Ins
Hy

1991/043

ARCHIVE:

PLEASE DO NOT DESTROY

Natural Environment Research Council
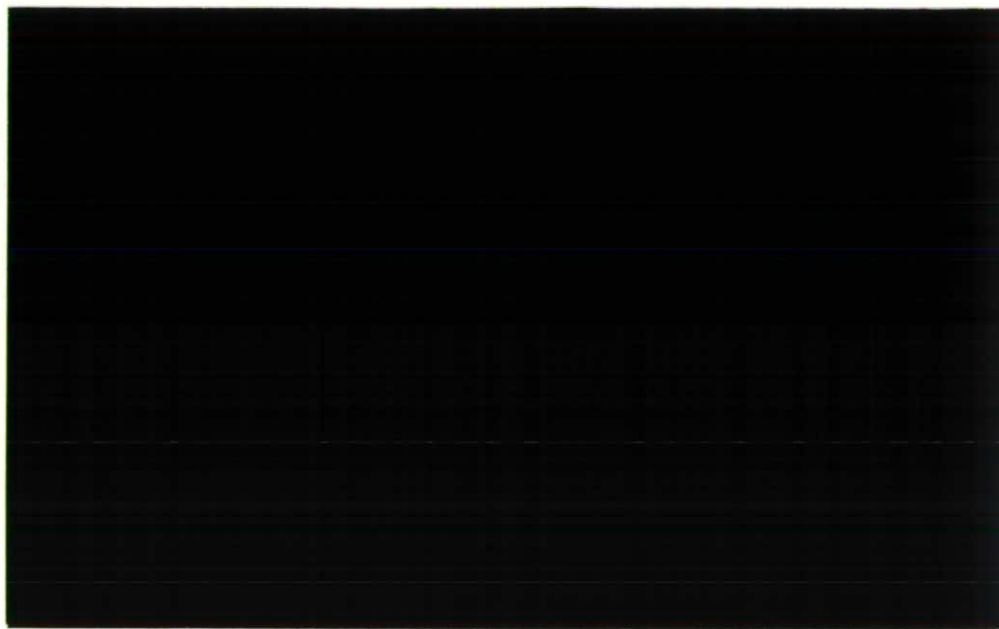
July 1, 1991

**The West Sedgemoor
hydrological study 1986-1991**

**Report prepared for
Nature Conservancy Council**

K Gilman [1]
D C W Marshall [2]

Appendices A1 to A5

[1] Institute of Hydrology
Staylittle
Llanbrynmair
Powys
SY19 7DB

[2] Institute of Hydrology
Maclean Building
Crowmarsh Gifford
Wallingford
Oxfordshire
OX10 8BB

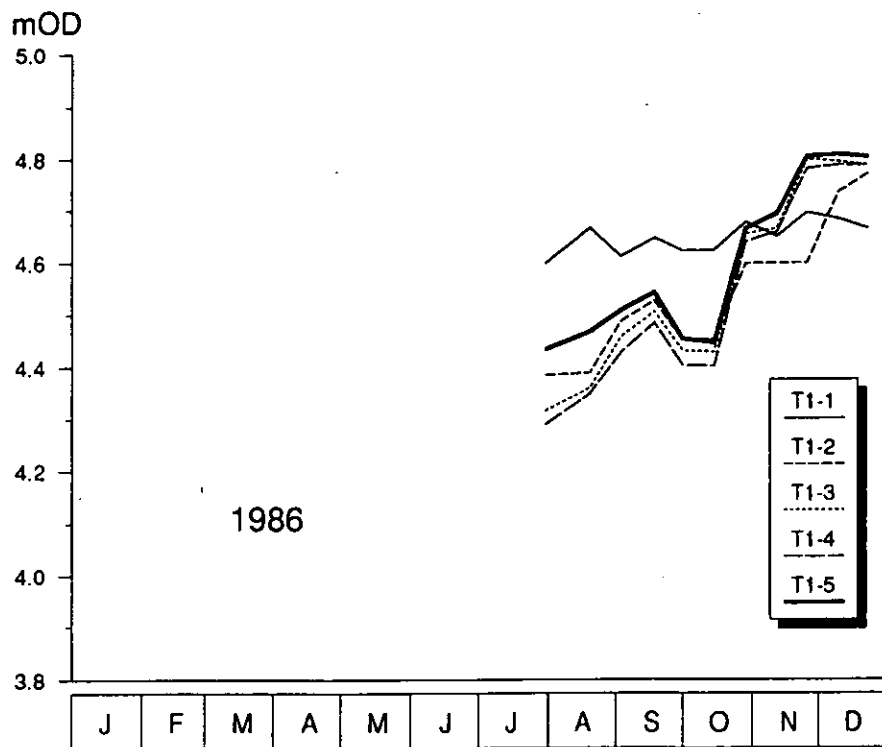# Appendix 1 GROUNDWATER LEVEL DATA FROM DIPWELL TRANSECTS

