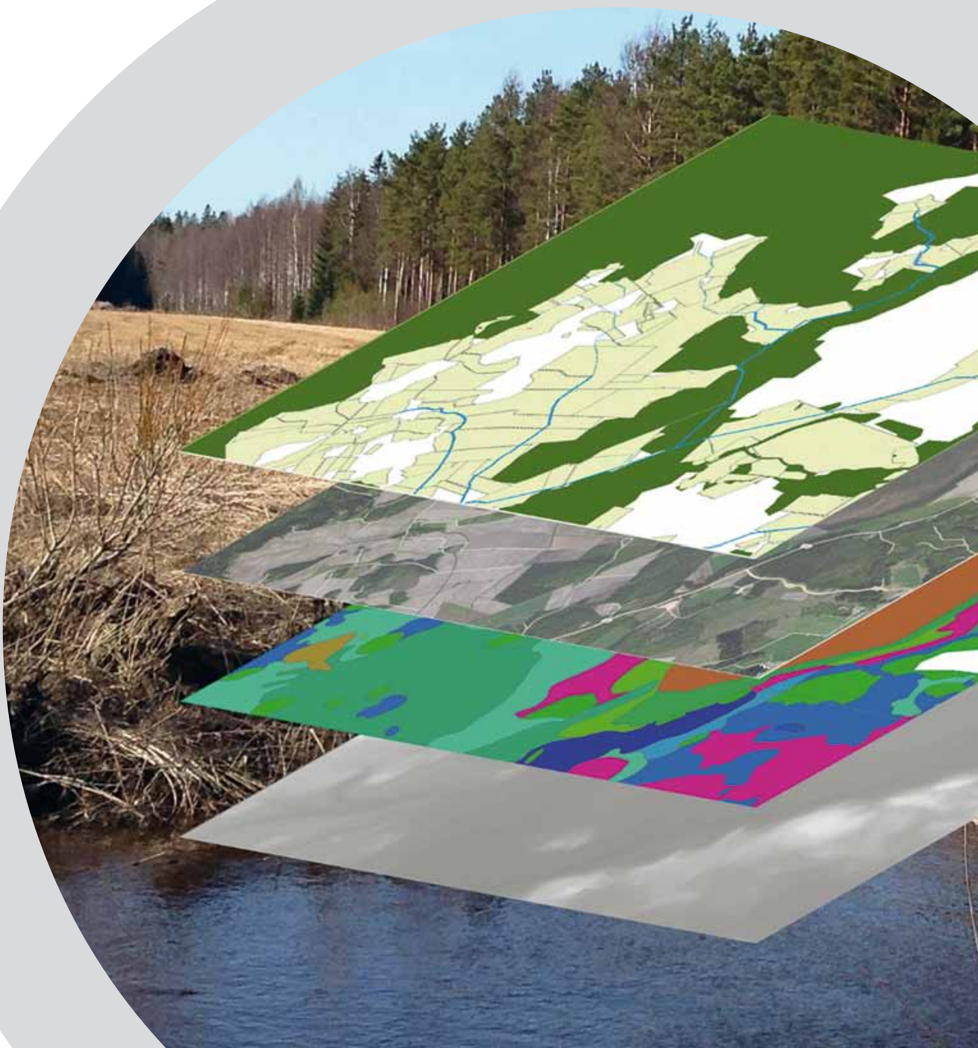
MTT SCIENCE 24

Cost-efficient nutrient load reduction in agriculture

A short-run perspective on reducing nitrogen and phosphorus in Finland

Doctoral Dissertation

Janne Helin



MTT

MTT SCIENCE 24

Cost-efficient nutrient load reduction in agriculture

A short-run perspective on reducing nitrogen and phosphorus in Finland

Doctoral Dissertation

Janne Helin

Academic Dissertation:
To be presented, with the permission of the Faculty of Agriculture and Forestry of the University of Helsinki, for public criticism in the Auditorium I at the City Centre Campus of the University of Helsinki (Siltavuorenpenger 1 A, Helsinki) on December 13th, 2013, at 12 o'clock.

ISBN 978-952-487-494-6 (Print)
ISBN 978-952-487-495-3 (Electronic)
ISSN 1798-1824 (Printed version)
ISSN 1798-1840 (Electronic version)
http://urn.fi/URN:ISBN:978-952-487-495-3
http://www.mtt.fi/mtttiede/pdf/mtttiede24.pdf

Distribution and sale
MTT Agrifood Research Finland,
Media and Information services,
FI-31600 Jokioinen, phone +358 29 5300 700,
e-mail julkaisut@mtt.fi
Printing year 2013
Cover picture Janne Helin
Printing house Juvenes Print – Suomen Yliopistopaino Oy

Supervisors:
Professor Heikki Lehtonen
MTT Agrifood Research Finland
Helsinki, Finland

Professor Markku Ollikainen
Department of Economics and Management
University of Helsinki
Helsinki, Finland

Pre-examiners:
Professor Ing-Marie Gren
Swedish University of Agricultural Sciences, Uppsala, Sweden

Professor Rauli Svento
University of Oulu, Oulu, Finland

Opponent:
Professor Eirik Romstad
Norwegian University of Life Sciences, Aas, Norway

Custos:
Professor Markku Ollikainen
University of Helsinki
Helsinki, Finland

Cost-efficient nutrient load reduction in agriculture
A short-run perspective on reducing nitrogen and phosphorus in Finland

Janne Helin
MTT Economic Research, Latokartanonkaari 9, FI-00790 Helsinki
janne.helin@mtt.fi

Abstract

This dissertation examines the economic efficiency of nutrient abatement measures in agriculture, focusing on the case of Finland. The thesis consists of an introductory article and four separate studies, which consider the nutrient abatement problem from different angles. Nutrient abatement was put on the environmental policy agenda decades ago because of the adverse impacts of eutrophication in surface waters, and remains there as water quality, in Europe and elsewhere, has not reached satisfactory levels. In particular, the Water Framework Directive of the European Union requires the member states to reach a good surface water status. In Finland, agriculture accounts for a major share of nutrient loads and could play an important role in achieving the water quality targets. However, the current environmental policy, relying on subsidies paid to farmers, has not met the abatement targets set for agriculture. As the European farm subsidy regime is shifting, new environmental policies could be adopted. For identifying the policies that would reach water protection targets, more information on costs-efficient measures is required.

The objective of this dissertation is to estimate nutrient abatement costs in agriculture and to rank the measures in terms of cost-efficiency. The focus is on the measures that have been considered in the agri-environmental support scheme. The dissertation relies on empirical numerical models that are based on microeconomic theory. Numerical representative farm models were developed for dairy and crop production. Abatement costs were found to fluctuate, depending on the target abatement levels, environmental and market conditions as well as production structure of agriculture. It is argued that reaching the national abatement targets, set by the government for 2015, is unlikely; extending the measures currently common in Finland is insufficient. For such a high abatement targets, policies should aim at increasing the share of green fallow, which can be effective in cutting down both nitrogen and phosphorus loads and increases biodiversity.

Key words:
water pollution, nutrient load, cost-efficiency, agriculture, abatement cost, manure management

Maatalouden ravinnekuormituksen kustannustehokas vähentäminen
– lyhyen aikavälin näkökulma typen ja fosforin vähentämiseen Suomessa

Janne Helin

MTT Taloustutkimus, Latokartanonkaari 9, 00790 Helsinki
janne.helin@mtt.fi

Tiivistelmä

Tämän väitöksen tavoitteena on arvioida ravinteiden vähentämisen kustannuksia suomen maataloudessa ja asettaa vähennyskeinot järjestykseen kustannustehokkuuden perusteella. Tarkastelun pääpaino on maatalouden ympäristötukijärjestelmän toimenpiteillä. Väitös nojaa mikrotalousteorian pohjalta laadittuihin empiiriseen aineistoon perustuviin laskentamalleihin.

Tutkimuksessa ohjelmoitiin maito ja peltoviljely -tuotantosuuntia edustavat tilamallit. Tuloksista käy ilmi että ravinnekuormituksen vähentämisen kustannukset vaihtelevat riippuen vähennystasoista, ympäristöllisistä tekijöistä, markkinatilanteesta ja tuotantosuunnista.

Kansallisten maataloudelle vuoteen 2015 asetettujen vähennystavoitteiden saavuttaminen nykyisten keinojen avulla vaikuttaa epätodennäköiseltä. Kun vähennystavoitteet on asetettu korkealle, niiden saavuttamiseksi tähtäävän politiikan kannattaisi lisätä viherkesantojen osuutta, sillä siten on tehokasta vähentää sekä typpeä että fosforia ja toisaalta suojella biodiversiteettiä.

Avainsanat:
vesistönkuormitus, maatalous, kustannustehokkuus, päästövähennyskustannukset, karjanlanta

Acknowledgements

There have been many people who have supported me during the long and rocky road of my dissertation work. First of all, I would like to thank my supervisors: Markku Ollikainen and Heikki Lehtonen. Without Heikki, this geography master student would have never written a phd dissertation in environmental economics.

This research was carried out mainly while I was working in MTT. The first project, The farm- and regional level optimisation of phosphorus cycles in Finnish animal husbandry, got me started with the phd studies and provided invaluable data and perspectives for the third paper of this dissertation. I would like to thank Pekka Huhtanen and Eila Turtola for supporting my modeling efforts. The second project, Cost-efficiency in water protection of Finnish agriculture improved my understanding of water quality models and I thank Sirkka Tattari for this. Marita Laukkanen taught me a lot about writing scientific publications. Sincere thanks also to the rest of the MTT economists, and especially the environmental economics team, whose former team leader Anni Huhtala gave me a plenty of feedback on several of my manuscripts. I gratefully acknowledge the financial support from Ministry of Agriculture and Forestry for the projects and the Finnish Cultural Foundation for providing the funding for the final push. I am very grateful for the support of my fellow students in environmental economics, especially Janne Artell.

While working with the dissertation, I spent a part of my time in the International Institute for Applied Systems Analysis (IIASA) in Austria. I would like to thank Tatiana Ermolieva as well as my other colleagues and YSSPrs for the valuable discussions on modeling issues. At the time of writing this, I work at the University of Helsinki. I would like to thank my colleagues for their patient attitude towards finalising my work.

I am very grateful to the pre-examiners of my dissertation, Rauli Svento and Ing-Marie Gren for your encouraging and valuable comments. Sincere thanks to Eirik Romstad for agreeing to be my opponent. I would like to thank my sister Katja Frösen for making this dissertation more readable. Finally, I wish to thank rest of my family and friends for listening to my worries on this seemingly never ending quest. Especially, I would like to thank my girl friend Anna Stygar for providing plenty of peer support and showing how a PhD-project is done efficiently.

Helsinki, November 2013

List of essays

This thesis is based on the original papers listed below, which are referred to in the text by their Roman numerals. These papers are reprinted with the kind permission of the publishers, while the studies III and IV are the author versions of the submitted manuscripts:

I Helin, J., Laukkanen, M. and Koikkalainen K. 2006. Abatement costs for agricultural nitrogen and phosphorus loads: a case study of crop farming in south-western Finland. Agricultural and Food Science 15.

II Helin, J. and Tattari, S. 2012. How much can be gained by optimizing nutrient abatement spatially - Cost-efficiency comparison of nonpoint arable loads from different Finnish watersheds. Food Economics 9.

III Helin, J. Reducing nutrient loads from dairy farms: a bioeconomic model with endogenous feeding and land use (manuscript accepted to be published in Agricultural Economics).

IV Helin, J., Hyytiäinen, K., Korpela, E.-L. and Kuussaari, M. Model for quantifying the synergies between farmland biodiversity conservation and water protection at catchment scale (manuscript accepted to be published in Journal of Environmental Management).

Contents

1 Introduction

1.1 Background

Eutrophication is a well-recognised environmental problem following from overloading of nitrogen and phosphorus[1]. It is associated with changes in the structure and functioning of marine ecosystems, reduced biodiversity, and reduced income from fishery, mariculture and tourism(Aertebjerg, 2001). Eutrophication problems are common in rivers, lakes, estuaries, and coastal oceans all around the world(Aertebjerg, 2001; Carpenter et al., 1998; Smith, Tilman, and Nekola, 1999).

Even though the full extent of utility losses caused by eutrophication has not yet been covered, the severity of the associated damages has motivated plenty of research on controlling eutrophication. While environmental science has established that nitrogen and phosphorus play a major role in causing the problems in water bodies, the interactions between the nutrients in water ecosystems are complex. Theories that support prioritising either of the nutrients are supported by empirical evidence, and reducing both of the nutrients has become the environmental norm in the developed world. Consequently, significant investments have been made to reduce the loads of nitrogen and phosphorus in the past decades. However, globally, loads are forecasted to increase further, and problems with eutrophication have not vanished from regions such as Europe where load trends have been somewhat decreasing (Aertebjerg, 2001; Drecht et al., 2009). While it is possible that the reductions of nitrogen and phosphorus as such have more complex interactions in the system than is currently understood, the paradigm in policy and science keeps on calling for further reductions in nutrient loads. However, a larger share of the measurable concentrations in rivers transporting the nutrients are coming from diffuse sources, so establishing control policies relies more on models developed to quantify various sources and their impacts.

For eutrophication, agriculture is an easily identifiable, but a nebulously quantifiable source. Nutrient balances have dramatically grown since the industrial and green revolutions[2], but the produc-

[1] At the time of publishing this PhD there were ≈ 180000 hits in Google scholar and 26704 hits in science direct for eutrophication.

[2] Industrial revolution lead to manufacturing fertilisers and green revolution spread these and other technological innovations to developing countries (Erisman et al., 2008; Gaud, 1968)

tion conditions of agriculture are very heterogeneous, making the connection between nutrient use and load ambiguous. The spatial and temporal variation in the complex processes detaching the nutrients from fields are affecting also the non-anthropogenic sources, hindering measurement of the anthropogenic load and the effect of abatement measures on the catchment or national scale. For example, separating the origins of nutrients from agriculture and forests at the river outlet is practically infeasible for estuary catchments. Furthermore, there are several variables such as the distribution of rainfall or the temperature range which are stochastic and cannot be practically managed.

Despite the lack of information at the relevant scale regarding many of the processes causing eutrophication, policies to control it exist, and new ones continue to be developed and implemented. For understanding and evaluating such efforts, economic models can be used in concert with environmental ones. Some environmental policy advice can be derived even from limited information. As the lack of direct observation of diffuse emissions implies relying on indirect emission control, the questions of what and who to target are policy-relevant.

lish effectiveness in both environmental and economic sense. Recognising the true complexity of nonpoint source pollution (NPS) control problem, means that the simplified analytical models offer little guidance without empirical knowledge of the magnitude or functional forms of the interacting processes. Nevertheless, modeling can provide guidance in decision-making, by, for example, in determining some general causalities or directing the empirical work in natural sciences towards economically viable management options. Hence this dissertation compiles information on several abatement methods for both macro-nutrients. The economic setting is a classical one, where a firm, in case of these four studies a farm enterprise, is described as a risk-neutral profit-maximising entity. Farms produce an external effect on the society by contributing to eutrophication and since the pollution share of each individual farm can not be verified, their joint nutrient load at the watershed level is described by static nonpoint production functions. The backbone of this dissertation is numerical optimisation modeling, which is used to approximate the complex processes that transport nutrients from agriculture to water. The methods of reducing the nutrient load and their effectiveness differ.

1.2 Subject matter

This dissertation examines the nutrient abatement strategies in agriculture. The common theme of the four separate studies lies in identifying the least-cost measures for both nitrogen and phosphorus abatement. Since empirical data is scarce, bioeconomic modeling is used to estab-

Study I estimates nutrient abatement costs given the cost-efficient measures available for a representative farm in South-Finland with uniform nutrient loads. It considers vegetated buffer strips, fertilisation reductions, tillage type, fallow and crop choices as potential abatement measures available for farmers. It demonstrates the effects of the Common Agriculture Policy (CAP) reform on the abatement costs.

Study II shows that spatially uniform nutrient load parametrisation can lead to overestimating nutrient load abatement costs under the Finnish conditions, since targeting of measures on the field areas with the largest load potential is not considered among abatement measure choices. It estimates abatement costs for two different types of watershed based on both homogeneous and heterogeneous description of farm land nutrient loads.

Study III indicates that fertiliser reductions and tillage choices precede feeding changes in cost-efficient nutrient abatement strategies. It estimates abatement costs for a representative dairy farm, given uniform agricultural land and endogenous manure composition.

Study IV demonstrates the synergy between cost-efficient nutrient load reductions and biodiversity conservation at a spatially heterogeneous watershed. The study demonstates that nutrient abatement by spatially targeted measures such as green fallow is supported by considering its biodiversity benefits.

2 Cost-efficient nutrient abatement

2.1 Theory

2.1.1 Eutrophication as a social problem

Eutrophication as a social problem can be analysed as an externality, an effect on the welfare of some third party not considered by the decision-maker. Baumol and Oates (1989) define externality by a condition "'*An externality is present whenever some individual's (say A's) utility or production relationships include real (that is non-monetary) variables, whose values are chosen by others (person's corporations, governments) without particular attention to the effects on A's welfare.*"' By definition, the problem of eutrophication then cannot be solved by markets as such. The economic agents are maximising their own utilities and ignoring the negative environmental effects on the utility of others. While the real world political processes to solve environmental problems are complex, involving multiple and conflicting interests, to simplify modeling of government intervention, one can postulate a social planner that would have some power over the economic agents and a goal to maximise the total social welfare, but lacking information on externalities. To implement policies for reaching the goal of maximum social welfare, the planner would require information on both the utility lost resulting from eutrophication, as well as utility lost by the agents adopting less nutrient polluting production. This dissertation focuses only on the latter problem.

2.1.2 Informational challenges

The first step in the classical pollution control problem is to identify the polluting agents (Shortle and Horan, 2001). In case of eutrophication, the task is not trivial. Nutrients are essential for all primary production and can be found in various quantities, not only in vulnerable water ecosystems, but in natural terrestrial sources as well as in different anthropogenic sources. This dissertation is limited to agricultural sources. Focusing on one, albeit on a significant sector, means that the results of this dissertation should be combined with information on other polluting sectors to establish cost-efficient abatement required for finding social optima. It can be shown that given any social optimum, it is necessary for all the polluters' marginal abatement costs to be equal (for example Baumol and Oates (1989)).

While agriculture as an economic sector can be identified as a source of nutrient pollution, the individual contributions of farms are far more difficult to quantify. Following the necessary condition of equal marginal costs, farms should be made to reduce their nutrient loads relative to their costs. Since farms are not identical in terms of their polluting loads or available measures and their impacts, the marginal costs are expected to be heterogeneous. This implies that setting equal nutrient abatement quantities for farms would not lead to a cost-efficient outcome. The burden of obtaining load and abatement information from each individual farm is great, since agricultural production is decentralised compared to many other production sectors. Nevertheless, many of the cost-efficient abatement policy schemes, such as input charges, rely on information on the private abatement costs (Shortle and Abler, 2001).

The literature on the classical pollution control problem under uncertainty shows that the information burden for the social planner can be decreased by designing environmental auctions (Adar and Griffin, 1976). Shortle and Dunn (1986) extend the policy analysis to nonpoint source pollution with uncertain knowledge about both weather and farm profits. However, in these studies, firms are assumed to have information on how their production choices affect the environment; an assumption which is ill-suited for dealing with the scientifically demanding quantification of the nutrient load processes at small agricultural enterprises. So while farms might be aware of their own control costs (adopting certain farming inputs), they likely are less informed on the load effects than the social planner.

Griffin and Bromley (1982) sidestep the asymmetric information between the planner and the farmers. Given the profit maximising behaviour and competitive markets, the joint supply of similar agricultural goods can be described by a single farm that represents the entire production. Griffin and Bromley (1982) call such representation a nonpoint production function. Heterogeneous production conditions characterising agricultural production can be accounted for in the function's arguments. However, if the policy is not directed towards management practices, also this approach requires the individual farmers to know the nonpoint production function for the least cost abatement choices.

There are informational challenges also for the social planner. According to Griffin and Bromley (1982), it is not necessary to monitor all inputs and outputs, just the ones related to pollution generation. However, while they claim that most production factors would not need to be considered in the nonpoint production function, Wossink, Lansink, and Struik (2001) argue that agriculture's production sets should be characterised as non-separable and heterogeneous. Both non-separability and heterogeneity add to the information required for establishing nonpoint source production functions. However, representing all the possible ecological and economic system linkages in a model is not feasible. Hence, the question remains: which properties of the nonpoint production function should be considered and which could be ignored when planning environmental policies?

Economic theory provides some selection criteria. For establishing a social optimal policy, it is necessary to consider the cost-efficient set of abatement measures (for example Baumol and Oates (1989)), which means that some aspects of production affecting nutrient loads will not need to be modeled. However, the cost-efficient set is ex ante unknown. Economic analysis including the cost-inefficient measures is required to separate the inferior measures from the cost-efficient ones. Furthermore, due to the uncertain benefits of abatement or uncertainties in abatement efficiency in other polluting sectors, determining the efficient abatement costs curve rather than just a single abatement target, is justified. These problems in outlining the extent of the required information can be illustrated with a simple set of three measures.

In Figure 1 the cost-efficient set for the lower abatement target is simply formed of only the lowest cost measure. Since the low abatement target a' can be achieved with a single measure, the remaining two measures do not need to be considered or analysed further. With the more pervasive environmental pollution problems, the reduction target is not as low compared to the effectiveness of the abatement measures. Such a situation, represented by a'', requires using more than the lowest cost abatement measure, since its reduction potential runs out before the societal target is reached. In Figure 1A the contribution of the other two measures depends on their relative costs. Even though either of the two more expensive measures has enough capacity to reach the abatement target, both should be used to abate cost-efficiently. As illustrated by the intersection of target level a'' and either of the joint marginal cost curves in Figure 1A, combining the measures non-exclusively allows reaching the target with lower costs than using single measure (intersection of a'' with either of single measure curves in Figure 1B). Hence, information on both costs and effectiveness are needed for all three measures. When multiple measures are needed, they can also interact to various degrees. Consider the vertical distance between the cost curves. In Figure 1B, the marginal costs of the second measure will begin from level c' when it is unaffected by the lowest cost abatement measure, and from c'' if the abatement processes were completely overlapping. For example, if measure 1 is reducing the emissions through the same mechanism as measure 2, the costs for adopting measure 2 will be higher (c''). Reduced efficiency due to overlapping measures may also imply that even more measures are required. Thus, interactions of the measures need to be understood for defining the set of cost-efficient measures.

All these concerns can be related to cost-efficient nutrient abatement in agriculture. Ranking the measures similar to Figure 1A requires a considerable amount of empirical information which is usually not available for all production conditions or abatement levels even for a single measure due to heterogeneity. Many of the conceived measures do not have fixed effects, but depend on heterogeneous production conditions such as soil structure or climate. Increasing marginal costs for a measure can stem from heterogeneity too; extending the measure from the most effective environment (for example, a crop area with the largest loads) to less suitable environment decreases the achieved abatement but not the cost. The effectiveness of measures can also be limited to a subset of environmental conditions, such as steep slopes, and the reduction potential of a single measure can be exhausted before reaching the tar-

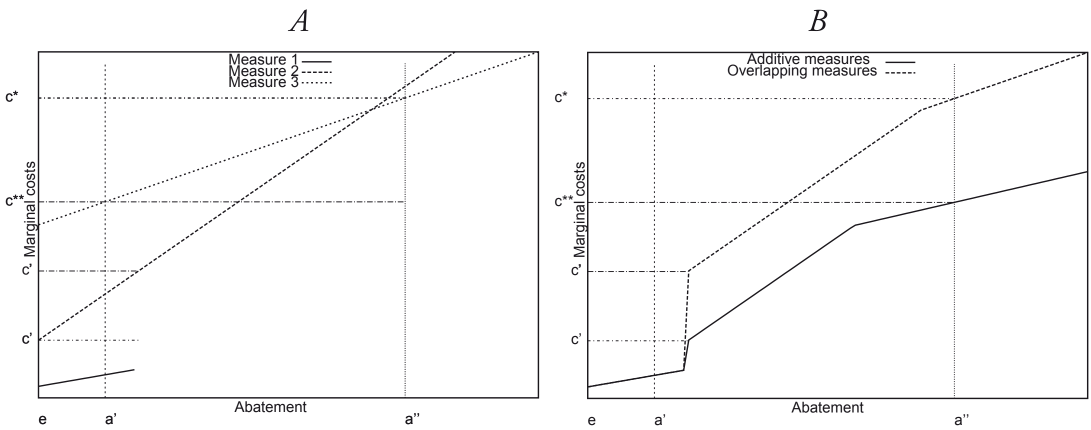


Figure 1: Three types of different abatement methods (A) and their cost-efficient combinations under different assumptions on the mutual exlusiveness of the measures (B). In figure A, the solid line illustrates a low-cost/low-potential measure, while the high-cost/high-potential and medium-cost/high-potential are represented with dotted and dashed lines, respectively. Vertical lines depict the two different abatement targets a' and a''. Baseline load is marked with e. The first abatement unit for measure 2 costs c' when there is no interaction between the effectiviness of measures 1 and 2 and c'' when the measures are overlapping. The abatement target a'' is more costly to reach when the measures are overlapping (c^*) than when they are additive (c^{**}).

get. The reduction targets are not fixed and are influenced by political decisions. Opportunity costs in foregone crop production are variable due to the stochasticity of weather and the related fluctuations in output prices. Concurring with Shortle and Horan (2001), there seems to be no universal ”easy” solution for reducing nonpoint source pollution.

Shortle and Horan (2001) point out that reducing the input tax/subsidy base to a subset of choices that are both relatively easy to observe and highly correlate with ambient impacts could address problems related with the moral hazard. So are there some more general factors that affect the farmers' abatement sets and that could be monitored? In previous nonpoint pollution literature, animal densities have been employed as convenient indicators of nonpoint source pollution (for example Letson et al. (1998) and Saam et al. (2005)). Farm production characteristics such as animal production densities will affect the feasible set of abatement choices (Schnitkey and Miranda, 1993; Innes, 2000; Feinerman, Bosch, and Pease, 2004; Bosch, Wolfe, and Knowlton, 2006). As environmental production conditions are heterogeneous, abatement can be achieved by relocating more nutrient-intensive land use to environmental conditions less prone to nonpoint source pollution (Braden et al., 1989). However, when fixed capital investments, such as animal housing, are part of production, relocation could be costly compared to other measures. Consequently, the nonpoint production function and the abatement set in animal farming cover more or at least different possibilities than mere crop farming. Ignoring animal production could lead to considering only a subset of the management practices, and thus incentivising inferior abatement measures (i.e. similar to leaving out one of the measures in Figure 1 when target is a''). However, without an empirical analysis of the abatement sets, the cost-efficiency of measures not-involving animals (similar to measure 1 and target a' in Figure 1) cannot be ruled out either.

2.1.3 The Model

To formally compare the optimal abatement on a farm with and without animals, suppose that there exists a watershed for which the social planner considers a nutrient load target level $\hat{E}$ for agriculture. The current load is composed of contributions of i farms, which the planner cannot monitor without prohibitive expenses, but can estimate the load e_i from each farm based on some normal weather conditions and known farm characteristics including nutrient $N_{i,j,s}$ and land $X_{i,j,s}$ use. Let j be an index of the crop type and s the index of land characteristics and management practices[1].

$$E = \sum_i e_i(N_{i,j,s}, X_{i,j,s}) \qquad (2.1)$$

The load from the farms adds to the total load E, and to have a social problem $E > \hat{E}$. Defining the difference $E - \hat{E} \equiv A$, there is a total social abatement target A.

[1] It may be helpful to think of variables in terms of annual sums i.e. total load $\hat{E}$ kilos per year, area $X_{i,j,s}$ in hectares, and fertilisation $N_{i,j,s}$ in kilos per hectar.

For cost-efficient A, the necessary condition is that those farms that abate, do so cost-efficiently. This is equal to reaching the abatement target with the combination of measures having the lowest costs (as in Figure 1). For farm i, the cost-efficient abatement is $a_i \equiv e_i - \hat{e}_i$, which maximises the constrained farm profits denoted by $\hat{\pi}_i^*$. Thus, the abatement costs $C(\hat{a}_i)$ for the farm are defined by

$$C_i(\hat{a}_i) = \pi_i^*(N_{i,j,s}, X_{i,j,s}) - \hat{\pi}_i^*(\hat{N}_{i,j,s}, \hat{X}_{i,j,s}) \qquad (2.2)$$

where π_i^* is the optimal profit without the load constraint[2]. Reaching A cost-efficiently requires marginal abatement costs, $\partial C_i(a_i)/\partial a_i$, for farms to be equal[3]. Otherwise, reallocating abatement between the farms could be used to decrease the total costs, $\sum_i C_i(\hat{a}_i)$. This condition for the socially optimal solution assumes that the units of nutrient load from different sources are perfect substitutes[4].

[2] It is also possible to formulate the problem as a cost-minimisation problem (the dual of constrained profit-maximisation problem), but the maximisation formulation follows the approach taken in the studies I-IV

[3] For proof (not including existence of equilibrium), see Baumol and Oates (1989).

[4] A theoretically precise formulation would require establishing transport functions for capturing the effect of various hydrological processes, since the fate of nutrients from different sources is not identical between the farms. However, this would unnecessarily complicate this pedagogical presentation with elements that could be accounted in e_i by defining the set s to contain the required information such as location of the farms. Generally, the freshwater systems use and lose some of the nutrients, and only part of the total load from land flows to estuaries. This share could be based on location i.e.

Given competitive input and output markets, farmers are not able to influence prices. It is assumed that farmers aim at maximising the profits and are not motivated by other factors when taking decisiona affecting the expected nutrient loads. Under these assumptions, the economic abatement problem of single farmer can be generalised to a nonpoint production problem of the whole watershed by specifying the yield and load functions according to the watershed's properties.

Crop production

Consider farm i which produces only crops. Notwithstanding any prior regulation, the private profit maximising level for the representative farmer (dropping subscript i from arguments)

$$\hat{\pi}_i^*(\hat{N}_{j,s}, \hat{X}_{j,s}) = \sum_j \sum_s (p_j y_{j,s}(N_{j,s}) - p_N N_{j,s} - c_{j,s}) X_{j,s} \quad (2.3)$$

s.t.

$$\sum_j X_{j,s} r_{j,s,l} \leq \bar{X}_{s,l} \; \forall \; s, l \quad (2.4)$$

$$\sum_j \sum_s z_{j,s}(N_{j,s}) X_{j,s} \leq \hat{e} \quad (2.5)$$

$$N_{j,s}, X_{j,s} \geq 0 \quad (2.6)$$

For the farmer, nutrient vector $N_{j,s}$ (consists of both synthetic fertiliser and manure) and $X_{j,s}$, the land use vector, are endogenous variables which determine the expected nutrient load $z_{j,s}$ and yield $y_{j,s}$ per area unit. Output prices are given by p_j. Manure and the price of its nutrients, as well as the synthetic fertilisation, are exogenous (price vector p_N). Costs of farming per area unit, $c_{j,s}$, depend on the crop type, land characteristics and management practices. The distribution of fixed land characteristics at the watershed defines $\bar{X}_{s,l}$ for the representative farm. Set l consists of limits to land use, including the total area constraint. Parameter $r_{j,s,l}$ defines the limitations in production technology and land characteristics. For example, certain crops might be suitable only for a part of the field area due to different soil types. The target nutrient load $\hat{e}$ for the representative farm is determined by the social planner and is proportional to $\hat{E}$ so that $\hat{E}/\sum_{i,j,s} X_{i,j,s} = \hat{e}/\sum_s \bar{X}_{s,l}$. Thus, the expected load (and abatement costs) of the representative farm can be scaled up to the watershed level. Solving for $N_{j,s}$ and $X_{j,s}$ without the (binding) constraint in Equation 2.5 will give the baseline private optimal profit π_i^*.

Karush-Kuhn-Tucker conditions (KKT) for the optimal solution are

$$\frac{\partial L}{\partial X_{j,s}} = p_j y(N_{j,s}) - p_N N_{j,s} - c_{j,s} - \lambda_1 z_{j,s}(N_{j,s}) - \mu_l r_{j,l,s} \leq 0 \quad (= 0 \; if \; X_{j,s} > 0) \quad (2.7)$$

the distance from the river outlet. However, euthrophication of both fresh water bodies and seas represents an externality. Therefore, the effect of load on both fresh and coastal water nutrient concentrations should be traced.

$$\frac{\partial L}{\partial N_{j,s}} = p_j \frac{\partial y_{j,s}}{\partial N_{j,s}} X_{j,s} - p_N$$
$$-\lambda_1 \frac{\partial z_{j,s}(N_{j,s})}{\partial N_{j,s}} X_{j,s} \leq 0 \quad (2.8)$$
$$(= 0 \; if \; N_{j,s} > 0)$$

$$\frac{\partial L}{\partial \mu_l} = \bar{X}_{s,l} - \sum_j X_{j,s} r_{j,s,l} \geq 0 \quad (2.9)$$
$$(= 0 \; if \; \mu_l > 0)$$

$$\frac{\partial L}{\partial \lambda_1} = \hat{e} - \sum_j \sum_s z_{j,s}(N_{j,s}) X_{j,s} \geq 0$$
$$(= 0 \; if \; \lambda_1 > 0) \quad (2.10)$$

which have to hold for all j,s and l. The marginal benefits from the optimal land allocation equal the marginal costs. The productivity of land is influenced by nutrient use. The constraint in Equation 2.4 forms a shadow price (μ_l) for the short run land availability and technical farming limitations for each binding limitation. Without production heterogeneity in $X_{j,s}$, a single crop/technology combination dominates until a binding resource limit $r_{j,s,l}$ is reached. Hence, the shadow price is determined by the difference between the most profitable and the next most profitable combination. Under production heterogeneity, the yield response function is conditional on set s. For example, one might define $s = 1$ as sandy soil and $s = 2$ as clay soil and give different parameters in $y_{j,s}(N_{j,s})$.

Without government intervention there are no limitations on the nutrient load, e, and no effect on the farmer's profits. Capping the load to $\hat{e}$ creates a shadow price ($\lambda_1 \neq 0$) for the difference between socially allowed and privately expected nutrient load. Private optimal fertilisation maximises the profit from a hectare of land. Assuming concave yield leads to decreasing marginal returns to nutrient use. Besides the shape of the yield function, the optimal solution is determined by the crop output and nutrient input prices. Without further assumptions, only the lowest cost nutrient source is used, and the mixed use of synthetic fertilisers and manure at the watershed level is not optimal. Considering the external effect of the nutrient load would decrease the optimal fertilisation.

Within this frame, the abatement set consists of (joint) production choices influencing e. When e is an increasing function of nutrient use, ($N_{j,s}$), decreasing the nutrient input quantities leads to abatement. Decreasing the amount of farm land decreases agricultural load, but since land does not truly vanish from the watershed, this option is better represented by a "back stop" land use class such as fallow or forestry for j and holding the total land area constant. Furthermore, it can be postulated that some crop j and farming conditions and technologies in s are leading to a larger expected load than others. Choosing $X_{j,s}$ within the constraints can be used for abatement. Thus, it is possible to model measures such as a direct tillage or an extended vegetation cover period.

Livestock production

For a simple representation considering livestock in addition to crops, assume that the k farmer has animals and decides the stock size Q based on the fixed exogenous net return p_q from each animal (similar to Schnitkey and Miranda (1993)). In the short run, farmer's capital such as animal sheds and machinery are limited to a fixed capacity $\bar{Q}$. Furthermore, assume that as a byproduct of animals ϵ of manure nutrients is excreted and needs to be disposed of annually. Compared to synthetic fertilisation, the nutrients in manure (a subset of N identified with superscript Q in Equation 2.13) are not in a compact form and would normally cost more to haul and apply. Hence, the distance between the animal shelter and fields becomes a significant factor in the nutrient allocation problem. Separating this distance from other field characteristics and denoting it by d helps to illustrate how the optimal nutrient allocation changes. The private profit maximising problem of a representative farm with animals:

$$\begin{aligned}\hat{\pi}_k^*(N_{j,s,d}, X_{j,s,d}, Q) = p_q Q \\ + \sum_j \sum_s \sum_d (p_y y_{j,s,d}(N_{j,s,d}) \\ - p_{N,d} N_{j,s,d} - c_{j,s,d}) X_{j,s,d}\end{aligned} \tag{2.11}$$

s.t.

$$Q \leq \bar{Q} \tag{2.12}$$

$$\sum_j \sum_s \sum_d (N^Q_{j,s,d}) X_{j,s,d} = \epsilon Q \tag{2.13}$$

$$\sum_j X_{j,s,d} r_{j,s,l,d} \leq \bar{X}_{s,l,d} \tag{2.14}$$

$$\sum_j \sum_s \sum_d z_{j,s,d}(N_{j,s,d}) X_{j,s,d} \leq \hat{e} \tag{2.15}$$

$$N_{j,s,d}, X_{j,s,d}, Q \geq 0 \tag{2.16}$$

The prices for synthetic fertilisation and crops are as above. The price of manure nutrients is determined by the cost of transporting them and depends on the fixed distance between the farm and its fields. Thus, the price parameter $p_{N,d}$ depends on the nutrient origin. For nutrients from animal production ($N^Q_{s,l,d}$), their price is increasing with the transport distance. The field area is distributed to $\bar{X}_{s,l,d}$. KKT-conditions for the optimal solution are:

$$\begin{aligned}\frac{\partial L}{\partial X_{j,s,d}} = p_j y_{j,s,d}(N_{j,s,d}) - p_{N,d}(N_{j,s,d}) - c_{j,s,d} \\ -\lambda_1 z_{j,s,d}(N_{j,s,d}) - \mu_l r_{j,s,l,d} - \lambda_3 N^Q_{j,s,d} \leq 0 \\ (= 0 \; if \; X_{j,s,d} > 0)\end{aligned} \tag{2.17}$$

$$\begin{aligned}\frac{\partial L}{\partial N_{j,s,d}} = p_j \frac{\partial y_{j,s,d}}{\partial N_{j,s,d}} X_{j,s,d} - p_{N,d} \\ -\lambda_3 X_{j,s,d} - \lambda_1 \frac{\partial z_{j,s,d}(N_{j,s,d})}{\partial N_{j,s,d}} X_{j,s,d} \leq 0 \\ (= 0 \; if \; N_{j,s,d} > 0)\end{aligned} \tag{2.18}$$

$$\begin{aligned}\frac{\partial L}{\partial Q} = p_q - \lambda_2 + \lambda_3 \epsilon \leq 0 \\ (= 0 \; if \; Q > 0)\end{aligned} \tag{2.19}$$

$$\frac{\partial L}{\partial \mu_l} = \bar{X}_{s,l,d} - \sum_j X_{j,s,d} r_{j,s,l,d} \geq 0 \quad (= 0 \; if \; \mu_l > 0) \tag{2.20}$$

$$\frac{\partial L}{\partial \lambda_2} = \bar{Q} - Q \geq 0 \quad (= 0 \; if \; \lambda_2 > 0) \tag{2.21}$$

$$\frac{\partial L}{\partial \lambda_3} = \epsilon Q - \sum_j \sum_s \sum_d (N^Q_{j,s,d}) X_{j,s,d} = 0 \tag{2.22}$$

$$\frac{\partial L}{\partial \lambda_1} = \hat{e} - \sum_j \sum_s \sum_d z_{j,s,d}(N_{k,j,d}) X_{j,s,d} \geq 0 (= 0 \; if \; \lambda_1 > 0) \tag{2.23}$$

For a holding to classify as an animal farm, $Q > 0$. Thus, Equation 2.19 holds as an equality. In a case in which animal capacity is not constraining production, $\lambda_2 = 0$ (Equation 2.21), both increasing the price of the animal product and its contribution to nutrients increase the optimal quantity of animals. Furthermore, the shadow price of manure is determined by profits gained in animal production $\lambda_3 = -p_q/\epsilon$.

The optimal fertilisation in Equation 2.18 is affected by the distances and the animals (through λ_3). The cost of manure nutrient application, $p_{N,d}$, is increasing in d, but synthetic fertiliser use is still determined by solely the market price. The relative prices of transport for manure and the market price for synthetic fertiliser per kilogramme of nutrient determine which one will be used. It is theoretically possible that transport costs are lower than fertiliser prices for each d and thus manure would be used at every distance. Given the limited animal capacity of the farm, the simultaneous use of synthetic fertiliser on the fields further away can still be optimal, since manure nutrients would not be sufficient to reach the optimal fertilisation levels. In the opposite case, where the fertiliser prices are lower than the transport prices per nutrient, manure needs to be disposed of following Equation 2.13, even though profits are decreased. The lowest costs are given by the smallest distance, so for a strictly non-decreasing $y_{j,s,d}$, all manure would end up to the closest field. Between these two limiting cases regarding $p_{N,d}$, manure hauling costs increase with the distance until the fertiliser price (or animal capacity) is reached. At this distance, manure nutrient use equals the quantities that would be used on a crop farm without manure. On the fields beyond this distance, optimal fertilisation is similar to the solution on the crop farm. Within this distance, the optimal manure use can exceed the optimal nutrient quantities from the perspective of the yield response. For a yield function with negative marginal yield (e.g. the commonly used quadratic yield), a marginal reduction in the yield revenue increases the profitable manure transport distance.

As can be seen from Equation 2.17, the optimal land use is influenced by nutrient application costs, which now depend on the distance from the farm to the fields. Since this distance is fixed, there is a shadow price for the inability to relocate the fields closer to the farm. When the constraint in Equation 2.13 is binding, the limited manure supply can reduce the profits from farming the land, if, for that distance, it would have been cheaper to use manure instead of synthetic fertiliser. It is possible that also other farming costs increase with the distance, and $c_{j,s,d}$ can affect not only the overall profit from more remote fields, but also the optimal crop or farming technology choices.

For the abatement set, reduction in Q is a measure not available at the crop farm. Relative efficiency compared to other abatement measures requires further assumptions or empirical knowledge. Given the nutrient load functions suggested by some studies (Simmelsgaard and Djurhuus, 1998; Koopmans et al., 2002), exceeding the biological uptake norms of the crop leads to a rapid increase in the load. Thus, the circumstances leading to excess nutrients at close fields would provide a potential reduction target, which can be obtained at the costs of manure transport. On the other hand, marginal transport costs can still be higher than the value of the yield loss from an equivalent load reduction by other means. Further abatement options at the animal farm stem from the different load propensity of manure and synthetic fertilisation. Generally, manure nutrients are regarded as prone to runoff compared to synthetic fertilisers, but technologies to reduce manure losses are implemented world-wide.

Synthesis at the watershed

From the necessary condition for the cost-efficient optimal solution, the marginal cost of the farms need to be equal,

$$\frac{\partial C_i(a_i)}{\partial a_i} = \frac{\partial C_k(a_k)}{\partial a_k} \forall i \qquad (2.24)$$

.

When $C_i \neq C_k$, the social planner's abatement problem is not characterised completely by the solution to the single representative farm (either i or k). Without stronger assumptions, it cannot be concluded that more abatement should be targeted towards either of the farms. In the special case where the socially pursued levels of abatement can be reached with the least-cost at $C_k(a_k) = 0$, less complicated models in terms of the farm production economy would be required, allowing research efforts to focus on many of the other problematic issues in the nonpoint source pollution control problem. Since farms with both animals and fields share farming management technology with farms growing only crops, satisfying conditions for only the crop farms to abate in a cost-efficient solution, seems unlikely.

In the mixed case (both i and k farms abate), animal density becomes a factor affecting the abatement decisions. In case of homogeneous land, the inter-farm manure transport could be expected to occur only if crop farms have some fields closer to animals farms than some fields owned by animal farms. To capture such interaction, a representative single farm nonpoint pollution function is inadequate from theoretical grounds.

If animal related measures are superior to measures available for all farms (in the sense of measure a' in Figure 1), single representative animal farm contains the elements for a nonpoint production function of the whole watershed. However, for many conceivable abatement options specific to animal farms, the distance to fields and animal capacity, which are farm specific parameters, would need to be accounted. For example reducing surface loading risk of manure by injecting it, would increase the cost and thus reduce the optimal transport distance. So while one farm can theoretically represent the nonpoint production of a watershed, the information required from each farm for the cost-efficient solution increases. Thus, the formulation of the nonpoint production function approaches the level of complexity of modeling the watershed farm-by-farm.

Even with a very simple illustration of animal farming, it is possible to see that the definition of cost-efficiency at watershed level is more complex to derive than the nonpoint source pollution problem based on single representative farm. Given these basic analytical models of representative farms, the abatement at joint animal and field operations could be more or less effective than abatement at farms without animals. More analysis based on empirical data is necessary. Furthermore, the simple formulation of models above disregards several issues potentially affecting cost-efficient abatement.

Innes (2000) shows that in animal production, the density of animals is a factor influencing the nutrient abatement. From the social point of view, animal production operations can be excessively concentrated and lead to larger damages from externalities than from a less dense production structure. While Innes (2000) emphasizes both leakages and spills from manure storage facility, the model presented as part of this dissertation extends the simple model presented above with a choice of covering the manure storage. Compared to Innes (2000) the links between feeding and fields are emphasized. For example, the feeding of animals affects the optimal field allocation. Endogenous feeding is covered in study III, which simultaneously analyses manure nutrient ratios from the perspective of crop growth. Farmers are able to choose from different synthetic fertiliser N:P ratios suitable for a variety of crop needs, but the N:P ratio of manure is determined by feeding and volatilisation and hence not typically matching the optimum ratio for crop growth. In short-run nitrogen has a more immediate impact on yield than phosphorus, thus in contrast to Schnitkey and Miranda (1993), nitrogen content in manure could play a larger role in the manure allocation problem. In contrast to both Innes (2000) and Schnitkey and Miranda (1993) manure nutrient content is determined by an endogenous diet. In addition to the diet several nutrient interactions are modeled for the other abatement methods covered in studies I-IV.

2.2 Empirical literature

2.2.1 Effectiveness of agricultural measures

Empirical information on the relative impacts of different abatement measures is needed for defining nonpoint source abatement sets. This section aims to provide a short summary of potential measures for nutrient abatement in Northern-Europe. Cherry et al. (2008) identify potential diffuse pollution control measures for UK, of which the following 15 (Table 2.2.1) are studied in this dissertation.

Land use and soil management

Transport of the nutrients from the fields depends not only on their concentration in arable land, but also on the structure of the land and field surface. These variables in turn are affected by the crop type and cultivation.

Ekholm et al. (2005) compare grassland with cereal farming. Irrespective of the soil type, total algae available P load[5] from grassland is higher than the load from cereals, although the difference is more evident on fine sand soils. Puustinen, Koskiaho, and Peltonen (2005) measure the load from grass ley compared to cereal cultivation. According to Uusi-Kämppä and Jauhiainen (2010), the dissolved reactive phosphorus (DRP) load is higher and particulate phosphorus (PP) lower for pasture than for autumn ploughed spring wheat. Kutra and Aksomaitiene (2003) measure the drainage nutrient concentrations of different crop rotations including winter and spring grain crops, sugar beet and perennial grasses in Norway. The nitrogen load from cereals and sugar beet is several times higher than from the perennial grasses, but the grasses have the highest P leaching.

[5]Phosphorus load can stem from different processes. Typically water transports phosphorus to water with eroded land particles or dissolves it from the soil. Part of the eroded material will be sedimented before the phosphorus has time to react and to be utilised by algae. Therefore this particle phosphorus share is regarded less bioavailable than the phosphorus already dissolved. McDowell (2012)

Koskiaho, Kivisaari, et al. (2002) show that harrowing reduces erosion and nitrogen load compared to ploughing, while only a minor difference in total P (TP) load are observed at clayey fields in Southern-Finland. In Puustinen, Koskiaho, and Peltonen (2005), ploughing and cultivation treatments in Autumn produce the highest particle P concentrations at slope clayey fields, whereas P concentration is 31% lower for no till treatment. According to Turtola, Alakukku, et al. (2007) even on flat clay soil, adopting no tillage reduces erosion by 48-12 %. Shallow autumn stubble cultivation does not reduce erosion significantly compared to mouldboard ploughing. Puustinen, Koskiaho, and Peltonen (2005) report higher DRP concentrations for conservation tillage treatment than for autumn ploughed winter wheat.

The reduction of erosion by changing tillage (timing and method) has been studied widely also elsewhere(Cannell, 1985; Holland, 2004; Soane et al., 2012). According to a study in US by Zeimen et al. (2006) the chisel/disk cultivation results in sediment losses two times higher compared with the no-till, but the soluble P losses are 3.0 and 2.1 times higher for the no-till technology. In Scandinavia, the erosion and leaching of nitrogen seems to increase with more intensive cultivation (Rasmussen, 1999).

Establishing perennial grass cover at the edge of non-cultivated field seems to carry some potential for reducing the nutrient load from the rest of field. Puustinen, Koskiaho, and Peltonen (2005) find that a 14-meter timothy grass buffer zone reduces the flow-weighted PP concentration by 74 % and increases the DRP

Measure	Study
Convert arable land to extensive grassland	I,II,III,IV
Cultivate land for crop establishment in spring rather than in autumn	I,II,III,IV
Adopt minimal cultivation systems	I,II,III
Establish in-field grass buffer strips	II,IV
Reduce overall stocking rates on livestock farms	III
Reduce dietary N and P intakes	III
Use a fertilizer recommendation system	I,II,III,IV
Integrate fertiliser and manure nutrient supply	III
Reduce fertiliser application rates	I,II,III,IV
Do not apply P fertilisers to high P index soils or other high-risk areas	II,IV
Increase the capacity of farm manure (slurry) stores	III
Minimise the volume of dirty water produced	III
Do not apply manure to high-risk areas	III
Incorporate manure into the soil	III
Establish riparian buffer strips	I,II,III,IV
Establish cover crops in the autumn	
Allow field drainage systems to deteriorate	
Cultivate compacted tillage soils	
Cultivate and drill across the slope	
Leave autumn seedbeds rough	
Avoid tramlines over winter	
Loosen compacted soil layers in grassland fields	
Maintain and enhance soil organic matter levels	
Reduce the length of the grazing day or grazing season	
Adopt phase feeding of livestock	
Adopt batch storage of manure	
Compost solid manure	
Change from slurry to a solid manure handling system	
Site solid manure heaps away from watercourses and field drains	
Site solid manure heaps on concrete and collect the effluent	
Do not spread manure to fields at high-risk times	
Transport manure to neighbouring farms	
Manure treatment including incineration of poultry litter	
Fence off rivers and streams from livestock	
Construct bridges for livestock crossing rivers and streams	
Re-site gateways away from high-risk areas	
Establish new hedges	
Establish and maintain artificial (constructed) wetlands	

Adopted from Cherry et al. (2008)

only marginally. In Uusi-Kämppä (2005), the mean annual TP loss from 10-meter wide grass buffer and natural vegetation buffer plots is 40% lower than the TP loss from non-buffer plots. However, the loss of DRP was 70% higher from the natural vegetation buffer plot than from the other plots. Uusi-Kämppä and Jauhiainen (2010) show that buffer zones can reduce the TP and DRP loads also from grazing and direct till field areas.

Mander, Hayakawa, and Kuusemets (2005) summarize several vegetated filter strips studies and present equations for nutrient removal. In US, Daniels and Gilliam (1996) observe that a 6-meter wide vegetated filter strip reduces the total phosphorus and nitrogen loads approximately by 50%, but also notes an increase in the soluble P concentration. In Norway, Syversen (2005) report average removal efficiencies of 60–89%, 37–81% and 81–91% for phosphorus, nitrogen and particles, respectively

Empirical evidence on nutrient load reduction by tramline direction is scarce. In Finland, cross-plowing results in halving the flow-weighted nutrient concentration on a sloped clayey winter wheat field (Puustinen, Koskiaho, and Peltonen, 2005). According to Withers et al. (2006), tramlines aligned with slope increase runoff and phosphorus load on ploughed fields, but not on fields with less intensive cultivation.

Nutrient management

Turtola and Kemppainen (1998) compare nutrient loads from non-fertilised grass with manure-fertilised grass or synthetic fertilized grass. For nitrogen, the loads are 1.8 to 14.7 times higher and for phosphorus 5.4 to 74 times higher than on non-fertilised grass, depending on the source, timing and method of nutrient input. Simmelsgaard and Djurhuus (1998) estimate the relationship between nitrogen fertilisation and load from Danish empirical data. In Finland, Salo and Turtola (2006) show that the nitrogen balance can be used to predict the nitrogen load of poorly managed farmland, but that on fields under good agricultural practice, nitrogen balance alone is not sufficient to explain the load variation. In Norway, Kutra and Aksomaitiene (2003) demonstrate that high nitrogen fertilisation leads to high load. Ekholm et al. (2005) estimate phosphorus reduction from decreasing the phosphorus balance through the effect of the balance on soil test phosphorus.

Animal management

Around 70% of world's agricultural area is used for producing animal fodder (Steinfeld, 2006). Thus, the nonpoint source loads are very much influenced by the production decisions of animal farms. However, some abatement measures are specific to animal husbandry. The most direct of these is limiting the excretion straight to water bodies, which can occur when grazing of cattle is free. Also on land, grazing leaves manure susceptible to runoff, so limiting grazing in specific vulnerable areas or in general has been considered as an abatement measure (McGechan and Topp, 2004; Kurz, O'Reilly, and Tunney, 2006; Butler et al., 2008). However, the nutrient load of manure is not limited to grazing. Confining animals to enclosed areas makes it necessary to manage and in most cases store the manure in one way or the other. Moreover, such a manure management chain is prone to depositing nutrients to water bodies. Hence, manure management is generally legislated to ban practices that would lead to the most severe nutrient loads. For example, manure cannot be disposed of by dumping it to the nearby ditch or lake. Other high load risk areas are also regarded as unsuitable for manure application, although the impact of this kind of abatement measure has not been extensively studied.

A more substantial body of research can be found on the load impact of timing and methods of manure dispersal on agricultural land, since manure has to be disposed of somewhere, and in agriculture it can substitute for other inputs. In Finland, Ekholm et al. (2005) compare grassland with no surface manure application with broadcasting of manure to surface. The surface broadcasting of manure doubles the total algae available P load. This impact follows from the DRP component, while the PP share is virtually unaffected. However, the Finnish law regulates the surface broadcasting to be consequently followed by incorporation within 24 hours. Also the manure application periods are limited to the snow free period between 15.4.-15.10. No peer-reviewed sources are available to quantify the average abatement pursued with this legislation. A load reduction can be achieved by drilling the manure into the soil or by hose trailer application. Ball Coelho, Roy, and Bruin (2006) find that the injection of slurry improves the N efficiency, whereas excess surface application leads to elevated N load. According to Daverede et al. (2004), the injection of swine slurry reduces the total phosphorus by 94% compared to the surface application. Another US study shows that liquid swine manure increases the NO3–N losses with tile flows by 53% compared to urea ammonium nitrate fertilisation (Bakhsh, Kanwar, and Karlen, 2005).

The manure storages themselves can form significant nutrient load sources. Containing heaps of manure in exposed conditions can lead to nutrient losses. For this reason, the manure storage conditions are regulated in Europe by the Nitrate Directive (91/676/EEC). Lack of data on the impacts of the existing regulation hinders evaluations on manure storage guidelines, which require a standard manure storage capacity per cattle, composting or other treatments.

Nutrient abatement by changing the diet of animals is widely supported by the literature, although the efforts have rather focused on the modeling and nutrient balances than on trials with load monitoring. Rotz et al. (1999) demonstrate with the DAFOSYM model used in the US that shifting away from feeding the dairy cow herd solely on soybean protein to less rumen degradable protein diet leads to a reduction in N leaching by 1kg/ha. The same model shows the effect of reducing the dietary P supplement on farm P balance (Ghebremichael and Watzin, 2011). In Finland, Huhtanen, Nousiainen, and Turtola (2011) show that switching to sugar-beet pulp soybean meal supplements from least-cost ration formulation improves P efficiency in milk production and decreases the modeled nutrient balance. With addition of phytase to the pig diet, the manure phosphorus content can be decreased (Cromwell et al., 1993).

A majority of the identified measures that are not modeled in this dissertation relate to animal husbandry farms. While Study III considers animal husbandry, many of the potential measures lack empirical data to quantify their impact on the load or, in some cases, the costs. For example, a recent review by Kay, Edwards, and Foulger (2009) did not report studies on the effectiveness of nutrient-specific measures available for implementation in the UK, such as preventing runoff from in-field manure heaps or not applying organic fertilisers when the soil is saturated. There is generally very little data on the efficiency of some specific measures, such as preventing manure application within 10 m of a surface water or within 50 m of a borehole under Finnish conditions.

Limiting the animal contact or proximity to the water reduces the phosphorus load (Rao et al., 2009). In North Carolina, fencing off streams from livestock reduces organic nitrogen, Kjeldahl nitrogen [6] and TP by 33%, 78%, and 76%, respectively. Similarly, arranging cattle to not to drink from streams can reduce total phosphorus concentrations by 54%, whilst total nitrogen concentrations can be reduced by 81% (Sheffield et al., 1997). The effect of constructing bridges for livestock to cross rivers and streams seems smaller since the time to excrete directly to water body is likely smaller in cattle transport than grazing. Reducing grazing decreases the time animals excrete on pastures, and hence reduces the nutrient load potential Kurz, O'Reilly, and Tunney (2006). However, for nitrogen, the more significant nitrate load can occur from the renovation of pastures (Saarijärvi et al., 2004). In Uusi-Kämppä and Jauhiainen (2010), it is shown that the DRP load can be greater from grazing than from ploughed cereals. An experiment by Turtola and Kemppainen (1998) shows that manure application in autumn and winter increases the nutrient loads from grass ley.

[6] (Total) Kjeldahl nitrogen is the sum of organic nitrogen, ammonia (NH_3), and ammonium (NH_4^+)

Wetlands and catch-crops

Catch crops have been shown to reduce average nitrogen load compared to autumn ploughing (Gustafson, Fleischer, and Joelsson, 2000). In the UK, catch crops reduce nitrate leaching by 53 % compared to leaving the field bare (Shepherd, 1999). Empirical studies on catch crops' effects on nutrient loads in Finland are still scarce. The autumn growing season after the main harvest is short due to the colder winter conditions in Finland compared to the UK or southern Sweden, Thus, establishing a sufficient capacity for winter hardening, and reducing overwintering damage has only a small time window (Peltonen-Sainio, 2012). Indeed, catch crops are not as common as in warmer climates.

Wetlands are not, as such, an abatement measure limited to agricultural nonpoint sources, but since they can be constructed on fields and are often established to control nonpoint pollution from agriculture, they have been frequently evaluated as an abatement measure (Cherry et al., 2008; Kay, Edwards, and Foulger, 2009). In a recent review, Kumar and Zhao (2011) conclude that the future direction of constructed wetland modeling work should be focused to quantify the rates of individual processes happening inside the wetland, and that process-based modeling of constructed wetlands is still in its infancy. Empirical studies suggest that wetlands can be very effective at removing nutrients from runoff (Koskiaho, Ekholm, et al., 2003). However, the operational efficiencies vary seasonally and with time, and some wetlands perform poorly in nutrient retention (Braskerud, 2002a; Braskerud, 2002b; Braskerud et al., 2005). Generally, the efficiency of wetland systems is reduced during high flow periods when retention times are shorter (Koskiaho, Ekholm, et al., 2003).

2.2.2 Abatement cost models

Difficulties in establishing empirical evidence on the effectiveness of abatement measures lead to the use of modeling in policy-oriented studies. Gustafson, Fleischer, and Joelsson (2000) discuss potentially effective measures, including fertiliser reduction, constructed wetlands and catch crops but do not attempt to quantify the costs. In practice, all applied economics on abatement relies on one or more environmental modeling tools used for establishing the nutrient load reduction for a combination of measures and environmental conditions not covered directly by empirical data.

Braden et al. (1989) develop an economic model to study sediment transport in a spatially explicit model setting. While their work is not directly quantifying nutrient loads, the results on sediments provide insight particularly on PP abatement.

The state-of-the-art nutrient load models and recent nonpoint pollution control literature consider a wide range of joint production choices influencing the loads and thus, ultimately, the optimal control policies.

A variety of agricultural abatement measures including reductions in the use of fertilizer and the number of animals, cover crop changes, buzzer zones and wetland, have been analysed in the economic context in Sweden(Gren, Elofsson, and Jannke, 1997). Yiridoe and Weersink (1998) study the abatement costs of groundwater N pollution. Byström (1998) and Turner et al. (2000) consider only wetlands for abatement. More recently in Sweden, Brady (2003) include more non-linear abatement options relating to crop choices and regional differences. Johansson (2004) develops a meta-model of agricultural drainage and pesticide transport model (ADAPT) and includes reduced fertiliser treatments, incorporated and broadcast fertiliser treatments, cropping choices, and residue management practices as abatement measures in the US. Goetz and Keusch (2005) combine economic and biophysical models to evaluate the dynamic efficiency of soil erosion and phosphorus reduction policies. Schou et al. (2006) examine several nonpoint source abatement measures including wetland restoration, reduced fertiliser use, the introduction of catch crops into agriculture and livestock reduction in agriculture. Cools et al. (2011) derive nitrogen abatement costs for a Belgian watershed with a SWAT model combined with mixed integer programming in GAMS. Their results, based on constant abatement efficiency of measures, recommend using more productive dairy cattle, implementing basic measures as defined in the WFD, winter cover crops, improving the efficiency of waste water treatment plants, enhancing fodder efficiency for pigs, further treatment of industrial waste water and tuned fertilization. As for Kling (2011), who combines a SWAT model with an evolutionary algorithm to estimate the simultaneous abatement costs for N and P at the Boone River Watershed in the US. Using the MONERIS model, Mewes (2012) points out that improving the farm advisory service could provide cost-effective abatement measure compared to land use changes. Doole (2012) studied nitrate abatement on dairy farms in New Zealand. Abatement measures cover grazing, N fertiliser application, reducing the number of the cows, a change in total milk production, and using maize silage.

2.3 The Case of Finland

Finland is internationally renown as the country of thousands of lakes. The catchment areas of Finnish rivers and lakes are characterised by a wide variety of hydrological and geological features and land-use patterns. The population density is the third lowest in Europe. Climatological variation is high, and the freezing and break-up of ice on rivers and lakes varies considerably. Due to Finland's northern location, lakes are covered by ice for approximately 4 months in the south and up to 8 months in the north. The average rainfall (between 1961-1990) in Finland was 660 mm/year. Spring thaw discharges as much as 40 % of the annual water into the Baltic Sea. Both the sea (23m) and lakes (7m) are shallow, and hence sensitive to eutrophication. The water nutrient concentrations have been increasing in the coastal waters between 1965-1995 (Bonsdorff et al., 1997), whereas inland water bodies are showing mixed trends (Räike et al., 2003). In southern and western Finland, especially in rivers flowing through agricultural areas, there are still increasing trends in nutrient concentrations (Räike et al., 2003).

The political process to control eutrophication in Finland has been long. The problem was observed in Helsinki, the current capital, already at the very beginning of last century, but the national policies for waste water treatment were formulated later, from 1960s (Laakkonen and Parpola, 2010). The national concerns propelled an international process for the protection of the Baltic Sea. In 1974, Finland and the other countries surrounding the Baltic Sea signed a convention to abate pollution to avoid harmful eutrophication. The current objectives set by the Baltic Marine Environment Protection Commission (HELCOM) call for Finland to reduce 1,200 tonnes of nitrogen and 150 tonnes of phosphorus from land-based sources.

The eutrophication of Baltic Sea is further legislated by the EU Marine Strategy Framework Directive 2008/56/EC (MSFD), which aims at reaching a good environmental status of the marine areas. While the MSFD introduced in 2008 points out to eutrophication, more progress has so far been pushed by the Water Framework Directive 2000/60/EC

(WFD). Following WFD, the EU member states committed to ensuring the good ecological quality of surface and ground waters and to report to the Commission on the measures taken to the attainment and progress towards the goal. In their reports to the Commission, the member states incorporate a ”Programme of measures”, which, for the diffuse sources, should include abatement measures and legislative control, as well as the judgement about the cost-efficient combination of the measures along with the estimated costs in general. These controls are to be periodically reviewed and, where necessary, updated. WFD was nationally implemented in Finland in the Act on Water Resources Management (1299/2004).

The Decree on River Basin Districts (2004) divides Finland into 8 river basin districts of which 2 are international. Currently, the water protection act is at the implementation stage. The programme of measures will last until 2021. Relatively fast abatement methods would be needed in order to achieve a positive water quality impact by 2021 and the government adoption of water management plans in 2015 should be preceded by policy advice.

Already in 1998, the Finnish Government issued a Decision-in-Principle on the water protection targets to 2005. The decision states that by the year 2005, annual P and N loads from field cultivation should be reduced by 50 % from the estimated levels in the beginning of the 1990s. Granlund et al. (2005) suggest that water protection measures for agricultural production need to be further intensified, since their results showed that little or no reduction of loads was achieved. In 2006, the Finnish Government issued new targets for 2015 (MoE, 2006), setting load reductions of one third from the average level of 2001-2005. The decision also calls for considerations on the economic profitability, research on cost-efficiency in water protection, and the targeting of environmental support measures. Furthermore, erosion control by developing farming technology and establishing vegetated buffer zones and wetlands, as well as reduced fertilisation and improved manure use are emphasised.

Agriculture is a significant source of nutrients to surface waters. Mitikka and Ekholm (2003) report that lakes with agriculture-dominated catchments were the most eutrophied of the Finnish lake types. Based on data from small Finnish catchments, the mean annual nutrient loss from agricultural land was estimated to be on average 110 kg km^{-2} annually for total phosphorus and 1500 kg km^{-2} for total nitrogen (Vuorenmaa et al., 2002). Agriculture has been estimated to represent 63 % of phosphorus and 51 % of nitrogen loads in Finland (MoE, 2006). Fields cover 2.3 million hectares of land, which is only 6.8 % of Finland's surface area. Thus, the characteristics of agricultural land are important determinants of the loading processes. The dominant soil type for agricultural land is clay, but also sandy and organic soils are cultivated. Finnish soils are geologically young and high in organic matter content (Lilja et al., 2006). The median slope of Finnish fields is 0.77 % and half of the fields are closer than one kilometer from the nearest waterway (Puustinen, Merilä, et al., 1994).

The share of grasslands has been decreasing from total Finnish farm land from the beginning of the 20th century (TIKE, 2012). The share of silage of the total farmland has increased steadily at the expense of hay for the last 15 years and now accounts for 478 600 hectares, making its share of the farmland the largest. Barley has been the dominant cereal in Finnish agriculture. Other popular crops include oats and wheat, whereas oilseeds are less cultivated. Fallows (non-productive fields) cover 10-13 % of total agricultural land. Silage production in Finland is mainly explained by the dairy sector, since specialised beef production farms account for only 6 % of farms. Dairy farms account for approximately half of the market value of agricultural production (Niemi and Ahlstedt, 2010).

As part of the European Union, Finnish agriculture is supported by the Common Agricultural Policy (CAP). While single farm payments per hectare are lower than in some older EU member states, the subsidies including the least favoured area payments, form a considerable share of the farm income (Niemi and Ahlstedt, 2010). Since the MacSharry reform, environmental protection has progressively received more attention in CAP. Some would argue that this interest in environment is legitimising the continuation of policy entitlements to farmers (Clark et al., 1997; Potter and Goodwin, 1998; Winter, 2000), but the ongoing reforms towards a "greener" CAP demonstrate that concrete changes away from direct production subsidies are happening. The Agenda2000 reform, the "Midterm Review of 2003" and the "Health Check" 2009 have lead to a more decoupled "single farm payment" system, which for most parts levels the playing field for crops and fallow and contains environmental cross-compliance requirements. Cross-compliance requirements make single farm payments conditional on following the EU directives and "good agricultural practice and environmental condition", which is specified at the national level.

In Finland, the single farm payment (168-249 €/ha) and the least favoured area payment (150-210 €/ha) require the farmers to maintain good farming practices and environmental conditions on their fields. Common crop farms are required to cultivate at least two types of crops every year. Fallow has to be mowed and cannot be fertilised. Straw burning is allowed only under special circumstances, and heavy machinery on wet soils should be avoided. More relevantly for nutrient abatement, water bodies and main ditches require 0.6 m wide non-farmed zones, and the maximum nitrogen application is between 120 and 250 kg/ha following the Nitrate Directive which is applied in the whole country. Manure storages are required to have capacity for 12 months, and nutrient leaks from the farms should be prevented. Manure cannot be surface applied on fields steeper than 10 %, and nitrogen in general cannot be applied to water saturated, frozen or snow covered fields. Also, nitrogen cannot be applied within 5 meters of water bodies (if the field is steeper than 2 %, then surface application has 10 meter limit).

The Agenda 2000 reforms to CAP gave Finland a significant opportunity to redirect up to 20 % of the subsidy into the CAP's accompanying measures, including agri-environment schemes. While the CAP income component has stayed approximately the same since 2006, especially the national funding of the environmental subsidy system has increased (from 192 to 265 M/€between 2006 and 2011) (Niemi and Ahlstedt, 2010). Together with the EU share, the Finnish agri-environmental scheme, including animal welfare and non-production investments, currently pays out approximately 365 M/€every year.

Compared to the other European Union member states, the adoption rate of the environmental subsidy system is high (circa 95 % of total field area) in Finland (Niemi and Ahlstedt, 2010). The voluntary scheme is divided to three categories of measures, of which the first receives the largest share. This first category consist of several measures such as nutrient standards and environmental education, which are all required for a farm to qualify for a hectare-based compensation. Generally, some second category measures are compulsory for the farmers in the scheme, but they can choose which measures to implement and which not from a broader set. The exact number required/allowed depends on the region. Measures in the third category require participation in the scheme, but are completely optional for the farmers. One of the main stated objectives of the system is water protection (MAF, 2009). The evaluation of the programme period 2000-2006 points out that while the environmental subsidy scheme has flattened the increasing nutrient load trends, it did not, and cannot, stop the trend of increasing nutrient balances at animal-intensive regions, which was considered one of the major threats to nutrient load abatement (Turtola and Lemola, 2004).

Despite the fact that the agri-environmental system has been in place for several programme periods, the costs to farmers for implementing the measures has not been studied widely, at least partly due to farmer's reluctance to provide data. Hence, the costs used for subsidy levels are based on estimates of costs approved in national committees influenced by political stakeholders such as the agricultural producer's union. Nevertheless, the subsidy calculations have been approved by the European commission and been used as proxies for estimating the cost-effectiveness of some of the abatement measures introduced in the previous chapter (Väisänen and Puustinen, 2010).

To avoid mixing cost-efficient nutrient abatement measures with the effects of the current agri-environmental policies, the subsidies in the agri-environmental system are not included in the baseline solutions in this dissertation.

3 Summaries of the studies

3.1 Study I. Abatement costs for agricultural nitrogen and phosphorus loads: a case study of crop farming in south-western Finland

The first study develops an empirical framework for estimating the abatement costs for nutrient loading from agricultural land using the typical measures advocated in Finland. These include the reduction in fertilisation, changes in tillage technologies, change of crop type, and the establishment of green fallow and vegetated buffer zones. A nonpoint production function is formulated based on crop production. Since both the yield response of crop fertilisation and the load response are non-linear, a mathematical model is developed accordingly. A representative farmer decides the fertilisation quantities based on the nitrogen content to maximise the profits. The phosphorus quantity is determined by the nitrogen input and the N:P ratio in compound fertiliser.

Crop allocation is constrained to account for production limitations associated especially with marginal crops. In addition to conventional ploughing, cultivation and no-tillage are considered. The nitrogen load potential is based on a relationship between the load and the ratio of the applied N quantity to the agronomic N recommendation in Finland. The phosphorus load is divided into a dissolved reactive component and a particulate component, which are determined respectively by the average runoff and erosion in addition to an average phosphorus stock and an annual phosphorus input.

Quadratic cost functions are fitted for simulation results of two Common Agricultural Policy regimes. The results indicate that an efficiently designed policy aimed at a 50 % reduction in agricultural nitrogen load of 2003 would cost 99 €/ha. According to the results, decoupling would decrease the abatement costs by 23 %. These nitrogen load reductions would only decrease algae available phosphorus loads by 2 %. Cost-efficient abatement requires multiple modeled measures. A reduction in fertilisation quantities is cost-effective for the first abated units, but a decreasing marginal load and an increasing marginal yield loss from reduction of fertilisation lead to the consequent adoption of buffer zones and eventually increasing the share of fallow of the total arable area. Buffer zones and fallow are established on the fields of the crop with the lowest marginal net revenue.

3.2 Study II. How much can be gained by optimising nutrient abatement spatially - A cost-efficiency comparison of nonpoint arable loads from different Finnish watersheds

The model developed in the second study aims to demonstrate how the environmental heterogeneity of the agricultural nutrient load potential can affect cost-efficient abatement policies. We construct a metamodel of dynamic nutrient load model (ICECREAM) to establish load parameters for non-linear economic optimisation, and derive abatement cost functions for the nutrient loads from two Finnish catchments. Our nonpoint production function includes the geographical variation of nitrogen and phosphorus loads following from topography, soil types and the phosphorus stock in arable land. The abatement choice set includes the change and relocation of crop types, reductions in fertilisation, changing tillage from conventional ploughing to cultivation or direct-tillage, and establishing green fallow or vegetated buffer zones.

Digital elevation models, soil maps, crop cover and municipal statistics of soil phosphorus are processed, classified to discrete categories for River Aurajoki and River Kalajoki catchments. Linear nitrogen load functions for all parameter class combinations are estimated from ICECREAM simulation results of ten years. Nitrogen load is calculated using an exponential function for annually fertilised quantity for each combination of slope, crop, soil and tillage. The total phosphorus load is a sum of particulate and dissolved reactive phosphorus, for which the geographical variation is determined by erosion and runoff, as well as by the phosphorus stock. Erosion parameters are estimated with exponential functions and runoff parameters as linear functions of slope from the ICECREAM results. These results are benchmarked with the land use data from 2009 to the results of other nutrient load models. The metamodel predicts the ICECREAM results accurately, but the inconsistency of the model in the flat Kalajoki fields distorts erosion and the consequent particulate phosphorus load compared to other established load models.

Abatement costs are calculated for each nutrient separately and the reduction target of 30 % set by the Finnish Government is used to illustrate the results. Given the heterogeneous model specification, the nitrogen abatement costs for this target are 2.8 €/ha for Aurajoki and 3.3 €/ha for Kalajoki. For phosphorus, the costs at Aurajoki were 21.8 €/ha, but for Kalajoki this reduction target was not feasible. We calculate the difference in costs of the spatially optimal allocation of reduction measures and compare it with the costs of average non-targeted measures. For the feasible abatement target of 16 % P (N) abatement costs of the heterogeneous model were only 2 % (43 %) of homogeneous costs at Aurajoki and 5 % (20 %) at Kalajoki.

The cost-efficient abatement set includes multiple measures for both nutrients. For nitrogen, the main method is reducing fertilisation, while for phosphorus abatement, establishing grass cover for erosion prone areas is the most economically viable option. Overall reductions in phosphorus loads are challenging, since once the more erosion prone areas are covered, the modeled measures achieve diminutive abatement

3.3 Study III. How to reduce nutrient loads from Dairy farms? - An analytical framework with endogenous feeding and land use

The third study presents a theoretical framework that covers both animal and crop operations at farms. It describes the farmers' decision-making problem with non-linear functions, which capture the economic and biological aspects of the nutrient abatement problem. The representative farmer chooses the land allocation, tillage, the fertilisation of N (and P), the animal head count and feeding, as well as the manure storage and dispersal technologies. Farm land is divided to below and above median distance classes, but otherwise it is treated as homogeneous. Manure and synthetic nutrients are treated equally for yield and load effects.

The representative Finnish dairy farm model is applied at the Kalajoki watershed using data from 2003, and the abatement cost functions are derived separately for nitrogen and total phosphorus. Reaching the national abatement target (1/3 of 2000-2005 load levels) would cost 88 €/ha for N and 354 €/ha for P. Reducing the nitrogen load by 50 % would cost 433 €/ha. Reaching the reduction targets cost-efficiently requires multiple measures at the fields, such as reductions in the fertiliser quantities purchased, changes in tillage, increasing the share of silage production in crops, and setting aside land as green fallow. Dietary changes are utilised for non-lactating cattle, while the optimal milk production is not compromised.

The synergy between cost-efficient N and P abatement is strongest for establishing grass ley fallow. A sensitivity analysis with respect to fertiliser, fodder and milk prices shows that both private optimal nutrient loads and abatement costs are influenced by price changes.

3.4 Study IV. Model for quantifying the synergies between farmland biodiversity conservation and water protection at catchment scale

The fourth study combines farmland biodiversity conservation with water protection at catchment scale and provides an optimising tool for analysing the effect of spatial dependencies on multifunctional agriculture. We use a nonpoint production function based on a profit maximising representative farm with a choice of nitrogen fertilisation and land use allocation. Phosphorus fertilisation is determined by nitrogen fertilisation. The total nutrient load is a sum of the nitrogen and phosphorus loads, which consists of the algae available share of particulate P and DRP, both multiplied by Redfield ratio. Furthermore, the total nutrient load is influenced by a fixed soil phosphorus stock, slope and soil type, which are spatially heterogeneous. Biodiversity is estimated by a habitat suitability index for pollinator insects and is determined by the crop choice, slope, aspect and distance to the closest forest edge from the fields.

The numerical model is applied at the sub-catchment of River Lepsämänjoki in Southern-Finland using data from 2009. For land use choices, we consider barley, conventional timothy grass ley and more diverse meadow plant ley. We depict the trade-off curves of nutrient loads and habitat indices, and illustrate the spatial distribution of cost-efficient allocation, as well as the distribution of loads and habitat using maps. Cutting down the total nutrient load by 50 % would cost 249 €/ha.

We show that significant efficiency gains can be achieved by coordinating the spatial allocation and the type of grass covered ley at the catchment level. An initial least-cost abatement is achieved by reducing fertilisation. Further reductions are reached by establishing grass leys, starting with the most erosion and runoff prone areas of the parcels. When an extended vegetation cover is justified by water protection targets, higher farmland diversity benefits can be gained by including meadow nectar plants in the founding grass seed mixture. The most favourable locations for joint water protection and farmland biodiversity conservation measures are identified at the catchment scale. Targeting the erosion control measures on steep slopes is supported by provision of improved habitats for pollinators.

4 Discussion and conclusions

This dissertation presents an overview of nutrient abatement measures and their costs and effectiveness in Finland. Besides gathering the empirical data and deriving the abatement cost functions, this study contributes to the scientific literature on nonpoint source pollution abatement. This study suggests that in the short-run, the farmers' abatement potential is limited; despite a wide range of available measures, farmers do not possess a solution that would dramatically reduce eutrophication in the short-run. The lack of effective abatement measures stipulates more geospatial information such as topography and relative distances of farms and fields, to identify the cost-efficient environmental policies. It is shown that the cost-efficient abatement of phosphorus and nitrogen are inter-dependent problems under a variety of agricultural production conditions. Given the Finnish data, the main synergy in the nutrient abatement occurs at high reduction targets and costs, when green fallow is endorsed. The adoption of green fallow is further justified by its biodiversity benefits at spatially defined hot spots, but this potential has not yet been fully exploited. The simultaneous solution of animal feeding and crop production demonstrates that in achieving short run abatement, field measures precede feeding changes.

It is argued that reaching the national abatement targets set by the government for 2015 is unlikely, using the measures currently common in Finland. Despite using baselines which do not capture the nutrient reduction of the existing measures, the modeled abatement costs increase rapidly when the national targets are proximated. The exponential abatement costs illustrate that we are approaching, and in some cases reaching, the boundary of the feasible reduction set. Irrespective of the model chosen, even the joint effects of all the measures studied are not able to reduce the phosphorus load sufficiently in the short run. It seems that in the Finnish conditions, erosion control has too small an effect on the total P load for achieving major reductions. The load function for each of the models is derived from studies of (Uusitalo, Turtola, et al., 2001; Uusitalo and Jansson, 2002). However, similar conclusion can be drawn by using ICECREAM model results directly for the total phosphorus load.

The models described in this dissertation do not explicitly deal with weather variation, which introduces an important stochastic element to the decision-making problem. This dissertation uses an approach in which the efficiency of abatement measures is estimated using a simulation model of nutrient loads with ten-year weather data. Like any modeling choice, this limits describing some real life situations. Abnormal years are supported by scarce weather data and the effects of abatement measures are averaged out. However, as climate is changing, the data of the past years should not be extrapolated into the future without due concern of the assumptions made in the model. The uncertainty of abatement decisions is by no means limited to climate or environmental change. The effects of farmers risk aversion and management strategies are important research topics as such for the economics of nutrient abatement. For example, if farmers could predict with more certainty which years will be problematic for crop production, they could reduce the nutrient inputs accordingly and hence save in fertilisation costs and reduce loads. Thus, the abatement costs would be lower than suggested by the deterministic models.

Following the theoretical framework, the least-cost analyses in this dissertation do not include the impacts or costs of the environmental subsidy system. The non-point production functions contain the income subsidy elements, but at the level of a representative farmer, the decisions can be characterised as a private profit-maximisation problem i.e. the income policies are given. Some nutrient load effects of the environmental regulation within CAP, such as abatement related to the maximum nitrogen application regulated in the Nitrate Directive, or the minimum distance of field edges from water ways, are not considered in the dissertation. Moreover, leaving out the subsidy effects on the entry/exit from the sector means that the nutrient load from purely private land allocation might be smaller than the currently observed load. According to Lankoski and Ollikainen (2011) the Finnish agriculture shifted towards more nitrogen-intensive land uses during the CAP period.

Eliminating farm subsidies would surely have implications on cost-efficient nutrient abatement, but the projected consequences are dependent on the scale of market liberalisation. Retaining the income subsidies of agriculture in the analysis represents the current institutional framework for the environmental policies better than assuming open markets. This assumption has been commonly used in an empirical economic setting. However, a more detailed interpretation of abatement cost results requires separating the model baseline from the current status quo in Finland, which includes both market distortions and the nutrient reduction effects of subsidies. If some cost-efficient measures have already been implemented, the remaining reduction potential is smaller and the costs will be higher. Since the modeled measures are mainly based on the options already existing in the national environmental subsidy system, and since they are generally not very effective relative to the magnitude of the abatement targets, it is evident that there is some overlap between the existing and cost-efficient measures. Similarly, the environmental gains of CAP, including the effects of cross compliance are not decreasing the model baseline loads. Thus, achieving reductions from the current load levels will be more expensive than indicated by the results derived from the private profit maximisation baselines. Furthermore, as a part of the limited abatement potential is already utilized, reaching the policy targets with the modeled abatement measures seems not only more expsensive, but also infeasible. While the focus of this dissertation was not in the environmental policy analysis, the developed models offer a reasonable starting point for such efforts.

The choice of baseline has other effects. Even when retaining the behavioural assumption of a profit maximising farmer, it is possible to question if agricultural producers are perfectly informed. The existence of other market failures could play a role in attaining nutrient abatement. For example, if the farmers are overestimating the marginal yield gains from fertilisation, some nutrient abatement could be achieved with profit. This direction for future research would be interesting since such win-win situations have not been generally identified in the nonpoint source abatement literature.

Another more prominent direction of research is dynamic modeling. The time dimension allows more thorough representation of crop rotations, manure accumulation, as well as phosphorus stocks and loads. Thus, dynamic modeling could provide understanding on efficient long-run abatement, but with a computational cost.

Establishing the abatement cost functions for nonpoint source loads is a complex modeling task. Academic literature on the topic contains the basic descriptions of the models, but generally leaves the researchers, wishing to replicate or improve the existing models, to programme their own numerical models. This dissertation includes an electronic supplement containing the mathematical programmes that could save considerable time from future researchers wishing to study the topic.

References

Adar, Zvi and James M. Griffin (1976). "Uncertainty and the Choice of Pollution Control Instruments". In: *Journal of Environmental Economics and Management* 3, pp. 178–188.

Aertebjerg, Gunni (2001). *Eutrophication in Europe's coastal waters*. Vol. 7/2001. Topic report. Luxembourg: Office for Official Publications of the European Communities, p. 115.

Bakhsh, A., R.S. Kanwar, and D.L. Karlen (2005). "Effects of liquid swine manure applications on NO3-N leaching losses to subsurface drainage water from loamy soils in Iowa". In: *Agriculture, Ecosystems & Environment* 109 (1-2), pp. 118–128.

Ball Coelho, B.R., R.C. Roy, and A.J. Bruin (2006). "Nitrogen recovery and partitioning with different rates and methods of sidedressed manure". In: *Soil Science Society of America Journal* 70 (2), pp. 464–473.

Baumol, William J. and Wallace E. Oates (1989). *The theory of environmental policy*. Cambridge: Cambridge University Press, p. 299.

Bonsdorff, E. et al. (1997). "Coastal eutrophication: Causes, consequences and perspectives in the Archipelago areas of the northern Baltic Sea". In: *Estuarine, Coastal and Shelf Science* 44, Supplement 1, pp. 63–72.

Bosch, D-J, M-L Wolfe, and K-F Knowlton (2006). "Reducing phosphorus runoff from dairy farms." In: *Journal of Environmental Quality* 35(3), pp. 918–927.

Braden, J.B. et al. (1989). "Optimal spatial management of agricultural pollution". In: *American Journal of Agricultural Economics* 71, pp. 404–413.

Brady, Mark (2003). *Managing agriculture and water quality four essays on the control of large-scale nitrogen pollution*. Vol. 369. Uppsala: Department of Economics, Swedish University of Agricultural Sciences.

Braskerud, B. C. et al. (2005). "Can Constructed Wetlands Reduce the Diffuse Phosphorus Loads to Eutrophic Water in Cold Temperate Regions?" In: *Journal of Environmental Quality* 34 (6), pp. 2145–2155.

Braskerud, B.C (2002a). "Factors affecting nitrogen retention in small constructed wetlands treating agricultural non-point source pollution". In: *Ecological Engineering* 18 (3), pp. 351–370.

– (2002b). "Factors affecting phosphorus retention in small constructed wetlands treating agricultural non-point source pollution". In: *Ecological Engineering* 19 (1), pp. 41–61.

Butler, David M. et al. (2008). "Runoff water quality from manured riparian grasslands with contrasting drainage and simulated grazing pressure". In: *Agriculture, Ecosystems & Environment* 126 (3), pp. 250–260.

Byström, Olof (1998). "The nitrogen abatement cost in wetlands". In: *Ecological Economics* 26 (3), pp. 321–331.

Cannell, R.Q. (1985). "Reduced tillage in north-west Europe - A review". In: *Soil and Tillage Research* 5 (2), pp. 129–177.

Carpenter, S.R. et al. (1998). "Nonpoint pollution of surface waters with phosphorus and nitrogen". In: *Ecological Applications* 8, pp. 559–568.

Cherry, K.A. et al. (2008). "Assessing the effectiveness of actions to mitigate nutrient loss from agriculture: A review of methods". In: *Science of The Total Environment* 406 (1-2), pp. 1–23.

Clark, J R A et al. (1997). "Conceptualising the evolution of the European Union's agri-environment policy: a discourse approach". In: *Environment and Planning A* 29 (10), pp. 1869–1885.

Cools, Jan et al. (2011). "Coupling a hydrological water quality model and an economic optimization model to set up a cost-effective emission reduction scenario for nitrogen". In: *Environmental Modelling & Software* 26 (1), pp. 44–51.

Cromwell, G.L. et al. (1993). "Efficacy of Phytase in Improving the Bioavailability of Phosphorus in Soybean Meal and Corn-Soybean Meal Diets for Pigs". In: *Journal of Animal Science* 71, p. 1831.

Daniels, R. B. and J. W. Gilliam (1996). “Sediment and Chemical Load Reduction by Grass and Riparian Filters”. In: *Soil Sci. Soc. Am. J.* 60 (1), pp. 246–251.

Daverede, IC et al. (2004). “Phosphorus runoff from incorporated and surface-applied liquid swine manure and phosphorus fertilizer”. In: *Journal of environmental quality* 33 (4), pp. 1535–1544.

Doole, Graeme J. (2012). “Cost-effective policies for improving water quality by reducing nitrate emissions from diverse dairy farms: An abatement-cost perspective”. In: *Agricultural Water Management* 104, pp. 10–20.

Drecht, G. Van et al. (2009). “Global nitrogen and phosphate in urban wastewater for the period 1970 to 2050”. In: *Global Biogeochemical Cycles* 23.

Drud, A. (2004). *Conopt User Manual*. Denmark: Arki Consulting and Development.

Drud, Arne (1985). “CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems”. In: *Mathematical Programming* 31 (2).

Ekholm, Petri et al. (2005). “Phosphorus loss from different farming systems estimated from soil surface phosphorus balance”. In: *Agriculture, Ecosystems & Environment* 110 (3-4), pp. 266–278.

Erisman, Jan Willem et al. (2008). “How a century of ammonia synthesis changed the world”. In: *Nature Geoscience* 1, pp. 636–639.

Feinerman, Eli, Darrell J. Bosch, and James W. Pease (Feb. 2004). “Manure Applications and Nutrient Standards”. In: *American Journal of Agricultural Economics* 86 (1), pp. 14–25.

Gaud, William S. (Mar. 1968). *The Green Revolution: Accomplishments and Apprehensions*. Agency for International Development Department of State. URL: http://www.agbioworld.org/biotech-info/topics/borlaug/borlaug-green.html.

Ghebremichael, Lula T. and Mary C. Watzin (2011). “Identifying and controlling critical sources of farm phosphorus imbalances for Vermont dairy farms”. In: *Agricultural Systems* 104 (7), pp. 551–561.

Goetz, Renan-Ulrich and Alois Keusch (2005). “Dynamic efficiency of soil erosion and phosphor reduction policies combining economic and biophysical models”. In: *Ecological Economics* 52 (2), pp. 201–218.

Granlund, Kirsti et al. (2005). “Assessment of water protection targets for agricultural nutrient loading in Finland”. In: *Journal of Hydrology*, pp. 251–260.

Gren, Ing-Marie, Katarina Elofsson, and Paul Jannke (1997). “Cost-Effective Nutrient Reductions to the Baltic Sea”. In: *Environmental and Resource Economics* 10, pp. 341–362.

Griffin, R. C. and D. W. Bromley (1982). “Agricultural runoff as a nonpoint externality: theoretical development”. In: *American Journal of Agricultural Economics* 64, pp. 547–552.

Gustafson, A., S. Fleischer, and A. Joelsson (2000). “A catchment-oriented and cost-effective policy for water protection”. In: *Ecological Engineering* 14 (4), pp. 419–427.

Holland, J.M. (2004). “The environmental consequences of adopting conservation tillage in Europe: reviewing the evidence”. In: *Agriculture, Ecosystems & Environment* 103 (1), pp. 1–25.

Huhtanen, Pekka, Juha Nousiainen, and Eila Turtola (2011). “Dairy farm nutrient management model: 2. Evaluation of different strategies to mitigate phosphorus surplus”. In: *Agricultural Systems* 104, pp. 383–391.

Innes, Robert (2000). “The economics of livestock waste and its regulation”. In: *American Journal of Agricultural Economics* 92, pp. 97–117.

Johansson, Robert C. (2004). “Metamodelling Phosphorus Best Management Practices for Policy Use: A Frontier Approach”. In: *Agricultural Economics* 30 (1), pp. 63–74.

Kay, Paul, Anthony C. Edwards, and Miles Foulger (2009). “A review of the efficacy of contemporary agricultural stewardship measures for ameliorating water pollution problems of key concern to the UK water industry”. In: *Agricultural Systems* 99 (2-3), pp. 67–75.

Kling, Catherine L. (2011). “Economic Incentives to Improve Water Quality in Agricultural Landscapes: Some New Variations on Old Ideas”. In: *American Journal of Agricultural Economics* 93 (2), pp. 297–309. eprint: http : / / ajae . oxfordjournals.org/content/93/2/297.full.pdf+html.

Koopmans, G.F et al. (2002). “Soil phosphorus quantity-intensity relationships to predict increased soil phosphorus loss to overland and subsurface flow”. In: *Chemosphere* 48 (7), pp. 679–687.

Koskiaho, J., S. Kivisaari, et al. (2002). “Reduced tillage: influence on erosion and nutrient losses in a claye yfield in southern-Finland”. In: *Agricultural and Food Science* 11, pp. 37–50.

Koskiaho, Jari, Petri Ekholm, et al. (2003). “Retaining agricultural nutrients in constructed wetlands - experiences under boreal conditions". In: *Ecological Engineering* 20 (1), pp. 89–103.

Kumar, J.L.G. and Y.Q. Zhao (2011). "A review on numerous modeling approaches for effective, economical and ecological treatment wetlands". In: *Journal of Environmental Management* 92, pp. 400–406.

Kurz, Isabelle, Colin Daniel O'Reilly, and Hubert Tunney (2006). "Impact of cattle on soil physical properties and nutrient concentrations in overland flow from pasture in Ireland". In: *Agriculture, Ecosystems & Environment* 113 (1-4), pp. 378–390.

Kutra, Ginutis and Ramute Aksomaitiene (2003). "Use of nutrient balances for environmental impact calculations on experimental field scale". In: *European Journal of Agronomy* 20 (1-2), pp. 127–135.

Laakkonen, Simo and Antti Parpola (2010). "Itämeren tulevaisuus". In: ed. by Saara Bäck et al. Gaudeamus. Chap. Rehevöitymiskäsitysten historiaa, pp. 87–97.

Lankoski, Jussi E. and Markku Ollikainen (2011). "Counterfactual approach for assessing agri-environmental policy: The case of the Finnish water protection policy". In: *122nd Seminar European Association of Agricultural Economists*.

Letson, David et al. (1998). "Confined Animal Production and Groundwater Protection". In: *Review of Agricultural Economics* 20 (2), pp. 348–364. eprint: http://aepp.oxfordjournals.org/content/20/2/348.full.pdf+html.

Lilja, Harri et al. (2006). *Suomen maannostietokanta. Maannoskartta 1:250000 ja maaperän ominaisuuksia*. Finnish. Agrifood Research Finland. URL: www.mtt.fi/mtts/pdf/mtts114.pdf.

MAF (2009). *Manner-Suomen maaseudun kehittämisohjelma 2007-2013*. Finnish. Ministry of Agriculture and Forestry.

Mander, Ulo, Yoshihiko Hayakawa, and Valdo Kuusemets (2005). "Purification processes, ecological functions, planning and design of riparian buffer zones in agricultural watersheds". In: *Ecological Engineering* 24 (5), pp. 421–432.

McDowell, Richard W (2012). "Minimising phosphorus losses from the soil matrix". In: *Current Opinion in Biotechnology* 23 (6), pp. 860–865.

McGechan, M. B. and C. F. E. Topp (2004). "Modelling environmental impacts of deposition of excreted nitrogen by grazing dairy cows". In: *Agriculture, Ecosystems & Environment* 103 (1), pp. 149–164.