Figure A1.1 Dipwell levels in transect **T1**, 1986
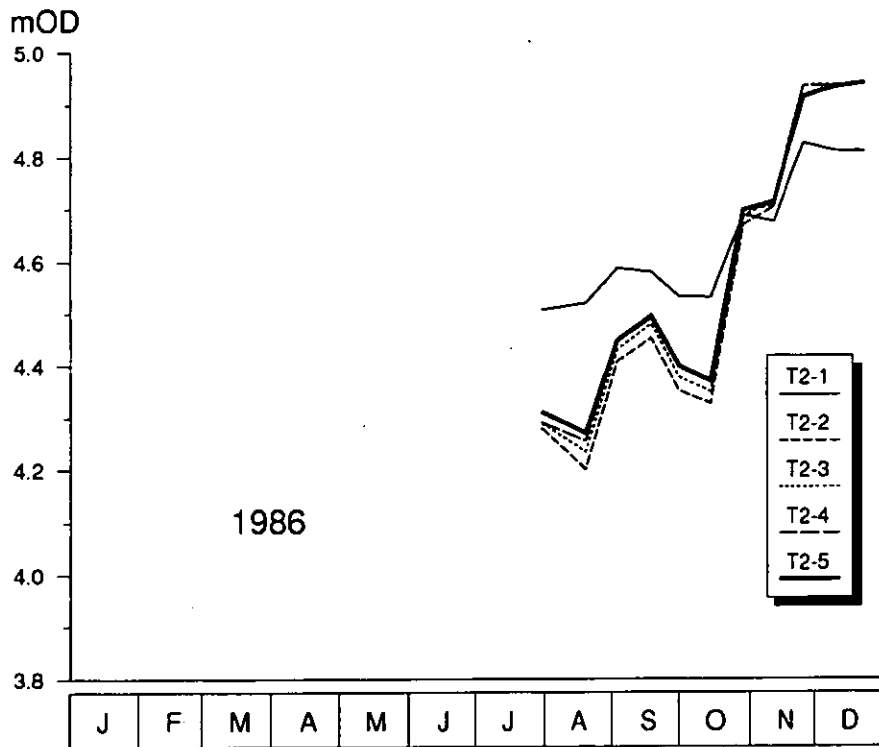
Figure A1-2 Dipwell levels in transect **T2**, 1986