Mewes, Melanie (2012). "Diffuse nutrient reduction in the German Baltic Sea catchment: Cost-effectiveness analysis of water protection measures". In: *Ecological Indicators* 22, pp. 16–26.

Mitikka, Sari and Petri Ekholm (2003). "Lakes in the Finnish Eurowaternet: status and trends". In: *Science of The Total Environment* 310 (1-3), pp. 37–45.

MoE (2006). *Water protection to targets to 2015*. online.

Niemi, J. and J. Ahlstedt (2010). *Finnish Agriculture and Rural Industries 2010*. Helsinki: MTT Agrifood Research, p. 94.

Peltonen-Sainio, Pirjo (2012). "Crop production in a northern climate". In: *FAO/OECD Workshop: Building Resilience for Adaptation to Climate Change in the Agriculture Sector*.

Potter, Clive and Philip Goodwin (1998). "Agricultural liberalization in the European union: an analysis of the implications for nature conservation". In: *Journal of Rural Studies* 14 (3), pp. 287–298.

Puustinen, Markku, Jari Koskiaho, and Kimmo Peltonen (2005). "Influence of cultivation methods on suspended solids and phosphorus concentrations in surface runoff on clayey sloped fields in boreal climate". In: *Agriculture, Ecosystems & Environment* 105 (4), pp. 565–579.

Puustinen, Markku, Eero Merilä, et al. (1994). *Kuivatustila, viljelykäytäntö ja vesistökuormitukseen vaikuttavat ominaisuudet Suomen pelloilla*. Vesi- ja ympäristöhallitus.

Räike, A. et al. (2003). "Trends of phosphorus, nitrogen and chlorophyll a concentrations in Finnish rivers and lakes in 1975-2000". In: *Science of The Total Environment* 310 (1-3), pp. 47–59.

Rao, Nalini S. et al. (2009). "Modeling watershed-scale effectiveness of agricultural best management practices to reduce phosphorus loading". In: *Journal of Environmental Management* 90 (3), pp. 1385–1395.

Rasmussen, K.J. (1999). "Impact of ploughless soil tillage on yield and soil quality: A Scandinavian review". In: *Soil and Tillage Research* 53 (1), pp. 3–14.

Rotz, C-A et al. (1999). "A dairy herd model for use in whole farm simulations." In: *Journal of Dairy Science* 82(12), pp. 2826–2840.

Saam, H. et al. (2005). "Use of animal density to estimate manure nutrient recycling ability of Wisconsin dairy farms". In: *Agricultural Systems* 84 (3), pp. 343–357.

Saarijärvi, K. et al. (2004). "N and P leaching and microbial contamination from intensively managed pasture and cut sward on sandy soil in Finland”. In: *Agriculture, Ecosystems & Environment* 104 (3), pp. 621–630.

Salo, T. and E. Turtola (2006). “Nitrogen balance as an indicator of nitrogen leaching in Finland”. In: *Agriculture, Ecosystems & Environment* 113 (1), pp. 98–107.

Schnitkey, GD and MJ Miranda (1993). “The Impactt of Pollution Controls on Livestock-Crop Producers”. English. In: *Journal of Agricultural and Resource Economics* 18 (1), pp. 25–36.

Schou, J.S. et al. (2006). *Modelling Cost-efficient Reductions of Nutrient Loads to the Baltic Sea - Concept, Data and Cost Functions for the Cost Minimisation Model*. English. National Environmental Research Institute. URL: http://www2.dmu.dk/pub/FR592.pdf.

Sheffield, R.E. et al. (1997). “Off-stream water sources for grazing cattle as a stream bank stabilization and water quality BMP”. In: *Transactions of the ASAE* 40 (3), pp. 595–604.

Shepherd, M. A. (1999). “The effectiveness of cover crops during eight years of a UK sandland rotation”. In: *Soil Use and Management* 15 (1), pp. 41–48.

Shortle, J. S. and D. G. Abler (2001). *Environmental policies for agricultural pollution control*. edited by J.S. Shortle and D.G. Abler. Wallingford: CABI Publishing, p. 224.

Shortle, J. S. and J. W. Dunn (1986). “The relative efficiency of agricultural source water pollution control policies”. In: *American Journal of Agricultural Economics* 68, pp. 668–677.

Shortle, James S. and Richard D. Horan (2001). “The economics of nonpoint pollution control”. In: *Journal of Economic Surveys* 15, pp. 255–289.

Simmelsgaard, S. E. and J. Djurhuus (1998). “An empirical model for estimating nitrate leaching as affected by crop type and the long-term fertilizer rate”. In: *Soil Use and Management* 14, pp. 37–43.

Smith, V. H., G. D. Tilman, and J. C. Nekola (1999). “Eutrophication: impacts of excess nutrient inputs on freshwater, marine, and terrestrial ecosystems”. In: *Environmental Pollution* 100 (1-3), pp. 179–196.

Soane, B.D. et al. (2012). “No-till in northern, western and south-western Europe: A review of problems and opportunities for crop production and the environment”. In: *Soil and Tillage Research* 118, pp. 66–87.

Steinfeld, H. (2006). *Livestock's Long Shadow: Environmental Issues and Options*. Livestock, Environment and Development, FAO.

Syversen, Nina (2005). “Effect and design of buffer zones in the Nordic climate: The influence of width, amount of surface runoff, seasonal variation and vegetation type on retention efficiency for nutrient and particle runoff”. In: *Ecological Engineering* 24 (5), pp. 483–490.

TIKE (2012). *Peltoalat 1910-2011*. Finnish. Maa- ja metsätalousministeriön tietopalvelukeskus. URL: http://www.maataloustilastot.fi/sites/default/modules/pubdlcnt/pubdlcnt.php?file=/sites/default/files/aikasarjat_viljelykasvien_pelltoalan_kaytto_1910_ja_1920-2011.xls&nid=35.

Turner, R.Kerry et al. (2000). “Ecological-economic analysis of wetlands: scientific integration for management and policy”. In: *Ecological Economics* 35 (1), pp. 7–23.

Turtola, E and E Kemppainen (1998). “Nitrogen and phosphorus losses in surface runoff and drainage water after application of slurry and mineral fertilizer to perennial grass ley.” In: *Agricultural and Food Science in Finland* 7(5/6), pp. 569–581.

Turtola, E. and R. Lemola (2004). *Maatalouden ympäristötuen seuranta MYTVAS 2*. 1st ed. Vol. 59. Jokioinen: Maa- ja elintarviketalouden tutkimuskeskus, p. 176.

Turtola, Eila, Laura Alakukku, et al. (2007). “Surface runoff, subsurface drainflow and soil erosion as affected by tillage in a clayey Finnish soil”. In: *Agricultural and Food Science* 16, pp. 332–351.

Uusi-Kämppä, Jaana (2005). “Phosphorus purification in buffer zones in cold climates”. In: *Ecological Engineering* 24, pp. 491–502.

Uusi-Kämppä, Jaana and Lauri Jauhiainen (2010). “Long-term monitoring of buffer zone efficiency under different cultivation techniques in boreal conditions”. In: *Agriculture, Ecosystems & Environment* 137 (1-2), pp. 75–85.

Uusitalo, R. and H. Jansson (2002). “Dissolved reactive phosphorus in runoff assessed by soil extraction with an acetate buffer”. In: *Agricultural and Food Science in Finland* 11, pp. 343–353.

Uusitalo, R., E. Turtola, et al. (2001). “Particulate phosphorus and sediment in surface runoff and drainflow from clayey soils”. In: *Journal of Environmental Quality* 30 (2), pp. 589–595.

Väisänen, Sari and Markku Puustinen (2010). *Maatalouden vesistökuormituksen hallinta - Seuranta, mallit ja kustannustehokkaat toimenpiteet vesienhoidon toimenpideohjelmissa*. Vol. 23. Suomen ympäristökeskus, p. 134.

Vuorenmaa, J. et al. (2002). "Losses of nitrogen and phosphorus from agricultural and forest areas in Finland during the 1980s and 1990s". In: *Environmental Monitoring and Assessment* 76, pp. 213–248.

Winter, Michael (2000). "Strong policy or weak policy? The environmental impact of the 1992 reforms to the CAP arable regime in Great Britain". In: *Journal of Rural Studies* 16 (1), pp. 47–59.

Withers, P. J. A. et al. (2006). "Some effects of tramlines on surface runoff, sediment and phosphorus mobilization on an erosion-prone soil". In: *Soil Use and Management* 22 (3), pp. 245–255.

Wossink, G.A.A, A.G.J.M Oude Lansink, and P.C Struik (2001). "Non-separability and heterogeneity in integrated agronomic-economic analysis of nonpoint-source pollution". In: *Ecological Economics* 38 (3), pp. 345–357.

Yiridoe, E.K. and A. Weersink (1998). "Marginal abatement costs of reducing groundwater-N pollution with intensive and extensive farm management choices." In: *Agricultural and Resource Economics Review* 27, pp. 169–185.

Zeimen, M. B. et al. (2006). "Combining management practices to reduce sediment, nutrients, and herbicides in runoff". In: *Journal of Soil and Water Conservation* 61 (5), pp. 258–267. eprint: http://www.jswconline.org/content/61/5/258.full.pdf+html.

Appendix: Mathematical programming

The nonpoint production functions in this study have been programmed with General Algebraic Modeling System (GAMS). GAMS is a modeling tool designed for solving large complex optimisation problems independent from computing hardware and with easily approachable syntax. All optimisation problems were solved with a nonlinear CONOPT3 solver which is based on the generalised reduced gradient approach (Drud, 1985). CONOPT3 was chosen since according to its developers it is well suited for models with very nonlinear constraints (Drud, 2004). Furthermore, CONOPT3 performed better in solving the models than MINOS.

CONOPT3 algorithm contains several procedures for finding an optimal solution. In the pre-processing stage constraints-variable pairs that can be solved a priori are assigned the final values and constraints always satisfied are excluded. Algorithm moves as many nonlinearities as possible to the objective function, but is affected by user defined bounds on intermediate variables leading to nonbinding constraints potentially affecting the optimisation results. Then model is scaled by the algorithm. A feasible starting point for CONOPT3 is formulated in GAMS and CONOPT3 tries to remain feasible and follow a path of improving feasible points until it reaches a local optimum. This is started with Phase 0, which uses Newtonian methods for sub set of variables and identifies "'difficult"' nonlinear or large infeasibility equations. In Phase 1 artificial variables are added to the infeasible equations and the sum of these artificial variables is minimised (subject to the feasible constraints remaining feasible). The optimisation consists of generalised reduced gradient algorithm with procedures for sequential linear programming, steepest edge, and sequential quadratic programming. The selected method depends on memory availability and performance statistics. (Drud, 2004)

All reported results are based on local optima reached by the solver. The model code follows the good practice recommended in the solver manual. Initial values have been provided to give the solver algorithm a feasible starting point. The crash procedure of the preprocessing state was not used to enforce the initial values. The decision and intermediate variables have algorithmic upper and lower bounds. These are required since CONOPT3, and GRG algorithms in general, struggle with maintaining feasibility far from the optimum. Many of the equations in the models contain divisions, so nonlinearity from small arguments is present and solving has been facilitated by the use of intermediate variables. Assuming constant land quantity in short run allows representing some of the constraints as equalities instead of inequalities and hence should be facilitating solver challenges in determining which constraints are binding. The models have not been written to the most convenient model solving scale, but rely on CONOPT3 auto-scaling features. While some of the modeled processes could be described naturally by discontinuous functions, the solver capacity is not sufficient to reliably deal with dynamic linear program type of problems. Following from use of intermediate variables and algorithmic bounds, finding feasible solutions is large part of the model optimisation problem. Generally, the abatement cost problems in studies I-IV have a considerable share of nonlinear equations. Of the 450 dairy model equations circa one-third are "'difficult"'.

To keep the mathematical notation in the study compact, some of the equations and intermediate variables in the GAMS code have been condensed for the article versions. In CONOPT3 the intermediate variables only used to define objective terms are eliminated from the models and the constraints are moved into the objective functions. For crops and tillage only a single index is used in the GAMS code. The code of the models is divided to several files. Each of the studies (I-IV) has a file for the economic abatement model. Sensitivity analyses and ordinary least-square estimations of the various functions are performed in separate files called in or by the economic abatement model files. Data is scattered in Microsoft outlook database (.mdb files), excel-sheets and GAMS files. Thus, the code for models as such is not sufficient for replicating the results. Graphs are produced with Gnuplot 4.6 and gnuplotxyz.

Appendix: Code

Title Finnish Agricultural Nutrient Abatement Model

```
$Title Finnish Agricultural Nutrient Abatement Model
$ontext
This model describes the nutrient abatement problem of nitrogen (and phosphorus) for agriculture of
South-West Finland. Abatement problem is described with a constrained profit maximation problem of a
representative farm. The nutrient load parameters are from ICECREAM model for phosphorus and
Swedish/Danish studies for nitrogen.
$offtext
* SETS
SETS
        j crop types and tillage
/
        wwhe1   winterwheat conventional
        wwhe2   winterwheat chisel plough
        wwhe3   winterwheat no till
        swhe1   springwheat conventional
        swhe2   springwheat chisel plough
        swhe3   springwheat no till
        sbar1   spring barley  conventional
        sbar2   spring barley  chisel plough
        sbar3   spring barley  no till
        trap1   spring turnip rape conventional
        trap2   spring turnip rape chisel plough
        trap3   spring turnip rape no till
        oats1   oats conventional
        oats2   oats chisel plough
        oats3   oats no till
        silag   conventional
        sbeet   sugarbeet conventional
        fall    fallow
/
*Crop subsets
        sb(j) spring barleys
/       sbar1*sbar3  /
        oa(j) oats
/       oats1*oats3  /
        wh(j) wheats
/       wwhe1*wwhe3, swhe1*swhe3  /
        tr(j) turnip rapes
/       trap1*trap3  /
*Yield function subsets
        i(j) Mitscherlich crops
            /
             wwhe1, wwhe2, wwhe3
             swhe1, swhe2, swhe3
             sbar1, sbar2, sbar3
             oats1, oats2, oats3
            /
            h(j) Quadratic crops
            /
```

```
            trap1, trap2, trap3
            silag
            sbeet
          /
        MACON max resource constraints
/
        maxtr      maximum rape
        maxfa      maximum EU fallow
        maxsb      maximum sugarbeet (quota)
        maxwhe     maximum wheat under new acreage subsidy  2006
        maxbar     maximum barley under new acreage subsidy 2006
        maxoat     maximum oats under new acreage subsidy   2006
/
       EQCON equalised constraints
/
       ha         arable land available for model farm
/
        MIC   min resource constraints
/
        minfall      minimum EU fallow
/
        Y yield function coefficients for Mitscherlich and quadratic
/
        a            intercept or yield when no chemical fertilization occurs
        b            positive slope as yield increases with N application
        c            curvature implies diminishing productivity
        m            Mitscherlich parameter m
        k            Mitscherlich parameter k
        d            Mitscherlich parameter b
/
        lea leaching and load parameters

/       drp phosphorus runoff and technology
        pp  phosphoruserosion and technology
        nic nitrogen leaching based on icecream
/
;
* POLICY PARAMETERS
SCALARS
        N_TAX         nitrogen tax euro per kg N       / 0 /
        RETENTION      retention coefficient          /1/
        REDUCTION      dummy reduction                /-10/
        RED_L         abatement rate                 /-10/
        B_NLOADt0      baseline prof max N load       /120/
        B_PLOADt0      dummy baseline prof max P load   /700/
        STP0          average initial soil P mg per l from Sami Myyrä /10.6/
        SF            the share of P surface flow Puustinen(SYKE)  /0.4/
        DF            the share of P drainage flow             / 0.6/
        fallreq06      fallow requirement of 2006 for A area / 0.088/
        fallreq03      fallow requirement of 2006 for A area / 0.1/
        farmsize       average farm size of the region       /38/
        studreg        size of the study region            /481500/
```

```
;
* Revenue and cost

PARAMETERS
      N_PRICE(j) cost of mineral fertilizer euro per kg
/
      wwhe1*wwhe3    1.1
      swhe1*swhe3    1.2
      sbar1*sbar3    1.2
      trap1*trap3    1.2
      oats1*oats3    1.2
      silag          1.2
      sbeet          1.56

/
      D_PRICE(j) cost of drying euro per kg
/
      wwhe1*wwhe3    0.01
      swhe1*swhe3    0.01
      sbar1*sbar3    0.01
      oats1*oats3    0.01
/
      C_PRICE(j) Year 2003 commodity price euro per kg
/
      wwhe1*wwhe3    0.127
      swhe1*swhe3    0.127
*       barley from malt barley price 80\% and fodder barley price 20\%
      sbar1*sbar3    0.130
      trap1*trap3    0.260
      oats1*oats3    0.099
      silag          0.149
      sbeet          0.054
/
      SUBS(j)   Year 2003 sum of subsidies  euro per ha
/
      wwhe1*wwhe3    534
      swhe1*swhe3    534
      sbar1*sbar3    513
      trap1*trap3    572
      oats1*oats3    438
      silag          364
      sbeet          352
      fall           364
/
      SUBS06(j) Year 2006 sum of subsidies  euroa per ha
/
      wwhe1*wwhe3    565
      swhe1*swhe3    565
      sbar1*sbar3    454
      trap1*trap3    589
      oats1*oats3    416
      silag          410
```

```
        sbeet        539
        fall         410
/
        CAP2003(j) Only CAP subsidy for 2003 euro per ha
/
        wwhe1*wwhe3     279
        swhe1*swhe3     279
        sbar1*sbar3     279
        trap1*trap3     279
        oats1*oats3     279
        silag          214
        fall           214
/
        NATSUB2003(j) Only national subsidies for 2003
/
        wwhe1*wwhe3     105
        swhe1*swhe3     105
        sbar1*sbar3     84
        trap1*trap3     279
        oats1*oats3     9
        sbeet          202
        silag          247
        fall           214
/
        LFA(j) Least favoured area subsdies for 2003
/
        wwhe1*wwhe3     150
        swhe1*swhe3     150
        sbar1*sbar3     150
        trap1*trap3     150
        oats1*oats3     150
        sbeet          150
        silag          150
        fall           150
/
        LFA06(j) Least favoured area subsdies for 2006
/
        wwhe1*wwhe3     170
        swhe1*swhe3     170
        sbar1*sbar3     170
        trap1*trap3     170
        oats1*oats3     170
        sbeet          170
        silag          170
        fall           170
/
        ESUBS(j) Environmental subsidies (not in the model) euro per ha
/
        wwhe1*wwhe3    94
        swhe1*swhe3    94
        sbar1*sbar3    94
        trap1*trap3    94
```

```
    oats1*oats3    94
    silag          94
    sbeet          94
    fall           94
/
    FIX_C(j) fixed costs euro per ha  from Koikkalainen (unpublished)
/
    wwhe1   436
    wwhe2   433
    wwhe3   423
    swhe1   436
    swhe2   433
    swhe3   423
    sbar1   436
    sbar2   433
    sbar3   423
    trap1   436
    trap2   433
    trap3   423
    oats1   436
    oats2   433
    oats3   423
    silag   382.75
    sbeet   711
    fall    166.47
/
    BUFZ_C(j) buffer costs euro per ha from Koikkalainen (unpublished)
/
    wwhe1*wwhe3     231.47
    swhe1*swhe3     231.47
    sbar1*sbar3     231.47
    trap1*trap3     231.47
    oats1*oats3     231.47
    silag           221.47
    sbeet           244.47
    fall            231.47
/
    P_RATIO(j)  Fixed ratio between P and N (Kg of P per kg of N )
/
    wwhe1*wwhe3    0.12
    swhe1*swhe3    0.15
    sbar1*sbar3    0.15
    trap1*trap3    0.15
    oats1*oats3    0.15
    silag          0.14
    sbeet          0.11
    fall           1
/
;
* Resource Limits ---------------------------------------------
PARAMETERS
     VALCON(MACON) resource endowment and technology constraints
```

```
/
        maxtr         9.5
        maxfa         19
        maxsb         0.5
        maxwhe         19
        maxbar        19
        maxoat        19
/
       HACON(EQCON)   Total land (equality constraint)
/      ha           38
/
       MCON(MIC)     Minimum constraints (minimum fallow required by EU)
/
       minfall       3.8
/
;
* Yield function parameters (Lehtonen 2001 and other sources)
TABLE YC(j,Y)  yield function coefficients for both mitscherlich and quadratic
              m          k          d        a       b         c
wwhe1       4871.0      0.7623      0.0104
wwhe2       4747.2      0.7623      0.0104
wwhe3       3937.3      0.7623      0.0104
swhe1       4871.0      0.7623      0.0104
swhe2       4747.2      0.7623      0.0104
swhe3       3937.3      0.7623      0.0104
sbar1       5309.6      0.8280      0.0168
sbar2       5421.2       0.8280     0.0168
sbar3       5105.1      0.8280      0.0168
oats1       5659.1      0.7075      0.0197
oats2       5677.0      0.7075      0.0197
oats3       5368.4      0.7075      0.0197
trap1                               1096.1    9.82    -0.0354
trap2                               1052.26    9.82    -0.0354
trap3                               986.49    9.82    -0.0354
silag                              1182.9   24.24    -0.0394
sbeet                               23630    53.21    -0.083
;
* LEACHING PARAMETERS
table leach(j,lea) N and P load coefficients
$set cmdfile leach.txt
$onecho > "\%cmdfile\%"
I="C:\gamsdir\crop_farm\parametrit113.xls"
O="C:\gamsdir\crop_farm\leach.inc"
R="LEACH!F2:I20"
$offecho
$call =xls2gms \at"\%cmdfile\%"
$include leach.inc
;
PARAMETERS
       N_N(j) Normal N-dose from Tuottopehtori(compaire with INIT(j))
/
       wwhe1*wwhe3    120
```

```
        swhe1*swhe3     100
        sbar1*sbar3     90
        trap1*trap3     100
        oats1*oats3     90
        silag           180
        sbeet            120
        fall            0.00001
/
*CONSTRAINT PARAMETERS
TABLE BACON(j, EQCON) Total farm land balance
             ha
        wwhe1  1
        wwhe2  1
        wwhe3  1
        swhe1  1
        swhe2  1
        swhe3  1
        sbar1  1
        sbar2  1
        sbar3  1
        trap1  1
        trap2  1
        trap3  1
        oats1  1
        oats2  1
        oats3  1
        silag  1
        sbeet  1
        fall   1
;
TABLE TECON(j, MACON) Technological constraints
                maxfa   maxtr   maxsb   maxwhe  maxbar   maxoat
        wwhe1
        wwhe2
        wwhe3
        swhe1
        swhe2
        swhe3
        sbar1
        sbar2
        sbar3
        trap1            1
        trap2            1
        trap3            1
        oats1
        oats2
        oats3
        silag
        sbeet                    1
*FALLOW DISABLED
        fall       0
;
```

```
TABLE MICON(j, MIC)  Minimum constraints
           minfall
      wwhe1
      wwhe2
      wwhe3
      swhe1
      swhe2
      swhe3
      sbar1
      sbar2
      sbar3
      trap1
      trap2
      trap3
      oats1
      oats2
      oats3
      silag
      sbeet
      fall     1
;
* VARIABLE INITIAL VALUES

PARAMETERS
N1(j) normal or prof max N dose for individual quad crops without constr
N2(j) normal or prof max N dose for individual mitsch crops without constr
Y1(j) prof max yield levels for quadratic crops
Y2(j) prof max yield levels for mitscherlich crops
INIT(j) unconstrained optimal N fertilizer dose
YIELD(j) unconstrained optimal yield levels ;
;
* Calculating unconstrained optimal N manually for solver initial values
N1(j) $(YC(j,"a") AND YC(j, "b") AND YC(j, "c") NE 0)
           =  [N_PRICE(j)/(C_PRICE(j)) - YC(j,"b")] / [2 * YC(j,"c")]   ;

N2(j) $(YC(j,"m") AND YC(j, "k") AND YC(j, "d") NE 0)
           =  log[(C_PRICE(j) * YC(j, "m")* YC(j, "k") * YC(j, "d"))
                / N_PRICE(j)] / YC(j, "d")   ;

Y1(j) $(YC(j,"a") AND YC(j, "b") AND YC(j, "c") NE 0)
      =  YC(j,"a") + YC(j, "b") * N1(j) +  YC(j, "c") * POWER(N1(j),2) ;

Y2(j) $(YC(j,"m") AND YC(j, "k") AND YC(j, "d") NE 0)
      =  YC(j,"m") * {1 - YC(j, "k") * exp[-(YC(j,"d")* N2(j))]} ;

* To prevent fallow from beeing evaluated at 0
N1("fall") = 1 ;

* Summing up over  different yield function specifications
           INIT(j)  = N1(j) + N2(j);
           YIELD(j) = Y1(j) + Y2(j)  ;
```

```
* initial levels based on average allocation (2003-2001) in the area
PARAMETER      X_INIT(j)  calculating the N-Baseload from unconstrained
/
        wwhe1  0.5
        wwhe2  0.5
        wwhe3  0.5
        swhe1  3.7
        swhe2  3.7
        swhe3  3.7
        sbar1  3.2
        sbar2  3.2
        sbar3  3.2
        trap1  0.7
        trap2  0.7
        trap3  0.7
        oats1  2
        oats2  2
        oats3  2
        silag  2.7
        sbeet  0.6
        fall   5.1
/
* area between different farming techniques is allocated equally
;
* SOME SENSITIVITY ANALYSIS-----------------------------------------------

*C_PRICE(j) = C_PRICE(j)* 0.9  ;
*C_PRICE(j) = C_PRICE(j)*2  ;
*X_INIT(j) = 0
*N_PRICE(j)=  N_PRICE(j)* 1.5 ;

* ENDOGENEOUS VARIABLES ----------------------------------------------
VARIABLES
        Z      Total annual producer surplus for modelling period euro
        X(j)    Activity level (ha)
        N(j)    N dose to crop j (kg per ha)
        B(j)    Buffer
*Based on the region's water ways required for env subsidies and allowed for CAP
*So 3 types of buffer area based on subsidy eligibility
        B1(j)   Buffer area
        B2(j)   Buffer area \% of 38ha when losing CAP
        B3(j)   Buffer area \% of 38ha classified as buffer zone
        cropsubsidy(j) crop specific subsidies
        MAX
;
FREE VARIABLE Z ;
POSITIVE VARIABLES X, N, B, B1, B2, B3, cropsubsidy ;

* VARIABLE INITIALIZATION AND BOUNDS -------------------------------
* Initial activity level is set to profit maximising level
* N cannot exceed profit max level +10\%  (limit for solver performance purpose)
*COMMON BOUNDS & INTIAL VALUES
```

```
N.L(j)          = INIT(j)            ;
N.up(j)         = 1.1 * INIT(j)        ;
*N.lo(j)        = 0.2 * INIT(j)        ;
N.fx("fall")    = 0.0000001           ;
*X.L(j)         = X_INIT(j)           ;
X.up("fall")    = 38                  ;
X.up(j)         = 38                  ;
*Silage limit as the current land share
X.up("silag")   =  2.7                ;
*Buffer zone bound depend on the subsidy system, below different options
*------------------------------------------
* FREE MARKET
*B.up(j)         = 0.1             ;
*B1.up("fall")    = 0.000001        ;
*B3.up("fall")    = 0.000001        ;
*B1.lo(j)        = 0.000001        ;
*B2.fx(j)         = 0              ;
*B3.lo(j)        = 0.000001        ;
*------------------------------------------
* INCOME SUBSIDIES 2003
*B.lo(j)         = 0.000001        ;
*B.up(j)         = 0.005829969      ;
*B3.up(j)    = 0.03        ;
*B1.up("fall")    = 0.000001        ;
*B3.up("fall")    = 0.000001        ;
*B2.fx(j)         = 0              ;
*B2.L(j)         = 0              ;
*------------------------------------------
* INCOME & ENVIRONMENT SUBSIDIES 2003
*B.lo(j)         = 0.004688474      ;
*B1.lo(j)        = 0.000001        ;
*B1.up(j)         = 0.005829969      ;
*B1.up("fall")    = 0.000001        ;
*B3.up("fall")    = 0.000001        ;
*B2.fx(j)         = 0              ;
*B3.up(j)        = 0.03            ;
*CONSIDER *B3.lo(j)        = 0.024333213       ;
*------------------------------------------
* CAP REFORM 2006 (NO ENVIRONMENTAL SUBSIDIES)
*0.6m * main ditch and water ways
B1.lo(j)         = 0.001752897      ;
B.up(j)         = 0.5              ;
* 1m * main ditch and 3m * water ways
B1.up(j)        = 0.005829969      ;
B1.up("fall")    = 0.001752897      ;
B3.up(j)        = 0.5              ;
B3.up("fall")    = 0.000001        ;
B2.fx(j)        = 0              ;
B2.L(j)         = 0              ;
*------------------------------------------
* Minumum is required for base measure environmental subsidies,
*any more looses subsidies
```

```
*B1.up(j)       = 0.005829969       ;
*B1.lo(j)      = 0.004688474       ;
* FREE MARKET & NO ENVIRONMENTAl SUBSIDIES
*B1.lo(j)      = 0.000001          ;
*B2.lo(j)      = 0.005829969      ;
*B2.up(j)       = 0.024333213      ;
*B2.up(j)        = 0               ;
* Minumum is required for buffer subsidies
*B3.lo(j)      = 0.000001          ;
* ENVIRONMENTAL SUBSIDY REQUIREMENT
*B.up("fall")    = 0.004688474        ;

* EQUATION DECLARATION ----------------------------------------------------------
*Different versions of the objective function to compare model performance
EQUATIONS
        SIMPLE        objfunc for max profits (only one buffer type)
        BASE          objfunc for max profits (baseline subsidies)
        CAPREF        objfunc for max profits (CAP reform)
        EBSUBS        objfunc for max profits (no environmental subs)
        NOSUBS        objfunc for max profits (no subsidies)
        TECH(MACON)     max (technological & political) constraints
        LAND(EQCON)     equality constraints (amount of land stays constant)
        MINFAL(MIC)     minimum constraints
        LOADN         the restricted N load
        LOADP         the restricted P load
        BUFUP1        buffer constraint balance for buffer strip
        BUFUP2        buffer constraint balance for limits
        BUFUP3        buffer constraint balance for buffer zone
        BUFLO1        buffer constraint balance for buffer strip
        BUFLO2        buffer constraint balance for limits
        BUFLO3        buffer constraint balance for buffer zone
        BUFCOM(j)      combining buffers to one variable
*Diversifying constraints
        CROCO1        binding as long as less than 50\% of the total field
        CROCO2        binding when more than 50\% is used.
        CROCO3        binding as long as less than 50\% of the total field
        CROCO4        binding when more than 50\% is used.
        CROCO5        binding as long as less than 50\% of the total field
        CROCO6        binding when more than 50\% is used.
        TSMAX         test on smax
;
*OBJECTIVE FUNCTIONS-------------------------------------
*only one these is maximised (different specifications)

* BASE YEAR 2003 reported----------------------------------------------------
*REMEMBER sugar price, buffer constraints, fallow requirement, output lines

NOESUBS ..  Z =E=  SUM{j,  [ SUBS(j) - FIX_C(j) +
         (C_PRICE(j) - D_PRICE(j)) *
         [(YC(j, "a") +  YC(j, "b") * N(j) + YC(j, "c") * POWER(N(j),2))
         + YC(j,"m")*[1 - YC(j,"k") * exp(-(YC(j,"d")*  N(j)))]]
          - N_PRICE(j) * N(j) ]  * X(j) * (1 - [B1(j)+ B3(j)]) }
```

```
          + SUM{j,  X(j) * B1(j) * (SUBS(j)- BUFZ_C(j)) }
          + SUM{j,  X(j) * B3(j) * (LFA(j)- BUFZ_C(j))}     ;
* Buffer strip and zone decrease costt from field cultivation, but have their
* own costs
*--------------------------------------------------------------------------

*THE REPORTED CAP scenario------------------------------------------
*REMEMBER:*sugar price, buffer constraints, fallow requirement, output lines
CAPREF ..  Z =E=  SUM{j,  [ SUBS06(j) - FIX_C(j) +
         (C_PRICE(j) - D_PRICE(j)) *
         [(YC(j, "a") +  YC(j, "b") * N(j) + YC(j, "c") * POWER(N(j),2))
         + YC(j,"m")*[1 - YC(j,"k") * exp(-(YC(j,"d")*  N(j)))]]
          - N_PRICE(j) * N(j) ]  * X(j) * (1 - [B1(j)+ B3(j)]) }
          + SUM{j,  X(j) * B1(j) * (SUBS06(j)- BUFZ_C(j))}
          + SUM{j,  X(j) * B3(j) * (LFA06(j) - BUFZ_C(j))}

* Constraints for cereal and oil seed plant subsidy with diversification 2006
TSMAX.. MAX =E= SMAX{j, [X(j) - 19]or 0}*50                 ;
CROCO1.. SUM{sb,cropsubsidy(sb)} =L= SUM{sb,50 * X(sb)}      ;
CROCO2.. SUM{sb,cropsubsidy(sb)} =L= 50 * 0.5 * farmsize     ;
CROCO3.. SUM{wh,cropsubsidy(wh)} =L= SUM{wh,50 * X(wh)}      ;
CROCO4.. SUM{wh,cropsubsidy(wh)} =L= 50 * 0.5 * farmsize     ;
CROCO5.. SUM{tr,cropsubsidy(tr)} =L= SUM{tr,50 * X(tr)}      ;
CROCO6.. SUM{tr,cropsubsidy(tr)} =L= 50 * 0.5 * farmsize     ;

*---------------------------------------------------------------------------

*Alternative objective functions

EBSUBS ..  Z =E=  SUM{j,  [
         (C_PRICE(j) - D_PRICE(j)) *
         [(YC(j, "a") +  YC(j, "b") * N(j) + YC(j, "c") * POWER(N(j),2))
         + YC(j,"m")*[1 - YC(j,"k") * exp(-(YC(j,"d")*  N(j)))]]
          - N_PRICE(j) * N(j)  ]  * X(j) * (1 - [B1(j)+ B3(j)]) } +
            SUM{j,  X(j) * (ESUBS(j) + SUBS(j))} - SUM{j,  X(j) *  FIX_C(j)} ;

NOSUBS ..  Z =E=  SUM{j,  [ - FIX_C(j) +
         (C_PRICE(j) - D_PRICE(j)) *
         [(YC(j, "a") +  YC(j, "b") * N(j)
         + YC(j, "c") * POWER(N(j),2))
         + YC(j,"m")*[1 - YC(j,"k") * exp(-(YC(j,"d")*  N(j)))]]
          - N_PRICE(j) * N(j)  ]  * X(j) * (1 - [B1(j)+ B3(j)]) }
          ;

SIMPLE ..  Z =E=  SUM{j,  [SUBS(j) - FIX_C(j) +
         (C_PRICE(j) - D_PRICE(j)) *
         [(YC(j, "a") +  YC(j, "b") * N(j)
         + YC(j, "c") * POWER(N(j),2))
         + YC(j,"m")*[1 - YC(j,"k") * exp(-(YC(j,"d")* N(j)))]]
          - N_PRICE(j) * N(j)  ] * (1 - B(j)) * X(j) }
         ;
```

```
*RESOURCE AND TECHNOLOGY CONSTRAINTS -----------------------------

 TECH(MACON) .. SUM{j, TECON(j, MACON) * X(j) } =L= VALCON(MACON)  ;
 LAND(EQCON) .. SUM{j, BACON(j, EQCON) * X(j) } =E= HACON(EQCON)   ;
 MINFAL(MIC) .. SUM{j, MICON(j, MIC) *   X(j) } =G=
*fallow for 2003
 farmsize * fallreq03
*fallow for 2006
*farmsize * fallreq06
         ;

*BUFFER CONSTRAINTS-------------------------------------------------
* Right hand side already divided by hundred
* Upper limit to strips still valid for CAP subsidies, parametrit113:BUFFER:O4
BUFUP1..   SUM{j, B1(j)* X(j)} =L= 0.005829969 * farmsize                  ;

*BUFUP2..   SUM{j, B2(j)* X(j)} =L= 0.024333213 * farmsize                  ;
*upper limit of (from regional environmental centres)
* buffer zone constraint  parametrit113:BUFFER:Q8
BUFUP3..   SUM{j, B3(j)* X(j)} =L= 0.024170031 * farmsize                  ;

*minimum buffer strip (required for CAP subsidy compliance)
BUFLO1..   SUM{j, B1(j)* X(j)} =G= 0.001752897 * farmsize                  ;
*BUFLO2..   SUM{j, B2(j)} =G= 0.00                                ;
*BUFLO3..   SUM{j, B3(j)} =G= 0.00                                ;
*BUFLO3..   SUM{j, B3(j)} =G= 0.024333213                               ;

*upper constraint for total buffer capacity of strips and zones
BUFCOM(j).. B(j) =E= B1(j) + B3(j)                                 ;

* Load constraints ------------------------------------------
LOADN .. SUM{j,
        [
         [(1 - B(j)**0.2)* SF + DF ] * LEACH(j, "NIC") *
         EXP[0.71 * ((1 - B(j)) * N(j) / N_N(j) - 1 )]
        ] * X(j) } * RETENTION
        =L= B_NLOADt0 * (1 - RED_L) ;

LOADP.. SUM{j,
       [
*         DRP
          [(1 - B(j)**1.3)* SF + DF] * LEACH(j, "DRP") *
          [0.021*(STP0 + 0.01 * (1 - B(j)) * (P_RATIO(j) * N(j))) - 0.015] /100
           +
*         PP
          [(1 - B(j)**0.3) * SF + DF] * LEACH(j, "PP") *
         (250 * log[STP0 + 0.01 * (1 - B(j)) * (P_RATIO(j) * N(j))]- 150) *
         POWER(10, -6)
       ] *  X(j) }
        =L= B_PLOADt0 * (1 - RED_L)     ;

* MODEL ---------------------------------------------------------
```

```
*Choose which functions to include based on the subsidy regime
MODEL PROFMAX /
*BASE,
*TEST,
*CAPREF,
*TSMAX,
NOESUBS,
*NOSUBS,
*LOADP,
BUFCOM,
BUFLO1,
BUFUP1,
BUFUP3,
MINFAL,
*CROCO1, CROCO2, CROCO3, CROCO4, CROCO5, CROCO6,
TECH, LAND,  LOADN  /;

SOLVE PROFMAX using NLP maximising Z ;

* ECOSYSTEM INDICATORS ------------------------------------------

PARAMETER BASENLOAD amount of N leaching from fields ;
PARAMETER BASEPLOAD amount of P leaching from fields ;
PARAMETER NETNLOAD share of N-load reaching waterways ;
PARAMETER NETPLOAD share of P-load reaching waterways ;

BASENLOAD = SUM{j,
        [
         [(1 - B.L(j)**0.2)* SF + DF ] * LEACH(j, "NIC") *
         EXP[0.71 * ((1 - B.L(j)) * N.L(j) / N_N(j) - 1 )]
        ] * X.L(j) }  ;
BASEPLOAD = SUM{j,
        [
         [(1 - B.L(j)**1.3) * SF + DF] * LEACH(j, "DRP") *
         [0.021*(STP0 + 0.01 * (1 - B.L(j))*
         (P_RATIO(j) * N.L(j))) - 0.015]/100
          +
         [(1 - B.L(j)**0.3) * SF + DF] * LEACH(j, "PP") *
         (250 * log[STP0 + 0.01 * (1 - B.L(j))*
         (P_RATIO(j) * N.L(j))]- 150) *
         POWER(10, -6)
        ] * X.L(j) }                    ;
NETNLOAD =  BASENLOAD * RETENTION            ;
NETPLOAD =  BASEPLOAD * RETENTION            ;
* Initially assumed that all of the N contributes to the N stock in water
* Initialise Baseload for iterations

B_NLOADt0 = NETNLOAD                       ;
B_PLOADt0 = NETPLOAD                       ;

* Implementation of N-reduction policy------------------------------------
```

```
$libinclude xlexport

SET COUNT iterative increases in N load restriction / 1-1 * 1-32 /     ;
SCALAR c     ;
      c = 1 ;
* Report writer parameters------------------------------------------------
PARAMETER      REDPRO(COUNT)      Abatement percentage reduction from Pmax  ;
PARAMETER      PROFIT(COUNT)      Profits                          ;
PARAMETER      redcost(COUNT)     Reduction cost                   ;
PARAMETER      loadred(COUNT)     Load reduction                   ;
PARAMETER      redcostha(COUNT)    Reduction cost per ha                ;
PARAMETER      loadredha(COUNT)    Load reduction per ha                ;
PARAMETER      redcostreg(COUNT)   Reduction cost for the study region      ;
PARAMETER      loadredreg(COUNT)   Load reduction for the study region      ;
PARAMETER      BUFFER(COUNT, j)    Buffer width (proportion of every ha     ;
PARAMETER      BUFFER1(COUNT, j)   Buffer width (proportion of every ha     ;
PARAMETER      BUFFER2(COUNT, j)   Buffer width (proportion of every ha     ;
PARAMETER      BUFFER3(COUNT, j)   Buffer width (proportion of every ha     ;
PARAMETER      TBUFFER(COUNT, j)   Total buffer proportion                ;
PARAMETER      NFERT(COUNT)       Amount of N ferlizer used             ;
PARAMETER      PFERT(COUNT)       Amount of P ferlizer used             ;
PARAMETER      NLOAD(COUNT)       Amount of N leached                   ;
PARAMETER      PLOAD(COUNT)       Amount of P leached                   ;
PARAMETER      NDOSE(COUNT, j)     N dose kg per ha                     ;
PARAMETER      PDOSE(COUNT, j)     P dose kg per ha                     ;
PARAMETER      RELNDOSE(COUNT, j)  N dose as percentage of the normal N use  ;
PARAMETER      TYIELD(COUNT, j)    Yield for each plant                 ;
PARAMETER      RELYIELD(COUNT, j)  Yield as percentage of normal yield      ;
PARAMETER      LAND_SP(COUNT)      Shadow price of agricultural land        ;
PARAMETER      NLOAD_SP(COUNT)     Shadow price of N emission restrictions   ;
PARAMETER      PLOAD_SP(COUNT)     Shadow price of P emission restrictions   ;
PARAMETER      HACUL(COUNT)       Cultivated area                      ;
PARAMETER      HAALL(COUNT, j)     Allocation of hectares for crops and fall ;
PARAMETER      HAYIELD(COUNT, j)   Yield per ha                         ;
PARAMETER      B_NLOAD(COUNT)      Baseload N (check no change)            ;
PARAMETER      YIELDREV(COUNT)     Revenue from yield                    ;
PARAMETER      SUBSREV(COUNT)      Revenue from ha subsidies              ;
PARAMETER      BUFFREV(COUNT)      Revenue from buffer subsidies           ;
PARAMETER      FERTCOST(COUNT)     Cost of fertilisers                   ;
PARAMETER      FIXCOST(COUNT)      Fixed Costs                         ;
PARAMETER      DRYCOST(COUNT)      Drying costs                         ;
PARAMETER      MONOHA(COUNT)       Hectares of monoculture over half        ;
PARAMETER      MONOPEN(COUNT)      Subsidy loss due monoculture            ;
PARAMETER      crosubs(COUNT,j)    Crop specific subsidies               ;
PARAMETER      crosubsha(COUNT,j)  Crop specific subsidies per ha           ;
* Note that parameters cannot be declared in a loop-------------------------

*The loop for abatement reduction iterations
LOOP (COUNT,
*X.L(j)         = 1          ;
SOLVE PROFMAX using NLP maximising Z   ;
*Recording the solved parameters for the current member of the set "COUNT"
```

```
REDPRO(COUNT) = REDUCTION            ;
PROFIT(COUNT) = Z.L                  ;
HACUL(COUNT) = SUM {j, X.L(j)}        ;
HAALL(COUNT, j) = X.L(j)              ;
BUFFER(COUNT, j) = B.L(j)             ;
BUFFER1(COUNT, j) = B1.L(j)            ;
BUFFER2(COUNT, j) = B2.L(j)            ;
BUFFER3(COUNT, j) = B3.L(j)            ;
TBUFFER(COUNT, j)$(X.L(j) gt 0) = B1.L(j)+ B2.L(j) + B3.L(j)             ;
B_NLOAD(COUNT)= B_NLOADt0            ;
NLOAD(COUNT) = SUM {j,
                 [
                  [(1 - B.L(j)**0.2) * SF + DF] *
                   LEACH(j, "NIC") *
                  EXP[0.71 * ((1 - B.L(j))*
                     N.L(j) / N_N(j) - 1 ) ]
                 ] * X.L(j)}     ;
PLOAD(COUNT) = SUM{j,
        [
         [(1 - B.L(j)**1.3) * SF + DF] * LEACH(j, "DRP") *
         [0.021*(STP0 + 0.01 *(1 - B.L(j))*(P_RATIO(j) * N.L(j))) - 0.015] /100
          +
          [(1 - B.L(j)**0.3) * SF + DF]* LEACH(j, "PP") *
          (250 * log[STP0 + 0.01 * (1 - B.L(j))*(P_RATIO(j) * N.L(j))]- 150) *
           POWER(10, -6)
        ] * X.L(j) }  ;
TYIELD(COUNT, j)
= [YC(j, "a") + YC(j, "b") * N.L(j) + YC(j, "c") * POWER(N.L(j),2)
  + YC(j,"m")*(1 - YC(j,"k") * exp(-(YC(j,"d")* N.L(j))))]
                  * X.L(j) * (1 - B.L(j))                          ;

HAYIELD(COUNT, j)$(X.L(j) gt 0)
= {[YC(j, "a") + YC(j, "b") * N.L(j) + YC(j, "c") * POWER(N.L(j),2)
                 + YC(j,"m")*(1 - YC(j,"k") * exp(-(YC(j,"d")* N.L(j))))]
                  }                                ;
* (1 - B.L(j))]/X.L(j)               ;
RELYIELD(COUNT, j)$(YIELD(j) gt 0) = [HAYIELD(COUNT, j)/YIELD(j)]*100        ;
YIELDREV(COUNT) = SUM{j, C_PRICE(j) * [
YC(j, "a") + YC(j, "b") * N.L(j) + YC(j, "c") * POWER(N.L(j),2)+
YC(j,"m")*(1 - YC(j,"k") * exp(-(YC(j,"d")* N.L(j))))]
                  * X.L(j) * (1 - B.L(j))}                          ;
*REMEMBER TO CHANCE FOR DIFFERENT SCENARIOS!
SUBSREV(COUNT) = SUM{j,  X.L(j) *  (SUBS(j)+ ESUBS(j))}               ;
BUFFREV(COUNT) = SUM{j,  [450*0.2 -(CAP2003(j)+ NATSUB2003(j))] * 38 * B3.L(j)};
NDOSE(COUNT, j)$(X.L(j) gt 0) = N.L(j)      ;
NFERT(COUNT) = SUM{j, NDOSE(COUNT, j) * X.L(j) * (1-[B1.L(j)+ B3.L(j)]) }      ;
FERTCOST(COUNT) = SUM{j, N_PRICE(j) * N.L(j) * X.L(j) * (1-[B1.L(j)+ B3.L(j)])};
FIXCOST(COUNT) = SUM{j, FIX_C(j)* X.L(j)}      ;
DRYCOST(COUNT) = SUM{j, D_PRICE(j) * [
YC(j, "a") + YC(j, "b") * N.L(j) + YC(j, "c") * POWER(N.L(j),2)+
YC(j,"m")*(1 - YC(j,"k") * exp(-(YC(j,"d")* N.L(j))))]
                  * X.L(j) * (1 - B.L(j))  };
```

```
PDOSE(COUNT, j)$(X.L(j) gt 0) = (P_RATIO(j) * N.L(j))      ;
PFERT(COUNT) = SUM {j, P_RATIO(j) * NDOSE(COUNT, j) * X.L(j)}  ;
RELNDOSE(COUNT, j) $(X.L(j) gt 0) = [N.L(j)/N_N(j)]    ;
LAND_SP(COUNT)  =  LAND.M("ha") ;
NLOAD_SP(COUNT)  =  LOADN.M ;
*PLOAD_SP(COUNT)  =  LOADP.M ;
REDUCTION = 1 - c             ;
* Implement policy reduction requirements
*Making the N reduction constraint tighter
RED_L = REDUCTION             ;
       c = c - .02 ;
) ;
* Resolving until all members of set COUNT have been covered

*Some basic results
redcost(COUNT)= PROFIT("1-1") - PROFIT(COUNT)               ;
loadred(COUNT)= NLOAD("1-1")  - NLOAD(COUNT)                ;
redcostha(COUNT) =     redcost(COUNT)/HACUL(COUNT)          ;
loadredha(COUNT) =     loadred(COUNT)/HACUL(COUNT)          ;
redcostreg(COUNT)=     redcostha(COUNT) * studreg           ;
loadredreg(COUNT)=     loadredha(COUNT) * studreg           ;

*OPTIONAL TO REPORT PARAMETERS IN EXCEL--------
REDPRO(COUNT)$(NOT REDPRO(COUNT)) = EPS   ;
*$libinclude xlexport REDPRO FresultsNP_B_2.xls OUTPUT!b1:ag2
*$libinclude xlexport PROFIT FresultsNP_B_2.xls OUTPUT!b3:ag4
*$libinclude xlexport HACUL FresultsNP_B_2.xls OUTPUT!b5:ag6
LAND_SP(COUNT)$(NOT LAND_SP(COUNT)) = EPS ;
*$libinclude xlexport LAND_SP FresultsNP_B_2.xls OUTPUT!b7:ag8
NLOAD_SP(COUNT)$(NOT NLOAD_SP(COUNT)) = EPS ;
*$libinclude xlexport NLOAD_SP FresultsNP_B_2.xls OUTPUT!b9:ag10
NLOAD(COUNT)$(NOT NLOAD(COUNT)) = EPS     ;
*$libinclude xlexport NLOAD FresultsNP_B_2.xls OUTPUT!b11:ag12
NFERT(COUNT)$(NOT NFERT(COUNT)) = EPS     ;
*$libinclude xlexport NFERT FresultsNP_B_2.xls OUTPUT!b13:ag14
TYIELD(COUNT, j)$(NOT TYIELD(COUNT, j)) = EPS     ;
*$libinclude xlexport TYIELD FresultsNP_B_2.xls OUTPUT!a18:r50    ;
RELYIELD(COUNT, j)$(NOT RELYIELD(COUNT, j)) = EPS     ;
*$libinclude xlexport RELYIELD FresultsNP_B_2.xls OUTPUT!a58:r90  ;
NDOSE(COUNT, j)$(NOT NDOSE(COUNT, j)) = EPS     ;
*$libinclude xlexport NDOSE FresultsNP_B_2.xls OUTPUT!a95:r127    ;
PDOSE(COUNT, j)$(NOT PDOSE(COUNT, j)) = EPS     ;
*$libinclude xlexport PDOSE FresultsNP_B_2.xls OUTPUT!a136:r168
*PLOAD_SP(COUNT)$(NOT PLOAD_SP(COUNT)) = EPS ;
*$libinclude xlexport PLOAD_SP FresultsNP_B_2.xls OUTPUT!b130:ag131
PLOAD(COUNT)$(NOT PLOAD(COUNT)) = EPS     ;
*$libinclude xlexport PLOAD FresultsNP_B_2.xls OUTPUT!b132:ag133
HAALL(COUNT, j)$(NOT HAALL(COUNT, j)) = EPS     ;
*$libinclude xlexport HAALL FresultsNP_B_2.xls OUTPUT!a171:s203
TBUFFER(COUNT, j)$(NOT TBUFFER(COUNT, j)) = EPS     ;
*$libinclude xlexport TBUFFER FresultsNP_B_2.xls OUTPUT!a206:s238
BUFFER(COUNT, j)$(NOT BUFFER(COUNT, j)) = EPS     ;
```

```
*$libinclude xlexport BUFFER FresultsNP_B_2.xls OUTPUT!a206:s238
BUFFER1(COUNT, j)$(NOT BUFFER1(COUNT, j)) = EPS     ;
*$libinclude xlexport BUFFER1 FresultsNP_B_2.xls OUTPUT!a276:s308
*$libinclude xlexport YIELDREV FresultsNP_B_2.xls OUTPUT!b310:ag311
*$libinclude xlexport SUBSREV FresultsNP_B_2.xls OUTPUT!b312:ag313
*$libinclude xlexport BUFFREV FresultsNP_B_2.xls OUTPUT!b314:ag315
*$libinclude xlexport FERTCOST FresultsNP_B_2.xls OUTPUT!b319:ag320
*$libinclude xlexport FIXCOST FresultsNP_B_2.xls OUTPUT!b321:ag322
*$libinclude xlexport DRYCOST FresultsNP_B_2.xls OUTPUT!b323:ag324

* SCENARIO SUMMARY REPORTING-----------------------------------------
* BASE 2003
$libinclude xlexport REDPRO NP_RESULT_COMPARISON.xls PROFITS!b1:ag2
$libinclude xlexport PROFIT NP_RESULT_COMPARISON.xls PROFITS!b4:ag5
$libinclude xlexport NLOAD NP_RESULT_COMPARISON.xls LOAD!b2:ag3
$libinclude xlexport PLOAD NP_RESULT_COMPARISON.xls LOAD!b11:ag12
$libinclude xlexport TYIELD NP_RESULT_COMPARISON.xls YIELD!a3:s35
$libinclude xlexport HAALL NP_RESULT_COMPARISON.xls HA!a3:s35
$libinclude xlexport NDOSE NP_RESULT_COMPARISON.xls FERTILIZATION!a3:s35
$libinclude xlexport BUFFER NP_RESULT_COMPARISON.xls BUFFER!a3:s35
$libinclude xlexport RELYIELD NP_RESULT_COMPARISON.xls YIELD!a110:s142
*--------------------------------------------------------------------
* FREE MARKET 2003
*$libinclude xlexport REDPRO NP_RESULT_COMPARISON.xls PROFITS!b1:ag2
*$libinclude xlexport PROFIT NP_RESULT_COMPARISON.xls PROFITS!b6:ag7
*$libinclude xlexport NLOAD NP_RESULT_COMPARISON.xls LOAD!b5:ag6
*$libinclude xlexport PLOAD NP_RESULT_COMPARISON.xls LOAD!b14:ag15
*$libinclude xlexport TYIELD NP_RESULT_COMPARISON.xls YIELD!a38:s70
*$libinclude xlexport HAALL NP_RESULT_COMPARISON.xls HA!a38:s70
*$libinclude xlexport NDOSE NP_RESULT_COMPARISON.xls FERTILIZATION!a38:s70
*$libinclude xlexport BUFFER NP_RESULT_COMPARISON.xls BUFFER!a38:s70
*--------------------------------------------------------------------
* CAP REFORM 2006
*$libinclude xlexport REDPRO NP_RESULT_COMPARISON.xls PROFITS!b1:ag2
*$libinclude xlexport PROFIT NP_RESULT_COMPARISON.xls PROFITS!b8:ag9
*$libinclude xlexport NLOAD NP_RESULT_COMPARISON.xls LOAD!b8:ag9
*$libinclude xlexport PLOAD NP_RESULT_COMPARISON.xls LOAD!b17:ag18
*$libinclude xlexport TYIELD NP_RESULT_COMPARISON.xls YIELD!a73:s105
*$libinclude xlexport HAALL NP_RESULT_COMPARISON.xls HA!a73:s105
*$libinclude xlexport NDOSE NP_RESULT_COMPARISON.xls FERTILIZATION!a73:s105
*$libinclude xlexport BUFFER NP_RESULT_COMPARISON.xls BUFFER!a73:s105
*$libinclude xlexport RELYIELD NP_RESULT_COMPARISON.xls YIELD!t110:al142
*--------------------------------------------------------------------
*INCOME & ENVIRONMENT SUBSIDIES 2003
*$libinclude xlexport PROFIT NP_RESULT_COMPARISON.xls PROFITS!b10:ag11

*SENSITIVITY ANALYSIS OUTPUT---------------------------------------------------
*BASE
$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b3:ag4
$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b25:ag26
*BASE cprice*0.9
*$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b5:ag6
```

```
*$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b27:ag28
*BASE nprice*1.5
*$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b7:ag8
*$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b29:ag30

*CAP
*$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b13:ag14
*$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b35:ag36
*CAP cprice*0.9
*$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b15:ag16
*$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b37:ag38
*CAP nprice*1.5
*$libinclude xlexport redcostreg NP_RESULT_COMPARISON.xls RESULTS!b17:ag18
*$libinclude xlexport loadredreg NP_RESULT_COMPARISON.xls RESULTS!b39:ag40

;

*For result listing
DISPLAY REDPRO, PROFIT, HACUL, LAND_SP, NLOAD_SP, YIELD ;
DISPLAY BASENLOAD, NFERT, INIT, NLOAD  ;
DISPLAY BASEPLOAD, PDOSE, PFERT, PLOAD ;
DISPLAY HAALL, RELYIELD ;
*DISPLAY BUFFER       ;
DISPLAY TBUFFER, BUFFER1, BUFFER2, BUFFER3, B_NLOAD, NLOAD, B_NLOADt0,NETNLOAD ;
DISPLAY YIELD, Y1, Y2, N1, N2, INIT, RELNDOSE ;
*DISPLAY MAX.L, MONOPEN, HAALL, MONOHA ;
*DISPLAY crosubs, crosubsha ;
DISPLAY redcost, loadred, redcostha, loadredha, redcostreg, loadredreg ;
```

Finnish nutrient abatement model for crop farm (heterogeneous load parameters)

$Title Finnish nutrient abatement model for crop farm (heterogeneous load parameters)
$ontext
This file imports some general land use and distribution data from Access/GIS and load parameters from GDX file produced in Sum_Aura_load .gms File. Yield parameters are imported from Aurajoki_yields_45param.gdx. Some of the parameter data is given in scalars and parameters before the variable declarations in GAMS
The subset of non cereal functions, as well as some of the constraints for crops/fallow, are defined in GAMS. Initial values and upper limits of nitrogen fertilisation are based on an analytical solution with given prices.
Objective functions of the model maximise the profits of the reprentative farmer of the watershed. Crop yield (per ha) is defined in a separate function. Nitrogen fertilisation (which determines fixed phophorus fertilisation) and the allocation of farm land (including crop type and tillage)are the decision variables. The objective is subject to constraint keeping the slope, soil and Pst of the fields fixed, while allowing change of tillage and crops. Load functions are based on ICECREAM model results and determine soil and slope dependent erosion and runoff, which are combined to PP and DRP predecting equations of R.Uusitalo. Nitrogen is predected directly from the ICECREAM model.
First solve calculates the optimal results without the load constraints. Then, load constraints are made binding by tightening the constraint in a loop with 2\% iterations. Then, the unconstrained base line is resolved and loop is calculated for phosphorus.
OLS estimates for abatmenet cost functions are done in a separate file.
The main results from this file are:
- Aura_N_redcost(count) (abetement costs for 2-60\% reduction)
- baseTP and baseN loads (profit maximsing unconstrained load levels)
- Optimal levels for decision variables
This file should have GDX= Aura_results_het.gdx for OLS estimates of the cost functions.
$offtext

```
*Some solver options:
$eolcom //
option iterlim=999999999; // avoid limit on it
option reslim=600; // timelimit for sol
*option optcr=0.0; // gap tolerance
*option solprint=ON; // include solution
*option limrow=100; // limit number of r
*option limcol=100; // limit number of c

*option decimals = 2;
*Switch default solution printing off.
*option solprint = off;
//-

*DATA IMPORT FROM ACCESS/GIS----------------------------------------------------
* Requires mdb2gms  (http://www.gams.com/dd/docs/tools/mdb2gms.pdf)
* This is multiquery batch version
* Set correct file path for the database location I="????"

$onecho > cmd.txt
I="D:\GIS\SAMA_database2003_INPUT.mdb"
X=Aura_data_param.gdx
```

```
Q1=select j from setj order by id
s1=j
Q2=select flw from setflw
s2=flw
Q3=select slp from 6slopes
s3=slp
Q4=select sltclass from soil4set
s4=slt
Q6=select Pst from set_Aura_Pst
s6=Pst
Q7=select crop_id from crop
s7=croptype
Q8=select con from setcon
s8=con
Q9=select fcon from setfcon
s9=fcon
Q10=select Pfert from setPfert
s10=pfert
Q11=select Nfert from setNfert
s11=nfert
Q12=select fn from Aura_cropfarm_area
s12=fn
Q20=select j, price2009 from crop_price2009
p20=c_price
Q21=select j, FixedCost2008 from FixedCost2008
p21=fix_c
Q19=select flw, FixedCostFallow2008 from FixedCostFallow2008
p19=fall_cost
Q22=select flw, fcon, facon from FACON
p22=facon
Q24=select j, Aura_mingrass from setj
p24=mingrass
Q26=select slp, sltclass, Pst, Area from GROUPBY_AuraCropFarmDataDistr
p26=land_distr
Q28=select j, slp, sltclass, Pst, Area from AuraCropFarmCropDistr
p28=crop_distr
Q29=select flw, slp, sltclass, Pst, Area from AuraCropFarmFallowDistr
p29=fallow_distr
Q27=select Pst, Pst_no from set_Aura_Pst
p27=Pst_no
Q30=select j, national_subsidy from crop_farm_subsidies_2009_A
p30=ntlsubs
Q31=select j, cap_subsidy from crop_farm_subsidies_2009_A
p31=capsubs
Q32=select j, env from crop_farm_subsidies_2009_A
p32=envsubs
Q33=select j, lfa from crop_farm_subsidies_2009_A
p33=lfasubs
Q34=select flw, capfallow from crop_farm_fallow_subsidies_2009_A
p34=capfallsubs
Q35=select flw, envfallow from crop_farm_fallow_subsidies_2009_A
p35=envfallsubs
```

```
Q36=select flw, lfafallow from crop_farm_fallow_subsidies_2009_A
p36=lfafallsubs
Q40=select fn, m2 from Aura_cropfarm_area
p40=Aura_cropfarm_areas
Q41=select j, con, tecon from tecon
p41=tecon
Q42=select SUM_F_AREA from AuraBuffer15m
p42=AuraBufferArea15m
Q43=select SUM_F_AREA from AuraBuffer3m
p43=AuraBufferArea3m
Q44=select SUM_F_AREA from AuraBuffer1m
p44=AuraBufferArea1m
Q48=select SUM_F_AREA from AuraBuffer160m
p48=AuraBufferArea160m
Q49=select SUM_F_AREA from AuraBuffer500m
p49=AuraBufferArea500m
Q45=select ReductionCoefficient from wetlandNutrientReductionP
p45=wtldCoefficientP
Q46=select ReductionCoefficient from wetlandNutrientReductionN
p46=wtldCoefficientN
Q47=select F_AREA from AuraCatchmentTotalArea
p47=CatchmentTotalArea
$offecho
$call =mdb2gms \atcmd.txt
* SETS -----------------------------------------------------------
SETS

    flw fallow types   gfs1 gfs2 gfs3 gf1n gfn2 gfn3 bfs1 bfs2 bfs3 bf1n bfn2 bfn3 ffn g=green b=bare
s=subsidy eligible n=no subsidies ff=forest
    slp slope classes  1=0-0.5 2=0.5-1 3=1-2 4=2-3 5=3-6 6=>6
    slt soil types  1=HsS 2=HtS 3=HHt 4=KHt
    j farmed crops  f1-6=wheat f7-12=rye f13-18=barley f19-21=oats f25-27=peas f28-33=potato f34-
36=sugar beet f37-42=rape f43-45=silage
    fn farms
    count iterative increases in P load restriction / 1-1 * 1-32 /
    feas(count) feasible solution subset of count has not been defined here   / 1-1 * 1-32 /
    y yield function coefficients
/
    a     Quadratic parameter a intercept
    d     Quadratic parameter b positive slope
    c     Quadratic parameter c for diminishing marginal
    m      Mitscherlich parameter m
    k     Mitscherlich parameter k
    b     Mitscherlich parameter b
    pyc1   yield coefficient for P fertilisers
    pyc2   yield coefficient for P fertilisers
    pyc3   negative yield coefficient of P stock
    pyc4   positive yield coefficient of P stock
    pcon   constant increase in P yield
    pcoe   crop type yield coefficient for P
    sry    saarelas fodder unit
    tillfac tillage yield factor
```

```
/
        lc load coefficients for P and N
/        r runoff
        e erosion
        o omega leaching based on different technology
        v delta leaching based on different technology
        nic nitrogen load coefficient
        psini initial phosphorus level
        PStm1 dummy for last period P level
/
        con max an min crop resource constraints
        fcon  various fallow maximum subsidy constraints
       eqcon equalised constraints
       ha arable land available for model farm
/
        Pst Soil Phosphorus levels
        pfert P fertilisation levels in ICECREAM data
        nfert N fertilisation levels in ICECREAM data
;
*        soil
alias (soil, slt) ;
parameters
*LAND PARAMETERS---------------------------------------------------------------
        Aura_cropfarm_areas(fn)        Field areas of farms within Aurajoki watershed with no production
animals
        crop_distr(j,slp,slt,Pst)      Crop types on different soil slope and P status land    (different from
calibration file classes)
        fallow_distr(flw,slp,slt,Pst)  Crop types on different soil slope and P status land  (different from
calibration file classes)
        land_distr(slp,slt,Pst)        Land distribution in m2 to ha different soil and slope classes on farm scale
        land_distr_share(slp,slt,Pst)  Land distribution percent share to different soil and slope classes on farm
scale
        land_distr_farm(slp,slt,Pst)   Land distribution in ha to different soil and slope classes on farm scale
        initCropShare(j,slp,slt,Pst)   Share of crops from total CROP land 2003
        N_cropdistr_hom(count,j,slp,slt,Pst)   Crop distribution (for homogeneous parameter values)
        initFallowShare(flw,slp,slt,Pst)       Share of fallow from total FALLOW land 2003
*Coeficient for the constraint matrix
        facon(flw, fcon)               Constraint matrix left-hand side (fallow)
        tecon(j, con)                  Constraint matrix left-hand side
        minfall(flw)                   Constraint matrix right-hand side (minimum fallow)
        maxfall(flw)                   Constraint matrix right-hand side (maximum fallow)
*LOAD PARAMETERS---------------------------------------------------------------
        runoff(j,slp,slt)              Runoff in mm per ha
        sloss(j,slp,slt)               Soil loss in kg ha-1 from crop area
        runoff_flw(flw,slp,slt)        Runoff in mm per ha    from fallow
        sloss_flw(flw,slp,slt)         Soil loss in kg ha-1   from fallow
        TPloadIC_flw(flw,slp,slt,Pst,pfert)  Total P load ha-1 from fallow (according icecream)
        NloadIC(j,slp,slt,nfert)       N load ha-1 from crop area        (according icecream)
        NloadIC_flw(flw,slp,slt)       N load ha-1 from fallow          (according icecream)
        loadCoeffN(j,slp,slt)          Constant for crop nitrogen load
        loadCoeffN_flw(flw,slp,slt)    Constant for fallow nitrogen load
        Ncoeff_frt_b(j,slp,slt)        Coefficient b for y= b * exp(x*c) in nitrogen load function
```

```
      Ncoeff_frt_c(j,slp,slt)        Coefficient c for y= b * exp(x*c) in nitrogen load function
      leach(j,lc)                    Load coefficients  (from previous model versions)
*       loadCoeff_N(j,slp,soil)        Constant for crop nitrogen load   (obsolete 20 soil type set)
*       loadCoeff_N_flw(flw,slp,soil)   Constant for fallow nitrogen load (obsolete 20 soil type set)
*       rnoff(j,slp,slt)              Runoff in mm per ha              (obsolete 20 soil type set)
*       erosion(j,slp,soil)           Erosion from crops (in kg ha-1)   (obsolete 20 soil type set)
*       rnoff_flw(flw,slp,soil)        Runoff from fallow              (obsolete 20 soil type set)
*       erosion_flw(flw,slp,soil)      Erosion from fallow (in kg ha-1 ) (obsolete 20 soil type set)
      sloss_m(j)                    Mean erosion in kg from each crop
      runoff_m(j)                   Runoff of mean slope and soil (HtS)
      runoff_flw_m(flw)             Mean runoff from fallow
      sloss_flw_m(flw)              Mean soil loss in kg ha-1 from fallow
      pmax(Pst,j,slt)               Maximum base fertilisation level as intial P level
      nmax(j,slt)                   Maximum base fertilisation level as intial N level
      Pst_no(Pst)                   soil P status mg per l
*FOR BETTER INITIAL ALLOCATION WHEN P IS CONSTRAINED, TAKEN FROM 1-5 ITERATION ->
CONSEQUETIVE RUNS WONT WORK
      best_P_profit_land(j,slp,slt,Pst)      Land allocation of 1-5 iteration of P load which was the best when
equal intial values were given
      best10_P_profit_land(j,slp,slt,Pst)     Land allocation of 1-10 iteration of P load which was the best
when equal intial values were given
*ECONOMIC PARAMETERS
      c_price(j)            Product prices
      fix_c(j)              Fixed costs euro per ha of field
      fall_cost(flw)        Fixed costs of fallow
      ntlsubs(j)            National ha subsidies for crops
      capsubs(j)            EU ha subsidies for crops
      envsubs(j)            Environmental subsidy for crop
      lfasubs(j)            LFA subsidy for crop
      mingrass(j)           Demand of grass modeled as minimum land constraint
      capfallsubs(flw)      CAP subsidy for fallow
      envfallsubs(flw)      Environmental subsidy for fallow
      lfafallsubs(flw)      LFA subsidy for fallow
;
scalars
      CatchmentTotalArea
      farm_size             Average crop farm size for Aurajoki
      Aura_farm_no          Number of farms with
      grass_land_share      Share of grass land of total modeled area
      reduction             Dummy reduction                     /-10/
      red_l                 Abatement rate                      /0/
      tpinit                Check tp in the beginning
      bio_coef              Conversion factor from PP to algae available
      retention             No retention is assumed (included in the ICECREAM parameters)  /1/
      PsAvg                 Average initial soil P stock mg per l
      PStm                  Initial soil P stock mg per l  if variable
      baseN_load            N load which is scaled down to VEPS levels
      baseP_load            P load which is scaled down to VEPS levels
      veps_N_2002           28 total N load (kg) for 2002
      veps_TP_2002          28 total P load (kg) for 2002
      TotalNrevised         Total nitrogen load from the representative farm (ICECREAM N load functions)
```

TotalN_MYTVAS_Hi Total nitrogen load from the representative farm (given high N use of MYTVAS study and ICECREAM N load functions)
TotalN_MYTVAS_Lo Total nitrogen load from the representative farm (given low N use of MYTVAS study and ICECREAM N load functions)
MYTVASmodTP Total phophprus load from the representative farm (given low P use of MYTVAS study and ICECREAM erosion and runoff functions)
drp_load Dissolved reactive phosphorus load from the representative farm
pp_load Particle phosphorus load from the representative farm
DRP_TP_share Share of drp of total p
*Scaling was given up, so these woul not be needed
baseTPload TP load which is NOT scaled down to VEPS levels and the basis for first non-restricted model solution
baseNload N load which is NOT scaled down to VEPS levels and the basis for first non-restricted model solution
*Model runs sensitive to these, should be implemented in the homogeneous version too
baseTPmulti Multiplier for base TP load to ensure that the intial contstraint is not a binding one / 2 /
baseNmulti Multiplier for base N load to ensure that the intial contstraint is not a binding one / 3 /
modTP Modeled Total Phosphorus
veps_ha Area of modeled land
model_ha Land area covered by the model
fallow_ha Total fallow area
fallow_ha_rev Total fallow area of the watershed directly from Access with smaller sets
crop_ha Total crop area of the watershed
crop_ha_rev Total crop area of the watershed directly from Access with smaller sets
model_ha_rev Total model area of the watershed directly from Access with smaller sets
*BELOW NOT CURRENTLY USED IN THE MODEL
AuraBufferArea500m Area within 500 m of the river (could be used for buffer zone or separate load (flooding) functions)
AuraBufferArea160m Area within 160 m of the river (could be used for buffer zone or separate load (flooding) functions)
AuraBufferArea15m Area within 15 m of the river (could be used for buffer zone or separate load (flooding) functions)
AuraBufferArea3m Area within 3 m of the river (could be used for buffer zone or separate load (flooding) functions)
AuraBufferArea1m Area within 1 m of the river (could be used for buffer zone or separate load (flooding) functions)
AuraBufferArea500mShare Share of Area500m of total field area
AuraBufferArea160mShare Share of Area160m of total field area
AuraBufferArea15mShare Share of field within 15 meters of total modeled watershed farm area
AuraBufferArea3mShare Share of field within 3 meters of total modeled watershed farm area
AuraBufferArea1mShare Share of field within 1 meters of total modeled watershed farm area
wetlandInvestCostkgP Investment cost for wetland (test version) / 411 /
wetlandInvestCostkgha Investment costs per ha of wetland regional_inputdata.xls (regional farms) Majoinen 2005 Tuusula / 30611 /
wtldCoefficientP Reduction coefficient of wetland for P (constant for wetland share of total watershed area)
wtldCoefficientN Reduction coefficient of wetland for N (constant for wetland share of total watershed area)
catchmentFieldShare Share of fields from the catchment total area
;

```
*GET THE DATA FROM THE GDX FILE CREATED FROM ACCESS DATABASE
*SPECIFY THE CORRECT PATH!
$gdxin D:\gamsdir\crop_farm_2009\Aura_data_param.gdx
$load fn j flw slp slt Pst pfert nfert land_distr Pst_no
$load crop_distr fallow_distr CatchmentTotalArea
$load ntlsubs capsubs envsubs lfasubs c_price fix_c mingrass Aura_cropfarm_areas fall_cost
$load con fcon tecon facon capfallsubs envfallsubs lfafallsubs
$load AuraBufferArea500m AuraBufferArea160m AuraBufferArea15m AuraBufferArea3m
AuraBufferArea1m wtldCoefficientP wtldCoefficientN

*GET THE DATA FROM THE GDX FILE CREATE BY Load_Sum_Aura.gms
*SPECIFY THE CORRECT PATH AND REMEMER TO CHECK THAT GDX WAS CREATED!
$gdxin D:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx
*PARAMETER DATA FOR THE 20 SOIL TYPE  SET (NOT USED ANYMORE)
*$load sloss rnoff runoff_flw sloss_flw  loadCoeffN_flw baseN_load modTP
*PARAMETER DATA FOR THE 4 SOIL TYPE  SET
$load rnoff erosion rnoff_flw erosion_flw loadCoeff_N_flw baseN_load
$load Ncoeff_frt_b Ncoeff_frt_c
$load veps_ha model_ha fallow_ha crop_ha grass_land_share
$load veps_N_2002 veps_TP_2002 PsAvg pmax nmax
$load modTP bio_coef
$load TotalNrevised TotalN_MYTVAS_Hi TotalN_MYTVAS_Lo MYTVASmodTP
;
*USING THE BASELINE CROP LAND ALLOCATION OF HOGENEOUS VERSION AS THE INITIAL VALUES FOR
CROP DISTRIBUTION
*SPECIFY THE CORRECT PATH AND REMEMER TO CHECK THAT GDX WAS CREATED! (NEED RUNNING OF
HOMOGENOUS LOAD OPTIMIZATION VERSION)
$gdxin D:\gamsdir\crop_farm_2009\Aura_results_hom.gdx
$load N_cropdistr_hom

*CROP TYPE SUBSETS
sets potato(j) /f28,f29,f30,f31,f32,f33 / , peas(j) /f25, f26, f27 / , oilseed(j) / f37, f38, f39, f40, f41, f42  / ,
grass(j) / f43, f44, f45 /  , subeet(j) / f34, f35, f36 /
;

*TO IMPROVE THE BASELINE SOLUTION READ THE BEST SOLUTION FROM THE P LOOP
*$gdxin C:\gamsdir\crop_farm_2009\Aura_results.gdx
*$load best_P_profit_land best10_P_profit_land

*TRANSFER PARAMETERS TO THE SMALLER SOIL SET-----------------------------------
 loadCoeffN_flw(flw,slp,'slt1')      = loadCoeff_N_flw(flw,slp,'slt1')          ;
 loadCoeffN_flw(flw,slp,'slt2')      = loadCoeff_N_flw(flw,slp,'slt2')          ;
 loadCoeffN_flw(flw,slp,'slt3')      = loadCoeff_N_flw(flw,slp,'slt3')          ;
 loadCoeffN_flw(flw,slp,'slt4')      = loadCoeff_N_flw(flw,slp,'slt4')          ;
 runoff(j,slp,'slt1')      = rnoff(j,slp,'slt1')          ;
 runoff(j,slp,'slt2')      = rnoff(j,slp,'slt2')          ;
 runoff(j,slp,'slt3')      = rnoff(j,slp,'slt3')          ;
 runoff(j,slp,'slt4')      = rnoff(j,slp,'slt4')          ;
 sloss(j,slp,'slt1')      = erosion(j,slp,'slt1')          ;
 sloss(j,slp,'slt2')      = erosion(j,slp,'slt2')          ;
 sloss(j,slp,'slt3')      = erosion(j,slp,'slt3')          ;
```

```
 sloss(j,slp,'slt4')      = erosion(j,slp,'slt4')           ;
 runoff_flw(flw,slp,'slt1')      = rnoff_flw(flw,slp,'slt1')           ;
 runoff_flw(flw,slp,'slt2')      = rnoff_flw(flw,slp,'slt2')           ;
 runoff_flw(flw,slp,'slt3')      = rnoff_flw(flw,slp,'slt3')           ;
 runoff_flw(flw,slp,'slt4')      = rnoff_flw(flw,slp,'slt4')           ;
 sloss_flw(flw,slp,'slt1')      = erosion_flw(flw,slp,'slt1')           ;
 sloss_flw(flw,slp,'slt2')      = erosion_flw(flw,slp,'slt2')           ;
 sloss_flw(flw,slp,'slt3')      = erosion_flw(flw,slp,'slt3')           ;
 sloss_flw(flw,slp,'slt4')      = erosion_flw(flw,slp,'slt4')           ;

display loadCoeffN_flw, runoff, sloss, runoff_flw, sloss_flw       ;

*LAND PARAMETERS---------------------------------------------------------------
* calculate number of farms from GIS data
 Aura_farm_no = CARD(fn)
* calculate farm mean size in ha                              ;
 farm_size = SUM{fn,  Aura_cropfarm_areas(fn)} /  Aura_farm_no / 10000 ;
* convert land distribution parameters from m2 to ha
 land_distr(slp,slt,Pst) = land_distr(slp,slt,Pst)           /10000      ;
 crop_distr(j,slp,slt,Pst) = crop_distr(j,slp,slt,Pst)        /10000      ;
 fallow_distr(flw,slp,slt,Pst) = fallow_distr(flw,slp,slt,Pst) /10000      ;
*Field along streams with different buffer widths (in ha)
 AuraBufferArea15m = AuraBufferArea15m / 10000 ;
 AuraBufferArea3m  = AuraBufferArea3m / 10000 ;
 AuraBufferArea1m  = AuraBufferArea1m / 10000 ;
 AuraBufferArea500m = AuraBufferArea500m / 10000 ;
 AuraBufferArea160m = AuraBufferArea160m / 10000 ;

*WATERSHED AREA---------------------------------------------------------------
* calculate total watershed area covered by the model
 model_ha = SUM{(slp,slt,Pst), land_distr(slp,slt,Pst) }          ;
* calculate share of land distribution classes from the total modeled area
 land_distr_share(slp,slt,Pst)  =  land_distr(slp,slt,Pst)/model_ha ;
* scale the land distribution to reprensetative farm size
 land_distr_farm(slp,slt,Pst)   = land_distr_share(slp,slt,Pst) * farm_size          ;
* calculate share of total modeled farm land from the total catchement area
 catchmentFieldShare =  model_ha  / CatchmentTotalArea/10000 ;
* calculate shares of buffer areas from the modeled area
 AuraBufferArea500mShare = AuraBufferArea500m / model_ha ;
 AuraBufferArea160mShare = AuraBufferArea160m / model_ha ;
 AuraBufferArea15mShare = AuraBufferArea15m / model_ha ;
 AuraBufferArea3mShare  = AuraBufferArea3m / model_ha  ;
 AuraBufferArea1mShare  = AuraBufferArea1m / model_ha  ;

*LAND USE IN 2003 SUBSIDY STATISTICS---------------------------------
* calculate the total subdized crop area at the watershed    (2003)
 crop_ha_rev   = SUM{(j,slp,slt,Pst), crop_distr(j,slp,slt,Pst) }      ;
* calculate the total subdized fallow area at the watershed  (2003)
 fallow_ha_rev = SUM{(flw,slp,slt,Pst), fallow_distr(flw,slp,slt,Pst) } ;
* calculate the total modeled field area (should match with land_distr)
 model_ha_rev = crop_ha_rev + fallow_ha_rev   ;
* calculate initial land distribution for crops and fallow
```

```
 initCropShare(j,slp,slt,Pst)     =  crop_distr(j,slp,slt,Pst) / crop_ha_rev       ;
 initFallowShare(flw,slp,slt,Pst) =  fallow_distr(flw,slp,slt,Pst) / fallow_ha_rev ;

*EXPECTED LOAD FOR HA MULTIPLIED BY THE FARM SIZE AND COEFFICIENT TO MAKE IT NOT BINDING IN
INITIAL MODEL SOLUTION
* setting too large non binding intial values can affect the solver performance and worsen the solution
 baseTPload = modTP/model_ha*farm_size*baseTPmulti ;
*basetpload = 40 ;
 baseNload = baseN_load ;
 baseNload = baseNload/model_ha*farm_size*baseNmulti  ;
 tpinit = baseTPload/model_ha*farm_size  ;
display crop_ha_rev, crop_ha, model_ha, model_ha_rev ;

*RESOURCE LIMITS ---------------------------------------------------------------
parameters
        valcon(con)          resource endowment and technology constraints right side
        hacon(eqcon)           right hand side for all ha
        fallLimits(fcon)       right hand side for fallow constraints (both min and max)
;
*SETTING LIMIT CONDITIONS FOR CROPS
 hacon("ha")       = farm_size                      ;
 valcon("maxtr")   = farm_size/3                    ;
 valcon("maxsub")  = farm_size * SUM{(subeet,slp,slt,Pst),initCropShare(subeet,slp,slt,Pst)}  ;
 valcon("maxpot")  = farm_size * SUM{(potato,slp,slt,Pst),initCropShare(potato,slp,slt,Pst)}  ;
 valcon("maxgr")   = farm_size * 0.9 ;
 valcon("maxpea") = farm_size * SUM{(peas,slp,slt,Pst),initCropShare(peas,slp,slt,Pst)}  ;
 valcon("mingr")   = (farm_size * grass_land_share) *(-1)  ;
* AND FALLOW
 fallLimits("maxsubfall")  = farm_size * 0.5          ;
 fallLimits("maxbarefall") = farm_size * 0.3          ;
 fallLimits("maxwtld") = farm_size * 0.5          ;
*NO MIN REQUIREMENT FOR FALLOW ANYMORE
 fallLimits("mingrfall")  = farm_size     ;
 fallLimits("mincapfall") = farm_size     ;

*Making the ffn as wetland (NOT USED CURRENTLY)
*fall_cost('ffn')  = 30611  ;
 sloss_flw('ffn',slp,slt)  =  0     ;
 runoff_flw('ffn',slp,slt) =  0     ;

display farm_size, tecon, valcon, facon, fall_cost, fallLimits ;

*YIELD PARAMETERS---------------------------------------------------------------
parameters
 yc(j,y) Old experimental coefficients and P-coefficients
 yieldA(j,y) Yield function coefficients for both mitscherlich and quadratic for Ostrobothnia C2
 qyieldA N yield parameters for quadratic specification
 myieldA N yield parameters for mitscherlich specification
;
*CHECK PATH FOR YIELD FUNCTION PARAMETERS (FROM LEHTONEN 2001)
$GDXIN D:\gamsdir\yield_parameters\Aurajoki_yields_45param.gdx
$load yc qyieldA myieldA yieldA
```

```
;
display yc, qyieldA, myieldA, yieldA   ;

* PRICE DATA-----------------------------------------------------------------
parameters
*Ravinteiden arvot tammikuussa 2009 Hyötylanta_ohry slides   (Koikkailainen?)
        k_price  kalium                                   /  1.22  /
        p_price cost of mineral fertilizer euro per kg        /  1.09  /
        n_price cost of mineral fertilizer euro per kg        /  1.43  /
;
*COSTS-----------------------------------------------------------------------
*Price of land (discounted with 5\%), can be included to compare farm profits with other sources, not
affecting the optimal solution
scalar       land_p                       / 350 /
 ;
*FOR NITROGEN INITIAL VALUES (OPTIMAL)----------------------------------------
parameters
 n1(j) Profit max N dose for individual quad crops without constr
 n2(j) Profit max N dose for individual mitsch crops without constr
 ninit(j) unconstrained profit max N fertilizer dose irrespective of crop yield function
 Y1(j) Profit max yield levels for quadratic crops
 Y2(j) Profit max yield levels for mitscherlich crops
;
 n1(j) $(qyieldA(j,"a") AND qyieldA(j, "d") AND qyieldA(j, "c") NE 0)
              =  [n_price/(c_price(j)) - qyieldA(j,"d")] / [2 * qyieldA(j,"c")]   ;
 n2(j) $(myieldA(j,"m") AND myieldA(j, "k") AND myieldA(j, "b") NE 0)
              =  log[(c_price(j) * myieldA(j, "m")* myieldA(j, "k") * myieldA(j, "b"))
                     / n_price] / myieldA(j, "b")   ;
 Y1(j) =  yieldA(j, "a") + yieldA(j, "d") * n1(j)  + yieldA(j, "c") * POWER(n1(j),2) ;
* calculate all initial N levels to same paramter
              ninit(j) = n1(j) + n2(j);
*Grass, hay and silage prices are very low -> optimal N very low
* -> use more realistic one for initial values
*n1("f43") = 180 ;
*n1("f44") = 180 ;
*n1("f45") = 180 ;

VARIABLES
       Z                     Total annual producer surplus for modelling period euro
       X(j,slp,slt,Pst)        Activity level ha
       Xfa(flw,slp,slt,Pst)     Fallow (ha)
       PF(j,slp,slt,Pst)       phosphorus fertiliser applied on the field kg per ha
       PS(j,slp,slt,Pst)       phosphorus stock  mg per litre
       DRP(j,slp,slt,Pst)      Dissolved reactive phosphorus
       PP(j,slp,slt,Pst)       Particle phosphorus
       FDRP(flw,slp,slt,Pst)    Dissolved reactive phosphorus from fallow land
       FPP(flw,slp,slt,Pst)     Particle phosphorus from fallow land
       FAN(flw,slp,slt)        Nitrogen loss from fallow land
       NC(j,slp,slt)          Nitrogen load from all possible combinations of soil slope and crop
       PU(j,slp,slt,Pst)       Phosphorus removal by plants kg for plant
       CY(j,slp,slt,Pst)       Crop yield per ha kg DM in year
*       B(j,slp,slt,Pst)        Buffer
```

```
      NF(j,slp,slt)          Nitrogen fertiliser kg per ha of plant j
      wetlandEffectPred       Reduction \% of wetland calculated from its area share of drainage area
      wetlandEffectNred       Reduction \% of wetland calculated from its area share of drainage area
;
FREE VARIABLE Z ;
POSITIVE VARIABLES X, Xfa, PF, PS, DRP, PP, PU, FDRP, FPP, FAN, NF ;

* VARIABLE INITIALIZATION AND BOUNDS --------------------------------
 PF.up(j,slp,slt,Pst) = 30          ;
 PF.L(j,slp,slt,Pst)   = 0.1     ;
 PF.lo(j,slp,slt,Pst)   = 0.001     ;
*analytical optimum as the maximum level (note grass, silage and hay)
 NF.up(j,slp,slt) = ninit(j)    ;
 NF.L(j,slp,slt) = ninit(j)     ;
** 0.8
 NF.lo(j,slp,slt) = 0.001       ;
*HOW TO MODEL THE VARIANCE IN P STOCK CORRECTLY
 PS.fx(j,slp,slt,Pst) =  Pst_no(Pst)      ;
*CY.up(j,slp,slt) = 15000       ;
 X.up(j,slp,slt,Pst) = land_distr_share(slp,slt,Pst) * farm_size           ;
* farm_size * 0.9 WORKED FOR EQUALING N and P baselines before grass minimum levels were introduced
-after grass limit no feasible solutions
*X.L(j,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size  * 0.5   ;
 X.L(j,slp,slt,Pst) = N_cropdistr_hom('1-1',j,slp,slt,Pst)   ;
*X.L(j,slp,slt,Pst) = initCropShare(j,slp,slt,Pst)  *  farm_size         ;
*Xfa.L(flw,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size  * 0.5           ;
 Xfa.L(flw,slp,slt,Pst) = 0 ;
 Xfa.up(flw,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size           ;

* WETLANDS (NOT CURRENTLY USED)
 Xfa.L('ffn',slp,slt,Pst) = 0       ;
 Xfa.fx('ffn',slp,slt,Pst) = 0       ;
*Wetland reduction effect now fixed to 0
 wetlandEffectNred.fx = 1 ;
 wetlandEffectPred.fx = 1 ;

*Xfa.L("ffn",slp,slt,Pst) = 0                  ;
*Xfa.up("ffn",slp,slt,Pst) = farm_size * 0.05    ;
*------------------------------------------------------------------------

* EQUATIONS --------------------------------------------------------
EQUATIONS
        MAXAFARM              Objective function to maximise farmers profits
        tech(con)             Technological and political constraints
        land                  Amount of land (currently reduction allowed)
        landistr(slp,slt,Pst)    Keeping the soil and slope combinations fixed
        fallowConstraint(fcon)   Area contraints concerning fallow
        phobal(j,slp,slt,Pst)    Phosphorus balance
        Pupptake(j,slp,slt,Pst)  Definition of phosphorus uptake by plants
        DRPload(j,slp,slt,Pst)   Definition of dissolved phosphorus loss
        PPload(j,slp,slt,Pst)    Definition of particle phosphorus loss
        DRPfall(flw,slp,slt,Pst) Definition of dissolved phosphorus loss for fallow
```

```
        PPfall(flw,slp,slt,Pst)  Definition of particle phosphorus loss for fallow
        Nfall(flw,slp,slt)       Definition of N load for fallow(currently just a constant)
        Ncrop(j,slp,slt)         Definition of N load for crops (Simmelsgaard 1998)
        NcropREV(j,slp,slt)      Definition of N load for crops (metamodel from ICECREAM)
        loadP                    P load constraint
        loadN                    N load constraint
        cropyield(j,slp,slt,Pst) Total crop yield for year in kgs DM
        minimumgrass             Test for different formulation of grass minimum requirement constraint
        wetlandP                 Defining wetland as part of the fallow set
        wetlandN                 Defining wetland as part of the fallow set
;
*YIELD FUNCTIONS---------------------------------------------------------
 cropyield(j,slp,slt,Pst)..  CY(j,slp,slt,Pst)  =E=
 {
* quadratic base yield
          yieldA(j, "a")
* positive slope for nitrogen
          + yieldA(j, "d") * (NF(j,slp,slt) )
*negative term for nitrogen
          + yieldA(j, "c") * POWER((NF(j,slp,slt) ),2)
* mitserlich base yield
          + yieldA(j,"m")* [
                   1 - yieldA(j,"k") *
* mitserlich nitrogen curve
                   exp(-(yieldA(j,"b")*(NF(j,slp,slt))))
                  ] +
           [
* Saari 1995 p68 for P-fertiliser and manure effect   (ry/ha)
             + yc(j,"pcon")
             + yc(j,"pyc1") * (PF(j,slp,slt,Pst) )**0.5
             - yc(j,"pyc2") * PS(j,slp,slt,Pst) * (PF(j,slp,slt,Pst) )**0.5
             - yc(j,"pyc3") * (SQR[PF(j,slp,slt,Pst) ]/Pst_no(Pst) ) * yc(j,"pcoe")
             + yc(j,"pyc4") * (PF(j,slp,slt,Pst) )/Pst_no(Pst)  * yc(j,"pcoe")
*muunnos ry ha
           ] / yc(j, "sry")
 }

 ;
*RESOURCE AND TECHNOLOGY CONSTRAINTS ----------------------------
* Crop production constraints and balances
 land.. SUM{(j,slp,slt,Pst), X(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst), Xfa(flw,slp,slt,Pst)} =E= farm_size   ;
*FIX FOR PST
 tech(con) .. SUM{(j,slp,slt,Pst), tecon(j, con) * X(j,slp,slt,Pst) } =L= valcon(con)  ;
 landistr(slp,slt,Pst).. SUM{j, X(j,slp,slt,Pst)} + SUM{flw, Xfa(flw,slp,slt,Pst)}  =E=
land_distr_share(slp,slt,Pst)* farm_size  ;
 fallowConstraint(fcon)..  SUM{(flw,slp,slt,Pst), facon(flw, fcon) * Xfa(flw,slp,slt,Pst)}  =L= fallLimits(fcon)   ;

* Only phosphorus uptake in seeds is removed  (THIS IS JUST A RESULT PARAMETER SINCE THE MODEL IS
STATIC)
 Pupptake(j,slp,slt,Pst).. PU(j,slp,slt,Pst)   =E= nuval(j,"pho")/1000 * CY(j,slp,slt,Pst)  * X(j,slp,slt,Pst)
;
```

```
*LOAD FUNCTIONS  (Uusitalo and ICECREAM for P and Simmelsgaard and ICECREAM for N)
 DRPload(j,slp,slt,Pst).. DRP(j,slp,slt,Pst) =E=  runoff(j,slp,slt)* [2*(PS(j,slp,slt,Pst)  + 0.01*[PF(j,slp,slt,Pst) ]) -
1.5]  * 0.0001          ;
 PPload(j,slp,slt,Pst)..  PP(j,slp,slt,Pst)  =E=  sloss(j,slp,slt) * [250 * log(PS(j,slp,slt,Pst)  + 0.01 * PF(j,slp,slt,Pst)
)- 150] * 0.000001  ;
 DRPfall(flw,slp,slt,Pst).. FDRP(flw,slp,slt,Pst) =E= runoff_flw(flw,slp,slt)  * [2*Pst_no(Pst) - 1.5] * 0.0001
;
 PPfall(flw,slp,slt,Pst)..  FPP(flw,slp,slt,Pst)  =E= sloss_flw(flw,slp,slt)   * [250 * log[Pst_no(Pst)] - 150] *
0.000001  ;
 Ncrop(j,slp,slt)..     NC(j,slp,slt)     =E= loadCoeffN(j,slp,slt)    *  EXP[0.71 * ((NF(j,slp,slt) )  / nmax(j,slt) - 1 ) ] ;
 NcropREV(j,slp,slt)..  NC(j,slp,slt)     =E= Ncoeff_frt_b(j,slp,slt) * exp[NF(j,slp,slt) * Ncoeff_frt_c(j,slp,slt)] ;
 Nfall(flw,slp,slt)..   FAN(flw,slp,slt)  =E= loadCoeffN_flw(flw,slp,slt)          ;

*WETLAND RETENTION FUNCTIONS (CURRENTLY DISABLED)
 wetlandP.. wetlandEffectPred =E= 1 -(wtldCoefficientP * [SUM{(slp,slt,Pst), Xfa('ffn', slp,slt,Pst) } /
farm_size ])    ;
 wetlandN.. wetlandEffectNred =E= 1 -(wtldCoefficientN * [SUM{(slp,slt,Pst), Xfa('ffn', slp,slt,Pst) } /
farm_size ])    ;

* CONSTRAINT FOR ABATEMENT COST PURPOSES-----------------------------------
* summing up DRP and PP load (PP load transformed from bioavailable share to the total particle bound)
 loadP.. (SUM{(j,slp,slt,Pst), DRP(j,slp,slt,Pst)* X(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst), FDRP(flw,slp,slt,Pst)*
Xfa(flw,slp,slt,Pst)}
  + (SUM{(j,slp,slt,Pst), PP(j,slp,slt,Pst) * X(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst), FPP(flw,slp,slt,Pst) *
Xfa(flw,slp,slt,Pst)} )/bio_coef
   ) * wetlandEffectPred
   =L= baseTPload * (1 - red_l)     ;

 loadN .. [ SUM{(j,slp,slt,Pst), NC(j,slp,slt)  * X(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst), FAN(flw,slp,slt) *
Xfa(flw,slp,slt,Pst)}
        ] * retention * wetlandEffectNred
        =L= baseNload * (1 - red_l) ;

*OBJECTIVE FUNCTION----------------------------------------------------------
* MAX  CROP PROFIT - FIXED COSTS
 MAXAFARM.. Z =E= SUM{(j,slp,slt,Pst),
 [ ntlsubs(j) + capsubs(j) + lfasubs(j)   -  fix_c(j)     +
          c_price(j) *  CY(j,slp,slt,Pst)
       - [n_price* NF(j,slp,slt) + p_price * PF(j,slp,slt,Pst)]
 ]* X(j,slp,slt,Pst)
 }
* Revenue/cost from fallow 2009
 + SUM{(flw,slp,slt,Pst), Xfa(flw,slp,slt,Pst) * (capfallsubs(flw)
       + lfafallsubs(flw) - fall_cost(flw) ) }
 ;

* MODELS --------------------------------------------------------
MODEL BASELINE / MAXAFARM, cropyield
 landistr
 tech
 DRPload, PPload, DRPfall, PPfall
 NcropREV
```

```
 Nfall
/   ;

MODEL PROFMAXP / MAXAFARM, cropyield
*land
* minimumgrass
 landistr
 tech
* fallowConstraint
 DRPload, PPload, DRPfall, PPfall
 NcropREV
 Nfall
 loadP
* wetlandP
* wetlandN
/   ;

*WORKS WITH EITHER LAND DISTR OR LAND NOT WITH BOTH
MODEL PROFMAXN / MAXAFARM, cropyield
tech
*minimumgrass
*land
 landistr
* fallowConstraint
 DRPload, PPload
 DRPfall
 PPfall
 NcropREV
 Nfall
 loadN
* wetlandP
* wetlandN
/   ;

SOLVE BASELINE using NLP maximising Z ;

*Result parameters for baseline solution
 parameters cropsonDifferentslpslt(slp,slt)
          fallonDIfferentslpslt(slp,slt)
          farm_land_distr_share(slp,slt,Pst)
          total_land      to check that the land area stays constant as there is no upper bound specified for total
land
           ;
*calculate total represntative farm area from solved variable levels
 total_land = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst), Xfa.L(flw,slp,slt,Pst)} ;
* land result distribution without split to soil P classes
 cropsonDifferentslpslt(slp,slt)   = SUM{(j, Pst),X.L(j,slp,slt,Pst)}      ;
 fallonDIfferentslpslt(slp,slt)  = SUM{(flw, Pst), Xfa.L(flw,slp,slt,Pst)}  ;
 farm_land_distr_share(slp,slt,Pst) = land_distr_share(slp,slt,Pst) * farm_size  ;
 display cropsonDifferentslpslt, fallonDIfferentslpslt, farm_land_distr_share, ninit, total_land ;

* Adjusting the base load values-----------------------------------------------
```

```
 drp_load =  SUM{(j,slp,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FDRP.L(flw,slp,slt,Pst) * Xfa.L(flw,slp,slt,Pst)}  ;
 pp_load  =  SUM{(j,slp,slt,Pst), PP.L(j,slp,slt,Pst) * X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FPP.L(flw,slp,slt,Pst)  * Xfa.L(flw,slp,slt,Pst)}  ;
 basetpload = (drp_load + pp_load/bio_coef) * wetlandEffectPred.L    ;
 basenload  = SUM{(j,slp,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst), FAN.L(flw,slp,slt) *
Xfa.L(flw,slp,slt,Pst)} * wetlandEffectNred.L  ;

* PHOSPHORUS & NITROGEN ABATEMENT ITERATIONS------------------------------------
 scalar ite iteration (between 0 and 1)  starting from 1  /1/    ;

*PARAMETERS FOR LOOP RESULTS
* (storing the value for each member of count)
* N parameters are N reduction results, P for P reduction
parameters
*PROFITS & COSTS
 N_profit(count), P_profit(count) Profit parameter for NP version
 N_redcost(count), P_redcost(count) Reduction costs when NP is constrained
 Aura_N_redcost(count), Aura_P_redcost(count)  Reduction costs when NP is constrained (for watershed
comparison)
 Aura_N_Nload_het(count), Aura_P_TPload_het(count) NP load (NP constraint and heterogeneous
parameters)

*LOAD PARAMETERS
 redpro(count) Reduction percent
 VEPSvsSAMA_N, VEPSvsSAMA_TP   parameters for comparing with VEPS model results
 N_basetpload(count), P_basetpload(count) TP baseload (needs to remain constant)
 N_tpload(count), P_tpload(count)       TP load for the farm (needs to change for P constraint)
 N_bioTP(count), P_bioTP(count)         Algae available P
 N_nload(count), P_nload(count)         N load for iterations
 N_drpload(count), P_drpload(count)      DRP load for iterations
 N_ppload(count), P_ppload(count)       PP load for iterations
 N_nloadred(count), P_nloadred(count)    N reduction for iterations
 N_tploadred(count), P_tploadred(count)  TP reduction
 N_nloadred_f(count), P_tploadred_f(count) NP load reduction for feasible iterations
 watershedBaseTPload     The economic optimum base load for P
 watershedBaseNload      The economic optimum base load for N
 N_meanErosion(count), P_meanErosion(count)
 N_totalErosion(count), P_totalErosion(count)
 watershed_tpload(count) TP load scaled up for the watershed
 watershed_nload(count)  N load scaled up for the watershed
*Parameters for comparison of watersheds
 P_cropLoadDRP_Aura_het(count, j,slp,slt,Pst), N_cropLoadDRP_Aura_het(count, j,slp,slt,Pst)
 P_cropLoadPP_Aura_het(count, j,slp,slt,Pst),  N_cropLoadPP_Aura_het(count, j,slp,slt,Pst)
 P_cropLoadN_Aura_het(count, j,slp,slt,Pst),   N_cropLoadN_Aura_het(count, j,slp,slt,Pst)
 P_fallLoadDRP_Aura_het(count, flw,slp,slt,Pst), N_fallLoadDRP_Aura_het(count, flw,slp,slt,Pst)
 P_fallLoadPP_Aura_het(count, flw,slp,slt,Pst),  N_fallLoadPP_Aura_het(count, flw,slp,slt,Pst)
 P_fallLoadN_Aura_het(count, flw,slp,slt,Pst),   N_fallLoadN_Aura_het(count, flw,slp,slt,Pst)

*LAND USE
 N_totland(count), P_totland(count)  Should stay constant all the time (when NP is constrained)
 N_land(count, j), P_land(count, j)  Land allocation of crop land (when NP is constrained)
```