Figure A1-3 Dipwell levels in transect **T3**, 1986

mOD

Figure A1-4 Dipwell levels in transect **T4**, 1986

mOD

Figure A1-5 Dipwell levels in transect **T1**, 1987

Figure A1-6 Dipwell levels in transect **T2**, 1987

Figure A1-7 Dipwell levels in transect **T3**, 1987

Figure A1-8 Dipwell levels in transect **T4**, 1987

Figure A1-9 Dipwell levels in transect **T1**, 1988

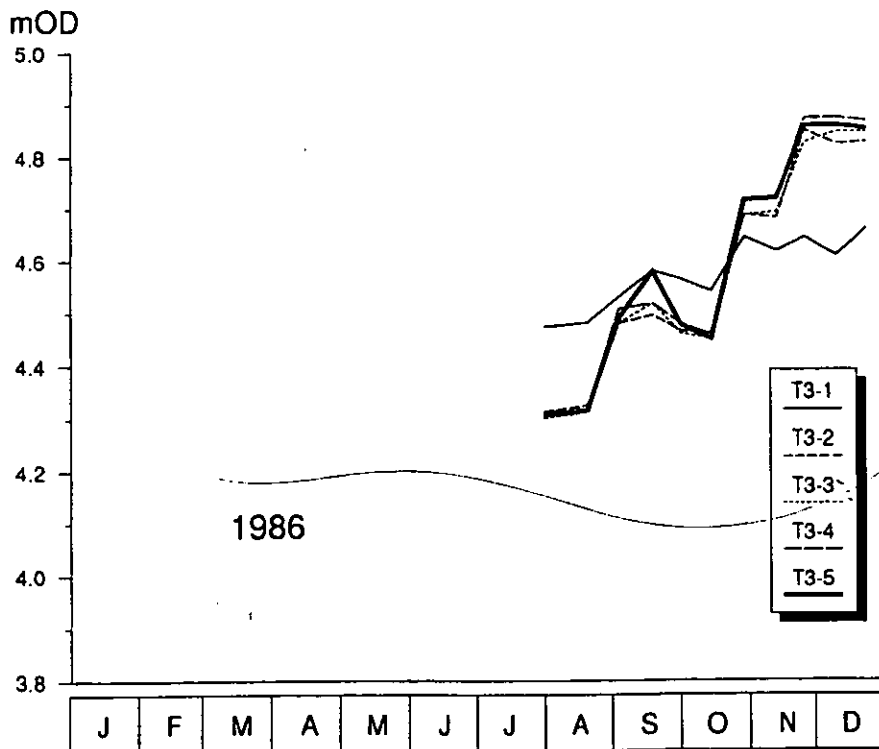Figure A1-10 Dipwell levels in transect **T2**, 1988
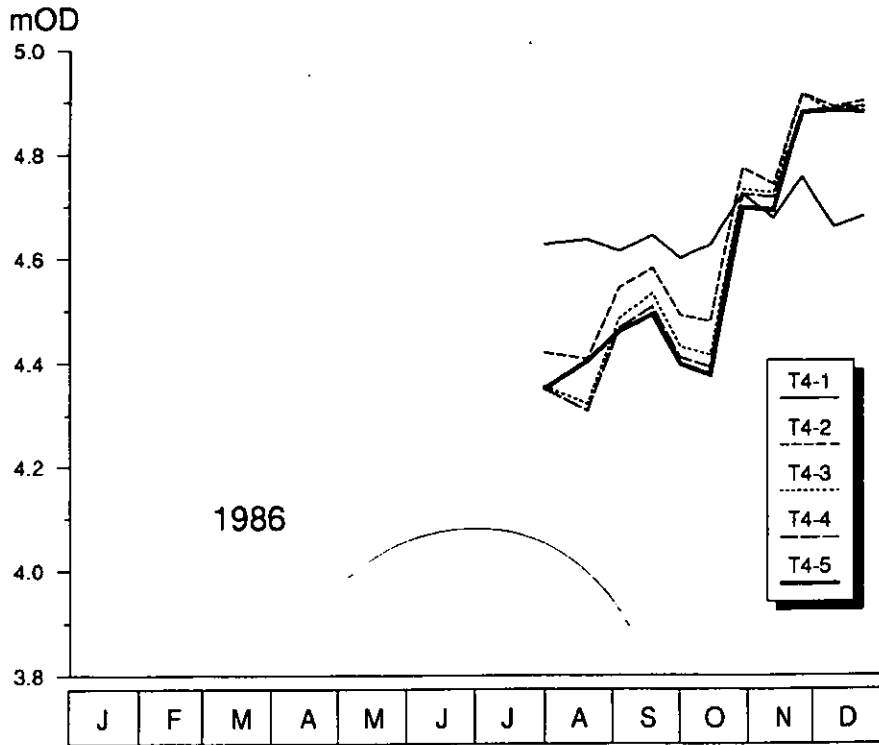