N_cropdistr(count,j,slp,slt,Pst), P_cropdistr(count,j,slp,slt,Pst) Distribution of crop land (NP constraint)
N_fallDistr(count,flw,slp,slt,Pst), P_fallDistr(count,flw,slp,slt,Pst) Distribution of fallow land (NP constraint)
N_cropshare(count,j), P_cropshare(count,j) Share of land for each cultivated crop (NP constraint)
N_landtotal(count), P_landtotal(count) Land allocation of crop land (when NP is constrained)
N_fallow(count, flw), P_fallow(count, flw) Fallow land of different types (when NP is constrained)
N_totfallow(count), P_totfallow(count) Total amount of fallow (when NP is constrained)
*yields
N_totalyield(count, j), P_totalyield(count, j) Total yield for each crop in kg DM (N constraint)
*wetland
N_wetlandShare(count), P_wetlandShare(count) Share of wetland of total farm area (NP)
wtlandRedN(count), wtlandRedP(count) Wetland reduction efficiency \% of Puustinen 2007 formula
wtlndRedPrN(count), wtlndRedPrP(count) Wetland reduction \% of NP

*NUTRIENT BALANCE PARAMETERS--
N_Nuse(count, j,slp,slt) N for each crop slope and soil combination
N_Puse(count, j,slp,slt,Pst) Total P for farmed crops
N_NuseFarmed(count,j,slp,slt,Pst)N fertilisation (N constraint)
N_NuseCrop(count, j) N fertilisation (N constraint)
N_PuseCrop(count, j) P fertilisation (N constraint)
N_PuseFarmed(count,j,slp,slt,Pst)P fertilisation (N constraint)
P_NuseCrop(count, j) Total N for crops (P constraint)
P_NuseFarmed(count,j,slp,slt,Pst) N fertilisation (P constraint)
P_Nuse(count, j) Total N for each farmed crops
P_Puse(count, j) Total P for each farmed crops
N_fertNsum(count) Total Fert N
N_fertPsum(count) Total Fert P
P_fertNsum(count) Total Fert N
P_fertPsum(count) Total Fert P
P_fertPmean_ha(count) Average Fert P on average
N_fertNmean_ha(count) Average Fert N on average
N_fert(j,slp,slt) N fertilisation
P_fert(j,slp,slt,Pst) P fertilisation
allfieldP(count) Sum of applied P
allfieldN(count) Sum of applied N
allfieldNperha(count) Sum of applied N per ha
seedP_in_c(count) P in seeds
fertP_in_c(count) P in fertiliser
yieldP_out_c(count) P out in yield
manureP_out_c(count) P out in manure
gateP_in_c(count) P gate in
gateP_out_c(count) P gate out
gatePbalance_c(count) P gate in - out
fieldPbalance_c(count) P field balance
P_Pstock(count,j,slp,slt,Pst) P stock variable (should remain constant in the static model)
N_Pstock(count,j,slp,slt,Pst) P stock variable (should remain constant in the static model)

*COST PARAMETERS--
fallowcost_c(count) Costs of fallow maintenance
fertcost_c(count) Costs of chemical fertilising
fieldcost_c(count) Costs of outsourced field work
landcost_c(count) Costs of land capital
all_c(count) All costs in euro per farm annually

*EFFECTS WITH SLOPE AND SOIL---
N_fallowSharebySlopeClass(count, slp), P_fallowSharebySlopeClass(count, slp) Share of fallow from total field area devided in slope classes under NP constraint
N_fallowSharebySoilType(count, slt), P_fallowSharebySoilType(count, slt) Share of fallow from total field area devided in soil classes under NP constraint
N_fallowSLP_SLT_PST(count,slp,slt,Pst), P_fallowSLP_SLT_PST(count,slp,slt,Pst) Total fallow over different slopes and soils
N_fallowShare_SLP_SLT_PST(count, slp, slt), P_fallowShare_SLP_SLT_PST(count, slp, slt) Total fallow share over different slopes and soils
N_fallow_SLP_SLT(count,slp,slt), P_fallow_SLP_SLT(count,slp,slt) Total fallow over different slopes and soils and pst
N_fallowSharebySLPandSLT(count, slp, slt), P_fallowSharebySLPandSLT(count, slp, slt) Total fallow share over different slopes and soils
N_fallowShareSLP_SLT_PST(count,slp,slt,Pst), P_fallowShareSLP_SLT_PST(count,slp,slt,Pst) Share of total fallow over different slopes and soils from total of that combination
totalFarmLandbySoil(slt) total farm land between soil types
totalFarmLandbySlope(slp) total farm land between slope classes
totalFarmLandbyPST(Pst) total farm land between soil P classes
N_landbySlopeClass(count, slp), P_landbySlopeClass(count, slp) Farm land in slope classes (should stay constant)
N_landbySoilType(count, slt), P_landbySoilType(count, slt) Farm land in soil classes (should stay constant)
N_Nfertbyslopeclass(count, slp), P_Nfertbyslopeclass(count, slp) N fertilization divided between slope classes
N_Nfertbysoiltype(count, slt), P_Nfertbysoiltype(count, slt) N fertilization divided between soil classes
N_Pfertbyslopeclass(count, slp), P_Pfertbyslopeclass(count, slp) P fertilization divided between slope classes
N_Pfertbysoiltype(count, slt), P_Pfertbysoiltype(count, slt) P fertilization divided between soil classes
N_fallowbyslopeclass(count, slp), P_fallowbyslopeclass(count, slp) Fallow area by slope classes (ha)
N_fallowbysoiltype(count, slt), P_fallowbysoiltype(count, slt) Fallow area by soil classes (ha)
N_cropbyslopetype(count, j, slp), P_cropbyslopetype(count, j, slp) Crop area by slope classes (ha)
N_cropbysoiltype(count, j, slt), P_cropbysoiltype(count, j, slt) Crop area by soil classes (ha)
N_cropbyPst(count,j,pst), P_cropbyPst(count,j,pst) Crop area by soil P classes (ha)
N_DRPbySlope(count,slp) , P_DRPbySlope(count,slp) DRP kg a-1 from each slope
N_PPbySlope(count,slp), P_PPbySlope(count,slp) PP kg a-1 from each slope
N_TPbySlope(count,slp), P_TPbySlope(count,slp) TP kg a-1 from each slope
N_NbySlope(count,slp), P_NbySlope(count,slp) N kg a-1 from each slope
N_DRPbySoil(count,slt), P_DRPbySoil(count,slt) DRP kg a-1 from each slope
N_PPbySoil(count,slt), P_PPbySoil(count,slt) PP kg a-1 from each slope
N_TPbySoil(count,slt), P_TPbySoil(count,slt) TPkg a-1 from each slope
N_NbySoil(count,slt), P_NbySoil(count,slt) N kg a-1 from each slope
N_DRPbySlopeShare(count,slp), P_DRPbySlopeShare(count,slp) Share of DRP from each slope of total DRP load
N_PPbySlopeShare(count,slp), P_PPbySlopeShare(count,slp) Share of PP from each slope of total PP load
N_TPbySlopeShare(count,slp), P_TPbySlopeShare(count,slp) Share of TP from each slope of total TP load
N_NbySlopeShare(count,slp), P_NbySlopeShare(count,slp) Share of N from each slope of total N load
N_DRPbySoilShare(count,slt), P_DRPbySoilShare(count,slt) Share DRP from each soil class of total DRP load
N_PPbySoilShare(count,slt), P_PPbySoilShare(count,slt) Share PP from each soil class of total PP load
N_TPbySoilShare(count,slt), P_TPbySoilShare(count,slt) Share TP from each soil class of total TP load
N_NbySoilShare(count,slt), P_NbySoilShare(count,slt) Share N from each soil class of total N load
N_DRPbyPst(count,Pst), P_DRPbyPst(count,Pst) DRP kg a-1 from each soil P class

```
 N_PPbyPst(count,Pst),  P_PPbyPst(count,Pst)       PP kg a-1  from each soil P class
 N_TPbyPst(count,Pst),  P_TPbyPst(count,Pst)       TP kg a-1  from each soil P class
 N_NbyPst(count,Pst),  P_NbyPst(count,Pst)        N kg a-1   from each soil P class
 N_DRPbyPstShare(count,Pst), P_DRPbyPstShare(count,Pst)      Share of each soil P class from total load
 N_PPbyPsSharet(count,Pst), P_PPbyPstShare(count,Pst)       Share of each soil P class from total load
 N_TPbyPstShare(count,Pst), P_TPbyPstShare(count,Pst)       Share of each soil P class from total load
 N_NbyPstShare(count,Pst), P_NbyPstShare(count,Pst)        Share of each soil P class from total load
;

*THE ACTUAL ABATEMENT LOOP STARTS HERE (FIRST FOR NITROGEN)
LOOP (count,

 SOLVE PROFMAXN using NLP maximising Z   ;
*BASIC PARAMETERS-------------------------------------------------------------
 N_totland(count) = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst), Xfa.L(flw,slp,slt,Pst)} ;
 N_profit(count) = Z.L                                  ;
 N_cropDistr(count,j,slp,slt,Pst) = X.L(j,slp,slt,Pst)              ;
 N_fallDistr(count,flw,slp,slt,Pst) = Xfa.L(flw,slp,slt,Pst)              ;
 N_land(count, j)   = SUM{(slp,slt,Pst), X.L(j,slp,slt,Pst)}         ;
 N_fallow(count, flw) = SUM{(slp,slt,Pst), Xfa.L(flw,slp,slt,Pst) }    ;
 N_totfallow(count)   = SUM{(flw), N_fallow(count, flw)}      ;
 N_landtotal(count)  = SUM{(j,slp,slt,Pst),X.L(j,slp,slt,Pst)} + N_totfallow(count)          ;
 N_Pstock(count,j,slp,slt,Pst) = PS.L(j,slp,slt,Pst)  ;
 N_wetlandShare(count) = SUM{(slp,slt,Pst), Xfa.L('ffn',slp,slt,Pst)} / farm_size     ;
 wtlandRedN(count) = 10.47 * SUM{(slp,slt,Pst), Xfa.L('ffn', slp,slt,Pst) } / farm_size  ;
 wtlndRedPrP(count)   =  wetlandEffectPred.L  ;
 wtlndRedPrN(count)   =  wetlandEffectNred.L  ;

*LOAD PARAMETERS-------------------------------------------------------------
 N_drpload(count) = SUM{(j,slp,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FDRP.L(flw,slp,slt,Pst) * Xfa.L(flw,slp,slt,Pst)}  ;
 N_ppload(count) = SUM{(j,slp,slt,Pst), PP.L(j,slp,slt,Pst) * X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FPP.L(flw,slp,slt,Pst)  * Xfa.L(flw,slp,slt,Pst)}  ;
 N_basetpload(count)  =  basetpload  ;
 N_tpload(count)  =  ( N_drpload(count) + N_ppload(count) / bio_coef ) * wetlandEffectPred.L   ;
 N_bioTP(count) = (N_drpload(count) + N_ppload(count))*  wetlandEffectPred.L    ;
 N_nload(count)   =  (SUM{(j,slp,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}) * wetlandEffectNred.L ;
 N_redcost(count) = N_profit("1-2") - N_profit(count)                ;
 N_nloadred(count) = N_nload("1-2")  - N_nload(count)         ;
 N_tploadred(count)= N_tpload("1-2")  - N_tpload(count)  ;
 N_cropLoadDRP_Aura_het(count, j,slp,slt,Pst) = DRP.L(j,slp,slt,Pst) ;
 N_cropLoadPP_Aura_het(count, j,slp,slt,Pst)  = PP.L(j,slp,slt,Pst)  ;
 N_cropLoadN_Aura_het(count, j,slp,slt,Pst)   = NC.L(j,slp,slt)  ;
 N_fallLoadDRP_Aura_het(count, flw,slp,slt,Pst) = FDRP.L(flw,slp,slt,Pst) ;
 N_fallLoadPP_Aura_het(count, flw,slp,slt,Pst)  = FPP.L(flw,slp,slt,Pst)  ;
 N_fallLoadN_Aura_het(count, flw,slp,slt,Pst)   = FAN.L(flw,slp,slt)  ;
 N_totalerosion(count) =   SUM{(j,slp,slt,Pst), sloss(j,slp,slt) * X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst),
sloss_flw(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)} ;
 N_meanErosion(count)  = N_totalerosion(count) / farm_size ;

*FERTILISATION-------------------------------------------------------------
```

```
 N_Nuse(count,j,slp,slt)   = NF.L(j,slp,slt)  ;
 N_NuseFarmed(count,j,slp,slt,Pst) =  NF.L(j,slp,slt) * X.L(j,slp,slt,Pst)      ;
 N_NuseCrop(count, j)  =  SUM{(slp,slt,Pst),N_NuseFarmed(count,j,slp,slt,Pst) } ;
 N_Puse(count, j,slp,slt,Pst)$(X.L(j,slp,slt,Pst) ge 0)  = PF.L(j,slp,slt,Pst) *  X.L(j,slp,slt,Pst)  ;
 N_PuseFarmed(count,j,slp,slt,Pst) = PF.L(j,slp,slt,Pst) * X.L(j,slp,slt,Pst)      ;
 N_PuseCrop(count, j) =  SUM{(slp,slt,Pst), N_PuseFarmed(count,j,slp,slt,Pst) } ;
*N_fert(j,slp,slt)$(X.L(j,slp,slt,Pst) ge 0)  = NF.L(j,slp,slt)       ;
*N_fertNsum(count) = SUM{(j,slp,slt), N_fert(j,slp,slt)}            ;
 P_fert(j,slp,slt,Pst) = 0 ;
 P_fert(j,slp,slt,Pst)$(X.L(j,slp,slt,Pst) ge 0)  = PF.L(j,slp,slt,Pst)  *  X.L(j,slp,slt,Pst)     ;
 N_fertNsum(count)  =  SUM{j, N_NuseCrop(count, j)} ;
 N_fertNmean_ha(count) = N_fertNsum(count) / farm_size ;

*REVENUE PARAMETERS-------------------------------------------------------------
 N_totalyield(count, j) = SUM{(slp,slt,Pst), CY.L(j,slp,slt,Pst)  * X.L(j,slp,slt,Pst)}             ;

*COST PARAMETERS --------------------------------------------------------------
* the land costs not here at the moment
 fallowcost_c(count) = SUM{(flw,slp,slt,Pst), Xfa.L(flw,slp,slt,Pst) *  (fall_cost(flw) + land_p)}                ;
 fertcost_c(count)  =  SUM{(j,slp,slt,Pst), n_price* NF.L(j,slp,slt)* X.L(j,slp,slt,Pst)}  + SUM{(j,slp,slt,Pst),
p_price * PF.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)};
 fieldcost_c(count) = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst) * fix_c(j)}            ;
 landcost_c(count) = land_p * N_landtotal(count)                           ;

*EFFECTS WITH SLOPE AND SOIL----------------------------------------------------
 N_Nuse(count, j,slp,slt)   = NF.L(j,slp,slt)  ;
 N_NuseFarmed(count,j,slp,slt,Pst) =  NF.L(j,slp,slt) * X.L(j,slp,slt,Pst)      ;
 N_NuseCrop(count, j)  =  SUM{(slp,slt,Pst),N_NuseFarmed(count,j,slp,slt,Pst) } ;
*N_fert(j,slp,slt)$(X.L(j,slp,slt,Pst) ge 0)  = NF.L(j,slp,slt)       ;
*N_fertNsum(count) = SUM{(j,slp,slt), N_fert(j,slp,slt)}            ;
 P_fert(j,slp,slt,Pst)$(X.L(j,slp,slt,Pst) ge 0)  = PF.L(j,slp,slt,Pst)      ;
*N_Nfertbyslopeclass(count, slp) =  SUM{(j,slt), N_fert(j,slp,slt)}   ;
*N_Nfertbysoiltype(count, slt)  =   SUM{(j,slp), N_fert(j,slp,slt)}   ;
 N_Pfertbyslopeclass(count, slp) = SUM{(j,slt,Pst), P_fert(j,slp,slt,Pst)}    ;
 N_Pfertbysoiltype(count, slt)   = SUM{(j,slp,Pst), P_fert(j,slp,slt,Pst)}    ;
 N_fallowSLP_SLT_PST(count,slp,slt,Pst) = SUM{(flw), Xfa.L(flw,slp,slt,Pst)}  ;
 N_fallow_SLP_SLT(count,slp,slt) = SUM{(flw,Pst), Xfa.L(flw,slp,slt,Pst)}  ;
 N_fallowbyslopeclass(count, slp) = SUM{(flw,slt,Pst), Xfa.L(flw,slp,slt,Pst)}   ;
 N_fallowbysoiltype(count, slt)   = SUM{(flw,slp,Pst), Xfa.L(flw,slp,slt,Pst)}   ;
 N_cropbysoiltype(count,j,slt)   = SUM{(slp,Pst), X.L(j,slp,slt,Pst)}   ;
 N_cropbyslopetype(count,j,slp)   = SUM{(slt,Pst), X.L(j,slp,slt,Pst)}   ;
 N_cropbyPst(count,j,pst)   = SUM{(slt,slp), X.L(j,slp,slt,Pst)}   ;

*THESE SHOULD NOT CHANGE---------------------------------------------------
 N_landbySlopeClass(count, slp)  =  SUM{(j,slt,Pst), X.L(j,slp,slt,Pst)}   ;
 N_landbySoilType(count, slt)    =  SUM{(j,slp,Pst), X.L(j,slp,slt,Pst)}   ;
*LOAD WITH SLOPE AND SOIL
 N_DRPbySlope(count,slp)      = SUM{(j,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,Pst),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 N_PPbySlope(count,slp)       = SUM{(j,slt,Pst), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)}  + SUM{(flw,slt,Pst),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}    ;
```

```
 N_TPbySlope(count,slp)      = N_DRPbySlope(count,slp) +  N_PPbySlope(count,slp)/bio_coef
;
 N_NbySlope(count,slp)       = SUM{(j,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slt,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}          ;
 N_DRPbySoil(count,slt)      = SUM{(j,slp,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,Pst),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 N_PPbySoil(count,slt)       = SUM{(j,slp,Pst), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)}  + SUM{(flw,slp,Pst),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}    ;
 N_TPbySoil(count,slt)       = N_DRPbySoil(count,slt) +  N_PPbySoil(count,slt)/bio_coef            ;
 N_NbySoil(count,slt)        = SUM{(j,slp,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}          ;
 N_DRPbySlopeShare(count,slp)  = N_DRPbySlope(count,slp) /  N_drpload(count) ;
 N_PPbySlopeShare(count,slp)   = N_PPbySlope(count,slp)  /  N_ppload(count)  ;
 N_TPbySlopeShare(count,slp)   = N_TPbySlope(count,slp)  /  N_tpload(count)  ;
 N_NbySlopeShare(count,slp)    = N_NbySlope(count,slp)   /  N_nload(count)   ;
 N_DRPbySoilShare(count,slt)   = N_DRPbySoil(count,slt) /  N_drpload(count) ;
 N_PPbySoilShare(count,slt)    = N_PPbySoil(count,slt)  /  N_ppload(count)  ;
 N_TPbySoilShare(count,slt)    = N_TPbySoil(count,slt)  /  N_tpload(count)  ;
 N_NbySoilShare(count,slt)     = N_NbySoil(count,slt)   /  N_nload(count)   ;
 N_DRPbyPst(count,Pst)       = SUM{(j,slt,slp), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,slp),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 N_PPbyPst(count,Pst)        = SUM{(j,slt,slp), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,slp),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}     ;
 N_TPbyPst(count,Pst)        = N_DRPbyPst(count,Pst) +  N_PPbyPst(count,Pst)/bio_coef                  ;
 N_NbyPst(count,Pst)         = SUM{(j,slp,slt), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}          ;
 N_DRPbyPstShare(count,Pst)  = N_DRPbyPst(count,Pst) /  N_drpload(count) ;
 N_PPbyPsSharet(count,Pst)   = N_PPbyPst(count,Pst)  /  N_ppload(count)  ;
 N_TPbyPstShare(count,Pst)   = N_TPbyPst(count,Pst)  /  N_tpload(count)  ;
 N_NbyPstShare(count,Pst)    = N_NbyPst(count,Pst)   /  N_nload(count)   ;

*Storing the reduction \% of previous solve
redpro(count) = reduction  * 100      ;
*calculate reduction \% for next iteration (in the loop)
reduction = 1 - ite            ;
red_l = reduction              ;
*Increase the reduction by 2\%  ( number of iterations given by the set "count" )
       ite = ite - .02        ;
) ;

*USE THESE PARAMETERS IF LOOP SOLUTIONS ARE FEASIBLE
 Aura_N_redcost(count) =  N_redcost(count)   ;
 Aura_N_Nload_het(count)  = N_nload(count)   ;
 N_nloadred_f(count) =  N_nloadred(count)   ;

*LOOP FOR P------------------------------------------------------------------

* VARIABLE INITIALIZATION AND BOUNDS --------------------------------
 PF.up(j,slp,slt,Pst) = 30          ;
 PF.L(j,slp,slt,Pst)   = 0.1     ;
 PF.lo(j,slp,slt,Pst)   = 0.001     ;
 NF.up(j,slp,slt) = ninit(j)    ;
```

```
 NF.L(j,slp,slt) = ninit(j) * 0.8   ;
 NF.lo(j,slp,slt) = 0.001       ;
 PS.fx(j,slp,slt,Pst) =  Pst_no(Pst)      ;
 X.up(j,slp,slt,Pst) = land_distr_share(slp,slt,Pst) * farm_size            ;
 X.L(j,slp,slt,Pst) = N_cropdistr_hom('1-1',j,slp,slt,Pst)   ;

*X.L(j,slp,slt,Pst) =  best_P_profit_land(j,slp,slt,Pst)  ;
*X.L(j,slp,slt,Pst) = best10_P_profit_land(j,slp,slt,Pst) ;

 Xfa.L(flw,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size  * 0.5          ;
 Xfa.up(flw,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size             ;

*NO WETLAND
*Xfa.L("ffn",slp,slt,Pst) = 0                   ;
 Xfa.fx("ffn",slp,slt,Pst) = 0                  ;
 wetlandEffectNred.fx = 1 ;
 wetlandEffectPred.fx = 1 ;

*------------------------------------------------------------------------

*Re-init also the base scalars
 reduction     = -10  ;
 red_l        = 0    ;
 ite          = 1    ;

*Readjust so that load constraints are again not binding
 baseTPload = modTP/model_ha*farm_size*baseTPmulti ;
*basetpload = 40 ;
 baseNload = baseN_load ;
 baseNload = baseNload/model_ha*farm_size*baseNmulti  ;
 tpinit = baseTPload/model_ha*farm_size  ;

BASELINE.OPTFILE = 0;
* Profit maximistion without load constraints
SOLVE BASELINE using NLP maximising Z ;

* Adjusting the base load values back to profit maximisng level
 drp_load =  SUM{(j,slp,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FDRP.L(flw,slp,slt,Pst) * Xfa.L(flw,slp,slt,Pst)}  ;
 pp_load  =  SUM{(j,slp,slt,Pst), PP.L(j,slp,slt,Pst) * X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FPP.L(flw,slp,slt,Pst)  * Xfa.L(flw,slp,slt,Pst)}  ;
 basetpload = (drp_load + pp_load/bio_coef ) * wetlandEffectPred.L   ;
 basenload  = (SUM{(j,slp,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst), FAN.L(flw,slp,slt) *
Xfa.L(flw,slp,slt,Pst)}) * wetlandEffectNred.L  ;

*New loop, this time for P reductions
LOOP (count,

 SOLVE PROFMAXP using NLP maximising Z   ;
*BASIC PARAMETERS----------------------------------------------------------------
 P_totland(count) = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst), Xfa.L(flw,slp,slt,Pst)} ;
 P_profit(count) = Z.L                  ;
```

```
 P_cropdistr(count,j,slp,slt,Pst) = X.L(j,slp,slt,Pst)           ;
 P_falldistr(count,flw,slp,slt,Pst) = Xfa.L(flw,slp,slt,Pst)            ;
 P_land(count, j)   = SUM{(slp,slt,Pst), X.L(j,slp,slt,Pst)}       ;
 P_fallow(count, flw) = SUM{(slp,slt,Pst),Xfa.L(flw,slp,slt,Pst) }    ;
 P_totfallow(count)  =  SUM{flw, P_fallow(count, flw)}        ;
 P_landtotal(count)  = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst)} + P_totfallow(count)          ;
 P_Pstock(count,j,slp,slt,Pst) = PS.L(j,slp,slt,Pst)  ;
 P_wetlandShare(count) = SUM{(slp,slt,Pst), Xfa.L('ffn',slp,slt,Pst)} / farm_size     ;
 wtlandRedP(count) = POWER[3.2, ROUND(SUM{(slp,slt,Pst), Xfa.L('ffn', slp,slt,Pst) } / farm_size * 0.57) ]  ;
 wtlndRedPrP(count)   =  wetlandEffectPred.L  ;

*LOAD PARAMETERS
 P_drpload(count) =  SUM{(j,slp,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FDRP.L(flw,slp,slt,Pst) * Xfa.L(flw,slp,slt,Pst)}  ;
 P_ppload(count) =  SUM{(j,slp,slt,Pst), PP.L(j,slp,slt,Pst) * X.L(j,slp,slt,Pst)} + SUM{(flw,slp,slt,Pst),
FPP.L(flw,slp,slt,Pst)  * Xfa.L(flw,slp,slt,Pst)}  ;
 P_basetpload(count)  =  basetpload  ;
 P_tpload(count)  =  (P_drpload(count) + P_ppload(count) / bio_coef) * wetlandEffectPred.L    ;
 P_bioTP(count) = (P_drpload(count) + P_ppload(count))*  wetlandEffectPred.L    ;
 P_nload(count)  = SUM{(j,slp,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)} * wetlandEffectNred.L ;
 P_redcost(count) = P_profit("1-2") - P_profit(count)               ;
 P_nloadred(count) = P_nload("1-2")  - P_nload(count)         ;
 P_tploadred(count)= P_tpload("1-2")  - P_tpload(count)  ;
 P_cropLoadDRP_Aura_het(count, j,slp,slt,Pst) = DRP.L(j,slp,slt,Pst) ;
 P_cropLoadPP_Aura_het(count, j,slp,slt,Pst)  = PP.L(j,slp,slt,Pst)  ;
 P_cropLoadN_Aura_het(count, j,slp,slt,Pst)   = NC.L(j,slp,slt)  ;
 P_fallLoadDRP_Aura_het(count, flw,slp,slt,Pst) = FDRP.L(flw,slp,slt,Pst) ;
 P_fallLoadPP_Aura_het(count, flw,slp,slt,Pst)  = FPP.L(flw,slp,slt,Pst)  ;
 P_fallLoadN_Aura_het(count, flw,slp,slt,Pst)   = FAN.L(flw,slp,slt)  ;
 P_totalerosion(count) =   SUM{(j,slp,slt,Pst), sloss(j,slp,slt) * X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt,Pst),
sloss_flw(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)} ;
 P_meanErosion(count)  = P_totalerosion(count) / farm_size ;

*FERTILISATION------------------------------------------------------------------
 P_fert(j,slp,slt,Pst) = 0 ;
 P_fert(j,slp,slt,Pst)$(X.L(j,slp,slt,Pst) ge 0)  = PF.L(j,slp,slt,Pst) *  X.L(j,slp,slt,Pst)       ;
 P_Puse(count,j)  = SUM{(slp,slt,Pst), P_fert(j,slp,slt,Pst)}                ;
 P_fertPsum(count)= SUM{j, P_Puse(count,j)}                ;
* P_Nuse(count,j,slp,slt)   = NF.L(j,slp,slt)               ;
 P_NuseFarmed(count,j,slp,slt,Pst) =  NF.L(j,slp,slt) * X.L(j,slp,slt,Pst)      ;
 P_NuseCrop(count, j)  =  SUM{(slp,slt,Pst),P_NuseFarmed(count,j,slp,slt,Pst) } ;
 P_fertNsum(count)= SUM{j, P_NuseCrop(count, j) }                         ;
 P_fertPmean_ha(count) = P_fertPsum(count) / farm_size                       ;

*REVENUE PARAMETERS------------------------------------------------------------
 P_totalyield(count, j) = SUM{(slp,slt,Pst), CY.L(j,slp,slt,Pst)  * X.L(j,slp,slt,Pst)}             ;

*COST PARAMETERS --------------------------------------------------------------
 fallowcost_c(count) = SUM{(flw,slp,slt,Pst), Xfa.L(flw,slp,slt,Pst) *  (fall_cost(flw) + land_p)}                ;
 fertcost_c(count)  =  SUM{(j,slp,slt,Pst), n_price* NF.L(j,slp,slt)* X.L(j,slp,slt,Pst)}  + SUM{(j,slp,slt,Pst),
p_price * PF.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)};
```

```
 fieldcost_c(count) = SUM{(j,slp,slt,Pst), X.L(j,slp,slt,Pst) * fix_c(j)}              ;
 landcost_c(count) = land_p * P_landtotal(count)                           ;

*EFFECTS WITH SLOPE AND SOIL----------------------------------------------------
*P_Nfertbyslopeclass(count,slp) =  SUM{(j,slt,Pst), N_fert(j,slp,slt)}   ;
*P_Nfertbysoiltype(count,slt)  =   SUM{(j,slp,Pst), N_fert(j,slp,slt)}   ;
 P_Pfertbyslopeclass(count,slp) = SUM{(j,slt,Pst), P_fert(j,slp,slt,Pst)}    ;
 P_Pfertbysoiltype(count,slt)   = SUM{(j,slp,Pst), P_fert(j,slp,slt,Pst)}    ;
 P_fallowSLP_SLT_PST(count,slp,slt,Pst) = SUM{(flw), Xfa.L(flw,slp,slt,Pst)}  ;
 P_fallow_SLP_SLT(count,slp,slt) = SUM{(flw,Pst), Xfa.L(flw,slp,slt,Pst)}  ;
 P_fallowbyslopeclass(count,slp) = SUM{(flw,slt,Pst), Xfa.L(flw,slp,slt,Pst)} ;
 P_fallowbysoiltype(count,slt)   = SUM{(flw,slp,Pst), Xfa.L(flw,slp,slt,Pst)} ;
 P_cropbysoiltype(count,j,slt)   = SUM{(slp,Pst), X.L(j,slp,slt,Pst)}   ;
 P_cropbyslopetype(count,j,slp)   = SUM{(slt,Pst), X.L(j,slp,slt,Pst)}   ;
 P_cropbyPst(count,j,pst)   = SUM{(slt,slp), X.L(j,slp,slt,Pst)}   ;
 P_landbySlopeClass(count, slp)  =  SUM{(j,slt,Pst), X.L(j,slp,slt,Pst)}   ;
 P_landbySoilType(count, slt)    =  SUM{(j,slp,Pst), X.L(j,slp,slt,Pst)}   ;

*LOAD WITH SLOPE AND SOIL-------------------------------------------------------
 P_DRPbySlope(count,slp)      = SUM{(j,slt,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,Pst),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}  ;
 P_PPbySlope(count,slp)       = SUM{(j,slt,Pst), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)}  + SUM{(flw,slt,Pst),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 P_TPbySlope(count,slp)       = N_DRPbySlope(count,slp) +  N_PPbySlope(count,slp)/bio_coef              ;
 P_NbySlope(count,slp)        = SUM{(j,slt,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slt,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}         ;
 P_DRPbySoil(count,slt)       = SUM{(j,slp,Pst), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slp,Pst),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}  ;
 P_PPbySoil(count,slt)        = SUM{(j,slp,Pst), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)}  + SUM{(flw,slp,Pst),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 P_TPbySoil(count,slt)        = N_DRPbySoil(count,slt) +  N_PPbySoil(count,slt)/bio_coef                        ;
 P_NbySoil(count,slt)         = SUM{(j,slp,Pst), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,Pst),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}         ;
 P_DRPbySlopeShare(count,slp)  = P_DRPbySlope(count,slp) /  P_drpload(count) ;
 P_PPbySlopeShare(count,slp)   = P_PPbySlope(count,slp)  /  P_ppload(count)  ;
 P_TPbySlopeShare(count,slp)   = P_TPbySlope(count,slp)  /  P_tpload(count)  ;
 P_NbySlopeShare(count,slp)    = P_NbySlope(count,slp)   /  P_nload(count)   ;
 P_DRPbySoilShare(count,slt)   = P_DRPbySoil(count,slt)  /  P_drpload(count) ;
 P_PPbySoilShare(count,slt)    = P_PPbySoil(count,slt)   /  P_ppload(count)  ;
 P_TPbySoilShare(count,slt)    = P_TPbySoil(count,slt)   /  P_tpload(count)  ;
 P_NbySoilShare(count,slt)     = P_NbySoil(count,slt)    /  P_nload(count)   ;
 P_DRPbyPst(count,Pst)       = SUM{(j,slt,slp), DRP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,slp),
FDRP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}   ;
 P_PPbyPst(count,Pst)        = SUM{(j,slt,slp), PP.L(j,slp,slt,Pst)* X.L(j,slp,slt,Pst)} + SUM{(flw,slt,slp),
FPP.L(flw,slp,slt,Pst)* Xfa.L(flw,slp,slt,Pst)}     ;
 P_TPbyPst(count,Pst)        = P_DRPbyPst(count,Pst) +  P_PPbyPst(count,Pst)/bio_coef                        ;
 P_NbyPst(count,Pst)         = SUM{(j,slp,slt), NC.L(j,slp,slt)* X.L(j,slp,slt,Pst)} +  SUM{(flw,slp,slt),
FAN.L(flw,slp,slt) * Xfa.L(flw,slp,slt,Pst)}          ;
 P_DRPbyPstShare(count,Pst)  = P_DRPbyPst(count,Pst) /  P_drpload(count) ;
 P_PPbyPstShare(count,Pst)   = P_PPbyPst(count,Pst)  /  P_ppload(count)  ;
 P_TPbyPstShare(count,Pst)   = P_TPbyPst(count,Pst)  /  P_tpload(count)  ;
 P_NbyPstShare(count,Pst)    = P_NbyPst(count,Pst)   /  P_nload(count)   ;
```

```
*Parameter for storing the reduction percent
 redpro(count) = reduction * 100 ;
*Setting reduction percent for next solution in the loop
 reduction = 1 - ite           ;
 red_l = reduction             ;
*Reducing the allowed load by two percent from the previous iteration
       ite = ite - .02        ;
*Closing the loop
) ;

*USE THESE PARAMETERS IF SOLUTIONS ARE FEASIBLE
 Aura_P_redcost(count) =  P_redcost(count)   ;
 P_tploadred_f(count) = P_tploadred(count)  ;
 best_P_profit_land(j,slp,slt,Pst)   =  P_cropdistr('1-5',j,slp,slt,Pst)     ;
 best10_P_profit_land(j,slp,slt,Pst) =  P_cropdistr('1-10',j,slp,slt,Pst)    ;

*EXPORT THESE SHARES TO MDB FOR MAP REPRESENTATION OF RESULTS
 totalFarmLandbySoil(slt) = SUM{(slp,Pst), land_distr_farm(slp,slt,Pst)}  ;
 totalFarmLandbySlope(slp) = SUM{(slt,Pst), land_distr_farm(slp,slt,Pst)} ;
 totalFarmLandbyPST(Pst) = SUM{(slp,slt), land_distr_farm(slp,slt,Pst)} ;
 N_fallowShareSLP_SLT_PST(count,slp,slt,Pst)$(land_distr_farm(slp,slt,Pst) > 0) =
N_fallowSLP_SLT_PST(count,slp,slt,Pst) /  land_distr_farm(slp,slt,Pst)   ;
 P_fallowShareSLP_SLT_PST(count,slp,slt,Pst)$(land_distr_farm(slp,slt,Pst) > 0) =
P_fallowSLP_SLT_PST(count,slp,slt,Pst) /  land_distr_farm(slp,slt,Pst)   ;
 N_fallowSharebySlopeClass(count, slp)$(totalFarmLandbySlope(slp) > 0) =  N_fallowbyslopeclass(count,
slp) /  totalFarmLandbySlope(slp) ;
 N_fallowSharebySoilType(count, slt)$(totalFarmLandbySoil(slt) > 0)   =  N_fallowbysoiltype(count, slt)   /
totalFarmLandbySoil(slt)  ;
 P_fallowSharebySlopeClass(count, slp)$(totalFarmLandbySlope(slp) > 0) =  P_fallowbySlopeClass(count, slp)
/  totalFarmLandbySlope(slp) ;
 P_fallowSharebySoilType(count, slt)$(totalFarmLandbySoil(slt) > 0)   =  P_fallowbySoilType(count, slt) /
totalFarmLandbySoil(slt)  ;

*UP-SCALING THE REPRESENTATIVE FARM RESULTS TO WHOLE WATERSHED
 watershed_tpload(count) = P_tpload(count)/farm_size  * model_ha             ;
 watershed_nload(count)  = N_nload(count)/farm_size  * model_ha              ;
 watershedbaseTPload  = watershed_tpload('1-2')                            ;
 watershedbaseNload   = watershed_nload('1-2')                             ;
 VEPSvsSAMA_TP   =   (P_tpload("1-2")/farm_size * model_ha )/ veps_TP_2002     ;
 VEPSvsSAMA_N    =   (N_nload("1-2")/farm_size * model_ha )/ veps_N_2002       ;
*CROP SHARE UNDER DIFFERENT CONSTRAINTS
 N_cropshare(count,j) = N_land(count, j) / N_landtotal(count)   ;
 P_cropshare(count,j) = P_land(count, j) / P_landtotal(count)   ;

* FOR MAPS AND OLS  (watershedResultsComparison.gms)---------------------------
 parameters      P_cropDistr_Aura_het(count,j,slp,slt,Pst), N_cropDistr_Aura_het(count,j,slp,slt,Pst) crop
distributions with "Aura" to separate from other watersheds
              P_fallDistr_Aura_het(count,flw,slp,slt,Pst), N_fallDistr_Aura_het(count,flw,slp,slt,Pst) fallow
distributions with "Aura" to separate from other watersheds
              Aura_land_distr(slp,slt,Pst) farm land distribution
              Aura_farm_size            Representative farm size (ha)
```

```
 ;
 Aura_farm_size =  farm_size ;
 Aura_land_distr(slp,slt,Pst)$(land_distr_share(slp,slt,Pst) > 0 ) = land_distr_share(slp,slt,Pst)* farm_size ;
 P_cropDistr_Aura_het(count,j,slp,slt,Pst) = P_cropdistr(count,j,slp,slt,Pst)   ;
 N_cropDistr_Aura_het(count,j,slp,slt,Pst) = N_cropdistr(count,j,slp,slt,Pst)   ;
 P_fallDistr_Aura_het(count,flw,slp,slt,Pst) = P_fallDistr(count,flw,slp,slt,Pst) ;
 N_fallDistr_Aura_het(count,flw,slp,slt,Pst) = N_fallDistr(count,flw,slp,slt,Pst) ;

*DISPLAY PARAMETER VALUES
 display farm_size, P_totland, N_totland , Aura_cropfarm_areas, Aura_farm_no ;
 display land_distr_share, land_distr,  model_ha, yieldA, ninit       ;
 display P_cropshare, N_cropshare                          ;
 display N_NuseFarmed, N_NuseCrop  ;
 display ntlsubs, capsubs, lfasubs, capfallsubs, lfafallsubs, c_price ;
 display tecon, valcon ;
 display Aura_land_distr, P_tpload, P_nload, N_tpload, N_nload ;
 display P_bioTP, N_bioTP     ;
 display P_fallowbyslopeclass, P_fallowbysoiltype, P_cropbysoiltype, P_cropbyslopetype,
P_fallowSharebySoilType ;
 display P_profit, N_profit ;
 display P_fallow_SLP_SLT, P_totfallow, P_fallow, P_land, N_fallow_SLP_SLT,  N_land,  N_totfallow,
N_fallow  ;
 display PS.L, sloss_flw, total_land                          ;
 display P_totalErosion, N_totalErosion, P_meanErosion, N_meanErosion                         ;
 display N_land, N_totfallow, N_fallowSLP_SLT_PST, N_fallowbyslopeclass, N_fallowbysoiltype,
N_cropbysoiltype, N_cropbyslopetype ;
 display
P_DRPbySlope,P_PPbySlope,P_TPbySlope,P_NbySlope,P_DRPbySoil,P_PPbySoil,P_TPbySoil,P_NbySoil      ;
 display
P_DRPbySlopeShare,P_PPbySlopeShare,P_TPbySlopeShare,P_NbySlopeShare,P_DRPbySoilShare,P_PPbySoil
Share,P_TPbySoilShare,P_NbySoilShare ;
 display
N_DRPbySlope,N_PPbySlope,N_TPbySlope,N_NbySlope,N_DRPbySoil,N_PPbySoil,N_TPbySoil,N_NbySoil      ;
 display
N_DRPbySlopeShare,N_PPbySlopeShare,N_TPbySlopeShare,N_NbySlopeShare,N_DRPbySoilShare,N_PPbyS
oilShare,N_TPbySoilShare,N_NbySoilShare ;
 display N_DRPbyPst,N_PPbyPst,N_TPbyPst,N_NbyPst                   ;
 display P_DRPbyPst,P_PPbyPst,P_TPbyPst,P_NbyPst                   ;
 display N_DRPbyPstShare, N_PPbyPsSharet,N_TPbyPstShare, N_NbyPstShare ;
 display P_DRPbyPstShare,P_PPbyPstShare,P_TPbyPstShare, P_NbyPstShare  ;
 display N_cropbyPst, P_cropbyPst    ;
 display Ncoeff_frt_b, Ncoeff_frt_c ;
 display N_NuseFarmed, N_NuseCrop, P_fertPsum, N_fertNsum, N_Puse, P_Puse, N_PuseFarmed,
P_fertPmean_ha, N_fertNmean_ha     ;
* These comparison between the models runs should be updated (if used)
 display best_P_profit_land, best10_P_profit_land ;
*Wetland should be disabled
 display wtlandRedP, wtlndRedPrP, N_wetlandShare, P_wetlandShare  ;
```

OLS estimation of load function from ICECREAM parameters of Aurajoki

```
$Title OLS estimation of load function from ICECREAM parameters of Aurajoki

$ontext
ESTIMATING THE METAMODEL FOR N AND P LOAD
Since ICECREAM results have not been calculated for all land use data types, some data classes in economic
abatement model would not have load parameters.
This file is the first step in creating load functions that cover all data types specified in the economic model.
As result all load results are based on the functions estimated here instead of the original ICECREAM
results.
This model contains the code for importing ICECREAM results from a mdb database to GAMS, specifying
erosion, runoff and nitrogen as functions of field slope & some other factors, minimizing the error terms
and drawing result graphs.
The main results from this file are the coefficients for calculating erosion and runoff as functions of slope
(best fits were for exponential and linear, respectively) and the coefficient as nitrogen load as function os
slope (linear).  In later stages erosion is used to calculate particulate phosphorus and runoff the dissolved
reactive phosphorus, which sum to total phosphorus.
The operation is split between the 2 files because, the slope set between the ICECREAM results and the
model slope categories is different. In this file the slope classes are defined by the ICECREAM results and in
the extrapolate file by the GIS data and the economic abatement problem. Similarly, the soil class are
defined separately.
This load can be compared with directly obtained ICECREAM total P results. These are also used to estimate
direct load functions as a function of slope and soil P.The soil P function could be compared with the
function that the model uses (byUusitalo)
For comparison, the file also includes the other functional forms for erosion.
$offtext

*IMPORT FROM MDB database----------------------------------------------------
* Requires mdb2gms  (http://www.gams.com/dd/docs/tools/mdb2gms.pdf)
* This is multiquery batch version
$onecho > cmd.txt
I="D:\GIS\SAMA_database2003_INPUT.mdb"
X=Aurajoki_load_param.gdx
Q1=select j from setj order by id
s1=j
Q2=select flw from setflw
s2=flw
Q3=select slp from ICECREAM_slope_set
s3=slp
Q4=select slt from soil_set
s4=soil
Q5=select Nfert from SetNfert order by Nf
s5=nfert
Q6=select Pfert from setPfert
s6=pfert
Q7=select PAC from setPAC
s7=pac
Q14=select Pst from setPstatus
s14=Pst
Q8=select slp, slopepoint_REV from ICECREAM_slope_set
```

```
p8=slopepoint
Q18=select slp, slopepoint_REV from ICECREAM_slope_set
p18=slopedata
Q15=select Pst, Pst_no from setPstatus
p15=Pst_no
Q9=select j, slp, soil, PAC, Pfert, RunoffP from ICECREAM_Aura_P_crop_param
p9=runoff_P
Q10=select j, slp, soil, PAC, Pfert, PercP from ICECREAM_Aura_P_crop_param
p10=perc_P
Q11=select j, slp, soil, PAC, Pfert, soilP from ICECREAM_Aura_P_crop_param
p11=soilP
Q12=select j, slp, soil, PAC, Pfert, Pf from ICECREAM_Aura_P_crop_param
p12=fertP
Q13=select j, slp, soil, PAC, Pfert, Runoff from ICECREAM_Aura_P_crop_param
p13=runoff
Q16=select j, slp, soil, PAC, Pfert, Erosion from ICECREAM_Aura_P_crop_param
p16=sloss
Q17=select flw, slp, soil, PAC, Pfert, SumP from ICECREAM_Aura_P_fall_param
p17=TP_load_flw
Q19=select flw, slp, soil, PAC, Pfert, RunoffP from ICECREAM_Aura_P_fall_param
p19=runoff_P_flw
Q20=select flw, slp, soil, PAC, Pfert, PercP from ICECREAM_Aura_P_fall_param
p20=perc_P_flw
Q21=select flw, slp, soil, PAC, Pfert, soilP from ICECREAM_Aura_P_fall_param
p21=soilP_flw
Q22=select flw, slp, soil, PAC, Pfert, Pf from ICECREAM_Aura_P_fall_param
p22=fertP_flw
Q23=select flw, slp, soil, PAC, Pfert, Runoff from ICECREAM_Aura_P_fall_param
p23=runoff_flw
Q24=select flw, slp, soil, PAC, Pfert, Erosion from ICECREAM_Aura_P_fall_param
p24=sloss_flw
Q25=select flw, slp, soil,  Soil_loss from bareVSgreen
p25=OLD_sloss_flw
Q29=select flw, slp, soil,  Runoff from bareVSgreen
p29=OLD_runoff_flw
Q26=select j, slp, soil, nfert, TotN from ICECREAM_Aura_N_crop_param
p26=N_load
Q27=select flw, slp, soil, nfert, TotN from ICECREAM_Aura_N_fall_param
p27=N_load_flw
*Q28=select j, slp, soil, nfert, Runoffmm from ICECREAM_Aura_N_crop_param
*p28=runoffN
Q30=select j, slp, soil, PAC, Pfert, SumP from ICECREAM_Aura_P_crop_param
p30=TP_load
Q31=select j, slp, soil, nfert, Nf from ICECREAM_Aura_N_crop_param order by Nf
p31=N_fert
$offecho

$call =mdb2gms \atcmd.txt

*The members of these set are defined in the dabase
sets    j      farmed crops
        soil   soil types
```

```
      flw    fallow types
      slp    slope classes
      pac    different P content in soil
      pfert  annual P application classes (kg per ha)
      nfert  annual N application classes (kg per ha)
      Pst    P soil status classes (mg in l of soil)
 ;

* parameters for imported data and intermediate results
PARAMETERS
 Pst_no(Pst)                      P-status (P in mg l-1 in soil)
 slopepoint(slp)                  Field slopes in ICECREAM results (point estimates)
 slopedata(slp)                   Slopes of which point estimates exist
 TP_load(j,slp,soil,pac,pfert)        P load from a crop parcel (kg per ha) (sum of runoff and percolating P)
 runoff_P(j,slp,soil,pac,pfert)       Runoff P from a crop parcel
 perc_P(j,slp,soil,pac,pfert)         Percolating P from a crop parcel
 fertP(j,slp,soil,pac,pfert)          Crop P fert quantity (in ICECREAM load estimate data)
 soilP(j,slp,soil,pac,pfert)          Crop P soil status (point estimates)
 runoff(j,slp,soil,PAC,pfert)         Crop Runoff  (mm per ha)
 runoffdata(j,slp,soil)               Crop Runoff  (mm per ha) in right dimensions
 sloss(j,slp,soil,PAC,pfert)          Crop Soil loss due erosion (kg per ha)
 erosiondata(j,slp,soil)              Erosion in kg per ha  (right dimensions)
 erosiondata_original(j,slp,soil)     Erosion in kg per ha in right dimensions (original icecream results
without recalibration)
 erosionHtSslope1(j)                  Crop erosion for only Hts soil class and slope class 1
*Load parameters for fallow
 TP_load_flw(flw,slp,soil,pac,pfert)    Fallow P load from a parcel
 runoff_P_flw(flw,slp,soil,pac,pfert)   Fallow runoff
 perc_P_flw(flw,slp,soil,pac,pfert)     Fallow
 fertP_flw(flw,slp,soil,pac,pfert)      P fertilisation amounts with load estimate data
 soilP_flw(flw,slp,soil,pac,pfert)      P soil status of which point estimates exist
 runoff_flw(flw,slp,soil,PAC,pfert)     Fallow Runoff  (mm per ha)
 sloss_flw(flw,slp,soil,PAC,pfert)      Fallow soil loss due erosion (kg per ha)
* Load parameters reduced to right dimensions (no effect from pfert or soil P)
 runoff__flw(flw,slp,soil)            Fallow Runoff in (mm per ha)
 sloss__flw(flw,slp,soil)             Fallow soil loss due erosion (kg per ha)
 Aura_flw_runoff_coeff(flw,slp,soil)
 Aura_flw_sloss_coeff(flw,slp,soil)
 Aura_flw_TP_coeff(flw,slp,soil,pac,pfert)
*Nitrogen
 runoffN(j,slp,soil,nfert)            Crop Runoff for N
 N_load(j,slp,soil,nfert)             N load per crop ha of land annually
 Nloaddata(j,slp,soil,nfert)            ICECREAM N load data
 N_fert(j,slp,soil,nfert)             N fertilisation per crop ha of land
 N_load_flw(flw,slp,soil,nfert)         N load per fallow ha of land (kg per ha per a)
 NloadData_flw(flw,slp,soil)            N load per fallow ha of land (reduced dimensions)
 slopepointN(j,slp,soil)              Slope values for N parameters from ICECREAM
* Some parameters to check if there is missing data (NODATA parameter = 1)
 slopeNODATA(j,slp,soil,pac,pfert)      point estimates DO NOT exist
 soilNODATA(j,slp,soil,pac,pfert)       point estimates NO NOT exist
 soilNODATA_flw(flw,slp,soil,pac,pfert)  point estimates NO NOT exist
*Mean parameter values  (average of mean values of slope classes)
```

```
 meanErosion                    Crop mean erosion
 meanErosion_flw                   Fallow mean erosion
 meanRunoff                     Crop mean runoff
 meanRunoff_flw                    Fallow mean runoff
;
*Loading the above declared sets and parameters from the gdx file created by the database import
$gdxin 'Aurajoki_load_param.gdx'
$load j slp flw soil pac Pst pfert nfert TP_load  runoff_P perc_P fertP runoff sloss soilP Pst_no slopedata
slopepoint TP_load_flw runoff_P_flw perc_P_flw fertP_flw runoff_flw soilP_flw sloss_flw N_load
N_load_flw N_fert  OLD_sloss_flw OLD_runoff_flw

*for checking that no data is missing
display slp, soil, pac, slopedata, soilp, soilP_flw, nfert  ;
soilNODATA(j,slp,soil,pac,pfert)$(soilP(j,slp,soil,pac,pfert) EQ 0) = 1 ;
slopeNODATA(j,slp,soil,pac,pfert)$(soilP(j,slp,soil,pac,pfert) EQ 0) = 1 ;
soilNODATA_flw(flw,slp,soil,pac,pfert)$(soilP_flw(flw,slp,soil,pac,pfert) EQ 0) = 1 ;
display soilNODATA, slopeNODATA, soilNODATA_flw, N_load_flw    ;

*FIXING SOME MISTAKES IN THE ICECREAM RESULTS----------------------------------
*CORRECTION FOR GRASS LOAD
* Exceptionally high N load for grass/silage resulted (probably) from double the fertilisation compared to
given values
* Since the relationship is not linear, the parameter calculation is only an estimate for those classes were
no
* exact match can be found from ICECREAM
N_load('f43',slp,soil,'nfer200')     =  N_load('f43',slp,soil,'nfer100')  ;
N_load('f44',slp,soil,'nfer200')     =  N_load('f44',slp,soil,'nfer100')  ;
N_load('f45',slp,soil,'nfer200')     =  N_load('f45',slp,soil,'nfer100')  ;
* Not exact approximation
N_load('f43',slp,soil,'nfer150')     =  N_load('f43',slp,soil,'nfer80')  ;
N_load('f44',slp,soil,'nfer150')     =  N_load('f44',slp,soil,'nfer80')  ;
N_load('f45',slp,soil,'nfer150')     =  N_load('f45',slp,soil,'nfer80')  ;

*NOT EXACT
N_load('f43',slp,soil,'nfer120')     =  [N_load('f43',slp,soil,'nfer50')  + N_load('f43',slp,soil,'nfer80')] / 2 ;
N_load('f44',slp,soil,'nfer120')     =  [N_load('f44',slp,soil,'nfer50')  + N_load('f43',slp,soil,'nfer80')] / 2 ;
N_load('f45',slp,soil,'nfer120')     =  [N_load('f45',slp,soil,'nfer50')  + N_load('f43',slp,soil,'nfer80')] / 2 ;

N_load('f43',slp,soil,'nfer100')     =  N_load('f43',slp,soil,'nfer50')  ;
N_load('f44',slp,soil,'nfer100')     =  N_load('f44',slp,soil,'nfer50')  ;
N_load('f45',slp,soil,'nfer100')     =  N_load('f45',slp,soil,'nfer50')  ;

* Not exact approximation
N_load('f43',slp,soil,'nfer80')     =  [N_load('f43',slp,soil,'nfer50') + N_load('f43',slp,soil,'nfer20')] / 2 ;
N_load('f44',slp,soil,'nfer80')     =  [N_load('f44',slp,soil,'nfer50') + N_load('f44',slp,soil,'nfer20')] / 2 ;
N_load('f45',slp,soil,'nfer80')     =  [N_load('f45',slp,soil,'nfer50') + N_load('f45',slp,soil,'nfer20')] / 2 ;

* Not exact approximation
N_load('f43',slp,soil,'nfer50')     =   N_load('f43',slp,soil,'nfer20')  ;
N_load('f44',slp,soil,'nfer50')     =   N_load('f44',slp,soil,'nfer20')  ;
N_load('f45',slp,soil,'nfer50')     =   N_load('f45',slp,soil,'nfer20')  ;
```

```
* Not exact approximation
N_load('f43',slp,soil,'nfer30')     =  [N_load('f43',slp,soil,'nfer0') + N_load('f43',slp,soil,'nfer20')] / 2 ;
N_load('f44',slp,soil,'nfer30')     =  [N_load('f44',slp,soil,'nfer0') + N_load('f44',slp,soil,'nfer20')] / 2 ;
N_load('f45',slp,soil,'nfer30')     =  [N_load('f45',slp,soil,'nfer0') + N_load('f45',slp,soil,'nfer20')] / 2 ;

N_load('f43',slp,soil,'nfer20')     =  N_load('f43',slp,soil,'nfer10')  ;
N_load('f44',slp,soil,'nfer20')     =  N_load('f44',slp,soil,'nfer10')  ;
N_load('f45',slp,soil,'nfer20')     =  N_load('f45',slp,soil,'nfer10')  ;

*NOT EXACT  -MAYBE REMOVE
N_load('f43',slp,soil,'nfer10')     =  [N_load('f43',slp,soil,'nfer0')  + N_load('f43',slp,soil,'nfer10')] / 2 ;
N_load('f44',slp,soil,'nfer10')     =  [N_load('f44',slp,soil,'nfer0')  + N_load('f44',slp,soil,'nfer10')] / 2 ;
N_load('f45',slp,soil,'nfer10')     =  [N_load('f45',slp,soil,'nfer0')  + N_load('f45',slp,soil,'nfer10')] / 2 ;

*Saving the modified nitrogen load parameters to a separate load parameter
NloadData(j,slp,soil,nfert) = N_load(j,slp,soil,nfert)        ;
NloadData_flw(flw,slp,soil) = N_load_flw(flw,slp,soil,'nfer0')    ;

*REDUCING ADDITIONAL DIMENSION FROM THE FALLOW RUNOFF AND EROSION DATA
* since P fertilisation and soil P status do not effect runoff or erosion in ICECREAM results
sloss__flw(flw,slp,soil)  = sloss_flw(flw,slp,soil,'pac12','pfer15')    ;
runoff__flw(flw,slp,soil) = runoff_flw(flw,slp,soil,'pac12','pfer15')   ;
runoffData(j,slp,soil) = runoff(j,slp,soil,'pac12','pfer15')     ;
erosionData(j,slp,soil) = sloss(j,slp,soil,'pac12','pfer15')     ;

*BARE FALLOW FROM GREEN FALLOW (SINCE THE FINAL ICECREAM RESULTS EXCLUDED BARE FALLOW)
*subsets of the fallow class
set bare(flw) / bf1s, bf2s, bf3s,  bf1n, bf2n, bf3n  /
   grsc(flw) / gf1s, gf2s, gf3s,  gf1n, gf2n, gf3n  /
   subFal(flw) / bf1s, bf2s, bf3s, gf1s, gf2s, gf3s  /
   nsbFal(flw) / bf1n, bf2n, bf3n, gf1n, gf2n, gf3n  /
 ;
parameter gr2brErat(flw,slp,soil), gr2brRrat(flw,slp,soil)  ;
*Missing HsS bare fallow (assume same relative effect as for HtS )
OLD_sloss_flw(bare,slp,'HsS') = OLD_sloss_flw(bare,slp,'HtS')     ;
gr2brErat('bf1n',slp,soil) = OLD_sloss_flw('gf1n',slp,soil)  / OLD_sloss_flw('bf1n',slp,soil)      ;
gr2brErat('bf2n',slp,soil) = OLD_sloss_flw('gf2n',slp,soil)  / OLD_sloss_flw('bf2n',slp,soil)      ;
gr2brErat('bf3n',slp,soil) = OLD_sloss_flw('gf3n',slp,soil)  / OLD_sloss_flw('bf3n',slp,soil)      ;
gr2brErat(bare,slp,'HsS') = gr2brErat(bare,slp,'HtS')     ;
* No difference between subsidy vs no subsidy
gr2brErat('bf1s',slp,soil) = gr2brErat('bf1n',slp,soil) ;
gr2brErat('bf2s',slp,soil) = gr2brErat('bf2n',slp,soil) ;
gr2brErat('bf3s',slp,soil) = gr2brErat('bf3n',slp,soil) ;
*Missing HsS bare fallow (assume same relative effect as for HtS )
OLD_runoff_flw(bare,slp,'HsS') = OLD_runoff_flw(bare,slp,'HtS')     ;
gr2brRrat('bf1n',slp,soil) = OLD_runoff_flw('gf1n',slp,soil)  / OLD_runoff_flw('bf1n',slp,soil)      ;
gr2brRrat('bf2n',slp,soil) = OLD_runoff_flw('gf2n',slp,soil)  / OLD_runoff_flw('bf2n',slp,soil)      ;
gr2brRrat('bf3n',slp,soil) = OLD_runoff_flw('gf3n',slp,soil)  / OLD_runoff_flw('bf3n',slp,soil)      ;
gr2brRrat('bf1s',slp,soil) = gr2brRrat('bf1n',slp,soil) ;
gr2brRrat('bf2s',slp,soil) = gr2brRrat('bf2n',slp,soil) ;
gr2brRrat('bf3s',slp,soil) = gr2brRrat('bf3n',slp,soil) ;
gr2brRrat(bare,slp,'HsS') = gr2brRrat(bare,slp,'HtS')     ;
```

```
sloss__flw(bare,slp,soil) = sloss__flw(bare,slp,soil) / gr2brErat(bare,slp,soil)   ;
runoff__flw(bare,slp,soil) = runoff__flw(bare,slp,soil) / gr2brRrat(bare,slp,soil) ;

display OLD_sloss_flw, OLD_runoff_flw, gr2brErat, gr2brRrat    ;

*Parameters for comparing mean values (mean of the value of each class) with distributed ones
parameters
        meanNitrogenLoad           Crop mean N load (average of class means)
        meanNitrogenLoad_flw        Fallow mean N load (average of class means)
* Total sums of squares between the class mean and the parameter
        totalSumSquaresErosion
        totalSumSquaresRunoff
        totalSumSquaresNload
        totalSumSquaresErosion_flw
        totalSumSquaresRunoff_flw
        totalSumSquaresNload_flw
 ;
 meanErosion      = 1 / [card(j) * card(slp)* card(soil)]  * SUM{(j,slp,soil), erosionData(j,slp,soil) }   ;
 meanErosion_flw   = 1 / [card(flw) * card(slp)* card(soil)]  * SUM{(flw,slp,soil), sloss__flw(flw,slp,soil) } ;
 meanRunoff       = 1 / [card(j) * card(slp)* card(soil)]   * SUM{(j,slp,soil), runoffData(j,slp,soil) }    ;
 meanRunoff_flw    = 1 / [card(flw) * card(slp)* card(soil)] * SUM{(flw,slp,soil), runoff__flw(flw,slp,soil) };

 meanNitrogenLoad  = 1 / [card(j) * card(slp)* card(soil)* card(nfert)]  * SUM{(j,slp,soil,nfert),
NloadData(j,slp,soil,nfert) }   ;
 meanNitrogenLoad_flw = 1 / [card(j) * card(slp)* card(soil)]  * SUM{(flw,slp,soil),
NloadData_flw(flw,slp,soil) }   ;

 totalSumSquaresErosion =   SUM{(j,slp,soil), SQR(erosionData(j,slp,soil) - meanErosion) }   ;
 totalSumSquaresRunoff  =   SUM{(j,slp,soil), SQR(runoffData(j,slp,soil) - meanRunoff) }   ;
 totalSumSquaresNload   =   SUM{(j,slp,soil,nfert), SQR(NloadData(j,slp,soil,nfert) - meanNitrogenLoad) }   ;

 totalSumSquaresErosion_flw =   SUM{(flw,slp,soil), SQR(sloss__flw(flw,slp,soil) - meanErosion_flw) }   ;
 totalSumSquaresRunoff_flw  =   SUM{(flw,slp,soil), SQR(runoff__flw(flw,slp,soil) - meanRunoff_flw) }   ;
 totalSumSquaresNload_flw   =   SUM{(flw,slp,soil), SQR(Nloaddata_flw(flw,slp,soil) -
meanNitrogenLoad_flw) }   ;

*TESTED SCALING THE RESULTS BASED ON LOAD PARAMETER VALUES BY PUUSTINEN
*parameters
*         sloss_flw_relf13slp0HsS(flw,slp,soil)     Erosion of fallow as share of conventional barley HsS slp1
*         erosion_brl_slp0_HsS                  Erosion of barley on slope 1 and HsS soil
*         erosiondata_rel_f13slp0HsS(j,slp,soil)    Erosion as share of conventional barley HsS slp1
*         erosiondata_rel_gf1slp0HsS(flw,slp,soil)  Erosion as share of grass ley HsS slp1
*         sloss__flw_noscaling(flw,slp,soil)       Erosion before scaling with liperi results

*erosion_brl_slp0_HsS  = erosiondata('f13','slp0','HsS') ;
*erosiondata_rel_f13slp0HsS(j,slp,soil) = erosiondata(j,slp,soil)/ erosion_brl_slp0_HsS  ;
* SET BASELINE AS LIPERI CONVENTIONAL PLOUGH EROSION 125 (TABLE FROM PUUSTINEN)
*Keeping the original erosion data for comparison
*erosiondata_original(j,slp,soil) = erosiondata(j,slp,soil)             ;
*erosiondata(j,slp,soil) = 125 * erosiondata_rel_f13slp0HsS(j,slp,soil)  ;
*TEST SCALE ALSO FALLOW TO BARLEY
*sloss_flw_relf13slp0HsS(flw,slp,soil)= sloss__flw(flw,slp,soil) / erosion_brl_slp0_HsS ;
```

```
*sloss__flw_noscaling(flw,slp,soil)   =  sloss__flw(flw,slp,soil)        ;
*display  erosion_brl_slp0_HsS, erosiondata_rel_f13slp0HsS, erosiondata_original ;
*display sloss__flw_noscaling, sloss__flw, sloss_flw_relf13slp0HsS                    ;

display runoff, runoffdata, sloss, erosiondata       ;

VARIABLES
*Coefficients to be estimated for crops
P_coeff_slp_b(j,soil,pac,pfert)        Coefficient b (P load as linear function of slope)
P_coeff_slp_c(j,soil,pac,pfert)        Coefficient c (P load as linear function of slope)
P_coeff_soilP_a(j,soil,slp,pfert)      Coefficient a (P load as function of soil P)
P_coeff_soilP_b(j,soil,slp,pfert)      Coefficient b (P load as function of soil P)
P_coeff_fertP_a(j,soil,slp,pac)        Coefficient a (P load as function of fertilisation P)
P_coeff_fertP_b(j,soil,slp,pac)        Coefficient b (P load as function of fertilisation P)
E_coeff_slp_b_lin(j,soil)              Erosion coefficient b for lin func form
E_coeff_slp_c_lin(j,soil)              Erosion coefficient c for lin func form
E_coeff_slp_b_exp(j,soil)              Erosion coefficient b for exp func form
E_coeff_slp_c_exp(j,soil)              Erosion coefficient c for exp func form
E_coeff_slp_b_sqr(j,soil)              Erosion coefficient b for squared func form
E_coeff_slp_c_sqr(j,soil)              Erosion coefficient c for squared func form
R_coeff_slp_b(j,soil)                  Runoff coefficient b (as a function of slope)
R_coeff_slp_c(j,soil)                  Runoff coefficient c (as a function of slope)
*Coefficients to be estimated for fallow
P_coeff_slp_b_flw(flw,soil,pac,pfert)    Linear slope func coefficient b (as a function of slope)
P_coeff_slp_c_flw(flw,soil,pac,pfert)    Linear slope func coefficient c (as a function of slope)
E_coeff_slp_b_flw_lin(flw,soil)        Erosion coefficient b (as a function of slope)
E_coeff_slp_c_flw_lin(flw,soil)        Erosion coefficient c (as a function of slope)
E_coeff_slp_b_flw_exp(flw,soil)        Erosion coefficient b (as a function of slope)
E_coeff_slp_c_flw_exp(flw,soil)        Erosion coefficient c (as a function of slope)
E_coeff_slp_b_flw_sqr(flw,soil)        Erosion coefficient b (as a function of slope)
E_coeff_slp_c_flw_sqr(flw,soil)        Erosion coefficient c (as a function of slope)
R_coeff_slp_b_flw(flw,soil)            Runoff coefficient  b (as a function of slope)
R_coeff_slp_c_flw(flw,soil)            Runoff coefficient  c (as a function of slope)
P_coeff_soilP_a_flw(flw,soil,slp,pfert)  P load coefficient  a (as a function of soil P ) fallow
P_coeff_soilP_b_flw(flw,soil,slp,pfert)  P load coefficient  b (as a function of soil P ) fallow
*For Nitrogen load coefficients
N_coeff_slp_b(j,soil,nfert)            Coefficient (for N as function of slope)
N_coeff_slp_c(j,soil,nfert)            Coefficient (for N as function of slope)
N_coeff_frt_b(j,soil,slp)              Coefficient (for N as function of fertilisation)
N_coeff_frt_c(j,soil,slp)              Coefficient (for N as function of fertilisation)
N_coeff_slp_b_flw(flw,soil,nfert)      Coefficient (fallow for N as function of slope)
N_coeff_slp_c_flw(flw,soil,nfert)      Coefficient (fallow for N as function of slope)
* Residuals (variables to be minimised)
residualtermN(j,slp,soil,nfert)        N residuals crops
residualtermN_flw(flw,slp,soil,nfert)    N residuals fallow
residual(slp)                          residual
*Separate variables for residuals for avoiding giving bad initial values between solves
Eresidualterm(j,slp,soil)              erosion residual term
Rresidualterm(j,slp,soil)              runoff residual term
residualterm(j,slp,soil,pac,pfert)     residual term for crops
residualterm_flw(flw,slp,soil,pac,pfert) residual term for fallow
Eresidualterm_flw(flw,slp,soil)        erosion residual term for fallow
```

```
Rresidualterm_flw(flw,slp,soil)        runoff residual term for fallow
OLS_N                              ordinary sum of least squares (objective to be minimised)
OLS_N_flw                            ordinary sum of least squares (objective to be minimised)
OLS_P                              ordinary sum of least squares (objective to be minimised)
OLS_P_flw                            ordinary sum of least squares (objective to be minimised)
;

Positive variables P_coeff_soilP_a,E_coeff_slp_b_lin,E_coeff_slp_c_lin
E_coeff_slp_b_exp,E_coeff_slp_c_exp,E_coeff_slp_b_exp2,E_coeff_slp_c_exp2
E_coeff_slp_b_sqr,E_coeff_slp_c_sqr,E_coeff_slp_b_flw_lin,E_coeff_slp_c_flw_lin
E_coeff_slp_b_flw_exp,E_coeff_slp_c_flw_exp,E_coeff_slp_b_flw_exp2,E_coeff_slp_c_flw_exp2
E_coeff_slp_b_flw_sqr,E_coeff_slp_c_flw_sqr,N_coeff_frt_b(j,soil,slp),N_coeff_frt_c(j,soil,slp)
;
Free variable OLS_P, OLS_N, OLS_N_flw, OLS_P_flw      ;

*Equation declarations
EQUATIONS
OBJF_P12        the objective function of least squares to be minimised
OBJF_P          the objective function of least squares to be minimised   (for P load)
OBJF_R          the objective function of least squares to be minimised  (for runoff)
OBJF_E          the objective function of least squares to be minimised  (for erosion)
OBJF_P_flw      the objective function of least squares to be minimised   (for P load) fallow
OBJF_E_flw      the objective function of least squares to be minimised  (for runoff)  fallow
OBJF_R_flw      the objective function of least squares to be minimised  (for erosion) fallow
OBJF_N          the objective function of least squares to be minimised
OBJF_N_flw      the objective function of least squares to be minimised fallow
TPA_P12(slp)                         effect of tillage as slopes vary
P_slope_lin(j,slp,soil,pac,pfert)       effect of slope on P load
runoff_slope_lin(j,slp,soil)           effect of slope on runoff
erosion_slope_lin(j,slp,soil)          effect of slope on erosion
erosion_slope_exp(j,slp,soil)          effect of slope on erosion exp
erosion_slope_exp2(j,slp,soil)          effect of slope on erosion exp (a(x) = c1*exp(x*d1))
erosion_slope_sqr(j,slp,soil)          effect of slope on erosion squared
P_soilP_exp(j,slp,soil,pac,pfert)       effect of soil P status  on P load
P_fertP_lin(j,slp,soil,pac,pfert)       effect of soil P status  on P load
flw_P_slope_lin(flw,slp,soil,pac,pfert)  effect of slope on P load
flw_erosion_slope_exp(flw,slp,soil)      effect of slope on erosion exp
flw_erosion_slope_exp2(flw,slp,soil)     effect of slope on erosion exp (a(x) = c1*exp(x*d1))
flw_erosion_slope_sqr(flw,slp,soil)      effect of slope on erosion squared
flw_erosion_slope_lin(flw,slp,soil)      effect of slope on erosion linear
flw_runoff_slope_lin(flw,slp,soil)       effect of slope on runoff linear
flw_P_soilP_exp(flw,slp,soil,pac,pfert)
*Nitrogen load equations
N_slope_lin(j,slp,soil,nfert)          Nitrogen load as function of slope
flw_N_slope_lin(flw,slp,soil,nfert)      Nitrogen load as function of slope (fallow)
N_fert_exp(j,slp,soil,nfert)           Nitrogen load as function of fertilisation -> next file
;
*EQUATIONS---------------------------------------------------------------------
*Objective functions
OBJF_P..        OLS_P =E= SUM{(j,slp,soil,pac,pfert), SQR[residualterm(j,slp,soil,pac,pfert)]}    ;
OBJF_R..        OLS_P =E= SUM{(j,slp,soil), SQR[Rresidualterm(j,slp,soil)]}    ;
OBJF_E..        OLS_P =E= SUM{(j,slp,soil), SQR[Eresidualterm(j,slp,soil)]}    ;
```

```
OBJF_N..        OLS_N =E= SUM{(j,slp,soil,nfert), SQR[residualtermN(j,slp,soil,nfert)]}    ;
OBJF_N_flw..    OLS_N_flw =E= SUM{(flw,slp,soil,nfert), SQR[residualtermN_flw(flw,slp,soil,nfert)]}    ;
OBJF_P_flw..    OLS_P_flw =E= SUM{(flw,slp,soil,pac,pfert), SQR[residualterm_flw(flw,slp,soil,pac,pfert)]}    ;
OBJF_E_flw..    OLS_P_flw =E= SUM{(flw,slp,soil), SQR[Eresidualterm_flw(flw,slp,soil)]}    ;
OBJF_R_flw..    OLS_P_flw =E= SUM{(flw,slp,soil), SQR[Rresidualterm_flw(flw,slp,soil)]}    ;
*OBJF_P..   OLS_P =E= SUM{(j,slp,soilfinn,PAC), SQR[TP_load(j,slp,soilfinn,PAC) - Pload(j,slp,soilfinn,PAC)]}
;

*Load functions to be estimated---------------------------------------------
N_fert_exp(j,slp,soil,nfert)..       N_load(j,slp,soil,nfert) =E=  N_coeff_frt_b(j,soil,slp) *
exp[N_fert(j,slp,soil,nfert) * N_coeff_frt_c(j,soil,slp)] + residualtermN(j,slp,soil,nfert)    ;
*flw_N_fert_exp(flw,slp,soil,nfert).. N_load_flw(flw,slp,soil,nfert) =E= N_coeff_frt_b_flw(flw,soil,nfert) *
exp[ slopedata(slp) + N_coeff_frt_c_flw(flw,soil,nfert)] + residualtermN_flw(flw,slp,soil,nfert)    ;
N_slope_lin(j,slp,soil,nfert).. N_load(j,slp,soil,nfert) =E= N_coeff_slp_b(j,soil,nfert) * slopedata(slp) +
N_coeff_slp_c(j,soil,nfert) + residualtermN(j,slp,soil,nfert)    ;
*TPA_P(j,slp,soilfinn,PAC)..    Pload(j,slp,soilfinn,PAC) =E= P_coeff_a(j,soilfinn,PAC) *
SQR[slopepoint(j,slp,soilfinn,PAC)]  + P_coeff_b(j,soilfinn,PAC) * slopepoint(j,slp,soilfinn,PAC) +
P_coeff_c(j,soilfinn,PAC)  ;
P_slope_lin(j,slp,soil,pac,pfert).. TP_load(j,slp,soil,pac,pfert) =E= P_coeff_slp_b(j,soil,pac,pfert) *
slopedata(slp) + P_coeff_slp_c(j,soil,pac,pfert) + residualterm(j,slp,soil,pac,pfert)    ;
P_soilP_exp(j,slp,soil,pac,pfert).. TP_load(j,slp,soil,pac,pfert) =E= P_coeff_soilP_a(j,soil,slp,pfert) *
log(soilP(j,slp,soil,pac,pfert)) + P_coeff_soilP_b(j,soil,slp,pfert) + residualterm(j,slp,soil,pac,pfert)    ;
P_fertP_lin(j,slp,soil,pac,pfert).. TP_load(j,slp,soil,pac,pfert) =E= P_coeff_fertP_a(j,soil,slp,pac) *
fertP(j,slp,soil,pac,pfert) + P_coeff_fertP_b(j,soil,slp,pac) + residualterm(j,slp,soil,pac,pfert)    ;
* Crop runoff and erosion
runoff_slope_lin(j,slp,soil)..   runoffdata(j,slp,soil)  =E= R_coeff_slp_b(j,soil)      * slopedata(slp)      +
R_coeff_slp_c(j,soil)     + Rresidualterm(j,slp,soil)    ;
erosion_slope_lin(j,slp,soil)..  erosiondata(j,slp,soil) =E= E_coeff_slp_b_lin(j,soil)  * slopedata(slp)      +
E_coeff_slp_c_lin(j,soil) + Eresidualterm(j,slp,soil)    ;
erosion_slope_exp(j,slp,soil)..  erosiondata(j,slp,soil) =E= E_coeff_slp_b_exp(j,soil)  * exp[slopedata(slp)] +
E_coeff_slp_c_exp(j,soil) + Eresidualterm(j,slp,soil)    ;
erosion_slope_exp2(j,slp,soil).. erosiondata(j,slp,soil) =E= E_coeff_slp_b_exp2(j,soil) * exp[slopedata(slp)  *
E_coeff_slp_c_exp2(j,soil)] + Eresidualterm(j,slp,soil)  ;
erosion_slope_sqr(j,slp,soil)..  erosiondata(j,slp,soil) =E= E_coeff_slp_b_sqr(j,soil)  * sqr[slopedata(slp)] +
E_coeff_slp_c_sqr(j,soil) + Eresidualterm(j,slp,soil)    ;
* Fallow nitrogen load, runoff and erosion and TP load
flw_N_slope_lin(flw,slp,soil,nfert).. N_load_flw(flw,slp,soil,nfert) =E= N_coeff_slp_b_flw(flw,soil,nfert) *
slopedata(slp) + N_coeff_slp_c_flw(flw,soil,nfert) + residualtermN_flw(flw,slp,soil,nfert)    ;
flw_P_slope_lin(flw,slp,soil,pac,pfert).. TP_load_flw(flw,slp,soil,pac,pfert) =E=
P_coeff_slp_b_flw(flw,soil,pac,pfert) * slopedata(slp) + P_coeff_slp_c_flw(flw,soil,pac,pfert) +
residualterm_flw(flw,slp,soil,pac,pfert)    ;
flw_erosion_slope_exp(flw,slp,soil).. sloss__flw(flw,slp,soil) =E= E_coeff_slp_b_flw_exp(flw,soil) *
exp[slopedata(slp)] + E_coeff_slp_c_flw_exp(flw,soil) + Eresidualterm_flw(flw,slp,soil)    ;
flw_erosion_slope_exp2(flw,slp,soil).. sloss__flw(flw,slp,soil) =E= E_coeff_slp_b_flw_exp2(flw,soil) *
exp[slopedata(slp) * E_coeff_slp_c_flw_exp2(flw,soil)] + Eresidualterm_flw(flw,slp,soil)    ;
flw_erosion_slope_sqr(flw,slp,soil).. sloss__flw(flw,slp,soil) =E= E_coeff_slp_b_flw_sqr(flw,soil) *
sqr[slopedata(slp)] + E_coeff_slp_c_flw_sqr(flw,soil) + Eresidualterm_flw(flw,slp,soil)    ;
flw_erosion_slope_lin(flw,slp,soil).. sloss__flw(flw,slp,soil) =E= E_coeff_slp_b_flw_lin(flw,soil) *
slopedata(slp) + E_coeff_slp_c_flw_lin(flw,soil) + Eresidualterm_flw(flw,slp,soil)    ;
flw_runoff_slope_lin(flw,slp,soil).. runoff__flw(flw,slp,soil) =E= R_coeff_slp_b_flw(flw,soil) * slopedata(slp)
+ R_coeff_slp_c_flw(flw,soil) + Rresidualterm_flw(flw,slp,soil)        ;
```

```
flw_P_soilP_exp(flw,slp,soil,pac,pfert).. TP_load_flw(flw,slp,soil,pac,pfert) =E=
P_coeff_soilP_a_flw(flw,soil,slp,pfert) * log(soilP_flw(flw,slp,soil,pac,pfert)) +
P_coeff_soilP_b_flw(flw,soil,slp,pfert) + residualterm_flw(flw,slp,soil,pac,pfert)    ;

*Different model specifications
*CROP P
MODEl P_OLS_SLOPE_LIN    / OBJF_P, P_slope_lin /                  ;
MODEl P_OLS_SOILP_EXP    / OBJF_P, P_soilP_exp /                  ;
MODEl P_OLS_FERTP_LIN    / OBJF_P, P_fertP_lin /                  ;
MODEl RUN_OLS_SLOPE_LIN  / OBJF_R, runoff_slope_lin /               ;
MODEl ERO_OLS_SLOPE_LIN  / OBJF_E, erosion_slope_lin /              ;
MODEl ERO_OLS_SLOPE_EXP  / OBJF_E, erosion_slope_exp /              ;
MODEl ERO_OLS_SLOPE_EXP2  / OBJF_E, erosion_slope_exp2 /            ;
MODEl ERO_OLS_SLOPE_SQR  / OBJF_E, erosion_slope_sqr /              ;
*FALLOW P
MODEl P_OLS_SLOPE_LIN_FLW    / OBJF_P_flw, flw_P_slope_lin /        ;
MODEl P_OLS_SOILP_EXP_FLW    / OBJF_P_flw, flw_P_soilP_exp /        ;
MODEl RUN_OLS_SLOPE_LIN_FLW  / OBJF_R_flw, flw_runoff_slope_lin /      ;
MODEl ERO_OLS_SLOPE_EXP_FLW  / OBJF_E_flw, flw_erosion_slope_exp /     ;
MODEl ERO_OLS_SLOPE_LIN_FLW  / OBJF_E_flw, flw_erosion_slope_lin /     ;
MODEl ERO_OLS_SLOPE_SQR_FLW  / OBJF_E_flw, flw_erosion_slope_sqr /     ;
*CROP N
MODEl N_OLS_FERT_EXP       / OBJF_N, N_fert_exp /                 ;
MODEl ERO_OLS_SLOPE_EXP2_FLW  / OBJF_E_flw, flw_erosion_slope_exp2 /    ;
MODEl N_OLS_SLOPE_LIN       / OBJF_N, N_slope_lin /                ;
*FALLOW N
MODEl N_OLS_SLOPE_LIN_FLW   / OBJF_N_flw, flw_N_slope_lin /         ;

*Parameters for results and reports -values predicted by the estimated functions
parameters
       P_load_slp(j,slp,soil,pac,pfert)            predicted P load
       P_load_fert(j,slp,soil,pac,pfert)           predicted P load
       TPloadPredDifference(j,slp,soil,pac,pfert)    difference in predicted P loads (slope - fert)
       Prunoff(j,slp,soil)                    predicted runoff from the parcel
       Perosion(j,slp,soil)                   predicted erosion from the parcel
       Perosion_lin(j,slp,soil)                predicted erosion from the parcel
       Perosion_exp(j,slp,soil)                predicted erosion from the parcel
       Perosion_exp2(j,slp,soil)               predicted erosion from the parcel
       Perosion_sqr(j,slp,soil)                predicted erosion from the parcel
       P_load_slp_flw(flw,slp,soil,pac,pfert)       predicted P load
       Prunoff_flw(flw,slp,soil)               predicted runoff from the parcel of fallow
       Perosion_flw(flw,slp,soil)              predicted erosion from the parcel of fallow
       Perosion_flw_lin(flw,slp,soil)           predicted erosion from the parcel of fallow
       Perosion_flw_exp(flw,slp,soil)           predicted erosion from the parcel of fallow
       Perosion_flw_exp2(flw,slp,soil)          predicted erosion from the parcel of fallow
       Perosion_flw_sqr(flw,slp,soil)           predicted erosion from the parcel of fallow
       Nload(j,slp,soil,nfert)                 predicted N load
       sse_erosio_exp2
       sse_runoff_lin
       sse_nitron_exp2
*        R2_erosio_exp2                   R2 values (not calculated)
*        R2_runoff_lin                   R2 values (not calculated)
```

```
*        R2_nitron_exp2                      R2 values (not calculated)
;

*VARIABLE INITIAL VALUES-----------------------------------------------------
P_coeff_soilP_a_flw.L(flw,soil,slp,pfert) = 1  ;
P_coeff_soilP_b_flw.L(flw,soil,slp,pfert)  = 1 ;
P_coeff_soilP_a_flw.lo(flw,soil,slp,pfert) = 0.0001  ;
P_coeff_soilP_b_flw.lo(flw,soil,slp,pfert)  = 0.0001  ;
P_coeff_soilP_a.L(j,soil,slp,pfert) = 1  ;
P_coeff_soilP_b.L(j,soil,slp,pfert)  = 1 ;
P_coeff_soilP_a.lo(j,soil,slp,pfert) = 0.0001  ;
P_coeff_soilP_b.lo(j,soil,slp,pfert)  = 0.0001  ;

*For the exponential functional form
*(estimation for exp currently fails)
E_coeff_slp_b_exp.L(j,soil) = 0.5  ;
E_coeff_slp_c_exp.L(j,soil) = 10  ;
E_coeff_slp_b_exp.lo(j,soil) = 0.0001  ;
E_coeff_slp_c_exp.lo(j,soil) = 0.0001  ;

E_coeff_slp_b_exp2.L(j,soil) = 10  ;
E_coeff_slp_c_exp2.L(j,soil) = 0.05  ;
E_coeff_slp_b_exp2.lo(j,soil) = 0.000001  ;
E_coeff_slp_c_exp2.lo(j,soil) = 0.000001  ;

E_coeff_slp_b_sqr.L(j,soil) = 10  ;
*E_coeff_slp_c_sqr.L(j,soil) = 5  ;
*E_coeff_slp_b_sqr.lo(j,soil) = 0.0001  ;
*E_coeff_slp_c_sqr.lo(j,soil) = 0.0001  ;

E_coeff_slp_b_lin.L(j,soil) = 10  ;
*E_coeff_slp_c_lin.L(j,soil) = 5  ;
*E_coeff_slp_b_lin.lo(j,soil) = 0.0001  ;
*E_coeff_slp_c_lin.lo(j,soil) = 0.0001  ;

*option Rtmaxv=1.5e+11 ;

*SOLVING THE LEAST SQUARE MINIMISATION PROBLEMS

*TOT P                              ;
SOLVE P_OLS_SLOPE_LIN_FLW using NLP minimizing OLS_P_flw              ;
*SOLVE P_OLS_SOILP_EXP_FLW using NLP minimizing OLS_P_flw              ;
SOLVE P_OLS_SLOPE_LIN using NLP minimizing OLS_P                      ;
SOLVE P_OLS_SOILP_EXP using NLP minimizing OLS_P                      ;
SOLVE P_OLS_FERTP_LIN using NLP minimizing OLS_P                      ;
*RUNOFF
SOLVE RUN_OLS_SLOPE_LIN using NLP minimizing OLS_P                    ;
sse_runoff_lin = OLS_P.L                                      ;
R2_runoff_lin  = 1 -  sse_runoff_lin / totalSumSquaresRunoff          ;

SOLVE RUN_OLS_SLOPE_LIN_FLW  using NLP minimizing OLS_P_flw           ;
```

```
*EROSION
SOLVE ERO_OLS_SLOPE_EXP using NLP minimizing OLS_P                    ;
SOLVE ERO_OLS_SLOPE_EXP2 using NLP minimizing OLS_P                   ;
sse_erosio_exp2 = OLS_P.L                                   ;
R2_erosio_exp2  =   1 - sse_erosio_exp2 / totalSumSquaresErosion        ;

*Eresidualterm.L(j,slp,soil) = 1000 ;
SOLVE ERO_OLS_SLOPE_LIN using NLP minimizing OLS_P                    ;
*Eresidualterm.L(j,slp,soil) = 1000 ;
SOLVE ERO_OLS_SLOPE_SQR using NLP minimizing OLS_P                    ;
SOLVE ERO_OLS_SLOPE_EXP_FLW  using NLP minimizing OLS_P_flw            ;
SOLVE ERO_OLS_SLOPE_EXP2_FLW  using NLP minimizing OLS_P_flw           ;
SOLVE ERO_OLS_SLOPE_LIN_FLW  using NLP minimizing OLS_P_flw            ;
SOLVE ERO_OLS_SLOPE_SQR_FLW  using NLP minimizing OLS_P_flw            ;

*NITROGEN
SOLVE N_OLS_FERT_EXP using NLP minimizing OLS_N                       ;

sse_nitron_exp2 = OLS_N.L                                   ;
R2_nitron_exp2   = 1 - sse_nitron_exp2 / totalSumSquaresNload          ;


SOLVE N_OLS_SLOPE_LIN using NLP minimizing OLS_N                      ;
SOLVE N_OLS_SLOPE_LIN_FLW using NLP minimizing OLS_N_flw              ;

*CALCULATE LOAD PARAMETERS BASES ON ESTIMATED EQUATIONS (COEFFICIENTS)----------
P_load_slp_flw(flw,slp,soil,pac,pfert) =  P_coeff_slp_b_flw.L(flw,soil,pac,pfert) * slopedata(slp) +
P_coeff_slp_c_flw.L(flw,soil,pac,pfert)   ;

*Predicted erosion for crop (for different function form estimates)
Perosion_lin(j,slp,soil)  = E_coeff_slp_b_lin.L(j,soil)  * slopedata(slp)      + E_coeff_slp_c_lin.L(j,soil)   ;
Perosion_exp(j,slp,soil)  = E_coeff_slp_b_exp.L(j,soil)  * exp[slopedata(slp)] + E_coeff_slp_c_exp.L(j,soil)   ;
Perosion_exp2(j,slp,soil) = E_coeff_slp_b_exp2.L(j,soil) * exp[slopedata(slp)  * E_coeff_slp_c_exp2.L(j,soil)] ;
Perosion_sqr(j,slp,soil)  = E_coeff_slp_b_sqr.L(j,soil)  * sqr[slopedata(slp)] + E_coeff_slp_c_sqr.L(j,soil)   ;
*Predicted erosion for fallow (for different function form estimates)
Perosion_flw_exp(flw,slp,soil)  =  E_coeff_slp_b_flw_exp.L(flw,soil)  * exp[slopedata(slp)] +
E_coeff_slp_c_flw_exp.L(flw,soil)   ;
Perosion_flw_exp2(flw,slp,soil) =  E_coeff_slp_b_flw_exp2.L(flw,soil) * exp[slopedata(slp) *
E_coeff_slp_c_flw_exp2.L(flw,soil)]  ;
Perosion_flw_lin(flw,slp,soil)  =  E_coeff_slp_b_flw_lin.L(flw,soil)  * slopedata(slp) +
E_coeff_slp_c_flw_lin.L(flw,soil)       ;
Perosion_flw_sqr(flw,slp,soil)  =  E_coeff_slp_b_flw_sqr.L(flw,soil)  * sqr[slopedata(slp)] +
E_coeff_slp_c_flw_sqr.L(flw,soil)   ;
*Predicted runoff for crop (as linear function of slope)
Prunoff(j,slp,soil) = R_coeff_slp_b.L(j,soil) * slopedata(slp) + R_coeff_slp_c.L(j,soil)   ;
*Predicted runoff for fallow (as linear function of slope)
Prunoff_flw(flw,slp,soil) = R_coeff_slp_b_flw.L(flw,soil) * slopedata(slp) + R_coeff_slp_c_flw.L(flw,soil)   ;

*Predicted P load (as linear function of slope) NOTE THAT THIS PREDICTION IS NOT USED IN THE ACTUAL
LOAD CALCULATION
* It is for comparing with the results based on erosion and runoff derived P-load
```

```
P_load_slp(j,slp,soil,pac,pfert)  =  P_coeff_slp_b.L(j,soil,pac,pfert) * slopedata(slp) +
P_coeff_slp_c.L(j,soil,pac,pfert)   ;
P_load_fert(j,slp,soil,pac,pfert) = P_coeff_fertP_a.L(j,soil,slp,pac)  * fertP(j,slp,soil,pac,pfert) +
P_coeff_fertP_b.L(j,soil,slp,pac)    ;
TPloadPredDifference(j,slp,soil,pac,pfert) = P_load_slp(j,slp,soil,pac,pfert)  - P_load_fert(j,slp,soil,pac,pfert)
;

*Predicted N load ( as linear function of slope) This load estimate is used in extrapolate file to calculate
Nload = b*exp(Nfert*c)
Nload(j,slp,soil,nfert) = N_coeff_slp_b.L(j,soil,nfert) * slopedata(slp) + N_coeff_slp_c.L(j,soil,nfert)


*PASSING LOAD COEFFICIENTS FROM VARIABLES TO PARAMETERS  (which will be passed to the
extrapolation file)
parameters
*P-load coeffcients (for comparison purposes)
        Pcoeff_slp_b(j,soil,pac,pfert)       Coefficient b (for TP = b * slope + c)
        Pcoeff_slp_c(j,soil,pac,pfert)       Coefficient c (for TP = b * slope + c)
        Pcoeff_fertP_a(j,soil,slp,pac)       Coefficient a (for TP = a * Pfert + b)
        Pcoeff_fertP_b(j,soil,slp,pac)       Coefficient b (for TP = a * Pfert + b)
        Pcoeff_soilP_a(j,soil,slp,pfert)     Coefficient a (for TP = a * log(soilP) + b)
        Pcoeff_soilP_b(j,soil,slp,pfert)     Coefficient b (for TP = a * log(soilP) + b)
        Pcoeff_slp_b_flw(flw,soil,pac,pfert)  Coefficient b (for TP = b * slope + c)  (fallow)
        Pcoeff_slp_c_flw(flw,soil,pac,pfert)  Coefficient c (for TP = b * slope + c)  (fallow)
        Pcoeff_soilP_a_flw(flw,soil,slp,pfert) Coefficient a (for TP = a * log(soilP) + b) (fallow)
        Pcoeff_soilP_b_flw(flw,soil,slp,pfert) Coefficient b (for TP = a * log(soilP) + b) (fallow)
*Coefficients for erosion with alternative functional forms
        Ecoeff_slp_b_lin(j,soil)           Coefficient b (for erosion(y) = b * slope + c)
        Ecoeff_slp_c_lin(j,soil)           Coefficient c (for erosion(y) = b * slope + c)
        Ecoeff_slp_b_exp(j,soil)           Coefficient b (for erosion(y) = b*exp(x) + c )
        Ecoeff_slp_c_exp(j,soil)           Coefficient c (for erosion(y) = b*exp(x) + c )
        Ecoeff_slp_b_exp2(j,soil)          Coefficient b (for erosion(y) = b*exp(x*c)  )
        Ecoeff_slp_c_exp2(j,soil)          Coefficient c (for erosion(y) = b*exp(x*c)  )
        Ecoeff_slp_b_sqr(j,soil)           Coefficient b (for erosion(y) = b*sqr(x)+ c )
        Ecoeff_slp_c_sqr(j,soil)           Coefficient c (for erosion(y) = b*sqr(x) + c )
        Ecoeff_slp_b_flw_lin(flw,soil)       Coefficient b (for erosion(y) = b * x + c (fallow) )
        Ecoeff_slp_c_flw_lin(flw,soil)       Coefficient c (for erosion(y) = b * x + c (fallow) )
        Ecoeff_slp_b_flw_exp(flw,soil)       Coefficient b (for erosion(y) = b*exp(x) + c (fallow) )
        Ecoeff_slp_c_flw_exp(flw,soil)       Coefficient c (for erosion(y) = b*exp(x) + c (fallow) )
        Ecoeff_slp_b_flw_exp2(flw,soil)      Coefficient b (for erosion(y) = b*exp(x*c) (fallow) )
        Ecoeff_slp_c_flw_exp2(flw,soil)      Coefficient c (for erosion(y) = b*exp(x*c) (fallow) )
        Ecoeff_slp_b_flw_sqr(flw,soil)       Coefficient b (for erosion(y) = b*sqr(x)+ c (fallow) )
        Ecoeff_slp_c_flw_sqr(flw,soil)       Coefficient c (for erosion(y) = b*sqr(x)+ c (fallow) )
*Erosion (to be passes to the extrapolate file)
        Ecoeff_slp_b(j,soil)               The chosen erosion coefficient b (between different functional forms)
        Ecoeff_slp_c(j,soil)               The chosen erosion coefficient c (between different functional forms)
        Ecoeff_slp_b_flw(flw,soil)         The chosen erosion coefficient b (between different functional
forms)  fallow
        Ecoeff_slp_c_flw(flw,soil)         The chosen erosion coefficient c (between different functional forms)
fallow
*Runoff  (used in the extrapolation file)
        Rcoeff_slp_b(j,soil)               Coefficient b (for runoff(y) = b * slope + c           )
```

```
        Rcoeff_slp_c(j,soil)              Coefficient c (for runoff(y) = b * slope + c           )
        Rcoeff_slp_b_flw(flw,soil)          Coefficient b (for runoff(y) = b * slope + c   (fallow)  )
        Rcoeff_slp_c_flw(flw,soil)          Coefficient b (for runoff(y) = b * slope + c   (fallow)  )
*Nitrogen cofficients to parameters
        Ncoeff_frt_b(j,soil,slp)           Coefficient b (for Nload(y) = b*exp(Nfert*c)  ) This estimation is redone
in extrapolation file
        Ncoeff_frt_c(j,soil,slp)           Coefficient c (for Nload(y) = b*exp(Nfert*c)  ) This estimation is redone
in extrapolation file
        Ncoeff_slp_b(j,soil,nfert)          Coefficient b (for Nload(y) = b * slope + c)  These esimates are used
        Ncoeff_slp_c(j,soil,nfert)          Coefficient c (for Nload(y) = b * slope + c)  These esimates are used
        Ncoeff_slp_b_flw(flw,soil,nfert)    Coefficient b (for Nload(y) = b * slope + c) (fallow)
        Ncoeff_slp_c_flw(flw,soil,nfert)    Coefficient c (for Nload(y) = b * slope + c) (fallow)
* Parameters for some result checks
        TPL_soilP(j,slp,soil,pac,pfert)     total P load when calculated from the soilP regression param
        NegTPloads(j,slp,soil,pac,pfert)    check for negative load values from the soil P estimate
        sumNegTPloads
        P_load_slp_flw(flw,slp,soil,pac,pfert)  predicted P load from slope function (fallow)
        P_load_fert_flw(flw,slp,soil,pac,pfert) predicted P load from Pfert function (fallow)
;
*crop coeffcient results to parameters
Pcoeff_slp_b(j,soil,pac,pfert)   = P_coeff_slp_b.L(j,soil,pac,pfert)     ;
Pcoeff_slp_c(j,soil,pac,pfert)   = P_coeff_slp_c.L(j,soil,pac,pfert)     ;
Pcoeff_fertP_a(j,soil,slp,pac)   = P_coeff_fertP_a.L(j,soil,slp,pac)     ;
Pcoeff_fertP_b(j,soil,slp,pac)   = P_coeff_fertP_b.L(j,soil,slp,pac)     ;
Pcoeff_soilP_a(j,soil,slp,pfert) = P_coeff_soilP_a.L(j,soil,slp,pfert)   ;
Pcoeff_soilP_b(j,soil,slp,pfert) = P_coeff_soilP_b.L(j,soil,slp,pfert)   ;
TPL_soilP(j,slp,soil,pac,pfert)  = Pcoeff_soilP_a(j,soil,slp,pfert) * log(soilP(j,slp,soil,pac,pfert)) +
P_coeff_soilP_b.L(j,soil,slp,pfert)   ;
Rcoeff_slp_b(j,soil) = R_coeff_slp_b.L(j,soil)  ;
Rcoeff_slp_c(j,soil) = R_coeff_slp_c.L(j,soil)  ;

*Erosion coefficients - crops
Ecoeff_slp_b_lin(j,soil)  = E_coeff_slp_b_lin.L(j,soil)  ;
Ecoeff_slp_c_lin(j,soil)  = E_coeff_slp_c_lin.L(j,soil)  ;
Ecoeff_slp_b_exp(j,soil)  = E_coeff_slp_b_exp.L(j,soil)  ;
Ecoeff_slp_c_exp(j,soil)  = E_coeff_slp_c_exp.L(j,soil)  ;
Ecoeff_slp_b_exp2(j,soil) = E_coeff_slp_b_exp2.L(j,soil) ;
Ecoeff_slp_c_exp2(j,soil) = E_coeff_slp_c_exp2.L(j,soil) ;
Ecoeff_slp_b_sqr(j,soil)  = E_coeff_slp_b_sqr.L(j,soil)  ;
Ecoeff_slp_c_sqr(j,soil)  = E_coeff_slp_c_sqr.L(j,soil)  ;
*Erosion coefficients - fallow
Ecoeff_slp_b_flw_lin(flw,soil)  = E_coeff_slp_b_flw_lin.L(flw,soil)  ;
Ecoeff_slp_c_flw_lin(flw,soil)  = E_coeff_slp_c_flw_lin.L(flw,soil)  ;
Ecoeff_slp_b_flw_sqr(flw,soil)  = E_coeff_slp_b_flw_sqr.L(flw,soil)  ;
Ecoeff_slp_c_flw_sqr(flw,soil)  = E_coeff_slp_c_flw_sqr.L(flw,soil)  ;
Ecoeff_slp_b_flw_exp(flw,soil)  = E_coeff_slp_b_flw_exp.L(flw,soil)  ;
Ecoeff_slp_c_flw_exp(flw,soil)  = E_coeff_slp_c_flw_exp.L(flw,soil)  ;
Ecoeff_slp_b_flw_exp2(flw,soil) = E_coeff_slp_b_flw_exp2.L(flw,soil) ;
Ecoeff_slp_c_flw_exp2(flw,soil) = E_coeff_slp_c_flw_exp2.L(flw,soil) ;

*Choose according to the best functional form (exp2)  NOTE! This specifies the functional form to be used
in all the model files
```

```
Ecoeff_slp_b(j,soil) = Ecoeff_slp_b_exp2(j,soil) ;
Ecoeff_slp_c(j,soil) = Ecoeff_slp_c_exp2(j,soil) ;
Ecoeff_slp_b_flw(flw,soil) = E_coeff_slp_b_flw_exp2.L(flw,soil)  ;
Ecoeff_slp_c_flw(flw,soil) = E_coeff_slp_c_flw_exp2.L(flw,soil)  ;

*Check that none of estimated coefficients do not produce negative load values
NegTPloads(j,slp,soil,pac,pfert)$(TPL_soilP(j,slp,soil,pac,pfert) \textless 0) = 1 ;
sumNegTPloads = SUM{(j,slp,soil,pac,pfert), NegTPloads(j,slp,soil,pac,pfert) } ;

*Nitrogen load function coefficients
Ncoeff_frt_b(j,soil,slp)   = N_coeff_frt_b.L(j,soil,slp);
Ncoeff_frt_c(j,soil,slp)   = N_coeff_frt_c.L(j,soil,slp) ;
Ncoeff_slp_b(j,soil,nfert) = N_coeff_slp_b.L(j,soil,nfert) ;
Ncoeff_slp_c(j,soil,nfert) = N_coeff_slp_c.L(j,soil,nfert) ;

*Fallow results to runoff and P load parameters
Pcoeff_slp_b_flw(flw,soil,pac,pfert) = P_coeff_slp_b_flw.L(flw,soil,pac,pfert) ;
Pcoeff_slp_c_flw(flw,soil,pac,pfert) = P_coeff_slp_c_flw.L(flw,soil,pac,pfert) ;
Rcoeff_slp_b_flw(flw,soil) = R_coeff_slp_b_flw.L(flw,soil)  ;
Rcoeff_slp_c_flw(flw,soil) = R_coeff_slp_c_flw.L(flw,soil)  ;
*total P  (for comparison)
Pcoeff_soilP_a_flw(flw,soil,slp,pfert) = P_coeff_soilP_a_flw.L(flw,soil,slp,pfert)  ;
Pcoeff_soilP_b_flw(flw,soil,slp,pfert) = P_coeff_soilP_b_flw.L(flw,soil,slp,pfert)  ;
*N for fallow
Ncoeff_slp_b_flw(flw,soil,nfert) = N_coeff_slp_b_flw.L(flw,soil,nfert) ;
Ncoeff_slp_c_flw(flw,soil,nfert) = N_coeff_slp_c_flw.L(flw,soil,nfert) ;
*CODE FOR DRAWING FIGURES-----------------------------------------------------

*Set declaration for figure purposes
set

lines1
/ line1*line6 /


lines3
/ ds_data, ds_pred, cc_data, cc_pred, cu_data, cu_pred /

datvspred
/ data, linear, square, exponent /
lines
/ brl_cc_finesand, brl_cc_fineclay, brl_cc_coursesand, brl_cc_courseclay, brl_cu_finesand, brl_cu_fineclay,
brl_cu_coursesand, brl_cu_courseclay, brl_ds_finesand, brl_ds_fineclay, brl_ds_coursesand,
brl_ds_courseclay
/
linesallcrop
/
brl_cc_finesand, brl_cc_fineclay, brl_cc_coursesand, brl_cc_courseclay, brl_cu_finesand, brl_cu_fineclay,
brl_cu_coursesand, brl_cu_courseclay, brl_ds_finesand, brl_ds_fineclay, brl_ds_coursesand,
brl_ds_courseclay
```

wwh_cc_finesand, wwh_cc_fineclay, wwh_cc_coursesand, wwh_cc_courseclay, wwh_cu_finesand, wwh_cu_fineclay, wwh_cu_coursesand, wwh_cu_courseclay, wwh_ds_finesand, wwh_ds_fineclay, wwh_ds_coursesand, wwh_ds_courseclay
sbt_cc_finesand, sbt_cc_fineclay, sbt_cc_coursesand, sbt_cc_courseclay, sbt_cu_finesand, sbt_cu_fineclay, sbt_cu_coursesand, sbt_cu_courseclay, sbt_ds_finesand, sbt_ds_fineclay, sbt_ds_coursesand, sbt_ds_courseclay
ols_cc_finesand, ols_cc_fineclay, ols_cc_coursesand, ols_cc_courseclay, ols_cu_finesand, ols_cu_fineclay, ols_cu_coursesand, ols_cu_courseclay, ols_ds_finesand, ols_ds_fineclay, ols_ds_coursesand, ols_ds_courseclay
grs_cc_finesand, grs_cc_fineclay, grs_cc_coursesand, grs_cc_courseclay, grs_cu_finesand, grs_cu_fineclay, grs_cu_coursesand, grs_cu_courseclay, grs_ds_finesand, grs_ds_fineclay, grs_ds_coursesand, grs_ds_courseclay
pot_cc_finesand, pot_cc_fineclay, pot_cc_coursesand, pot_cc_courseclay, pot_cu_finesand, pot_cu_fineclay, pot_cu_coursesand, pot_cu_courseclay, pot_ds_finesand, pot_ds_fineclay, pot_ds_coursesand, pot_ds_courseclay
/
linesallcrop2
/
brl_cc_fs, brl_cc_fc, brl_cc_cs, brl_cc_cl, brl_cu_fs, brl_cu_fc, brl_cu_cs, brl_cu_cl, brl_ds_fs, brl_ds_fc, brl_ds_cs, brl_ds_cl
wwh_cc_fs, wwh_cc_fc, wwh_cc_cs, wwh_cc_cl, wwh_cu_fs, wwh_cu_fc, wwh_cu_cs, wwh_cu_cl, wwh_ds_fs, wwh_ds_fc, wwh_ds_cs, wwh_ds_cl
sbt_cc_fs, sbt_cc_fc, sbt_cc_cs, sbt_cc_cl, sbt_cu_fs, sbt_cu_fc, sbt_cu_cs, sbt_cu_cl, sbt_ds_fs, sbt_ds_fc, sbt_ds_cs, sbt_ds_cl
ols_cc_fs, ols_cc_fc, ols_cc_cs, ols_cc_cl, ols_cu_fs, ols_cu_fc, ols_cu_cs, ols_cu_cl, ols_ds_fs, ols_ds_fc, ols_ds_cs, ols_ds_cl
grs_cc_fs, grs_cc_fc, grs_cc_cs, grs_cc_cl, grs_cu_fs, grs_cu_fc, grs_cu_cs, grs_cu_cl, grs_ds_fs, grs_ds_fc, grs_ds_cs, grs_ds_cl
pot_cc_fs, pot_cc_fc, pot_cc_cs, pot_cc_cl, pot_cu_fs, pot_cu_fc, pot_cu_cs, pot_cu_cl, pot_ds_fs, pot_ds_fc, pot_ds_cs, pot_ds_cl
/
singleCrop
/
pred_cc_fs, pred_cc_fc, pred_cc_cs, pred_cc_cl, pred_cu_fs, pred_cu_fc, pred_cu_cs, pred_cu_cl, pred_ds_fs, pred_ds_fc, pred_ds_cs, pred_ds_cl
IC_cc_fs, IC_cc_fc, IC_cc_cs, IC_cc_cl, IC_cu_fs, IC_cu_fc, IC_cu_cs, IC_cu_cl, IC_ds_fs, IC_ds_fc, IC_ds_cs, IC_ds_cl
/
predLines
/
pred_cc_fs, pred_cc_fc, pred_cc_cs, pred_cc_cl, pred_cu_fs, pred_cu_fc, pred_cu_cs, pred_cu_cl, pred_ds_fs, pred_ds_fc, pred_ds_cs, pred_ds_cl
/
ICreamLines
/
IC_cc_fs, IC_cc_fc, IC_cc_cs, IC_cc_cl, IC_cu_fs, IC_cu_fc, IC_cu_cs, IC_cu_cl, IC_ds_fs, IC_ds_fc, IC_ds_cs, IC_ds_cl
/
ICreamLinesN
/
IC_cc_fs_s0, IC_cc_fc_s0, IC_cc_cs_s0, IC_cc_cl_s0, IC_cu_fs_s0, IC_cu_fc_s0, IC_cu_cs_s0, IC_cu_cl_s0, IC_ds_fs_s0, IC_ds_fc_s0, IC_ds_cs_s0, IC_ds_cl_s0

IC_cc_fs_s1, IC_cc_fc_s1, IC_cc_cs_s1, IC_cc_cl_s1, IC_cu_fs_s1, IC_cu_fc_s1, IC_cu_cs_s1, IC_cu_cl_s1,
IC_ds_fs_s1, IC_ds_fc_s1, IC_ds_cs_s1, IC_ds_cl_s1
IC_cc_fs_s3, IC_cc_fc_s3, IC_cc_cs_s3, IC_cc_cl_s3, IC_cu_fs_s3, IC_cu_fc_s3, IC_cu_cs_s3, IC_cu_cl_s3,
IC_ds_fs_s3, IC_ds_fc_s3, IC_ds_cs_s3, IC_ds_cl_s3
IC_cc_fs_s7, IC_cc_fc_s7, IC_cc_cs_s7, IC_cc_cl_s7, IC_cu_fs_s7, IC_cu_fc_s7, IC_cu_cs_s7, IC_cu_cl_s7,
IC_ds_fs_s7, IC_ds_fc_s7, IC_ds_cs_s7, IC_ds_cl_s7
/
subeet_line
/sbt_cc_finesand, sbt_cc_fineclay, sbt_cc_coursesand, sbt_cc_courseclay, sbt_cu_finesand,
sbt_cu_fineclay, sbt_cu_coursesand, sbt_cu_courseclay, sbt_ds_finesand, sbt_ds_fineclay,
sbt_ds_coursesand, sbt_ds_courseclay /
grass_line
/ grs_cc_finesand, grs_cc_fineclay, grs_cc_coursesand, grs_cc_courseclay, grs_cu_finesand,
grs_cu_fineclay, grs_cu_coursesand, grs_cu_courseclay, grs_ds_finesand, grs_ds_fineclay,
grs_ds_coursesand, grs_ds_courseclay /
convcult
/
brl_cc_finesand, brl_cc_fineclay, brl_cc_coursesand, brl_cc_courseclay
wwh_cc_finesand, wwh_cc_fineclay, wwh_cc_coursesand, wwh_cc_courseclay
sbt_cc_finesand, sbt_cc_fineclay, sbt_cc_coursesand, sbt_cc_courseclay
ols_cc_finesand, ols_cc_fineclay, ols_cc_coursesand, ols_cc_courseclay
grs_cc_finesand, grs_cc_fineclay, grs_cc_coursesand, grs_cc_courseclay
pot_cc_finesand, pot_cc_fineclay, pot_cc_coursesand, pot_cc_courseclay
/
convcultHtS
/ brl_cc_courseclay, wwh_cc_courseclay, sbt_cc_courseclay, ols_cc_courseclay, grs_cc_courseclay,
pot_cc_courseclay /
plines
/
brl_cc_finesand_lin, brl_cc_fineclay_lin, brl_cc_coursesand_lin, brl_cc_courseclay_lin, brl_cu_finesand_lin,
brl_cu_fineclay_lin, brl_cu_coursesand_lin, brl_cu_courseclay_lin, brl_ds_finesand_lin, brl_ds_fineclay_lin,
brl_ds_coursesand_lin, brl_ds_courseclay_lin
brl_cc_finesand_exp, brl_cc_fineclay_exp, brl_cc_coursesand_exp, brl_cc_courseclay_exp,
brl_cu_finesand_exp, brl_cu_fineclay_exp, brl_cu_coursesand_exp, brl_cu_courseclay_exp,
brl_ds_finesand_exp, brl_ds_fineclay_exp, brl_ds_coursesand_exp, brl_ds_courseclay_exp
brl_cc_finesand_sqr, brl_cc_fineclay_sqr, brl_cc_coursesand_sqr, brl_cc_courseclay_sqr,
brl_cu_finesand_sqr, brl_cu_fineclay_sqr, brl_cu_coursesand_sqr, brl_cu_courseclay_sqr,
brl_ds_finesand_sqr, brl_ds_fineclay_sqr, brl_ds_coursesand_sqr, brl_ds_courseclay_sqr
/

scen Scenarios of which each is a line
/
brl_cc_slt1, brl_cc_slt4, brl_cc_slt9, brl_cc_slt16, brl_cu_slt1, brl_cu_slt4, brl_cu_slt9, brl_cu_slt16,
brl_ds_slt1, brl_ds_slt4, brl_ds_slt9, brl_ds_slt16
cultivation_data, cultivation_pred, ds_data, ds_pred, cc_data, cu_data, cu_pred, cc_pred,
ds_data1, ds_pred1, cc_data1, cu_data1, cu_pred1, cc_pred1, ds_data2, ds_pred2, cc_data2, cu_data2,
cu_pred2, cc_pred2, ds_data3, ds_pred3, cc_data3, cu_data3, cu_pred3, cc_pred3
/
bscen
/
predicted_cc_finesand, predicted_cc_fineclay, predicted_cc_coursesand, predicted_cc_courseclay
predicted_cu_finesand, predicted_cu_fineclay, predicted_cu_coursesand, predicted_cu_courseclay

```
predicted_ds_finesand, predicted_ds_fineclay, predicted_ds_coursesand, predicted_ds_courseclay
data_cc_finesand, data_cc_fineclay, data_cc_coursesand, data_cc_courseclay
data_cu_finesand, data_cu_fineclay, data_cu_coursesand, data_cu_courseclay
data_ds_finesand, data_ds_fineclay, data_ds_coursesand, data_ds_courseclay
/
bscen2
/
predicted_cc_finesand, predicted_cc_fineclay, predicted_cc_coursesand, predicted_cc_courseclay
data_cc_finesand, data_cc_fineclay, data_cc_coursesand, data_cc_courseclay
/

bICdata
/
cc_finesand, cc_fineclay, cc_coursesand, cc_courseclay
cu_finesand, cu_fineclay, cu_coursesand, cu_courseclay
ds_finesand, ds_fineclay, ds_coursesand, ds_courseclay
/

item        y and x axis items       /Pred, CDataP, slopepoints, totP, soilP, P_level, runoff, erosion, totalN,
Nfert, Pfert, slope /
Naxisitems                      / N_fertilization, ICdata, totalNload   /
;

*Parameters for different graphs (need to be 3 dimensional)
parameters
NloadBrlSlp1HtS(lines3, nfert, item)          N load as a function N fert for HtS soil slope class 1
NloadBrlSlp3HtS(lines3, nfert, item)          N load as a function N fert for HtS soil slope class 3
NloadBrlSlp3HsS(lines3, nfert, item)          N load as a function N fert for HsS soil slope class 3
NloadBrlSlp3KHt(lines1, nfert, Naxisitems)      N load as a function N fert for KHt soil slope class 3
N_dataGraphBrl_slp1
P_12_HsS_15F_barley(scen, slp, item)          The graph of TP load of barley as P stock is 12 and P
fertilisation is 15
P_12_HsS_barley(scen, pfert, item)            The graph of TP load of barley as P stock is 12 and P fertilisation
varies
runoff_12_HsS_15F_barley(scen, slp, item)       The graph of runoff of barley as P stock is 12 and P
fertilisation is 15
runoff_barley_cc(bscen2, slp, item)           Runoff data and OLS predicted values for barley (conventional
cultivation)
runoff_barley(singleCrop, slp, item)          Runoff data and OLS predicted values for barley (all cultivation)
NloadBrlSlp3(bICdata, nfert, item)            Barley N load for slope class3
N_dataGraphBrl_slp1(bICdata, nfert, item)       N load as function of fertilization  (IC data)  for SLP1
N_barley_slp_data(bscen, slp, item)           N load of barley as function of slope when N fertilisation is 80
(data)
N_barley_data(ICreamLinesN, nfert, item)        N load of barley as function of fertilization  (IC data)
N_dataGraph_slp1(linesallcrop, nfert, item)      Nitrogen load plotted for fert levels
N_data_CC_SLP1(convcult, nfert, item)          Nitrogen load of conventionally cultivated crops plotted for
different fertilization levels
N_load_HtS_CC_SLP1(convcultHtS, nfert, item)     Nitrogen load of conventionally cultivated crops plotted
for different fertilization levels
N_subeet_slp1(subeet_line, nfert, item)        Nitrogen load for sugarbeet plotted for fert levels
N_grass_slp1(grass_line, nfert, item)          Nitrogen load for sugarbeet plotted for fert levels
```

erosion_12_HsS_15F_barley(scen, slp, item)   The graph of NON-SCALED erosion of barley as P stock is 12 and P fertilisation is 15
erosion_all_crops(linesallcrop, slp, item)   The rescaled erosion data
erosion_all_crops2(linesallcrop2, slp, item)   The rescaled erosion data
erosion_predicted(plines, slp, item)   Preditected erosion using OLS estimated functions
ErosionBrl_cc_HtS(datvspred, slp, item)   IC-data and preditected erosion using OLS estimated functions for barley cc Hts
ErosionBrl_cu_HtS(datvspred, slp, item)   IC-data and preditected erosion using OLS estimated functions for barley cu Hts
ErosionBrl_ds_HtS(datvspred, slp, item)   IC-data and preditected erosion using OLS estimated functions for barley ds Hts
ErosionGrs_cc_HtS(datvspred, slp, item)   IC-data and preditected erosion using OLS estimated functions for grass cc Hts
erosion_barley(bscen, slp, item)   Barley average erosion as function of slope
erosion_ICbarley(bICdata, slp, item)   ICECREMA-data for barley
totPfert_barley(bscen, pfert, item)   Total P-loas as a function of P fertilisation for barley (all tillage and soil and slope) OLS and data
;

```
*GRAPH DATA VALUES
*X-axis
totPfert_barley(bscen, pfert, "Pfert") = fertP('f13','slp1','HHt','pac12',pfert)                 ;
*Y-axis
totPfert_barley("data_cc_finesand", pfert, "totP") = TP_load("f13",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("data_cc_fineclay", pfert, "totP") = TP_load("f13",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("data_cc_coursesand", pfert, "totP") = TP_load("f13",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("data_cc_courseclay", pfert, "totP") = TP_load("f13",'slp1','HtS','pac12',pfert)   ;
totPfert_barley("predicted_cc_finesand", pfert, "totP") = P_load_fert("f13",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("predicted_cc_fineclay", pfert, "totP") = P_load_fert("f13",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("predicted_cc_coursesand", pfert, "totP") = P_load_fert("f13",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("predicted_cc_courseclay", pfert, "totP") = P_load_fert("f13",'slp1','HtS','pac12',pfert)   ;

totPfert_barley("data_cu_finesand", pfert, "totP") = TP_load("f14",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("data_cu_fineclay", pfert, "totP") = TP_load("f14",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("data_cu_coursesand", pfert, "totP") = TP_load("f14",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("data_cu_courseclay", pfert, "totP") = TP_load("f14",'slp1','HtS','pac12',pfert)   ;
totPfert_barley("predicted_cu_finesand", pfert, "totP") = P_load_fert("f14",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("predicted_cu_fineclay", pfert, "totP") = P_load_fert("f14",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("predicted_cu_coursesand", pfert, "totP") = P_load_fert("f14",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("predicted_cu_courseclay", pfert, "totP") = P_load_fert("f14",'slp1','HtS','pac12',pfert)   ;

totPfert_barley("data_ds_finesand", pfert, "totP") = TP_load("f15",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("data_ds_fineclay", pfert, "totP") = TP_load("f15",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("data_ds_coursesand", pfert, "totP") = TP_load("f15",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("data_ds_courseclay", pfert, "totP") = TP_load("f15",'slp1','HtS','pac12',pfert)   ;
totPfert_barley("predicted_ds_finesand", pfert, "totP") = P_load_fert("f15",'slp1','HHt','pac12',pfert)    ;
totPfert_barley("predicted_ds_fineclay", pfert, "totP") = P_load_fert("f15",'slp1','HsS','pac12',pfert)    ;
totPfert_barley("predicted_ds_coursesand", pfert, "totP") = P_load_fert("f15",'slp1','KHt','pac12',pfert)   ;
totPfert_barley("predicted_ds_courseclay", pfert, "totP") = P_load_fert("f15",'slp1','HtS','pac12',pfert)   ;

*X-axis
NloadBrlSlp3(bICdata, nfert, "Nfert")  =  N_fert('f34','slp1','HHt',nfert)  ;
```

```
*Y-axis
NloadBrlSlp3("cc_finesand", nfert, "totalN") = Nloaddata("f13","slp3",'HHt',nfert)     ;
NloadBrlSlp3("cc_fineclay", nfert, "totalN") = Nloaddata("f13","slp3",'HsS',nfert)     ;
NloadBrlSlp3("cc_coursesand", nfert, "totalN") = Nloaddata("f13","slp3",'KHt',nfert)   ;
NloadBrlSlp3("cc_courseclay", nfert, "totalN") = Nloaddata("f13","slp3",'HtS',nfert)   ;
NloadBrlSlp3("cu_finesand", nfert, "totalN") = Nloaddata("f14","slp3",'HHt',nfert)     ;
NloadBrlSlp3("cu_fineclay", nfert, "totalN") = Nloaddata("f14","slp3",'HsS',nfert)     ;
NloadBrlSlp3("cu_coursesand", nfert, "totalN") = Nloaddata("f14","slp3",'KHt',nfert)   ;
NloadBrlSlp3("cu_courseclay", nfert, "totalN") = Nloaddata("f14","slp3",'HtS',nfert)   ;
NloadBrlSlp3("ds_finesand", nfert, "totalN") = Nloaddata("f15","slp3",'HHt',nfert)     ;
NloadBrlSlp3("ds_fineclay", nfert, "totalN") = Nloaddata("f15","slp3",'HsS',nfert)     ;
NloadBrlSlp3("ds_coursesand", nfert, "totalN") = Nloaddata("f15","slp3",'KHt',nfert)   ;
NloadBrlSlp3("ds_courseclay", nfert, "totalN") = Nloaddata("f15","slp3",'HtS',nfert)   ;


erosion_ICbarley(bICdata, slp, "slope") = slopepoint(slp)                  ;
erosion_ICbarley("cc_finesand", slp, "erosion") = erosiondata("f13",slp,'HHt')     ;
erosion_ICbarley("cc_fineclay", slp, "erosion") = erosiondata("f13",slp,'HsS')     ;
erosion_ICbarley("cc_coursesand", slp, "erosion") = erosiondata("f13",slp,'KHt')   ;
erosion_ICbarley("cc_courseclay", slp, "erosion") = erosiondata("f13",slp,'HtS')   ;
erosion_ICbarley("cu_finesand", slp, "erosion") = erosiondata("f14",slp,'HHt')     ;
erosion_ICbarley("cu_fineclay", slp, "erosion") = erosiondata("f14",slp,'HsS')     ;
erosion_ICbarley("cu_coursesand", slp, "erosion") = erosiondata("f14",slp,'KHt')   ;
erosion_ICbarley("cu_courseclay", slp, "erosion") = erosiondata("f14",slp,'HtS')   ;
erosion_ICbarley("ds_finesand", slp, "erosion") = erosiondata("f15",slp,'HHt')     ;
erosion_ICbarley("ds_fineclay", slp, "erosion") = erosiondata("f15",slp,'HsS')     ;
erosion_ICbarley("ds_coursesand", slp, "erosion") = erosiondata("f15",slp,'KHt')   ;
erosion_ICbarley("ds_courseclay", slp, "erosion") = erosiondata("f15",slp,'HtS')   ;

*EROSION FOR BARLEY DATA VS OLS MODEL (TILLAGE AND SOIL)
erosion_barley(bscen, slp, "slope") = slopepoint(slp)                  ;
erosion_barley("data_cc_finesand", slp, "erosion") = erosiondata("f13",slp,'HHt')     ;
erosion_barley("data_cc_fineclay", slp, "erosion") = erosiondata("f13",slp,'HsS')     ;
erosion_barley("data_cc_coursesand", slp, "erosion") = erosiondata("f13",slp,'KHt')   ;
erosion_barley("data_cc_courseclay", slp, "erosion") = erosiondata("f13",slp,'HtS')   ;
erosion_barley("predicted_cc_finesand", slp, "erosion") = Perosion_exp("f13",slp,'HHt')     ;
erosion_barley("predicted_cc_fineclay", slp, "erosion") = Perosion_exp("f13",slp,'HsS')     ;
erosion_barley("predicted_cc_coursesand", slp, "erosion") = Perosion_exp("f13",slp,'KHt')   ;
erosion_barley("predicted_cc_courseclay", slp, "erosion") = Perosion_exp("f13",slp,'HtS')   ;

erosion_barley("data_cu_finesand", slp, "erosion") = erosiondata("f14",slp,'HHt')     ;
erosion_barley("data_cu_fineclay", slp, "erosion") = erosiondata("f14",slp,'HsS')     ;
erosion_barley("data_cu_coursesand", slp, "erosion") = erosiondata("f14",slp,'KHt')   ;
erosion_barley("data_cu_courseclay", slp, "erosion") = erosiondata("f14",slp,'HtS')   ;
erosion_barley("predicted_cu_finesand", slp, "erosion") = Perosion_exp("f14",slp,'HHt')     ;
erosion_barley("predicted_cu_fineclay", slp, "erosion") = Perosion_exp("f14",slp,'HsS')     ;
erosion_barley("predicted_cu_coursesand", slp, "erosion") = Perosion_exp("f14",slp,'KHt')   ;
erosion_barley("predicted_cu_courseclay", slp, "erosion") = Perosion_exp("f14",slp,'HtS')   ;

erosion_barley("data_ds_finesand", slp, "erosion") = erosiondata("f15",slp,'HHt')     ;
erosion_barley("data_ds_fineclay", slp, "erosion") = erosiondata("f15",slp,'HsS')     ;
erosion_barley("data_ds_coursesand", slp, "erosion") = erosiondata("f15",slp,'KHt')   ;
```

```
erosion_barley("data_ds_courseclay", slp, "erosion") = erosiondata("f15",slp,'HtS')   ;
erosion_barley("predicted_ds_finesand", slp, "erosion") = Perosion_exp("f15",slp,'HHt')     ;
erosion_barley("predicted_ds_fineclay", slp, "erosion") = Perosion_exp("f15",slp,'HsS')     ;
erosion_barley("predicted_ds_coursesand", slp, "erosion") = Perosion_exp("f15",slp,'KHt')   ;
erosion_barley("predicted_ds_courseclay", slp, "erosion") = Perosion_exp("f15",slp,'HtS')   ;

*X-AXIS
ErosionBrl_cc_HtS(datvspred, slp, 'slope') = slopedata(slp)                    ;
*Y-AXIS
ErosionBrl_cc_HtS('data', slp, 'erosion')     =    Erosiondata('f13',slp,'HtS')    ;
ErosionBrl_cc_HtS('linear', slp, 'erosion')   =    Perosion_lin('f13',slp,'HtS')   ;
ErosionBrl_cc_HtS('square', slp, 'erosion')   =    Perosion_sqr('f13',slp,'HtS')   ;
ErosionBrl_cc_HtS('exponent', slp, 'erosion') =    Perosion_exp('f13',slp,'HtS')   ;

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley erosion conventionally cultivated IC-data vs predictions'
*$libinclude gnuplotxyz ErosionBrl_cc_HtS slope erosion

*X-AXIS
ErosionBrl_cu_HtS(datvspred, slp, 'slope') = slopedata(slp)                    ;
*Y-AXIS
ErosionBrl_cu_HtS('data', slp, 'erosion')     =    Erosiondata('f14',slp,'HtS')    ;
ErosionBrl_cu_HtS('linear', slp, 'erosion')   =    Perosion_lin('f14',slp,'HtS')   ;
ErosionBrl_cu_HtS('square', slp, 'erosion')   =    Perosion_sqr('f14',slp,'HtS')   ;
ErosionBrl_cu_HtS('exponent', slp, 'erosion') =    Perosion_exp('f14',slp,'HtS')   ;

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley erosion cultivated IC-data vs predictions'
*$libinclude gnuplotxyz ErosionBrl_cu_HtS slope erosion

*X-AXIS
ErosionBrl_ds_HtS(datvspred, slp, 'slope') = slopedata(slp)                    ;
*Y-AXIS
ErosionBrl_ds_HtS('data', slp, 'erosion')     =    Erosiondata('f15',slp,'HtS')    ;
ErosionBrl_ds_HtS('linear', slp, 'erosion')   =    Perosion_lin('f15',slp,'HtS')   ;
ErosionBrl_ds_HtS('square', slp, 'erosion')   =    Perosion_sqr('f15',slp,'HtS')   ;
ErosionBrl_ds_HtS('exponent', slp, 'erosion') =    Perosion_exp('f15',slp,'HtS')   ;

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley erosion direct till IC-data vs predictions'
*$libinclude gnuplotxyz ErosionBrl_ds_HtS slope erosion

*X-AXIS
ErosionGrs_cc_HtS(datvspred, slp, 'slope') = slopedata(slp)                    ;
*Y-AXIS
ErosionGrs_cc_HtS('data', slp, 'erosion')     =    Erosiondata('f43',slp,'HtS')    ;
ErosionGrs_cc_HtS('linear', slp, 'erosion')   =    Perosion_lin('f43',slp,'HtS')   ;
ErosionGrs_cc_HtS('square', slp, 'erosion')   =    Perosion_sqr('f43',slp,'HtS')   ;
ErosionGrs_cc_HtS('exponent', slp, 'erosion') =    Perosion_exp('f43',slp,'HtS')   ;

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Grass erosion conventionally cultivated IC-data vs predictions'
```

```
*$libinclude gnuplotxyz ErosionGrs_cc_HtS slope erosion

*erosion_all_crops_soils(scen, slp, item) As function of slope
*EROSION DATA
*Conventional tillage
erosion_all_crops(linesallcrop, slp, 'slopepoints') = slopedata(slp)    ;
erosion_all_crops('brl_cc_finesand', slp, 'erosion') = erosiondata('f13',slp,'HHt')  ;
erosion_all_crops('brl_cc_fineclay', slp, 'erosion') = erosiondata("f13",slp,"HsS")  ;
erosion_all_crops('brl_cc_coursesand', slp, 'erosion') = erosiondata("f13",slp,"KHt")  ;
erosion_all_crops('brl_cc_courseclay', slp, 'erosion') = erosiondata("f13",slp,"HtS")  ;
*Cultivation on different soils
erosion_all_crops('brl_cu_finesand', slp, 'erosion') = erosiondata("f14",slp,"HHt")  ;
erosion_all_crops('brl_cu_fineclay', slp, 'erosion') = erosiondata("f14",slp,"HsS")  ;
erosion_all_crops('brl_cu_coursesand', slp, 'erosion') = erosiondata("f14",slp,"KHt")  ;
erosion_all_crops('brl_cu_courseclay', slp, 'erosion') = erosiondata("f14",slp,"HtS")  ;
*Direct sowing on different soils
erosion_all_crops('brl_ds_finesand', slp, 'erosion') = erosiondata("f15",slp,"HHt")  ;
erosion_all_crops('brl_ds_fineclay', slp, 'erosion') = erosiondata("f15",slp,"HsS")  ;
erosion_all_crops('brl_ds_coursesand', slp, 'erosion') = erosiondata("f15",slp,"KHt")  ;
erosion_all_crops('brl_ds_courseclay', slp, 'erosion') = erosiondata("f15",slp,"HtS")  ;
*Conventional tillage
erosion_all_crops('wwh_cc_finesand', slp, 'erosion') = erosiondata('f1',slp,'HHt')  ;
erosion_all_crops('wwh_cc_fineclay', slp, 'erosion') = erosiondata("f1",slp,"HsS")  ;
erosion_all_crops('wwh_cc_coursesand', slp, 'erosion') = erosiondata("f1",slp,"KHt")  ;
erosion_all_crops('wwh_cc_courseclay', slp, 'erosion') = erosiondata("f1",slp,"HtS")  ;
*Cultivation on different soils
erosion_all_crops('wwh_cu_finesand', slp, 'erosion') = erosiondata("f2",slp,"HHt")  ;
erosion_all_crops('wwh_cu_fineclay', slp, 'erosion') = erosiondata("f2",slp,"HsS")  ;
erosion_all_crops('wwh_cu_coursesand', slp, 'erosion') = erosiondata("f2",slp,"KHt")  ;
erosion_all_crops('wwh_cu_courseclay', slp, 'erosion') = erosiondata("f2",slp,"HtS")  ;
*Direct sowing on different soils
erosion_all_crops('wwh_ds_finesand', slp, 'erosion') = erosiondata("f3",slp,"HHt")  ;
erosion_all_crops('wwh_ds_fineclay', slp, 'erosion') = erosiondata("f3",slp,"HsS")  ;
erosion_all_crops('wwh_ds_coursesand', slp, 'erosion') = erosiondata("f3",slp,"KHt")  ;
erosion_all_crops('wwh_ds_courseclay', slp, 'erosion') = erosiondata("f3",slp,"HtS")  ;
*Grass
erosion_all_crops('grs_cc_finesand', slp, 'erosion') = erosiondata('f43',slp,'HHt')  ;
erosion_all_crops('grs_cc_fineclay', slp, 'erosion') = erosiondata("f43",slp,"HsS")  ;
erosion_all_crops('grs_cc_coursesand', slp, 'erosion') = erosiondata("f43",slp,"KHt")  ;
erosion_all_crops('grs_cc_courseclay', slp, 'erosion') = erosiondata("f43",slp,"HtS")  ;
*Sugarbeet
erosion_all_crops('sbt_cc_finesand', slp, 'erosion') = erosiondata('f34',slp,'HHt')  ;
erosion_all_crops('sbt_cc_fineclay', slp, 'erosion') = erosiondata("f34",slp,"HsS")  ;
erosion_all_crops('sbt_cc_coursesand', slp, 'erosion') = erosiondata("f34",slp,"KHt")  ;
erosion_all_crops('sbt_cc_courseclay', slp, 'erosion') = erosiondata("f34",slp,"HtS")  ;
*Oilseed
erosion_all_crops('ols_cc_finesand', slp, 'erosion') = erosiondata('f37',slp,'HHt')  ;
erosion_all_crops('ols_cc_fineclay', slp, 'erosion') = erosiondata("f37",slp,"HsS")  ;
erosion_all_crops('ols_cc_coursesand', slp, 'erosion') = erosiondata("f37",slp,"KHt")  ;
erosion_all_crops('ols_cc_courseclay', slp, 'erosion') = erosiondata("f37",slp,"HtS")  ;
*Potato
erosion_all_crops('pot_cc_finesand', slp, 'erosion') = erosiondata('f28',slp,'HHt')  ;
```

```
erosion_all_crops('pot_cc_fineclay', slp, 'erosion') = erosiondata("f28",slp,"HsS")  ;
erosion_all_crops('pot_cc_coursesand', slp, 'erosion') = erosiondata("f28",slp,"KHt")  ;
erosion_all_crops('pot_cc_courseclay', slp, 'erosion') = erosiondata("f28",slp,"HtS")  ;

*EROSION DATA2
*Conventional tillage
erosion_all_crops2(linesallcrop2, slp, 'slopepoints') = slopedata(slp)    ;
erosion_all_crops2('brl_cc_fs', slp, 'erosion') = erosiondata('f13',slp,'HHt')  ;
erosion_all_crops2('brl_cc_fc', slp, 'erosion') = erosiondata("f13",slp,"HsS")  ;
erosion_all_crops2('brl_cc_cs', slp, 'erosion') = erosiondata("f13",slp,"KHt")  ;
erosion_all_crops2('brl_cc_cl', slp, 'erosion') = erosiondata("f13",slp,"HtS")  ;
*Cultivation on different soils
erosion_all_crops2('brl_cu_fs', slp, 'erosion') = erosiondata("f14",slp,"HHt")  ;
erosion_all_crops2('brl_cu_fc', slp, 'erosion') = erosiondata("f14",slp,"HsS")  ;
erosion_all_crops2('brl_cu_cs', slp, 'erosion') = erosiondata("f14",slp,"KHt")  ;
erosion_all_crops2('brl_cu_cl', slp, 'erosion') = erosiondata("f14",slp,"HtS")  ;
*Direct sowing on different soils
erosion_all_crops2('brl_ds_fs', slp, 'erosion') = erosiondata("f15",slp,"HHt")  ;
erosion_all_crops2('brl_ds_fc', slp, 'erosion') = erosiondata("f15",slp,"HsS")  ;
erosion_all_crops2('brl_ds_cs', slp, 'erosion') = erosiondata("f15",slp,"KHt")  ;
erosion_all_crops2('brl_ds_cl', slp, 'erosion') = erosiondata("f15",slp,"HtS")  ;
*Conventional tillage
erosion_all_crops2('wwh_cc_fs', slp, 'erosion') = erosiondata('f1',slp,'HHt')  ;
erosion_all_crops2('wwh_cc_fc', slp, 'erosion') = erosiondata("f1",slp,"HsS")  ;
erosion_all_crops2('wwh_cc_cs', slp, 'erosion') = erosiondata("f1",slp,"KHt")  ;
erosion_all_crops2('wwh_cc_cl', slp, 'erosion') = erosiondata("f1",slp,"HtS")  ;
*Cultivation on different soils
erosion_all_crops2('wwh_cu_fs', slp, 'erosion') = erosiondata("f2",slp,"HHt")  ;
erosion_all_crops2('wwh_cu_fc', slp, 'erosion') = erosiondata("f2",slp,"HsS")  ;
erosion_all_crops2('wwh_cu_cs', slp, 'erosion') = erosiondata("f2",slp,"KHt")  ;
erosion_all_crops2('wwh_cu_cl', slp, 'erosion') = erosiondata("f2",slp,"HtS")  ;
*Direct sowing on different soils
erosion_all_crops2('wwh_ds_fs', slp, 'erosion') = erosiondata("f3",slp,"HHt")  ;
erosion_all_crops2('wwh_ds_fc', slp, 'erosion') = erosiondata("f3",slp,"HsS")  ;
erosion_all_crops2('wwh_ds_cs', slp, 'erosion') = erosiondata("f3",slp,"KHt")  ;
erosion_all_crops2('wwh_ds_cl', slp, 'erosion') = erosiondata("f3",slp,"HtS")  ;
*Grass
erosion_all_crops2('grs_cc_fs', slp, 'erosion') = erosiondata('f43',slp,'HHt')  ;
erosion_all_crops2('grs_cc_fc', slp, 'erosion') = erosiondata("f43",slp,"HsS")  ;
erosion_all_crops2('grs_cc_cs', slp, 'erosion') = erosiondata("f43",slp,"KHt")  ;
erosion_all_crops2('grs_cc_cl', slp, 'erosion') = erosiondata("f43",slp,"HtS")  ;
*Sugarbeet
erosion_all_crops2('sbt_cc_fs', slp, 'erosion') = erosiondata('f34',slp,'HHt')  ;
erosion_all_crops2('sbt_cc_fc', slp, 'erosion') = erosiondata("f34",slp,"HsS")  ;
erosion_all_crops2('sbt_cc_cs', slp, 'erosion') = erosiondata("f34",slp,"KHt")  ;
erosion_all_crops2('sbt_cc_cl', slp, 'erosion') = erosiondata("f34",slp,"HtS")  ;
*Oilseed
erosion_all_crops2('ols_cc_fs', slp, 'erosion') = erosiondata('f37',slp,'HHt')  ;
erosion_all_crops2('ols_cc_fc', slp, 'erosion') = erosiondata("f37",slp,"HsS")  ;
erosion_all_crops2('ols_cc_cs', slp, 'erosion') = erosiondata("f37",slp,"KHt")  ;
erosion_all_crops2('ols_cc_cl', slp, 'erosion') = erosiondata("f37",slp,"HtS")  ;
*Potato
```

```
erosion_all_crops2('pot_cc_fs', slp, 'erosion') = erosiondata('f28',slp,'HHt') ;
erosion_all_crops2('pot_cc_fc', slp, 'erosion') = erosiondata("f28",slp,"HsS") ;
erosion_all_crops2('pot_cc_cs', slp, 'erosion') = erosiondata("f28",slp,"KHt") ;
erosion_all_crops2('pot_cc_cl', slp, 'erosion') = erosiondata("f28",slp,"HtS") ;

*EROSION PREDICTIONS
*LINEAR
*Conventional tillage
erosion_predicted(plines, slp, 'slopepoints') = slopedata(slp)    ;
erosion_predicted('brl_cc_finesand_lin', slp, 'erosion') = Perosion_lin('f13',slp,'HHt') ;
erosion_predicted('brl_cc_fineclay_lin', slp, 'erosion') = Perosion_lin("f13",slp,"HsS") ;
erosion_predicted('brl_cc_coursesand_lin', slp, 'erosion') = Perosion_lin("f13",slp,"KHt") ;
erosion_predicted('brl_cc_courseclay_lin', slp, 'erosion') = Perosion_lin("f13",slp,"HtS") ;
*Cultivation on different soils
erosion_predicted('brl_cu_finesand_lin', slp, 'erosion') = Perosion_lin("f14",slp,"HHt") ;
erosion_predicted('brl_cu_fineclay_lin', slp, 'erosion') = Perosion_lin("f14",slp,"HsS") ;
erosion_predicted('brl_cu_coursesand_lin', slp, 'erosion') = Perosion_lin("f14",slp,"KHt") ;
erosion_predicted('brl_cu_courseclay_lin', slp, 'erosion') = Perosion_lin("f14",slp,"HtS") ;
*Direct sowing on different soils
erosion_predicted('brl_ds_finesand_lin', slp, 'erosion') = Perosion_lin("f15",slp,"HHt") ;
erosion_predicted('brl_ds_fineclay_lin', slp, 'erosion') = Perosion_lin("f15",slp,"HsS") ;
erosion_predicted('brl_ds_coursesand_lin', slp, 'erosion') = Perosion_lin("f15",slp,"KHt") ;
erosion_predicted('brl_ds_courseclay_lin', slp, 'erosion') = Perosion_lin("f15",slp,"HtS") ;

*QUADRATIC
*Conventional tillage
erosion_predicted('brl_cc_finesand_sqr', slp, 'erosion') = Perosion_sqr('f13',slp,'HHt') ;
erosion_predicted('brl_cc_fineclay_sqr', slp, 'erosion') = Perosion_sqr("f13",slp,"HsS") ;
erosion_predicted('brl_cc_coursesand_sqr', slp, 'erosion') = Perosion_sqr("f13",slp,"KHt") ;
erosion_predicted('brl_cc_courseclay_sqr', slp, 'erosion') = Perosion_sqr("f13",slp,"HtS") ;
*Cultivation on different soils
erosion_predicted('brl_cu_finesand_sqr', slp, 'erosion') = Perosion_sqr("f14",slp,"HHt") ;
erosion_predicted('brl_cu_fineclay_sqr', slp, 'erosion') = Perosion_sqr("f14",slp,"HsS") ;
erosion_predicted('brl_cu_coursesand_sqr', slp, 'erosion') = Perosion_sqr("f14",slp,"KHt") ;
erosion_predicted('brl_cu_courseclay_sqr', slp, 'erosion') = Perosion_sqr("f14",slp,"HtS") ;
*Direct sowing on different soils
erosion_predicted('brl_ds_finesand_sqr', slp, 'erosion') = Perosion_sqr("f15",slp,"HHt") ;
erosion_predicted('brl_ds_fineclay_sqr', slp, 'erosion') = Perosion_sqr("f15",slp,"HsS") ;
erosion_predicted('brl_ds_coursesand_sqr', slp, 'erosion') = Perosion_sqr("f15",slp,"KHt") ;
erosion_predicted('brl_ds_courseclay_sqr', slp, 'erosion') = Perosion_sqr("f15",slp,"HtS") ;

*EXP
*Conventional tillage
erosion_predicted('brl_cc_finesand_exp', slp, 'erosion') = Perosion_exp('f13',slp,'HHt') ;
erosion_predicted('brl_cc_fineclay_exp', slp, 'erosion') = Perosion_exp("f13",slp,"HsS") ;
erosion_predicted('brl_cc_coursesand_exp', slp, 'erosion') = Perosion_exp("f13",slp,"KHt") ;
erosion_predicted('brl_cc_courseclay_exp', slp, 'erosion') = Perosion_exp("f13",slp,"HtS") ;
*Cultivation on different soils
erosion_predicted('brl_cu_finesand_exp', slp, 'erosion') = Perosion_exp("f14",slp,"HHt") ;
erosion_predicted('brl_cu_fineclay_exp', slp, 'erosion') = Perosion_exp("f14",slp,"HsS") ;
erosion_predicted('brl_cu_coursesand_exp', slp, 'erosion') = Perosion_exp("f14",slp,"KHt") ;
erosion_predicted('brl_cu_courseclay_exp', slp, 'erosion') = Perosion_exp("f14",slp,"HtS") ;
```

```
*Direct sowing on different soils
erosion_predicted('brl_ds_finesand_exp', slp, 'erosion') = Perosion_exp("f15",slp,"HHt")  ;
erosion_predicted('brl_ds_fineclay_exp', slp, 'erosion') = Perosion_exp("f15",slp,"HsS")  ;
erosion_predicted('brl_ds_coursesand_exp', slp, 'erosion') = Perosion_exp("f15",slp,"KHt")  ;
erosion_predicted('brl_ds_courseclay_exp', slp, 'erosion') = Perosion_exp("f15",slp,"HtS")  ;


*Data on the effect of slopes on P load under different tillage methods
P_12_HsS_15F_barley("ds_data", slp, "totP") = TP_load("f15",slp,"HsS","PAC12","pfer15")  ;
P_12_HsS_15F_barley("ds_data", slp, "slopepoints") = slopepoint(slp)   ;
P_12_HsS_15F_barley("cu_data", slp, "totP") = TP_load("f14",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_15F_barley("cu_data", slp, "slopepoints") =slopepoint(slp)   ;
P_12_HsS_15F_barley("cc_data", slp, "totP") = TP_load("f13",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_15F_barley("cc_data", slp, "slopepoints") = slopepoint(slp)  ;

*OLS linear model on the effect of slopes on P load under different tillage methods
P_12_HsS_15F_barley("ds_pred", slp, "totP") = P_load_slp("f15",slp,"HsS","PAC12","pfer15")  ;
P_12_HsS_15F_barley("ds_pred", slp, "slopepoints") = slopepoint(slp)   ;
P_12_HsS_15F_barley("cu_pred", slp, "totP") = P_load_slp("f14",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_15F_barley("cu_pred", slp, "slopepoints") = slopepoint(slp)    ;
P_12_HsS_15F_barley("cc_pred", slp, "totP") = P_load_slp("f13",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_15F_barley("cc_pred", slp, "slopepoints") = slopepoint(slp)   ;


*Data on the effect of slopes on P load under different tillage methods AS P FERTILISATION CHANGES  slp 1
P_12_HsS_barley("ds_data1", pfert, "totP") = TP_load("f15","slp1","HsS","PAC12", pfert)  ;
P_12_HsS_barley("ds_data1", pfert, "P_level") = fertP("f15","slp1","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cu_data1", pfert, "totP") = TP_load("f14","slp1","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cu_data1", pfert, "P_level") = fertP("f14","slp1","HsS","PAC12", pfert)    ;
P_12_HsS_barley("cc_data1", pfert, "totP") = TP_load("f13","slp1","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cc_data1", pfert, "P_level") = fertP("f13","slp1","HsS","PAC12", pfert)   ;

*Data on the effect of slopes on P load under different tillage methods AS P FERTILISATION CHANGES  slp 2
*P_12_HsS_barley("ds_data2", pfert, "totP") = TP_load("f15","slp2","HsS","PAC12", pfert)  ;
*P_12_HsS_barley("ds_data2", pfert, "P_level") = fertP("f15","slp2","HsS","PAC12", pfert)   ;
*P_12_HsS_barley("cu_data2", pfert, "totP") = TP_load("f14","slp2","HsS","PAC12", pfert)   ;
*P_12_HsS_barley("cu_data2", pfert, "P_level") = fertP("f14","slp2","HsS","PAC12", pfert)    ;
*P_12_HsS_barley("cc_data2", pfert, "totP") = TP_load("f13","slp2","HsS","PAC12", pfert)   ;
*P_12_HsS_barley("cc_data2", pfert, "P_level") = fertP("f13","slp2","HsS","PAC12", pfert)   ;

*Data on the effect of slopes on P load under different tillage methods AS P FERTILISATION CHANGES  slp 3
P_12_HsS_barley("ds_data3", pfert, "totP") = TP_load("f15","slp3","HsS","PAC12", pfert)  ;
P_12_HsS_barley("ds_data3", pfert, "P_level") = fertP("f15","slp3","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cu_data3", pfert, "totP") = TP_load("f14","slp3","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cu_data3", pfert, "P_level") = fertP("f14","slp3","HsS","PAC12", pfert)    ;
P_12_HsS_barley("cc_data3", pfert, "totP") = TP_load("f13","slp3","HsS","PAC12", pfert)   ;
P_12_HsS_barley("cc_data3", pfert, "P_level") = fertP("f13","slp3","HsS","PAC12", pfert)   ;

*Data on the effect of slopes on runoff under different tillage methods
runoff_12_HsS_15F_barley("ds_data", slp, "runoff") = runoff("f15",slp,"HsS","PAC12","pfer15")  ;
runoff_12_HsS_15F_barley("ds_data", slp, "slopepoints") = slopepoint(slp)   ;
runoff_12_HsS_15F_barley("cu_data", slp, "runoff") = runoff("f14",slp,"HsS","PAC12","pfer15")   ;
```

```
runoff_12_HsS_15F_barley("cu_data", slp, "slopepoints") = slopepoint(slp)    ;
runoff_12_HsS_15F_barley("cc_data", slp, "runoff") = runoff("f13",slp,"HsS","PAC12","pfer15")   ;
runoff_12_HsS_15F_barley("cc_data", slp, "slopepoints") = slopepoint(slp)   ;

*X-axis
runoff_barley_cc(bscen2, slp, "slopepoints") = slopepoint(slp)                   ;
*Y-axis
runoff_barley_cc("data_cc_finesand", slp, "runoff") = runoffdata("f13",slp,'HHt')     ;
runoff_barley_cc("data_cc_fineclay", slp, "runoff") = runoffdata("f13",slp,'HsS')     ;
runoff_barley_cc("data_cc_coursesand", slp, "runoff") = runoffdata("f13",slp,'KHt')   ;
runoff_barley_cc("data_cc_courseclay", slp, "runoff") = runoffdata("f13",slp,'HtS')   ;

runoff_barley_cc("predicted_cc_finesand", slp, "runoff") = Prunoff("f13",slp,'HHt')     ;
runoff_barley_cc("predicted_cc_fineclay", slp, "runoff") = Prunoff("f13",slp,'HsS')     ;
runoff_barley_cc("predicted_cc_coursesand", slp, "runoff") = Prunoff("f13",slp,'KHt')   ;
runoff_barley_cc("predicted_cc_courseclay", slp, "runoff") = Prunoff("f13",slp,'HtS')   ;


*X-axis
runoff_barley(singleCrop, slp, "slopepoints") = slopepoint(slp)                   ;
*Y-axis
*Conventional cultivation data
runoff_barley("IC_cc_fs", slp, "runoff") = runoffdata("f13",slp,'HHt')     ;
runoff_barley("IC_cc_fc", slp, "runoff") = runoffdata("f13",slp,'HsS')     ;
runoff_barley("IC_cc_cs", slp, "runoff") = runoffdata("f13",slp,'KHt')   ;
runoff_barley("IC_cc_cl", slp, "runoff") = runoffdata("f13",slp,'HtS')   ;
*Conventional cultivation prediction
runoff_barley("pred_cc_fs", slp, "runoff") = Prunoff("f13",slp,'HHt')     ;
runoff_barley("pred_cc_fc", slp, "runoff") = Prunoff("f13",slp,'HsS')     ;
runoff_barley("pred_cc_cs", slp, "runoff") = Prunoff("f13",slp,'KHt')   ;
runoff_barley("pred_cc_cl", slp, "runoff") = Prunoff("f13",slp,'HtS')   ;
*Cultivation data
runoff_barley("IC_cu_fs", slp, "runoff") = runoffdata("f14",slp,'HHt')     ;
runoff_barley("IC_cu_fc", slp, "runoff") = runoffdata("f14",slp,'HsS')     ;
runoff_barley("IC_cu_cs", slp, "runoff") = runoffdata("f14",slp,'KHt')   ;
runoff_barley("IC_cu_cl", slp, "runoff") = runoffdata("f14",slp,'HtS')   ;
*Conventional cultivation prediction
runoff_barley("pred_cu_fs", slp, "runoff") = Prunoff("f14",slp,'HHt')     ;
runoff_barley("pred_cu_fc", slp, "runoff") = Prunoff("f14",slp,'HsS')     ;
runoff_barley("pred_cu_cs", slp, "runoff") = Prunoff("f14",slp,'KHt')   ;
runoff_barley("pred_cu_cl", slp, "runoff") = Prunoff("f14",slp,'HtS')   ;
*Direct till  data
runoff_barley("IC_ds_fs", slp, "runoff") = runoffdata("f15",slp,'HHt')     ;
runoff_barley("IC_ds_fc", slp, "runoff") = runoffdata("f15",slp,'HsS')     ;
runoff_barley("IC_ds_cs", slp, "runoff") = runoffdata("f15",slp,'KHt')   ;
runoff_barley("IC_ds_cl", slp, "runoff") = runoffdata("f15",slp,'HtS')   ;
*Conventional cultivation prediction
runoff_barley("pred_ds_fs", slp, "runoff") = Prunoff("f15",slp,'HHt')     ;
runoff_barley("pred_ds_fc", slp, "runoff") = Prunoff("f15",slp,'HsS')     ;
runoff_barley("pred_ds_cs", slp, "runoff") = Prunoff("f15",slp,'KHt')   ;
runoff_barley("pred_ds_cl", slp, "runoff") = Prunoff("f15",slp,'HtS')   ;
```

```
*NON-SCALED Data on the effect of slopes on erosion under different tillage methods
erosion_12_HsS_15F_barley("ds_data", slp, "erosion") = sloss("f15",slp,"HsS","PAC12","pfer15")  ;
erosion_12_HsS_15F_barley("ds_data", slp, "slopepoints") = slopepoint(slp)  ;
erosion_12_HsS_15F_barley("cu_data", slp, "erosion") = sloss("f14",slp,"HsS","PAC12","pfer15")   ;
erosion_12_HsS_15F_barley("cu_data", slp, "slopepoints") = slopepoint(slp)    ;
erosion_12_HsS_15F_barley("cc_data", slp, "erosion") = sloss("f13",slp,"HsS","PAC12","pfer15")   ;
erosion_12_HsS_15F_barley("cc_data", slp, "slopepoints") = slopepoint(slp)   ;

*NITROGEN GRAPHS-------------------------------------------------------------------
*N DATA BARLEY ONLY  ALL SOILS, ALL TILL, SLP1
*X-axis
N_dataGraphBrl_slp1(bICdata , nfert, 'Nfert') =  N_fert('f15','slp1','HHt',nfert)  ;
*Y-AXIS
N_dataGraphBrl_slp1('cc_fineclay', nfert, 'totalN') = Nloaddata('f13','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('cu_fineclay', nfert, 'totalN') = Nloaddata('f14','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('ds_fineclay', nfert, 'totalN') = Nloaddata('f15','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('cc_courseclay', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('cu_courseclay', nfert, 'totalN') = Nloaddata('f14','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('ds_courseclay', nfert, 'totalN') = Nloaddata('f15','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('cc_finesand', nfert, 'totalN') = Nloaddata('f13','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('cu_finesand', nfert, 'totalN') = Nloaddata('f14','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('ds_finesand', nfert, 'totalN') = Nloaddata('f15','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('cc_coursesand', nfert, 'totalN') = Nloaddata('f13','slp1','KHt',nfert)  ;
N_dataGraphBrl_slp1('cu_coursesand', nfert, 'totalN') = Nloaddata('f14','slp1','KHt',nfert)  ;
N_dataGraphBrl_slp1('ds_coursesand', nfert, 'totalN') = Nloaddata('f15','slp1','KHt',nfert)  ;

*COMPARISON OF 100 kg fert effect on load for different slopes
*X-axis
N_barley_slp_data(bscen, slp, 'slopepoints') = slopedata(slp)    ;
*Conventional cultivation on different soils
N_barley_slp_data('data_cc_finesand', slp, 'totalN') = N_load('f13',slp,'HHt','nfer100')  ;
N_barley_slp_data('data_cc_fineclay', slp, 'totalN') = N_load("f13",slp,"HsS",'nfer100')  ;
N_barley_slp_data('data_cc_coursesand', slp, 'totalN') = N_load("f13",slp,"KHt",'nfer100')  ;
N_barley_slp_data('data_cc_courseclay', slp, 'totalN') = N_load("f13",slp,"HtS",'nfer100')  ;
*Cultivation on different soils
N_barley_slp_data('data_cu_finesand', slp, 'totalN') = N_load("f14",slp,"HHt",'nfer100')  ;
N_barley_slp_data('data_cu_fineclay', slp, 'totalN') = N_load("f14",slp,"HsS",'nfer100')  ;
N_barley_slp_data('data_cu_coursesand', slp, 'totalN') = N_load("f14",slp,"KHt",'nfer100')  ;
N_barley_slp_data('data_cu_courseclay', slp, 'totalN') = N_load("f14",slp,"HtS",'nfer100')  ;
*Direct sowing on different soils
N_barley_slp_data('data_ds_finesand', slp, 'totalN') = N_load("f15",slp,"HHt",'nfer100')  ;
N_barley_slp_data('data_ds_fineclay', slp, 'totalN') = N_load("f15",slp,"HsS",'nfer100')  ;
N_barley_slp_data('data_ds_coursesand', slp, 'totalN') = N_load("f15",slp,"KHt",'nfer100')  ;
N_barley_slp_data('data_ds_courseclay', slp, 'totalN') = N_load("f15",slp,"HtS",'nfer100') ;

*Conventional cultivation on different soils
N_barley_slp_data('predicted_cc_finesand', slp, 'totalN') = Nload('f13',slp,'HHt','nfer100')  ;
N_barley_slp_data('predicted_cc_fineclay', slp, 'totalN') = Nload("f13",slp,"HsS",'nfer100')  ;
N_barley_slp_data('predicted_cc_coursesand', slp, 'totalN') = Nload("f13",slp,"KHt",'nfer100')  ;
N_barley_slp_data('predicted_cc_courseclay', slp, 'totalN') = Nload("f13",slp,"HtS",'nfer100')  ;
*Cultivation on different soils
```

```
N_barley_slp_data('predicted_cu_finesand', slp, 'totalN') = Nload("f14",slp,"HHt",'nfer100')  ;
N_barley_slp_data('predicted_cu_fineclay', slp, 'totalN') = Nload("f14",slp,"HsS",'nfer100')  ;
N_barley_slp_data('predicted_cu_coursesand', slp, 'totalN') = Nload("f14",slp,"KHt",'nfer100')  ;
N_barley_slp_data('predicted_cu_courseclay', slp, 'totalN') = Nload("f14",slp,"HtS",'nfer100')  ;
*Direct sowing on different soils
N_barley_slp_data('predicted_ds_finesand', slp, 'totalN') = Nload("f15",slp,"HHt",'nfer100')  ;
N_barley_slp_data('predicted_ds_fineclay', slp, 'totalN') = Nload("f15",slp,"HsS",'nfer100')  ;
N_barley_slp_data('predicted_ds_coursesand', slp, 'totalN') = Nload("f15",slp,"KHt",'nfer100')  ;
N_barley_slp_data('predicted_ds_courseclay', slp, 'totalN') = Nload("f15",slp,"HtS",'nfer100') ;

*COMPARISON BETWEEN CROPS AND TILLAGE ON SLP1-----------------------------------
*X-axis
N_dataGraph_slp1(linesallcrop, nfert, 'Nfert') =  N_fert('f34','slp1','HHt',nfert)  ;
*Y-AXIS
*sugarbeet
N_dataGraph_slp1('sbt_cc_fineclay', nfert, 'totalN') = Nloaddata('f34','slp1','HsS',nfert)  ;
N_dataGraph_slp1('sbt_cu_fineclay', nfert, 'totalN') = Nloaddata('f35','slp1','HsS',nfert)  ;
N_dataGraph_slp1('sbt_ds_fineclay', nfert, 'totalN') = Nloaddata('f36','slp1','HsS',nfert)  ;
N_dataGraph_slp1('sbt_cc_courseclay', nfert, 'totalN') = Nloaddata('f34','slp1','HtS',nfert)  ;
N_dataGraph_slp1('sbt_cu_courseclay', nfert, 'totalN') = Nloaddata('f35','slp1','HtS',nfert)  ;
N_dataGraph_slp1('sbt_ds_courseclay', nfert, 'totalN') = Nloaddata('f36','slp1','HtS',nfert)  ;
N_dataGraph_slp1('sbt_cc_finesand', nfert, 'totalN') = Nloaddata('f34','slp1','HHt',nfert)  ;
N_dataGraph_slp1('sbt_cu_finesand', nfert, 'totalN') = Nloaddata('f35','slp1','HHt',nfert)  ;
N_dataGraph_slp1('sbt_ds_finesand', nfert, 'totalN') = Nloaddata('f36','slp1','HHt',nfert)  ;
N_dataGraph_slp1('sbt_cc_coursesand', nfert, 'totalN') = Nloaddata('f34','slp1','KHt',nfert)  ;
N_dataGraph_slp1('sbt_cu_coursesand', nfert, 'totalN') = Nloaddata('f35','slp1','KHt',nfert)  ;
N_dataGraph_slp1('sbt_ds_coursesand', nfert, 'totalN') = Nloaddata('f36','slp1','KHt',nfert)  ;
*barley
N_dataGraph_slp1('brl_cc_fineclay', nfert, 'totalN') = Nloaddata('f13','slp1','HsS',nfert)  ;
N_dataGraph_slp1('brl_cu_fineclay', nfert, 'totalN') = Nloaddata('f14','slp1','HsS',nfert)  ;
N_dataGraph_slp1('brl_ds_fineclay', nfert, 'totalN') = Nloaddata('f15','slp1','HsS',nfert)  ;
N_dataGraph_slp1('brl_cc_courseclay', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
N_dataGraph_slp1('brl_cu_courseclay', nfert, 'totalN') = Nloaddata('f14','slp1','HtS',nfert)  ;
N_dataGraph_slp1('brl_ds_courseclay', nfert, 'totalN') = Nloaddata('f15','slp1','HtS',nfert)  ;
N_dataGraph_slp1('brl_cc_finesand', nfert, 'totalN') = Nloaddata('f13','slp1','HHt',nfert)  ;
N_dataGraph_slp1('brl_cu_finesand', nfert, 'totalN') = Nloaddata('f14','slp1','HHt',nfert)  ;
N_dataGraph_slp1('brl_ds_finesand', nfert, 'totalN') = Nloaddata('f15','slp1','HHt',nfert)  ;
N_dataGraph_slp1('brl_cc_coursesand', nfert, 'totalN') = Nloaddata('f13','slp1','KHt',nfert)  ;
N_dataGraph_slp1('brl_cu_coursesand', nfert, 'totalN') = Nloaddata('f14','slp1','KHt',nfert)  ;
N_dataGraph_slp1('brl_ds_coursesand', nfert, 'totalN') = Nloaddata('f15','slp1','KHt',nfert)  ;
*grass
N_dataGraph_slp1('grs_cc_fineclay', nfert, 'totalN') = Nloaddata('f43','slp1','HsS',nfert)  ;
N_dataGraph_slp1('grs_cu_fineclay', nfert, 'totalN') = Nloaddata('f44','slp1','HsS',nfert)  ;
N_dataGraph_slp1('grs_ds_fineclay', nfert, 'totalN') = Nloaddata('f45','slp1','HsS',nfert)  ;
N_dataGraph_slp1('grs_cc_courseclay', nfert, 'totalN') = Nloaddata('f43','slp1','HtS',nfert)  ;
N_dataGraph_slp1('grs_cu_courseclay', nfert, 'totalN') = Nloaddata('f44','slp1','HtS',nfert)  ;
N_dataGraph_slp1('grs_ds_courseclay', nfert, 'totalN') = Nloaddata('f45','slp1','HtS',nfert)  ;
N_dataGraph_slp1('grs_cc_finesand', nfert, 'totalN') = Nloaddata('f43','slp1','HHt',nfert)  ;
N_dataGraph_slp1('grs_cu_finesand', nfert, 'totalN') = Nloaddata('f44','slp1','HHt',nfert)  ;
N_dataGraph_slp1('grs_ds_finesand', nfert, 'totalN') = Nloaddata('f45','slp1','HHt',nfert)  ;
N_dataGraph_slp1('grs_cc_coursesand', nfert, 'totalN') = Nloaddata('f43','slp1','KHt',nfert)  ;
N_dataGraph_slp1('grs_cu_coursesand', nfert, 'totalN') = Nloaddata('f44','slp1','KHt',nfert)  ;
```

```
N_dataGraph_slp1('grs_ds_coursesand', nfert, 'totalN') = Nloaddata('f45','slp1','KHt',nfert)  ;
*oilseed
N_dataGraph_slp1('ols_cc_fineclay', nfert, 'totalN') = Nloaddata('f37','slp1','HsS',nfert)  ;
N_dataGraph_slp1('ols_cu_fineclay', nfert, 'totalN') = Nloaddata('f38','slp1','HsS',nfert)  ;
N_dataGraph_slp1('ols_ds_fineclay', nfert, 'totalN') = Nloaddata('f39','slp1','HsS',nfert)  ;
N_dataGraph_slp1('ols_cc_courseclay', nfert, 'totalN') = Nloaddata('f37','slp1','HtS',nfert)  ;
N_dataGraph_slp1('ols_cu_courseclay', nfert, 'totalN') = Nloaddata('f38','slp1','HtS',nfert)  ;
N_dataGraph_slp1('ols_ds_courseclay', nfert, 'totalN') = Nloaddata('f39','slp1','HtS',nfert)  ;
N_dataGraph_slp1('ols_cc_finesand', nfert, 'totalN') = Nloaddata('f37','slp1','HHt',nfert)  ;
N_dataGraph_slp1('ols_cu_finesand', nfert, 'totalN') = Nloaddata('f38','slp1','HHt',nfert)  ;
N_dataGraph_slp1('ols_ds_finesand', nfert, 'totalN') = Nloaddata('f39','slp1','HHt',nfert)  ;
N_dataGraph_slp1('ols_cc_coursesand', nfert, 'totalN') = Nloaddata('f37','slp1','KHt',nfert)  ;
N_dataGraph_slp1('ols_cu_coursesand', nfert, 'totalN') = Nloaddata('f38','slp1','KHt',nfert)  ;
N_dataGraph_slp1('ols_ds_coursesand', nfert, 'totalN') = Nloaddata('f39','slp1','KHt',nfert)  ;
*potato
N_dataGraph_slp1('pot_cc_fineclay', nfert, 'totalN') = Nloaddata('f31','slp1','HsS',nfert)  ;
N_dataGraph_slp1('pot_cu_fineclay', nfert, 'totalN') = Nloaddata('f32','slp1','HsS',nfert)  ;
N_dataGraph_slp1('pot_ds_fineclay', nfert, 'totalN') = Nloaddata('f33','slp1','HsS',nfert)  ;
N_dataGraph_slp1('pot_cc_courseclay', nfert, 'totalN') = Nloaddata('f31','slp1','HtS',nfert)  ;
N_dataGraph_slp1('pot_cu_courseclay', nfert, 'totalN') = Nloaddata('f32','slp1','HtS',nfert)  ;
N_dataGraph_slp1('pot_ds_courseclay', nfert, 'totalN') = Nloaddata('f33','slp1','HtS',nfert)  ;
N_dataGraph_slp1('pot_cc_finesand', nfert, 'totalN') = Nloaddata('f31','slp1','HHt',nfert)  ;
N_dataGraph_slp1('pot_cu_finesand', nfert, 'totalN') = Nloaddata('f32','slp1','HHt',nfert)  ;
N_dataGraph_slp1('pot_ds_finesand', nfert, 'totalN') = Nloaddata('f33','slp1','HHt',nfert)  ;
N_dataGraph_slp1('pot_cc_coursesand', nfert, 'totalN') = Nloaddata('f31','slp1','KHt',nfert)  ;
N_dataGraph_slp1('pot_cu_coursesand', nfert, 'totalN') = Nloaddata('f32','slp1','KHt',nfert)  ;
N_dataGraph_slp1('pot_ds_coursesand', nfert, 'totalN') = Nloaddata('f33','slp1','KHt',nfert)  ;
*winter wheat
N_dataGraph_slp1('wwh_cc_fineclay', nfert, 'totalN') = Nloaddata('f1','slp1','HsS',nfert)  ;
N_dataGraph_slp1('wwh_cu_fineclay', nfert, 'totalN') = Nloaddata('f2','slp1','HsS',nfert)  ;
N_dataGraph_slp1('wwh_ds_fineclay', nfert, 'totalN') = Nloaddata('f3','slp1','HsS',nfert)  ;
N_dataGraph_slp1('wwh_cc_courseclay', nfert, 'totalN') = Nloaddata('f1','slp1','HtS',nfert)  ;
N_dataGraph_slp1('wwh_cu_courseclay', nfert, 'totalN') = Nloaddata('f2','slp1','HtS',nfert)  ;
N_dataGraph_slp1('wwh_ds_courseclay', nfert, 'totalN') = Nloaddata('f3','slp1','HtS',nfert)  ;
N_dataGraph_slp1('wwh_cc_finesand', nfert, 'totalN') = Nloaddata('f1','slp1','HHt',nfert)  ;
N_dataGraph_slp1('wwh_cu_finesand', nfert, 'totalN') = Nloaddata('f2','slp1','HHt',nfert)  ;
N_dataGraph_slp1('wwh_ds_finesand', nfert, 'totalN') = Nloaddata('f3','slp1','HHt',nfert)  ;
N_dataGraph_slp1('wwh_cc_coursesand', nfert, 'totalN') = Nloaddata('f1','slp1','KHt',nfert)  ;
N_dataGraph_slp1('wwh_cu_coursesand', nfert, 'totalN') = Nloaddata('f2','slp1','KHt',nfert)  ;
N_dataGraph_slp1('wwh_ds_coursesand', nfert, 'totalN') = Nloaddata('f3','slp1','KHt',nfert)  ;
*sugar beet
*X-axis
N_subeet_slp1(subeet_line, nfert, 'Nfert') =  N_fert('f34','slp1','HHt',nfert)  ;
*Y-axis
N_subeet_slp1('sbt_cc_fineclay', nfert, 'totalN') = Nloaddata('f34','slp1','HsS',nfert)  ;
N_subeet_slp1('sbt_cu_fineclay', nfert, 'totalN') = Nloaddata('f35','slp1','HsS',nfert)  ;
N_subeet_slp1('sbt_ds_fineclay', nfert, 'totalN') = Nloaddata('f36','slp1','HsS',nfert)  ;
N_subeet_slp1('sbt_cc_courseclay', nfert, 'totalN') = Nloaddata('f34','slp1','HtS',nfert)  ;
N_subeet_slp1('sbt_cu_courseclay', nfert, 'totalN') = Nloaddata('f35','slp1','HtS',nfert)  ;
N_subeet_slp1('sbt_ds_courseclay', nfert, 'totalN') = Nloaddata('f36','slp1','HtS',nfert)  ;
N_subeet_slp1('sbt_cc_finesand', nfert, 'totalN') = Nloaddata('f34','slp1','HHt',nfert)  ;
N_subeet_slp1('sbt_cu_finesand', nfert, 'totalN') = Nloaddata('f35','slp1','HHt',nfert)  ;
```

```
N_subeet_slp1('sbt_ds_finesand', nfert, 'totalN') = Nloaddata('f36','slp1','HHt',nfert)  ;
N_subeet_slp1('sbt_cc_coursesand', nfert, 'totalN') = Nloaddata('f34','slp1','KHt',nfert)  ;
N_subeet_slp1('sbt_cu_coursesand', nfert, 'totalN') = Nloaddata('f35','slp1','KHt',nfert)  ;
N_subeet_slp1('sbt_ds_coursesand', nfert, 'totalN') = Nloaddata('f36','slp1','KHt',nfert)  ;

*X-axis
N_grass_slp1(grass_line, nfert, 'Nfert') =  N_fert('f43','slp1','HHt',nfert)  ;
*Y-axis
N_grass_slp1('grs_cc_fineclay', nfert, 'totalN') = Nloaddata('f43','slp1','HsS',nfert)  ;
N_grass_slp1('grs_cu_fineclay', nfert, 'totalN') = Nloaddata('f44','slp1','HsS',nfert)  ;
N_grass_slp1('grs_ds_fineclay', nfert, 'totalN') = Nloaddata('f45','slp1','HsS',nfert)  ;
N_grass_slp1('grs_cc_courseclay', nfert, 'totalN') = Nloaddata('f43','slp1','HtS',nfert)  ;
N_grass_slp1('grs_cu_courseclay', nfert, 'totalN') = Nloaddata('f44','slp1','HtS',nfert)  ;
N_grass_slp1('grs_ds_courseclay', nfert, 'totalN') = Nloaddata('f45','slp1','HtS',nfert)  ;
N_grass_slp1('grs_cc_finesand', nfert, 'totalN') = Nloaddata('f43','slp1','HHt',nfert)  ;
N_grass_slp1('grs_cu_finesand', nfert, 'totalN') = Nloaddata('f44','slp1','HHt',nfert)  ;
N_grass_slp1('grs_ds_finesand', nfert, 'totalN') = Nloaddata('f45','slp1','HHt',nfert)  ;
N_grass_slp1('grs_cc_coursesand', nfert, 'totalN') = Nloaddata('f43','slp1','KHt',nfert)  ;
N_grass_slp1('grs_cu_coursesand', nfert, 'totalN') = Nloaddata('f44','slp1','KHt',nfert)  ;
N_grass_slp1('grs_ds_coursesand', nfert, 'totalN') = Nloaddata('f45','slp1','KHt',nfert)  ;

*COMPARISON BETWEEN CROPS AT CONVENTIONAL CULTIVATION
*X-AXIS
N_data_CC_SLP1(convcult, nfert, 'Nfert')  =  N_fert('f43','slp1','HHt',nfert)  ;
*Y-AXIS
*sugarbeet
N_data_CC_SLP1('sbt_cc_fineclay', nfert, 'totalN') = Nloaddata('f34','slp1','HsS',nfert)  ;
N_data_CC_SLP1('sbt_cc_courseclay', nfert, 'totalN') = Nloaddata('f34','slp1','HtS',nfert)  ;
N_data_CC_SLP1('sbt_cc_finesand', nfert, 'totalN') = Nloaddata('f34','slp1','HHt',nfert)  ;
N_data_CC_SLP1('sbt_cc_coursesand', nfert, 'totalN') = Nloaddata('f34','slp1','KHt',nfert)  ;
*barley
N_data_CC_SLP1('brl_cc_fineclay', nfert, 'totalN') = Nloaddata('f13','slp1','HsS',nfert)  ;
N_data_CC_SLP1('brl_cc_courseclay', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
N_data_CC_SLP1('brl_cc_finesand', nfert, 'totalN') = Nloaddata('f13','slp1','HHt',nfert)  ;
N_data_CC_SLP1('brl_cc_coursesand', nfert, 'totalN') = Nloaddata('f13','slp1','KHt',nfert)  ;
*grass
N_data_CC_SLP1('grs_cc_fineclay', nfert, 'totalN') = Nloaddata('f43','slp1','HsS',nfert)  ;
N_data_CC_SLP1('grs_cc_courseclay', nfert, 'totalN') = Nloaddata('f43','slp1','HtS',nfert)  ;
N_data_CC_SLP1('grs_cc_finesand', nfert, 'totalN') = Nloaddata('f43','slp1','HHt',nfert)  ;
N_data_CC_SLP1('grs_cc_coursesand', nfert, 'totalN') = Nloaddata('f43','slp1','KHt',nfert)  ;
*wwheat
N_data_CC_SLP1('wwh_cc_fineclay', nfert, 'totalN') = Nloaddata('f1','slp1','HsS',nfert)  ;
N_data_CC_SLP1('wwh_cc_courseclay', nfert, 'totalN') = Nloaddata('f1','slp1','HtS',nfert)  ;
N_data_CC_SLP1('wwh_cc_finesand', nfert, 'totalN') = Nloaddata('f1','slp1','HHt',nfert)  ;
N_data_CC_SLP1('wwh_cc_coursesand', nfert, 'totalN') = Nloaddata('f1','slp1','KHt',nfert)  ;
*potato
N_data_CC_SLP1('pot_cc_fineclay', nfert, 'totalN') = Nloaddata('f28','slp1','HsS',nfert)  ;
N_data_CC_SLP1('pot_cc_courseclay', nfert, 'totalN') = Nloaddata('f28','slp1','HtS',nfert)  ;
N_data_CC_SLP1('pot_cc_finesand', nfert, 'totalN') = Nloaddata('f28','slp1','HHt',nfert)  ;
N_data_CC_SLP1('pot_cc_coursesand', nfert, 'totalN') = Nloaddata('f28','slp1','KHt',nfert)  ;
*oilseed
N_data_CC_SLP1('ols_cc_fineclay', nfert, 'totalN') = Nloaddata('f37','slp1','HsS',nfert)  ;
```

```
N_data_CC_SLP1('ols_cc_courseclay', nfert, 'totalN') = Nloaddata('f37','slp1','HtS',nfert)  ;
N_data_CC_SLP1('ols_cc_finesand', nfert, 'totalN') = Nloaddata('f37','slp1','HHt',nfert)  ;
N_data_CC_SLP1('ols_cc_coursesand', nfert, 'totalN') = Nloaddata('f37','slp1','KHt',nfert)  ;

*COMPARISON OF ICECREAM DATA OF BARLEY NFERT EFFECT ON NLOAD ON ALL SOILS, SLOPES AND
TILLAGE
N_barley_data(ICreamLinesN, nfert, 'Nfert') =  N_fert('f13','slp1','HHt',nfert)  ;
*SLP1 ( 1 \%)
*cc
N_barley_data('IC_cc_fc_s1', nfert, 'totalN') = Nloaddata('f13','slp1','HsS',nfert)  ;
N_barley_data('IC_cc_cl_s1', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
N_barley_data('IC_cc_fs_s1', nfert, 'totalN') = Nloaddata('f13','slp1','HHt',nfert)  ;
N_barley_data('IC_cc_cs_s1', nfert, 'totalN') = Nloaddata('f13','slp1','KHt',nfert)  ;
*cu
N_barley_data('IC_cu_fc_s1', nfert, 'totalN') = Nloaddata('f14','slp1','HsS',nfert)  ;
N_barley_data('IC_cu_cl_s1', nfert, 'totalN') = Nloaddata('f14','slp1','HtS',nfert)  ;
N_barley_data('IC_cu_fs_s1', nfert, 'totalN') = Nloaddata('f14','slp1','HHt',nfert)  ;
N_barley_data('IC_cu_cs_s1', nfert, 'totalN') = Nloaddata('f14','slp1','KHt',nfert)  ;
*ds
N_barley_data('IC_ds_fc_s1', nfert, 'totalN') = Nloaddata('f15','slp1','HsS',nfert)  ;
N_barley_data('IC_ds_cl_s1', nfert, 'totalN') = Nloaddata('f15','slp1','HtS',nfert)  ;
N_barley_data('IC_ds_fs_s1', nfert, 'totalN') = Nloaddata('f15','slp1','HHt',nfert)  ;
N_barley_data('IC_ds_cs_s1', nfert, 'totalN') = Nloaddata('f15','slp1','KHt',nfert)  ;
*SLP0 ( 0.5 \%)
*cc
N_barley_data('IC_cc_fc_s0', nfert, 'totalN') = Nloaddata('f13','slp0','HsS',nfert)  ;
N_barley_data('IC_cc_cl_s0', nfert, 'totalN') = Nloaddata('f13','slp0','HtS',nfert)  ;
N_barley_data('IC_cc_fs_s0', nfert, 'totalN') = Nloaddata('f13','slp0','HHt',nfert)  ;
N_barley_data('IC_cc_cs_s0', nfert, 'totalN') = Nloaddata('f13','slp0','KHt',nfert)  ;
*cu
N_barley_data('IC_cu_fc_s0', nfert, 'totalN') = Nloaddata('f14','slp0','HsS',nfert)  ;
N_barley_data('IC_cu_cl_s0', nfert, 'totalN') = Nloaddata('f14','slp0','HtS',nfert)  ;
N_barley_data('IC_cu_fs_s0', nfert, 'totalN') = Nloaddata('f14','slp0','HHt',nfert)  ;
N_barley_data('IC_cu_cs_s0', nfert, 'totalN') = Nloaddata('f14','slp0','KHt',nfert)  ;
*ds
N_barley_data('IC_ds_fc_s0', nfert, 'totalN') = Nloaddata('f15','slp0','HsS',nfert)  ;
N_barley_data('IC_ds_cl_s0', nfert, 'totalN') = Nloaddata('f15','slp0','HtS',nfert)  ;
N_barley_data('IC_ds_fs_s0', nfert, 'totalN') = Nloaddata('f15','slp0','HHt',nfert)  ;
N_barley_data('IC_ds_cs_s0', nfert, 'totalN') = Nloaddata('f15','slp0','KHt',nfert)  ;
*SLP3 ( 3 \%)
*cc
N_barley_data('IC_cc_fc_s3', nfert, 'totalN') = Nloaddata('f13','slp3','HsS',nfert)  ;
N_barley_data('IC_cc_cl_s3', nfert, 'totalN') = Nloaddata('f13','slp3','HtS',nfert)  ;
N_barley_data('IC_cc_fs_s3', nfert, 'totalN') = Nloaddata('f13','slp3','HHt',nfert)  ;
N_barley_data('IC_cc_cs_s3', nfert, 'totalN') = Nloaddata('f13','slp3','KHt',nfert)  ;
*cu
N_barley_data('IC_cu_fc_s3', nfert, 'totalN') = Nloaddata('f14','slp3','HsS',nfert)  ;
N_barley_data('IC_cu_cl_s3', nfert, 'totalN') = Nloaddata('f14','slp3','HtS',nfert)  ;
N_barley_data('IC_cu_fs_s3', nfert, 'totalN') = Nloaddata('f14','slp3','HHt',nfert)  ;
N_barley_data('IC_cu_cs_s3', nfert, 'totalN') = Nloaddata('f14','slp3','KHt',nfert)  ;
*ds
N_barley_data('IC_ds_fc_s3', nfert, 'totalN') = Nloaddata('f15','slp3','HsS',nfert)  ;
```

```
N_barley_data('IC_ds_cl_s3', nfert, 'totalN') = Nloaddata('f15','slp3','HtS',nfert)  ;
N_barley_data('IC_ds_fs_s3', nfert, 'totalN') = Nloaddata('f15','slp3','HHt',nfert)  ;
N_barley_data('IC_ds_cs_s3', nfert, 'totalN') = Nloaddata('f15','slp3','KHt',nfert)  ;
*SLP3 ( 7 \%)
*cc
N_barley_data('IC_cc_fc_s7', nfert, 'totalN') = Nloaddata('f13','slp7','HsS',nfert)  ;
N_barley_data('IC_cc_cl_s7', nfert, 'totalN') = Nloaddata('f13','slp7','HtS',nfert)  ;
N_barley_data('IC_cc_fs_s7', nfert, 'totalN') = Nloaddata('f13','slp7','HHt',nfert)  ;
N_barley_data('IC_cc_cs_s7', nfert, 'totalN') = Nloaddata('f13','slp7','KHt',nfert)  ;
*cu
N_barley_data('IC_cu_fc_s7', nfert, 'totalN') = Nloaddata('f14','slp7','HsS',nfert)  ;
N_barley_data('IC_cu_cl_s7', nfert, 'totalN') = Nloaddata('f14','slp7','HtS',nfert)  ;
N_barley_data('IC_cu_fs_s7', nfert, 'totalN') = Nloaddata('f14','slp7','HHt',nfert)  ;
N_barley_data('IC_cu_cs_s7', nfert, 'totalN') = Nloaddata('f14','slp7','KHt',nfert)  ;
*ds
N_barley_data('IC_ds_fc_s7', nfert, 'totalN') = Nloaddata('f15','slp7','HsS',nfert)  ;
N_barley_data('IC_ds_cl_s7', nfert, 'totalN') = Nloaddata('f15','slp7','HtS',nfert)  ;
N_barley_data('IC_ds_fs_s7', nfert, 'totalN') = Nloaddata('f15','slp7','HHt',nfert)  ;
N_barley_data('IC_ds_cs_s7', nfert, 'totalN') = Nloaddata('f15','slp7','KHt',nfert)  ;

*X-AXIS
N_load_HtS_CC_SLP1(convcultHtS, nfert, 'Nfert')  =  N_fert('f43','slp1','HtS',nfert)  ;
*Y-AXIS
*sugarbeet
N_load_HtS_CC_SLP1('sbt_cc_courseclay', nfert, 'totalN') = Nloaddata('f34','slp1','HtS',nfert)  ;
*barley
N_load_HtS_CC_SLP1('brl_cc_courseclay', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
*grass
N_load_HtS_CC_SLP1('grs_cc_courseclay', nfert, 'totalN') = Nloaddata('f43','slp1','HtS',nfert)  ;
*wwheat
N_load_HtS_CC_SLP1('wwh_cc_courseclay', nfert, 'totalN') = Nloaddata('f1','slp1','HtS',nfert)  ;
*potato
N_load_HtS_CC_SLP1('pot_cc_courseclay', nfert, 'totalN') = Nloaddata('f28','slp1','HtS',nfert)  ;
*oil seed
N_load_HtS_CC_SLP1('ols_cc_courseclay', nfert, 'totalN') = Nloaddata('f37','slp1','HtS',nfert)  ;

*X-axis
NloadBrlSlp1HtS(lines3, nfert, 'Nfert')  =  N_fert('f13','slp1','HtS',nfert)   ;
*Y-axis
NloadBrlSlp1HtS('cc_data', nfert, 'totalN')  =  Nloaddata('f13','slp1','HtS',nfert)   ;
NloadBrlSlp1HtS('cc_pred', nfert, 'totalN')  =  Nload('f13','slp1','HtS',nfert)       ;
NloadBrlSlp1HtS('cu_data', nfert, 'totalN')  =  Nloaddata('f14','slp1','HtS',nfert)   ;
NloadBrlSlp1HtS('cu_pred', nfert, 'totalN')  =  Nload('f14','slp1','HtS',nfert)       ;
NloadBrlSlp1HtS('ds_data', nfert, 'totalN')  =  Nloaddata('f15','slp1','HtS',nfert)   ;
NloadBrlSlp1HtS('ds_pred', nfert, 'totalN')  =  Nload('f15','slp1','HtS',nfert)       ;

*X-axis
NloadBrlSlp3HtS(lines3, nfert, 'Nfert')  =  N_fert('f13','slp1','HtS',nfert)   ;
*Y-axis
NloadBrlSlp3HtS('cc_data', nfert, 'totalN')  =  Nloaddata('f13','slp3','HtS',nfert)   ;
NloadBrlSlp3HtS('cc_pred', nfert, 'totalN')  =  Nload('f13','slp3','HtS',nfert)       ;
NloadBrlSlp3HtS('cu_data', nfert, 'totalN')  =  Nloaddata('f14','slp3','HtS',nfert)   ;
```

```
NloadBrlSlp3HtS('cu_pred', nfert, 'totalN')  =  Nload('f14','slp3','HtS',nfert)      ;
NloadBrlSlp3HtS('ds_data', nfert, 'totalN')  =  Nloaddata('f15','slp3','HtS',nfert)   ;
NloadBrlSlp3HtS('ds_pred', nfert, 'totalN')  =  Nload('f15','slp3','HtS',nfert)      ;

*X-AXIS
NloadBrlSlp3KHt(lines1, nfert, 'N_fertilization')  =  N_fert('f13','slp1','HtS',nfert)   ;
*Y-axis
* data
NloadBrlSlp3KHt('line1', nfert, 'ICdata')  =  Nloaddata('f13','slp3','KHt',nfert)   ;
NloadBrlSlp3KHt('line2', nfert, 'ICdata')  =  Nloaddata('f14','slp3','KHt',nfert)   ;
NloadBrlSlp3KHt('line3', nfert, 'ICdata')  =  Nloaddata('f15','slp3','KHt',nfert)   ;
* prediction
NloadBrlSlp3KHt('line4', nfert, 'totalNload')  =  Nload('f13','slp3','KHt',nfert)      ;
NloadBrlSlp3KHt('line5', nfert, 'totalNload')  =  Nload('f14','slp3','KHt',nfert)      ;
NloadBrlSlp3KHt('line6', nfert, 'totalNload')  =  Nload('f15','slp3','KHt',nfert)      ;

*X-axis
NloadBrlSlp3HsS(lines3, nfert, 'Nfert')  =  N_fert('f13','slp1','HtS',nfert)   ;
*Y-axis
NloadBrlSlp3HsS('cc_data', nfert, 'totalN')  =  Nloaddata('f13','slp3','HsS',nfert)   ;
NloadBrlSlp3HsS('cc_pred', nfert, 'totalN')  =  Nload('f13','slp3','HsS',nfert)      ;
NloadBrlSlp3HsS('cu_data', nfert, 'totalN')  =  Nloaddata('f14','slp3','HtS',nfert)   ;
NloadBrlSlp3HsS('cu_pred', nfert, 'totalN')  =  Nload('f14','slp3','HsS',nfert)      ;
NloadBrlSlp3HsS('ds_data', nfert, 'totalN')  =  Nloaddata('f15','slp3','HsS',nfert)   ;
NloadBrlSlp3HsS('ds_pred', nfert, 'totalN')  =  Nload('f15','slp3','HsS',nfert)      ;

*barley
N_dataGraphBrl_slp1('cc_fineclay', nfert, 'totalN') = Nloaddata('f13','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('cu_fineclay', nfert, 'totalN') = Nloaddata('f14','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('ds_fineclay', nfert, 'totalN') = Nloaddata('f15','slp1','HsS',nfert)  ;
N_dataGraphBrl_slp1('cc_courseclay', nfert, 'totalN') = Nloaddata('f13','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('cu_courseclay', nfert, 'totalN') = Nloaddata('f14','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('ds_courseclay', nfert, 'totalN') = Nloaddata('f15','slp1','HtS',nfert)  ;
N_dataGraphBrl_slp1('cc_finesand', nfert, 'totalN') = Nloaddata('f13','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('cu_finesand', nfert, 'totalN') = Nloaddata('f14','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('ds_finesand', nfert, 'totalN') = Nloaddata('f15','slp1','HHt',nfert)  ;
N_dataGraphBrl_slp1('cc_coursesand', nfert, 'totalN') = Nloaddata('f13','slp1','KHt',nfert)  ;
N_dataGraphBrl_slp1('cu_coursesand', nfert, 'totalN') = Nloaddata('f14','slp1','KHt',nfert)  ;
N_dataGraphBrl_slp1('ds_coursesand', nfert, 'totalN') = Nloaddata('f15','slp1','KHt',nfert)  ;

*$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley X: Nitrogen load plotted against N fertilisation data and OLS prediction HtS
SLP1'
*$libinclude gnuplotxyz NloadBrlSlp1HtS Nfert totalN

*$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley X: Nitrogen load plotted against N fertilisation data and OLS prediction SLP3'
*$libinclude gnuplotxyz NloadBrlSlp3HtS Nfert totalN

*$setglobal gp_style 'linespoints'
$setglobal gp_title  'Barley: Nitrogen load plotted against N fertilisation data and OLS prediction KHT SLP3'
$setglobal gp_ylabel "Nitrogen load (kg per ha)"
```

```
$setglobal gp_xlabel "Nitrogen fertilisation (kg per ha)"
$setglobal gp_style 'lines'
$setglobal gp_l1style 'points'
$setglobal gp_l2style 'points'
$setglobal gp_l3style 'points'
$setglobal gp_l4style 'lines'
$setglobal gp_l5style 'lines'
$setglobal gp_l6style 'lines'
$libinclude gnuplotxyz NloadBrlSlp3KHt N_fertilization ICdata totalNload
display NloadBrlSlp3KHt ;
*$setglobal gp_title   'Figure 1: Effect of slope on cultivation as P reducer'
*$libinclude gnuplotxyz P_12_cult_slope_graph slopepoints Pred
*$setglobal gp_title   'Barley: The effect of slope on P load under different tillage methods on HsS Pst12
PF15'
*$libinclude gnuplotxyz P_12_HsS_15F_barley slopepoints totP
$setglobal gp_title   'Barley: The effect of tillage on P load as P fertilisation is altered on HsS Pst12'
*$libinclude gnuplotxyz P_12_HsS_barley P_level totP
$setglobal gp_title   'Barley: The effect of slope on runoff under different tillage methods on HsS Pst12 PF15
'
*$libinclude gnuplotxyz runoff_12_HsS_15F_barley slopepoints runoff
$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley: The effect of slope, soil and tillage on erosion'
*$libinclude gnuplotxyz erosion_barley slope erosion
$setglobal gp_style 'points'
*$setglobal gp_title   'Barley: The effect of slope on soil erosion original ICECREAM on HsS Pst12 PF15'
*$libinclude gnuplotxyz erosion_12_HsS_15F_barley slopepoints erosion

$setglobal gp_style 'points'
$setglobal gp_title   'Re-scaled erosiond data: The effect of slope on soil erosion under different tillage
methods'
$setglobal gp_ylabel "Erosion (kg per ha) "
$setglobal gp_xlabel "Slope (\%) "
*$libinclude gnuplotxyz erosion_all_crops slopepoints erosion
$setglobal gp_style 'points'
$setglobal gp_title   'Erosiond data: The effect of slope on soil erosion under different tillage methods'
$setglobal gp_ylabel "Erosion (kg per ha) "
$setglobal gp_xlabel "Slope (\%) "
*$libinclude gnuplotxyz erosion_all_crops2 slopepoints erosion
$setglobal gp_style 'lines'
$setglobal gp_title   'The effect of slope on soil erosion of barley (different functional forms)'
$setglobal gp_ylabel "Erosion (kg per ha) "
$setglobal gp_xlabel "Slope (\%) "
*$libinclude gnuplotxyz erosion_predicted slopepoints erosion
$setglobal gp_style 'dots'
$setglobal gp_title   'Nitrogen load for all IC crop types on all soils and tillage plotted against N fertilisation'
$setglobal gp_ylabel "Total N load (kg per ha) "
$setglobal gp_xlabel "N fertilization (kg per ha) "
*$libinclude gnuplotxyz N_dataGraph_slp1 Nfert totalN
$setglobal gp_style 'linespoints'
$setglobal gp_title   'Sugarbeet nitrogen load plotted against N fertilisation on slp1'
*$libinclude gnuplotxyz N_subeet_slp1 Nfert totalN
$setglobal gp_style 'linespoints'
```

```
$setglobal gp_title   'Grass nitrogen load plotted against N fertilisation on slp1'
*$libinclude gnuplotxyz N_grass_slp1 Nfert totalN
$setglobal gp_style 'linespoints'
$setglobal gp_title   'N load for all different conventionally cultivated crops and soils on slp1'
*$libinclude gnuplotxyz N_data_CC_SLP1 Nfert totalN
$setglobal gp_style 'linespoints'
$setglobal gp_title   'N load for all different conventionally cultivated crops on Hts and slp1'
*$libinclude gnuplotxyz N_load_HtS_CC_SLP1 Nfert totalN

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley: ICECREAM model data on the effect of slope on N load on main soil types
under 100kg N fert'
*$libinclude gnuplotxyz N_barley_slp_data slopepoints totalN

$setglobal gp_style 'points'
$setglobal gp_title   'Barley: ICECREAM model data on the effect of slope on erosion'
$setglobal gp_ylabel "Erosion (kg per ha) "
$setglobal gp_xlabel "Slope (\%) "
$libinclude gnuplotxyz erosion_ICbarley slope erosion

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley: ICECREAM model data on the effect of fertilisation on Nitrogen load'
$libinclude gnuplotxyz NloadBrlSlp3 Nfert totalN

$setglobal gp_style 'linespoints'
$setglobal gp_title   'Barley: ICECREAM model data on the effect of fertilisation on Phosphorus load'
$libinclude gnuplotxyz totPfert_barley Pfert totP


$setglobal gp_style 'points'
$setglobal gp_title 'Barley N load ICECREAM data -Aurajoki'
$setglobal gp_ylabel "N load (kg per ha)"
$setglobal gp_xlabel "N fertilization (kg per ha)"
$setglobal gp_yrange 0:200
$setglobal gp_xrange 0:250
$libinclude gnuplotxyz N_barley_data Nfert totalN

$setglobal gp_style 'points'
$setglobal gp_title   'Barley -Nitrogen load plotted against N fertilisation data for SLP1)'
$setglobal gp_yrange 0:200
$setglobal gp_xrange 0:250
$libinclude gnuplotxyz N_dataGraphBrl_slp1 Nfert totalN

$ontext
set term postscript eps enhanced
set output 'C:\Documents and Settings\ttl207\Omat tiedostot\EAAE\Chania\article\Fig4.eps'
set title 'Aurajoki'
replot
exit
$offtext
$setglobal gp_title   'RUNOFF The effect of slope and soil on runoff for barley '
$setglobal gp_ylabel "Runoff (mm per year)"
```

```
$setglobal gp_xlabel "Slope (\%)"
$setglobal gp_yrange 0:250
$setglobal gp_xrange 0:8

$setglobal gp_style 'lines'
$setglobal gp_l1style 'lines'
$setglobal gp_l2style 'lines'
$setglobal gp_l3style 'lines'
$setglobal gp_l4style 'lines'
$setglobal gp_l5style 'points'
$setglobal gp_l6style 'points'
$setglobal gp_l7style 'points'
$setglobal gp_l8style 'points'
$setglobal gp_key 'bottom left'
*$setglobal gp_label_1 'pred cc fs'
*$setglobal gp_label_1 'pred cc fs'
*$setglobal gp_label_2  'pred cc fc'
*$setglobal gp_label_3  'pred cc cs'
*$setglobal gp_label_4  'pred cc cl'
*$setglobal gp_label_5  'IC cc fs'
*$setglobal gp_label_6  'IC cc fc'
*$setglobal gp_label_7  'IC cc cs'
*$setglobal gp_label_8  'IC cc cl'

$libinclude gnuplotxyz runoff_barley_cc slopepoints runoff

display runoff_barley_cc, N_dataGraph_slp1, N_load_HtS_CC_SLP1, N_data_CC_SLP1 ;
display Ncoeff_frt_b, Ncoeff_frt_c    ;

*PARAMETERS EXPORTED TO EXTRAPOLATE_LOAD_PARAM_Aura
display j, pac, pfert, nfert, soil, slopepoint, soilP, soilP_flw, flw, slopedata, runoffdata, erosiondata,
erosion_predicted, Nloaddata, N_fert      ;
display Pcoeff_slp_b, Pcoeff_slp_c,  Rcoeff_slp_b, Rcoeff_slp_c, Ecoeff_slp_b, Ecoeff_slp_c,  Pcoeff_soilP_a,
Pcoeff_soilP_b,  Ecoeff_slp_b_lin, Ecoeff_slp_c_lin, Ecoeff_slp_b_exp, Ecoeff_slp_c_exp, Ecoeff_slp_b_sqr,
Ecoeff_slp_c_sqr ;
display Pcoeff_slp_b_flw, Pcoeff_slp_c_flw, Rcoeff_slp_b_flw, Rcoeff_slp_c_flw, Ecoeff_slp_b_flw,
Ecoeff_slp_c_flw, Pcoeff_soilP_a_flw, Pcoeff_soilP_b_flw     ;
display Ncoeff_slp_b, Ncoeff_slp_c, Ncoeff_frt_b, Ncoeff_frt_c, Ncoeff_slp_b_flw, Ncoeff_slp_c_flw ;
display TP_load, TP_load_flw ;
display N_barley_slp_data ;
display runoffdata, runoff_barley, erosion_barley ;
*display runoff_12_HsS_15F_barley, ErosionBrl_cc_HtS, ErosionBrl_cu_HtS, ErosionBrl_ds_HtS,
ErosionGrs_cc_HtS, NloadBrlSlp3HsS ;

*statistical validation - not implemented
*display sse_erosio_exp2, sse_runoff_lin, sse_nitron_exp2  ;
*display R2_erosio_exp2, R2_runoff_lin, R2_nitron_exp2     ;

*GNUPLOT SCRIPTING--------------------------------------------------------------
$ontext
The figures for functions are easier to produce in GNUPLOT if the OLS is redone
in the gnuplot application. The coefficients used for graphs should be double
```