Figure A1-11 Dipwell levels in transect **T3**, 1988

Figure A1-12 Dipwell levels in transect **T4**, 1988

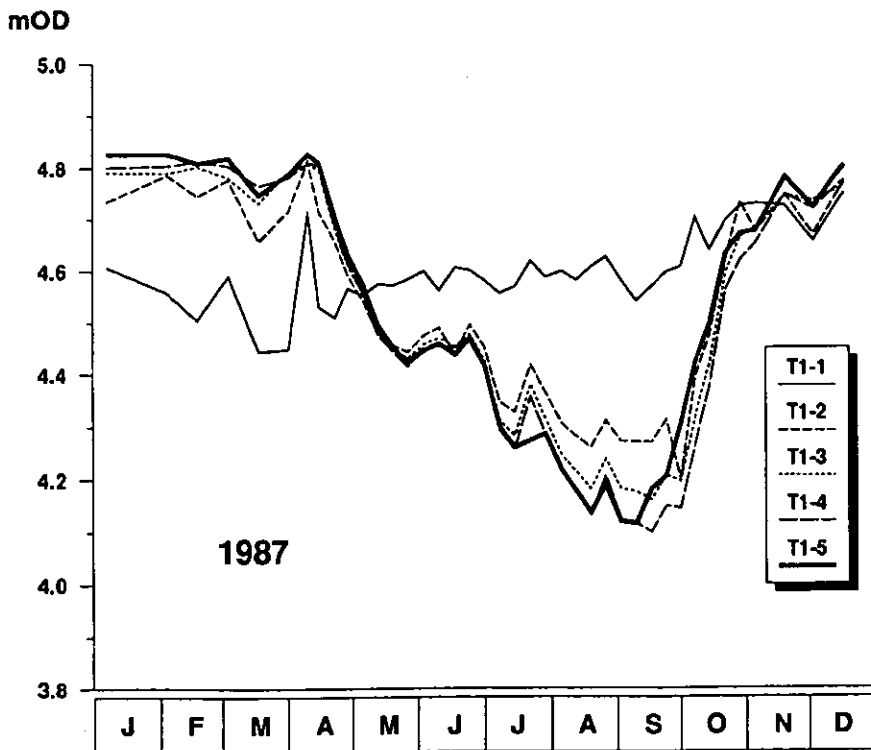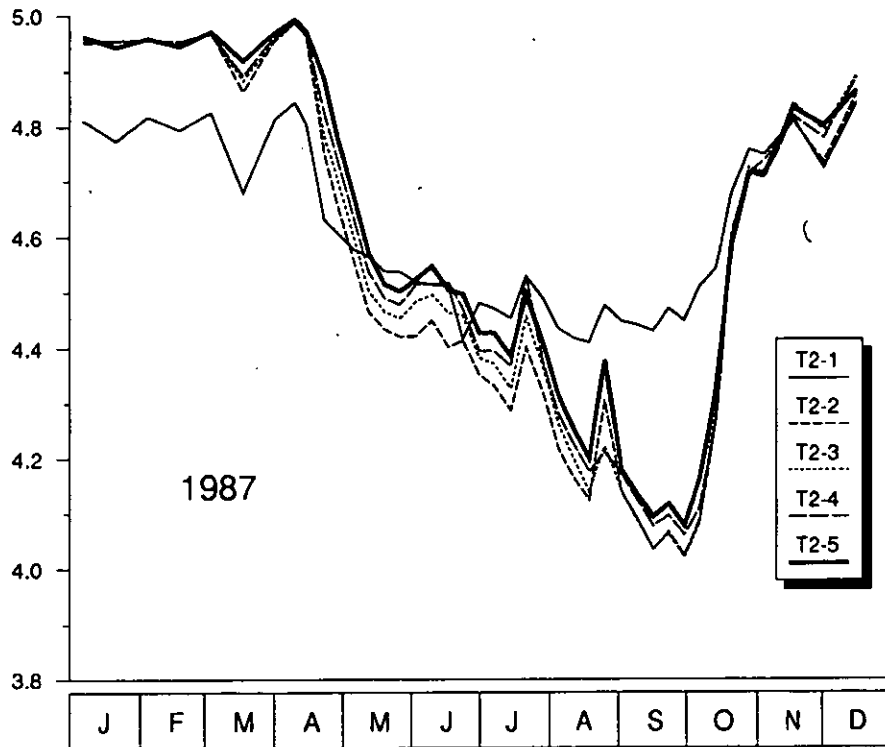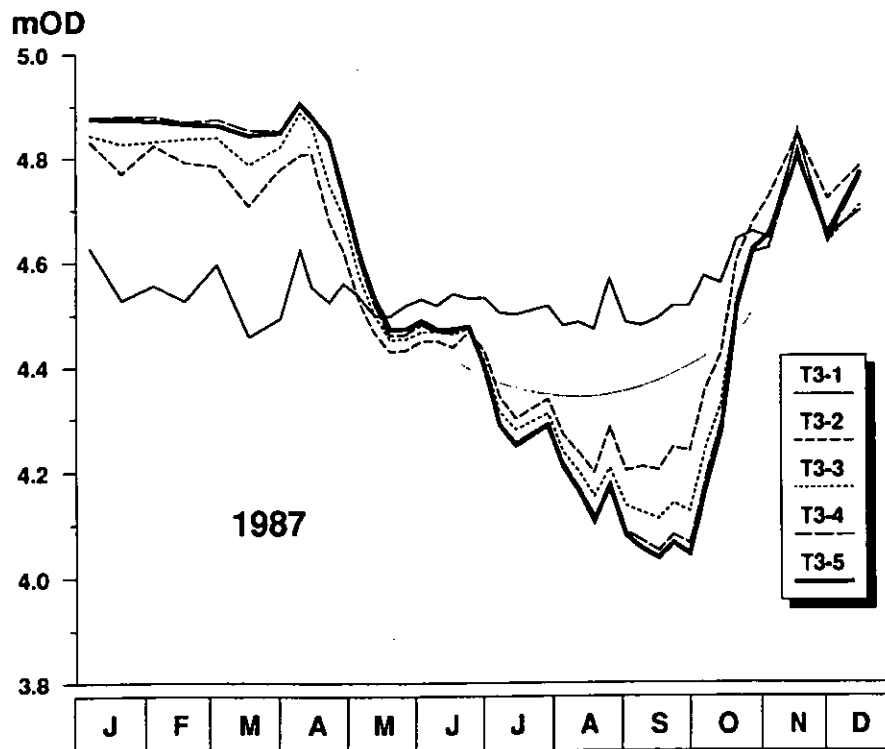Figure