checked with the one estimated in GAMS.

EROSION
OLS in gnuplot (since functions look better when plotting the function in gnuplot)
CREATE gnuplot_erosion_Aura.dat WITH $libinclude gnuplotxyz erosion_ICbarley slope erosion (needs to be the last output from gams xyz)
*set key 'top left vertical inside'

```
set terminal postscript eps enhanced
set output 'D:\Documents\EAAE\Chania\article\Fig2.eps'

f1(x) = n1*exp(x*m1)
f2(x) = n2*exp(x*m2)
f3(x) = n3*exp(x*m3)
f4(x) = n4*exp(x*m4)
f5(x) = n5*exp(x*m5)
f6(x) = n6*exp(x*m6)
f7(x) = n7*exp(x*m7)
f8(x) = n8*exp(x*m8)
f9(x) = n1*exp(x*m9)
f10(x) = n10*exp(x*m10)
f11(x) = n11*exp(x*m11)
f12(x) = n12*exp(x*m12)

fit f1(x) "gnuplot_erosion_Aura.dat" index 0:0 using 1:2 via n1, m1
fit f2(x) "gnuplot_erosion_Aura.dat" index 1:1 using 1:2 via n2, m2
fit f3(x) "gnuplot_erosion_Aura.dat" index 2:2 using 1:2 via n3, m3
fit f4(x) "gnuplot_erosion_Aura.dat" index 3:3 using 1:2 via n4, m4
fit f5(x) "gnuplot_erosion_Aura.dat" index 4:4 using 1:2 via n5, m5
fit f6(x) "gnuplot_erosion_Aura.dat" index 5:5 using 1:2 via n6, m6
fit f7(x) "gnuplot_erosion_Aura.dat" index 6:6 using 1:2 via n7, m7
fit f8(x) "gnuplot_erosion_Aura.dat" index 7:7 using 1:2 via n8, m8
fit f9(x) "gnuplot_erosion_Aura.dat" index 8:8 using 1:2 via n9, m9
fit f10(x) "gnuplot_erosion_Aura.dat" index 9:9 using 1:2 via n10, m10
fit f11(x) "gnuplot_erosion_Aura.dat" index 10:10 using 1:2 via n11, m11
fit f12(x) "gnuplot_erosion_Aura.dat" index 11:11 using 1:2 via n12, m12
set title "Aurajoki"
set size 0.7,0.7
set yrange [0:6000]
set xrange [0:8]
set key 0,5900
set xlabel "Slope (\%)"
set ylabel "Erosion (kg per ha)"
plot f1(x) title "pred cc fs" with line ls 1 ,\
f2(x) title "pred cc fc" with line ls 2 ,\
f3(x) title "pred cc cs" with line ls 3 ,\
f4(x) title "pred cc cl" with line ls 4 ,\
f5(x) title "pred cu fs" with line ls 5 ,\
f6(x) title "pred cu fc" with line ls 6 ,\
f7(x) title "pred cu cs" with line ls 7 ,\
f8(x) title "pred cu cl" with line ls 8 ,\
f9(x) title "pred ds fs" with line ls 9 ,\
```

```
f10(x) title "pred ds fc" with line ls 10 ,\
f11(x) title "pred ds cs" with line ls 11 ,\
f12(x) title "pred ds cl" with line ls 12 ,\
"gnuplot_erosion_Aura.dat" index 0:0 using 1:2 title "IC cc fs" with points 1 ,\
"gnuplot_erosion_Aura.dat" index 1:1 using 1:2 title "IC cc fc" with points 2 ,\
"gnuplot_erosion_Aura.dat" index 2:2 using 1:2 title "IC cc cs" with points 3 ,\
"gnuplot_erosion_Aura.dat" index 3:3 using 1:2 title "IC cc cl" with points 4 ,\
"gnuplot_erosion_Aura.dat" index 4:4 using 1:2 title "IC cu fs" with points 5 ,\
"gnuplot_erosion_Aura.dat" index 5:5 using 1:2 title "IC cu fc" with points 6 ,\
"gnuplot_erosion_Aura.dat" index 6:6 using 1:2 title "IC cu cs" with points 7 ,\
"gnuplot_erosion_Aura.dat" index 7:7 using 1:2 title "IC cu cl" with points 8 ,\
"gnuplot_erosion_Aura.dat" index 8:8 using 1:2 title "IC ds fs" with points 9 ,\
"gnuplot_erosion_Aura.dat" index 9:9 using 1:2 title "IC ds fc" with points 10 ,\
"gnuplot_erosion_Aura.dat" index 10:10 using 1:2 title "IC ds cs" with points 11 ,\
"gnuplot_erosion_Aura.dat" index 11:11 using 1:2 title "IC ds cl" with points 12 ;
exit
```

RUNOFF
-linear plots directly from gnuplot xyz utility for gams, code in the previous section
-replotted since changing labels did not work with the gp_label option

```
set terminal postscript eps enhanced
set output 'D:\Documents\EAAE\Chania\article\Fig4.eps'
set title "Aurajoki"
set key 0,140
set key height 10
set size 0.7,0.7
set yrange [0:250]
set xrange [0:8]
set xlabel "Slope (\%)"
set ylabel "Runoff (ml per ha)"
plot "gnuplot_runoff_Aura.dat" index 0:0 using 1:2 title "pred fs" with line 1 ,\
"gnuplot_runoff_Aura.dat" index 1:1 using 1:2 title "pred fc" with line 2 ,\
"gnuplot_runoff_Aura.dat" index 2:2 using 1:2 title "pred cs" with line 3 ,\
"gnuplot_runoff_Aura.dat" index 3:3 using 1:2 title "pred cl" with line 4 ,\
"gnuplot_runoff_Aura.dat" index 4:4 using 1:2 title "IC fs" with points 5 ,\
"gnuplot_runoff_Aura.dat" index 5:5 using 1:2 title "IC fc" with points 6 ,\
"gnuplot_runoff_Aura.dat" index 6:6 using 1:2 title "IC cs" with points 7 ,\
"gnuplot_runoff_Aura.dat" index 7:7 using 1:2 title "IC cl" with points 8
exit
```

NITROGEN

CREATE gnuplot_Nload_Aura.dat WITH $libinclude gnuplotxyz N_dataGraphBrl_slp1 Nfert totN (needs to be the last output from gams xyz)

```
set terminal postscript eps enhanced
set output 'D:\Documents\EAAE\Chania\article\Fig6.eps'

g1(x) = k1*exp(x*l1)
g2(x) = k2*exp(x*l2)
g3(x) = k3*exp(x*l3)
```

```
g4(x) = k4*exp(x*l4)
g5(x) = k5*exp(x*l5)
g6(x) = k6*exp(x*l6)
g7(x) = k7*exp(x*l7)
g8(x) = k8*exp(x*l8)
g9(x) = k1*exp(x*l9)
g10(x) = k10*exp(x*l10)
g11(x) = k11*exp(x*l11)
g12(x) = k12*exp(x*l12)

k1 = 3
l1 = 0.01
k2 = 3
l2 = 0.01
k3 = 3
l3 = 0.01
k4 = 3
l4 = 0.01
k5 = 3
l5 = 0.01
k6 = 3
l6 = 0.01
k7 = 3
l7 = 0.01
k8 = 3
l8 = 0.01
k9 = 3
l9 = 0.01
k10 = 3
l10 = 0.01
k11 = 3
l11 = 0.01
k12 = 3
l12 = 0.01
set yrange [0:150]
set xrange [0:250]
fit g1(x) "gnuplot_Nload_Aura.dat" index 0:0 using 1:2 via k1, l1
fit g2(x) "gnuplot_Nload_Aura.dat" index 1:1 using 1:2 via k2, l2
fit g3(x) "gnuplot_Nload_Aura.dat" index 2:2 using 1:2 via k3, l3
fit g4(x) "gnuplot_Nload_Aura.dat" index 3:3 using 1:2 via k4, l4
fit g5(x) "gnuplot_Nload_Aura.dat" index 4:4 using 1:2 via k5, l5
fit g6(x) "gnuplot_Nload_Aura.dat" index 5:5 using 1:2 via k6, l6
fit g7(x) "gnuplot_Nload_Aura.dat" index 6:6 using 1:2 via k7, l7
fit g8(x) "gnuplot_Nload_Aura.dat" index 7:7 using 1:2 via k8, l8
fit g9(x) "gnuplot_Nload_Aura.dat" index 8:8 using 1:2 via k9, l9
fit g10(x) "gnuplot_Nload_Aura.dat" index 9:9 using 1:2 via k10, l10
fit g11(x) "gnuplot_Nload_Aura.dat" index 10:10 using 1:2 via k11, l11
fit g12(x) "gnuplot_Nload_Aura.dat" index 11:11 using 1:2 via k12, l12
set title "Aurajoki"
set size 0.7,0.7
set key 0,140
set xlabel "Nitrogen fertilization (kg per ha)"
```

```
set ylabel "Nitrogen load (kg per ha)"
plot g1(x) title "pred cc fs" with line ls 1 ,\
g2(x) title "pred cc fc" with line ls 2 ,\
g3(x) title "pred cc cs" with line ls 3 ,\
g4(x) title "pred cc cl" with line ls 4 ,\
g5(x) title "pred cu fs" with line ls 5 ,\
g6(x) title "pred cu fc" with line ls 6 ,\
g7(x) title "pred cu cs" with line ls 7 ,\
g8(x) title "pred cu cl" with line ls 8 ,\
g9(x) title "pred ds fs" with line ls 9 ,\
g10(x) title "pred ds fc" with line ls 10 ,\
g11(x) title "pred ds cs" with line ls 11 ,\
g12(x) title "pred ds cl" with line ls 12 ,\
"gnuplot_Nload_Aura.dat" index 0:0 using 1:2 title "IC cc fs" with points 1 ,\
"gnuplot_Nload_Aura.dat" index 1:1 using 1:2 title "IC cc fc" with points 2 ,\
"gnuplot_Nload_Aura.dat" index 2:2 using 1:2 title "IC cc cs" with points 3 ,\
"gnuplot_Nload_Aura.dat" index 3:3 using 1:2 title "IC cc cl" with points 4 ,\
"gnuplot_Nload_Aura.dat" index 4:4 using 1:2 title "IC cu fs" with points 5 ,\
"gnuplot_Nload_Aura.dat" index 5:5 using 1:2 title "IC cu fc" with points 6 ,\
"gnuplot_Nload_Aura.dat" index 6:6 using 1:2 title "IC cu cs" with points 7 ,\
"gnuplot_Nload_Aura.dat" index 7:7 using 1:2 title "IC cu cl" with points 8 ,\
"gnuplot_Nload_Aura.dat" index 8:8 using 1:2 title "IC ds fs" with points 9 ,\
"gnuplot_Nload_Aura.dat" index 9:9 using 1:2 title "IC ds fc" with points 10 ,\
"gnuplot_Nload_Aura.dat" index 10:10 using 1:2 title "IC ds cs" with points 11 ,\
"gnuplot_Nload_Aura.dat" index 11:11 using 1:2 title "IC ds cl" with points 12 ;
exit
$offtext
```

Load parameters from OLS estimated functions to the model slope classes

```
$title Load parameters from OLS estimated functions to the model slope classes

$ontext
* SOLVING THE SLOPE PROBLEM: FIVE CLASSES IN THE IC, 6 CLASSES IN THE DATA
-> EXTRA/INTERPOLATE FOR 6 CLASSES IN DATA
Since ICECREAM results have not been calculated for all land use data types, some data classes in economic abatement model would not have load parameters.
This file is the second step in creating load functions that cover all data types specified in the economic model. It contains the code for estimating the load parameter for the new slope classes based on the equations in the previous step. First coefficients are imported, then the new sets defined by the GIS are imported. Then coefficients are used to calculate erosion and runoff parameters for the new slope classes.
Similarly, total P and N loads are calculated for each slope class. These will be then used for estimating direct TP and N load functions. (However TP load is used just for comparison and loads are later calculated from erosion and runoff.)
The ICECREAM results are for four (Finnish) soil types. Data is based on FAO classes, which will be all given one of the ICECREAM soil types.
slp  -> new slp from GIS
soil -> slt (20 FAO classes)
The main results from this file are P load function parameters: runoff and sloss as well as  nitrogen load function coefficients Ncoeff_frtREV_b, Ncoeff_frtREV_c
$offtext
*SETS
sets
 j      farmed crops
 flw    fallow classes
 soil   Finnish soil types  of ICECREAM parameters
 pac    different P content in soil of ICECREAM parameters
 pfert  different levels of P fertilisation of ICECREAM parameters
 nfert  different levels of N fertilisation of ICECREAM parameters
;
*Coefficient parameters from Load_OLS_Aura.gms file
parameters
 Pcoeff_slp_b(j,soil,pac,pfert)         Coefficient P load
 Pcoeff_slp_c(j,soil,pac,pfert)         Coefficient P load
 Rcoeff_slp_b(j,soil)                   Coefficient runoff
 Rcoeff_slp_c(j,soil)                   Coefficient runoff
 Ecoeff_slp_b(j,soil)                   Coefficient erosion
 Ecoeff_slp_c(j,soil)                   Coefficient erosion
 Ecoeff_slp_b_exp(j,soil)               Coefficient erosion
 Ecoeff_slp_c_exp(j,soil)               Coefficient erosion
 Ecoeff_slp_b_exp2(j,soil)              Coefficient erosion
 Ecoeff_slp_c_exp2(j,soil)              Coefficient erosion
 Ecoeff_slp_b_lin(j,soil)               Coefficient erosion
 Ecoeff_slp_c_lin(j,soil)               Coefficient erosion
 Ecoeff_slp_b_sqr(j,soil)               Coefficient erosion
 Ecoeff_slp_c_sqr(j,soil)               Coefficient erosion
 Pcoeff_slp_b_flw(flw,soil,pac,pfert)   Coefficient P load
 Pcoeff_slp_c_flw(flw,soil,pac,pfert)   Coefficient P load
 Rcoeff_slp_b_flw(flw,soil)             Coefficient runoff
```

```
 Rcoeff_slp_c_flw(flw,soil)            Coefficient runoff
 Ecoeff_slp_b_flw(flw,soil)            Coefficient erosion
 Ecoeff_slp_c_flw(flw,soil)            Coefficient erosion
 Ncoeff_slp_b(j,soil,nfert)            Coefficient nitrogen
 Ncoeff_slp_c(j,soil,nfert)            Coefficient nitrogen
;

*REDEFINING THE SLOPE SET TO MATCH LAND DATA INSTEAD OF LOAD PARAMETERS
* Requires mdb2gms  (http://www.gams.com/dd/docs/tools/mdb2gms.pdf)
* This is multiquery batch version

$onecho > cmd.txt
I="D:\GIS\SAMA_database2003_INPUT.mdb"
X=Aurajoki_load_param2.gdx
Q1=select slp from 6slopes
s1=slp
Q2=select soil from setsoil
s2=slt
Q3=select Pst from setPstatus
s3=Pst
Q5=select slp, slopeclassmean from setslp_Aura_6class
p5=slopeclassmean
Q4=select Pst, Pst_no from setPstatus
p4=soil_P
$offecho
set slp slope classes in GIS data  (6 slope classes define in GIS and Access)
   slt soil types  of GIS data (based on FAO classification)
   Pst Phosphorus status of GIS data (average municipal soil P rounded to integers)
;
scalar meanSlope from ArcGIS zonal stastistics / 1.42  / ;
*Parameters for the data import from GIS and some of the results
parameters
 slopeclassmean(slp)                 Value of the slope class in the GIS DATA
 slopepoint(slp)                     Slope of ICECREAM point estimates
 slopedata(slp)                      Value of the slope class data point
 slopeClassUpLim(slp)                Slope class upper bound values of GIS data
 soilP(j,slp,soil,pac,pfert)         Soil P status for the ICECREAM data points
 soil_P(Pst)                         Soil P status for all integers in the GIS data
 soilPs(pac)                         Soil P status for the ICECREAM data points
*Total P, erosion and runoff parameters for crops
 TPload(j,slp,soil,pac,pfert)        Predicted total P load
 TPL_soilP(j,slp,soil,Pst,pfert)     Predicted total P load as soil P status varies
 TPload_IC(j,slp,soil,Pst,pfert)     Predicted total P load on different soil types  (from ICECREAM instead of
Uusitalo)
 TPloadIC(j,slp,slt,Pst,pfert)       Predicted total P load on different soil types  (from ICECREAM instead of
Uusitalo)
 TP_load(j,slp,soil,pac,pfert)       Data on total P load (directly from ICECREAM but note that the slope
classes are not from GIS)
 Prunoff(j,slp,soil)                 Predicted runoff  (ICECREAM soil types)
 runoff(j,slp,slt)                   Predicted runoff  (GIS data soil types)
 Perosion(j,slp,soil)                Predicted erosion (ICECREAM soil types)
 sloss(j,slp,slt)                    Predicted erosion (GIS data soil types)
```

runoffdata(j,slp,soil) Runoff data with ICECREAM slope classes (not calculated by the estiamated functions)
erosiondata(j,slp,soil) Erosion data with ICECREAM slope classes (not calculated by the estiamated functions)
*The mean loads are not calculated in this file
meanPrunoff(j) Mean runoff for each crop
meanPerosion(j) Mean erosion for each crop
*Total P, erosion and runoff parameters for fallow
TP_load_flw(flw,slp,soil,pac,pfert) Data on total P load from ICECREAM
TPload_IC_flw(flw,slp,soil,Pst,pfert) Predicted total P load on different soil types on fallow
TPloadIC_flw(flw,slp,slt,Pst,pfert) Predicted total P load on different soil types on fallow corrected for 20 soil classes
Prunoff_flw(flw,slp,soil) Predicted runoff
runoff_flw(flw,slp,slt) Generalised runoff for all soil types and different soil P levels
Perosion_flw(flw,slp,soil) Predicted erosion
sloss_flw(flw,slp,slt) Generalised runoff for all soil types and different soil P levels
OLD_sloss_flw(flw,slp,soil) Old model version soil loss
OLD_runoff_flw(flw,slp,soil) Old model version runoff
soilP_flw(flw,slp,soil,pac,pfert) Soil P values for fallow
* Coefficients for direct P loads (as function of soil P and slope/soil)
Pcoeff_soilP_a(j,soil,slp,pfert) Coefficient a P load as soil P function
Pcoeff_soilP_b(j,soil,slp,pfert) Coefficient b P load as soil P function
Pcoeff_slt_slp_a(j,soil,slp,pfert) Coefficient for log function of soil P status effect on TP load calculated from linear OLS for slope results
Pcoeff_slt_slp_b(j,soil,slp,pfert) Coefficient for log function of soil P status effect on TP load calculated from linear OLS for slope results (constant)
*Fallow
Pcoeff_slt_slp_a_flw(flw,soil,slp,pfert) Coefficient for log function of soil P status effect on TP load calculated from linear OLS for slope results
Pcoeff_slt_slp_b_flw(flw,soil,slp,pfert) Coefficient for log function of soil P status effect on TP load calculated from linear OLS for slope results (constant)
Pcoeff_soilP_a_flw(flw,soil,slp,pfert) Coefficient a P load as soil P function
Pcoeff_soilP_b_flw(flw,soil,slp,pfert) Coefficient a P load as soil P function
*Parameters for checking for missing and inconsistent data
soilNODATA(j,slp,soil,pac,pfert) Data with NO SOIL CLASS
TPload_nodata(j,slp,soil,pac,pfert) Combinations without any load (no data)
diffTPL(j,slp,soil,Pst,pfert) Difference between the total P load predictions of linear estimation and original model data (not valid for slope 5 and 6)
diffTPL4(j,soil,Pst,pfert) Difference between the total P load predictions of linear estimation and original model data for 4 slope class
sumNegTPloads SUM of count of TP loads that have negative value
NegTPloads(j,slp,soil,Pst,pfert) Count of TP loads that have negative value
sloss_test(j,slp,slt) Checking new soil loss values
*Nitrogen parameters
Ncoeff_slp_b(j,soil,nfert) N load coefficients as function of slope (to estimate N load in this file as function of fertilisation)
Ncoeff_slp_c(j,soil,nfert) N load coefficients as function of slope (to estimate N load in this file as function of fertilisation)
Ncoeff_slp_b_flw(flw,soil,nfert) N load coefficients as function of slope fallow
Ncoeff_slp_c_flw(flw,soil,nfert) N load coefficients as function of slope fallow
Nload(j,slp,soil,nfert) Estimated nitrogen load from crops

```
 Nloaddata(j,slp,soil,nfert)          Nitrogen load with grass correction (not calculated by the estimated
functions)
 NloadIC(j,slp,slt,nfert)             Estimated nitrogen load from crops (ICECREAM)
 NloadIC_flw(flw,slp,slt,nfert)        Estimated nitrogen load from fallow (ICECREAM)
 Nload_flw(flw,slp,soil,nfert)         Estimated nitrogen load from fallow
 Ncoeff_frt_b(j,soil,slp)              Estimated nitrogen load function param b (from previous results as N load
as function of slope)
 Ncoeff_frt_c(j,soil,slp)              Estimated nitrogen load function param c (from previous results as N load
as function of slope)
 N_fert(j,slp,soil,nfert)              Nitrogen fertiliser quantities in the N load function
 N_frtlzn(nfert)                       Nitrogen fertiliser quantities in the N load function  (reduced extra
dimensions away)
 Ncoeff_frtREV_b(j,slp,slt)            Revised coefficients for N load estimated from ICECREAM parameters
directly (to load_Aurajoki -calculations)
 Ncoeff_frtREV_c(j,slp,slt)            Revised coefficients for N load estimated from ICECREAM parameters
directly (to load_Aurajoki -calculations)
 NloadICestm(j,slp,soil,nfert)         Revised Nitrogen load based solely on ICECREAM parameters (no
simmelsgaard etc)
 meanNload(j,soil,nfert)               Mean nitrogen slope for crops  (over the slope classes)
 meanNload_flw(flw,soil)               Mean nitrogen slope for fallow (over the slope classes)
 mean_N_coeff_frt_b(j,soil)            Coefficient b for N load (mean over slope)
 mean_N_coeff_frt_c(j,soil)            Coefficient c for N load (mean over slope)
;
*Code for getting GIS data classes from MDB database
$call =mdb2gms \atcmd.txt
*$call =gdxviewer Aurajoki_load_param2.gdx
$gdxin 'Aurajoki_load_param2.gdx'
$load slp slt Pst slopeclassmean soil_P

*Code for getting data from gdx produced by Load_OLS_Aura
$GDXIN Aura_load_OLS.gdx
$load j pac pfert soil Pcoeff_slp_b Pcoeff_slp_c TP_load slopepoint Rcoeff_slp_b Rcoeff_slp_c Ecoeff_slp_b
Ecoeff_slp_c  Pcoeff_soilP_a Pcoeff_soilP_b soilP Ecoeff_slp_b_lin Ecoeff_slp_c_lin Ecoeff_slp_b_exp
Ecoeff_slp_c_exp Ecoeff_slp_b_sqr Ecoeff_slp_c_sqr Ecoeff_slp_b_exp2 Ecoeff_slp_c_exp2
$load flw slopedata Pcoeff_slp_b_flw Pcoeff_slp_c_flw Rcoeff_slp_b_flw Rcoeff_slp_c_flw
Ecoeff_slp_b_flw Ecoeff_slp_c_flw Pcoeff_soilP_a_flw Pcoeff_soilP_b_flw TP_load_flw soilP_flw
$load nfert Ncoeff_slp_b Ncoeff_slp_c Ncoeff_frt_b Ncoeff_frt_c Ncoeff_slp_b_flw Ncoeff_slp_c_flw
runoffdata erosiondata Nloaddata N_fert
*Simplyfy the soil P dimensions
soilPs(pac) =  soilP('f13','slp1','HHt',pac,'pfer15')   ;
*Simplyfy the soil N_fert dimensions (ALL crop soil slope combinations should have the same fertilization
quantities
N_frtlzn(nfert) =  N_fert('f1','slp1','HHt',nfert)     ;

*CALCULATE THE EROSION AND RUNOFF PARAMETERS FROM THE ESTIMATED FUNCTIONS - CHOOSE
WHICH FUNCTIONAL FORM TO USE
*(note that slope classes here correspond with data, not the previous file)
*crop
 Prunoff(j,slp,soil) =     Rcoeff_slp_b(j,soil) * slopeclassmean(slp) + Rcoeff_slp_c(j,soil)      ;
*Perosion(j,slp,soil) =    Ecoeff_slp_b_lin(j,soil) * slopeclassmean(slp) + Ecoeff_slp_c_lin(j,soil)      ;
*Perosion(j,slp,soil) =    Ecoeff_slp_b_sqr(j,soil) * sqr[slopeclassmean(slp)] + Ecoeff_slp_c_sqr(j,soil) ;
*Perosion(j,slp,soil) =    Ecoeff_slp_b_exp(j,soil) * exp[slopeclassmean(slp)] + Ecoeff_slp_c_exp(j,soil) ;
```

```
 Perosion(j,slp,soil) =     Ecoeff_slp_b_exp2(j,soil) * exp[slopeclassmean(slp)* Ecoeff_slp_c_exp2(j,soil) ]  ;
*fallow
 Prunoff_flw(flw,slp,soil) =     Rcoeff_slp_b_flw(flw,soil) * slopeclassmean(slp) + Rcoeff_slp_c_flw(flw,soil)
;
*Perosion_flw(flw,slp,soil) =    Ecoeff_slp_b_flw(flw,soil) * slopeclassmean(slp) + Ecoeff_slp_c_flw(flw,soil)
;
*Perosion_flw(flw,slp,soil) =    Ecoeff_slp_b_flw(flw,soil) * sqr[slopeclassmean(slp)] +
Ecoeff_slp_c_flw(flw,soil) ;
*Perosion_flw(flw,slp,soil) =    Ecoeff_slp_b_flw(flw,soil) * exp[slopeclassmean(slp)] +
Ecoeff_slp_c_flw(flw,soil) ;
 Perosion_flw(flw,slp,soil) =    Ecoeff_slp_b_flw(flw,soil) * exp[slopeclassmean(slp) *
Ecoeff_slp_c_flw(flw,soil)]   ;

*EROSION SCALING--------------------------------------------------------------------
*scalars   Perosion_brl_slp01_HsS ;
*parameter RelPrErosion_f13slp01HsS(j,slp,soil) Predicted erosion relative to the barley on HsS slope less
than 0.2
*        Perosion_noscale(j,slp,soil)        Non scaled predicted erosions for crops
*        RelPrEro_f13slp01HsS_flw(flw,slp,soil) Predicted erosion of fallow relative to the erosion of barley
on HsS slope less than 0.2
*        Perosion_noscale_flw(flw,slp,soil)     Non scaled predicted erosions for fallow
*;
* Perosion_brl_slp01_HsS  = Ecoeff_slp_b_exp('f13','HsS') * exp[0.1] + Ecoeff_slp_c_exp('f13','HsS')   ;
*Perosion_brl_slp01_HsS  = Ecoeff_slp_b_sqr('f13','HsS') * exp[0.1] + Ecoeff_slp_c_sqr('f13','HsS')   ;
* RelPrErosion_f13slp01HsS(j,slp,soil) = Perosion(j,slp,soil)/ Perosion_brl_slp01_HsS  ;
* SET BASELINE AS LIPERI CONVENTIONAL PLOUGH EROSION 125 (TABLE FROM PUUSTINEN)
* Perosion_noscale(j,slp,soil) = Perosion(j,slp,soil)               ;
*Perosion(j,slp,soil) = 125 * RelPrErosion_f13slp01HsS(j,slp,soil)    ;
*SCALE ALSO FALLOW TO BARLEY
* RelPrEro_f13slp01HsS_flw(flw,slp,soil) = Perosion_flw(flw,slp,soil) / Perosion_brl_slp01_HsS ;
* Perosion_noscale_flw(flw,slp,soil)     =  Perosion_flw(flw,slp,soil)         ;
*Perosion_flw(flw,slp,soil) = 125 *  RelPrEro_f13slp01HsS_flw(flw,slp,soil) ;

*CALCULATE THE TOTAL P LOAD PARAMETERS PREDICTED BY THE OLS ESTIMATED FUNCTIONS-----------------
----------------------------
 TPload(j,slp,soil,pac,pfert) = Pcoeff_slp_b(j,soil,pac,pfert) * slopeclassmean(slp) +
Pcoeff_slp_c(j,soil,pac,pfert)       ;
 TPL_soilP(j,slp,soil,Pst,pfert)  =  Pcoeff_soilP_a(j,soil,slp,pfert) * log[soil_P(Pst)] +
Pcoeff_soilP_b(j,soil,slp,pfert)  ;
*CALCULATE THE (TOTAL) N LOAD PARAMETERS PREDICTED BY THE OLS ESTIMATED SLOPE FUNCTIONS------
-----------------------------------------
 Nload(j,slp,soil,nfert) = Ncoeff_slp_b(j,soil,nfert) * slopeclassmean(slp) + Ncoeff_slp_c(j,soil,nfert)       ;
 Nload_flw(flw,slp,soil,nfert) = Ncoeff_slp_b_flw(flw,soil,nfert) * slopeclassmean(slp) +
Ncoeff_slp_c_flw(flw,soil,nfert)       ;
*calculate the N load parameters for mean slope
 meanNload(j,soil,nfert) = Ncoeff_slp_b(j,soil,nfert) * meanSlope + Ncoeff_slp_c(j,soil,nfert)       ;
 meanNload_flw(flw,soil) = Ncoeff_slp_b_flw(flw,soil,'nfer0') * meanSlope +
Ncoeff_slp_c_flw(flw,soil,'nfer0')       ;
*Indentifying missing data (if any)
 soilNODATA(j,slp,soil,pac,pfert)$(soilP(j,slp,soil,pac,pfert) EQ 0) = 1 ;
 TPload_nodata(j,slp,soil,pac,pfert)$(TPload(j,slp,soil,pac,pfert) EQ 0) = 1 ;
 display TPload_nodata, Perosion, Perosion_noscale, Perosion_flw    ;
```

```
*display Perosion_noscale_flw, RelPrErosion_f13slp01HsS, Perosion_brl_slp01_HsS ;

*ESTIMATE A TP LOAD FUNCTION BASED ON SOIL P (SIMILAR TO UUSITALO) SO ALL SOIL P CLASSES IN
DATA CAN BE REPRESENTED
*IS BASED ON METAMODEL PARAMETERS OF SLOPE FUNCTION
*RECALIBRATION OF THE SOIL P STATUS LOAD COEFFICIENTS FROM THE TP LOAD OF HIGHER SLOPE
CLASSES
Variables
 P_coeff_soilP_a(j,soil,slp,pfert)         Coefficient  a  a*log(soilP) + b
 P_coeff_soilP_b(j,soil,slp,pfert)         Coefficient  b (constant)
 residualterm(j,slp,soil,pac,pfert)        Residual
 P_coeff_soilP_a_flw(flw,soil,slp,pfert)   Coefficient  a  a*log(soilP) + b
 P_coeff_soilP_b_flw(flw,soil,slp,pfert)   Coefficient  b (constant)
 residualterm_flw(flw,slp,soil,pac,pfert)  Residual
;
*Positive variables P_coeff_soilP_a
Free variable OLS_P, OLS_P_flw
;
*Variable initiation
P_coeff_soilP_a.L(j,soil,slp,pfert)       = Pcoeff_soilP_a(j,soil,slp,pfert)     ;
P_coeff_soilP_b.L(j,soil,slp,pfert)       = Pcoeff_soilP_b(j,soil,slp,pfert)    ;
P_coeff_soilP_a.lo(j,soil,slp,pfert)      = 0.0001  ;
P_coeff_soilP_b.lo(j,soil,slp,pfert)      = 0.0001  ;
P_coeff_soilP_a_flw.L(flw,soil,slp,pfert)  = Pcoeff_soilP_a_flw(flw,soil,slp,pfert)     ;
P_coeff_soilP_b_flw.L(flw,soil,slp,pfert)  = Pcoeff_soilP_b_flw(flw,soil,slp,pfert)    ;
P_coeff_soilP_a_flw.lo(flw,soil,slp,pfert) = 0.0001  ;
P_coeff_soilP_b_flw.lo(flw,soil,slp,pfert) = 0.0001  ;

Equations
 P_soilP_exp(j,slp,soil,pac,pfert)       re-estimating the soil P effect on loads for all slope classes based on
prediction of slope effect
 OBJF_P                             the objective function of least squares to be minimised
 flw_P_soilP_exp(flw,slp,soil,pac,pfert) re-estimating the soil P effect on loads for all slope classes based on
the linear prediction of slope effect
 OBJF_P_flw                         the objective function of least squares to be minimised  (fallow)
;
*Defining the equations
P_soilP_exp(j,slp,soil,pac,pfert).. TPload(j,slp,soil,pac,pfert) =E= P_coeff_soilP_a(j,soil,slp,pfert) *
log[soilPs(pac)] + P_coeff_soilP_b(j,soil,slp,pfert) + residualterm(j,slp,soil,pac,pfert)    ;
OBJF_P..   OLS_P =E= SUM{(j,slp,soil,pac,pfert), SQR[residualterm(j,slp,soil,pac,pfert)]}    ;
flw_P_soilP_exp(flw,slp,soil,pac,pfert).. TP_load_flw(flw,slp,soil,pac,pfert) =E=
P_coeff_soilP_a_flw(flw,soil,slp,pfert) * log[soilPs(pac)] + P_coeff_soilP_b_flw(flw,soil,slp,pfert) +
residualterm_flw(flw,slp,soil,pac,pfert)    ;
OBJF_P_flw..   OLS_P_flw =E= SUM{(flw,slp,soil,pac,pfert), SQR[residualterm_flw(flw,slp,soil,pac,pfert)]}    ;

*Models and solves
 Model P_OLS_SOILP_EXP  / OBJF_P, P_soilP_exp /                       ;
 Solve P_OLS_SOILP_EXP using NLP minimizing OLS_P                      ;
 Model P_OLS_SOILP_EXP_FLW  / OBJF_P_flw, flw_P_soilP_exp /            ;
 Solve P_OLS_SOILP_EXP_FLW using NLP minimizing OLS_P_flw              ;

*TP coefficients to parameters (TP as a function of soil P)
```

```
 Pcoeff_slt_slp_a(j,soil,slp,pfert) =   P_coeff_soilP_a.L(j,soil,slp,pfert)                 ;
 Pcoeff_slt_slp_b(j,soil,slp,pfert) =   P_coeff_soilP_b.L(j,soil,slp,pfert)                 ;
 Pcoeff_slt_slp_a_flw(flw,soil,slp,pfert) =   P_coeff_soilP_a_flw.L(flw,soil,slp,pfert)        ;
 Pcoeff_slt_slp_b_flw(flw,soil,slp,pfert) =   P_coeff_soilP_b_flw.L(flw,soil,slp,pfert)        ;

*CALCULATING THE TP LOAD FOR ALL SOIL P BETWEEN 1 and 50
*( in contrast to the ICECREAM data which only for the set "pac")
 TPload_IC(j,slp,soil,Pst,pfert) =  Pcoeff_slt_slp_a(j,soil,slp,pfert) * log[soil_P(Pst)]  +
Pcoeff_slt_slp_b(j,soil,slp,pfert) ;
 TPload_IC_flw(flw,slp,soil,Pst,pfert) =  Pcoeff_slt_slp_a_flw(flw,soil,slp,pfert) * log[soil_P(Pst)]  +
Pcoeff_slt_slp_b_flw(flw,soil,slp,pfert) ;
*Any negative loads from "poorly estimated functions"
 NegTPloads(j,slp,soil,Pst,pfert)$(TPload_IC(j,slp,soil,Pst,pfert) \textless 0) = 1                    ;
 sumNegTPloads = SUM{(j,slp,soil,Pst,pfert), NegTPloads(j,slp,soil,Pst,pfert) }                  ;
 diffTPL(j,slp,soil,Pst,pfert)   = TPL_soilP(j,slp,soil,Pst,pfert) -  TPload_IC(j,slp,soil,Pst,pfert)          ;
 diffTPL4(j,soil,Pst,pfert)      = TPL_soilP(j,'slp4',soil,Pst,pfert) -  TPload_IC(j,'slp4',soil,Pst,pfert)     ;

*ESTIMATING THE NITROGEN LOAD AS FUNCTION OF FERTILISATION---------------------------------------------
* (by using the nitrogen loads calculated earlier for each slope class)
Variables
 N_coeff_frt_b(j,slp,soil)        Coefficient b ((x) = b1*exp(x*c1))
 N_coeff_frt_c(j,slp,soil)        Coefficient c ((x) = b1*exp(x*c1))
 meanN_coeff_frt_b(j,soil)        Coefficient b ((x) = b1*exp(x*c1))  (mean over slopes)
 meanN_coeff_frt_c(j,soil)        Coefficient c ((x) = b1*exp(x*c1))  (mean over slopes)
 residualtermN(j,slp,soil,nfert)   Residual
 meanResidualtermN(j,soil,nfert)   Residual
 ;
Free variable OLS_N ;

Equations
OBJF_N                     Objective function (sum of residuals)
meanOBJF_N                     Objective function for mean version (sum of residuals)
N_fert_exp(j,slp,soil,nfert)     N load as function of N fertilisation
meanN_fert_exp(j,soil,nfert)     N load as function of N fertilisation (mean over slopes)
;
OBJF_N..                     OLS_N =E= SUM{(j,slp,soil,nfert), SQR[residualtermN(j,slp,soil,nfert)]}  ;
meanOBJF_N..                   OLS_N =E= SUM{(j,slp,soil,nfert), SQR[meanResidualtermN(j,soil,nfert)]}  ;
N_fert_exp(j,slp,soil,nfert)..  Nload(j,slp,soil,nfert) =E=  N_coeff_frt_b(j,slp,soil) * exp[N_frtlzn(nfert) *
N_coeff_frt_c(j,slp,soil)] + residualtermN(j,slp,soil,nfert)    ;
meanN_fert_exp(j,soil,nfert)..  meanNload(j,soil,nfert) =E=  meanN_coeff_frt_b(j,soil) * exp[N_frtlzn(nfert)
* meanN_coeff_frt_c(j,soil)] + meanResidualtermN(j,soil,nfert)    ;

*MODELS AND SOLVES
MODEL N_LOAD_FUNCTION_OLS /  OBJF_N, N_fert_exp /  ;
MODEL meanLoadFunctionN_OLS /  meanOBJF_N, meanN_fert_exp /  ;

SOLVE  N_LOAD_FUNCTION_OLS using NLP minimizing OLS_N     ;
SOLVE  meanLoadFunctionN_OLS using NLP minimizing OLS_N   ;

*Storing the solved coefficient values to parameters
 mean_N_coeff_frt_b(j,soil) = meanN_coeff_frt_b.L(j,soil) ;
 mean_N_coeff_frt_c(j,soil) = meanN_coeff_frt_c.L(j,soil) ;
```

```
*N load calculated with estimated parameters (for the orginal fertilisation levels, but new slope classes)
NloadICestm(j,slp,soil,nfert) =   N_coeff_frt_b.L(j,slp,soil) * exp[N_frtlzn(nfert) * N_coeff_frt_c.L(j,slp,soil)]
;

display NloadICestm, N_frtlzn ;


*PARAMETER CONVERSION FROM FINNISH SOIL TYPE TO SLT CLASSES------------------------------------------
$ontext
This step would not be needed in the current model specification since the soil types in the economic
model were reduced to 4 due feasibility problems. Thus, the data could be directly divided between the
Finnish soil types. However, the economic model code is written in for slt soil classes and the soil type sets
should be changed to simplyfy the model.
$offtext

*subset of soils which are assigned with Finnish ICECREAM parameters
sets HsS_soils(slt) /slt3, slt4, slt5, slt6 /
    HtS_soils(slt) /slt8, slt16, slt17, slt18, slt19, slt20 /
    HHt_soils(slt) /slt1, slt2, slt7  /
    KHt_soils(slt) /slt9, slt10, slt11, slt12, slt13, slt14, slt15  /
 ;

sloss(j,slp,HsS_soils)   =  Perosion(j,slp,'HsS')   ;
sloss(j,slp,HtS_soils)   =  Perosion(j,slp,'HtS')   ;
sloss(j,slp,HHt_soils)   =  Perosion(j,slp,'HHt')   ;
sloss(j,slp,KHt_soils)   =  Perosion(j,slp,'KHt')   ;

runoff(j,slp,HsS_soils)   =  Prunoff(j,slp,'HsS')   ;
runoff(j,slp,HtS_soils)   =  Prunoff(j,slp,'HtS')   ;
runoff(j,slp,HHt_soils)   =  Prunoff(j,slp,'HHt')   ;
runoff(j,slp,KHt_soils)   =  Prunoff(j,slp,'KHt')   ;

TPloadIC(j,slp,HsS_soils,Pst,pfert)   =  TPload_IC(j,slp,'HsS',Pst,pfert)   ;
TPloadIC(j,slp,HtS_soils,Pst,pfert)   =  TPload_IC(j,slp,'HtS',Pst,pfert)   ;
TPloadIC(j,slp,HHt_soils,Pst,pfert)   =  TPload_IC(j,slp,'HHt',Pst,pfert)   ;
TPloadIC(j,slp,KHt_soils,Pst,pfert)   =  TPload_IC(j,slp,'KHt',Pst,pfert)   ;

NloadIC(j,slp,HsS_soils,nfert)   =  Nload(j,slp,'HsS',nfert)   ;
NloadIC(j,slp,HtS_soils,nfert)   =  Nload(j,slp,'HtS',nfert)   ;
NloadIC(j,slp,HHt_soils,nfert)   =  Nload(j,slp,'HHt',nfert)   ;
NloadIC(j,slp,KHt_soils,nfert)   =  Nload(j,slp,'KHt',nfert)   ;

Ncoeff_frtREV_b(j,slp,HsS_soils)   =  N_coeff_frt_b.L(j,slp,'HsS')   ;
Ncoeff_frtREV_b(j,slp,HtS_soils)   =  N_coeff_frt_b.L(j,slp,'HtS')   ;
Ncoeff_frtREV_b(j,slp,HHt_soils)   =  N_coeff_frt_b.L(j,slp,'HHt')   ;
Ncoeff_frtREV_b(j,slp,KHt_soils)   =  N_coeff_frt_b.L(j,slp,'KHt')   ;

Ncoeff_frtREV_c(j,slp,HsS_soils)   =  N_coeff_frt_c.L(j,slp,'HsS')   ;
Ncoeff_frtREV_c(j,slp,HtS_soils)   =  N_coeff_frt_c.L(j,slp,'HtS')   ;
Ncoeff_frtREV_c(j,slp,HHt_soils)   =  N_coeff_frt_c.L(j,slp,'HHt')   ;
Ncoeff_frtREV_c(j,slp,KHt_soils)   =  N_coeff_frt_c.L(j,slp,'KHt')   ;
```

```
sloss_flw(flw,slp,HsS_soils)   =  Perosion_flw(flw,slp,'HsS')   ;
sloss_flw(flw,slp,HtS_soils)   =  Perosion_flw(flw,slp,'HtS')   ;
sloss_flw(flw,slp,HHt_soils)   =  Perosion_flw(flw,slp,'HHt')   ;
sloss_flw(flw,slp,KHt_soils)   =  Perosion_flw(flw,slp,'KHt')   ;

runoff_flw(flw,slp,HsS_soils)   =  Prunoff_flw(flw,slp,'HsS')   ;
runoff_flw(flw,slp,HtS_soils)   =  Prunoff_flw(flw,slp,'HtS')   ;
runoff_flw(flw,slp,HHt_soils)   =  Prunoff_flw(flw,slp,'HHt')   ;
runoff_flw(flw,slp,KHt_soils)   =  Prunoff_flw(flw,slp,'KHt')   ;

TPloadIC_flw(flw,slp,HsS_soils,Pst,pfert)   =  TPload_IC_flw(flw,slp,'HsS',Pst,pfert)   ;
TPloadIC_flw(flw,slp,HtS_soils,Pst,pfert)   =  TPload_IC_flw(flw,slp,'HtS',Pst,pfert)   ;
TPloadIC_flw(flw,slp,HHt_soils,Pst,pfert)   =  TPload_IC_flw(flw,slp,'HHt',Pst,pfert)   ;
TPloadIC_flw(flw,slp,KHt_soils,Pst,pfert)   =  TPload_IC_flw(flw,slp,'KHt',Pst,pfert)   ;

NloadIC_flw(flw,slp,HsS_soils,nfert)   =  Nload_flw(flw,slp,'HsS',nfert)   ;
NloadIC_flw(flw,slp,HtS_soils,nfert)   =  Nload_flw(flw,slp,'HtS',nfert)   ;
NloadIC_flw(flw,slp,HHt_soils,nfert)   =  Nload_flw(flw,slp,'HHt',nfert)   ;
NloadIC_flw(flw,slp,KHt_soils,nfert)   =  Nload_flw(flw,slp,'KHt',nfert)   ;

*DISPLAY slp, slt, Pcoeff_slp_b, Pcoeff_slp_c, Ecoeff_slp_b, Ecoeff_slp_c, Perosion, slopeclassmean, TPload,
runoff, sloss, TPL_soilP,  diffTPL4, diffTPL  ;
DISPLAY TPload_IC, sumNegTPloads  ;

*Figure DRAWING---------------------------------------------------------------------------------------------------------
*barley on eutric cambisol2, eutric regosol,haplic podsol1, fibric-terric histosol1    (98\% coverage)
set
scen        Scenarios  of which each is a line    /
        brl_cc_HHt, brl_cc_HsS, brl_cc_KHt, brl_cc_HtS, brl_cu_HHt,  brl_cu_HsS, brl_cu_KHt, brl_cu_HtS,
brl_ds_HHt, brl_ds_HsS, brl_ds_KHt, brl_ds_HtS
        brl_cc_HHt_data, brl_cc_HsS_data, brl_cc_KHt_data, brl_cc_HtS_data, brl_cu_HHt_data,
brl_cu_HsS_data, brl_cu_KHt_data, brl_cu_HtS_data, brl_ds_HHt_data, brl_ds_HsS_data, brl_ds_KHt_data,
brl_ds_HtS_data
        /

brlscen2  / cc_fine_sand_estmt, cc_fine_clay_estmt, cc_coarse_sand_estmt, cc_coarse_clay_estmt,
cu_fine_sand_estmt,  cu_fine_clay_estmt, cu_coarse_sand_estmt, cu_coarse_clay_estmt,
ds_fine_sand_estmt, ds_fine_clay_estmt, ds_coarse_sand_estmt, ds_coarse_clay_estmt
        cc_fine_sand_data, cc_fine_clay_data, cc_coarse_sand_data, cc_coarse_clay_data,
cu_fine_sand_data, cu_fine_clay_data, cu_coarse_sand_data, cu_coarse_clay_data, ds_fine_sand_data,
ds_fine_clay_data, ds_coarse_sand_data, ds_coarse_clay_data /

brlscen  / cc_HHt, cc_HsS, cc_KHt,cc_HtS, cu_HHt,  cu_HsS, cu_KHt, cu_HtS, ds_HHt, ds_HsS, ds_KHt,
ds_HtS
        cc_HHt_data, cc_HsS_data, cc_KHt_data, cc_HtS_data, cu_HHt_data, cu_HsS_data, cu_KHt_data,
cu_HtS_data, ds_HHt_data, ds_HsS_data, ds_KHt_data, ds_HtS_data /

brldata  / cc_HHt_data, cc_HsS_data, cc_KHt_data, cc_HtS_data, cu_HHt_data, cu_HsS_data, cu_KHt_data,
cu_HtS_data, ds_HHt_data, ds_HsS_data, ds_KHt_data, ds_HtS_data /

brlscencc  / HHt, HsS, KHt, HtS, HHt_data, HsS_data, KHt_data, HtS_data /
```

```
brlNscen  / cc_HHt, cc_HsS, cc_KHt,cc_HtS, cu_HHt,  cu_HsS, cu_KHt, cu_HtS, ds_HHt, ds_HsS, ds_KHt,
ds_HtS /

p12scen / cu_data, cu_pred, ds_data, ds_pred, cc_pred, cc_data /

greenflw / HHt, HsS, KHt, HtS /


item        y and x axis items       /slope, totP, soil_loss, soilPst, totN, run_off, Nfertilisation, N_load_per_ha,
soilP /
;

parameters
 P_barley_soils_pred(scen, slp, item)     P load from normal P stock level fields on main soil types
 N_barley_soils_pred(brlscen, slp, item)  N load from 80kg per ha fertilisation on main soil types
 P_barley_psoils_pred(scen, pac, item)    P load from normal variable soil P level fields on main soil types
 P_barley_soils_pred_slp(scen, slp, item) P load from normal P stock level fields on main soil types where
different Pst levels are estimated from the linear OLS regression results of slope
 P_12_HsS_barley(p12scen, slp, item)      P load from normal P stock level fields on clay soil
 P_12_KHt_barley(p12scen, slp, item)      P load from normal P stock level fields on sandy soil
 runoff_barley(scen, slp, item)          P load from normal P stock level fields on sandy soil
 erosion_barley(brlscen, slp, item)       P load from normal P stock level fields on sandy soil
 erosion_barley_cc(brlscencc, slp, item)  P load from normal P stock level fields on sandy soil
 erosion_barley_data(brldata, slp, item)  P load from normal P stock level fields on sandy soil
 nload_barley_flat(brlscen2, nfert, item) N load from conventionally cultivated barley on different soils and
1\% slope
 TPload_barley_cc_slp1(brlscencc, pac, item) TP load from conventionally cultivated barley on different
soils and 1\% slope as a function of soil P
 erosion_fallow(greenflw, slp, item)      Fallow erosion on different soils
;

*Fallow erosion
*X-axis
erosion_fallow(greenflw, slp, 'slope') = slopeclassmean(slp) ;
*Y-axis
erosion_fallow('HHt', slp, 'soil_loss') = sloss_flw('gf1s',slp, 'slt1') ;
erosion_fallow('HsS', slp, 'soil_loss') = sloss_flw('gf1s',slp, 'slt3') ;
erosion_fallow('KHt', slp, 'soil_loss') = sloss_flw('gf1s',slp, 'slt9') ;
erosion_fallow('HtS', slp, 'soil_loss') = sloss_flw('gf1s',slp, 'slt8') ;

*Total P load from conventionally cultivated barley graph for different soil P status and soils
*X-axis
TPload_barley_cc_slp1(brlscencc, pac, 'soilP') = soilPs(pac) ;
*Y-axis
TPload_barley_cc_slp1('HHt_data', pac, 'totP') = TPload('f13','slp1','HHt',pac,'pfer15')  ;
TPload_barley_cc_slp1('HsS_data', pac, 'totP') = TPload('f13','slp1','HsS',pac,'pfer15')  ;
TPload_barley_cc_slp1('KHt_data', pac, 'totP') = TPload('f13','slp1','KHt',pac,'pfer15')  ;
TPload_barley_cc_slp1('HtS_data', pac, 'totP') = TPload('f13','slp1','HtS',pac,'pfer15')  ;
*Y-axis
TPload_barley_cc_slp1('HHt', 'pac5', 'totP') = TPload_IC('f13','slp1','HHt','pst5','pfer15')  ;
TPload_barley_cc_slp1('HsS', 'pac5', 'totP') = TPload_IC('f13','slp1','HsS','pst5','pfer15')  ;
```

```
TPload_barley_cc_slp1('KHt', 'pac5', 'totP') = TPload_IC('f13','slp1','KHt','pst5','pfer15')  ;
TPload_barley_cc_slp1('HtS', 'pac5', 'totP') = TPload_IC('f13','slp1','HtS','pst5','pfer15')  ;
TPload_barley_cc_slp1('HHt', 'pac12', 'totP') = TPload_IC('f13','slp1','HHt','pst12','pfer15')  ;
TPload_barley_cc_slp1('HsS', 'pac12', 'totP') = TPload_IC('f13','slp1','HsS','pst12','pfer15')  ;
TPload_barley_cc_slp1('KHt', 'pac12', 'totP') = TPload_IC('f13','slp1','KHt','pst12','pfer15')  ;
TPload_barley_cc_slp1('HtS', 'pac12', 'totP') = TPload_IC('f13','slp1','HtS','pst12','pfer15')  ;
TPload_barley_cc_slp1('HHt', 'pac30', 'totP') = TPload_IC('f13','slp1','HHt','pst30','pfer15')  ;
TPload_barley_cc_slp1('HsS', 'pac30', 'totP') = TPload_IC('f13','slp1','HsS','pst30','pfer15')  ;
TPload_barley_cc_slp1('KHt', 'pac30', 'totP') = TPload_IC('f13','slp1','KHt','pst30','pfer15')  ;
TPload_barley_cc_slp1('HtS', 'pac30', 'totP') = TPload_IC('f13','slp1','HtS','pst30','pfer15')  ;
TPload_barley_cc_slp1('HHt', 'pac50', 'totP') = TPload_IC('f13','slp1','HHt','pst50','pfer15')  ;
TPload_barley_cc_slp1('HsS', 'pac50', 'totP') = TPload_IC('f13','slp1','HsS','pst50','pfer15')  ;
TPload_barley_cc_slp1('KHt', 'pac50', 'totP') = TPload_IC('f13','slp1','KHt','pst50','pfer15')  ;
TPload_barley_cc_slp1('HtS', 'pac50', 'totP') = TPload_IC('f13','slp1','HtS','pst50','pfer15')  ;

*PHOSPHORUS LOAD PREDICTION FROM THE LINEAR OLS REGRESSION - TP LOAD AS FUNCTION OF SLOPE-
*X-AXIS
P_barley_soils_pred(scen, slp, 'slope') = slopeclassmean(slp)    ;
*Y-AXIS
*Conventional cultivation on different soils
P_barley_soils_pred('brl_cc_HHt', slp, 'totP') = TPload('f13',slp,'HHt','pac12','pfer15')  ;
P_barley_soils_pred('brl_cc_HsS', slp, 'totP') = TPload("f13",slp,"HsS",'pac12','pfer15')  ;
P_barley_soils_pred('brl_cc_KHt', slp, 'totP') = TPload("f13",slp,"KHt",'pac12','pfer15')  ;
P_barley_soils_pred('brl_cc_HtS', slp, 'totP') = TPload("f13",slp,"HtS",'pac12','pfer15')  ;
*Cultivation on different soils
P_barley_soils_pred('brl_cu_HHt', slp, 'totP') = TPload("f14",slp,"HHt",'pac12','pfer15')  ;
P_barley_soils_pred('brl_cu_HsS', slp, 'totP') = TPload("f14",slp,"HsS",'pac12','pfer15')  ;
P_barley_soils_pred('brl_cu_KHt', slp, 'totP') = TPload("f14",slp,"KHt",'pac12','pfer15')  ;
P_barley_soils_pred('brl_cu_HtS', slp, 'totP') = TPload("f14",slp,"HtS",'pac12','pfer15')  ;
*Cultivation on different soils
P_barley_soils_pred('brl_ds_HHt', slp, 'totP') = TPload("f15",slp,"HHt",'pac12','pfer15')  ;
P_barley_soils_pred('brl_ds_HsS', slp, 'totP') = TPload("f15",slp,"HsS",'pac12','pfer15')  ;
P_barley_soils_pred('brl_ds_KHt', slp, 'totP') = TPload("f15",slp,"KHt",'pac12','pfer15')  ;
P_barley_soils_pred('brl_ds_HtS', slp, 'totP') = TPload("f15",slp,"HtS",'pac12','pfer15')  ;

*PHOSPHORUS LOAD PREDICTION FROM THE LINEAR OLS REGRESSION  - TP LOAD AS FUNCTION OF SOIL P-
*Conventional cultivation on different soils
P_barley_psoils_pred('brl_cc_HHt', pac, 'totP') = TPload('f13','slp4','HHt',pac,'pfer15')  ;
P_barley_psoils_pred('brl_cc_HHt', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_cc_HsS', pac, 'totP') = TPload("f13",'slp4',"HsS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cc_HsS', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_cc_KHt', pac, 'totP') = TPload("f13",'slp4',"KHt",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cc_KHt', pac, 'soilPst') = soilPs(pac)  ;
P_barley_psoils_pred('brl_cc_HtS', pac, 'totP') = TPload("f13",'slp4',"HtS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cc_HtS', pac, 'soilPst') = soilPs(pac)    ;

*Cultivation on different soils
P_barley_psoils_pred('brl_cu_HHt', pac, 'totP') = TPload("f14",'slp4',"HHt",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cu_HHt', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_cu_HsS', pac, 'totP') = TPload("f14",'slp4',"HsS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cu_HsS', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_cu_KHt', pac, 'totP') = TPload("f14",'slp4',"KHt",pac,'pfer15')  ;
```

```
P_barley_psoils_pred('brl_cu_KHt', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_cu_HtS', pac, 'totP') = TPload("f14",'slp4',"HtS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_cu_HtS', pac, 'soilPst') = soilPs(pac)    ;

*Direct sowing on different soils
P_barley_psoils_pred('brl_ds_HHt', pac, 'totP') = TPload("f15",'slp4',"HHt",pac,'pfer15')  ;
P_barley_psoils_pred('brl_ds_HHt', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_ds_HsS', pac, 'totP') = TPload("f15",'slp4',"HsS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_ds_HsS', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_ds_KHt', pac, 'totP') = TPload("f15",'slp4',"KHt",pac,'pfer15')  ;
P_barley_psoils_pred('brl_ds_KHt', pac, 'soilPst') = soilPs(pac)    ;
P_barley_psoils_pred('brl_ds_HtS', pac, 'totP') = TPload("f15",'slp4',"HtS",pac,'pfer15')  ;
P_barley_psoils_pred('brl_ds_HtS', pac, 'soilPst') = soilPs(pac)    ;

*PREDICTION FROM THE LINEAR OLS REGRESSION AND FROM THAT LN FUNCTION OF PAC for different Pst levels--------------------------
*Conventional cultivation on different soils
P_barley_soils_pred_slp('brl_cc_HHt', slp, 'totP') = TPload_IC('f13',slp,'HHt','pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cc_HHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cc_HsS', slp, 'totP') = TPload_IC("f13",slp,"HsS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cc_HsS', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cc_KHt', slp, 'totP') = TPload_IC("f13",slp,"KHt",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cc_KHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cc_HtS', slp, 'totP') = TPload_IC("f13",slp,"HtS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cc_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*Cultivation on different soils
P_barley_soils_pred_slp('brl_cu_HHt', slp, 'totP') = TPload_IC("f14",slp,"HHt",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cu_HHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cu_HsS', slp, 'totP') = TPload_IC("f14",slp,"HsS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cu_HsS', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cu_KHt', slp, 'totP') = TPload_IC("f14",slp,"KHt",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cu_KHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_cu_HtS', slp, 'totP') = TPload_IC("f14",slp,"HtS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_cu_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*Direct sowing on different soils
P_barley_soils_pred_slp('brl_ds_HHt', slp, 'totP') = TPload_IC("f15",slp,"HHt",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_ds_HHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_ds_HsS', slp, 'totP') = TPload_IC("f15",slp,"HsS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_ds_HsS', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_ds_KHt', slp, 'totP') = TPload_IC("f15",slp,"KHt",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_ds_KHt', slp, 'slope') = slopeclassmean(slp)    ;
P_barley_soils_pred_slp('brl_ds_HtS', slp, 'totP') = TPload_IC("f15",slp,"HtS",'pst12','pfer15')  ;
P_barley_soils_pred_slp('brl_ds_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*PREDICTION OF TP LOAD BASED ON LINEAR OLS REGRESSION OF SLOPE
*Claye soil
P_12_HsS_barley("ds_pred", slp, "totP") = TPload("f15",slp,"HsS","PAC12","pfer15")  ;
P_12_HsS_barley("ds_pred", slp, "slope") = slopeclassmean(slp)   ;
P_12_HsS_barley("cu_pred", slp, "totP") = TPload("f14",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_barley("cu_pred", slp, "slope") = slopeclassmean(slp)    ;
```

```
P_12_HsS_barley("cc_pred", slp, "totP") = TPload("f13",slp,"HsS","PAC12","pfer15")   ;
P_12_HsS_barley("cc_pred", slp, "slope") = slopeclassmean(slp)    ;

*Sandy soil
P_12_KHt_barley("ds_pred", slp, "totP") = TPload("f15",slp,"KHt","PAC12","pfer15")  ;
P_12_KHt_barley("ds_pred", slp, "slope") = slopeclassmean(slp)   ;
P_12_KHt_barley("cu_pred", slp, "totP") = TPload("f14",slp,"KHt","PAC12","pfer15")   ;
P_12_KHt_barley("cu_pred", slp, "slope") = slopeclassmean(slp)    ;
P_12_KHt_barley("cc_pred", slp, "totP") = TPload("f13",slp,"KHt","PAC12","pfer15")   ;
P_12_KHt_barley("cc_pred", slp, "slope") = slopeclassmean(slp)    ;


*EROSION BARLEY------------------------------------------------------------------
*Conventional cultivation on different soils
erosion_barley('cc_HHt', slp, 'soil_loss') = sloss("f13",slp,"slt1")  ;
erosion_barley('cc_HHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cc_HsS', slp, 'soil_loss') = sloss("f13",slp,"slt4")  ;
erosion_barley('cc_HsS', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cc_KHt', slp, 'soil_loss') = sloss("f13",slp,"slt9")  ;
erosion_barley('cc_KHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cc_HtS', slp, 'soil_loss') = sloss("f13",slp,"slt16")  ;
erosion_barley('cc_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*Cultivation on different soils
erosion_barley('cu_HHt', slp, 'soil_loss') = sloss("f14",slp,"slt1")  ;
erosion_barley('cu_HHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cu_HsS', slp, 'soil_loss') = sloss("f14",slp,"slt4")  ;
erosion_barley('cu_HsS', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cu_KHt', slp, 'soil_loss') = sloss("f14",slp,"slt9")  ;
erosion_barley('cu_KHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('cu_HtS', slp, 'soil_loss') = sloss("f14",slp,"slt16")  ;
erosion_barley('cu_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*Direct sowing on different soils
erosion_barley('ds_HHt', slp, 'soil_loss') = sloss("f15",slp,"slt1")  ;
erosion_barley('ds_HHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('ds_HsS', slp, 'soil_loss') = sloss("f15",slp,"slt4")  ;
erosion_barley('ds_HsS', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('ds_KHt', slp, 'soil_loss') = sloss("f15",slp,"slt9")  ;
erosion_barley('ds_KHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley('ds_HtS', slp, 'soil_loss') = sloss("f15",slp,"slt16")  ;
erosion_barley('ds_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*EROSION BARLEY -DATA----------------------------------------------------------
erosion_barley('cc_HHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"HHt")  ;
erosion_barley('cc_HHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley('cc_HsS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HsS")  ;
erosion_barley('cc_HsS_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley('cc_KHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"KHt")  ;
erosion_barley('cc_KHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley('cc_HtS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HtS")  ;
erosion_barley('cc_HtS_data', slp, 'slope') = slopedata(slp)      ;
```

```
*Cultivation on different soils
erosion_barley('cu_HHt_data', slp, 'soil_loss') = erosiondata("f14",slp,"HHt")  ;
erosion_barley('cu_HHt_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley('cu_HsS_data', slp, 'soil_loss') = erosiondata("f14",slp,"HsS")  ;
erosion_barley('cu_HsS_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley('cu_KHt_data', slp, 'soil_loss') = erosiondata("f14",slp,"KHt")  ;
erosion_barley('cu_KHt_data', slp, 'slope') = slopedata(slp)     ;
erosion_barley('cu_HtS_data', slp, 'soil_loss') = erosiondata("f14",slp,"HtS")  ;
erosion_barley('cu_HtS_data', slp, 'slope') = slopedata(slp)      ;

*Direct sowing on different soils
erosion_barley('ds_HHt_data', slp, 'soil_loss') = erosiondata("f15",slp,"HHt")  ;
erosion_barley('ds_HHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley('ds_HsS_data', slp, 'soil_loss') = erosiondata("f15",slp,"HsS")  ;
erosion_barley('ds_HsS_data', slp, 'slope') = slopedata(slp)     ;
erosion_barley('ds_KHt_data', slp, 'soil_loss') = erosiondata("f15",slp,"KHt")  ;
erosion_barley('ds_KHt_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley('ds_HtS_data', slp, 'soil_loss') = erosiondata("f15",slp,"HtS")  ;
erosion_barley('ds_HtS_data', slp, 'slope') = slopedata(slp)      ;

*EROSION BARLEY CONVENTIONAL TILLAGE ONLY-----------------------------------------------------------------
*prediction
erosion_barley_cc('HHt', slp, 'soil_loss') = sloss("f13",slp,"slt1")  ;
erosion_barley_cc('HHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley_cc('HsS', slp, 'soil_loss') = sloss("f13",slp,"slt4")  ;
erosion_barley_cc('HsS', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley_cc('KHt', slp, 'soil_loss') = sloss("f13",slp,"slt9")  ;
erosion_barley_cc('KHt', slp, 'slope') = slopeclassmean(slp)    ;
erosion_barley_cc('HtS', slp, 'soil_loss') = sloss("f13",slp,"slt16")  ;
erosion_barley_cc('HtS', slp, 'slope') = slopeclassmean(slp)    ;
*ICECREAM data
erosion_barley_cc('HHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"HHt")  ;
erosion_barley_cc('HHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_cc('HsS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HsS")  ;
erosion_barley_cc('HsS_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley_cc('KHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"KHt")  ;
erosion_barley_cc('KHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_cc('HtS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HtS")  ;
erosion_barley_cc('HtS_data', slp, 'slope') = slopedata(slp)      ;

*RUNOFF BARLEY--------------------------------------------------------------------
*Conventional cultivation on different soils
runoff_barley('brl_cc_HHt', slp, 'run_off') = runoff("f13",slp,"slt1")  ;
runoff_barley('brl_cc_HHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cc_HsS', slp, 'run_off') = runoff("f13",slp,"slt4")  ;
runoff_barley('brl_cc_HsS', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cc_KHt', slp, 'run_off') = runoff("f13",slp,"slt9")  ;
runoff_barley('brl_cc_KHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cc_HtS', slp, 'run_off') = runoff("f13",slp,"slt16")  ;
runoff_barley('brl_cc_HtS', slp, 'slope') = slopeclassmean(slp)    ;
```

```
*Cultivation on different soils
runoff_barley('brl_cu_HHt', slp, 'run_off') = runoff("f14",slp,"slt1")  ;
runoff_barley('brl_cu_HHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cu_HsS', slp, 'run_off') = runoff("f14",slp,"slt4")  ;
runoff_barley('brl_cu_HsS', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cu_KHt', slp, 'run_off') = runoff("f14",slp,"slt9")  ;
runoff_barley('brl_cu_KHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_cu_HtS', slp, 'run_off') = runoff("f14",slp,"slt16")  ;
runoff_barley('brl_cu_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*Direct sowing on different soils
runoff_barley('brl_ds_HHt', slp, 'run_off') = runoff("f15",slp,"slt1")  ;
runoff_barley('brl_ds_HHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_ds_HsS', slp, 'run_off') = runoff("f15",slp,"slt4")  ;
runoff_barley('brl_ds_HsS', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_ds_KHt', slp, 'run_off') = runoff("f15",slp,"slt9")  ;
runoff_barley('brl_ds_KHt', slp, 'slope') = slopeclassmean(slp)    ;
runoff_barley('brl_ds_HtS', slp, 'run_off') = runoff("f15",slp,"slt16")  ;
runoff_barley('brl_ds_HtS', slp, 'slope') = slopeclassmean(slp)    ;

*RUNOFF BARLEY -DATA POINTS----------------------------------------------------
*Conventional cultivation on different soils
runoff_barley('brl_cc_HHt_data', slp, 'run_off') = runoffdata("f13",slp,"HHt")  ;
runoff_barley('brl_cc_HHt_data', slp, 'slope') = slopedata(slp)    ;
runoff_barley('brl_cc_HsS_data', slp, 'run_off') = runoffdata("f13",slp,"HsS")  ;
runoff_barley('brl_cc_HsS_data', slp, 'slope') = slopedata(slp)      ;
runoff_barley('brl_cc_KHt_data', slp, 'run_off') = runoffdata("f13",slp,"KHt")  ;
runoff_barley('brl_cc_KHt_data', slp, 'slope') = slopedata(slp)    ;
runoff_barley('brl_cc_HtS_data', slp, 'run_off') = runoffdata("f13",slp,"HtS")  ;
runoff_barley('brl_cc_HtS_data', slp, 'slope') = slopedata(slp)      ;

*Cultivation on different soils
runoff_barley('brl_cu_HHt_data', slp, 'run_off') = runoffdata("f14",slp,"HHt")  ;
runoff_barley('brl_cu_HHt_data', slp, 'slope') = slopedata(slp)      ;
runoff_barley('brl_cu_HsS_data', slp, 'run_off') = runoffdata("f14",slp,"HsS")  ;
runoff_barley('brl_cu_HsS_data', slp, 'slope') = slopedata(slp)    ;
runoff_barley('brl_cu_KHt_data', slp, 'run_off') = runoffdata("f14",slp,"KHt")  ;
runoff_barley('brl_cu_KHt_data', slp, 'slope') = slopedata(slp)     ;
runoff_barley('brl_cu_HtS_data', slp, 'run_off') = runoffdata("f14",slp,"HtS")  ;
runoff_barley('brl_cu_HtS_data', slp, 'slope') = slopedata(slp)      ;

*Direct sowing on different soils
runoff_barley('brl_ds_HHt_data', slp, 'run_off') = runoffdata("f15",slp,"HHt")  ;
runoff_barley('brl_ds_HHt_data', slp, 'slope') = slopedata(slp)    ;
runoff_barley('brl_ds_HsS_data', slp, 'run_off') = runoffdata("f15",slp,"HsS")  ;
runoff_barley('brl_ds_HsS_data', slp, 'slope') = slopedata(slp)     ;
runoff_barley('brl_ds_KHt_data', slp, 'run_off') = runoffdata("f15",slp,"KHt")  ;
runoff_barley('brl_ds_KHt_data', slp, 'slope') = slopedata(slp)      ;
runoff_barley('brl_ds_HtS_data', slp, 'run_off') = runoffdata("f15",slp,"HtS")  ;
runoff_barley('brl_ds_HtS_data', slp, 'slope') = slopedata(slp)      ;

*EROSION BARLEY -DATA POINTS----------------------------------------------------
```

```
*Conventional cultivation on different soils
erosion_barley_data('cc_HHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"HHt")  ;
erosion_barley_data('cc_HHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_data('cc_HsS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HsS")  ;
erosion_barley_data('cc_HsS_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley_data('cc_KHt_data', slp, 'soil_loss') = erosiondata("f13",slp,"KHt")  ;
erosion_barley_data('cc_KHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_data('cc_HtS_data', slp, 'soil_loss') = erosiondata("f13",slp,"HtS")  ;
erosion_barley_data('cc_HtS_data', slp, 'slope') = slopedata(slp)      ;
*Cultivation on different soils
erosion_barley_data('cu_HHt_data', slp, 'soil_loss') = erosiondata("f14",slp,"HHt")  ;
erosion_barley_data('cu_HHt_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley_data('cu_HsS_data', slp, 'soil_loss') = erosiondata("f14",slp,"HsS")  ;
erosion_barley_data('cu_HsS_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_data('cu_KHt_data', slp, 'soil_loss') = erosiondata("f14",slp,"KHt")  ;
erosion_barley_data('cu_KHt_data', slp, 'slope') = slopedata(slp)     ;
erosion_barley_data('cu_HtS_data', slp, 'soil_loss') = erosiondata("f14",slp,"HtS")  ;
erosion_barley_data('cu_HtS_data', slp, 'slope') = slopedata(slp)      ;
*Direct sowing on different soils
erosion_barley_data('ds_HHt_data', slp, 'soil_loss') = erosiondata("f15",slp,"HHt")  ;
erosion_barley_data('ds_HHt_data', slp, 'slope') = slopedata(slp)    ;
erosion_barley_data('ds_HsS_data', slp, 'soil_loss') = erosiondata("f15",slp,"HsS")  ;
erosion_barley_data('ds_HsS_data', slp, 'slope') = slopedata(slp)     ;
erosion_barley_data('ds_KHt_data', slp, 'soil_loss') = erosiondata("f15",slp,"KHt")  ;
erosion_barley_data('ds_KHt_data', slp, 'slope') = slopedata(slp)      ;
erosion_barley_data('ds_HtS_data', slp, 'soil_loss') = erosiondata("f15",slp,"HtS")  ;
erosion_barley_data('ds_HtS_data', slp, 'slope') = slopedata(slp)      ;
DISPLAY runoffdata, runoff, erosiondata, sloss              ;

parameters
Nbarley_slp5_est(brlNscen, nfert, item)  barley nitrogen load graph for steepest slopes
Nbarley_slp1_est(brlNscen, nfert, item)  barley nitrogen load graph for flattest slopes
 ;
*X-axis
Nbarley_slp5_est(brlNscen, nfert, 'Nfertilisation') =  N_frtlzn(nfert)       ;
Nbarley_slp1_est(brlNscen, nfert, item)   =  N_frtlzn(nfert)       ;
*Y-axis
Nbarley_slp5_est('cc_HHt', nfert, 'totN') =  NloadICestm('f13','slp5','HHt',nfert)      ;
Nbarley_slp5_est('cu_HHt', nfert, 'totN') =  NloadICestm('f14','slp5','HHt',nfert)      ;
Nbarley_slp5_est('ds_HHt', nfert, 'totN') =  NloadICestm('f15','slp5','HHt',nfert)      ;
Nbarley_slp5_est('cc_HsS', nfert, 'totN') =  NloadICestm('f13','slp5','HsS',nfert)      ;
Nbarley_slp5_est('cu_HsS', nfert, 'totN') =  NloadICestm('f14','slp5','HsS',nfert)      ;
Nbarley_slp5_est('ds_HsS', nfert, 'totN') =  NloadICestm('f15','slp5','HsS',nfert)      ;
Nbarley_slp5_est('cc_KHt', nfert, 'totN') =  NloadICestm('f13','slp5','KHt',nfert)      ;
Nbarley_slp5_est('cu_KHt', nfert, 'totN') =  NloadICestm('f14','slp5','KHt',nfert)      ;
Nbarley_slp5_est('ds_KHt', nfert, 'totN') =  NloadICestm('f15','slp5','KHt',nfert)      ;
Nbarley_slp5_est('cc_HtS', nfert, 'totN') =  NloadICestm('f13','slp5','HtS',nfert)      ;
Nbarley_slp5_est('cu_HtS', nfert, 'totN') =  NloadICestm('f14','slp5','HtS',nfert)      ;
Nbarley_slp5_est('ds_HtS', nfert, 'totN') =  NloadICestm('f15','slp5','HtS',nfert)      ;

Nbarley_slp1_est('cc_HHt', nfert, 'totN') =  NloadICestm('f13','slp1','HHt',nfert)      ;
Nbarley_slp1_est('cu_HHt', nfert, 'totN') =  NloadICestm('f14','slp1','HHt',nfert)      ;
```

```
Nbarley_slp1_est('ds_HHt', nfert, 'totN') =  NloadICestm('f15','slp1','HHt',nfert)      ;
Nbarley_slp1_est('cc_HsS', nfert, 'totN') =  NloadICestm('f13','slp1','HsS',nfert)      ;
Nbarley_slp1_est('cu_HsS', nfert, 'totN') =  NloadICestm('f14','slp1','HsS',nfert)      ;
Nbarley_slp1_est('ds_HsS', nfert, 'totN') =  NloadICestm('f15','slp1','HsS',nfert)      ;
Nbarley_slp1_est('cc_KHt', nfert, 'totN') =  NloadICestm('f13','slp1','KHt',nfert)      ;
Nbarley_slp1_est('cu_KHt', nfert, 'totN') =  NloadICestm('f14','slp1','KHt',nfert)      ;
Nbarley_slp1_est('ds_KHt', nfert, 'totN') =  NloadICestm('f15','slp1','KHt',nfert)      ;
Nbarley_slp1_est('cc_HtS', nfert, 'totN') =  NloadICestm('f13','slp1','HtS',nfert)      ;
Nbarley_slp1_est('cu_HtS', nfert, 'totN') =  NloadICestm('f14','slp1','HtS',nfert)      ;
Nbarley_slp1_est('ds_HtS', nfert, 'totN') =  NloadICestm('f15','slp1','HtS',nfert)      ;


*X-axis
nload_barley_flat(brlscen2, nfert, 'Nfertilisation') =  N_frtlzn(nfert)       ;
*Y-axis
nload_barley_flat('cc_fine_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f13','slp1','HHt',nfert)      ;
nload_barley_flat('cu_fine_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f14','slp1','HHt',nfert)      ;
nload_barley_flat('ds_fine_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f15','slp1','HHt',nfert)      ;
nload_barley_flat('cc_fine_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f13','slp1','HsS',nfert)      ;
nload_barley_flat('cu_fine_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f14','slp1','HsS',nfert)      ;
nload_barley_flat('ds_fine_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f15','slp1','HsS',nfert)      ;
nload_barley_flat('cc_coarse_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f13','slp1','KHt',nfert)
;
nload_barley_flat('cu_coarse_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f14','slp1','KHt',nfert)
;
nload_barley_flat('ds_coarse_sand_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f15','slp1','KHt',nfert)
;
nload_barley_flat('cc_coarse_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f13','slp1','HtS',nfert)
;
nload_barley_flat('cu_coarse_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f14','slp1','HtS',nfert)
;
nload_barley_flat('ds_coarse_clay_estmt', nfert, 'N_load_per_ha') =  NloadICestm('f15','slp1','HtS',nfert)
;
* ICECREAM Data points
nload_barley_flat('cc_fine_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f13','slp1','slt1',nfert)      ;
nload_barley_flat('cu_fine_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f14','slp1','slt1',nfert)      ;
nload_barley_flat('ds_fine_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f15','slp1','slt1',nfert)      ;
nload_barley_flat('cc_fine_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f13','slp1','slt4',nfert)      ;
nload_barley_flat('cu_fine_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f14','slp1','slt4',nfert)      ;
nload_barley_flat('ds_fine_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f15','slp1','slt4',nfert)      ;
nload_barley_flat('cc_coarse_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f13','slp1','slt9',nfert)      ;
nload_barley_flat('cu_coarse_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f14','slp1','slt9',nfert)      ;
nload_barley_flat('ds_coarse_sand_data', nfert, 'N_load_per_ha') =  NloadIC('f15','slp1','slt9',nfert)      ;
nload_barley_flat('cc_coarse_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f13','slp1','slt16',nfert)      ;
nload_barley_flat('cu_coarse_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f14','slp1','slt16',nfert)      ;
nload_barley_flat('ds_coarse_clay_data', nfert, 'N_load_per_ha') =  NloadIC('f15','slp1','slt16',nfert)      ;


*COMMANDS FOR ACTUALLY DRAWING THE GRAPHS IN GNUPLOT
*and setting the graphs with titles-------------------------------------

*$setglobal gp_title   'Figure 1: Effect of slope and tillage measures on total P load on claye soils'
```

```
*$libinclude gnuplotxyz P_12_HsS_barley slope totP

*$setglobal gp_title   'Figure 2: Effect of slope and tillage measures on total P load on sandy soils'
*$libinclude gnuplotxyz P_12_KHt_barley slope totP

*$setglobal gp_style 'lines'
*$setglobal gp_title   'Figure 6: Predicted effect of slope, soil and tillage on TP load on main soil types with PAC12 and 15kg P fert'
*$libinclude gnuplotxyz P_barley_soils_pred slope totP

*$setglobal gp_title   'Figure 7: Predicted effect of slope, soil and tillage on TP load on main soil types with Pst12 and 15kg P fert'
*$libinclude gnuplotxyz P_barley_soils_pred_slp slope totP

*$setglobal gp_title   'Figure 8: Predicted effect of slope, soil and tillage on TP load on main soil types under 15kg P fert'
*$libinclude gnuplotxyz P_barley_psoils_pred soilPst totP

*$setglobal gp_title   'Figure 6: Predicted effect of fertilisation on the total nitrogen load of barley cultivation on steep slopes'
*$libinclude gnuplotxyz Nbarley_slp5_est Nfertilisation totN

*$setglobal gp_title   'Figure 7: Predicted effect of fertilisation on the total nitrogen load of barley cultivation on flat slopes'
*$libinclude gnuplotxyz Nbarley_slp1_est Nfertilisation totN

*$setglobal gp_style 'linespoints'
*$setglobal gp_title   'Barley: Predicted effect of slope and tillage on runoff on different soils'
*$libinclude gnuplotxyz runoff_barley slope run_off

*$setglobal gp_title   'Barley: Predicted effect of slope and tillage on erosion on different soils'
*$libinclude gnuplotxyz erosion_barley slope soil_loss

*$setglobal gp_title   'Figure 9: Predicted effect of slope on erosion on different soils under barley cultivation'
*$libinclude gnuplotxyz erosion_barley_cc slope soil_loss

*$setglobal gp_style 'points'
*$setglobal gp_title   'Barley: Data of slope and tillage on erosion on different soils'
*$libinclude gnuplotxyz erosion_barley_data slope soil_loss

*$setglobal gp_style 'linespoints'
*$setglobal gp_title   'Figure 6: Predicted effect of fertilization and tillage on annual nitrogen load for barley'
*$libinclude gnuplotxyz nload_barley_flat Nfertilisation N_load_per_ha

*$setglobal gp_style 'linespoints'
*$setglobal gp_title   'Figure 2: Predicted effect of slope, soil and tillage on TP load on main soil types '
*$libinclude gnuplotxyz TPload_barley_cc_slp1 soilP totP

*$setglobal gp_style 'linespoints'
*$setglobal gp_title   'Erosion from fallow on the main soil classes'
```

```
*$libinclude gnuplotxyz erosion_fallow slope soil_loss

*Parameters exported to load_Aurajoki_heterogen
display slopeclassmean, slopedata, slopepoint, runoff, sloss, TPloadIC, NloadIC, runoff_flw, sloss_flw,
TPloadIC_flw, NloadIC_flw, Ncoeff_frtREV_b, Ncoeff_frtREV_c, NloadICestm ;
display N_barley_soils_pred ;
display erosiondata ;

*display Nbarley_slp1_est,  Nbarley_slp5_est, TPload_barley_cc_slp1  ;
```