A1-13 Dipwell levels in transect **T1**, 1989
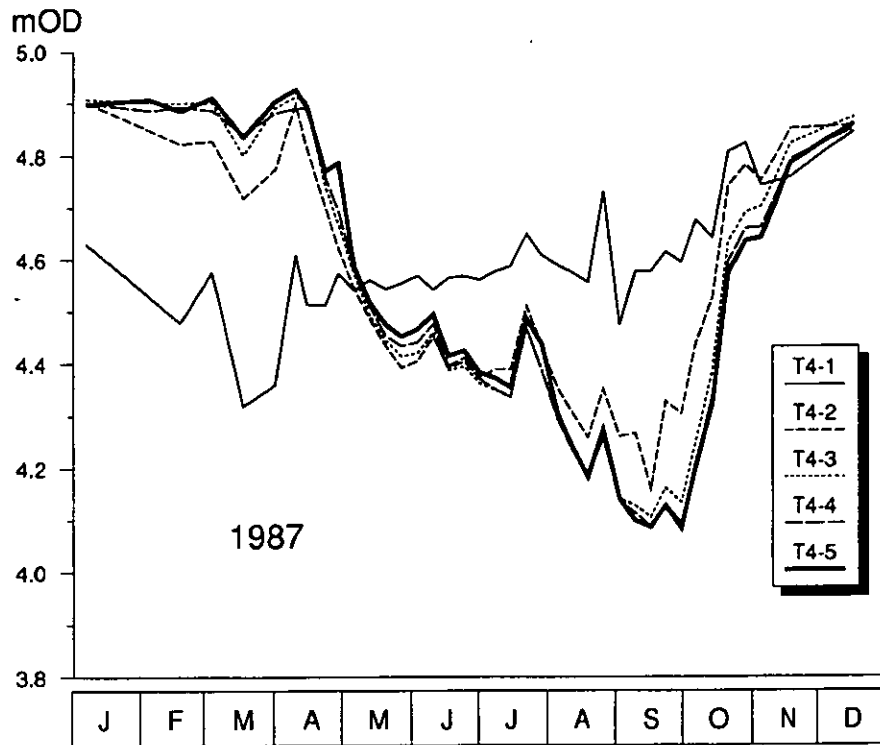