Nutrient load parameters –summary and comparisons

```
$Title Nutrient load parameters III
$ontext
METAMODEL FOR N AND P LOAD -combining land use data with load parameters Since ICECREAM results have not been calculated for all land use data types, some data classes in economic abatement model would not have load parameters.
 PARAMETERS DO NOT COVER ALL THE
A) SLOPES (LOAD AS A FUNCTION OF SLOPE, USING THIS LOAD FUNCTION SO THAT CLASSIFICATION CAN DIFFER FROM THE ORIGINAL ICECREAM CLASSES)
B) SOILS  (USE THE CLOSEST SOIL TYPE OP ICEREAM)
C) CROPS  (USE THE DIFFERENCE BETWEEN CROP TYPES, BUT MODIFY SO THAT CEREALS DO NOT HAVE WEIRD LOAD DIFFERENCES)
This file is the third step in creating load functions that cover all data types specified in the economic model.
This file imports the land use, slope, soil and soiP distribuition data of the study watershed and combines it with previously estimated load parameters. Then the outcome is compared with the other model results for the watershed and graphs are drawn.
Economic abatment model should import the load parameters from this file, NOT the previous two files.  --> THIS FILE SHOULD HAVE GDX=Aura_load_parameters
Land use data is result of ArcGIS analyses which have been exported to Access database.
$offtext

* Data import MS-ACCESS MDB databse-------------------------------------------
* Requires mdb2gms  (http://www.gams.com/dd/docs/tools/mdb2gms.pdf)
* This is multiquery batch version
$onecho > cmd.txt
I="D:\GIS\SAMA_database2003_INPUT.mdb"
X=Aurajoki_load.gdx
Q1=select j from setj order by ID
s1=j
Q2=select flw from setflw
s2=flw
Q3=select slp from 6slopes
s3=slp
Q4=select soil from setsoil
s4=slt
Q6=select Pst from set_Aura_Pst
s6=Pst
Q7=select crop_id from crop
s7=croptype
Q8=select PAC from setPAC
s8=pac
Q9=select crop_cat from land_cat_distinct
s9=cropcode
Q10=select Pfert from setPfert
s10=pfert
Q11=select Nfert from setNfert
s11=nfert
Q16=select slt from soil_set
s16=soil
```

```
Q17=select setPclass from setPf
s17=pfclass
Q5=select j from setj where k=1 order by ID
s5=k1
Q18=select j from setj where k=2 order by ID
s18=k2
Q19=select j from setj where k=3 order by ID
s19=k3
*Q20=select j, slp, soil, PAC, Pfert, Runoff from Aura_Pload_param
*p20=runoffdata
*Q21=select j, slp, soil, PAC, Pfert, Soil_loss from Aura_Pload_param
*p21=slossdata
Q25=select Pst, Pst_no from set_Aura_Pst
p25=Pst_no
Q28=select j, slp, sltclass, Pst, Area from AuraCropFarmCropDistr
p28=crop_distr
Q29=select flw, slp, sltclass, Pst, Area from AuraCropFarmFallowDistr
p29=fallow_distr
Q30=select j, slt, nmax from Nsublimits
p30=nmax
Q31=select j, pfclass, upperPlimit2007 from Prec2007
p31=upperPlimit2007
Q32=select slt, pfclass, Class_lower_limit from Pfertility
p32=PfclassLow
$offecho
*$call =gdxviewer Aurajoki_load.gdx
$call =mdb2gms \atcmd.txt

sets
*These sets defined in Access
      j      potentially farmed crops
      flw    fallow types
      slp    slopes classes 1 to 6 (mean of each class in GIS)
      slt    soil types 1 to 20   (GIS data)
      soil   back to 4 soil classes (HsS HtS HHt KHt)   (ICECREAM results)
      Pst    phosphorus soil status set (GIS data)
      pfclass phosphorus fertility class set
      PAC    phosphorus soil status (ICECREAM results)
      pfert  P fertilisation levels
      nfert  P fertilisation levels

      lc     load parameter set
/
      r runoff
      e erosion
      o omega leaching based on different technology
      v delta leaching based on different technology
      nic nitrogen load coefficient
      psini initial phosphorus level
      PStm1 dummy for last period P level
/
*subsets for tillage types
```

```
        k1(j) conventional tillage crops
        k2(j) cultivator tillage crops
        k3(j) direct tillage crops

        cropcode crop types including fallow codes with TIKE numbers
;

parameters
        Pst_no(Pst)                 Variable soil P mg per l
        N_frtlzn(nfert)             Fertilisation quantities in ICECREAM results
        crop_distr(j,slp,slt,Pst)      Crop types on different soil slope and P status land
        crop_distr_no0(j,slp,slt)      Land distribution between crops and slopes and soil types (no soil P class)
        fallow_distr(flw,slp,slt,Pst)   Fallow types on different soil slope and P status land
        runoff(j,slp,slt)            Crop Runoff in mm ha-1     (OLS results from ICECREAM)
        sloss(j,slp,slt)             Crop Soil loss in kg ha-1  (OLS results from ICECREAM)
        runoff_flw(flw,slp,slt)        Fallow Runoff in mm ha-1     (OLS results from ICECREAM)
        sloss_flw(flw,slp,slt)         Fallow Soil loss in kg ha-1  (OLS results from ICECREAM)
        runoffdata(j,slp,soil)         Runoff in mm ha-1      (original ICECREAM values with different slp classes)
        erosiondata(j,slp,soil)        Soil loss in kg ha-1   (original ICECREAM values with different slp classes)
        runoff__flw(flw,slp,soil)      Fallow Runoff mm ha-1  (original ICECREAM values with different slp
classes)
        sloss__flw (flw,slp,soil)      Fallow Erosion kg ha-1 (original ICECREAM values with different slp classes)
        Nloaddata(j,slp,soil,nfert)    Original ICECREAM values with different slp classes
        TPloadIC(j,slp,slt,Pst,pfert)   P load for crops (original ICECREAM data classes)
        TPloadIC_flw(flw,slp,slt,Pst,pfert) P load for fallow (original ICECREAM data classes)
        NloadIC(j,slp,slt,nfert)       N load for crops (original ICECREAM data classes)
        NloadIC_flw(flw,slp,slt,nfert)  N load for fallow (original ICECREAM data classes)
*Mean values calculated over the slope class for Hts soil and all crops (Original IC or OLS???)
        meanRunoffCrops(j)            Mean runoff for different crop types
        meanErosionCrops(j)           Mean erosion for different crop types
        meanRunoffFall(flw)           Mean runoff for different fallow types
        meanErosionFall(flw)          Mean erosion for different fallow types
        meanHtSRunoffCrops(j)          Mean runoff from each crop at Hts soil (average soil) 1\% slope
        meanHtSErosionCrops(j)         Mean erosion from each crop at Hts soil (average soil) 1\% slope
        meanHtSErosion               Mean erosion at Hts soil (average soil) 1\% slope
        meanHtSRunoff                Mean runoff at Hts soil (average soil) 1\% slope
        meanHtSRunoffFall(flw)         Mean erosion from each fallow type at Hts soil (average soil) 1\% slope
        meanHtSErosionFall(flw)        Mean erosion from each fallow type at Hts soil (average soil) 1\% slope
        finnsoilshare(soil)           Share of each Finnish soil class (4) of total watershed area
        soilshare(slt)               Share of each soil type of total arable land given original 20 FAO soil types
        slopeClassMean(slp)           Mean of the slope class  (in GIS data)
        slopeClassUpLim(slp)          Upper boundary of the slope class  (in GIS data)
        Ncoeff_frtREV_b(j,slp,slt)      Exp2 load function coefficient b for N
        Ncoeff_frtREV_c(j,slp,slt)      Exp2 load function coefficient c for N
        meanNcoeff_frt_b(j)           Nitrogen load function coefficient estimated directly from ICECREAM
(mean slope and soil)
        meanNcoeff_frt_c(j)           Nitrogen load function coefficient estimated directly from ICECREAM
(mean slope and soil)
        NloadICestm(j,slp,slt,nfert)    Nitrogen load estimated from ICECREAM   (for all ICECREAM fertilisation
levels)
        nmax(j,slt)                  Recommended base fertilisation level as baseline N-load level
        upperPlimit2007(j, pfclass)     Upper limit for allowed P fertilisation for given P-fertility class
```

```
        PfclassLow(slt, pfclass)       Lower bound for Pst status of given P-fertility class
        pmax(Pst,j,slt)                Recommended base fertilisation level as baseline P-load level
        upperPlimit2007slt(j, pfclass, slt) Soil class dependent upper limit for allowed P fertilisation for given P-
fertility class
        Pfcropmax(j, slt, pfclass)      Maximum P fertilisation
        totalCropPk1(k1,slp,slt,Pst)    Crop area for conventional tillage
        totalCropPk2(k2,slp,slt,Pst)    Crop area for cultivation tillage
        totalCropPk3(k3,slp,slt,Pst)    Crop area for direct tillage
        totalCropP_k1(cropcode,slp,slt,Pst)  Baseline total P load that can be updated to GIS crop data given
conventional tillage
        totalCropP_k2(cropcode,slp,slt,Pst)                                  given cultivation tillage
        totalCropP_k3(cropcode,slp,slt,Pst)                                  given no-till
        pmaxTPperha(cropcode,slp,slt,Pst)    Max P fert P load with conventional cultivated crops and existing
fallow dist
*Scaling test parameters
        erosiondata_rel_f13slp0HsS(j,slp,soil)
        erosion_brl_slp0_HsS
*For mean
        Ecoeff_slp_b(j,soil)          Coefficient b for mean erosion (over slope classes) for each soil type  (crops)
        Ecoeff_slp_c(j,soil)          Coefficient c for mean erosion (over slope classes) for each soil type  (crops)
        Rcoeff_slp_b(j,soil)          Coefficient b for mean runoff (over slope classes) for each soil type   (crops)
        Rcoeff_slp_c(j,soil)          Coefficient c for mean runoff (over slope classes) for each soil type   (crops)
        Ecoeff_slp_b_flw(flw,soil)     Coefficient b for mean erosion (over slope classes) for each soil type
(fallow)
        Ecoeff_slp_c_flw(flw,soil)     Coefficient c for mean erosion (over slope classes) for each soil type
(fallow)
        Rcoeff_slp_b_flw(flw,soil)     Coefficient b for mean runoff (over slope classes) for each soil type
(fallow)
        Rcoeff_slp_c_flw(flw,soil)     Coefficient c for mean runoff (over slope classes) for each soil type
(fallow)
        meanNload_flw(flw,soil)        Mean N load (over slope classes) from each soil type (fallow)
        mean_N_coeff_frt_b(j,soil)     Coefficient b for mean Nload (over slope classes) for each soil type
(crop)
        mean_N_coeff_frt_c(j,soil)     Coefficient c for mean Nload (over slope classes) for each soil type
(crop)

;

*GET THE DATA FROM THE GDX FILE CREATED FROM ACCESS DATABASE
$gdxin Aurajoki_load.gdx
$load j k1 k2 k3 cropcode flw slp slt Pst pfert nfert crop_distr fallow_distr Pst_no soil nmax
$load pfclass upperPlimit2007 PfclassLow

*GET THE EXTRAPOLATED PARAMETER DATA
$gdxin Aura_load_prams.gdx
$load runoff sloss TPloadIC NloadIC slopeclassmean
$load runoff_flw sloss_flw TPloadIC_flw NloadIC_flw
$load Ecoeff_slp_b Ecoeff_slp_c Rcoeff_slp_b Rcoeff_slp_c Ecoeff_slp_b_flw Ecoeff_slp_c_flw
Rcoeff_slp_b_flw Rcoeff_slp_c_flw
$load Ncoeff_frtREV_b, Ncoeff_frtREV_c NloadICestm N_frtlzn
$load mean_N_coeff_frt_b mean_N_coeff_frt_c meanNload_flw
```

```
*crop_distr_no0(j,slp,slt)$(crop_distr(j,slp,slt,"Pst18") = 0 AND crop_distr(j,slp,slt,"Pst12") = 0 AND
*crop_distr(j,slp,slt,"Pst11") = 0  AND crop_distr(j,slp,slt,"Pst9") = 0  ) = 1 ;

*GET THE ORIGINAL LOAD PARAMETERS  - NOTE THAT THESE ARE IN DIFFERENT SLOPE DIMENSIONS
$gdxin Aura_load_OLS
$load runoffdata erosiondata Nloaddata runoff__flw sloss__flw erosiondata_rel_f13slp0HsS
erosion_brl_slp0_HsS

*BARE FALLOW TAKEN FROM THE RELATIVE DIFFERENCE IN THE OLDER ICECREAM RUNS
set bare(flw) / bf1s, bf2s, bf3s, bf1n, bf2n, bf3n  /  ;

parameters
*Land distribution
        land_share(j,slp,slt)        Share of land area for each crop on each soil with given slope
        cropLandbySlope(slp)         Crop land in each slope category
        cropLandbySoil(slt)          Crop land in each soil category
        cropLandbyPst(Pst)           Crop land in each soil P category
        crop_distr18(j, slp, slt)     Crop distribution soil P = 18 class
        crop_distr_f13(slp, slt, Pst)  Distribution of conventionally cultivated barley between slope and soil
and soil P classes
        crop_dist_noP(j, slp, slt)    Distribution of crop land  with no soil P status
        crop_dist_no_slp(j, slt)      Distribution of crop land  with no soil P status or slope
        fall_dist_noP(flw, slp, slt)   Distribution of fallow land  with no soil P status
        fallowLandbySlope(slp)        Fallow land in each slope category
        fallowLandbySoil(slt)         Fallow land in each soil category
        fallowLandbyPst(Pst)          Fallow land in each soil P category
        arableLandbySlope(slp)        Crop + Fallow land in each slope category
        arableLandbySoil(slt)         Crop + Fallow land in each soil category
        arableLandbyPst(Pst)          Crop + Fallow land in each soil P category
        cropLandbySlopeShare(slp)     Share of crop land of total land of each slope class
        cropLandbySoilShare(slt)      Share of crop land of total land of each soil class
        fallowLandbySlopeShare(slp)    Share of fallow land of total land of each slope class
        fallowLandbySoilShare(slt)     Share of fallow land of total land of each soil class
        slopeShare(slp)              Share of different slope classes of total modeled area
        slopeAreaCrop(slp)           Ha of different slope classes of crop area
        slopeAreaFallow(slp)          Ha of different slope classes of fallow area
        slopeShareCrop(slp)           Share of different slope classes of crop area
        slopeShareFallow(slp)         Share of different slope classes of fallow area
        grassLandSharebySlope(slp)     Share of grass land (for production) of each slope class
        grassLandSharebySoil(slt)      Share of grass land (for production) of each soil class
        grassLandSharebyPst(Pst)       Share of grass land (for production) of each soil P class
        fallsoilshare(slt)           Share of fallow over different soils of total fallow area
        cropshare(j)                 Share of each crop from the total arable area summed over soil and slope
categories
        cropshare_cropland(j)         Share of each crop from the total CROP area summed over soil and slope
categories
        fallowshare(flw)             Share of each fallow from the total area summed over soil and slope
categories
        soilPstShareCrop(pst)         Share of each pst class from the total area summed over crop and soil and
slope categories
        soilPstShareFall(pst)         Share of pst class from the total area  (crop land)
        soilPstShare(pst)            Share of pst class from the total area  (fallow land)
```

*Other load parameters
ICECREAM_PP(j,slp,slt,Pst) PP load of ICECREAM for all the
ICECREAM_DRP(j,slp,slt,Pst) DRP load factors from ICECREAMC
Nload_share1to1(j) Share of N load assuming equal proportion as area
Pload_share1to1(j) Share of P load assuming equal proportion as area
landout_share Share of land which is not modeled
rel_N(j) Relative N coefficients (if fallow equals 1)
loadcoef_N(j,slp,slt) N kg per ha for each plant
loadcoeff_N N kg per ha for fallow
loadcoef_DRP(j,slp,slt) DRP kg for each plant
loadcoeff_DRP DRP kg for fallow
loadcoef_PP(j,slp,slt) PP each plant (kg per ha)
loadcoeff_PP PP for fallow (kg per ha)
runoff6(j,slt,Pst) Runoff parameters from the steepest slope class
PStm(j,slp,slt,Pst) P soil status (ammonium acetate soluable)
DRP_loadparam(j,slp,slt,Pst) DRP load parameters to be estimated
PP_loadparam(j,slp,slt,Pst) PP load parameters to be estimated
DRP_loadparam_flw(flw,slp,slt,Pst) DRP load parameters to be estimated
PP_loadparam_flw(flw,slp,slt,Pst) PP load parameters to be estimated
DRP_loadMYTVAS(j,slp,slt,Pst) DRP load parameters to be estimated
PP_loadMYTVAS(j,slp,slt,Pst) PP load parameters to be estimated
runofftest(j,slp,slt) Test version of runoff
DRP_test(j,slp,slt,Pst) Test version DRP load
DRPSUMtest Test version DRP load
sumDRPbyslopes(slp) Sum of DRP for each slope class
pmaxtest(j,slt) Parameter for testing load from max P fertilisation
testrunoffsum(j) SUM of runoff for crops
TcropP_test(j,slp,slt,Pst) Check if matches witht the TP load
NloadIC80(j,slp,slt) Nitrogen load if 80kg ha-1 for plants
DRP_HtS_pram(j) Test parameter for DRP load from HtS soil
PP_HtS_pram(j) Test parameter for PP load from HtS soil
cropNeach(j,slp,slt) N load from each crop (given soil and slope class)
cropNeachperha(j,slp,slt) N load per crop ha given the maximum env.sub fertilization
fallNrevised(flw) N load from falllow (revised version)
cropN_soils(slt) N load from crop area between different soils
cropN_slopes(slp) N load from crop area between different slopes
cropN_crops(j) N load from crop area between different crops
DRP_meanload(j) Average DRP load from crop types
DRP_meanload_flw(flw) Average DRP load from crop types
PP_meanload(j) Average PP load from crop types
PP_meanload_flw(flw) Average PP load from fallow types
mytvasPfertAvg(j) Average P fertilisation to each crop according to the MYTVAS study for the watershed
mytvasNfertAvg(j) Average N fertilisation to each crop according to the MYTVAS study for the watershed
meanNloadIC_flw(flw) Mean N load from different fallow types (ICECREAM based)
N_LoadbySlpMYTVAS(slp) N load from each slope given MYTVAS fertilisation
N_LoadbySltMYTVAS(slt) N load from each soil given MYTVAS fertilisation
totalCropP(j,slp,slt,Pst) DRP + PP for crops
totalFallP(flw,slp,slt,Pst) DRP + PP for fallow
PP_load15kg(j,slp,slt,Pst) PP load when using 15kg of P fertilisation
DRP_load15kg(j,slp,slt,Pst) DRP load when using 15kg of P fertilisation

MeanSlope_CropErosion(j,soil) Crop erosion for mean slope
MeanSlope_CropRunoff(j,soil)   Crop runoff for mean slope
MeanSlope_FallErosion(flw,soil) Fallow erosion for mean slope
MeanSlope_FallRunoff(flw,soil) Fallow runoff for mean slope
meanNcoeff_frt_b2(j)   Mean N load function coeffcient b (over slope and soil)
meanNcoeff_frt_c2(j)   Mean N load function coeffcient c (over slope and soil)
MeanSlope_Ncoeff_b(j,slt)   Mean N load function coeffcient b (over slope)
MeanSlope_Ncoeff_c(j,slt)   Mean N load function coeffcient c (over slope)
mean_Ncoeff_b1(j)   Mean N load function coeffcient b (over slope and soil)
mean_Ncoeff_c1(j)   Mean N load function coeffcient c (over slope and soil)
loadCoeff_N_flw(flw,slp,slt)   Nitrogen load function parameter for fallow
Ncoeff_frt_b(j,slp,slt)   Nitrogen load function coefficient estimated directly from ICECREAM
Ncoeff_frt_c(j,slp,slt)   Nitrogen load function coefficient estimated directly from ICECREAM
rnoff(j,slp,slt)   Runoff in mm ha-1
erosion(j,slp,slt)   Soil loss in kg ha-1
rnoff_flw(flw,slp,slt)   Runoff of fallow (4 soil classes)
erosion_flw(flw,slp,slt)   Erosion of fallow   (4 soil classes)
loadFSCoeff_N(j,slp,soil)   N coeff crops
loadFSCoeff_N_flw(flw,slp,soil) N coeff fallow
meanNcoeff(j)   Mean N coeff
meanNcoeff_flw(flw)   Mean N coeff (fallow)
meanNcoeff_old(j)   The N-load values for older smaller crop set
meanRunoffCrops_old(j) The Runoff values for older smaller crop set
meanErosionCrops_old(j) The Erosion values for older smaller crop set
* loadCoeff_N(j,slp,slt)
* meanNcoeff_frt_b(j,slp,slt) Nitrogen load function coefficient estimated directly from ICECREAM
* meanNcoeff_frt_c(j,slp,slt) Nitrogen load function coefficient estimated directly from ICECREAM
;
scalars
*   land_ha   total agricultural land (IN VEPS)   / /
veps_ha   Total agricultural land in VEPS 2002   / 32784 /
model_ha   Total modeled area including fallow
totX   Total modeled crop area calculated from the crop_dist_noP
crop_ha   Total modeled crop land
fallow_ha   Total modeled fallow land
totalCropShare   Share of crop land of total modeled land
grass_land_share   Share of grass (for production) of total modeled crop land
totalFallowShare   Share of fallow land of total modeled land
noLoadLand   Area not covered by the model
noLoadLand_pr   Area not covered by the model as percent of total arable land
meanWatershedSlope Calculated from arcgis slope map with zonal statistics mean
meanSlope   Mean slope of the farm land
meanSlopeCrop   Mean slope for the arable land calculated from the slope classes and their shares
meanSlopeFallow   Mean slope for the arable land calculated from the slope classes and their shares
meanSlopeArable   Mean slope for the arable land calculated from the slope classes and their shares from total crop and fallow area
meanErosion   Mean erosion
meanRunoff   Mean runoff
cropMeanErosion   Mean erosion ha-1 for the watershed crops (average over the years)
fallowMeanErosion Mean erosion ha-1 for the watershed fallow (average over the years)
arableMeanErosion Mean erosion ha-1 for the watershed arable area (including both fallow and crops) (average over the years)

cropMeanRunoff Mean runoff from crops
fallowMeanRunoff Mean runoff from fallow
arableMeanRunoff Mean runoff of all farm land
totalErosion Total erosion from the watershed farmland (average over the years)
totalErosion_perha Average erosion from the watershed farmland (average over the years and land)
*Other modelled loads for the watershed (VEPS,VEMALA, VIHMA)
veps_N_2002 Kg N in VEPS 2002 average weather /415135 /
veps_Nperha_2002 Kg N per ha from VEPS
veps_TP_2002 Kg TP VEPS 2002 average weather / 31119 /
veps_TPperha_2002 Kg TP per ha
VEMALA_TP_2003 TP load for watershed 2003 VEMALA simulation result excel sheet Kuormituslukuja_eri_malleista / 18670 /
VEMALA_TP_2002 TP load for watershed 2002 VEMALA simulation result excel sheet Kuormituslukuja_eri_malleista / 21751 /
VEMALA_TP_2008 TP load for watershed 2008 VEMALA simulation result excel sheet Kuormituslukuja_eri_malleista / 62121 /
VEMALA_TP_2009 TP load for watershed 2009 VEMALA simulation result excel sheet Kuormituslukuja_eri_malleista / 14811 /
VEMALA_N_2003 N load for watershed 2003 all sources excel sheet Kuormituslukuja_eri_malleista / 329000 /
VEMALA_N_2002 N load for watershed 2002 all sources excel sheet Kuormituslukuja_eri_malleista / 389000 /
VEMALA_N_2004 N load for watershed 2004 all sources excel sheet Kuormituslukuja_eri_malleista / 661000 /
VEMALA_N_2009 N load for watershed 2009 all sources excel sheet Kuormituslukuja_eri_malleista / 273000 /
VEMALA_N_AVG N load for watershed mean load between 1997-2006 email from S Tattari / 622600 /
VEMALA_P_AVG P load for watershed mean load between 1997-2006 email from S Tattari / 37408 /
VIHMA_N N load for watershed in VIHMA model / 514700 /
VIHMA_P P load for watershed in VIHMA model / 40925 /
*Nitrogen
TotalN_MYTVAS_Hi Total N load given functions estimated from ICECREAM and fertilisation given by upper end of MYTVAS study
TotalN_MYTVAS_Lo Total N load given functions estimated from ICECREAM and fertilisation given by lower end of MYTVAS study
cropN Nitrogen load from crop land
fallN Nitrogen load from fallow land
baseN_load Total Nitrogen load of area
TotalN_old Total nitrogen load of modeled arable area
TotalNperha_old Total nitrogen load per ha of modeled arable area
TotalNrevised Total annual N load kg given the maximum allowed fertilisation use and ICECREAM estimated functions
TotalNrevisedPerHa Total annual N load kg per ha
cropNrevised Total annual N load kg ha-1 for all crop area (revised from ICECREAM) MAX ENV SUB fertilization
cropNamx Total annual N load kg ha-1 for all crop area (revised from ICECREAM) MYVASS average N fertilization
cropNmeanBase Total annual N load kg ha-1 for all crop area (calculated from mean ICECREAM soil and slope parameters)

cropNmeanBase2 Total annual N load kg ha-1 for all crop area (calculated from mean ICECREAM soil and slope parameters estimated with same OLS as hetgen
cropN_sensitivity Total annual N load kg ha-1 sensitivity with half N fertilisation
cropN_mytvasHi Total annual N load kg ha-1 given MYTVAS P fertilisation
cropN_mytvasLo Total annual N load kg ha-1 given MYTVAS P fertilisation
fallmeanN Total annual N load kg
fallmeanN_perha Total annual N load kg ha-1
TotalN_mean Total N load per ha from mean parameters and MYTVAS fertilization and 2003 crop distribution
TotalN_mean2 Total N load per ha from mean parameters (b1 and c1) and MYTVAS fertilization and 2003 crop distribution
cropNsum Sum of annual N load from crop land given revised nload functions
cropNsumperha Average annual N load from crop land
ccbarleyNload N load from conventionally cultivated barley
ccbarleyNload_ha N load from conventionally cultivated barley per ha
TotNPerHa_Aura N load per ha
*Phosphorus
cropP_mytvasHi Total annual TP load kg ha-1 given MYTVAS P fertilisation
bio_coef share of PP available for algae / 0.16 /
Fall_DRP ICECREAM coefficient for fallow DRP / 0.60 /
Fall_PP ICECREAM coefficient for fallow PP / 0.04 /
DRP_TP_share Share of DRP from TP / 0.33 /
basePP The old shares applied to VEPS loads to produce baseline PP
baseDRP The old shares applied to VEPS loads to produce baseline DRP
modTP Modeled TP (PP converted back to non algae available form by coefficient)
modTPperha Modeled TP (PP converted back to non algae available form by coefficient)
mod_DRP_TP_share Modeled share of DRP from TP
DRP_VEPS share of the DRP in VEPS
TotalDRP Total DRP load from modelled area (extrapolation of ICECREAM runoff)
TotalPP Total PP load from modelled area (extrapolation of ICECREAM erosion)
DRPtoPPratio Ratio of DRP and PP at the model area
TotalP_IC Total P load of arable area (extrapolation of ICECREAM TP values constant P fert 15kg per ha)
TotalPperhaIC Total P load of arable area (per ha)
TcropP Total P load of modeled crop land
TfallP Total P load of modeled fallow land
TcropPperha Total P load per ha of modeled crop land
cropDRP SAMA modeled amount of DRP from crop land
cropDRPmeanland SAMA modeled amount of DRP from crop land
cropDRPperha SAMA modeled amount of DRP from crop land (per ha)
fallDRP SAMA modeled amount of DRP from fallow land
fallDRPmeanland SAMA modeled amount of DRP from fallow land
sumDRPt SAMA modeled amount of DRP in tonnes of P
cropPP SAMA modeled amount of PP annually from the research area
cropPPmeanland SAMA modeled amount of PP annually from the research area
fallPP SAMA modeled amount of PP annually from the research area
fallPPmeanland SAMA modeled amount of PP annually from the research area
modBioP modeled total algae available P of the research area
MYTVASmodBioP modeled algae available P of the research area (MYTVAS fertilisation)
MYTVASmodTP modeled TP (PP converted back to non algae available form by coefficient)
MYTVAScropPP PP from crop (MYTVAS fertilisation)
MYTVAScropDRP DRP from crop (MYTVAS fertilisation)

```
      MYTVAStotalDRP         DRPP from total farm land    (MYTVAS fertilisation)
      MYTVAStotalPP          PP from total farm land      (MYTVAS fertilisation)
      PsAvg                  Average soil P mg per l
      PsAvgcrop              Average soil P mg per l of the crop area given the reference year
      PsAvgfall              Average soil P mg per l of the fallow area given the reference year
      cropPPmeanland         Total PP from crop with mean land
      fallPPmeanland         Total PP from fallow with mean land
      meanlandTotDRP         Total DRP from mean farm land
      meanlandTotPP          Total PP from mean farm land
      meanlandmodTP          Total TP model with mean land
      cropDRPmeanload        Crop DRP from mean land
      fallDRPmeanload        Fallow DRP from mean land
      Nout1to1               N load which is not modeled when assuming 1:1
      Pout1to1               N load which is not modeled when assuming 1:1
      baseN_load             N load which is modeled
      baseP_load             P load which is modeled
      check15kgModTP         Total TP when 15 kg P fertilisation
      check15kgTotalDRP      Total DRP when 15 kg P fertilisation
      check15kgTotalPP       Total PP when 15 kg P fertilisation
      check15kgCropDRP       Crop DRP when 15 kg P fertilisation
      check15kgCropPP        Crop PP when 15 kg P fertilisation
      metaOverIC_TP          check15kgModTP divided by TotalP_IC
;
*meanwatershed slope from arcgis zonal statistics (use the watershed polygon ID as single zone)  NOTE NOT THE MEAN FIELD SLOPE
 meanWatershedSlope = 1.055143        ;
*ArcGIS zonal statistics AuraMeanFieldSlope (parcel data model builder script)
 meanSlope = 1.42   ;

parameters
 mytvasPfertHi(j)  P fertilization based on interviews of Savijoki farmers   (highest amounts)
 /
*winter wheat  (SAME USED AS FOR SPRING WHEAT)
 f1*f3        15
*spring wheat
 f4*f6        15
*other cereal
 f7*f15       15
 f16*f18      17
*oats
 f19*f21      21
*NO MYTVAS VALUE FOR PEAS, use same as limit for cereals
 f25*f27      15
 f28*f33      27
*sugarbeet MYTVAS value for Löytäneenoja p.45
 f34*f36      14
*NO MYTVAS VALUE FOR RAPE AT SAVIJOKI, USE YLÄNEENJOKI INSTEAD
 f37*f42      21
*FROM YLÄNEENJOKI INSTEAD
 f43*f45      14
 /
 mytvasPfertLo(j)  P fertilization based on interviews of Savijoki farmers   (lowest amounts)
```

```
 /
*winter wheat  (SAME USED AS FOR SPRING WHEAT)
 f1*f3          12
*spring wheat
 f4*f6          12
 f7*f15          12
*malt barley
 f16*f18        14
*oats
 f19*f21        16
*NO MYTVAS VALUE FOR PEAS, use same as limit for cereals
 f25*f27        12
 f28*f33        27
*sugarbeet MYTVAS value for Löytäneenoja p.45
 f34*f36        9
*NO MYTVAS VALUE FOR RAPE AT SAVIJOKI, USE YLÄNEENJOKI INSTEAD
 f37*f42        14
*FROM YLÄNEENJOKI INSTEAD
 f43*f45        14
 /

 mytvasNfertHi(j) N fertilization based on interviews of Savijoki farmers  (highest amounts)
 /
*syysvehnä  (SAME USES AS FOR SPRING WHEAT)
 f1*f3          140
*kevätvehnä
 f4*f6          110
 f7*f24          110
*NO MYTVAS VALUE FOR PEAS, use nmax level
 f25*f27         110
*potato
 f28*f33         80
*Sugarbeet
 f34*f36         125
*rapeseed
 f37*f42         120
 f43*f45        170
 /
 mytvasNfertLo(j) N fertilization based on interviews of Savijoki farmers (lowest amounts)
 /
* MYTVA2 p.43
 f1*f24          80
*NO MYTVAS VALUE FOR PEAS, use nmax level
 f25*f27         50
*potato      MYTVA2 p.43
 f28*f33         60
*Sugarbeet    MYTVA2 p.43
 f34*f36         100
*rapeseed   MYTVA2 p.43
 f37*f42         80
 f43*f45        80
 /
```

```
;

*mytvasPfertAvg(j) = mytvasPfertHi(j) + mytvasPfertLo(j)     ;
mytvasNfertAvg(j) = (mytvasNfertHi(j) + mytvasNfertLo(j))/2     ;

*ALLOCATE THE BASELINE FERTILISATION USE BY THE MAXIMUM ALLOWED LEVELS FOR EACH P-STATUS,
SOIL AND CROP COMBINATION
LOOP (pst,
 pmax(pst,j,slt)$( Pst_no(pst) \textless PfclassLow(slt, 'pfcls2') ) = upperPlimit2007(j, 'pfcls1') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls2') ) = upperPlimit2007(j, 'pfcls2') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls3') ) = upperPlimit2007(j, 'pfcls3') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls4') ) = upperPlimit2007(j, 'pfcls4') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls5') ) = upperPlimit2007(j, 'pfcls5') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls6') ) = upperPlimit2007(j, 'pfcls6') ;
 pmax(pst,j,slt)$( Pst_no(pst) ge PfclassLow(slt, 'pfcls7') ) = upperPlimit2007(j, 'pfcls7') ;
  ) ;

*Hence use 1\% for slope class which from the data is slp3, use HtS soil as proxy because between
extremes
*meanHtSErosionCrops(j) = erosiondata(j,'slp1','HtS')       ;
*meanHtSRunoffCrops(j)  = runoffdata(j,'slp1','HtS')       ;
*meanHtSRunoffFall(flw)  = runoff__flw(flw,"slp1","HtS")            ;
*meanHtSErosionFall(flw) = sloss__flw(flw,"slp1","HtS")             ;

 display crop_distr, fallow_distr, runoff_flw, sloss_flw, TPloadIC_flw, runoff, sloss, TPloadIC, NloadIC,
NloadIC_flw   ;

*Using VEPS results as a bench mark
 baseN_load  =  veps_N_2002 ;
 veps_Nperha_2002 = veps_N_2002/veps_ha ;

*CONVERTING CROP AREA DATA FROM M2 TO HA
 crop_distr(j, slp, slt, Pst) =  crop_distr(j, slp, slt, Pst) /10000       ;
 fallow_distr(flw, slp, slt, Pst) = fallow_distr(flw, slp, slt, Pst)/10000 ;

* Summing up the different combinations to total area of crop and fallow land
 fallow_ha = SUM{(flw,slp,slt,Pst), fallow_distr(flw, slp, slt, Pst) }  ;
 crop_ha = SUM{(j,slp,slt,Pst), crop_distr(j, slp, slt, Pst)}  ;
 DRP_VEPS = veps_TP_2002  * DRP_TP_share ;
*vepsTPperha = veps_TP_2002/land_ha ;
*vepsNperha  = veps_N_2002/land_ha ;

*Comparing watershed GIS total field area with VEPS area
 model_ha =  crop_ha + fallow_ha ;
 noLoadLand = veps_ha - model_ha   ;
 noLoadLand_pr = noLoadLand / veps_ha * 100 ;
*Calculating the shares for each attribute combination (from the total modeled farm land)
 cropshare(j) = SUM{(slp,slt,Pst), crop_distr(j,slp,slt,Pst)}           /model_ha       ;
*Calculating the shares for each attribute combination (from the total modeled crop land)
 cropshare_cropland(j) = SUM{(slp,slt,Pst), crop_distr(j,slp,slt,Pst)}   /crop_ha        ;
*Calculating land shares of different land classes of crops, fallow and total farm land
 cropLandbySlope(slp)   = SUM{(j,slt,Pst), crop_distr(j,slp,slt,Pst)}        ;
```

```
 cropLandbySoil(slt)    = SUM{(j,slp,Pst), crop_distr(j,slp,slt,Pst)}       ;
 cropLandbyPst(Pst)     = SUM{(j,slt,slp), crop_distr(j,slp,slt,Pst)}       ;
 fallowLandbySlope(slp) = SUM{(flw,slt,Pst), fallow_distr(flw,slp,slt,Pst)}  ;
 fallowLandbySoil(slt)  = SUM{(flw,slp,Pst), fallow_distr(flw,slp,slt,Pst)}  ;
 fallowLandbyPst(Pst)   = SUM{(flw,slt,slp), fallow_distr(flw,slp,slt,Pst)}  ;
 arableLandbySlope(slp) = cropLandbySlope(slp) + fallowLandbySlope(slp)      ;
 arableLandbySoil(slt)  = cropLandbySoil(slt)  + fallowLandbySoil(slt)       ;
 arableLandbyPst(Pst)   = cropLandbyPst(Pst)   + fallowLandbyPst(Pst)        ;
*Calculating land shares of different land classes of crops, fallow and total farm land
 cropLandbySlopeShare(slp)   =  cropLandbySlope(slp) / model_ha       ;
 cropLandbySoilShare(slt)    =  cropLandbySoil(slt)  / model_ha       ;
 fallowLandbySlopeShare(slp) =  fallowLandbySlope(slp) / model_ha      ;
 fallowLandbySoilShare(slt)  =  fallowLandbySoil(slt)  / model_ha      ;
 fallowshare(flw) = SUM{(slp,slt,Pst), fallow_distr(flw,slp,slt,Pst)}    /model_ha       ;
 grass_land_share = SUM{(slp,slt,Pst),   crop_distr('f43',slp,slt,Pst)}       / model_ha       ;
 grassLandSharebySlope(slp)  = SUM{(slt,Pst),   crop_distr('f43',slp,slt,Pst)} / arableLandbySlope(slp) ;
 grassLandSharebySoil(slt)$(arableLandbySoil(slt) > 0)   = SUM{(slp,Pst),   crop_distr('f43',slp,slt,Pst)} /
arableLandbySoil(slt) ;
 grassLandSharebyPst(Pst)$(arableLandbyPst(Pst) > 0)     = SUM{(slt,slp),   crop_distr('f43',slp,slt,Pst)} /
arableLandbyPst(Pst) ;
 totalCropShare    =    crop_ha   /  model_ha  ;
 totalFallowShare  =    fallow_ha /  model_ha  ;

* meanrunoffcrops(j) = SUM{(slp,slt,Pst), runoff(j,slp,slt,Pst) * crop_distr(j,slp,slt,Pst) } / SUM{(slp,slt,Pst),
crop_distr(j,slp,slt,Pst)};
* meanerosiocrops(j) = SUM{(slp,slt,Pst), sloss(j,slp,slt) * crop_distr(j,slp,slt,Pst) } / SUM{(slp,slt,Pst),
crop_distr(j,slp,slt,Pst)}     ;

*Slope shares
 slopeshare(slp)   =  [SUM{(j,slt,Pst), crop_distr(j,slp,slt,Pst)} + SUM{(flw,slt,Pst),fallow_distr(flw,slp,slt,Pst)}]
/ model_ha  ;
 slopeAreaCrop(slp)      = SUM{(j,slt,Pst), crop_distr(j,slp,slt,Pst) }                       ;
 slopeAreaFallow(slp)    = SUM{(flw,slt,Pst), fallow_distr(flw,slp,slt,Pst)}                   ;
 slopeShareCrop(slp)     =  slopeAreaCrop(slp)   / crop_ha                                     ;
 slopeShareFallow(slp)   =  slopeAreaFallow(slp) / fallow_ha                                   ;
 meanSlopeCrop          = SUM{slp, slopeclassmean(slp) *  slopeShareCrop(slp)}                  ;
 meanSlopeFallow        = SUM{slp, slopeclassmean(slp) *  slopeShareFallow(slp)}                ;
 meanSlopeArable        =  meanSlopeCrop * totalCropShare + meanSlopeFallow * totalFallowShare    ;
*Soil shares (FAO classes)
 soilshare(slt) = [SUM{(j,slp,Pst),crop_distr(j,slp,slt,Pst)} + SUM{(flw,slp,Pst),fallow_distr(flw,slp,slt,Pst)}]
/model_ha    ;
 fallsoilshare(slt)  =  SUM{(flw,slp,Pst),fallow_distr(flw,slp,slt,Pst)}  /   fallow_ha       ;
*Soil shares (FAO classes grouped to 4 ICECREAM result classes)
 finnsoilshare('HsS') =  soilshare('slt5') +  soilshare('slt3') +  soilshare('slt4') +  soilshare('slt6')        ;
 finnsoilshare('HtS')  = soilshare('slt16')  +  soilshare('slt8')  + soilshare('slt17')  +  soilshare('slt18')  +
soilshare('slt19')  + soilshare('slt20')  ;
 finnsoilshare('HHt')   = soilshare('slt1')  + soilshare('slt2')  + soilshare('slt7')   ;
 finnsoilshare('KHt')   = soilshare('slt9')  + soilshare('slt10') + soilshare('slt11') + soilshare('slt12') +
soilshare('slt13') + soilshare('slt14') + soilshare('slt15');
* Erosion and Runoff for all crops and fallow (and 4 soil types) given the mean slope of field area
 MeanSlope_CropErosion(j,soil)   =   Ecoeff_slp_b(j,soil) * exp[meanSlope * Ecoeff_slp_c(j,soil)]   ;
 MeanSlope_CropRunoff(j,soil)    =   Rcoeff_slp_b(j,soil) * meanSlope + Rcoeff_slp_c(j,soil)      ;
```

```
 MeanSlope_FallErosion(flw,soil)   =   Ecoeff_slp_b_flw(flw,soil) * exp[meanSlope *
Ecoeff_slp_c_flw(flw,soil)]   ;
 MeanSlope_FallRunoff(flw,soil)    =   Rcoeff_slp_b_flw(flw,soil) * meanSlope + Rcoeff_slp_c_flw(flw,soil)
;
* Nitrogen load function coefficients given the slope of field area in each slope class (and soil class)
 MeanSlope_Ncoeff_b(j,slt)  = SUM{slp, Ncoeff_frtREV_b(j,slp,slt)  * slopeshare(slp) } ;
 MeanSlope_Ncoeff_c(j,slt)  = SUM{slp, Ncoeff_frtREV_c(j,slp,slt)  * slopeshare(slp) } ;
 mean_Ncoeff_b1(j)   =   SUM{slt,  MeanSlope_Ncoeff_b(j,slt) * soilshare(slt)  }     ;
 mean_Ncoeff_c1(j)   =   SUM{slt,  MeanSlope_Ncoeff_c(j,slt) * soilshare(slt)  }     ;

*Average erosion, runoff and nitrogen parameters (weighted average over soil distribution)
 meanErosionCrops(j) = SUM{soil, finnsoilshare(soil) * MeanSlope_CropErosion(j,soil)}       ;
 meanRunoffCrops(j)  = SUM{soil, finnsoilshare(soil) * MeanSlope_CropRunoff(j,soil)}        ;
 meanRunoffFall(flw)  = SUM{soil, finnsoilshare(soil) * MeanSlope_FallErosion(flw,soil)}    ;
 meanErosionFall(flw) = SUM{soil, finnsoilshare(soil) * MeanSlope_FallRunoff(flw,soil) }    ;
 meanNcoeff_frt_b2(j) = SUM{soil, finnsoilshare(soil) * mean_N_coeff_frt_b(j,soil)}       ;
 meanNcoeff_frt_c2(j) = SUM{soil, finnsoilshare(soil) * mean_N_coeff_frt_c(j,soil)}       ;
 meanNloadIC_flw(flw) = SUM{soil, finnsoilshare(soil) * meanNload_flw(flw,soil)}         ;
 display  mean_Ncoeff_b1, meanNcoeff_frt_b2, mean_Ncoeff_c1, meanNcoeff_frt_c2, meanNloadIC_flw ;

*SHARE OF P STATUS OF TOTAL MODELED AREA
 soilPstShareCrop(pst) =  SUM{(j,slt,slp), crop_distr(j,slp,slt,Pst)} /  crop_ha        ;
 soilPstShareFall(pst) =  SUM{(flw,slt,slp),fallow_distr(flw,slp,slt,Pst)} / fallow_ha  ;
 soilPstShare(pst)     =  [SUM{(j,slt,slp), crop_distr(j,slp,slt,Pst)} +
SUM{(flw,slt,slp),fallow_distr(flw,slp,slt,Pst)}] / model_ha  ;
* Somewhere in code still references to soil P as PStm, so the parameter is given values from soil P GIS
integers
 PStm(j,slp,slt,Pst) = Pst_no(Pst)  ;
*AVERAGE P status of soil (given GIS data distribution)
 PsAvgcrop  = SUM{(j,slp,slt,Pst), crop_distr(j,slp,slt,Pst)* Pst_no(Pst)} /SUM{(j,slp,slt,Pst),
crop_distr(j,slp,slt,Pst)}              ;
 PsAvgfall = SUM{(flw,slp,slt,Pst), fallow_distr(flw,slp,slt,Pst)* Pst_no(Pst)} /SUM{(flw,slp,slt,Pst),
fallow_distr(flw,slp,slt,Pst)}     ;
 PsAvg = (PsAvgcrop * crop_ha +  PsAvgfall * fallow_ha) / model_ha         ;
*Checking the distribution of barley land to different data classes
 crop_distr_f13(slp, slt, Pst)  = crop_distr('f13', slp, slt, Pst) ;
*Checking nitrogen loads when "normal" 80 kg is used
NloadIC80(j,slp,slt)= NloadIC(j,slp,slt,'nfer80')                 ;
*crop_distr18(j, slp, slt) = crop_distr(j, slp, slt, 'Pst18')        ;

*Reducing P dimension from the crop and fallow distributions (not needed for Nitrogen load calculations)
 crop_dist_noP(j, slp, slt) = SUM{Pst, crop_distr(j, slp, slt, Pst)}   ;
 fall_dist_noP(flw, slp, slt) = SUM{Pst, fallow_distr(flw, slp, slt, Pst)} ;
*Checking that the total land area still matches the earlier distributions
 totX =  SUM{(j, slp, slt), crop_dist_noP(j, slp, slt) } ;
*Reducing the slope dimension from crop distribution (in earlier model specfications it was not used for N
load)
 crop_dist_no_slp(j, slt) = SUM{slp, crop_dist_noP(j, slp, slt)} ;

*NITROGEN LOAD--------------------------------------------------------------------
* ( calculating different compositions of the nitrogen load for the watershed )
 cropN = SUM{(j,slp,slt), NloadIC(j,slp,slt,'nfer80') * crop_dist_noP(j, slp, slt)}        ;
```

```
 fallN = SUM{(flw,slp,slt), NloadIC_flw(flw,slp,slt,'nfer0') * fall_dist_noP(flw, slp, slt)} ;
 fallmeanN = SUM{flw, meanNloadIC_flw(flw) *  SUM{(slp,slt), fall_dist_noP(flw,slp,slt) }  } ;
 TotalN_old  = cropN + fallN ;
 TotalNperha_old = TotalN_old/model_ha ;
 fallmeanN_perha   =  fallmeanN /  SUM{(flw,slp,slt), fall_dist_noP(flw,slp,slt) } ;
 cropNamx  = SUM{(j,slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slp,slt)])*
crop_dist_noP(j, slp, slt)  } ;
 cropNrevised  = SUM{(j,slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[mytvasNfertAvg(j) *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 cropNmeanBase  = SUM{j, (meanNcoeff_frt_b2(j) * exp[mytvasNfertAvg(j)  * meanNcoeff_frt_c2(j)])*
(cropshare(j)* model_ha)  } ;
 cropNmeanBase2  = SUM{j, (mean_Ncoeff_b1(j) * exp[mytvasNfertAvg(j)  * mean_Ncoeff_c1(j)])*
(cropshare(j)* model_ha)  } ;
 cropN_sensitivity  = SUM{(j,slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt)/2 *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 cropN_mytvasHi  = SUM{(j,slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[mytvasNfertHi(j) *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 cropN_mytvasLo  = SUM{(j,slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[mytvasNfertLo(j) *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 ccbarleyNload = SUM{(slp,slt), (Ncoeff_frtREV_b('f13',slp,slt) * exp[mytvasNfertHi('f13') *
Ncoeff_frtREV_c('f13',slp,slt)])* crop_dist_noP('f13', slp, slt)  } ;
 ccbarleyNload_ha = ccbarleyNload / SUM{(slp,slt), crop_dist_noP('f13', slp, slt)} ;
 cropNeach(j,slp,slt) = (Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slp,slt)])*
crop_dist_noP(j, slp, slt) ;
 cropNeachperha(j,slp,slt)$( crop_dist_noP(j, slp, slt) gt 0 ) =  cropNeach(j,slp,slt) /  crop_dist_noP(j, slp, slt)
;
 cropNsum =  SUM{(j,slp,slt), cropNeach(j,slp,slt) }   ;
 cropNsumperha = cropNsum / SUM{(j, slp, slt),  crop_dist_noP(j, slp, slt) } ;
 cropN_soils(slt) = SUM{(j,slp), (Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slp,slt)])*
crop_dist_noP(j, slp, slt)  } ;
 cropN_slopes(slp) = SUM{(j,slt),(Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slp,slt)])*
crop_dist_noP(j, slp, slt)  } ;
 cropN_crops(j) = SUM{(slp,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slp,slt)])*
crop_dist_noP(j, slp, slt)  } ;
 TotalNrevised =  cropNrevised + fallN        ;
 TotalN_MYTVAS_Hi =  cropN_mytvasHi + fallN        ;
 TotalN_MYTVAS_Lo =  cropN_mytvasLo + fallN        ;
 TotalN_mean =  cropNmeanBase + fallmeanN        ;
 TotalN_mean2 =  cropNmeanBase2 + fallmeanN        ;
 TotalNrevisedPerHa   = TotalNrevised /model_ha ;
 N_LoadbySlpMYTVAS(slp) = SUM{(j,slt), (Ncoeff_frtREV_b(j,slp,slt) * exp[mytvasNfertHi(j) *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 N_LoadbySltMYTVAS(slt) = SUM{(j,slp), (Ncoeff_frtREV_b(j,slp,slt) * exp[mytvasNfertHi(j) *
Ncoeff_frtREV_c(j,slp,slt)])* crop_dist_noP(j, slp, slt)  } ;
 TotNPerHa_Aura = TotalNrevisedPerHa ;

* MEAN EROSION AND RUNOFF-----------------------------------------------------------------
* ( calculating mean erosion and runoff )
 cropMeanErosion          = SUM{(j,slp,slt), sloss(j,slp,slt) *  crop_dist_noP(j,slp,slt)/crop_ha }              ;
 fallowMeanErosion        = SUM{(flw,slp,slt), sloss_flw(flw,slp,slt) *  fall_dist_noP(flw,slp,slt)/fallow_ha }
;
```

```
 arableMeanErosion        =  cropMeanErosion * crop_ha/model_ha + fallowMeanErosion *
fallow_ha/model_ha          ;
 cropMeanRunoff          = SUM{(j,slp,slt), runoff(j,slp,slt) *  crop_dist_noP(j,slp,slt)/crop_ha }            ;
 fallowMeanRunoff        = SUM{(flw,slp,slt), runoff_flw(flw,slp,slt) *  fall_dist_noP(flw,slp,slt)/fallow_ha }
;
 arableMeanRunoff        =  cropMeanRunoff * crop_ha/model_ha + fallowMeanRunoff *
fallow_ha/model_ha          ;

*PHOSPHORUS LOAD-------------------------------------------------------------------
* ( calculating P loads from erosion and runoff )

*Dissolved reactive P for the region    (DRP)
 DRP_loadparam(j,slp,slt,Pst)   = runoff(j,slp,slt) * [2*(Pst_no(Pst) + 0.01*pmax(Pst,j,slt)) - 1.5]  * POWER(10,
-4)   ;
 DRP_loadMYTVAS(j,slp,slt,Pst)  = runoff(j,slp,slt) * [2*(Pst_no(Pst) + 0.01*mytvasPfertHi(j)) - 1.5]  *
POWER(10, -4)   ;
 DRP_load15kg(j,slp,slt,Pst)  = runoff(j,slp,slt) * [2*(Pst_no(Pst) + 0.01*15) - 1.5]  * POWER(10, -4)   ;
 cropDRP = SUM{(j,slp,slt,Pst), DRP_loadparam(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
 MYTVAScropDRP = SUM{(j,slp,slt,Pst), DRP_loadMYTVAS(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
 check15kgCropDRP = SUM{(j,slp,slt,Pst), DRP_load15kg(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
*Assume no fertilisation on fallow
 DRP_loadparam_flw(flw,slp,slt,Pst)  = runoff_flw(flw,slp,slt) * [2*Pst_no(Pst)  - 1.5]  * POWER(10, -4)   ;
 fallDRP = SUM{(flw,slp,slt,Pst), DRP_loadparam_flw(flw,slp,slt,Pst) * fallow_distr(flw, slp, slt, Pst)}  ;
*Version calculated with mean slope and soil
 DRP_meanload(j)  = meanRunoffCrops(j) * [2*(PsAvgcrop + 0.01*mytvasPfertHi(j)) - 1.5]  * POWER(10, -4)
;
 DRP_meanload_flw(flw) =  meanRunoffFall(flw) * [2*PsAvgfall  - 1.5]  * POWER(10, -4)   ;
 cropDRPmeanland = SUM{j, DRP_meanload(j) * cropshare(j)* model_ha}  ;
 fallDRPmeanland = SUM{flw, DRP_meanload_flw(flw) * fallowshare(flw)* model_ha }  ;

*Particulate P for the region      (PP)
 PP_loadparam(j,slp,slt,Pst) = sloss(j,slp,slt) *  [250 * log(Pst_no(Pst) + 0.01 * pmax(Pst,j,slt))- 150] *
POWER(10, -6) ;
 PP_loadMYTVAS(j,slp,slt,Pst) = sloss(j,slp,slt) *  [250 * log(Pst_no(Pst) + 0.01 * mytvasPfertHi(j))- 150] *
POWER(10, -6) ;
 PP_load15kg(j,slp,slt,Pst) = sloss(j,slp,slt) *  [250 * log(Pst_no(Pst) + 0.01 * 15)- 150] * POWER(10, -6) ;
 cropPP = SUM{(j,slp,slt,Pst), PP_loadparam(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
 MYTVAScropPP = SUM{(j,slp,slt,Pst), PP_loadMYTVAS(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
 check15kgCropPP =  SUM{(j,slp,slt,Pst), PP_load15kg(j,slp,slt,Pst) * crop_distr(j, slp, slt, Pst)}  ;
* PP_HtS_pram(j) = sloss(j,"slp1","slt16") * [250 * log(Pst_no("Pst15") + 0.01 * pmax('pst14',j,"slt16"))- 150]
* POWER(10, -6) ;
 PP_loadparam_flw(flw,slp,slt,Pst) = sloss_flw(flw,slp,slt) *  [250 * log(Pst_no(Pst))- 150] * POWER(10, -6) ;
 fallPP = SUM{(flw,slp,slt,Pst), PP_loadparam_flw(flw,slp,slt,Pst) * fallow_distr(flw, slp, slt, Pst)}  ;
*Version calculated with mean slope and soil
 PP_meanload(j)      =  meanErosionCrops(j) * [250 * log(PsAvgcrop + 0.01 * mytvasPfertHi(j))- 150] *
POWER(10, -6) ;
 PP_meanload_flw(flw) =  meanErosionFall(flw) *  [250 * log(PsAvgfall)- 150] * POWER(10, -6) ;
 cropPPmeanland = SUM{j, PP_meanload(j) * cropshare(j)* model_ha}  ;
 fallPPmeanland = SUM{flw, PP_meanload_flw(flw) * fallowshare(flw)* model_ha }  ;

* TP for each combination (for map export of the baseline)
 totalCropP(j,slp,slt,Pst) =  DRP_loadparam(j,slp,slt,Pst)       + PP_loadparam(j,slp,slt,Pst)     / bio_coef   ;
```

```
 totalFallP(flw,slp,slt,Pst) =  DRP_loadparam_flw(flw,slp,slt,Pst) + PP_loadparam_flw(flw,slp,slt,Pst) /
bio_coef   ;

*Adding up total loads
 MYTVAStotalDRP  =  MYTVAScropDRP + fallDRP     ;
 MYTVAStotalPP   =  MYTVAScropPP + fallPP      ;
 check15kgTotalDRP  =  check15kgCropDRP + fallDRP     ;
 check15kgTotalPP   =  check15kgCropPP + fallPP      ;
 TotalDRP =  cropDRP + fallDRP     ;
 TotalPP  =  cropPP + fallPP      ;
 meanlandTotDRP  =  cropDRPmeanland +  fallDRPmeanland ;
 meanlandTotPP   =  cropPPmeanland +  fallPPmeanland   ;
 DRPtoPPratio  =  TotalDRP/(TotalPP/bio_coef) ;
 MYTVASmodBioP = MYTVAStotalDRP + MYTVAStotalPP ;
 modBioP = cropPP + cropDRP + fallPP + fallDRP  ;
 modTP = TotalDRP + TotalPP/bio_coef         ;
 MYTVASmodTP =  MYTVAStotalDRP + MYTVAStotalPP/bio_coef   ;
 check15kgModTP =  check15kgTotalDRP + check15kgTotalPP/bio_coef   ;
 meanlandmodTP =  meanlandTotDRP + meanlandTotPP/bio_coef   ;
 modTPperha = modTP/model_ha                 ;
 mod_DRP_TP_share = TotalDRP /  modTP         ;
 baseDRP =   veps_TP_2002 *  mod_DRP_TP_share ;
 basePP = veps_TP_2002 - baseDRP       ;

*An alternative way to model  WITHOUT Uusitalo functions, ICECREAM Total P -Assuming uniform 15 kg ha-
1 P-fert
 TcropP = SUM{(j,slp,slt,Pst), TPloadIC(j,slp,slt,Pst,'pfer15') * crop_distr(j, slp, slt, Pst)}           ;
 TfallP = SUM{(flw,slp,slt,Pst), TPloadIC_flw(flw,slp,slt,Pst,'pfer15') * fallow_distr(flw, slp, slt, Pst)} ;
 TotalP_IC = SUM{(j,slp,slt,Pst), TPloadIC(j,slp,slt,Pst,'pfer15') * crop_distr(j, slp, slt, Pst)} +
SUM{(flw,slp,slt,Pst), TPloadIC_flw(flw,slp,slt,Pst,'pfer15') * fallow_distr(flw, slp, slt, Pst)} ;
 TotalPperhaIC = TotalP_IC/ model_ha            ;
 TcropP_test(j,slp,slt,Pst) = TPloadIC(j,slp,slt,Pst,'pfer15') * crop_distr(j, slp, slt, Pst) ;
 TcropPperha =  TcropP/crop_ha       ;
 cropDRPperha = cropDRP/crop_ha      ;
 metaOverIC_TP = check15kgModTP / TotalP_IC  ;

*CONVERTING THE MODEL TO ORIGINAL PARAMETRIZED SOIL CLASSES---------------------
*(Directly) ICECREAM estimated nitrogen loads
 Ncoeff_frt_b(j,slp,'slt1') =  Ncoeff_frtREV_b(j,slp,'slt5')       ;
 Ncoeff_frt_b(j,slp,'slt2') =  Ncoeff_frtREV_b(j,slp,'slt16')      ;
 Ncoeff_frt_b(j,slp,'slt3') =  Ncoeff_frtREV_b(j,slp,'slt1')       ;
 Ncoeff_frt_b(j,slp,'slt4') =  Ncoeff_frtREV_b(j,slp,'slt9')       ;

 Ncoeff_frt_c(j,slp,'slt1') =  Ncoeff_frtREV_c(j,slp,'slt5')       ;
 Ncoeff_frt_c(j,slp,'slt2') =  Ncoeff_frtREV_c(j,slp,'slt16')      ;
 Ncoeff_frt_c(j,slp,'slt3') =  Ncoeff_frtREV_c(j,slp,'slt1')       ;
 Ncoeff_frt_c(j,slp,'slt4') =  Ncoeff_frtREV_c(j,slp,'slt9')       ;

 meanNcoeff_frt_b(j) = mean_Ncoeff_b1(j)              ;
 meanNcoeff_frt_c(j) = mean_Ncoeff_c1(j)              ;

 loadCoeff_N_flw(flw,slp,'slt1')   = NloadIC_flw(flw,slp,'slt5', 'nfer0')  ;
```

```
 loadCoeff_N_flw(flw,slp,'slt2')   = NloadIC_flw(flw,slp,'slt16', 'nfer0') ;
 loadCoeff_N_flw(flw,slp,'slt3')   = NloadIC_flw(flw,slp,'slt1', 'nfer0')  ;
 loadCoeff_N_flw(flw,slp,'slt4')   = NloadIC_flw(flw,slp,'slt9', 'nfer0')  ;

 meanNcoeff_flw(flw) = meanNloadIC_flw(flw) ;

 rnoff(j,slp,'slt1') =  runoff(j,slp,'slt5')       ;
 rnoff(j,slp,'slt2') =  runoff(j,slp,'slt16')      ;
 rnoff(j,slp,'slt3') =  runoff(j,slp,'slt1')       ;
 rnoff(j,slp,'slt4') =  runoff(j,slp,'slt9')       ;
 rnoff_flw(flw,slp,'slt1') =  runoff_flw(flw,slp,'slt5')       ;
 rnoff_flw(flw,slp,'slt2') =  runoff_flw(flw,slp,'slt16')      ;
 rnoff_flw(flw,slp,'slt3') =  runoff_flw(flw,slp,'slt1')       ;
 rnoff_flw(flw,slp,'slt4') =  runoff_flw(flw,slp,'slt9')       ;

 erosion(j,slp,'slt1') =  sloss(j,slp,'slt5')       ;
 erosion(j,slp,'slt2') =  sloss(j,slp,'slt16')      ;
 erosion(j,slp,'slt3') =  sloss(j,slp,'slt1')       ;
 erosion(j,slp,'slt4') =  sloss(j,slp,'slt9')       ;
 erosion_flw(flw,slp,'slt1') =  sloss_flw(flw,slp,'slt5')       ;
 erosion_flw(flw,slp,'slt2') =  sloss_flw(flw,slp,'slt16')      ;
 erosion_flw(flw,slp,'slt3') =  sloss_flw(flw,slp,'slt1')       ;
 erosion_flw(flw,slp,'slt4') =  sloss_flw(flw,slp,'slt9')       ;

*REASSIGN FOR THE OLDER CROP SET (smaller crop set defition in the economic abatement model)
 meanRunoffCrops_old('f1') = meanRunoffCrops('f1')  ;
 meanRunoffCrops_old('f2') = meanRunoffCrops('f2')  ;
 meanRunoffCrops_old('f3') = meanRunoffCrops('f3')  ;
 meanRunoffCrops_old('f4') = meanRunoffCrops('f4')  ;
 meanRunoffCrops_old('f5') = meanRunoffCrops('f5')  ;
 meanRunoffCrops_old('f6') = meanRunoffCrops('f6')  ;
 meanRunoffCrops_old('f7') = meanRunoffCrops('f13')  ;
 meanRunoffCrops_old('f8') = meanRunoffCrops('f14')  ;
 meanRunoffCrops_old('f9') = meanRunoffCrops('f15')  ;
 meanRunoffCrops_old('f10') = meanRunoffCrops('f19')  ;
 meanRunoffCrops_old('f11') = meanRunoffCrops('f20')  ;
 meanRunoffCrops_old('f12') = meanRunoffCrops('f21')  ;
 meanRunoffCrops_old('f13') = meanRunoffCrops('f37')  ;
 meanRunoffCrops_old('f14') = meanRunoffCrops('f38')  ;
 meanRunoffCrops_old('f15') = meanRunoffCrops('f39')  ;
 meanRunoffCrops_old('f16') = meanRunoffCrops('f43')  ;
 meanRunoffCrops_old('f17') = meanRunoffCrops('f43')  ;
 meanRunoffCrops_old('f18') = meanRunoffCrops('f43')  ;
 meanRunoffCrops_old('f19') = meanRunoffCrops('f28')  ;
 meanRunoffCrops_old('f20') = meanRunoffCrops('f25')  ;
 meanRunoffCrops_old('f21') = meanRunoffCrops('f34')  ;
 meanRunoffCrops_old('f22') = meanRunoffCrops('f22')  ;
 meanRunoffCrops_old('f23') = meanRunoffCrops('f23')  ;
 meanRunoffCrops_old('f24') = meanRunoffCrops('f24')  ;
 meanRunoffCrops_old('f25') = meanRunoffCrops('f16')  ;
 meanRunoffCrops_old('f26') = meanRunoffCrops('f17')  ;
 meanRunoffCrops_old('f27') = meanRunoffCrops('f18')  ;
```

```
meanErosionCrops_old('f1') = meanErosionCrops('f1')  ;
meanErosionCrops_old('f2') = meanErosionCrops('f2')  ;
meanErosionCrops_old('f3') = meanErosionCrops('f3')  ;
meanErosionCrops_old('f4') = meanErosionCrops('f4')  ;
meanErosionCrops_old('f5') = meanErosionCrops('f5')  ;
meanErosionCrops_old('f6') = meanErosionCrops('f6')  ;
meanErosionCrops_old('f7') = meanErosionCrops('f13')  ;
meanErosionCrops_old('f8') = meanErosionCrops('f14')  ;
meanErosionCrops_old('f9') = meanErosionCrops('f15')  ;
meanErosionCrops_old('f10') = meanErosionCrops('f19')  ;
meanErosionCrops_old('f11') = meanErosionCrops('f20')  ;
meanErosionCrops_old('f12') = meanErosionCrops('f21')  ;
meanErosionCrops_old('f13') = meanErosionCrops('f37')  ;
meanErosionCrops_old('f14') = meanErosionCrops('f38')  ;
meanErosionCrops_old('f15') = meanErosionCrops('f39')  ;
meanErosionCrops_old('f16') = meanErosionCrops('f43')  ;
meanErosionCrops_old('f17') = meanErosionCrops('f43')  ;
meanErosionCrops_old('f18') = meanErosionCrops('f43')  ;
meanErosionCrops_old('f19') = meanErosionCrops('f28')  ;
meanErosionCrops_old('f20') = meanErosionCrops('f25')  ;
meanErosionCrops_old('f21') = meanErosionCrops('f34')  ;
meanErosionCrops_old('f22') = meanErosionCrops('f22')  ;
meanErosionCrops_old('f23') = meanErosionCrops('f23')  ;
meanErosionCrops_old('f24') = meanErosionCrops('f24')  ;
meanErosionCrops_old('f25') = meanErosionCrops('f16')  ;
meanErosionCrops_old('f26') = meanErosionCrops('f17')  ;
meanErosionCrops_old('f27') = meanErosionCrops('f18')  ;

$ontext
meanNcoeff_old('f1') = meanNcoeff('f1')  ;
meanNcoeff_old('f2') = meanNcoeff('f2')  ;
meanNcoeff_old('f3') = meanNcoeff('f3')  ;
meanNcoeff_old('f4') = meanNcoeff('f4')  ;
meanNcoeff_old('f5') = meanNcoeff('f5')  ;
meanNcoeff_old('f6') = meanNcoeff('f6')  ;
meanNcoeff_old('f7') = meanNcoeff('f13')  ;
meanNcoeff_old('f8') = meanNcoeff('f14')  ;
meanNcoeff_old('f9') = meanNcoeff('f15')  ;
meanNcoeff_old('f10') = meanNcoeff('f19')  ;
meanNcoeff_old('f11') = meanNcoeff('f20')  ;
meanNcoeff_old('f12') = meanNcoeff('f21')  ;
meanNcoeff_old('f13') = meanNcoeff('f37')  ;
meanNcoeff_old('f14') = meanNcoeff('f38')  ;
meanNcoeff_old('f15') = meanNcoeff('f39')  ;
meanNcoeff_old('f16') = meanNcoeff('f43')  ;
meanNcoeff_old('f17') = meanNcoeff('f43')  ;
meanNcoeff_old('f18') = meanNcoeff('f43')  ;
meanNcoeff_old('f19') = meanNcoeff('f28')  ;
meanNcoeff_old('f20') = meanNcoeff('f25')  ;
meanNcoeff_old('f21') = meanNcoeff('f34')  ;
meanNcoeff_old('f22') = meanNcoeff('f22')  ;
meanNcoeff_old('f23') = meanNcoeff('f23')  ;
```

```
meanNcoeff_old('f24') = meanNcoeff('f24')  ;
meanNcoeff_old('f25') = meanNcoeff('f16')  ;
meanNcoeff_old('f26') = meanNcoeff('f17')  ;
meanNcoeff_old('f27') = meanNcoeff('f18')  ;
$offtext

*GRAPHS-------------------------------------------------------------------------
 set
 scen        Scenarios  of which each is a line    /crop_N_data, crop_N_est, crop_P_data, crop_P_est,
cropfarm_N_data,  cropfarm_P_data, crop_N_hom_data, crop_N_hom_est, crop_P_hom_data,
crop_P_hom_est, cropfarm_N_hom_data,  cropfarm_P_hom_data /
 item        y and x axis items       /Nredcost_f, nloadred_f, Predcost_f, tploadred_f, NredCost,
NloadAbatement, PredCost, PloadAbatement, share_of_fallow_land, abatement_percent, kg_a/
 litems    y and x axis items /n_load, tp_load, N_land, P_land, P_fallow, N_fallow, abatement_percent,
landShareDat, landshare_model, landshare_2003, landshare, kg_a /
 rescen / N_reduction, P_reduction  /
 litem /  landshare /
 landallo
* Mean_CN_A, Mean_CP_A, Heter_CN_A, Heter_CP_A, simu_mean_P, simu_mean_N
* / NTF, PTF, abat_percent /
*fitems y and x axis items / n_load, tp_load,  P_fallow, N_fallow /
 loadestmt / VEMALA_AVG, VEPS, SAMA_MYTVAS_Hi, SAMA_MYTVAS_Lo, SAMA_ENVSU,
SAMA_RESCALED, SAMA_IC_TP, VEMALA_2002, VEMALA_2003, VEMALA_2009   /
 loadestmt2 / VEPS_avg_weather, metamodel_baseline, VEMALA_2002, VEMALA_2003, VEMALA_2008,
VEMALA_AVG, VEMALA_2009, VIHMA   /
 box / xdummy, crop_types/
 coor / landuse /
 ;
 alias (j, crops) ;
*NITROGEN RESPONSE
 parameter meanNloadResponseGraph(j, nfert, litems) ;

 meanNloadResponseGraph(j, nfert, 'kg_a') = N_frtlzn(nfert) ;
 meanNloadResponseGraph(j, nfert, 'n_load') = meanNcoeff_frt_b(j) * exp[N_frtlzn(nfert) *
mean_Ncoeff_c1(j)] ;

$libinclude gnuplotxyz meanNloadResponseGraph kg_a n_load

*BAR CHART FOR LOAD COMPARISON
 parameter Pload_graph(loadestmt, item, box)  ;
 parameter Pload_graph2(loadestmt2, item, box)  ;
 parameter Nload_graph(loadestmt, item, box)  ;

 Pload_graph2('VEPS_avg_weather', 'kg_a', box)       = veps_TP_2002      ;
 Pload_graph2('VEMALA_2003', 'kg_a', box)           = VEMALA_TP_2003     ;
 Pload_graph2('VEMALA_2002', 'kg_a', box)           = VEMALA_TP_2002     ;
 Pload_graph2('VEMALA_2009', 'kg_a', box)           = VEMALA_TP_2009     ;
 Pload_graph2('VEMALA_AVG', 'kg_a', box)            = VEMALA_P_AVG      ;
 Pload_graph2('VIHMA', 'kg_a', box)                 = VIHMA_P      ;

 Pload_graph2('metamodel_baseline', 'kg_a', box)      = modTP            ;
```

```
 Pload_graph('VEPS', 'kg_a', box)            = veps_TP_2002      ;
 Pload_graph('SAMA_MYTVAS_Hi', 'kg_a', box)   = MYTVASmodTP       ;
 Pload_graph('SAMA_ENVSU', 'kg_a', box)       = modTP             ;
*Pload_graph('SAMA_RESCALED', 'kg_a', box)    = modTP_rescal      ;
 Pload_graph('SAMA_IC_TP', 'kg_a', box)       = TotalP_IC         ;

$setglobal gp_title 'Comparison of P load results'
$setglobal gp_style "histogram"
$setglobal gp_yrange "0:60000"
*$libinclude gnuplotxyz Pload_graph xdummy

$setglobal gp_title 'Comparison of P load results'
$setglobal gp_style "histogram"
$setglobal gp_yrange "0:100000"
*$libinclude gnuplotxyz Pload_graph2 xdummy

*Nitrogen load comparison histogram
 Nload_graph('VEPS', 'kg_a', box)            = veps_N_2002          ;
 Nload_graph('VEMALA_2002', 'kg_a', box)     = VEMALA_N_2002        ;
 Nload_graph('VEMALA_2003', 'kg_a', box)     = VEMALA_N_2003        ;
 Nload_graph('VEMALA_AVG', 'kg_a', box)      = VEMALA_N_AVG      ;
 Nload_graph('SAMA_MYTVAS_Hi', 'kg_a', box)  = TotalN_MYTVAS_Hi      ;
 Nload_graph('SAMA_MYTVAS_Lo', 'kg_a', box)  = TotalN_MYTVAS_Lo      ;
 Nload_graph('SAMA_ENVSU', 'kg_a', box)      = TotalNrevised        ;

$setglobal gp_title 'Comparison of N load results'
$setglobal gp_style "histogram"
$setglobal gp_yrange "0:1000000"
*$libinclude gnuplotxyz Nload_graph xdummy

*FOR MAP EXPORTS------------------------------------------------------------------

*NEED FALLOW WITH CROP CODE (totalFallP)
 totalCropPk1(k1,slp,slt,Pst)  =  totalCropP(k1,slp,slt,Pst)  ;
 totalCropPk2(k2,slp,slt,Pst)  =  totalCropP(k2,slp,slt,Pst)  ;
 totalCropPk3(k3,slp,slt,Pst)  =  totalCropP(k3,slp,slt,Pst)  ;

* totalCropP_k1(cropcode,slp,slt,Pst)  = totalCropPk1(k1,slp,slt,Pst) ;
 totalCropP_k1('1110',slp,slt,Pst)  = totalCropPk1('f1',slp,slt,Pst) ;
 totalCropP_k1('1120',slp,slt,Pst)  = totalCropPk1('f4',slp,slt,Pst) ;
 totalCropP_k1('1220',slp,slt,Pst)  = totalCropPk1('f7',slp,slt,Pst) ;
 totalCropP_k1('1230',slp,slt,Pst)  = totalCropPk1('f10',slp,slt,Pst) ;
 totalCropP_k1('1310',slp,slt,Pst)  = totalCropPk1('f13',slp,slt,Pst) ;
 totalCropP_k1('1320',slp,slt,Pst)  = totalCropPk1('f16',slp,slt,Pst) ;
 totalCropP_k1('1400',slp,slt,Pst)  = totalCropPk1('f19',slp,slt,Pst) ;
 totalCropP_k1('1510',slp,slt,Pst)  = totalCropPk1('f22',slp,slt,Pst) ;
 totalCropP_k1('2110',slp,slt,Pst)  = totalCropPk1('f25',slp,slt,Pst) ;
 totalCropP_k1('3110',slp,slt,Pst)  = totalCropPk1('f28',slp,slt,Pst) ;
* totalCropP_k1('3130',slp,slt,Pst)  = totalCropPk1('f31',slp,slt,Pst) ;
 totalCropP_k1('3210',slp,slt,Pst)  = totalCropPk1('f34',slp,slt,Pst) ;
 totalCropP_k1('4110',slp,slt,Pst)  = totalCropPk1('f37',slp,slt,Pst) ;
* totalCropP_k1('4120',slp,slt,Pst)  = totalCropPk1('f40',slp,slt,Pst) ;
```

```
 totalCropP_k1('6121',slp,slt,Pst)  = totalCropPk1('f43',slp,slt,Pst) ;

pmaxTPperha(cropcode,slp,slt,Pst)   =  totalCropP_k1(cropcode,slp,slt,Pst) ;
pmaxTPperha('9300',slp,slt,Pst)     =  totalFallP('ffn',slp,slt,Pst) ;
pmaxTPperha('9400',slp,slt,Pst)     =  totalFallP('gf1n',slp,slt,Pst) ;
pmaxTPperha('9620',slp,slt,Pst)     =  totalFallP('bf1n',slp,slt,Pst) ;

* totalCropP_k2(cropcode,slp,slt,Pst)  = totalCropPk2(k2,slp,slt,Pst) ;
* totalCropP_k3(cropcode,slp,slt,Pst)  = totalCropPk3(k3,slp,slt,Pst) ;

display slopeclassmean, slopeshare, Pst_no, pst ;
display crop_ha, totX, fallow_ha, noLoadLand, cropshare, fallowshare, fallsoilshare  ;
display Ncoeff_frtREV_b, Ncoeff_frtREV_c,  Ncoeff_frt_b,  Ncoeff_frt_c,  meanNcoeff_frt_b2,
meanNcoeff_frt_c2   ;
display cropN_soils, cropN_slopes, cropN_crops ;
display TotalN_old, TotalN_mean, TotalN_mean2, TotalNrevised, veps_N_2002 ;
display cropN, cropNmeanBase, cropNmeanBase2, cropNrevised, cropN_sensitivity, cropN_mytvasHi,
cropN_mytvasLo,   fallN, fallmeanN  ;
display cropNsum, cropNeach, cropNeachperha, cropNsumperha, veps_Nperha_2002, TotalNrevisedPerHa,
TotalNperha_old ;
display DRP_VEPS, crop_distr_f13, cropDRP, fallDRP, cropPP, fallPP  ;
display modBioP, modTP, meanlandmodTP, TotalP_IC,  check15kgModTP, metaOverIC_TP, veps_TP_2002,
MYTVASmodTP, TcropP, TfallP, TcropPperha, cropDRPperha, baseDRP, basePP, DRPtoPPratio,
mod_DRP_TP_share   ;
display noLoadLand_pr, modTPperha,  TotalPperhaIC, TotalNperha_old, noLoadLand_pr, PsAvgcrop,
PsAvgfall, PsAvg   ;
display erosion, runoff, rnoff  ;
display soilshare, finnsoilshare, runoff__flw, sloss__flw, erosion_flw ;
display runoffdata, erosiondata, meanRunoffCrops, meanRunoffFall, meanErosionCrops,  meanErosionFall
;
display pfclass, upperPlimit2007, PfclassLow, pmax  ;
display cropshare, fallowshare, slopeAreaCrop, slopeAreaFallow, meanSlopeCrop, meanSlopeFallow,
meanSlopeArable ;
display cropMeanErosion, fallowMeanErosion, arableMeanErosion, cropMeanRunoff, fallowMeanRunoff,
arableMeanRunoff   ;
display cropLandbySlope, cropLandbySoil, cropLandbyPst, fallowLandbySlope, fallowLandbySoil,
fallowLandbyPst, crop_distr       ;
display arableLandbySlope,arableLandbySoil, arableLandbyPst, cropLandbySlopeShare,
cropLandbySoilShare, fallowLandbySlopeShare, fallowLandbySoilShare  ;
display grassLandSharebySlope, grassLandSharebySoil, grassLandSharebyPst ;
display DRP_loadparam, PP_loadparam ;
display soilPstShareCrop, soilPstShareFall, soilPstShare ;
display Ncoeff_frtREV_b, N_LoadbySlpMYTVAS, N_LoadbySltMYTVAS, mytvasNfertHi, ccbarleyNload,
ccbarleyNload_ha, TotNPerHa_Aura ;
display k1, k2, k3, totalCropPk1, totalCropP_k1, fallmeanN_perha, pmaxTPperha ;

*PARAMETERS IN OPTIMIZATION IMPORTED FROM Aura_load_parameters.gdx
display erosion, rnoff, erosion_flw, rnoff_flw, Ncoeff_frt_b, Ncoeff_frt_c, loadCoeff_N_flw, fallmeanN,
meanNloadResponseGraph ;


*EXPORT RESULT GDX TO ACCESS------------------------------------------------------
```

```
*EXPORT TO MDB = add the script from below to the gdx viewer options
*ODBC SCRIPT
*ODBC DSN-less MS Access Driver
*Driver={Microsoft Access Driver (*.mdb)};Dbq= C:\GIS\SAMA_databaseGIS2003.mdb;

*OLE DB Provider for ODBC Access (Jet)
*Provider =MSDASQL; Driver={Microsoft Access Driver(*.mdb)};Dbq= C:\GIS\SAMA_databaseGIS2003.mdb;

*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', pmaxTPperha ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', totalCropP_k1  ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', landTPload  ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', totalCropPk2  ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', totalCropPk3  ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', totalFallP  ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', erosion     ;
*execute_unload 'C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx', erosion_flw ;
*execute_unload 'C:\gamsdir\crop_farm_IIASArev\Aura_cropfarm_results.gdx',Ploadred_ha53 ;

*View GDX
*execute 'gdxviewer.exe i=C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx sql
id=totalCropP_k1'
*execute 'gdxviewer.exe i=C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx sql
id=totalFallP'
*execute 'gdxviewer.exe i=C:\gamsdir\load_calibration_Aurajoki\Aura_load_parameters.gdx sql
id=pmaxTPperha'
$ontext
STEPS following the export from GAMS:

1. Creating long integer ID-field to the GIS source data (in results_Aura -ArcGIS model, add field, calculate
field operations)
 - check results_Aurajoki model from SAMA datamanagement toolbox
2. Creating text field to the GIS source field and it giving crop category values (text version needed to join
the GAMS result table)
3. Creating a new table with sql multijoin query (load table and GIS source data) matching the sets and
returning ID and load
SELECT LepsLandData.IDfield, LoadSW.level AS nutrientLoad INTO loadSW_to_map
FROM LoadSW INNER JOIN LepsLandData ON (LoadSW.dim5 = LepsLandData.dst) AND (LoadSW.dim4 =
LepsLandData.aspct) AND (LoadSW.dim3 = LepsLandData.slp) AND (LoadSW.dim2 = LepsLandData.Pst) AND
(LoadSW.dim1 = LepsLandData.sltclass);
4. In ArcGIS joining the new table with GIS source data based on the ID field
        - requires copying the both of the data to .gdb
$offtext
```

Finnish dairy nutrient abatement model

```
$Title Finnish dairy nutrient abatement model
*$sysinclude gams-f
$ontext
This file contains the profit maximisation problem of dairy farm with both field and cattle.  The choice variables are number of milking cows, fodder given tocattle type (milking, dry, heifer), land use (what crops to grow on which field area at given distances), nitrogen fertilisation per hectare (determining also the phosphorus quantity), volume of manure to be stored with different technology, volume of manure dispersed with different technology and manure dispersal location.
First some farm data is imported from Access/GIS. Load parameters from GDX file are produced in XXXXX.gms file. Yield parameters are imported from XXXXXX.gdx. Some of the parameter data is given in scalars and parametersof this file.
The subset of non cereal functions, as well as some of the constraints for crops/fallow, are defined in GAMS. Initial values and upper limits of nitrogen fertilisation are based on an analytical solution with given prices.
Objective functions of the model maximise the profits of the reprentative farmer of the watershed. Crop yield (per ha) is defined in a separate function. Nitrogen fertilisation (which determines fixed phophorus fertilisation) and the allocation of farm land (including crop type and tillage)are the decision variables.
The objective is subject to constraint keeping the slope, soil and Pst of the fields fixed, while allowing change of tillage and crops. Load functions are based on ICECREAM model results and determine soil and slope dependent erosion and runoff, which are combined to PP and DRP predecting equations of R.Uusitalo. Nitrogen is predected directly from the ICECREAM model.
First solve calculates the optimal results without the load constraints. Then,load constraints are made binding by tightening the constraint in a loop with 2\% iterations. Then, the unconstrained base line is resolved and loop is calculated for phosphorus.
OLS estimates for abatmenet cost functions are done in a separate file.
The main results from this file are:
-  Aura_N_redcost(count) (abetement costs for 2-60\% reduction)
-  baseTP and baseN loads (profit maximsing unconstrained load levels)
-  Optimal levels for decision variables
This file should have GDX= Aura_results_het.gdx for OLS estimates of the cost functions.
Options:
Malt barley share has been fixed to zero.
Uses the baseline land allocation of homogeneous load parameter version as the initial values for crop distribution. Minumum and maximum values for grass
$offtext

$onecho > cmd.txt
I="D:\GIS\SAMA_database2003.mdb"
X=KalaDairyData.gdx

Q1=select SUM_dairyDistances_Area from KalaDairyTotalFieldArea
p1=KalaDairyTotalFieldArea
Q2=select SUM_dairyDistances_Area from KalaDairyBelowMedianFieldArea
p2=KalaDairyBelowMedianFieldArea
Q3=select MED_DAIRY from KalaDairyMedianDistance
p3=KalaDairyMedianDistance
Q4=select MEAN_dairyDistances_Total_meters from BelowMedianFarmMeanDistance
p4=DairyBelowMedianDistance
Q5=select MEAN_dairyDistances_Total_meters from AboveMedianFarmMeanDistance
```

```
p5=DairyAboveMedianDistance
Q6=select MEAN_SUM_dairyDistances_Area from MeanDairyFarmSize
p6=MeanDairyFarmSize
$offecho
$call =mdb2gms \atcmd.txt

* SETS -----------------------------------------------------------

SETS
*       t time
*/     0*30 /
*GATHERING ALL INPUT ITEMS FOR SENSITIVITY ANALYSIS
  alli / milk, Nfertiliser, Pfertiliser, f0*f303, Total, gfs, gfn, bfs, bfn, ffn, sp, bs, is, cc, nc, fc, dstBM, dstAM,
meat, NPfertiliser, fuel, lubricant, labour, harvesting, calf, heifer, dry, milking, ALA, ADR, AHE, ACA,
ALA_DMI, bandSpreadVol,manVolperAU,manurePcont,manureNcont, manurePcomp,manureNcomp    /
   measures    / cropIncome, NetNutLoad, TotFallowShare, Profit, cropYield, dailyMilkYield, Dairy_cows,
manureN, manureP, manureVol, baseCropArea, baseFallowArea, AU, manStorSurface, rainWaterPercent,
lostManureNpercent, CostPerLitre, ShareOfSubsidyRevenue, ShareOfMilkRevenue, TotalForageShare, DMI,
DMY, transportCost, spreadCost, DispersalTechNetValue,dispCapitalShare
              PurchasedDM, FarmedDM, Nload, Pload, loadDRP, loadPP, PfieldBalance, NfieldBalance,
abvMedDistManureVolume, purchasedNitrogen, purchasedPhosphorus, annualMY, Nfertiliser, Pfertiliser,
ProfitDistance, ALA, ADR, AHE, ACA,ALA_DMI, bandSpreadVol,manVolperAU,manurePcont,manureNcont,
manurePcomp,manureNcomp, intakePcomp,intakeNcomp, lostManN
 /
   primary(alli)  primary products and inputs / f0*f303, gfs,gfn,bfs,bfn,ffn, milk, meat, Nfertiliser, Pfertiliser,
NPfertiliser, fuel, lubricant, labour, harvesting, calf, ALA, ADR, AHE, ACA, sp, bs, is   /

       dst distance classes
/
       dstBM
       dstAM
/
       f(primary) fodder types
/
f1*f15, f16 silage, f17*f30, f31 rape seed, f32*f237, f238 mash, f239 mash (dry), f240*f303
/
       j(f) farmed crops
/
*f0       green fodder  (silage parameters)
f1      winterwheat conventional
f2      winterwheat chisel plough
f3      winterwheat no till
f4      springwheat conventional
f5      springwheat chisel plough
f6      springwheat no till
f7      spring barley conventional
f8      spring barley chisel plough
f9      spring barley no till
f10      oats conventional
f11      oats chisel plough
f12      oats no till
f13      spring turnip rape conventional
```

```
f14     spring turnip rape chisel plough
f15     spring turnip rape no till
f16     silage
f17     hay
f18     grass
f19     potato
f20     peas
f21     sugarbeet (nut. values for "pressed")
f22     mixed grain conventional
f23     mixed grain  chisel plough
f24     mixed grain no till
f25     malting barley conventional
f26     malting barley chisel plough
f27     malting barley no till
/
  conv(j)
        /f1, f5, f7, f10, f13, f16, f17, f18, f19, f20, f21, f22, f25 /
        notill(j) no tillage crops
        /f3, f6, f9, f12, f15, f24, f27 /
        chplo(j) chisel plough crops
        /f2, f5, f8, f11, f14, f23, f26 /

*short version of purchased
        l(f) purchased fodder types
/
f28*f303
/
conc(f)  concentrates
/
        f1*f15,
        f19*f303
/

coars(j) coarsefodder
        /f16*f18/

cereals(j) / f1*f12, f22*f27 /
rapeseed(j) / f13*f15 /
tubers(j)   / f19, f21/
harvester(j)  / f1*f12, f22*f27,  f13*f15, f20 /

mitch(j) mitcherlich yield function crops / f1*f12, f20, f22*f27  /
quad(j)  quadratic yield function crops   / f13*f19, f21  /

        flw(primary) fallow types
/
        gfs     green fallow entitled to subsidies
        gfn     green fallow not entitled to subsidies
        bfs     bare fallow entitled to subsidies
        bfn     bare fallow not entitled to subsidies
        ffn     from fallow to forest with no subsidies
/
```

```
     i(primary) animal types
/
     ALA        lactating cow
     ADR        dry cow
     AHE        heifer
     ACA        calf
/

     n fodder characteristics
/
     ka     dry matter percentage
     ry     energy unit
     ryg    ry per kilogram
     oiv    protein units g per kg KA
     ndf    non digestive fiber g per kg KA
     pho    phosphorus content
     me     MJ per kg KA
     pvt    g per kg KA
     hvo    \%
     rv     g per kg KA
     rr     g per kg KA
     rk     g per kg KA
     tua    g per kg KA
     tu     g per kg KA
     p      € per kg KA
     dv     d-value (\% of digestible organic matter)
/
     y yield function coefficients
/
     a      Quadratic parameter a intercept
     d      Quadratic parameter b positive slope
     c      Quadratic parameter c for diminishing marginal
     m      Mitscherlich parameter m
     k      Mitscherlich parameter k
     b      Mitscherlich parameter b
     pyc1   yield coefficient for P fertilisers
     pyc2   yield coefficient for P fertilisers
     pyc3   negative yield coefficient of P stock
     pyc4   positive yield coefficient of P stock
     pcon   constant increase in P yield
     pcoe   crop type yield coefficient for P
     sry    saarelas fodder unit
     tillfac tillage yield factor
/
     lc load coefficients for P and N
/     r runoff
     e erosion
     o omega leaching based on different technology
     v delta leaching based on different technology
     nic nitrogen load coefficient
     psini initial phosphorus level
```

```
       PStm1 dummy for last period P level
/
       macon max resource constraints
/
       maxtr      maximum rape
       maxfa      maximum EU fallow
       maxsub      maximum sugar beet for subsidy
       maxgr      maximum grass with out preservation
       maxpot
/
      eqcon equalised constraints
/
      ha         arable land available for model farm
/
       micon   min resource constraints
/
       minfall      minimum EU fallow
/

       fcon  fallow resource constraints
/
       maxsubfall
       maxbarefall
*        mingrfall
       mincapfall
/
       spt(primary) spreading technologies
/
       sp      splash plate
       bs      band slurry speader
       is      injector slurry spreader
/
      splte(spt) subset of tech  /  sp, bs, is /

       stt storage technologies
/
       cc      concrete cover
       fc      flexible cover
       nc      no cover
/
       sttte(stt)   subset of storage tech    / cc /

set      count iterative increases in P load restriction / 1-1 * 1-32, N1-1 * N1-32,  P1-1 * P1-32 /
       countN(count) / N1-1 * N1-32 /
       countP(count) / P1-1 * P1-32 / ;

scalar   ite   iteration count                              /1/  ;

;

SCALARS
     farm_size      average farm size for 5304                    / 0  /
```

```
      dairy_share                                         / 0.5 /
* 5 \% rate used in the excel calculations
      discountr      Discount rate                              / 0.05 /
* LYPSYLEHMIEN_RUOKINNAN PERUSTEET.PDF - Juha Nousiainen, slides
       dim           Days in milk                             / 300 /
       diy           Days in year                             / 365 /
       heifer_oiv     Protein requirement per day per heifer       / 438.9 /
       heifer_ry      Energy requirement per day per heifer         / 5.2 /
*inlucluded in the purchased fodder prices
       trc           Transport and transaction cost                /0.01/
* Check numbers below (some of these just dummies)
       exp_coeff      the average difference between expriment and field /0.8/
       fodloss        loss of silage in storage                      /0.7/
       cow_w          cow weight in kilograms                        /600 /
       drycow_w
       heifer_w       heifer weight in kilograms
* Pro Agria 5.9.2003
       cowmeat        Kg of cow sold as meat                         / 240 /
       p_meat         constant in article = cow meat * meat_p
* Cow renewal every fith year
       calf_f         Renewal  rate of cows from heifers             / 0.4 /
* From Kauko personal communication
       calfmilk       Amount of milk per calf in kg                  / 140 /
* P\% from Antikainen, 2005, inputdata.xls:Pcont
       calf_pho                                         / 0.71 /
* P\% from Antikainen, 2005, inputdata.xls:Pcont
       cow_pho                                            / 0.71 /
*       meat_s         Subsidity of meat                 /  80  /
*Fuel consumption 51-60 kwh tracktor from Maidontuotanto30.xls:Konetyö (Koikkalainen)
       fuel_c         Fuel consumption   litre per hour                /7/
       loil_c         Lubricant consumption   litre per hour          /0.09/
       harvester_fuel_c        Fuel consumption   litre per hour                /11/
       harvester_loil_c        Lubricant consumption   litre per hour          /0.12/
       avspeed        average speed of transport  km per hour        / 20 /
       container_vol   volume of manure transport m3                 / 10 /
* Coefficient from Bosch et al. 2006.Reducing Phosphorus Runoff from Dairy Farms
      cowdigcoeff     digestibility coefficient(for manure dm)       / 0.65 /
*Cleaning water regional_inputdata.xls/dairycosts small sample obtained from other study
      cleaningwat     litre per day per cow                          / 20 /
*Kaukolta katsottu laiduntutkimus jossa sontaa ja virtsaa suunnilleen sama määrä eli sonnan paino
kerrotaan kahdella
*HUOM! Rotz et al 1999. wet-fecal to urea ratio 2.2 mature, 1.2 for heifers
       ureamultiplier   dry manure to urine                          / 2 /
       heiferman      Manure from heifers  m3 per year               / 15 /
       heiferdm       Max drymatter intake for heifers kg per day       / 15 /
*       manstor        Manure storage capacity             /  50   /
*       PMTm1          Initial manure amount (t-1)          /  5000   /
*LOAD PARAMETERS FROM load_paramters.gms - this for declaration
        PStm          average initial soil P stock mg per l
* Abatement iterations
        reduction      dummy reduction                               /-10/
        red_l          abatement rate                              /0/
```

```
        basetpload      initial for base p load
        tpinit          check tp in the beginning
        basenload       initial for base n load
        suflow          is the share of P surface flow Puustinen(SYKE)           /0.4/
        draflow         is the share of P drainage flow                         / 0.6/
        retention       no retention is assumed                                 /1/
*Source for average rainfall ...

*" Ohje kotieläintalouden ympäristönsuojelusta 1998, Ympäristöministeriö, page 11"
        dc_stor_req     manure storage size requirement for 12 months per dairy cow  /24/
        h_stor_req      manure  storage size requirement for 12 months per heifer    /15/
        cowman          Manure from cows  m3 per year                 / 24 /
        storageheight   Height (m) of manure storage                  / 4 /
        storagem2       storage surface area m2                       / 225 /
*Viljavuuspalvelu lantatilastot 2009 (2000-2004) (also from "Lannan käsittely ja käyttö", 2009 p. 19)
        viljavuusp_Pcont    fosfori                          / 0.5 /
        viljavuusp_Ncont    kokonaistyppi                    / 3 /
        mload_t         time (minutes) taken to load the manure per m3        / 0.3 /
        redfield        redfield ratio (mass to mass)                 / 7.2  /
*http://www.mtt.fi/mtts/pdf/mtts74.pdf
        meanMilkYield2003   Pohjanmaan  keskimaituotos 40 lehmää 2003             /   8200  /
        belowMedianFieldShare
        HartBradyNcost      50 \% N reduction approximately 10 \% from Hart and Brady 2002 JEM / 0.1  /
        HartBradyNcostComparison      applying  Hart and Brady 2002 JEM to these results
        Helin2006NcostPerHa     50 \% N reduction                              / 99  /
        Turner1999NcostPerKg    Finland 57–220 SEK per kg N                    /  14 /
;
parameters KalaDairyTotalFieldArea, KalaDairyBelowMedianFieldArea, KalaDairyMedianDistance,
DairyAboveMedianDistance, DairyBelowMedianDistance, MeanDairyFarmSize,
dairyFields(dst), distance(dst)  average distance of fields in km ;

*GET THE DATA FROM THE GDX FILE CREATED FROM ACCESS DATABASE
$gdxin D:\gamsdir\dairyfarm2011distance\KalaDairyData.gdx
$load KalaDairyTotalFieldArea KalaDairyBelowMedianFieldArea KalaDairyMedianDistance
DairyAboveMedianDistance DairyBelowMedianDistance MeanDairyFarmSize

belowMedianFieldShare =  KalaDairyBelowMedianFieldArea/KalaDairyTotalFieldArea ;
farm_size = MeanDairyFarmSize / 10000 ;
dairyFields('dstBM') = (farm_size * KalaDairyBelowMedianFieldArea/KalaDairyTotalFieldArea  )  ;
dairyFields('dstAM') = (farm_size * [1 - KalaDairyBelowMedianFieldArea/KalaDairyTotalFieldArea]  ) ;

*Assigning values from data to the model
distance('dstBM') =  DairyBelowMedianDistance /1000 ;
distance('dstAM') =  DairyAboveMedianDistance /1000;

*Assume heifers on average same as Nennich 2005 thinner than cows
heifer_w = cow_w  * 437/630  ;
drycow_w = cow_w * 755/630   ;

*CONSTRAINT PARAMETERS---------------------------------------------
$set cmdfile nuval.txt
$onecho > "\%cmdfile\%"
```

```
I="D:\gamsdir\dairyandcrop5304\inputdata.xls"
R1="nuval3!d3:t306"
O1="D:\gamsdir\dairyfarm2011distance\nuval.inc"
R2="tecon!d3:i30"
O2="D:\gamsdir\dairyfarm2011distance\tecon.inc"
R4="bacon!d3:e30"
O4="D:\gamsdir\dairyfarm2011distance\bacon.inc"
$offecho
$call =xls2gms \at"\%cmdfile\%"

table nuval(f,n)  Nutritional characteristics of fodder and crops and price of fodder
$include nuval.inc
;
table tecon(j, macon) Technological constraints
$include tecon.inc
;
*table facon(j, micon) Minimum fallow criteria
*$include facon.inc
*;
table bacon(j, eqcon) Maximum fallow criteria
$include bacon.inc
;

*LOAD PARAMETERS--------------------------------------------------------------
parameters
 leach(j,lc)          Load coefficients
 loadcoef_N(j)         N kg per ha for each plant
 loadcoeff_N(flw)       N kg per ha for fallow
 loadcoef_DRP(j)        DRP kg per ha for each plant
 loadcoeff_DRP(flw)      DRP kg per ha for fallow
 loadcoef_PP(j)         PP kg per ha for each plant
 loadcoeff_PP(flw)       PP kg per ha for fallow
 veps_N_2002           5304 total N load (kg) for 2002
 veps_TP_2002          5304 total P load (kg) for 2002
 land_ha              area of modeled land
 drp_load             dissolved reactive phosphorus load
 pp_load              particle phosphorus load
 DRP_TP_share          share of drp of total p
 bio_coef
;
$GDXIN load_param.gdx
$load loadcoef_N loadcoeff_N loadcoef_DRP loadcoeff_DRP loadcoef_PP loadcoeff_PP veps_N_2002
veps_TP_2002 land_ha DRP_TP_share PStm bio_coef
;
*Fixed in the load parameter file
*Weighted average based on municipal mean P values
*Pstm = 10.32113976  ;

*EXPECTED LOAD FOR HA MULTIPLIED BY THE FARM SIZE AND COEFFICIENT TO MAKE IT NOT BINDING
leach(j,"nic") = loadcoef_N(j) ;
*This initial value gives good profit results -SENSITIVE!
basetpload = veps_TP_2002/land_ha*farm_size*1.5 ;
```

```
*basetpload = 40 ;
basenload = veps_N_2002/land_ha*farm_size*2  ;
tpinit = veps_TP_2002/land_ha*farm_size*1.5 ;

*YIELD PARAMETERS---------------------------------------------------------------
parameters
yc(j,y) Old experimental coefficients and P-coefficients
yieldC2(j,y) Yield function coefficients for both mitscherlich and quadratic for Ostrobothnia C2
qyieldC2 N yield parameters for quadratic specification
myieldC2 N yield parameters for mitscherlich specification
quality_table(j)
quality_constraints(j)  minumum nitrogen fertilisation levels
;
$GDXIN D:\gamsdir\yield_parameters\ostrobothnia_yields_param.gdx
$load yc qyieldC2 myieldC2 yieldC2 quality_constraints quality_table
;
quality_constraints(j) =   quality_constraints(j)  ;

* PRICE DATA--------------------------------------------------------------------
 scalars
* Maataloustilastollinen vuosikirja 2004, average with standard milk  33.48, quality 1.39, adjustment
payment 1.81
*SHOULD BE CONVERTED TO KILO PRICE ?
       milk_p        Price of milk litre                      /0.3668  /

*MORE detailed milk pricing   Jokela_1998, Heikkilä_2006 p. 30 (need to divide by 100?)
      fat_p     price in cents for tenth of fat \%                /2.4 /
      prot_p    price in cents for tenth of protein \%             /6.5/
      milkq_p   price in cents for milk quality per litre          /1.447 /
      milks_p   price in cents for season per litre              /1.225 /
* price from Niemi & Ahlsted 2005, p. 34, producer price 2003
      meat_p    Price of meat kg                            / 1.8565  /
*Lehmänlihan ennakoitu keskimääräinen tuottajahinta noin vuonna 2007 noin 1,47 euroa/kg
tuottopehtori
*      meat_p    Price of meat kg                            / 1.47  /
* Calf sale price 2003, 60kg calf sale price, Pro Agria 5.9.2003  - the calf price has not changed in the new
2007 tuottopehtori calculations
       calf_p        Calf price per 60 kg calf           / 110 /
*       mill_cost    milling costs for grain              /0.02/
*Mill costs bias the crop vs dairy comparison
       mill_cost    milling costs for grain              /0/
 PARAMETERS
*Suomen Maatalous 2004, p31  & Maatilatilastollinen vuosikirja 2006, p 167
       c_price(j) 2003 commodity price euro per kg (converted to price per DM later)
/
*       f0           0.00001
       f1*f6         0.12566
       f7*f9         0.10557
       f10*f12        0.09221
       f13*f15        0.237
* pienet hinnat annettu nollan sijaan, jottei tulisi nollalla jako virhettä
* decimaalinkin siirto saattaa tosin vaikuttaa tuloksiin
```

```
	f16	0.0001
	f17	0.0001
	f18	0.0001
*More recent data (2006) not accurate as combines two estimates 75\% for starh and 0.22 price per starch
kilo
	f19	0.0357
*Fodder peas MTK 2007
	f20	0.180
	f21	0.054
*Price calculated as 1:1 barley to oats
	f22	0.0995
	f23	0.0995
	f24	0.0995
	f25	0.1347
	f26	0.1347
	f27	0.1347
/
;
parameter transport_cost(j) Maatilatilastollinen vuosikirja 2004 mallasohran keskihinnasta vähennetty 0.01
kuljetuskustannuksiin ;
transport_cost(cereals) = 0.01 ;
c_price(j) = c_price(j) - mill_cost  ;
c_price(cereals) = c_price(cereals) - transport_cost(cereals) ;
c_price(j) =   c_price(j)/(nuval(j, "ka")*0.01)     ;
*Fixing the prices of grass, silage and hay on equal levels again
*c_price("f0") =  0.00001   ;
c_price("f16") =  0.001    ;
c_price("f17") =  0.001    ;
c_price("f18") =  0.001    ;

display c_price ;

scalars
	grass_NP_ratio combined from 2 different tuottopehtori fertilisation types weighted average
	nk1_fert_ratio
	y3_2003_n_content /  0.2  /
	nk1_2003_n_content / 0.2 /
	y1peruna_2003_n_content  / 0.08 /
	yjuurikas_2003_n_content  / 0.15 /
	p_y3_2003 spring cereal "y3-lannoite" 20-3-8 NPK\% price per t / 206.84   /
	p_y1peruna_2003 spring cereal "y3-lannoite" 8-5-19 NPK\% maatalouden mallilaskelma price per t /
340   /
	p_yjuurikas_2003    sugar beet "y-lannoite" 15-4-8 price per t  / 260  /
	p_nk1_2003 grass fertiliser 20-0-7 /  200.96  /
	p_price cost of mineral fertilizer euro per kg  /  1.22 /
	n_price cost of mineral fertilizer euro per kg  / 0.75   /
	np_priceBarley         ;

nk1_fert_ratio = 395 /(490 + 395)    ;

parameter np_price(j) price for kg of compound fertilisers 2003
*/
```

```
*       f0     pasture
*       f1*f6   wheat        y3
*       f7*f9   barley       y3
*       f10*f12 oats         y3
*       f13*f15 turnip rape    y3
*       f16     silage        nk1 and y3
*       f17     hay           nk1 and y3
*       f18     grass         nk1 and y3
*       f19     potato        peruna y1
*       f20     peas         y3
*       f21     sugarbeet     juurikkaan y
*       f22*f24 mixed cereal  y3
*       f25*f27 malt barley   y3
*/
;
np_price(j) =   (p_y3_2003 /  y3_2003_n_content) / 1000       ;
np_price("f16") =  nk1_fert_ratio  * p_nk1_2003 /  nk1_2003_n_content / 1000  + (1 -  nk1_fert_ratio) *
p_y3_2003 /  y3_2003_n_content / 1000  ;
np_price("f17") =  nk1_fert_ratio  * p_nk1_2003 /  nk1_2003_n_content / 1000  + (1 -  nk1_fert_ratio) *
p_y3_2003 /  y3_2003_n_content / 1000  ;
np_price("f18") =  nk1_fert_ratio  * p_nk1_2003 /  nk1_2003_n_content / 1000  + (1 -  nk1_fert_ratio) *
p_y3_2003 /  y3_2003_n_content / 1000  ;
np_price("f19") =  (p_y1peruna_2003 /  y1peruna_2003_n_content) / 1000       ;
np_price("f21") =  (p_yjuurikas_2003 /  yjuurikas_2003_n_content) / 1000       ;

np_priceBarley =  np_price('f7') ;

p_meat = cowmeat * meat_p           ;

*SUBSIDIES--------------------------------------------------------------------
*SUBSIDIES FOR 2003
scalars
* From the "sitoutumus ehdot 2003"
      env_subs      basic environmental subsidy for animal farm  / 116.89 /
      mod_fert_subs                                      / 13.46/
* plant cover subsidies not available for animal farms
      plant_cover_subs                                   /23.55/
* Subsidy € per ha of field for covering the manure storage with concrete or lighter means
      man_cover_subs                                     /23.55 /
* price subsidy C2 region 2003
* SHOULD BE CONVERTED TO KILO SUBSIDY
      milk_s        Subsidity of milk litre  2003 C2              /0.094  /

parameters
* Fallow subsidy for C2-region 2003, Niemi & Ahlsted 2005 (CAP, LFA, LFA sup, youngfarmer, general C2)
      fall_subs(flw)      subsidy for fallow
/
gfs    389
gfn     0
bfs     145
bfn     0
ffn     0
```

```
/
* tuet C2 alueelta (inputdata.xls:SUBSC) not including environmental subsidies
       subsc(j) crop subsidies per ha (2003)C2 region
/
*f0       0
f1      608.7
f2      608.7
f3      608.7
f4      608.7
f5      608.7
f6      608.7
f7      512.7
f8      512.7
f9      512.7
f10      512.7
f11      512.7
f12      512.7
f13      570.7
f14      570.7
f15      570.7
f16      511
f17      366
f18      366
f19      439
f20      483
f21      638
f22      521
f23      512.7
f24      512.7
f25      587.7
f26      587.7
f27      587.7
/
;
*COSTS------------------------------------------------------------------------
scalars
* Capital cost of animal operation for 30 cow farm (capital, electricity, general expense, 60kw tracktor)
      fix_a        Animal shelter and milk machinery costs + tracktor capital cost  per year        /15692 /
* Labour costs per dairy cow (assuming that costs decrease as number of animals does)
      labour_c       Labour costs of animal feeding & care and general management                 /1274 /
* From Maidontuotanto_30:Työkooste G213
      labour_h       Hours assumed to be needed for animal care in year (including planning          / 3121.1 /
      lab_h_pc       Estimate of labour hours per dairy cow on 30 cow farm                        / 100 /
*Labour price from  Tuottopehtori 2003
      lab_p         Labour price                                             / 11.9 /
* lab_p         Labour price                              / 13.05 /
*Price of land from regional_inputdata.xls:dairy_farms C1 region discounted with 5\%
      land_p                                                           / 150 /
*Fuel prices from Maidontuotanto30.xls:Lähtötiedot (Koikkalainen) taken from 2005
      fuel_p         Fuel price € per l 80\% summer type 20\% winter type                         / 0.66 /
      loil_p        Lubricant oil price                                        / 1.23 /
* Kasvitila mallin kesannonhoitokustannus
```

```
;
lab_h_pc = labour_h/30     ;
labour_c = lab_h_pc * lab_p ;
parameter fix_c(j,dst) fixed costs euro per ha of field   ;

fix_c('f1',dst) =       436  ;
fix_c('f2',dst) =       433    ;
fix_c('f3',dst) =      423     ;
fix_c('f4',dst) =       436    ;
fix_c('f5',dst) =      433     ;
fix_c('f6',dst) =      423    ;
fix_c('f7',dst) =      436    ;
fix_c('f8',dst) =      433    ;
fix_c('f9',dst) =      423    ;
fix_c('f10',dst) =      436  ;
fix_c('f11',dst) =      433   ;
fix_c('f12',dst) =       423  ;
fix_c('f13',dst) =       436  ;
fix_c('f14',dst) =      433   ;
fix_c('f15',dst) =       423  ;
fix_c('f16',dst) =      382  ;
*Grass ley and hay should be bit cheaper than silage
fix_c('f17',dst) =       382  ;
fix_c('f18',dst) =      382  ;
*assumed to be same as sugar beet
fix_c('f19',dst) =      711  ;
* assumed to be same as barley
fix_c('f20',dst) =       436 ;
fix_c('f21',dst) =      711 ;
*assumed to be the as for barley
fix_c('f22',dst) =      436 ;
fix_c('f23',dst) =      433 ;
fix_c('f24',dst) =     423 ;
fix_c('f25',dst) =     436 ;
fix_c('f26',dst) =     433 ;
fix_c('f27',dst) =     423 ;

fix_c(coars,dst) = fix_c(coars,dst)  + (fuel_p*fuel_c+loil_p*loil_c+lab_p) * (distance(dst)*10/avspeed) +
(fuel_p*harvester_fuel_c+loil_p*harvester_loil_c+lab_p) * (distance(dst)*2/avspeed) ;
fix_c(harvester,dst) = fix_c(harvester,dst)  + (fuel_p*fuel_c+loil_p*loil_c+lab_p) * (distance(dst)*2/avspeed)
+ (fuel_p*harvester_fuel_c+loil_p*harvester_loil_c+lab_p) * (distance(dst)*2/avspeed)  ;
fix_c(tubers,dst) = fix_c(tubers,dst)  + (fuel_p*harvester_fuel_c+loil_p*harvester_loil_c+lab_p) *
(distance(dst)*2/avspeed) + (fuel_p*fuel_c+loil_p*loil_c+lab_p) * (distance(dst)*10/avspeed)  ;

*Test if can force the silage to closer area by increasing operating costs at fields further away
*fix_c(coars,'dstAM') = fix_c(coars,'dstAM') * 2 ;

parameter fall_cost(flw)      cost of green fallow per ha
/
gfs   166
gfn    166
bfs    50
```

```
bfn    50
ffn    300
/
;
*Prices for the sensitivity analysis
parameter prices(primary)            ;
prices('Nfertiliser') = n_price       ;
prices('Pfertiliser') = p_price       ;
prices('NPfertiliser') = np_priceBarley       ;
prices('milk') = milk_p       ;
prices('meat') = meat_p        ;
prices('fuel') = fuel_p       ;
prices('lubricant') = loil_p        ;
prices('labour') = lab_p        ;
prices(f) = nuval(f,'p')        ;
prices(j) = c_price(j)        ;

*Cost increases from 2003 to 2007---------------------------------------------
*Fert price increase from Suomen Maatalous 2007 p. 88, 2000 as a base year, 2003 = 104.2, 2006 = 116.1
scalar    pindex2006 increase of price index compare 2003 and 2006 /1.12/
          pindex2007 increase of price index compare 2003 and 2007 /1.156/
*From http://www.agronet.fi/cgi-bin/mkl/julk/malli.cgi?taulu=maito2007&alue=C2
;
*fix_c(j) * pindex2007   ;
*
*-----------------------------------------------------------------------------
Parameters
*From inputdata.xls:FIX_A, based on calculations done in regional data and Koikkalainen model
maidontuotanto30
*From regional_inputdata.xls:dairy_costs, calculated from required volume in the
ohje_kotieläintalouden_ympäristönsuojelusta_1998
*MMM-RMO E2.1- 1(5) neliökustannukset
m_stor_cover_m2(stt) Cost of cover only (do not use on top of storage costs below as already included)
/
cc      28.9
fc      14.45
nc      0
/
m_stor_A(stt) Cost of manure storage per dairy cow
/
cc      56
fc      64
nc      68
/
m_stor_m3(stt) Cost of manure storage per m3
/
cc      1.7
fc      1.6
nc      1.5
/
m_stor_P(stt)  Cost of manure storage per kg P given constant P content
/
```

```
cc      3.5
fc      3.2
nc       3
/
;
*From Haataja 1998, p. 94
table spread_t(spt, j) time in minutes required to apply m3 manure for each spreading tech
       f1*f27
sp      0.5
bs      0.5
is      1.5
;
PARAMETERS
*From regional_inputdata.xls:dairy_costs
spread_cap(spt) capital costs of spreading manure (tractor included in the fixed_c)
/
sp   2930
bs   4270
is   4610
/
*FOR ONLY comparison with the endogenous costs
*m_cost(spt) Cost of manure spreading tight to P content of 0.56 per m3
*/
*sp   6.8
*bs   7.9
*is   8.2
*/

*m_app from excel
*f_app(j) Costs of fertiliser application per kg
*/
*f1*f27 0
*/

*THESE ONLY FOR DYNAMIC ANALYSIS WHICH IS NOT DONE YET
parameter PSt0(j)  To contain P level t+1  ;
PSt0(j) = PStm ;
parameter PStm1(j) dummy values for last period p ;
PStm1(j) = PSt0(j)       ;
*---------------------------------------------------

PARAMETER pmax(j) Maximum base fertilisation level as intial p level
*note that pmax is 30 for silage, but for this parameter 15 for all more suitable
/
f1*f27       15
/
*Environemental subsidy guidelines http://www.finlex.fi/fi/laki/alkup/1995/19950768
*Grass plants set to equal though (f17 & f18 were 150 originally)
PARAMETER nmax(j) Recommended base fertilisation level as baseline N-load level
/
*f0     90
f1*f3   120
```

```
f4*f6   100
f7*f12  90
f13*f15 100
f16    180
*heinälle jos odelma korjataan muuten 90
f17    180
f18    180
f19   120
*for Karita peas
f20   50
f21   80
f22*f27 90
/
        P_ratio(j)   Kg of P per kg of N
/
*f0      0.1
f1*f3    0.15
f4*f6    0.15
f7*f12   0.15
f13*f15  0.15
f16      0.14
f17      0.14
f18      0.14
f19      0.625
f20      0.15
f21      0.27
f22*f27  0.15
/   ;

*DOES NOT MAKE SENSE TO CALCULATE THE GRASS P FROM Y, SINCE THE MANURE P AT ANIMAL FARM IS
A FACTOR DETERMINING THE P RATIO IN Y
grass_NP_ratio = nk1_fert_ratio  * 0 + (1 -  nk1_fert_ratio) * (3 / 20) ;

* Seed parameters-----------------------------------------------------------------
PARAMETERS
*input_data.xlc:Pcont parameters from REGGAE balance calculations
        seedN(j) seed nitrogen kg per kg seed (wet weight)
/
*        f0      0.005
        f1*f12   0.02
        f13*f15  0.03
        f16      0.007
        f17      0.008
        f18      0.008
        f19      0.003
        f20      0.03
*assume same as potato
        f21      0.002
        f22*f27  0.02
/

*input_data.xlc:Pcont parameters from REGGAE p -balance calculations, also Saarela 1995, p. 22
```

```
        seedP(j) seed phosphorus g per kg seed
/
*        f0      5
        f1*f12   3.5
        f13*f15  8
        f16*f18  5
        f19      2.3
        f20      4
*assume same as potato
        f21     2.3
        f22*f27  3.5
/
;
*USE SEED NP RATIO FOR GRASS
*Using the seed ratio would lead to very high P fertilisation level
*grass_NP_ratio = seedP('f16')/ (seedN('f16') * 1000) ;
*P_ratio('f16') = grass_NP_ratio   ;
*P_ratio('f17') = grass_NP_ratio   ;
*P_ratio('f18') = grass_NP_ratio   ;

parameters
        seeduse(j) amount of seed used in kg per ha
/
*        f0           9
        f1*f3        217
        f4*f6        205
        f7*f9        201
        f10*f12      182
        f13*f15      10
        f16          9
        f17          9
        f18          9
*Tuomisto Jussi phone call 20.8.2006 range 2000-2500
        f19          2250
*Herneen viljelyn uusi opas kansiossa yield
        f20          396
*Dummy for sugar beet 500
        f21          10
*Same as barley
        f22          201
        f23          201
        f24          201
        f25          201
        f26          201
        f27          201
/
;
*Conversion to N content in kg DM
seedN(j) = seedN(j) / (nuval(j, "ka" )   / 100) ;

* Resource Limits ---------------------------------------------
PARAMETERS
```

```
        valcon(macon)         right hand side for resource endowment and technology constraints
        hacon(eqcon)
        mini(micon)          minimum amount of fallow required by EU
        fallLimits(fcon)      right hand side for fallow constraints
        facon(flw, fcon)       left hand side for fallow constraints
;
hacon("ha") = farm_size      ;
mini("minfall") = farm_size/10 ;
valcon("maxfa") = farm_size/2 ;
valcon("maxtr") = farm_size/3 ;
valcon("maxgr") = farm_size/2 ;
valcon("maxsub") = 1 ;
mini("minfall")  =  farm_size/10 ;
fallLimits("maxsubfall")  = farm_size * 0.5          ;
fallLimits("maxbarefall") = farm_size * 0.3          ;
*fallLimits("maxwtld") = farm_size * 0.5          ;
*fallLimits("mingrfall")  = farm_size     ;
fallLimits("mincapfall") = farm_size * (-0.1)    ;
facon('gfs', 'maxsubfall')  = 1 ;
facon('bfs', 'maxsubfall')  = 1 ;
facon('gfn', 'maxsubfall')  = 0 ;
facon('bfn', 'maxsubfall')  = 0 ;
facon('bfs', 'maxbarefall') = 1 ;
facon('bfn', 'maxbarefall') = 1 ;
*facon(flw, 'mingrfall')
facon(flw, 'mincapfall')    = -1 ;

* Manure Storage: Nutrient losses
*Manure storage volume parameters
parameters
 avrainfall(stt)      average rainfaill mm per year
 /
       cc 1
       nc 650
       fc 600
 /
 evapor(stt)         evaporation fraction from uncovered manure storage
/
       cc 0.5
       nc 0.15
       fc 0.10
/

*THESE VALUES APPROXIAMATES BASED ON AMMONIUM STORAGE LOSSES in PALVA ET AL 2009
parameters

       storlossNU(stt) storage loss coefficient of NU
/
       cc     0.95
       fc     0.9
       nc      0.6
/
```

```
        storlossNM(stt) storage loss coefficient of NM
/
        cc     0.95
        fc     0.9
        nc     0.6
/
        storlossPM(stt) no storage loss coefficient of PM
/
        cc     1
        fc     1
        nc     1
/
*       storinvest(stt) capital costs of storage
*/
*       cc     500
*       fc     50
*       nc     0
*/
parameters
*NOTE! Loss coefficients for testing purposes
apPMloss(spt)  No phosphorus is lost on transport & application
/
        sp     1
        bs     1
        is     1
/
* Assume that 50\% of manure N in non-soluable form
*Kemppainen, 1989 p.177 in Mattila 2006 p.10 slurry non-soluable 56\%
* Some of the remaining 50\% might be converted back to soluable form by land microbes
apNMloss(spt) Loss of nitrogen (manure) in application & transport
/
        sp     0.5
        bs     0.5
        is     0.5
/
* On assumption that all N is soluable(ammoniacal) in urea and based on Haataja, 1998 (Mattila, 1997b)
*  This is lost as ammonium, what should one assume for the fate of this lost N
apNUloss(spt) Loss of nitrogen (urine) in application & transport
/
        sp     0.79
        bs     0.87
        is     0.99
/


depthFactor(spt)

/
        sp  1
        bs  1
        is  0
/
```

```
foddet(j)   Difference between eaten and grown amount of forage
* Silage based on presentation by Huhtanen (did not specify which)
/
      f1*f15    0.9
      f16       0.7
      f17       0.7
      f18       0.7
/
*ENVIRONMENTAL SUBSIDY REQUIREMENTS-------------------------------------------
scalar nitrateLimit  Limit set to manure N in the EU ditrate directive /170 / ;
parameters
volumeRequirement(i) required slurry capacity per animal in Nitraattiasetus 931 2000 (nitrate directive)
/
      ALA    24
      AHE    15
      ACA    4
/
Nenvbasic2003(j) Limits  nitrogen per ha in the basic subsidy scheme  2003
/
*       f0     150
      f1*f3   120
      f4*f6   100
      f7*f12  90
      f13*f15 100
      f16     180
      f17     180
      f18     180
      f19       80
      f20       90
      f21       120
      f22*f24       90
      f25*f27       90
/
Penvbasic2003(j) Limits  phosphorus per ha in the basic subsidy scheme  2003
/
*f0 put decreased from original 20
*       f0     15
      f1*f3   15
      f4*f6   15
      f7*f9   15
      f10*f12 15
      f13*f15 15
      f16     30
      f17     15
      f18     20
      f19     40
*assumed 15
      f20     15
      f21     30
      f22*f24 15
      f25*f27 15
/
```

```
Nenvmod2003(j) Limits  nitrogen per ha in the basic subsidy scheme  2003
* Limits for max yield levels, Keski-Suomi, eloperäiset maat
* heinäkasveille jako etelä-pohjoinen, valittu pohjoinen
/
*        f0      150
       f1*f3   90
       f4*f6   100
       f7*f9   90
       f10*f12 90
       f13*f15 100
*multiyear grass 100+100+40
       f16     240
       f17     180
       f18     240
*for starch potato yield of 40 tonnes, 30 tonnes would mean 90
       f19       120
*karita pea
       f20       50
* For subeet if "naatteja ei kylvetä maahan"
       f21      140
*oletettu 90
       f22*f24    90
       f25*f27     80
/
*"Tarkennettu fosforilannoitus 4000 tn hehtaari sadoille, fosfori luokka hyvä"
Penvmod2003(j) Limits  phosphorus per ha in the basic subsidy scheme  2003
/
*        f0      15
       f1*f3   10
       f4*f6   10
       f7*f9   13
       f10*f12  5
       f13*f15 10
       f16     30
       f17     15
       f18     20
       f19     35
       f20     10
       f21       40
       f22*f24    9
       f25*f27    13
/
Pmanure2007(j) Limits for manure phosphorus per ha in the basic subsidy scheme  2007
/
*        f0      15
       f1*f15  15
       f16*f18 20
       f19*f27 15
/
Nenvbasic2007(j) Limits  nitrogen per ha in the basic subsidy scheme  2007 karkeat kivennäismaat Etelä- ja
Keski-Suomi
/
```

```
*        f0      200
       f1*f3   110
       f4*f6   110
       f7*f12  90
       f13*f15 100
       f16     200
       f17     100
       f18     200
*satotaso 35 tn per ha
       f19     105
       f20       100
       f21       140
       f22*f24       90
       f25*f27       90
/
Nenvred2007(j) Limits  nitrogen per ha in the measure: reduced fertilisation  2007 karkeat kivennäismaat Etelä- ja Keski-Suomi
/
*        f0      150
       f1*f3   100
       f4*f6   100
       f7*f9    80
       f10*f12  70
       f13*f15  90
       f16     180
       f17     150
       f18     90
*satotaso 35 tn per ha
       f19     80
       f20     40
       f21     120
       f22*f24  80
       f25*f27  80
/
Penvbasic2007(j) Limits  phosphorus per ha in the basic subsidy scheme  2007 phosphorus class good
/
*        f0      0
       f1*f3   8
       f4*f6   8
       f7*f9   10
       f10*f12  5
       f13*f15 8
       f16     30
       f17     15
       f18     12
       f19     35
       f20     10
       f21       26
       f22*f24    7
       f25*f27    10
/
;
```

*fallow 5 kg when established

* ENDOGENOUS VARIABLES --

VARIABLES

Z   Total annual producer surplus for modelling period euro
X(j,dst)   Activity level ha
Xfa(flw,dst)   Fallow (ha)
A   Number of animals
FF(l)   Fodder kg DM per day per cow during lactation cycle (purchaced)
OFF(l)   Fodder kg DM per day per cow outside lactation cycle (purchaced)
EF(j)   Fodder kg DM per day per cow during lactation cycle (farmed)
OEF(j)   Fodder kg DM per day per cow outside lactation cycle (farmed)
HEF(j)   Fodder kg DM per day per heifer   (farmed)
HFF(l)   Fodder kg DM per day per heifer   (purchaced)
PF(j,dst)   phosphorus fertiliser applied on the field kg per ha
PM(j,dst)   Phosphorus from manure to plant j
PMT(stt)   Phosphorus manure in storage total kg per t
NUT(stt)   Nitrogen in urea in storage
NMT(stt)   Nitrogen in manure in storage
PMS(spt,dst)   Phosphorus in manure before dispersal losses (dispersed with tech spt at distance dst)
NMS(spt,dst)   Nitrogen in manure before dispersal losses   (dispersed with tech spt at distance dst)
NUS(spt,dst)   Nitrogen in urea before dispersal losses   (dispersed with tech spt at distance dst)
PS(dst)   phosphorus stock mg per litre
DRP(j,dst)   Dissolved reactive phosphorus
PP(j,dst)   Particle phosphorus
NC(j,dst)   Nitrogen loss from crop land
FDRP(flw,dst)   Dissolved reactive phosphorus from fallow land
FPP(flw,dst)   Particle phosphorus from fallow land
FAN(flw,dst)   Nitrogen loss from fallow land
PU(j,dst)   phosphorus removal by plants kg for plant
CY(j,dst)   crop yield per ha kg DM in year
MY   milk yield kg per day per cow
MF   milk fat yield g per day per cow
MP   milk protein yield g per day per cow
PSt1(j,dst)   Phosphorus stock for next period
B(j)   Buffer
NF(j,dst)   Nitrogen fertiliser kg per ha of plant j
NM(j,dst)   Nitrogen from manure kg per ha of plant j
NU(j,dst)   Nitrogen from urea kg per ha of plant j
RM   Ratio of N to P
UM   Ratio of manure over urea N
UoM   Ratio of urea over manure N
PoN   P over N in manure
PCT   Ratio of P per m3
NCT   Ratio of N per m3
NUCT   Ratio of N per m3 of urea
MVS(stt)   Manure volume in m3 for each storage type
MVP(spt,dst)   Manure volume in m3 for each spreading tech for each distance
MVS1   Manure volume in m3 for single storage technology
MVP1   Manure volume in m3 for single dispersal technology
MSA(stt)   Surface area of manure storage

```
      RVM(stt)        Rain volume
*       M3(j,dst)       Manure volume dispersed on each distance
;
FREE VARIABLES Z ;
POSITIVE VARIABLES X, Xfa, A, FF, EF, OFF, OEF, HEF, HFF,
  PF, PM, PMT, PS, DRP, PP, PU, FDRP, FPP, FAN, CY, MY, MF, MP,
  NM, NF, NU, NMT, NUT, PMS, NMS, NUS, RM, UM, MVP, MVS, PCT, NCT, NUCT, MSA, RVM, M3, PoN ;

*INITIAL VALUES
 parameters
 ninit(j) unconstrained profit max N fertilizer dose, X_init(j,dst) for initial land use, MVP_init(spt,dst) initial
manurevolume, A_init initial animals,
 n1(j) normal or prof max N dose for individual quad crops without constr
 n2(j) normal or prof max N dose for individual mitsch crops without constr
 optimalInitYields(j)
 ;
* n1(j) $(qyieldC2(j,"a") AND qyieldC2(j, "d") AND qyieldC2(j, "c") NE 0)
*              =  [prices('NPfertiliser')/(prices(j)) - qyieldC2(j,"d")] / [2 * qyieldC2(j,"c")]   ;
* n2(j) $(myieldC2(j,"m") AND myieldC2(j, "k") AND myieldC2(j, "b") NE 0)
*              =  log[(prices(j) * myieldC2(j, "m")* myieldC2(j, "k") * myieldC2(j, "b"))
*                     / prices('NPfertiliser')] / myieldC2(j, "b")   ;
*As price of silage, grass and hay set to very low use environmental subsidy levels instead
*n1("f0") = 90 ;
* n1("f17") = 150 ;
* n1("f18") = 180 ;
* n1("f16") = 180 ;

* VARIABLE INITIALIZATION------------------------------------------------------
 A_init = 20 ;
 MVP_init(splte,dst) =  A_init * 20 ;
*Revoked for each scenario
$include variable_init.gms

*AND BOUNDS ---------------------------------------------------------------------------
*29.8.2007 tested with 3 instead 2  -> seems to solve the domain error problem of ninit
NF.up(j,dst) = ninit(j) * 3     ;
NF.up(coars,dst) = 250    ;
NM.up(j,dst) = 500    ;
NU.up(j,dst) = 1000    ;
PS.up(dst) = 100           ;
PS.lo(dst) = PStm          ;
*PM.L("f17") = 500         ;
PM.up(j,dst) = 250         ;
PF.up(j,dst) = 250         ;
*PM.L(j,dst)   = pmax(j)      ;

*Upper limits based on environemntal subsidy system ----------------------------
*NF.up(j) = Nenvbasic2003(j)     ;
*NM.up(j) = Nenvbasic2003(j)     ;
*NU.up(j) = Nenvbasic2003(j)     ;
*PM.up(j) = Penvbasic2003(j)     ;
*PF.up(j) = Penvbasic2003(j)     ;
```

```
*Manual scaling for fertilser variables (added 14.8.2007)---------------------
*PM.lo(j,dst) = 0.3    ;
*PF.lo(j) = 0.3    ;
*NM.lo(j,dst) = 1    ;
*NU.lo(j,dst) = 0.002    ;
*NU.lo(j,dst) = 0.002    ;
*---------------------------------------------------------------------------
X.up(j,dst) = dairyFields(dst)          ;
Xfa.up(flw,dst) = dairyFields(dst)           ;
* WORKED FOR EQUALING N and P baselines
*X.fx("f0",dst) = 0     ;
*PM.fx("f0",dst) = 1     ;
*29.8.2007 Grass fix to see p and n constraints would share the common baseline  -27.5.2011 Removed the grass fix
*29.8.2007 Hay fix to see p and n constraints would share the common baseline -> DOMAIN ERROR iN NON-LINEAR
*X.up("f16",'dstAM') = 0     ;
X.fx("f17",dst) = 0     ;
X.fx("f18",dst) = 0     ;
X.fx('f21',dst) =  0    ;
* Potato constraint added after energy constraint for dairy cow makes potato farming most profitable under P load (but not under N load)
X.fx("f19",dst) = 0    ;
*A.fx =    40          ;
* 28.8.2008 tested if would work with above 20 values -> DOMAIN ERROR FOR SOME REASON
A.up = 30              ;
A.lo = 5               ;
* Removal of two mash based products (added in 27.8.2007)   -CAUSE DOMAIN ERROR  try increasing price instead
*20.2. attempt without price increases successfull
*nuval("f238","p") =  nuval("f239","p")* 10   ;
*nuval("f239","p") =  nuval("f238","p")* 10   ;
*Upper limits for fodder  (added in 14.8.2007, while removing scaling in option file)
EF.up(j) =  20  ;
OEF.up(j) =  20 ;
HEF.up(j) =  20 ;
FF.up(l) =  20  ;
OFF.up(l) =  20 ;
HFF.up(l) =  20 ;
EF.fx("f17") = 0      ;
OEF.fx("f17") = 0     ;
HEF.fx("f17") = 0     ;
EF.fx("f18") = 0      ;
OEF.fx("f18") = 0     ;
HEF.fx("f18") = 0     ;
*-----------------------------------------------------------------------
*PM.fx("f0") = 0.0000001  ;
*PF.fx("f0") = 0.0000001 ;
*PS.fx("f0") = 5        ;
B.fx(j) = 0            ;
*NU.fx("f0") = 0.0000001 ;
*NM.fx("f0") = 2 ;
```

```
*RM AND UM WERE  15                ;
RM.up = 10                   ;
RM.lo = 0.001                ;
*NLOAD WORKS NOW BETTER WITHOUT UM.L(OR UPDATE THESE CLOSER TO THE SOLUTION VALUES, P
VERY SENSITIVE TO CHANGE!)
*UM.L = 0.2                  ;
*UM.up = 10                  ;
*UM.lo = 0.001               ;
*PoN.lo =  0.001             ;
*PoN.up =  20             ;
*UoM.lo = 1            ;
*UoM.up = 20           ;
* Test to increase .lo values to avoid boundary errors in the last fodder price sceneario
*UM.up = 15                  ;
*UM.lo = 0.1               ;
PoN.lo =  0.1              ;
PoN.up =  20             ;
UoM.lo = 1            ;
UoM.up = 20           ;
*STORAGE TECH-----------------------------------------------------------------
* was 2000 before 10.5.2011
MVS.up(sttte) =  2000           ;
MVS.lo(sttte) =  1           ;
PMT.up(sttte)  = 2000           ;
NMT.up(sttte)  = 2000           ;
NUT.up(sttte)  = 5000           ;

PMT.lo(sttte)  = 10           ;
NMT.lo(sttte)  = 40           ;
NUT.lo(sttte)  = 80           ;

RVM.up(sttte) = 5000               ;
*RVM.up("cc") = 10               ;
*RVM.up("nc") = 150             ;
*RVM.up("fc") = 150             ;

MSA.up(sttte) = MVS.up(sttte)/storageheight                ;
MSA.lo(sttte) = 0.1                ;
RVM.lo(sttte) = 0.0001             ;

*DISPERSAL TECH--------------------------------------------------------------

MVP.up(splte,dst)  =   2000      ;

*WITHOUT LOWISH UPPER LIMITS -> DOMAIN ERROR
PMS.up(splte,dst) =      2000      ;
NMS.up(splte,dst) =      4000      ;
NUS.up(splte,dst) =      6000      ;

PMS.lo(splte,dst) =      0.01      ;
NMS.lo(splte,dst) =      0.01       ;
NUS.lo(splte,dst) =      0.02       ;
```

```
MVP.lo(splte,dst)  =    1        ;

PCT.up = 10             ;
*prev 0.01
*NCT.lo = 0.001         ;
*prev 10
NCT.up = 10             ;
*NUCT.lo = 0.0001       ;
NUCT.up = 100           ;

MF.up = 5000            ;
*MP.up = 5000           ;
*MP.lo = 1              ;
*MF.lo = 1              ;
MY.lo = 5               ;
MY.up = 50              ;
* Added 1.4.2012 as giving infeasible solution for too large free variable
NC.up(j,dst) = 250 ;

* EQUATIONS -------------------------------------------------------

EQUATIONS
      MAXAFARM        objective function to maximise farmers profits
      TECH(macon)     technological and political constraints
      LAND(eqcon)     amount of land (currently reduction allowed)
      fallowConstraint(fcon)
*       MINFAL(micon)    minimum crop constraints
*       MAXFAL          maximum of cap allowed fallow
      PHOBAL(j,dst)    phosphorus balance
      PMANURE         definition of manure phosphorus content
      NPRATIO         ratio of P and N in manure
      UMRATIO         ratio of N in manure and urea
      PNFIELD(j,dst)   ratio of P and N in manure        (on field)
      UMFIELD(j,dst)   ratio of N in manure and urea      (on field)
      PCONTENT(stt)    P in m3  (of manure after dispersal losses)
      NCONTENT(stt)    N in m3  (of manure after dispersal losses)
      NUCONTENT(stt)   N in m3  (of urea after dispersal losses)
      SPREADP         P content of applied manure must be same as that of stored
      SPREADN         N content of applied manure must be same as that of stored
      SPREADNU        N content of applied (urea) must be same as that of stored
      NTECH(spt,dst)   N in manure m3 (for each dispersal technology and distance) (with storage loss)
      NUTECH(spt,dst)  N in urine per m3 (for each dispersal technology and distance)(with storage loss)
      PTECH(spt,dst)   P in manure m3 for each dispersal technology and distance)(with storage loss)
      NMANURE         definition of manure N content from feeding equations
      NUREA           definition of urea N content from feeding equations
      PMSTOR          P in storage for various tech given the storage losses
      NMSTOR          N in storage for various tech given the storage losses
      NUSTOR          N (from urea) in storage for various tech given the storage losses
      NMANBAL(dst)        total (over all tech and distance) nitrogen from dispersed manure equals plant
available manure nitrogen
```

NUREABAL(dst) total (over all tech and distance) nitrogen from dispersed urea equals plant available urea nitrogen

PMANBAL(dst) total (over all tech and distance) nitrogen from dispersed manure equals plant available manure nitrogen

MAVOL volume of total manure calculated from number of animals with constant coefficient
MANWGHT volume of manure (kg dm per day)
PUPPTAKE(j,dst) definition of phosphorus uptake by plants
PdynConst steady state P for private optimal
DRPload(j,dst) definition of dissolved phosphorus loss
PPload(j,dst) definition of particle phosphorus loss
DRPFALL(flw,dst) definition of dissolved phosphorus loss for fallow
NCROP(j,dst) nitrogen loss from crop land (simmelsgaard)
PPFALL(flw,dst) definition of particle phosphorus loss for fallow
NFALL(flw,dst) definition of N leaching for fallow
LOADP the restricted P load
LOADN the restricted N load
NDF Neutral detergent fiber requirements
OIV minimum requirement for protein for lactating cows
APHO minimum animal phosphorus needs
PVT minimum animal protein balance needs
ENERGY minimum dairy cow energy requirement
DMMAX maximum of cow daily fodder intake kg dry matter
GFMAXDA maximum of fodder that can be supplied from grass not made silage for dairy
GFMAXDR maximum of fodder that can be supplied from grass not made silage for dry
GFMAXHE maximum of fodder that can be supplied from grass not made silage for he
DMRY cow daily fodder intake kg of DM
DRY_NDF Neutral detergent fiber requirements for dry period
DRY_OIV minimum requirement for protein for dry period
DRY_APHO minimum animal phosphorus needs for dry period
DRY_PVT minimum animal protein balance needs for dry period
DRY_ENERGY Energy requirements for dairy cattle dry period
DRY_DMAX Maximum of cow daily fodder intake kg dry matter
H_DMAX maximum of heifer daily fodder intake kg dry matter
H_PROT Recommended protein intake for heifer per day (oiv)
H_ENER Recommended energy intake for heifer per day (ry)
H_NDF Neutral detergent fiber requirements for heifer per day (g)
H_APHO Minimum animal phosphorus needs for phosphorus
MILKYIELD Definition of milk yield per cow per day in kg or litre ???
FATYIELD Milk fat yield per cow per day in g per day
PROTYIELD Milk protein yield per cow per day in g per day
PROTCONST Minimum requirement for protein in milk for I-class milk
cropYield(j,dst) Crop yield for each distance year in kgs DM
CYEFBAL(j) fodder from crop cannot be bigger than grown
PREQ(j,dst) prevents phosphorus decifit in the static model
MASPR from manure storage to fields volume does not change
AU Animal units cannot fall below 0.4 else the farm loses animal husbandry status
AREA(stt) Surface area for manure storage assuming fixed height for each storage technology
RAINVOLUME(stt) Volume of rain water allowed to the storage
PINOUT p input must equal output
ESNFERT(j,dst) Environmental subsidy scheme fertilisation maximum
ESPFERT(j,dst) Environmental subsidy scheme fertilisation maximum

```
        ESMNFERT(j,dst)  Environmental subsidy scheme fertilisation maximum modified by the special
measures
        ESMPFERT(j,dst)  Environmental subsidy scheme fertilisation maximum modified by the special
measures
        Measured_Pman2007(j,dst)      Env subsidy constraint for manure P if no chemical P used and
voluntary P-analysis taken
        Norm_Pman2007(j,dst)  Env subsidy constraint for manure P if no chemical P used and voluntary P-
analysis NOT taken
        ESMAN              Environmental subsidy scheme manure maximum
        ESPLOT             Environmental buffer requirements
        NormNITDIR(j,dst)     Limitations by the EU ditrate directive if voluntary N norm of env subsidy
        MeasuredNITDIR(j,dst) Limitations by the EU ditrate directive if measured
        quality(j,dst)       Quality recommendations of crops related to fertilisation
        phosphorusFert(j,dst) Phosphorus fertilisation determined by N fertilisation
        manureTransport      total transported volume must equal total stored volume (including rainwater)
        distanceConst(dst)    fields exist in different distances from the farm
        PNRATIO             P over in N in manure
        manureBalanceP(dst)   soil surface layer P stock  given by manure application
;
*YIELD FUNCTIONS----------------------------------------------------------

cropYield(j,dst)..  CY(j,dst) =E=
{
* quadratic base yield
         yieldC2(j, "a")
* positive slope for nitrogen
         + yieldC2(j, "d") * (NF(j,dst) + NM(j,dst)+ NU(j,dst))
*negative term for nitrogen
         + yieldC2(j, "c") * POWER((NF(j,dst) + NM(j,dst) + NU(j,dst)),2)
* mitserlich base yield
         + yieldC2(j,"m")* [
                  1 - yieldC2(j,"k") *
* mitserlich nitrogen curve
                  exp(-(yieldC2(j,"b")*(NF(j,dst)+ NM(j,dst)+ NU(j,dst))))
                 ]
 }
*Transform to dry matter (DM) yield
  * (nuval(j, "ka")*0.01)
 ;

quality(j,dst).. (NM(j,dst) + NU(j,dst) + NF(j,dst)) * quality_table(j) =G= quality_constraints(j)    ;
*P requirement instead of yield effect which depends on the Y fertiliser ratio
phosphorusFert(j,dst)..  PF(j,dst) + PM(j,dst) =E= [NF(j,dst)+ NU(j,dst) + NM(j,dst)]  *  P_ratio(j)  ;

* MILK KG PER DAY PER COW
$ontext
*Old milk yield
MILKYIELD.. MY =E=
0.174 * [SUM{j,nuval(j,"ry")* EF(j)} + SUM{l, nuval(l,"ry") * FF(l)}] -
0.00037 * [SUM{j,nuval(j,"ry")* EF(j)} + SUM{l, nuval(l,"ry") * FF(l)}] +
0.009049 *  [SUM{j,nuval(j,"oiv")* EF(j)} + SUM{l, nuval(l,"oiv") * FF(l)}]
*- 202.16 *  [EF("f16") + EF("f17") + EF("f18")] /
```

```
*        [SUM{j,EF(j)} + SUM{l,FF(l)} - [EF("f16") + EF("f17") + EF("f18")]]
- 8.82   ;
$offtext
*Milk function (3 parameters) for milk quantity
MILKYIELD.. MY =E=
*First term (B)
  0.174 *
* Alla tulkintani "mediest"
   [10.95 * [SUM{j,nuval(j,"ry")* EF(j)} + SUM{l, nuval(l,"ry") * FF(l)}]
    + 13.6 - 29.5*(1-[EF("f16") + EF("f17") + EF("f18")]*POWER[SUM{j,EF(j)} + SUM{l,FF(l)}, -1] )
   ]
*Second term (C)
-   0.00037 *
* "mediest2" 2 = squared
SQR[10.95 * [SUM{j,nuval(j,"ry")* EF(j)} + SUM{l, nuval(l,"ry") * FF(l)}]
    + 13.6 - 29.5*(1-[EF("f16") + EF("f17") + EF("f18")]*POWER[SUM{j,EF(j)} + SUM{l,FF(l)}, -1] )
   ]
*Third term (D)
 + 0.009049 *  [SUM{j,nuval(j,"oiv")* EF(j)} + SUM{l, nuval(l,"oiv") * FF(l)}]
* Negative constant term (A)
- 8.82
;
FATYIELD.. MF =E=
*First term (B)
  9.58 *
* Alla tulkintani "mediest" väkirehu prosentin käyttö ei onnistu sillä jako lasku ei toimi, pitäisi alustaa?
   [10.95237
 * [SUM{j,nuval(j,"ry")* EF(j)} + SUM{l, nuval(l,"ry") * FF(l)}]
    + 13.6 - 29.5*(1-[EF("f16") + EF("f17") + EF("f18")]/[SUM{j,EF(j)} + SUM{l,FF(l)}] )
   ]
*Second term (C)
-   0.01405 *
* "mediest2" 2 = squared
SQR[10.95237
 * [SUM{j,nuval(j,"ry")* EF.L(j)} + SUM{l, nuval(l,"ry") * FF.L(l)}]
    + 13.6 - 29.5*(1-[EF.L("f16") + EF.L("f17") + EF.L("f18")]/[SUM{j,EF.L(j)} + SUM{l,FF.L(l)}] )
   ]
+
*OIV term (D)
0.1471 *  [SUM{j,nuval(j,"oiv")* EF.L(j)} + SUM{l, nuval(l,"oiv") * FF.L(l)}]
* Negative constant term (A)
- 449 ;
PROTYIELD.. MP =G=
*First term (B)
  4.70 *
* Alla tulkintani "mediest" väkirehu prosentin käyttö ei onnistu sillä jako lasku ei toimi, pitäisi alustaa?
   [10.95237
 * [SUM{j,nuval(j,"ry")* EF.L(j)} + SUM{l, nuval(l,"ry") * FF.L(l)}]
    + 13.6 - 29.5*(1-[EF.L("f16") + EF.L("f17") + EF.L("f18")]/[SUM{j,EF.L(j)} + SUM{l,FF.L(l)}] )
   ]
*Second term (C)
-  0.00897 *
```

```
* "mediest2" 2 = squared
SQR[10.95237
 * [SUM{j,nuval(j,"ry")* EF.L(j)} + SUM{l, nuval(l,"ry") * FF.L(l)}]
   + 13.6 - 29.5*(1-[EF.L("f16") + EF.L("f17") + EF.L("f18")]/[SUM{j,EF.L(j)} + SUM{l,FF.L(l)}] )
  ]
+
*OIV term (D)
0.3554 *  [SUM{j,nuval(j,"oiv")* EF.L(j)} + SUM{l, nuval(l,"oiv") * FF.L(l)}]
* Negative constant term (A)
 -338 ;

*RESOURCE AND TECHNOLOGY CONSTRAINTS ----------------------------
* Crop production constraints and balances
TECH(macon).. SUM{(j,dst), tecon(j, macon) * X(j,dst) } =L= valcon(macon)  ;
LAND(eqcon).. SUM{(j,dst), bacon(j, eqcon) * X(j,dst) }+ SUM{(flw,dst), Xfa(flw,dst)}  =E= hacon(eqcon)   ;
distanceConst(dst)..  SUM{(j),  X(j,dst)} + SUM{(flw), Xfa(flw,dst)} =E=  dairyFields(dst)                    ;
fallowConstraint(fcon)..  SUM{(flw,dst), facon(flw, fcon) * Xfa(flw,dst)}  =L= fallLimits(fcon)   ;
*MINFAL(micon) ..  SUM{flw,Xfa(flw,dst)}  =G= mini(micon)  ;
*MAXFAL..   SUM{flw,Xfa(flw,dst)} =L= farm_size * 0.5             ;
* The animal units per hectare cannot fall below 0.4 or 10
AU.. A/[SUM{(j,dst), X(j,dst)}+ SUM{(flw,dst),Xfa(flw,dst)}] =G= 0.4     ;
* The amount of self grown crops fed must be smaller than the amount grown
CYEFBAL(j).. A * [dim * EF(j) + (diy - dim) * OEF(j) + diy * calf_f * HEF(j)] =L= SUM{dst, CY(j,dst) * X(j,dst)}  ;
* The amount of fresh grass must be smaller than silage
*GFMAX.. A * [dim * EF("f18") + (diy - dim) * OEF("f18") + diy * (1 - calf_f) * HEF("f18")]
* =L= A * [dim * EF("f16") + (diy - dim) * OEF("f16") + diy * (1 - calf_f) * HEF("f16")] ;
GFMAXDA.. EF("f18")  =L= EF("f16") *0.5  ;
GFMAXDR.. OEF("f18") =L= OEF("f16")*0.5  ;
GFMAXHE.. HEF("f18") =L= HEF("f16")*0.5  ;

* CATTLE NUTRIENT REQUIREMENTS (including milk production)---------------------
PROTCONST.. MP/10/MY =G= 3.3    ;
*Maximum capacity to digest fodder for both lactating...
DMMAX.. SUM{j, EF(j)} + SUM{l, FF(l)} =L= 20    ;
* HUOM KA syönti on RY-tarve / 0.98
*DMRY..  SUM{j, EF(j)} + SUM{l, FF(l)} =E= [MY * 0.47 + (0.71 + 0.0083 * cow_w)]/
*+ 0.0269*[(FF("f23")+ FF("f24")+ FF("f25"))/(SUM{j, EF(j)} + SUM{l, FF(l}* 100]
*- 0.481
*       0.98 ;

*Nousiainen et al in Press
* ECM = energy corrected milk yield, SDMI = relative intake potential of silages compared with standard
high quality primary growth grass silage
* CDMI  = The effects of the amount and composition of concentrate feeds on total DM intake
* DMImax =E= 3.26 + 0.011 * cow_w + 0.33 * ECM  * 0.1 * (SDMI index + CDMI index - 100)

* and dry periods
DRY_DMAX.. SUM{j, OEF(j)} + SUM{l, OFF(l)} =L= 20    ;
* NDF minimum 25\% of dry matter ingestion from grass, hay, silage
* 1000 yksikkömuunnos kiloista grammoihin, joilloin ndf-pitoisuus ja rehumäärä samassa yksikössä
NDF..  EF("f16")* 573 + EF("f17")* 635 + EF("f18") * 567
=G=  0.25 * [SUM{j, EF(j)} + SUM{l, FF(l)}] * 1000  ;
```

```
DRY_NDF.. OEF("f16")* 573 + OEF("f17")* 635 + OEF("f18") * 567
=G=  0.25 * [SUM{j, OEF(j)} + SUM{l, OFF(l)}] * 1000  ;

*Cattle phosphorus requirements inputdata.xls:Pcont linear regression from fodder recommendations
APHO.. SUM{j, EF(j)* nuval(j,"pho")} + SUM{l, FF(l)* nuval(l, "pho")   }
=G= (1.8214 * MY + 14.357)  ;

DRY_APHO.. SUM{j, OEF(j)* nuval(j,"pho")} + SUM{l, OFF(l)* nuval(l, "pho")   }
=G=  14.357  ;

* PVT Rumen protein balance minumum of -15g/kg DM
* Or SUM in a day -400g if cow eats 20 kg DM  (or -15g on average )
PVT..  SUM{j, EF(j)* nuval(j,"pvt")} + SUM{l, FF(l)* nuval(l, "pvt") }
*/[SUM{j, EF(j)} + SUM{l, FF(l)}]
=G= -400      ;
DRY_PVT..  SUM{j, OEF(j)* nuval(j,"pvt")} + SUM{l, OFF(l)* nuval(l, "pvt") }
=G= -15 * [SUM{j, OEF(j)} + SUM{l, OFF(l)}]      ;

ENERGY.. SUM{j, EF(j)* nuval(j,"ry")} + SUM{l, FF(l)* nuval(l, "ry")}
=G=  [(0.71 + 0.0083 * cow_w)] + MY * 0.47     ;
*Upkeep 0.71 + 0.0078 * live weight + 0.44RY * EKM
*Only upkeep in the version below
* Energy requirements during the dry period
DRY_ENERGY.. SUM{j, OEF(j)* nuval(j,"ry")} + SUM{l, OFF(l)* nuval(l, "ry")}
=G=  [(0.71 + 0.0083 * drycow_w)]           ;

*Ylläpito Elopaino potenssiin 0.75+ 19.0 x KA-syönti (kg KA/pv)
*Maidontuotanto (g/pv)        1.45 x valkuaistuotos (g/pv)
*https://portal.mtt.fi/portal/page/portal/AGRONET/REHUTAULUKOT/t3
OIV.. SUM{j, EF(j)* nuval(j,"oiv")} + SUM{l, FF(l)* nuval(l, "oiv") }
=G= cow_w**0.75 + 19 * [SUM{j, EF(j)} + SUM{l, FF(l)}] ;

DRY_OIV.. SUM{j, OEF(j)* nuval(j,"oiv")} + SUM{l, OFF(l)* nuval(l, "oiv") }
=G= drycow_w**0.75 + 19 * [SUM{j, OEF(j)} + SUM{l, OFF(l)}]      ;

* Feeding requirements for heifers
H_DMAX.. SUM{j, HEF(j)} + SUM{l, HFF(l)} =L= heiferdm    ;
H_PROT.. SUM{j, HEF(j)* nuval(j,"oiv")} + SUM{l, HFF(l)* nuval(l, "oiv")}
=G= heifer_oiv       ;
H_ENER..   SUM{j, HEF(j)* nuval(j,"ry")} + SUM{l, HFF(l)* nuval(l, "ry")}
=G=  heifer_ry       ;
H_NDF.. HEF("f16")* 573 + HEF("f17")* 635 + HEF("f18") * 567
=G=  0.25 * [SUM{j, HEF(j)} + SUM{l, HFF(l)}] * 1000  ;
* From inputdata.xls:Pcont based on the linear regression from Finnish dairy mineral requirements
H_APHO.. SUM{j, HEF(j)* nuval(j,"pho")} + SUM{l, HFF(l)* nuval(l, "pho")   }
=G=  14.357  ;

* Limits set by the environmental subsidies-------------------------------------
*Fertilisation 1:1
ESNFERT(j,dst)..   NF(j,dst)+ NM(j,dst)+ NU(j,dst) =L= Nenvbasic2003(j)    ;
ESPFERT(j,dst)..   PF(j,dst)+ PM(j,dst)       =L= Penvbasic2003(j)    ;
```

```
Measured_Pman2007(j,dst)..   PM(j,dst)* 0.85        =L= Pmanure2007(j)      ;
*Norm_Pman2007(j)..      MSA(stt) * 0.5 * 0.85    =L= Pmanure2007(j)      ;
ESMNFERT(j,dst)..          NF(j,dst)+ NM(j,dst)+ NU(j,dst) =L= Nenvmod2003(j)      ;
ESMPFERT(j,dst)..          PF(j,dst)+ PM(j,dst)       =L= Penvmod2003(j)      ;
MeasuredNITDIR(j,dst)..  NM(j,dst) + NU(j,dst)   =L= nitratelimit              ;
*NormNITDIR(j)..     (MSA * 1.8)/X(j,dst) =L= nitratelimit              ;
$ontext
*Manure content from ministry guides 2003
*ESNFERT..   NM(j,dst)+ MSA * 1.9 =L= Nenvbasic2003(j)    ;
*ESPFERT..   PM(j,dst)+ MSA * 0.6 =L= Penvbasic2003(j)    ;
* Mietipä intuiivisesti, tässä annetaan levittää noin tuplasti, koska malli pitoisuus on puolet enemmän
* Laimentaminen vedellä kannattaa vain jos pääsee alle tämän pitoisuuden ja ottaa lanta P testin
*Näihin sovelletaan vielä 50\% - 75\% käyttökelpoisuus lukua, jos haluaa levittää paljon levittää syksyllä
*Manure content from ministry guides 2003
*ESNFERT(j)..   NM(j,dst)+ MSA * 1.9 * 0.5 =L= Nenvbasic2003(j)    ;
*ESPFERT(j)..   PM(j,dst)+ MSA * 0.6 * 0.75 =L= Penvbasic2003(j)    ;
*Manure content from ministry guides with the autumn 50\% calculability
$offtext
* MANURE------------------------------------------------------------------------------
* PMT = Total Phosporus in Manure for year
*Pekka Huhtanen 2004 slides MRINNE_REHUTAULUKKOUUDISTUS 2004, p8
*Assume that heifers and calves can be calculated by the same formula

PMANURE.. SUM{sttte, PMT(sttte)} =E=   (
 0.6277 * [
          dim *(SUM{j, nuval(j,"pho")* EF(j)} + SUM{l, nuval(l,"pho")* FF(l)})
         + (diy-dim) * (SUM{j, nuval(j,"pho")* OEF(j)} + SUM{l, nuval(l,"pho")* OFF(l)})
         + (calf_f)* diy *(SUM{j, nuval(j,"pho")* HEF(j)} + SUM{l, nuval(l,"pho")* HFF(l)})
        ]
                + 2.0961
                    )/ 1000 * A ;
*Uusi-Kämppä MET25 Lypsykarjataloudesta tulevan ympäristökuormituksen vähentäminen, s. 36,
Nousiainen ym: Methods of improving N utilisation through feeding strategies on dairy farms
* inputdata:Ncontent
NMANURE.. SUM{sttte, NMT(sttte)} =E= (
   0.0315 *  [
           dim *( SUM{j, nuval(j,"pvt")* EF(j)} + SUM{l, nuval(l,"pvt")* FF(l)})
           + (diy-dim) * (SUM{j, nuval(j,"pvt")* OEF(j)} + SUM{l, nuval(l,"pvt")* OFF(l)})
           + (calf_f)* diy *(SUM{j, nuval(j,"pvt")* HEF(j)} + SUM{l, nuval(l,"pvt")* HFF(l)})
            ]
      + 151.24   )/ 1000 * A ;
NUREA..   SUM{sttte, NUT(sttte)} =E= (
  0.1772 * [
          dim *  (SUM{j, nuval(j,"pvt")* EF(j)} + SUM{l, nuval(l,"pvt")* FF(l)})
           + (diy-dim) *( SUM{j, nuval(j,"pvt")* OEF(j)} + SUM{l, nuval(l,"pvt")* OFF(l)})
           + (calf_f)* diy * (SUM{j, nuval(j,"pvt")* HEF(j)} + SUM{l, nuval(l,"pvt")* HFF(l)})
     + 184.94    ]
       )/ 1000 * A ;
*REMOVED CONSTANT FOR TEST PURPOSES  ) / 1000 * A

*Storage loss for storage types times the amount from cattle equals the amount of nutrients to be spread
(without storage loss)
```

```
PMSTOR.. SUM{(splte,dst), PMS(splte,dst)} =E= SUM{sttte, storlossPM(sttte)* PMT(sttte)}  ;
NMSTOR.. SUM{(splte,dst), NMS(splte,dst)} =E= SUM{sttte, storlossNM(sttte)* NMT(sttte)}  ;
NUSTOR.. SUM{(splte,dst), NUS(splte,dst)} =E= SUM{sttte, storlossNU(sttte)* NUT(sttte)}  ;

*Nennich 2005
*http://www.sciencedirect.com/science/article/pii/S0022030205730587
*Best predictor was linear
* for lactating cows     (ME = manure excrement kg/day)
*ME = [DMI × 2.63 (± 0.10)] + 9.4 (± 2.8)
*for dry cows Wilkerson 1997 in Nennich
*[ME =(0.00711 × BW) + (32.4 × Dietary CP, g/g of DM) + (25.9 × Dietary NDF, g/g of DM) + 8.05
*heifer
*ME = [DMI × 4.158 (± 0.536)] - [BW × 0.0246 (± 0.0103)]

MANWGHT..  SUM{sttte, MVS(sttte)} =E=
[
*during lactation
 [ dim * ( 2.63 * [SUM{j, EF(j) } + SUM{l, FF(l)}] + 9.4  )
* dry
+ (diy-dim) *[(0.00711 * drycow_w) + 32.4 * [SUM{j, OEF(j)* nuval(j,"rv")/1000} + SUM{l, OFF(l)*
nuval(l,"rv")/1000 } ] / [SUM{j, OEF(j)}+ SUM{l, OFF(l)}]
              + 25.9 * [SUM{j, OEF(j)* nuval(j,"ndf")/1000} + SUM{l, OFF(l)* nuval(l,"ndf")/1000}] / [SUM{j,
OEF(j)}+ SUM{l, OFF(l)}] + 8.05
           ]
*heifer
  + diy * (calf_f) * [ 4.158 * [SUM{j, HEF(j)} + SUM{l, HFF(l)}]  - heifer_w * 0.0246 ]
*calf  (approx. half as heifer per day  for 14 days)
  +   14 * 6.5
 ] * A
*cleaning
  + cleaningwat * A * diy
]/1000  ;

*Wilkerson 1997 non linear model param for dry   (table 6)
*CP \% of DM            1.497
*CP^2                 -0.0372
*NDF \% OF DM           -0.124
*BW (in kg)            -0.0179
*Days in pregnant (DOP)  0.00586
*BW * NDF              0.000384
*DMI, kg               2.603
*DMI * CP              -0.0132

*Weight of manure (kg/dm) calculated from digestibility of dry matter in diet
*What is not digested ends up in manure, amount of urea assumed equal
*DIGESTIVE ORGANIC (D-value) \% OF PURCHASED FODDER UNCLEAR, ASSUMED 95\%
*Assume same density for manure, urea and water (1kg equals 1 litre), further: litre = 0.001 m3
* Including the cleaning water (from Tuhkanen)
*MANWGHT..  SUM{sttte, MVS(sttte)} =E=
*[
* [ dim *             [SUM{j, EF(j) * (1-nuval(j,"dv")/100)} + SUM{l, FF(l)*(1-nuval(l,"dv")/100)}]
*  + diy * (1 - calf_f)*[SUM{j, HEF(j)*(1-nuval(j,"dv")/100)} + SUM{l, HFF(l)*(1-nuval(l,"dv")/100)}]
```

```
*  + (diy-dim) *       [SUM{j, OEF(j)*(1-nuval(j,"dv")/100)} + SUM{l, OFF(l)*(1-nuval(l,"dv")/100)}]
* ] * A * ureamultiplier + cleaningwat * A * diy
*]/1000  ;
*Kattamattoman lietesäiliön sademäärän laskenta
*WIKI:Yksi milli sadetta on litra neliömetrille.
*keskisadanta 650mm eli 650 litraa per neliö
*säiliön pinta-alan oletusarvo 30 lypsylehmän tilalle 225
* sadanta 146 m3 vuodessa
*onko evaporaatio merkittävää sikäli mikäli lantasäiliön pinnalle muodostuu kuorikerros
*oletaan vaikka 15\% haihtuvaksi
*RADIUS..(dc_stor_req * A.L + h_stor_req * (1 - calf_f) * A.L) = 3.14159265 * POWER(RAD,2) *
storageheight         ;
*AREA.. MSA =E= (dc_stor_req * A.L + h_stor_req * (1 - calf_f) * A.L)/ storageheight        ;
*AREA.. SUM{sttte, MSA(sttte)} =E= SUM{sttte, MVS(sttte)}/ storageheight        ;

AREA(sttte).. MSA(sttte) =E= MVS(sttte)/ storageheight        ;
RAINVOLUME(sttte).. RVM(sttte) =E= MSA(sttte) * avrainfall(sttte)/1000 * (1 - evapor(sttte))
;
*Katteen kustannus riippuu pinta-alasta, jos käytetään MMM-RMO E2.1- 1(5) MAA- JA
METSÄTALOUSMINISTERIÖ  -kustannuksia 2005
*taulukossa regional_inputdata.xls:dairy_costs
*MSA * m_stor_cover_m2(stt)

* All manure (and the rainwater) must be spread also in volume terms
*MASPR..  SUM{sttte, MVS(sttte) + RVM(sttte) } =E= SUM{(splte,dst), MVP(splte,dst) }              ;
* All manure (and the rainwater) must be transported in volume terms
manureTransport..  SUM{sttte, MVS(sttte) + RVM(sttte) } =E=  SUM{(splte, dst), MVP(splte,dst)  }      ;
*Common nitrogen to phosphorus ratio in manure for each dispersal technology
NPRATIO.. RM =E= SUM{(splte,dst), NMS(splte,dst) * apNMloss(splte)}/SUM{(splte,dst), PMS(splte,dst)*
apPMloss(splte)}      ;
PNRATIO.. PoN =E= SUM{(splte,dst), PMS(splte,dst)* apPMloss(splte)}/ SUM{(splte,dst), NMS(splte,dst) *
apNMloss(splte)}      ;

*NPRATIO(dst).. RM(dst) =E= SUM{(j,splte), NMS(j,splte,dst) * apNMloss(splte)}/SUM{(j,splte),
PMS(j,splte,dst)* apPMloss(splte)}      ;
*urea to manure ratio of N
*UMRATIO.. UM =E= SUM{(j,splte,dst), NMS(j,splte,dst) * apNMloss(splte)}/SUM{(j,splte,dst),
NUS(j,splte,dst)* apNUloss(splte)}      ;
UMRATIO.. UoM =E= SUM{(splte,dst), NUS(splte,dst)* apNUloss(splte)}  / SUM{(splte,dst), NMS(splte,dst) *
apNMloss(splte)}     ;
*UMRATIO(dst).. UM(dst) =E= SUM{(j,splte), NMS(j,splte,dst) * apNMloss(splte)}/SUM{(j,splte),
NUS(j,splte,dst)* apNUloss(splte)}      ;

*NPRATIO.. RM =E= SUM{(j,splte,dst), NMS(j,splte,dst) }/SUM{(j,splte,dst), PMS(j,splte,dst)}      ;
*urea to manure ratio of N
*UMRATIO.. UM =E= SUM{(j,splte,dst), NMS(j,splte,dst) }/SUM{(j,splte,dst), NUS(j,splte,dst)}      ;

* Ratio of NP must be same on field as in manure after storage and during spreading
*NOT WORKING WITH DISTANCES
*PNFIELD(j,dst)..   NM(j,dst)/PM(j,dst) =E= RM                              ;
*UMFIELD(j,dst)..   NM(j,dst)/NU(j,dst) =E= UM                              ;
```

```
* For each storage technology there is a common total concentration of nutrients per unit of volume
PCONTENT(sttte)..  PCT =E= storlossPM(sttte)* PMT(sttte)/ (MVS(sttte) + RVM(sttte))   ;
NCONTENT(sttte)..  NCT =E= storlossNM(sttte)* NMT(sttte)/ (MVS(sttte) + RVM(sttte))   ;
NUCONTENT(sttte).. NUCT =E= storlossNU(sttte)* NUT(sttte)/(MVS(sttte)+ RVM(sttte))   ;

*not working -  with no lower bound for MVP -> domain error, with lower bound -> infeasible
*Content (PCT,NCT) needs  to be positive, if MPV is positive for every crop PMS,NMS should be positive, if
that is true also PM,NM * X needs to positive -> not working model
*PTECH(j,splte,dst)..  PCT  =E= PMS(j,splte,dst)  / MVP(j,splte,dst)                 ;
*NTECH(j,splte,dst)..  NCT  =E= NMS(j,splte,dst)  / MVP(j,splte,dst)                 ;
*NUTECH(j,splte,dst).. NUCT =E= NUS(j,splte,dst)  / MVP(j,splte,dst)                 ;

*not working -  with no lower bound for MVP -> domain error, with lower bound -> infeasible
*PTECH(j,splte,dst)..  PCT  =E= PMS(j,splte,dst)  / MVP(j,splte,dst)                 ;
*NTECH(j,splte,dst)..  NCT  =E= NMS(j,splte,dst)  / MVP(j,splte,dst)                 ;
*NUTECH(j,splte,dst).. NUCT =E= NUS(j,splte,dst)  / MVP(j,splte,dst)                 ;

*For each tech and distance class, the ratio of SUM of nutrients and SUM of manure volume must be
EQUAL to CONTENT IN STORED MANURE
*NOT WORKING NICELY, since if the sum for longer distance class approached zero (as is should since the
costs are higher),,
* -> PCT becomes huge and was only working for PMS with strictly binding upper limit (which forces the
nutrients to both distance classes, and is not very large for any crop)
PTECH(splte,dst)..  PCT  =E= PMS(splte,dst)  / MVP(splte,dst)                 ;
NTECH(splte,dst)..  NCT  =E= NMS(splte,dst)  / MVP(splte,dst)                 ;
NUTECH(splte,dst).. NUCT =E= NUS(splte,dst)  / MVP(splte,dst)                 ;

*FORCES TO TO HAVE POSITIVE X FOR ALL POSITIVE PMS/NMS/NUS, WHICH CANNOT BE NEGATIVE DUE
THE MIX CONSTRAINTS
*For each distance the total of applied manure nutrients on fields cannot be greater than the amount of
nutrients in manure after storage and dispersal losses
*PMANBAL(dst)..   SUM{j, PM(j,dst) * X(j,dst)} =E= SUM{(j,splte), PMS(j,splte,dst)* apPMloss(splte)}    ;
*NMANBAL(dst)..   SUM{j, NM(j,dst) * X(j,dst)} =E= SUM{(j,splte), NMS(j,splte,dst)* apNMloss(splte)}    ;
*NUREABAL(dst)..  SUM{j, NU(j,dst) * X(j,dst)} =E= SUM{(j,splte), NUS(j,splte,dst)* apNUloss(splte)}    ;

*SHOULD WORK AS EQUALITY CONSTRAINT BUT:
* requires possibility for  PMS/NMS/NUS = 0 WHICH CANNOT BE  TRUE FOR NUTRIENT RATIO
CONSTRAINTS (DIVISION BY 0 ERROR)
*For each distance the total of applied manure nutrients on fields cannot be greater than the amount of
nutrients in manure after storage and dispersal losses
PMANBAL(dst)..   SUM{j, PM(j,dst) * X(j,dst)} =E= SUM{(splte), PMS(splte,dst) * apPMloss(splte)}    ;
NMANBAL(dst)..   SUM{j, NM(j,dst) * X(j,dst)} =E= SUM{(splte), NMS(splte,dst) * apNMloss(splte)}    ;
NUREABAL(dst)..  SUM{j, NU(j,dst) * X(j,dst)} =E= SUM{(splte), NUS(splte,dst) * apNUloss(splte)}    ;

* For balanced P stock all spread manure must be greater than Iho's estimate of steady state
PdynConst..  [SUM{(j,dst), PM(j,dst)*X(j,dst)} + SUM{(j,dst), PF(j,dst) * X(j,dst)}]/ SUM{(j,dst),X(j,dst)} =G=
25.7  ;
* phosphorus contained in seeds is removed  - what about planted seeds, husks etc.
PUPPTAKE(j,dst).. PU(j,dst) =E= nuval(j,"pho")/1000 * CY(j,dst) * X(j,dst)                 ;
* now assumed that PM and PF have similar leaching properties
DRPload(j,dst).. DRP(j,dst) =E= loadcoef_DRP(j) * [2*(PS(dst) + 0.01*[PF(j,dst)+PM(j,dst)]) - 1.5]  * 0.0001  *
X(j,dst)         ;
```

```
PPload(j,dst).. PP(j,dst) =E= loadcoef_PP(j) * [250 * log(PS(dst) + 0.01 * [PF(j,dst)+PM(j,dst)])- 150] *
0.000001 * X(j,dst) ;
*Phosphorus loads for fallow
DRPfall(flw,dst).. FDRP(flw,dst) =E= loadcoeff_DRP(flw) * [2*PS(dst) - 1.5] * 0.0001 * Xfa(flw,dst) ;
PPfall(flw,dst).. FPP(flw,dst) =E= loadcoeff_PP(flw) * [250 * log(PS(dst)) - 150] * 0.000001 * Xfa(flw,dst) ;
*Nitrogen loads for crops
NCrop(j,dst).. NC(j,dst) =E= leach(j, "NIC") * EXP[0.71 * ((NF(j,dst) + NM(j,dst)+ NU(j,dst)) / nmax(j) - 1 )]
;
Nfall(flw,dst).. FAN(flw,dst) =E= loadcoeff_N(flw) * 0.367879441 ;
* removed the buffer strip [(1 - B(j)**0.2)* suflow + draflow ] *

*manureBalanceP(dst).. PS(dst) =E= SUM{splte, depthFactor(splte) * PMS(splte,dst)} / (SUM{j, X(j,dst) } +
SUM{flw, Xfa(flw,dst)} ) + Pstm ;
manureBalanceP(dst).. PS(dst) =E= Pstm ;
* phosphorus balance in land for each hectare the stock is different (POISTA - DRP(j) - PP(j,dst))
PHOBAL(j,dst).. PSt1(j,dst) =E=
        PStm1(j) + (0.00084 * PStm1(j) + 0.0032) *
        [PF(j,dst) + PM(j,dst) - PU(j,dst) - DRP(j,dst) - PP(j,dst)] - 0.0184*PStm1(j) ;
*The P balance is non-negative (optional)
PINOUT.. SUM{(j,dst),(PM(j,dst) + PF(j,dst))* X(j,dst)} =G= SUM{(j,dst),PU(j,dst) + DRP(j,dst) + PP(j,dst)}
;

* CONSTRAINT FOR ABATEMENT COST PURPOSES------------------------------------
LOADP.. SUM{(j,dst), DRP(j,dst)} + SUM{(flw,dst), FDRP(flw,dst)} + (SUM{(j,dst), PP(j,dst)}+ SUM{(flw,dst),
FPP(flw,dst)})/bio_coef =L= basetpload * (1 - red_l) ;
LOADN.. [ SUM{(j,dst), NC(j,dst) * X(j,dst) } + SUM{(flw,dst), FAN(flw,dst)*Xfa(flw,dst)} ] * retention =L=
basenload * (1 - red_l) ;

* Uudistus hiehot heifer = A/5
* Hiehojen ruokinta = heifer * syöntivakio * rehuyksikkö säilörehusta EF("f16")
* Hiehojen lanta voluumi = h_man_m3 = heifer * 10
* Hiehojen lanta ravinteet = pitäisi tulla ruokinnan kautta suoraan pmanure yms funktioihin
* Kaikkien syntyvien lemmujen juotto maidolla, eli yks jälkikasvu per vuosi per lehmä = MY - A*juontivakio
* Välitys vasikkojen myynti = A*välitysvasikan hinta

*OBJECTIVE FUNCTION-----------------------------------------------------------

* MAX CROP PROFIT + ANIMAL PROFIT - CROP SALES * ANIMAL_NEED - FIXED COSTS
MAXAFARM.. Z =E=
*MILK PROFITS
*Milk yield revenue and price subsidies
* Quality pricing did not work
* Divide by 1000 to get kilos from grams multiply by 100 to get \% -> divide by 10
* Divide by hundred to get euros from cent price
*milk_p + milk_s/100) * (dim * MY - calfmilk)
*[(milkq_p + milks_p+ milk_s) * (dim * MY - calfmilk)
*   +( fat_p * MF/10/MY
*   + prot_p * MP/10/MY)* dim ]/100
[ (prices('milk') + milk_s) * (dim * MY - calfmilk)

* Feeding costs for lactation period
  - [ dim *
```

```
       (
            SUM{j, EF(j) * prices(j)} +
            SUM{l, FF(l) * prices(l) }
       )
* Feeding costs during dry period
     + (diy - dim) *
       (
            SUM{j, OEF(j) * prices(j)} +
            SUM{l, OFF(l) * prices(l) }
        )
* Heifer fodder cost
      + (calf_f) * diy *
        (
            SUM{j, HEF(j) * prices(j)}  +
            SUM{l, HFF(l) * prices(l) }
        )
      ]
* labour
 - labour_c
* One fifth of the cows are sold each year
 + prices('meat') * cowmeat * calf_f
*Frome each cow one gets single calf per year of which some are kept for renewal purposes while other are
sold
 + (1-calf_f) * calf_p
] * A
* Capital, electricity, general expense from  inputdata.xls:FIX_A
- fix_a
+
SUM{(j,dst),
* CROP FARMING PROFITS
* crop subsidies and fixed costs and capital costs of land
 [  subsc(j)  -  fix_c(j,dst) - land_p    +
*crop price euro per kg DM, yield function for crops for wet weight (Saarela fodder units) in CY conversion
to DM based on NUVAL DM
          prices(j) *  CY(j,dst)
* Cost of fertiliser and its application (now f_app=0, but included in the fixed costs)
  - [
     prices('NPfertiliser') * NF(j,dst)
    ]
 ]* X(j,dst)
}
* Revenue/cost from fallow 2003
+ SUM{(flw,dst), Xfa(flw,dst) * (fall_subs(flw) - fall_cost(flw) - land_p) }
* Revenue/cost from fallow 2007
* + Xfa * (fall_subs + capsub + lfasub + youngsub - fall_cost - land_p)
*Storage cost of manure
*(based on the modeled volume without legistlative requirements of minumum storage capacity per cow)
- (m_stor_m3("nc") * SUM{sttte, MVS(sttte) + RVM(sttte)} + SUM{sttte, MSA(sttte) *
m_stor_cover_m2(sttte) * discountr} )
* Loading cost
- mload_t * lab_p/60 * SUM{sttte,MVS(sttte) + RVM(sttte)}
*Transport costs
```

```
- SUM{(splte, dst), [(fuel_p*fuel_c+loil_p*loil_c+lab_p) * (distance(dst)*2/avspeed) *  MVP(splte,dst) /
container_vol ] }
* Spreading costs
- SUM{(splte,dst), spread_t(splte,"f1") * MVP(splte,dst)  * (fuel_p*fuel_c+loil_p*loil_c + lab_p)/60 }
;
* MODELS ----------------------------------------------------------
MODEL PROFMAX / MAXAFARM, MILKYIELD, CROPYIELD, CYEFBAL, APHO, NDF, PVT, DMMAX, ENERGY
PROTYIELD,  TECH, fallowConstraint, distanceConst
*LAND
* Dry period & Heifer feeding
 DRY_ENERGY, DRY_DMAX, DRY_APHO, DRY_PVT, DRY_NDF, DRY_OIV, H_DMAX, H_PROT, H_ENER, H_NDF,
H_APHO,
* Manure constraints
* NPRATIO
UMRATIO, PNRATIO
 PMSTOR, NMSTOR, NUSTOR, PMANURE,  NMANURE, NUREA, NUREABAL, PMANBAL, NMANBAL,
 MANWGHT, RAINVOLUME, AREA, PUPPTAKE
* PNFIELD,
*UMFIELD
 PCONTENT , NCONTENT, NUCONTENT
 PTECH, NTECH, NUTECH, manureTransport
 DRPload, PPload, DRPfall, PPfall, Nfall, NCrop
* LOADP
* LOADN
* Environmental subsidy related
* ESNFERT
* ESPFERT
* MeasuredNITDIR
*quality
*PdynConst
phosphorusFert
manureBalanceP
*MASPR
/   ;

MODEL PROFMAXP / MAXAFARM, MILKYIELD, CROPYIELD, CYEFBAL, APHO, NDF, PVT, DMMAX, ENERGY
PROTYIELD,
* Dry period & Heifer feeding
 DRY_ENERGY, DRY_DMAX, DRY_APHO, DRY_PVT, DRY_NDF, DRY_OIV, H_DMAX, H_PROT, H_ENER, H_NDF,
H_APHO,
 TECH, fallowConstraint, distanceConst
*LAND
*MINFAL, MAXFAL,
* Manure constraints
* NPRATIO
UMRATIO, PNRATIO, PMSTOR, NMSTOR, NUSTOR, PMANURE,  NMANURE, NUREA, NUREABAL, PMANBAL,
NMANBAL,
 MANWGHT, RAINVOLUME, AREA, PUPPTAKE
* PNFIELD,UMFIELD
 PCONTENT , NCONTENT, NUCONTENT
 PTECH, NTECH, NUTECH, manureTransport
* SPREADP, SPREADN, SPREADNU, NUCONTENT
```

```
 DRPload, PPload, DRPfall, PPfall, Nfall, NCrop
 LOADP
* LOADN
* Environmental subsidy related
* ESNFERT
* ESPFERT
* MeasuredNITDIR
*quality
*PdynConst
*MASPR
phosphorusFert
manureBalanceP
/   ;

MODEL PROFMAXN / MAXAFARM, MILKYIELD, CROPYIELD, CYEFBAL,  APHO, NDF, PVT, DMMAX, ENERGY
PROTYIELD,
* Dry period & Heifer feeding
 DRY_ENERGY, DRY_DMAX, DRY_APHO, DRY_PVT, DRY_NDF, DRY_OIV, H_DMAX, H_PROT, H_ENER, H_NDF,
H_APHO,
 TECH, fallowConstraint, distanceConst
*LAND
* MINFAL, MAXFAL,
* Manure constraints
* NPRATIO
UMRATIO, PNRATIO, PMSTOR, NMSTOR, NUSTOR, PMANURE, NMANURE, NUREA, NUREABAL, PMANBAL,
NMANBAL,
 MANWGHT, RAINVOLUME, AREA, PUPPTAKE
*PNFIELD, UMFIELD,
 PCONTENT,  NCONTENT, NUCONTENT
 PTECH, NTECH, NUTECH, manureTransport
* SPREADP, SPREADN, NUCONTENT, SPREADNU
 DRPload, PPload, DRPfall, PPfall, Nfall, NCrop
 LOADN
* LOADP
* Environmental subsidy related
* ESNFERT
* ESPFERT
*MeasuredNITDIR
*quality
*PdynConst
*MASPR
phosphorusFert
manureBalanceP
/   ;
*removed from the option file
*lsscal = f ;

PROFMAX.OPTFILE = 1;
SOLVE PROFMAX using NLP maximising Z   ;

display UoM.L, RM.L, PoN.L ;
```

```
********RESULT PARAMETERS************* - CHECK DairyResultReport and dairy_results_OLS_2012
files
parameters
*LOAD PARAMETERS--------------------------------------------------------------
 redpro(count) reduction percent
 dairy_drpload_2003_noenv  total farm drp load
 dairy_ppload_2003_noenv   total farm pp load
 dairy_tpload_crops(j,dst) tp load per ha
 dairy_tpload_fallw(flw,dst)  tp load per ha
 dairy_tpload_2003_noenv total farm to load
 dairy_nload_2003_noenv total farm n load
 dairy_nload_crops_farm(j,dst) farm level n load by crops and distance
 dairy_nload_fallw_farm(flw,dst) farm level n load by fallow type and distance
 dairy_nload_crops(j,dst) n load per ha
 dairy_nload_fallw(flw,dst)
 dairyTPperHa2003noEnv     average tp load per ha
 dairyNperHa2003noEnv      average n load per ha
 FallTP_load
 FallTP_loadperha
 dairy_DRP_TP

*LAND PARAMETERS---------------------------------------------------------------
 landtotal
 dairy_fallow_2003_noenv(flw,dst)
 dairy_fallowTot_2003_noenv
 crops_Dairy_2003_noenv(j)
 fallow_Dairy_2003_noenv(flw)
 Dairy2003_noenv_landTotal
 Pstock(dst)
 dairy2003TotalYield(j,dst)
 totalYield(j)
 totalDMyield
 Pyield(j)
*CATTLE & FODDER USE-------------------------------------------------------------
 auperha
 dcowperha
 milkyield_2003_noenv
 MYPC
 totalmilk
 N_milkYieldIFCNcomp
 N_milkProteinYield
 ffodder(l)
 efodder(j)
 heiferfod_purch(l)
 heiferfod_farm(j)
 nonmilkfodder_purch(l)
 nonmilkfodder_farm(j)
 totfarmfod
 totpurchfod
 purchfod(l)
 totfodder
 farmfodcons(j)
```

forageShareLact2003noenv
OIV_demand
OIV_eaten
lactationExcessOIV
lactationEatenP
lactationExcessP
milkYieldIFCNcomp
milkProteinYield
intakePcontComparison eaten P over recommended
intakeNcontComparison eaten OIV over recommended

*MANURE PARAMETERS---
*storage
manure_volume
heiferManVolPerDay
ManureVolumePerAU
storedManureNM
storedManureNU
storedManureP(sttte)
storedManureN(sttte)
totalStoredManureP
totalStoredManureN
storagem3_req
storagetech(sttte)
totalMStorSurface
manPMTsum
manureCoverCost
storManNcontent
storManPcontent
* rainWaterPercent
*dispersed
transportedManureVolume(dst)
transportedManureP(dst)
transportedManureN(dst)
lostManureN   total N received by plants over total N from animals
manurevolMVP
dispersedM3(splte)
spreadingDistance(dst)
manPsum
dispersedPMSsum
dispersedNMSsum
dispersedNUSsum
totalManureNsum
manNsum
ureNsum
* sumManures = N_manNsum(count) + N_ureNsum(count) ;
storcont
dispManNcontent
dispManPcontent
manureP_plants(j)
manureN_plants(j)
ureamanure_ratio

```
 NP_ratio
 dispersedNU(splte)
 dispersedNM(splte)
 dispersedPM(splte)
 notDispersedNM
 notDispersedNU
 notDispersedPM
 DispersalTechBenefit
 DispTechBenefitBase(splte)
 manureVolPerHa(dst)      manure volume per ha
 manurePcontComparison    manure P cont over farm data 2000-2003
 manureNcontComparison    manure N cont over farm data 2000-2003

*REVENUE PARAMETERS-----------------------------------------------------------
 cropsales(j)
 croprevenues(j)
 profitpercow
 TotalRevenues
 MilkRevenueShare
 SubsidyRevenueShare

*COST PARAMETERS -----------------------------------------------------------
 costownfodder(j)
 purchfodcost(l)
 totfarmfodcost
 totpurchfodcost
 totfoddercost
 fertcost
 fieldcost
 landcost
 animcapital
 labourcost
 yearlywage
 monthlywage
 fallowcost
*Storage
 totalStorageCost
 stor_c_m3_req
 stor_c_m3
 stor_c_A
*Spreading
 spreadCost
 spread_var
 spread_var2
 spread_m3
 spread_var_m3
 dispM3tech(splte)
 dispCapCost
 dispCapCostM3
 dispersalCostM3
 spreading_cc
 spread_per_P
```

```
*Transport
 transportCost
 transportTrips
 tripCost(dst)
 trans_m3
 trans_m3_per_km
 Profit_distance
 variableDispersalCost
*For CP comparison with the endogenous costs
 spread_cP
 all_cost
 costpercow
 costPerMilkLitre

*Gate balance parameters-------------------------------------------------------
 fertNdistances(dst)
 fertPdistances(dst)
 fertNsum
 fertPsum
 Dairy2003_noenv_Nfert(j,dst)
 Dairy2003_noenv_Pfert(j,dst)
 allfieldP
 allfieldN
 allfieldNperha
 inputN_dairy_2003_noenv(j,dst)
 inputP_dairy_2003_noenv(j,dst)
 seedP_in
 fodderP_in
 fertP_in
 yieldP_out
 milkP_out
 calfP_out
 cowP_out
 manureP_out
 gateP_in
 gateP_out
 gatePbalance
*Field balance parameters-------------------------------------------------------
 Dairy2003_noenv_Nmanu(j,dst)
 Dairy2003_noenv_Pmanu(j,dst)
 fertPsum
 yieldP_tot
 yieldN_tot
 yieldP_tot2
 fieldPbalance
 fieldPbalancePerHa
 fieldNbalance
 fieldNbalancePerHa
 fieldPbalancePerCropHa(j)
 fieldNbalancePerCropHa(j)
 dispCapCostM3Base(splte)
 DispTechBenefitBase(splte)
```

```
 DispersalTechNetValue
 DispTechValueComparison(splte)
 dispCapitalShare
 averageInputN(j)
 averageInputP(j)
 ;
* Checkin the baseline optimality analytically for crops
parameters initCropRevenue(j), initCropProfitTest(j), initCropProfitTest2(j), initCropProfit(j),
initCropProfitSubs(j), shouldBePositive(j), ninitTest(j),  ninitTest2(j) To check that Dm conversion did not
affect optimal fertilization  ;

 optimalInitYields(j) =
{
          yieldC2(j, "a") + yieldC2(j, "d") * ninit(j)  + yieldC2(j, "c") * POWER(ninit(j),2)
* mitserlich base yield
          + yieldC2(j,"m") * [  1 - yieldC2(j,"k") *  exp(-(yieldC2(j,"b")*ninit(j)))  ]
 }
*Transform to dry matter (DM) yield
  * (nuval(j, "ka")*0.01)
 ;
initCropRevenue(j) = optimalInitYields(j) * prices(j) ;
initCropProfit(j) = initCropRevenue(j) - fix_c(j,'dstBM') - prices('NPfertiliser') * ninit(j)  ;
initCropProfitSubs(j) = initCropProfit(j) + subsc(j)    ;
ninitTest(j) = ninit(j) - 1 ;
initCropProfitTest(j) =  {
* quadratic base yield
          yieldC2(j, "a")
* positive slope for nitrogen
          + yieldC2(j, "d") * ninitTest(j)
*negative term for nitrogen
          + yieldC2(j, "c") * POWER(ninitTest(j),2)
* mitserlich base yield
          + yieldC2(j,"m")* [
                    1 - yieldC2(j,"k") *
* mitserlich nitrogen curve
                    exp(-(yieldC2(j,"b") * ninitTest(j)))
                   ]
 }
*Transform to dry matter (DM) yield
  * (nuval(j, "ka")*0.01)    * prices(j) - fix_c(j,'dstBM') - prices('NPfertiliser') *  ninitTest(j)   ;

shouldBePositive(j) = (initCropProfit(j) + subsc(j)) -  (initCropProfitTest(j) + subsc(j) )  ;

$ontext
c_price(j) =   c_price(j) * (nuval(j, "ka")*0.01)     ;

n1(j) $(qyieldC2(j,"a") AND qyieldC2(j, "d") AND qyieldC2(j, "c") NE 0)
            =  [prices('NPfertiliser')/[c_price(j) ]  - qyieldC2(j,"d")] / [2 * qyieldC2(j,"c")  ]    ;
n2(j) $(myieldC2(j,"m") AND myieldC2(j, "k") AND myieldC2(j, "b") NE 0)
            =  log[(c_price(j)  * myieldC2(j, "m")* myieldC2(j, "k") * myieldC2(j, "b"))
                 / prices('NPfertiliser')] / myieldC2(j, "b")   ;
ninitTest2(j)  =  n1(j) + n2(j) ;
```

```
c_price(j) =   c_price(j) / (nuval(j, "ka")*0.01)     ;
display ninitTest2 ;
$offtext
display ninit, ninitTest,  optimalInitYields, initCropRevenue, initCropProfit, initCropProfitSubs,
initCropProfitTest, shouldBePositive ;
display dairyFields, fix_c, quality_table, quality_constraints, yieldC2, np_priceBarley, prices  ;
parameter heifers ;
display DairyBelowMedianDistance, DairyAboveMedianDistance    ;

* SOLVE PROFMAX using NLP maximising Z ;

*heifers = A.l * calf_f ;
*display heifers ;
```

Nutrient abatement and biodiversity model for crop farm

```
$Title Nutrient abatement and biodiversity model for crop farm

$onsymlist offsymxref
 OPTION LIMROW = 0
 OPTION LIMCOL = 0  ;


$eolcom //
option iterlim=999999999; // avoid limit on it
option reslim=600; // timelimit for sol
*option optcr=0.0; // gap tolerance
*option solprint=ON; // include solution
*option limrow=100; // limit number of r
*option limcol=100; // limit number of c

*option decimals = 2;
*Switch default solution printing off.
*option solprint = off;
//-------------------------------------

$onecho > cmd.txt
I="C:\GIS\SAMA_databaseGIS2003.mdb"
X=Leps_data_param.gdx

Q1=select crop from set_j_Leps
s1=crop
Q2=select flw from setflw_Leps
s2=fallw
Q3=select slp from 6slopes
s3=slope
Q4=select sltclass from soil4set
s4=soilt
Q6=select Pstat from setPst_Leps
s6=Pstat
Q7=select crop_id from crop
s7=croptype
Q9=select aspct from setAspct
s9=aspct
Q10=select dst from setDstnc
s10=dstnc
Q8=select con from setcon
s8=con
*Q9=select fcon from setfcon
*s9=fcon
Q12=select Pfert from setPfert
s12=pfert
Q11=select Nfert from setNfert
s11=nfert
*Q13=select fn from Leps_cropfarm_area
```

```
*s13=fn
Q20=select crop, price2009 from set_j_Leps
p20=c_price
Q21=select crop, FixedCost2008 from set_j_Leps
p21=fix_c
Q19=select fallw, FixedCost2008 from setflw_Leps
p19=fall_cost
*Q22=select fallw, fcon, facon from FACON
*p22=facon
*Q24=select crop, Leps_mingrass from setj
*p24=mingrass
Q26=select soilt, Pstat, slope, aspct, dstnc, Area from LepsArableDistr
p26=land_distr
Q28=select crop, soilt, Pstat, slope, aspct, dstnc, Area from LepsCropDistr
p28=crop_distr
Q29=select fallw, soilt, Pstat, slope, aspct, dstnc, Area from LepsFallowDistr
p29=fallow_distr
Q27=select Pstat, Pst_no from setPst_Leps
p27=Pst_no
*Q30=select crop, national_subsidy from crop_farm_subsidies_2009_A
*p30=ntlsubs
*Q31=select crop, cap_subsidy from crop_farm_subsidies_2009_A
*p31=capsubs
*Q32=select crop, env from crop_farm_subsidies_2009_A
*p32=envsubs
*Q33=select crop, lfa from crop_farm_subsidies_2009_A
*p33=lfasubs
*Q34=select fallw, capfallow from crop_farm_fallow_subsidies_2009_A
*p34=capfallsubs
*Q35=select fallw, envfallow from crop_farm_fallow_subsidies_2009_A
*p35=envfallsubs
*Q36=select fallw, lfafallow from crop_farm_fallow_subsidies_2009_A
*p36=lfafallsubs
*Q40=select fn, m2 from Leps_cropfarm_area
*p40=Leps_cropfarm_areas
*Q41=select crop, con, tecon from tecon
*p41=tecon
*Q42=select SUM_F_AREA from LepsBuffer15m
*p42=LepsBufferArea15m
*Q43=select SUM_F_AREA from LepsBuffer3m
*p43=LepsBufferArea3m
*Q44=select SUM_F_AREA from LepsBuffer1m
*p44=LepsBufferArea1m
*Q48=select SUM_F_AREA from LepsBuffer160m
*p48=LepsBufferArea160m
*Q49=select SUM_F_AREA from LepsBuffer500m
*p49=LepsBufferArea500m
*Q45=select ReductionCoefficient from wetlandNutrientReductionP
*p45=wtldCoefficientP
*Q46=select ReductionCoefficient from wetlandNutrientReductionN
*p46=wtldCoefficientN
Q47=select F_AREA from LepsCatchmentTotalArea
```

```
p47=CatchmentTotalArea
Q48=select fallw, bdfallow from setflw_Leps
p48=bdfallow
$offecho
$call =mdb2gms \atcmd.txt

* SETS ------------------------------------------------------------
* subsets for tweaking the optimization problem without touching the data dimensions
SETS
        fn       farms
        crop     crops
        fallw    fallow crops
        slope    slopes
        soilt    soil types
        aspct    aspect
        dstnc    distance
        Pstat    P status

        flw(fallw) fallow types  gs1 gs2 gn1 gn2   g=green  s=subsidy eligible n=no subsidies ff=forest 1= biodiv
2=normal 3= game cereal 4=buffer strip
        bdfall(fallw)  biodiversity fallow types
        slp(slope) slope classes  1=0-0.5 2=0.5-1 3=1-2 4=2-3 5=3-6 6=>6
        slt(soilt) soil types  1=HsS 2=HtS 3=HHt 4=KHt
        asp(aspct) aspect classes  asp1 asp2 asp3
        dst(dstnc) distance classes dst25 dst50 dst750
        j(crop)    farmed crops   f13 spring barley
        Pst(Pstat) P status
        count      iterative increases in P load restriction / 1-1 * 1-32 /
        feas(count) feasible solution subset of count has not been defined here   / 1-1 * 1-32 /
        pfert P fertilisation levels in ICECREAM data
        nfert N fertilisation levels in ICECREAM data

        y yield function coefficients
/
        a     Quadratic parameter a intercept
        d     Quadratic parameter b positive slope
        c     Quadratic parameter c for diminishing marginal
        m     Mitscherlich parameter m
        k     Mitscherlich parameter k
        b     Mitscherlich parameter b
        pyc1  yield coefficient for P fertilisers
        pyc2  yield coefficient for P fertilisers
        pyc3  negative yield coefficient of P stock
        pyc4  positive yield coefficient of P stock
        pcon  constant increase in P yield
        pcoe  crop type yield coefficient for P
        sry   saarelas fodder unit
        tillfac tillage yield factor
/
        lc load coefficients for P and N
/       r runoff
        e erosion
```

```
        o omega leaching based on different technology
        v delta leaching based on different technology
        nic nitrogen load coefficient
        psini initial phosphorus level
        PStm1 dummy for last period P level
/
        con max an min crop resource constraints

        fcon  various fallow maximum subsidy constraints

       eqcon equalised constraints
/
       ha         arable land available for model farm
/
       primary price set for habitats and nutrient loads

/
       habitat
       nutrient
/
;
*       soil
alias (soil, slt) ;
parameters
*LAND PARAMETERS---------------------------------------------------------------
        Leps_cropfarm_areas(fn)                          Field areas of farms within Lepsjoki watershed with no
production animals
        crop_distr(crop,soilt,Pstat,slope,aspct,dstnc)         Crop types on different soil slope and P status land
(different from calibration file classes)
        fallow_distr(fallw,soilt,Pstat,slope,aspct,dstnc)      Crop types on different soil slope and P status land
(different from calibration file classes)
        land_distr(soilt,Pstat,slope,aspct,dstnc)            Land distribution in m2 to ha different soil and slope
classes on farm scale
        land_distr_share(soilt,Pstat,slope,aspct,dstnc)        Land distribution percent share to different soil and
slope classes on farm scale
*       land_distr_farm(slope,soilt,Pstat,aspct,dstnc)        land distribution in ha to different soil and slope
classes on farm scale
        initCropShare(crop,soilt,Pstat,slope,aspct,dstnc)      Share of crops from total CROP land 2009
*       N_cropdistr_hom(count,j,slp,slt,Pst)
        initFallowShare(fallw,soilt,Pstat,slope,aspct,dstnc)    Share of fallow from total FALLOW land 2009
        TotalInitCropArea
        TotalInitFallArea
        veps_ha              area of modeled land in VEPS
        model_ha              arable land area covered by the model
        fallow_ha             total fallow area
        fallow_ha_rev          total fallow area of the watershed directly from Access with smaller sets
        crop_ha              total crop area of the watershed
        crop_ha_rev           total crop area of the watershed directly from Access with smaller sets
        model_ha_rev           total model area of the watershed directly from Access with smaller sets
        CatchmentTotalArea      Total area of the catchment
        catchmentFieldShare     Share of arable land of the catchment total land
        grass_land_share       Share of grass land of total modeled area  (all non fallow assumed barley)
```

```
*LOAD PARAMETERS---------------------------------------------------------------
       runoff(crop,slope,soilt)                Runoff in mm per ha
       rnoff(crop,slope,soilt)                 Runoff in mm per ha for the different soil type set
       runoff_m(crop)                          Runoff of mean slope and soil (HtS)
       sloss(crop,slope,soilt)                 Soil loss in kg ha-1 from crop area
       Pst_no(Pstat)                           Soil P status mg per l
       runoff_flw(fallw,slope,soilt)           Runoff in mm per ha    from fallow
       sloss_flw(fallw,slope,soilt)            Soil loss in kg ha-1   from fallow
       TPloadIC_flw(fallw,slope,soilt,Pstat,pfert)     Total P load ha-1 from fallow according icecream
       NloadIC(crop,slope,soilt,nfert)         N load ha-1 from crop area according icecream
       NloadIC_flw(fallw,slope,soilt)          N load ha-1 from fallow according icecream
       leach(crop,lc)                          Load coefficients  from previous model versions
       loadCoeffN(crop,slope,soilt)            Load coefficients  from previous model versions
       loadCoeffN_flw(fallw,slope,soilt)       Load coefficients for fallow from previous model versions
*reduced soil dimesion
       loadCoeff_N(crop,slope,soilt)
       loadCoeff_N_flw(fallw,slope,soilt)
       Ncoeff_frt_b(crop,slope,soilt)
       Ncoeff_frt_c(crop,slope,soilt)
*       rnoff(j,slp,soil) Runoff in mm per ha
       erosion(crop,slope,soilt)               Soil loss in kg ha-1
       rnoff_flw(fallw,slope,soilt)
       erosion_flw(fallw,slope,soilt)
       sloss_m(crop)
       runoff_flw_m(fallw)
       sloss_flw_m(fallw)
       pmax(Pstat,crop,soilt)                  Maximum base fertilisation level as intial P level
       nmax(crop,soilt)                        Maximum base fertilisation level as intial N level
       facon(fallw, fcon)                      Constraint matrix left-hand side  for fallow
       tecon(crop, con)                        Constraint matrix left-hand side

*ECONOMIC PARAMETERS---------------------------------------------------------------
       minfall(fallw)
       maxfall(fallw)
       c_price(crop)          product producer prices
       prices(primary)
       fix_c(crop)            fixed costs euro per ha of field
       fall_cost(fallw)       Fixed costs of fallow euro per ha
       ntlsubs(crop)          national ha subsidies for crops
       capsubs(crop)          EU ha subsidies for crops
       envsubs(crop)          env subsidy for crop
       lfasubs(crop)          lfa subsidy for crop
       mingrass(crop)         demand of grass modeled as minimum land constraint
       capfallsubs(fallw)
       envfallsubs(fallw)
       lfafallsubs(fallw)

*BIODISVERSITY PARAMETERS------------------------------------------------------
       fallBD(fallw,dstnc,aspct,slope)      Biodiversity index for fallow (from Kari)
       cropBD(crop,dstnc,aspct,slope)       Biodiversity index for crops (from Kari)
```

```
      TotalMaxBD                  Sum over the BD index with existing land distriubtion but best possible
land use
      TotalBaselineBD              Sum over the BD index with existing land distriubtion and current land
use
      bdfallow(fallw)              Dummy for biodiversity fallow
;
scalars
      farm_size           Average crop farm size for Lepsjoki
      Leps_farm_no         Number of farms with
      TotalMaxProfit       Maximum private profit
* Abatement iterations
      reduction           dummy reduction                    /-10/
      red_l              abatement rate                     /0/
*       basetpload     initial for base p load
      tpinit             check tp in the beginning
      bio_coef            conversion factor from PP to algae available
      redfield            redfield ratio (mass to mass)                  / 7.2  /
      retention           no retention is assumed (included in the ICECREAM parameters)  /1/
      PsAvg              average initial soil P stock mg per l
      PStm               initial soil P stock mg per l  if variable
      baseN_load           N load (given land distribution 2009 and MYTVAS average fertilization)
*       baseP_load           P load which is scaled down to VEPS levels
*       veps_N_2002          28 total N load (kg) for 2002
*       veps_TP_2002         28 total P load (kg) for 2002
      TotalNrevised
      TotalN_MYTVAS_Hi
      TotalN_MYTVAS_Lo
      MYTVASmodTP
      drp_load            dissolved reactive phosphorus load
      pp_load             particle phosphorus load
      DRP_TP_share         share of drp of total p
      baseTPload           TP load which is NOT scaled down to VEPS levels and the basis for first non-
restricted model solution
      baseNload            N load which is NOT scaled down to VEPS levels  and the basis for first non-
restricted model solution
*Model runs sensitive to these, should be implemented in the homogeneous version too
      baseTPmulti          Multiplier for base TP load to ensure that the intial contstraint is not a binding one
/ 2 /
      baseNmulti           Multiplier for base N load to ensure that the intial contstraint is not a binding one
/ 3 /
      modTP              modeled Total Phosphorus
      MYTVASmodBioP
      TotalMinLoad         Minimum load given land distribution and all fallow when N+P*redfield
      TotalBaseLoadRedfield   Base load given land distribution and all fallow when N+P*redfield

      LepsBufferArea500m
      LepsBufferArea160m
      LepsBufferArea15m
      LepsBufferArea3m
      LepsBufferArea1m
      LepsBufferArea500mShare
      LepsBufferArea160mShare
```

```
      LepsBufferArea15mShare share of field within 15 meters of total modeled watershed farm area
      LepsBufferArea3mShare  share of field within 3 meters of total modeled watershed farm area
      LepsBufferArea1mShare  share of field within 1 meters of total modeled watershed farm area
      wetlandInvestCostkgP   investment costs per kg P of wetland regional_inputdata.xls / 411  /
      wetlandInvestCostkgha  investment costs per ha of wetland regional_inputdata.xls (regional farms)
Majoinen 2005 Tuusula  / 30611  /
      wtldCoefficientP      P reduction coefficient of wetland
      wtldCoefficientN      N reduction coefficient of wetland
;
*GET THE DATA FROM THE GDX FILE CREATED FROM ACCESS DATABASE
$gdxin C:\gamsdir\biodiversity\Leps_data_param.gdx
$load crop fallw slope soilt Pstat aspct dstnc Pst_no
$load land_distr crop_distr fallow_distr CatchmentTotalArea bdfallow
$load c_price fix_c fall_cost
*$load pfert nfert
*$load capfallsubs envfallsubs lfafallsubs ntlsubs capsubs envsubs lfasubs
*$load LepsBufferArea500m LepsBufferArea160m LepsBufferArea15m LepsBufferArea3m
LepsBufferArea1m wtldCoefficientP wtldCoefficientN
*$load fn Leps_cropfarm_areas
$load con
* $load  facon  fcon tecon   mingrass
*Assigning subsets
slp(slope)      = yes  ;
j(crop)         = yes  ;
flw(fallw)      = yes  ;
slt(soilt)      = yes  ;
dst(dstnc)      = yes  ;
Pst(Pstat)      = yes  ;
asp(aspct)      = yes  ;
bdfall(fallw)$(bdfallow(fallw)= 1) = yes     ;

*Nutrient load parameters
$gdxin C:\gamsdir\load_calibration_Lepsjoki\Leps_load_parameters.gdx
*PARAMETER DATA FOR THE 20 SOIL TYPE  SET
*$load sloss rnoff runoff_flw sloss_flw  loadCoeffN_flw baseN_load modTP
*PARAMETER DATA FOR THE 4 SOIL TYPE  SET
$load rnoff erosion rnoff_flw erosion_flw loadCoeff_N loadCoeff_N_flw
$load Ncoeff_frt_b Ncoeff_frt_c baseN_load
$load model_ha fallow_ha crop_ha
*$load veps_ha  grass_land_share veps_N_2002 veps_TP_2002
$load  PsAvg pmax nmax
$load modTP bio_coef MYTVASmodBioP
$load TotalNrevised TotalN_MYTVAS_Hi TotalN_MYTVAS_Lo MYTVASmodTP
;
*Biodiversity parameters
$gdxin C:\gamsdir\biodiversity\biodiversityIndex.gdx
$load fallBD cropBD
display  fallBD, cropBD  ;

*MODEL DIMENSIONS
scalar model_size ;
model_size = card(j) * card(slp) * card(slt) *  card(Pst) * card(asp) * card(dst) ;
```

```
display model_size ;

*(Ncoeff_frtREV_b(j,slt,slp) * exp[nmax(j,slt) * Ncoeff_frtREV_c(j,slt,slp)])

*TRANSFER PARAMETERS TO THE SMALLER SOIL SET-------------------------------------
loadCoeffN(j,slp,'slt1')      = loadCoeff_N(j,slp,'slt1')          ;
loadCoeffN(j,slp,'slt2')      = loadCoeff_N(j,slp,'slt2')          ;
loadCoeffN(j,slp,'slt3')      = loadCoeff_N(j,slp,'slt3')          ;
loadCoeffN(j,slp,'slt4')      = loadCoeff_N(j,slp,'slt4')          ;

loadCoeffN_flw(flw,slp,'slt1')      = loadCoeff_N_flw(flw,slp,'slt1')          ;
loadCoeffN_flw(flw,slp,'slt2')      = loadCoeff_N_flw(flw,slp,'slt2')          ;
loadCoeffN_flw(flw,slp,'slt3')      = loadCoeff_N_flw(flw,slp,'slt3')          ;
loadCoeffN_flw(flw,slp,'slt4')      = loadCoeff_N_flw(flw,slp,'slt4')          ;

runoff(j,slp,'slt1')      = rnoff(j,slp,'slt1')          ;
runoff(j,slp,'slt2')      = rnoff(j,slp,'slt2')          ;
runoff(j,slp,'slt3')      = rnoff(j,slp,'slt3')          ;
runoff(j,slp,'slt4')      = rnoff(j,slp,'slt4')          ;

sloss(j,slp,'slt1')      = erosion(j,slp,'slt1')          ;
sloss(j,slp,'slt2')      = erosion(j,slp,'slt2')          ;
sloss(j,slp,'slt3')      = erosion(j,slp,'slt3')          ;
sloss(j,slp,'slt4')      = erosion(j,slp,'slt4')          ;

runoff_flw(flw,slp,'slt1')      = rnoff_flw(flw,slp,'slt1')          ;
runoff_flw(flw,slp,'slt2')      = rnoff_flw(flw,slp,'slt2')          ;
runoff_flw(flw,slp,'slt3')      = rnoff_flw(flw,slp,'slt3')          ;
runoff_flw(flw,slp,'slt4')      = rnoff_flw(flw,slp,'slt4')          ;

sloss_flw(flw,slp,'slt1')      = erosion_flw(flw,slp,'slt1')          ;
sloss_flw(flw,slp,'slt2')      = erosion_flw(flw,slp,'slt2')          ;
sloss_flw(flw,slp,'slt3')      = erosion_flw(flw,slp,'slt3')          ;
sloss_flw(flw,slp,'slt4')      = erosion_flw(flw,slp,'slt4')          ;

runoff_m(j) = runoff(j,"slp3","slt2")  ;
sloss_m(j) =  sloss(j,"slp3","slt2")   ;
runoff_flw_m(flw) =  runoff_flw(flw,"slp3","slt2")   ;
sloss_flw_m(flw)  =  sloss_flw(flw,"slp3","slt2")    ;

DISPLAY loadCoeffN, runoff, sloss, runoff_flw, sloss_flw       ;

*LAND PARAMETERS----------------------------------------------------------------
*Leps_farm_no = CARD(fn)                              ;
*farm_size = SUM{fn,  Leps_cropfarm_areas(fn)} /  Leps_farm_no / 10000 ;
*farm_size = 27 ;
*runoff(j,slp,slt) = rnoff(j,slp,slt)    ;

*CONVERSION TO HA
 land_distr(slt,Pst,slp,asp,dst) = land_distr(slt,Pst,slp,asp,dst)          /10000       ;
 crop_distr(j,slt,Pst,slp,asp,dst) = crop_distr(j,slt,Pst,slp,asp,dst)        /10000      ;
 fallow_distr(flw,slt,Pst,slp,asp,dst) = fallow_distr(flw,slt,Pst,slp,asp,dst) /10000       ;
```

```
*Total areas
TotalInitCropArea  =  SUM{(j,slt,Pst,slp,asp,dst), crop_distr(j,slt,Pst,slp,asp,dst)     } ;
TotalInitFallArea  =  SUM{(flw,slt,Pst,slp,asp,dst), fallow_distr(flw,slt,Pst,slp,asp,dst) } ;

*land_distr(slp,slt,Pst,asp,dst)
*Field along streams with different buffer width
* LepsBufferArea15m = LepsBufferArea15m / 10000 ;
* LepsBufferArea3m  = LepsBufferArea3m / 10000 ;
* LepsBufferArea1m  = LepsBufferArea1m / 10000 ;
* LepsBufferArea500m = LepsBufferArea500m / 10000 ;
* LepsBufferArea160m = LepsBufferArea160m / 10000 ;

*WATERSHED AREA---------------------------------------------------------------
 model_ha = SUM{(slp,slt,Pst,asp,dst), land_distr(slt,Pst,slp,asp,dst) }          ;
 land_distr_share(slt,Pst,slp,asp,dst)  =  land_distr(slt,Pst,slp,asp,dst)/model_ha ;
* land_distr_farm(slp,slt,Pst,asp,dst)   = land_distr_share(slp,slt,Pst,asp,dst) * farm_size          ;
 catchmentFieldShare =  model_ha  / (CatchmentTotalArea/10000) ;

* LepsBufferArea500mShare = LepsBufferArea500m / model_ha ;
* LepsBufferArea160mShare = LepsBufferArea160m / model_ha ;
* LepsBufferArea15mShare = LepsBufferArea15m / model_ha ;
* LepsBufferArea3mShare  = LepsBufferArea3m / model_ha  ;
* LepsBufferArea1mShare  = LepsBufferArea1m / model_ha  ;

*LAND USE IN 2003 SUBSIDY STATISTICS----------------------------------
*crop_ha_rev   = SUM{(j,slp,slt,Pst,asp,dst), crop_distr(j,slt,Pst,slp,asp,dst) }      ;
*fallow_ha_rev = SUM{(flw,slp,slt,Pst,asp,dst), fallow_distr(flw,slt,Pst,slp,asp,dst) } ;
*model_ha_rev = crop_ha_rev + fallow_ha_rev   ;
*initCropShare(j,slt,Pst,slp,asp,dst)     =  crop_distr(j,slt,Pst,slp,asp,dst) / crop_ha_rev      ;
*initFallowShare(flw,slt,Pst,slp,asp,dst) =  fallow_distr(flw,slt,Pst,slp,asp,dst) / fallow_ha_rev ;

*EXPECTED LOAD FOR HA MULTIPLIED BY THE FARM SIZE AND COEFFICIENT TO MAKE IT NOT BINDING IN
INITIAL MODEL SOLUTION
*THE MODEL AT WATERSHED LEVEL
*baseTPload = modTP/model_ha*farm_size*baseTPmulti ;
baseTPload = modTP * baseTPmulti ;
*basetpload = 40 ;
baseNload = baseN_load ;
*baseNload = baseNload/model_ha*farm_size*baseNmulti  ;
*baseNload = baseNload  ;
*tpinit = baseTPload/model_ha*farm_size  ;
tpinit = baseTPload ;

TotalMaxBD       = SUM{(slt,Pst,slp,asp,dst), fallBD("gf1s",dst,asp,slp) *  land_distr(slt,Pst,slp,asp,dst)} ;
TotalBaselineBD   = SUM{(flw,slt,Pst,slp,asp,dst), fallBD(flw,dst,asp,slp) *
fallow_distr(flw,slt,Pst,slp,asp,dst)} + SUM{(j,slt,Pst,slp,asp,dst), cropBD(j,dst,asp,slp) *
crop_distr(j,slt,Pst,slp,asp,dst)}  ;
TotalMinLoad      = SUM{(slt,Pst,slp,asp,dst), loadCoeffN_flw("gf1s",slp,slt)  *  land_distr(slt,Pst,slp,asp,dst)}
 + SUM{(slt,Pst,slp,asp,dst), ([runoff_flw("gf1s",slp,slt)* [2*Pst_no("pst10") - 1.5]  * 0.0001] +
[sloss_flw("gf1s",slp,slt)   * [250 * log[Pst_no("pst10")] - 150] * 0.000001 ] ) * land_distr(slt,Pst,slp,asp,dst) }
* redfield ;
```

```
TotalBaseLoadRedfield = baseN_load + MYTVASmodBioP * redfield ;

display  crop_distr, fallow_distr, TotalInitCropArea, TotalInitFallArea, model_ha, TotalMaxBD,
TotalBaselineBD, TotalMinLoad, TotalBaseLoadRedfield ;

*RESOURCE LIMITS ------------------------------------------------------------
*parameters
*        valcon(con)          resource endowment and technology constraints right side
*        hacon(eqcon)          right hand side for all ha
*        fallLimits(fcon)      right hand side for fallow constraints
*;

* display farm_size,  fall_cost, fallLimits ;
* displaytecon, valcon, facon ;

*YIELD PARAMETERS------------------------------------------------------------
parameters
yc(crop,y)      Old experimental coefficients and P-coefficients
yieldA(crop,y)   Yield function coefficients for both mitscherlich and quadratic for A region
qyieldA         N yield parameters for quadratic specification
myieldA         N yield parameters for mitscherlich specification
;
$GDXIN C:\gamsdir\yield_parameters\Lepsjoki_yields_2param.gdx
$load yc qyieldA myieldA yieldA
;
DISPLAY yc, qyieldA, myieldA, yieldA   ;

* PRICE DATA-----------------------------------------------------------------
parameters
*Ravinteiden arvot tammikuussa 2009 Hyötylanta_ohry slides   (Koikkailainen?)
       k_price  kalium                                 /  1.22  /
       p_price cost of mineral fertilizer euro per kg        /  1.09  /
       n_price cost of mineral fertilizer euro per kg        /  1.43  /

       bd_price                                  /  10     /
       wr_price                                  /  1     /

*COSTS----------------------------------------------------------------------
*Price of land from regional_inputdata.xls:dairy_farms  region discounted with 5\%
      land_p                                        / 350 /
 ;
parameter prices(primary)
        /
         habitat  10
         nutrient 1
       /

*FOR NITROGEN INITIAL VALUES-----------------------------------------------
PARAMETERS
n1(crop) normal or prof max N dose for individual quad crops without constr
n2(crop) normal or prof max N dose for individual mitsch crops without constr
Y1(crop) prof max yield levels for quadratic crops
```

```
Y2(crop) prof max yield levels for mitscherlich crops
ninit(crop) unconstrained profit max N fertilizer dose
 ;
n1(j) $(qyieldA(j,"a") AND qyieldA(j, "d") AND qyieldA(j, "c") NE 0)
            =  [n_price/(c_price(j)) - qyieldA(j,"d")] / [2 * qyieldA(j,"c")]   ;

n2(j) $(myieldA(j,"m") AND myieldA(j, "k") AND myieldA(j, "b") NE 0)
            =  log[(c_price(j) * myieldA(j, "m")* myieldA(j, "k") * myieldA(j, "b"))
                 / n_price] / myieldA(j, "b")   ;

Y1(j) =  yieldA(j, "a") + yieldA(j, "d") * n1(j)  + yieldA(j, "c") * POWER(n1(j),2) ;
*n1("f43") = 180 ;
*n1("f44") = 180 ;
*n1("f45") = 180 ;
ninit(j) = n1(j) + n2(j);

VARIABLES
      Z                                    Total annual producer surplus for modelling period euro
      X(crop,soilt,Pstat,slope,aspct,dstnc)          Activity level ha
      Xfa(fallw,soilt,Pstat,slope,aspct,dstnc)        Fallow (ha)
      PF(crop,soilt,Pstat,slope)                   phosphorus fertiliser applied on the field kg per ha
      PS(crop,soilt,Pstat,slope,aspct,dstnc)         phosphorus stock  mg per litre
      DRP(crop,soilt,Pstat,slope,aspct,dstnc)         Dissolved reactive phosphorus
      PP(crop,soilt,Pstat,slope,aspct,dstnc)         Particle phosphorus
      FDRP(fallw,soilt,Pstat,slope,aspct,dstnc)       Dissolved reactive phosphorus from fallow land
      FPP(fallw,soilt,Pstat,slope,aspct,dstnc)        Particle phosphorus from fallow land
      FAN(fallw,soilt,slope)                       Nitrogen loss from fallow land
      NC(crop,soilt,slope)                         Nitrogen load from all possible combinations of soil slope and crop
      PU(crop,soilt,Pstat,slope,aspct,dstnc)          Phosphorus removal by plants kg for plant
      CY(crop,soilt,Pstat,slope,aspct,dstnc)         Crop yield per ha kg DM in year
*       B(j,slp,slt,Pst)        Buffer
      NF(crop,soilt,slope)                         Nitrogen fertiliser kg per ha of plant j
;
FREE VARIABLE Z ;
POSITIVE VARIABLES X, Xfa, PF, PS, DRP, PP, PU, FDRP, FPP, FAN, NF ;

* VARIABLE INITIALIZATION AND BOUNDS --------------------------------
PF.up(j,slt,Pst,slp) = 30        ;
PF.L(j,slt,Pst,slp)  = 0.1       ;
PF.lo(j,slt,Pst,slp) = 0.001      ;
NF.up(j,slt,slp)     = ninit(j)   ;
NF.L(j,slt,slp)      = ninit(j)   ;
** 0.8
NF.lo(j,slt,slp) = 0.001       ;
*HOW TO MODEL THE VARIANCE IN P STOCK CORRECTLY
PS.fx(j,slt,Pst,slp,asp,dst) =  Pst_no(Pst)      ;
*CY.up(j,slp,slt) = 15000       ;
X.up(j,slt,Pst,slp,asp,dst) = land_distr(slt,Pst,slp,asp,dst)          ;
* farm_size * 0.9 WORKED FOR EQUALING N and P baselines before grass minimum levels were introduced
-after grass limit no feasible solutions
*X.L(j,slp,slt,Pst) = land_distr_share(slp,slt,Pst) *  farm_size  * 0.5   ;
*X.L(j,slp,slt,Pst,asp,dst) = N_cropdistr_hom('1-1',j,slp,slt,Pst)   ;
```

```
X.L(j,slt,Pst,slp,asp,dst)      = land_distr(slt,Pst,slp,asp,dst)        ;
Xfa.L(flw,slt,Pst,slp,asp,dst)  = land_distr(slt,Pst,slp,asp,dst)    / 2  ;
Xfa.up(flw,slt,Pst,slp,asp,dst) = land_distr(slt,Pst,slp,asp,dst)        ;

* WETLANDS!!!!!!!!!!!!!!!!!
*Xfa.L('ffn',slp,slt,Pst) = 0       ;
*Xfa.fx('ffn',slp,slt,Pst) = 0       ;

*wetlandEffectNred.fx = 1 ;
*wetlandEffectPred.fx = 1 ;

*Xfa.L("ffn",slp,slt,Pst) = 0                 ;
*Xfa.up("ffn",slp,slt,Pst) = farm_size * 0.05    ;
*-----------------------------------------------------------------------

* EQUATIONS -------------------------------------------------------
EQUATIONS
        MaxBenefits                           objective function to maximise social welfare
        MaxBenConst                             objective function to maximise social welfare (no bd or nutrient
price)
        PrivateProfit                         objective function to maximise crop yield private profit
        MaxBiodiversity                         objective function to maximise biodiversity
        MinNutrientLoad                          objective function to minimise nutrient load

        tech(con)                             technological and political constraints
        land                                amount of land (currently reduction allowed)
        landistr(slope,soilt,Pstat,aspct,dstnc)       trying to keep the soil and slope combinations fixed
*         fallowConstraint(fcon)
        phobal(crop,slope,soilt,Pstat,aspct,dstnc)     phosphorus balance
*         Pupptake(j,slp,slt,Pst)       definition of phosphorus uptake by plants
        DRPload(crop,slope,soilt,Pstat,aspct,dstnc)     definition of dissolved phosphorus loss
        PPload(crop,slope,soilt,Pstat,aspct,dstnc)      definition of particle phosphorus loss
        DRPfall(fallw,slope,soilt,Pstat,aspct,dstnc)    definition of dissolved phosphorus loss for fallow
        PPfall(fallw,slope,soilt,Pstat,aspct,dstnc)     definition of particle phosphorus loss for fallow
        Nfall(fallw,slope,soilt)                   definition of N leaching for fallow
*         Ncrop(j,slope,soilt,Pstat,aspct,dstce)          definition of N leaching for crops
        NcropREV(crop,slope,soilt)                  REVISED directly from ICECREAM definition of N leaching for
crops
        loadP                               P load
        loadN                               N load
        loadNP                              constraint for N and P
        cropyield(crop,slope,soilt,Pstat,aspct,dstnc)   Total crop yield for year in kgs DM
        DRPMeanSlopeSoil(crop)
        PPMeanSlopeSoil(crop)
        DRPfall_mean(fallw)                     definition of dissolved phosphorus loss for fallow
        PPfall_mean(fallw)                     definition of particle phosphorus loss for fallow
        meanloadP                           the restricted P load
        minimumgrass                          test for different formulation of grass minimum requirement
constraint
        wetlandP
        wetlandN
;
```

```
*YIELD FUNCTIONS------------------------------------------------------------

cropyield(j,slp,slt,Pst,asp,dst)..  CY(j,slt,Pst,slp,asp,dst)  =E=
{
* quadratic base yield
        yieldA(j, "a")
* positive slope for nitrogen
        + yieldA(j, "d") * (NF(j,slt,slp) )
*negative term for nitrogen
        + yieldA(j, "c") * POWER((NF(j,slt,slp) ),2)
* mitserlich base yield
        + yieldA(j,"m")* [
                1 - yieldA(j,"k") *
* mitserlich nitrogen curve
                exp(-(yieldA(j,"b")*(NF(j,slt,slp))))
               ] +
         [
* Saari 1995 p68 for P-fertiliser and manure effect   (ry/ha)
          + yc(j,"pcon")
          + yc(j,"pyc1") * (PF(j,slt,Pst,slp) )**0.5
          - yc(j,"pyc2") * PS(j,slt,Pst,slp,asp,dst) * (PF(j,slt,Pst,slp) )**0.5
          - yc(j,"pyc3") * (SQR[PF(j,slt,Pst,slp) ]/Pst_no(Pst) ) * yc(j,"pcoe")
          + yc(j,"pyc4") * (PF(j,slt,Pst,slp) )/Pst_no(Pst)  * yc(j,"pcoe")
*muunnos ry ha
         ] / yc(j, "sry")
 }
*Transform to dry matter (DM) yield
*  * (nuval(j, "ka")*0.01)

 ;

$ontext
      [
* Myyra Mitscherlich for P-stock effect  (kg/ha?)
          yc(j,"m") * [1 - yc(j,"k") * exp(-(yc(j,"b")* PS(j,slp,slt) ))]
$offtext

*RESOURCE AND TECHNOLOGY CONSTRAINTS -----------------------------

* Crop production constraints and balances
land.. SUM{(j,slp,slt,Pst,asp,dst), X(j,slt,Pst,slp,asp,dst)} + SUM{(flw,slp,slt,Pst,asp,dst),
Xfa(flw,slt,Pst,slp,asp,dst)} =E= model_ha   ;

*FIX FOR PST
*tech(con) .. SUM{(j,slp,slt,Pst,asp,dst), tecon(j, con) * X(j,slt,Pst,slp,asp,dst) } =L= valcon(con)  ;

landistr(slp,slt,Pst,asp,dst).. SUM{j, X(j,slt,Pst,slp,asp,dst)} + SUM{flw, Xfa(flw,slt,Pst,slp,asp,dst)}  =E=
land_distr(slt,Pst,slp,asp,dst)  ;

*fallowConstraint(fcon)..  SUM{(flw,slp,slt,Pst,asp,dst), facon(flw, fcon) * Xfa(flw,slp,slt,Pst,asp,dst)}  =L=
fallLimits(fcon)   ;
```

```
*minimumgrass..   SUM{(slp,slt,Pst), X('f43',slp,slt,Pst)} =G= farm_size * grass_land_share   ;
*+ SUM{(slp,slt,Pst), X('f44',slp,slt,Pst)} +  SUM{(slp,slt,Pst), X('f45',slp,slt,Pst)}
* phosphorus contained in seeds is removed  - what about planted seeds, husks etc.
*Pupptake(j,slp,slt,Pst)  .. PU(j,slp,slt,Pst)   =E= nuval(j,"pho")/1000 * CY(j,slp,slt,Pst)  * X(j,slp,slt,Pst)
;

DRPload(j,slp,slt,Pst,asp,dst).. DRP(j,slt,Pst,slp,asp,dst) =E=  runoff(j,slp,slt)* [2*(PS(j,slt,Pst,slp,asp,dst)  +
0.01*[PF(j,slt,Pst,slp) ]) - 1.5]  * 0.0001          ;

PPload(j,slp,slt,Pst,asp,dst)..  PP(j,slt,Pst,slp,asp,dst)  =E=  sloss(j,slp,slt) * [250 * log(PS(j,slt,Pst,slp,asp,dst)
+ 0.01 * PF(j,slt,Pst,slp) )- 150] * 0.000001  ;

DRPfall(flw,slp,slt,Pst,asp,dst).. FDRP(flw,slt,Pst,slp,asp,dst) =E= runoff_flw(flw,slp,slt)  * [2*Pst_no(Pst) -
1.5] * 0.0001           ;

PPfall(flw,slp,slt,Pst,asp,dst)..  FPP(flw,slt,Pst,slp,asp,dst)  =E= sloss_flw(flw,slp,slt)   * [250 *
log[Pst_no(Pst)] - 150] * 0.000001  ;

*Ncrop(j,slp,slt)..    NC(j,slt,slp)     =E= loadCoeffN(j,slp,slt)    *  EXP[0.71 * ((NF(j,slt,slp) )  / nmax(j,slt) - 1 ) ] ;

NcropREV(j,slp,slt)..  NC(j,slt,slp) =E= Ncoeff_frt_b(j,slp,slt) * exp[NF(j,slt,slp) * Ncoeff_frt_c(j,slp,slt)] ;

Nfall(flw,slp,slt)..   FAN(flw,slt,slp)  =E= loadCoeffN_flw(flw,slp,slt)          ;

*wetlandP.. wetlandEffectPred =E= 1 -(wtldCoefficientP * [SUM{(slp,slt,Pst), Xfa('ffn', slp,slt,Pst) } /
farm_size ])    ;
*wetlandN.. wetlandEffectNred =E= 1 -(wtldCoefficientN * [SUM{(slp,slt,Pst), Xfa('ffn', slp,slt,Pst) } /
farm_size ])    ;

* CONSTRAINT FOR ABATEMENT COST PURPOSES-----------------------------------
*NOTE PP PARAMETERS MIGHT BE WRONG SOMEWHERE
loadP.. (SUM{(j,slp,slt,Pst,asp,dst), DRP(j,slt,Pst,slp,asp,dst)* X(j,slt,Pst,slp,asp,dst)} +
SUM{(flw,slp,slt,Pst,asp,dst), FDRP(flw,slt,Pst,slp,asp,dst)* Xfa(flw,slt,Pst,slp,asp,dst)}
  + (SUM{(j,slp,slt,Pst,asp,dst), PP(j,slt,Pst,slp,asp,dst) * X(j,slt,Pst,slp,asp,dst)} +
SUM{(flw,slp,slt,Pst,asp,dst), FPP(flw,slt,Pst,slp,asp,dst) * Xfa(flw,slt,Pst,slp,asp,dst)} )/bio_coef
*   ) * wetlandEffectPred
  ) =L= baseTPload * (1 - red_l)     ;
*LOADP.. SUM{j, loadcoef_DRP(j,slp,slt)* [2*(PS(j,slp,slt)  + 0.01*[PF(j,slp,slt)+PM(j,slp,slt) ]) - 1.5]  *
POWER(10, -4)  * X(j,slp,slt) } + (loadcoeff_DRP * [2*PStm - 1.5] * POWER(10, -4)  * Xfa )  + (SUM{j,
(loadcoef_PP(j,slp,slt) *  [250 * log(PS(j,slp,slt)  + 0.01 * [PF(j,slp,slt)+PM(j,slp,slt) ])- 150] * POWER(10, -6)*
X(j,slp,slt) )}+ (loadcoeff_PP   * [250 * log(PStm) - 150] * POWER(10, -6) * Xfa))/0.16   =L= basetpload * (1 -
red_l)     ;
loadN .. [ SUM{(j,slp,slt,Pst,asp,dst), NC(j,slt,slp)  * X(j,slt,Pst,slp,asp,dst)} +  SUM{(flw,slp,slt,Pst,asp,dst),
FAN(flw,slt,slp) * Xfa(flw,slt,Pst,slp,asp,dst)}
*        ] * retention * wetlandEffectNred
       ] =L= basenload * (1 - red_l)
;

loadNP..  [ SUM{(j,slp,slt,Pst,asp,dst), NC(j,slt,slp)  * X(j,slt,Pst,slp,asp,dst)} +  SUM{(flw,slp,slt,Pst,asp,dst),
FAN(flw,slt,slp) * Xfa(flw,slt,Pst,slp,asp,dst)}
        ]
*       ] * retention * wetlandEffectNred
```

```
       +   redfield * [ SUM{(j,slp,slt,Pst,asp,dst), DRP(j,slt,Pst,slp,asp,dst)* X(j,slt,Pst,slp,asp,dst)} +
SUM{(flw,slp,slt,Pst,asp,dst), FDRP(flw,slt,Pst,slp,asp,dst)* Xfa(flw,slt,Pst,slp,asp,dst)}
  + SUM{(j,slp,slt,Pst,asp,dst), PP(j,slt,Pst,slp,asp,dst) * X(j,slt,Pst,slp,asp,dst)} + SUM{(flw,slp,slt,Pst,asp,dst),
FPP(flw,slt,Pst,slp,asp,dst) * Xfa(flw,slt,Pst,slp,asp,dst)}
                   ]
 =L= [redfield * basetpload + basenload] * (1 - red_l)     ;

*MEAN VERSIONS OF LOAD CONSTRAINTS
*meanloadP.. SUM{(j,slp,slt), DRP_m(j,slp,slt)* X(j,slp,slt)} +  SUM{(flw,slp,slt), FDRP_m(flw,slp,slt)*
Xfa(flw,slp,slt)}
*  + (SUM{(j,slp,slt), PP_m(j,slp,slt) * X(j,slp,slt)} + SUM{(flw,slp,slt), FPP_m(flw,slp,slt) * Xfa(flw,slp,slt)}
)/0.16
*   =L= basetpload * (1 - red_l)     ;


*OBJECTIVE FUNCTIONS--------------------------------------------------------

* MAX  CROP PROFIT - FIXED COSTS
PrivateProfit.. Z =E= SUM{(j,slp,slt,Pst,asp,dst),
* [ ntlsubs(j) + capsubs(j) + lfasubs(j)
[
        c_price(j) *  CY(j,slt,Pst,slp,asp,dst)
      - [n_price* NF(j,slt,slp) + p_price * PF(j,slt,Pst,slp)]  -  fix_c(j)
 ]* X(j,slt,Pst,slp,asp,dst)
}
* Revenue/cost from fallow 2009
+ SUM{(flw,slp,slt,Pst,asp,dst), - fall_cost(flw)  *    Xfa(flw,slt,Pst,slp,asp,dst)    }
* * (capfallsubs(flw) + lfafallsubs(flw)  ) }
;

* MAX  BIODIVERSITY - FIXED COSTS
MaxBiodiversity.. Z =E= SUM{(j,slp,slt,Pst,asp,dst),
* [ ntlsubs(j) + capsubs(j) + lfasubs(j)
[
*        c_price(j) *  CY(j,slt,Pst,slp,asp,dst)
       prices("habitat") * cropBD(j,dst,asp,slp)
*       - wr_price * [(DRP(j,slt,Pst,slp,asp,dst) + PP(j,slt,Pst,slp,asp,dst)) * redfield + NC(j,slt,slp) ]
*       - [n_price* NF(j,slt,slp) + p_price * PF(j,slt,Pst,slp)]
-  fix_c(j)
 ]* X(j,slt,Pst,slp,asp,dst)
}
+ SUM{(flw,slp,slt,Pst,asp,dst),
[ bd_price * fallBD(flw,dst,asp,slp)
*  - wr_price * [(FDRP(flw,slt,Pst,slp,asp,dst) + FPP(flw,slt,Pst,slp,asp,dst))* redfield + FAN(flw,slt,slp) ]
* +(capfallsubs(flw) + lfafallsubs(flw)
  - fall_cost(flw)
 ] * Xfa(flw,slt,Pst,slp,asp,dst)       }
;

* MIN  NUTRIENT LOSS - FIXED COSTS
MinNutrientLoad.. Z =E= SUM{(j,slp,slt,Pst,asp,dst),
* [ ntlsubs(j) + capsubs(j) + lfasubs(j) +
```

```
[
*        c_price(j) *  CY(j,slt,Pst,slp,asp,dst)
*       + bd_price * cropBD(j,dst,asp,slp)
       - prices("nutrient") * [(DRP(j,slt,Pst,slp,asp,dst) + PP(j,slt,Pst,slp,asp,dst)) * redfield + NC(j,slt,slp) ]
*       - [n_price* NF(j,slt,slp) + p_price * PF(j,slt,Pst,slp)]
-  fix_c(j)
 ]* X(j,slt,Pst,slp,asp,dst)
}
* Revenue/cost from fallow 2009
+ SUM{(flw,slp,slt,Pst,asp,dst),
[
*bd_price * fallBD(flw,dst,asp,slp)
  - wr_price * [(FDRP(flw,slt,Pst,slp,asp,dst) + FPP(flw,slt,Pst,slp,asp,dst))* redfield + FAN(flw,slt,slp) ]
*  + (capfallsubs(flw) + lfafallsubs(flw)
  - fall_cost(flw)
 ] * Xfa(flw,slt,Pst,slp,asp,dst)       }
;

* MAX "SOCIAL BENEFITS"
MaxBenefits.. Z =E= SUM{(j,slp,slt,Pst,asp,dst),
* [ ntlsubs(j) + capsubs(j) + lfasubs(j) +
[
        c_price(j) *  CY(j,slt,Pst,slp,asp,dst)
       + prices("habitat") * cropBD(j,dst,asp,slp)
       - prices("nutrient") * [(DRP(j,slt,Pst,slp,asp,dst) + PP(j,slt,Pst,slp,asp,dst)) * redfield + NC(j,slt,slp) ]
       - [n_price* NF(j,slt,slp) + p_price * PF(j,slt,Pst,slp)] -  fix_c(j)
 ]* X(j,slt,Pst,slp,asp,dst)
}
* Revenue/cost from fallow 2009
+ SUM{(flw,slp,slt,Pst,asp,dst),
[ prices("habitat") * fallBD(flw,dst,asp,slp)
  - prices("nutrient") * [(FDRP(flw,slt,Pst,slp,asp,dst) + FPP(flw,slt,Pst,slp,asp,dst))* redfield + FAN(flw,slt,slp)
]
*  + (capfallsubs(flw) + lfafallsubs(flw)
  - fall_cost(flw)
 ] * Xfa(flw,slt,Pst,slp,asp,dst)       }
;

* MAX "SOCIAL BENEFITS"
MaxBenConst.. Z =E= SUM{(j,slp,slt,Pst,asp,dst),
* [ ntlsubs(j) + capsubs(j) + lfasubs(j) +
[
        c_price(j) *  CY(j,slt,Pst,slp,asp,dst)
*       + prices("habitat") * cropBD(j,dst,asp,slp)
*       - prices("nutrient") * [(DRP(j,slt,Pst,slp,asp,dst) + PP(j,slt,Pst,slp,asp,dst)) * redfield + NC(j,slt,slp) ]
       - [n_price* NF(j,slt,slp) + p_price * PF(j,slt,Pst,slp)] -  fix_c(j)
 ]* X(j,slt,Pst,slp,asp,dst)
}
* Revenue/cost from fallow 2009
+ SUM{(flw,slp,slt,Pst,asp,dst),
[
* prices("habitat") * fallBD(flw,dst,asp,slp)
```

```
*  - prices("nutrient") * [(FDRP(flw,slt,Pst,slp,asp,dst) + FPP(flw,slt,Pst,slp,asp,dst))* redfield +
FAN(flw,slt,slp) ]
*  + (capfallsubs(flw) + lfafallsubs(flw)
  - fall_cost(flw)
 ] * Xfa(flw,slt,Pst,slp,asp,dst)       }
;

*RESULT PARAMETERS
parameters
* Privat Profit (PP), Biodiversity (BD), Nutrient Load (NL), Social Welfare (SW)
TotalYieldValuePP
TotalBdValuePP
TotalLoadValuePP
TotalYieldValuePerHaPP
TotalBdValuePerHaPP
TotalLoadValuePerHaPP
TotalLoadPP
TotalBiodiversityPP
CropLoadPP(crop,soilt,Pstat,slope,aspct,dstnc)
CropBiodiversityPP(crop,soilt,Pstat,slope,aspct,dstnc)
FallLoadPP(fallw,soilt,Pstat,slope,aspct,dstnc)
FallBiodiversityPP(fallw,soilt,Pstat,slope,aspct,dstnc)
YieldValuePP(crop,soilt,Pstat,slope,aspct,dstnc)
CropBdValuePP(crop,soilt,Pstat,slope,aspct,dstnc)
CropLoadValuePP(crop,soilt,Pstat,slope,aspct,dstnc)
FallBdValuePP(fallw,soilt,Pstat,slope,aspct,dstnc)
FallLoadValuePP(fallw,soilt,Pstat,slope,aspct,dstnc)
cropSharePP(crop,soilt,Pstat,slope,aspct,dstnc)
fallSharePP(fallw,soilt,Pstat,slope,aspct,dstnc)
BiodiversityPP(soilt,Pstat,slope,aspct,dstnc)
LoadPP(soilt,Pstat,slope,aspct,dstnc)
*BD
TotalYieldValueBD
TotalBdValueBD
TotalLoadValueBD
TotalYieldValuePerHaBD
TotalBdValuePerHaBD
TotalLoadValuePerHaBD
TotalLoadBD
TotalBiodiversityBD
CropLoadBD(crop,soilt,Pstat,slope,aspct,dstnc)
CropBiodiversityBD(crop,soilt,Pstat,slope,aspct,dstnc)
FallLoadBD(fallw,soilt,Pstat,slope,aspct,dstnc)
FallBiodiversityBD(fallw,soilt,Pstat,slope,aspct,dstnc)
YieldValueBD(crop,soilt,Pstat,slope,aspct,dstnc)
CropBdValueBD (crop,soilt,Pstat,slope,aspct,dstnc)
CropLoadValueBD(crop,soilt,Pstat,slope,aspct,dstnc)
FallBdValueBD(fallw,soilt,Pstat,slope,aspct,dstnc)
FallLoadValueBD(fallw,soilt,Pstat,slope,aspct,dstnc)
cropShareBD(crop,soilt,Pstat,slope,aspct,dstnc)
fallShareBD(fallw,soilt,Pstat,slope,aspct,dstnc)
BiodiversityBD(soilt,Pstat,slope,aspct,dstnc)
```

```
LoadBD(soilt,Pstat,slope,aspct,dstnc)
*NL
TotalYieldValueNL
TotalBdValueNL
TotalLoadValueNL
TotalYieldValuePerHaNL
TotalBdValuePerHaNL
TotalLoadValuePerHaNL
TotalLoadNL
TotalBiodiversityNL
CropLoadNL(crop,soilt,Pstat,slope,aspct,dstnc)
CropBiodiversityNL(crop,soilt,Pstat,slope,aspct,dstnc)
FallLoadNL(fallw,soilt,Pstat,slope,aspct,dstnc)
FallBiodiversityNL(fallw,soilt,Pstat,slope,aspct,dstnc)
YieldValueNL(crop,soilt,Pstat,slope,aspct,dstnc)
CropBdValueNL(crop,soilt,Pstat,slope,aspct,dstnc)
CropLoadValueNL(crop,soilt,Pstat,slope,aspct,dstnc)
FallBdValueNL(fallw,soilt,Pstat,slope,aspct,dstnc)
FallLoadValueNL(fallw,soilt,Pstat,slope,aspct,dstnc)
cropShareNL(crop,soilt,Pstat,slope,aspct,dstnc)
fallShareNL(fallw,soilt,Pstat,slope,aspct,dstnc)
BiodiversityNL(soilt,Pstat,slope,aspct,dstnc)
LoadNL(soilt,Pstat,slope,aspct,dstnc)
*SW
TotalYieldValueSW
TotalBdValueSW
TotalLoadValueSW
TotalYieldValuePerHaSW
TotalBdValuePerHaSW
TotalLoadValuePerHaSW
TotalLoadSW
TotalBiodiversitySW
CropLoadSW(crop,soilt,Pstat,slope,aspct,dstnc)
CropBiodiversitySW(crop,soilt,Pstat,slope,aspct,dstnc)
FallLoadSW(fallw,soilt,Pstat,slope,aspct,dstnc)
FallBiodiversitySW(fallw,soilt,Pstat,slope,aspct,dstnc)
YieldValueSW(crop,soilt,Pstat,slope,aspct,dstnc)
CropBdValueSW(crop,soilt,Pstat,slope,aspct,dstnc)
CropLoadValueSW(crop,soilt,Pstat,slope,aspct,dstnc)
FallBdValueSW(fallw,soilt,Pstat,slope,aspct,dstnc)
FallLoadValueSW(fallw,soilt,Pstat,slope,aspct,dstnc)
cropShareSW(crop,soilt,Pstat,slope,aspct,dstnc)
fallShareSW(fallw,soilt,Pstat,slope,aspct,dstnc)
BiodiversitySW(soilt,Pstat,slope,aspct,dstnc)
LoadSW(soilt,Pstat,slope,aspct,dstnc)
TotFallShareSW
TotBdvFallShareSW
;


* MODELS ------------------------------------------------------------
MODEL PROFMAX / PrivateProfit, cropyield
```

```
*land
* minimumgrass
 landistr
* tech
* fallowConstraint
 DRPload, PPload, DRPfall, PPfall
 NcropREV
 Nfall
* loadP
* wetlandP
* wetlandN
/   ;
SOLVE PROFMAX using NLP maximising Z ;


*RESULTS FOR PRIVATE PROFIT
CropLoadPP(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )          =  [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]  ;
FallLoadPP(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )        =  [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ]   ;
CropBiodiversityPP(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )    =  cropBD(j,dst,asp,slp)   ;
FallBiodiversityPP(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )  =  fallBD(flw,dst,asp,slp)   ;

YieldValuePP(j,slt,Pst,slp,asp,dst)     =   c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) * X.L(j,slt,Pst,slp,asp,dst)  ;
CropBdValuePP(j,slt,Pst,slp,asp,dst)    =   bd_price * cropBD(j,dst,asp,slp)   * X.L(j,slt,Pst,slp,asp,dst)        ;
FallBdValuePP(flw,slt,Pst,slp,asp,dst)  =   bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst)     ;
CropLoadValuePP(j,slt,Pst,slp,asp,dst)  =   wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) + PP.L(j,slt,Pst,slp,asp,dst))
* redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)   ;
FallLoadValuePP(flw,slt,Pst,slp,asp,dst) =   wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst)  ;

TotalYieldValuePP  =   SUM{(j,slt,Pst,slp,asp,dst), c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) *
X.L(j,slt,Pst,slp,asp,dst) } ;
TotalBdValuePP    =   SUM{(j,slt,Pst,slp,asp,dst), bd_price * cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) }
+  SUM{(flw,slt,Pst,slp,asp,dst), bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalLoadValuePP   =   SUM{(j,slt,Pst,slp,asp,dst), wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +
SUM{(flw,slt,Pst,slp,asp,dst), wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))*
redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

TotalYieldValuePerHaPP    =  TotalYieldValuePP    / model_ha ;
TotalBdValuePerHaPP       =  TotalBdValuePP       / model_ha ;
TotalLoadValuePerHaPP     =  TotalLoadValuePP     / model_ha ;

TotalLoadPP     =  SUM{(j,slt,Pst,slp,asp,dst), [(DRP.L(j,slt,Pst,slp,asp,dst) + PP.L(j,slt,Pst,slp,asp,dst)) *
redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +  SUM{(flw,slt,Pst,slp,asp,dst),
[(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] *
Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalBiodiversityPP  =  SUM{(j,slt,Pst,slp,asp,dst),  cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) } +
SUM{(flw,slt,Pst,slp,asp,dst),  fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
```

```
cropSharePP(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )        =  X.L(j,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;
fallSharePP(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )   =  Xfa.L(flw,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;

BiodiversityPP(slt,Pst,slp,asp,dst)   =  SUM{j,cropSharePP(j,slt,Pst,slp,asp,dst) *
CropBiodiversityPP(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallSharePP(flw,slt,Pst,slp,asp,dst) *
FallBiodiversityPP(flw,slt,Pst,slp,asp,dst)}   ;
LoadPP(slt,Pst,slp,asp,dst)          =  SUM{j,cropSharePP(j,slt,Pst,slp,asp,dst) *
CropLoadPP(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallSharePP(flw,slt,Pst,slp,asp,dst) *
FallLoadPP(flw,slt,Pst,slp,asp,dst)}   ;

*Setting the individual environmental aspects equal to value of max profit
TotalMaxProfit = Z.L          ;
prices("habitat") =  Z.L / TotalMaxBD   ;
prices("nutrient") =  Z.L / TotalMinLoad ;

display bd_price, wr_price ;

MODEL BDMAX / MaxBiodiversity, cropyield
*tech
*minimumgrass
*land
 landistr
* fallowConstraint
 DRPload, PPload
 DRPfall
 PPfall
 NcropREV
 Nfall
* loadN
* wetlandP
* wetlandN
/   ;
SOLVE BDMAX using NLP maximising Z ;

CropLoadBD(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )          =  [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]   ;
FallLoadBD(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )      =  [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ]   ;
CropBiodiversityBD(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )    =  cropBD(j,dst,asp,slp)    ;
FallBiodiversityBD(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )  =  fallBD(flw,dst,asp,slp)  ;

YieldValueBD(j,slt,Pst,slp,asp,dst)      =   c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) * X.L(j,slt,Pst,slp,asp,dst)  ;
CropBdValueBD(j,slt,Pst,slp,asp,dst)     =   bd_price * cropBD(j,dst,asp,slp)   * X.L(j,slt,Pst,slp,asp,dst)        ;
FallBdValueBD(flw,slt,Pst,slp,asp,dst)   =   bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst)     ;
CropLoadValueBD(j,slt,Pst,slp,asp,dst)   =   wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)   ;
FallLoadValueBD(flw,slt,Pst,slp,asp,dst) =   wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst)  ;
```

```
TotalYieldValueBD  =   SUM{(j,slt,Pst,slp,asp,dst), c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) *
X.L(j,slt,Pst,slp,asp,dst) } ;
TotalBdValueBD     =   SUM{(j,slt,Pst,slp,asp,dst), bd_price * cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) }
+  SUM{(flw,slt,Pst,slp,asp,dst), bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalLoadValueBD   =   SUM{(j,slt,Pst,slp,asp,dst), wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +
SUM{(flw,slt,Pst,slp,asp,dst), wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))*
redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

TotalYieldValuePerHaBD    = TotalYieldValueBD    /model_ha ;
TotalBdValuePerHaBD       = TotalBdValueBD       /model_ha ;
TotalLoadValuePerHaBD     = TotalLoadValueBD     /model_ha ;

TotalLoadBD         = SUM{(j,slt,Pst,slp,asp,dst), [(DRP.L(j,slt,Pst,slp,asp,dst) + PP.L(j,slt,Pst,slp,asp,dst)) *
redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +  SUM{(flw,slt,Pst,slp,asp,dst),
[(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] *
Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalBiodiversityBD  =  SUM{(j,slt,Pst,slp,asp,dst),  cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) } +
SUM{(flw,slt,Pst,slp,asp,dst),  fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

cropShareBD(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )        =  X.L(j,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;
fallShareBD(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )   =  Xfa.L(flw,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;

BiodiversityBD(slt,Pst,slp,asp,dst)   =  SUM{j,cropSharePP(j,slt,Pst,slp,asp,dst) *
CropBiodiversityBD(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareBD(flw,slt,Pst,slp,asp,dst) *
FallBiodiversityBD(flw,slt,Pst,slp,asp,dst)}   ;
LoadBD(slt,Pst,slp,asp,dst)          =  SUM{j,cropSharePP(j,slt,Pst,slp,asp,dst) *
CropLoadBD(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareBD(flw,slt,Pst,slp,asp,dst) *
FallLoadBD(flw,slt,Pst,slp,asp,dst)}   ;

MODEL NLOADMIN / MinNutrientLoad, cropyield
*tech
*minimumgrass
*land
 landistr
* fallowConstraint
 DRPload, PPload
 DRPfall
 PPfall
 NcropREV
 Nfall
* loadN
/   ;
SOLVE NLOADMIN using NLP maximising Z ;

CropLoadNL(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )          =  [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]   ;
FallLoadNL(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )        =  [(FDRP.L(flw,slt,Pst,slp,asp,dst)
+ FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ]   ;
CropBiodiversityNL(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )   =  cropBD(j,dst,asp,slp)    ;
```

```
FallBiodiversityNL(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )  =  fallBD(flw,dst,asp,slp)  ;

YieldValueNL(j,slt,Pst,slp,asp,dst)      =   c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) * X.L(j,slt,Pst,slp,asp,dst)  ;
CropBdValueNL(j,slt,Pst,slp,asp,dst)     =   bd_price * cropBD(j,dst,asp,slp)   * X.L(j,slt,Pst,slp,asp,dst)        ;
FallBdValueNL(flw,slt,Pst,slp,asp,dst)   =   bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst)     ;
CropLoadValueNL(j,slt,Pst,slp,asp,dst)   =   wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)   ;
FallLoadValueNL(flw,slt,Pst,slp,asp,dst) =   wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst)  ;

TotalYieldValueNL  =   SUM{(j,slt,Pst,slp,asp,dst), c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) *
X.L(j,slt,Pst,slp,asp,dst) } ;
TotalBdValueNL     =   SUM{(j,slt,Pst,slp,asp,dst), bd_price * cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) }
+  SUM{(flw,slt,Pst,slp,asp,dst), bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalLoadValueNL  =   SUM{(j,slt,Pst,slp,asp,dst), wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +
SUM{(flw,slt,Pst,slp,asp,dst), wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))*
redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

TotalYieldValuePerHaNL    =  TotalYieldValueNL    /model_ha ;
TotalBdValuePerHaNL       =  TotalBdValueNL       /model_ha ;
TotalLoadValuePerHaNL     =  TotalLoadValueNL     /model_ha ;

TotalLoadNL      =  SUM{(j,slt,Pst,slp,asp,dst), [(DRP.L(j,slt,Pst,slp,asp,dst) + PP.L(j,slt,Pst,slp,asp,dst)) *
redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +  SUM{(flw,slt,Pst,slp,asp,dst),
[(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] *
Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalBiodiversityNL  =  SUM{(j,slt,Pst,slp,asp,dst),  cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) } +
SUM{(flw,slt,Pst,slp,asp,dst),  fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

cropShareNL(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )        =  X.L(j,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;
fallShareNL(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )   =  Xfa.L(flw,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;

BiodiversityNL(slt,Pst,slp,asp,dst)   =  [SUM{j,cropShareNL(j,slt,Pst,slp,asp,dst) *
CropBiodiversityNL(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareNL(flw,slt,Pst,slp,asp,dst) *
FallBiodiversityNL(flw,slt,Pst,slp,asp,dst)}] / 2   ;
LoadNL(slt,Pst,slp,asp,dst)           =  [SUM{j,cropShareNL(j,slt,Pst,slp,asp,dst) *
CropLoadNL(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareNL(flw,slt,Pst,slp,asp,dst) *
FallLoadNL(flw,slt,Pst,slp,asp,dst)}] / 2   ;

MODEL WELFAREMAX / MaxBenefits, cropyield
*tech
*land
 landistr
* fallowConstraint
 DRPload, PPload
 DRPfall
 PPfall
 NcropREV
 Nfall
```

```
/   ;
SOLVE WELFAREMAX using NLP maximising Z ;

MODEL ConstrainedMax / MaxBenConst, cropyield
*tech
*land
 landistr
* fallowConstraint
 DRPload, PPload
 DRPfall
 PPfall
 NcropREV
 Nfall
 loadNP

/   ;
SOLVE ConstrainedMax using NLP maximising Z ;

*display TotalYieldValuePerHaPP, TotalBdValuePerHaPP, TotalLoadValuePerHaPP ;
*display TotalLoadPP, TotalBiodiversityPP ;

display c_price, bd_price, wr_price, fall_cost, fix_c  ;


*FOR SCENARIO COMPARISON AND REPORT WRITING
sets    alli all items in the scenario comparison / habitat, nutrient, Total /
        measures    / NetIncome, NetHabitat, NetNutLoad, TotFallowShare, TotBdvFallShare /
        plotmsrs(measures) / NetIncome, NetHabitat, NetNutLoad /
;
Parameter summary(alli,measures)   Farm Summary ;

display prices, alli, primary ;
```

Loop and sensitivity analysis

```
*SENSITIVITY ANALYSIS LOOPS
$onsymlist offsymxref
option limrow=0;
option solprint=off;
option limcol=0;
$include LepsCrop_6_2011.gms
*$include LepsCropReport_3_2011.gms

*step 1 - setup scenarios
sets    ordr /ScenarioSetup, ScenarioResults/
        scenarios        / base, zerozero, pScen1*pScen100, hab1*hab10, habZ1*habZ10, habH1*habH10,
habHZ1*habHZ10, pScenH1*pScenH100, pScenB1*pScenB100, pScenN1*pScenN100 /
        priceScen(scenarios)  / zerozero, pScen1*pScen100, hab1*hab10,habZ1*habZ10, habH1*habH10,
habHZ1*habHZ10, pScenH1*pScenH100, pScenB1*pScenB100, pScenN1*pScenN100 /
        priceScen1(priceScen) low price (except when nutrient price is constant)                    /
pScen1*pScen100 /
        priceScen2(priceScen) high price  (except when nutrient price is constant)                  /
pScenH1*pScenH100 /
        priceScen3(priceScen) high bd price low nutrient  (except when nutrient price is constant)      /
pScenB1*pScenB100 /
        priceScen4(priceScen) high nutrient price low bd price (except when nutrient price is constant) /
pScenN1*pScenN100 /
        habPchange(scenarios)  only habitat price changes  / hab1*hab10 /
        habPHchange(scenarios) only habitat price changes  / habH1*habH10 /
        habZeroP1(scenarios)   /  habZ1*habZ10  /
        habZeroP2(scenarios)   /  habHZ1*habHZ10  /

;
*alias(priceScen, priceScen1)
*display Newscenarios ;

parameter summary(alli,measures)       Farm Summary                  ;
parameter savsumm(ordr,*,alli,scenarios) Comparative Farm Summary;
parameter scenPrice(primary,scenarios)   Price alterations by scenario   ;
parameter scen1(primary,scenarios)   Price alterations by scenario   ;
parameter recPrice(primary,scenarios)   Price recorded within loop  ;
parameter ite1 iterations    /1 /  ;

*FOR CHECKING THE SCENARIO LOGIC
 LOOP(priceScen1,
   scen1("habitat",priceScen1)  =  mod(priceScen1.ord,10)/10    ;
   scen1("nutrient",priceScen1) =  ceil(priceScen1.ord/10)/10 ;
 );

scenprice(primary,scenarios) = prices(primary)            ;

*step 2 save data
parameter savprice(primary) saved non-market commodity prices;
savprice(primary)=prices(primary) ;
```

```
*DECREASING LOOP FROM THE BASE PRICE DOWN TO 0\% AT 10\% INTERVAL
* habitat = 0.1*baseline-baseline, nutrient = 0-0.9*baseline
 LOOP(priceScen1,
   scenPrice("habitat",priceScen1)  =  prices("habitat") * (1 - mod(priceScen1.ord,10)/10)    ;
   scenPrice("nutrient",priceScen1) =  prices("nutrient") * (1 - ceil(priceScen1.ord/10)/10) ;
 );
* habitat = 0.1baseline-baseline, nutrient = baseline
 LOOP(habPchange,
   scenPrice("habitat",habPchange)  = prices("habitat") *  (1 - mod(habPchange.ord,10)/10)    ;
 );
* habitat = 0, nutrient =  0.1*baseline-baseline
 LOOP(habZeroP1,
   scenPrice("nutrient",habZeroP1)  =  prices("nutrient") * (1 - mod(habZeroP1.ord,10)/10)    ;
   scenPrice("habitat",habZeroP1)  = 0 ;
 );

*setting the initial pricess higher so that there will be no barley solution in the scenarios
prices(primary) = prices(primary) * 10     ;

 LOOP(priceScen2,
   scenPrice("habitat",priceScen2)  =  prices("habitat") * (1 - mod(priceScen2.ord,10)/10)    ;
   scenPrice("nutrient",priceScen2) =  prices("nutrient") * (1 - ceil(priceScen2.ord/10)/10) ;
 );
 LOOP(habPHchange,
   scenPrice("habitat",habPHchange )  = prices("habitat") *  (1 - mod(habPHchange.ord,10)/10)    ;
 );

prices(primary) = savprice(primary) ;
prices("habitat") = prices("habitat") * 10     ;

 LOOP(priceScen3,
   scenPrice("habitat",priceScen3)  =  prices("habitat") * (1 - mod(priceScen3.ord,10)/10)    ;
   scenPrice("nutrient",priceScen3) =  prices("nutrient") * (1 - ceil(priceScen3.ord/10)/10) ;
 );

prices(primary) = savprice(primary) ;
prices("nutrient") = prices("nutrient") * 10     ;

 LOOP(priceScen4,
   scenPrice("habitat",priceScen4)  =  prices("habitat") * (1 - mod(priceScen4.ord,10)/10)    ;
   scenPrice("nutrient",priceScen4) =  prices("nutrient") * (1 - ceil(priceScen4.ord/10)/10) ;
 );

 LOOP(habZeroP2,
   scenPrice("nutrient",habZeroP2)  =  prices("nutrient") * (1 - mod(habZeroP2.ord,10)/10)    ;
   scenPrice("habitat",habZeroP2)  = 0 ;
 );
scenPrice(primary,"zerozero") = 0 ;

display scen1, prices, scenPrice,  savprice, priceScen ;
*$ontext
```

```
loop(scenarios,

*step 3 reestablish data to base level
*    prices(primary) = savprice(primary)                  ;

*step 4 change data to levels needed in scenario
*    prices(primary)$scenprice(primary,scenarios) = scenPrice(primary,scenarios) ;
   prices(primary) = scenPrice(primary,scenarios) ;
   recPrice(primary,scenarios) =  prices(primary) ;
*        display prices;

*step 5 -- solve model
  SOLVE WELFAREMAX using NLP maximising Z ;

*step 6 single scenario report writing
$include LepsCropReport_3_2011.gms

*step 7 cross scenario report writing
*    savsumm("ScenarioSetup","prices",primary,scenarios) = prices(primary)       ;
   savsumm("ScenarioResults",measures,alli,scenarios) = summary(alli,measures) ;

*step 8 end of loop
   );


*step 9 compute and display final results
option savsumm:2:3:1;display savsumm;

parameter savsummp(ordr,*,alli,scenarios) Comparative Farm Summary (percent chg) ;

savsummp(ordr,measures,alli,scenarios)$savsumm(ordr,measures,alli,"base")=
       round(  (savsumm(ordr,measures,alli,scenarios)
        -savsumm(ordr,measures,alli,"base"))*100
        /savsumm(ordr,measures,alli,"base"),1);
savsummp(ordr,measures,alli,scenarios)
        $(savsumm(ordr,measures,alli,"base") eq 0
       and savsumm(ordr,measures,alli,scenarios) ne 0)=na;
savsummp(ordr,"prices",alli,scenarios)$savsumm(ordr,"prices",alli,"base")=
       round(  (savsumm(ordr,"prices",alli,scenarios)
        -savsumm(ordr,"prices",alli,"base"))*100
        /savsumm(ordr,"prices",alli,"base"),1);
savsummp(ordr,"prices",alli,scenarios)
        $(savsumm(ordr,"prices",alli,"base") eq 0
       and savsumm(ordr,"prices",alli,scenarios) ne 0)=na;
option savsummp:1:3:1;display savsummp;
display scenprice, recPrice ;
*$offtext

sets
       NL_P0(priceScen)   set with nutrient load price equal to 0
       BD_P0(priceScen)   sets with nutrient load price equal to 0
       axisitem  / NetIncome, NetHabitat, NetNutLoad, BDprice, NLprice /
```

```
       graph Scenarios     / Results /
*           priceScen(scenarios)  / pScen1*pScen100 /
;

 NL_P0(priceScen)$(scenPrice('nutrient',priceScen) = 0) = yes ;
 BD_P0(priceScen)$(scenPrice('habitat',priceScen) = 0) = yes ;

parameter plot3D(graph, priceScen, priceScen, axisitem)  ;
parameter DataPlot3D(graph, priceScen, priceScen, axisitem)  ;
parameter habitatOnly(graph, priceScen, axisitem)  ;
parameter nutrientOnly(graph, priceScen, axisitem)  ;

habitatOnly(graph, NL_P0, "NetIncome")  = savsumm("ScenarioResults","NetIncome","Total",NL_P0)  ;
habitatOnly(graph, NL_P0, "NetHabitat") = savsumm("ScenarioResults","NetHabitat","Total",NL_P0) ;

nutrientOnly(graph, BD_P0, "NetIncome")  = savsumm("ScenarioResults","NetIncome","Total",BD_P0)  ;
nutrientOnly(graph, BD_P0, "NetNutLoad") = savsumm("ScenarioResults","NetNutLoad","Total",BD_P0) ;

plot3D(graph, priceScen, priceScen, "NetIncome")  =
savsumm("ScenarioResults","NetIncome","Total",priceScen)  ;
plot3D(graph, priceScen, priceScen, "NetHabitat") =
savsumm("ScenarioResults","NetHabitat","Total",priceScen) ;
plot3D(graph, priceScen, priceScen, "NetNutLoad") =
savsumm("ScenarioResults","NetNutLoad","Total",priceScen) ;

DataPlot3D(graph, priceScen, priceScen, "NetIncome")  =
savsumm("ScenarioResults","NetIncome","Total",priceScen)  ;
DataPlot3D(graph, priceScen, priceScen, "NetHabitat") =
savsumm("ScenarioResults","NetHabitat","Total",priceScen) ;
DataPlot3D(graph, priceScen, priceScen, "NetNutLoad") =
savsumm("ScenarioResults","NetNutLoad","Total",priceScen) ;
DataPlot3D(graph, priceScen, priceScen, "BDprice") = scenPrice("habitat",priceScen) ;
DataPlot3D(graph, priceScen, priceScen, "NLprice") = scenPrice("nutrient",priceScen) ;

display NL_P0, BD_P0, plot3D, habitatOnly, nutrientOnly ;
*plot3D,
*plotmsrs(measures) / NetIncome, NetHabitat, NetNutLoad /
*plotmsrs = axis items

*The data to be graphed as 3D-plot must be contained in a four dimensional parameter (a parameter which
has four sets/indexes/dimensions).
*The number of elements in the first argument determines the number of different planes in a plot.
*The number of elements in the second argument determines the number of x-axis values.
*The number of elements in the third argument determines the number of y-axis values.
*The order of the elements in the second and third index determines how the different data points
pertaining to one plane are connected.
*The remaining argument (fourth index) must contain at least three set elements of which one represents
the x-axis, another one the y-axis coordinate, and a third one the z-axis coordinate.

*$setglobal gp_style "lines"
*$libinclude gnuplotxyz plot3D NetHabitat NetNutLoad NetIncome
```

```
$libinclude gnuplotxyz habitatOnly NetIncome NetHabitat
$libinclude gnuplotxyz nutrientOnly NetIncome NetNutLoad

*file wtex /"D:\Documents\Econtools\article\abatementCostTable.tex"/; put wtex;
*$setglobal title "Abatement costs for N and P. Optimisation and OLS results for 16 \\% and 32 \\%
reduction targets "
*$setglobal format tex
*parameter decimals(columns) /a 0,b 4 ,c 1/  ;
*$setglobal row_label rows4
*$setglobal col_label columns4

*THESE PARAMETERS ARE ONLY ONE DIMENSIONAL AND HENCE THE TABLES DO NOT WORK
$setglobal format tex

*file wtex /"D:\Documents\Econtools\article\example.tex"/; put wtex;
*$setglobal texlabel "tab:example"
*$libinclude GAMS2tbl slope_share

*display LepsArableDistr ;

*EXPORT TO MDB = add the script from below to the gdx viewer options
*ODBC DSN-less MS Access Driver
*Driver={Microsoft Access Driver (*.mdb)};Dbq= D:\GIS\SAMA_databaseGIS2003.mdb;
*OLE DB Provider for ODBC Access (Jet)
*Provider =MSDASQL; Driver={Microsoft Access Driver(*.mdb)};Dbq= D:\GIS\SAMA_databaseGIS2003.mdb;
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=CropLoadSW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=FallLoadSW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=CropLoadNL'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=FallLoadNL'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=CropBiodiversityBD'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=FallBiodiversityBD'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=fallBD'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=CropBiodiversitySW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=FallBiodiversitySW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=CropLoadSW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=FallLoadSW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=BiodiversitySW'
*execute 'gdxviewer.exe i=D:\gamsdir\biodiversity\BdNlOpimizationResults.gdx sql id=LoadSW'
*$offtext
```

Loop parameter file

```
CropLoadSW(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )          = [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]   ;
FallLoadSW(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )        = [(FDRP.L(flw,slt,Pst,slp,asp,dst)
+ FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ]   ;
CropBiodiversitySW(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )    = cropBD(j,dst,asp,slp)    ;
FallBiodiversitySW(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )  = fallBD(flw,dst,asp,slp)   ;

YieldValueSW(j,slt,Pst,slp,asp,dst)      =  c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) * X.L(j,slt,Pst,slp,asp,dst)  ;
CropBdValueSW(j,slt,Pst,slp,asp,dst)     =   prices("habitat") * cropBD(j,dst,asp,slp)   * X.L(j,slt,Pst,slp,asp,dst)
;
FallBdValueSW(flw,slt,Pst,slp,asp,dst)   =   prices("habitat") * fallBD(flw,dst,asp,slp) *
Xfa.L(flw,slt,Pst,slp,asp,dst)     ;
CropLoadValueSW(j,slt,Pst,slp,asp,dst)   =   prices("nutrient") * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)   ;
FallLoadValueSW(flw,slt,Pst,slp,asp,dst) =   prices("nutrient") * [(FDRP.L(flw,slt,Pst,slp,asp,dst) +
FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst)  ;

TotalYieldValueSW  =   SUM{(j,slt,Pst,slp,asp,dst), c_price(j) *  CY.L(j,slt,Pst,slp,asp,dst) *
X.L(j,slt,Pst,slp,asp,dst) } ;
TotalBdValueSW     =   SUM{(j,slt,Pst,slp,asp,dst), bd_price * cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) }
+  SUM{(flw,slt,Pst,slp,asp,dst), bd_price * fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalLoadValueSW   =   SUM{(j,slt,Pst,slp,asp,dst), wr_price * [(DRP.L(j,slt,Pst,slp,asp,dst) +
PP.L(j,slt,Pst,slp,asp,dst)) * redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +
SUM{(flw,slt,Pst,slp,asp,dst), wr_price * [(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))*
redfield + FAN.L(flw,slt,slp) ] * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

TotalYieldValuePerHaSW     =  TotalYieldValueSW    /model_ha ;
TotalBdValuePerHaSW        =  TotalBdValueSW       /model_ha ;
TotalLoadValuePerHaSW      =  TotalLoadValueSW     /model_ha ;

TotalLoadSW     = SUM{(j,slt,Pst,slp,asp,dst), [(DRP.L(j,slt,Pst,slp,asp,dst) + PP.L(j,slt,Pst,slp,asp,dst)) *
redfield + NC.L(j,slt,slp) ]* X.L(j,slt,Pst,slp,asp,dst)  } +  SUM{(flw,slt,Pst,slp,asp,dst),
[(FDRP.L(flw,slt,Pst,slp,asp,dst) + FPP.L(flw,slt,Pst,slp,asp,dst))* redfield + FAN.L(flw,slt,slp) ] *
Xfa.L(flw,slt,Pst,slp,asp,dst) } ;
TotalBiodiversitySW  =  SUM{(j,slt,Pst,slp,asp,dst),  cropBD(j,dst,asp,slp)  * X.L(j,slt,Pst,slp,asp,dst) } +
SUM{(flw,slt,Pst,slp,asp,dst),  fallBD(flw,dst,asp,slp) * Xfa.L(flw,slt,Pst,slp,asp,dst) } ;

cropShareSW(j,slt,Pst,slp,asp,dst)$(X.L(j,slt,Pst,slp,asp,dst) > 0 )        = X.L(j,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;
fallShareSW(flw,slt,Pst,slp,asp,dst)$(Xfa.L(flw,slt,Pst,slp,asp,dst) > 0 )  = Xfa.L(flw,slt,Pst,slp,asp,dst)/
land_distr(slt,Pst,slp,asp,dst) ;

TotFallShareSW  =  SUM{(flw,slt,Pst,slp,asp,dst),Xfa.L(flw,slt,Pst,slp,asp,dst)} /
SUM{(slt,Pst,slp,asp,dst),land_distr(slt,Pst,slp,asp,dst)} ;
TotBdvFallShareSW  = SUM{(bdfall,slt,Pst,slp,asp,dst),Xfa.L(bdfall,slt,Pst,slp,asp,dst)} /
SUM{(slt,Pst,slp,asp,dst),land_distr(slt,Pst,slp,asp,dst)} ;
```

```
BiodiversitySW(slt,Pst,slp,asp,dst)   =  SUM{j,cropShareSW(j,slt,Pst,slp,asp,dst) *
CropBiodiversitySW(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareSW(flw,slt,Pst,slp,asp,dst) *
FallBiodiversitySW(flw,slt,Pst,slp,asp,dst)}   ;
LoadSW(slt,Pst,slp,asp,dst)          =  SUM{j,cropShareSW(j,slt,Pst,slp,asp,dst) *
CropLoadSW(j,slt,Pst,slp,asp,dst)} + SUM{flw,fallShareSW(flw,slt,Pst,slp,asp,dst) *
FallLoadSW(flw,slt,Pst,slp,asp,dst)}   ;

*display model_ha, model_size,  catchmentFieldShare, baseTPload, baseNload, TotalMinLoad, TotalMaxBD
;
*display TotalYieldValuePP, TotalYieldValueBD,TotalYieldValueNL, TotalYieldValueSW ;
*display TotalBiodiversityPP, TotalBiodiversityBD, TotalBiodiversityNL, TotalBiodiversitySW ;
*display TotalLoadPP, TotalLoadBD, TotalLoadNL, TotalLoadSW ;
*display CropBiodiversityBD, FallBiodiversityBD ;
*display FallBiodiversitySW, FallBiodiversityPP, FallBiodiversityBD, FallBiodiversityNL ;
*display BiodiversitySW, LoadSW ;
summary("Total","netIncome")  =  TotalYieldValueSW      ;
summary("Total","NetHabitat") =  TotalBiodiversitySW     ;
summary("Total","NetNutLoad") =  TotalLoadSW           ;
summary("Total","TotFallowShare")  =  TotFallShareSW     ;
summary("Total","TotBdvFallShare") =  TotBdvFallShareSW  ;
```

MTT is publishing its research findings in two series of publications:
MTT Science and MTT Growth.

MTT CREATES VITALITY THROUGH SCIENCE

MTT SCIENCE

www.mtt.fi/julkaisut

The MTT Science series includes scientific presentations and abstracts from conferences arranged by MTT Agrifood Research Finland.
Doctoral dissertations by MTT research scientists will also be published in this series.
The topics range from agricultural and food research to environmental research in the field of agriculture.

MTT, FI-31600 Jokioinen, Finland.
Tel. +358 29 5300 700, email julkaisut@mtt.fi