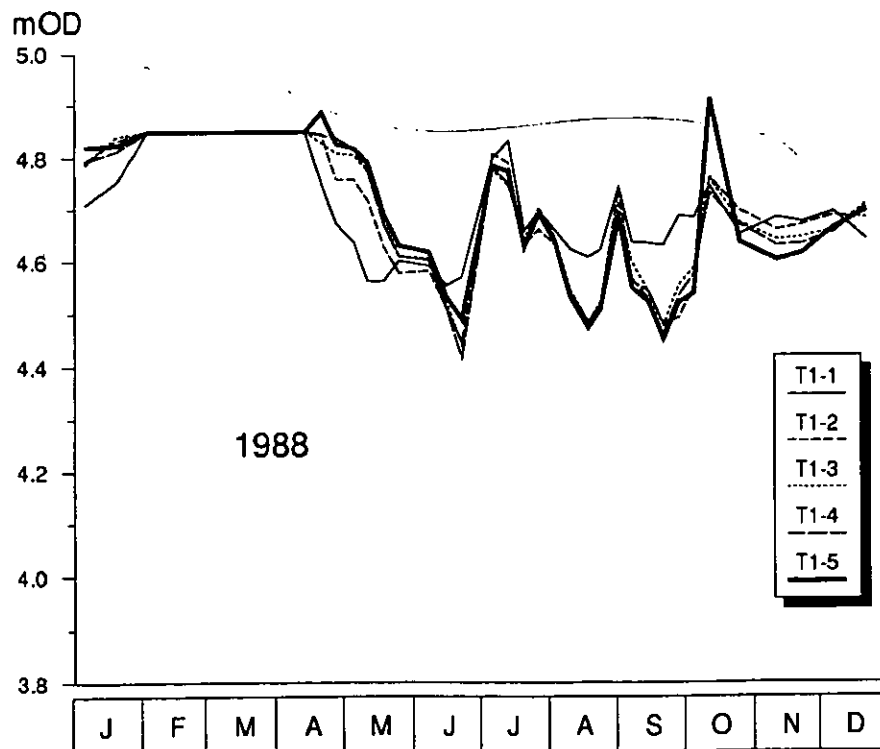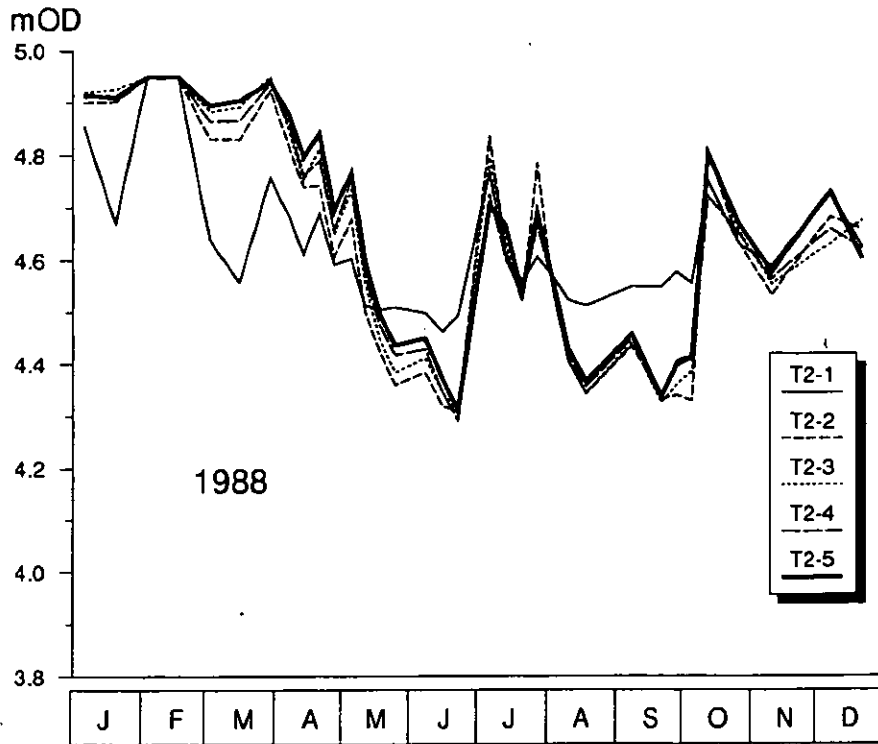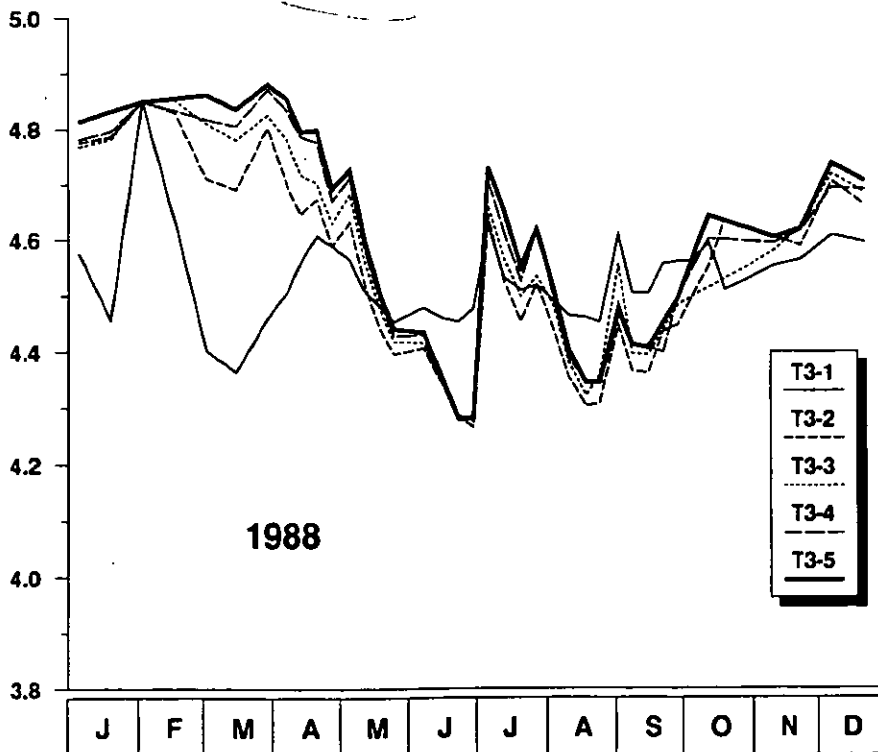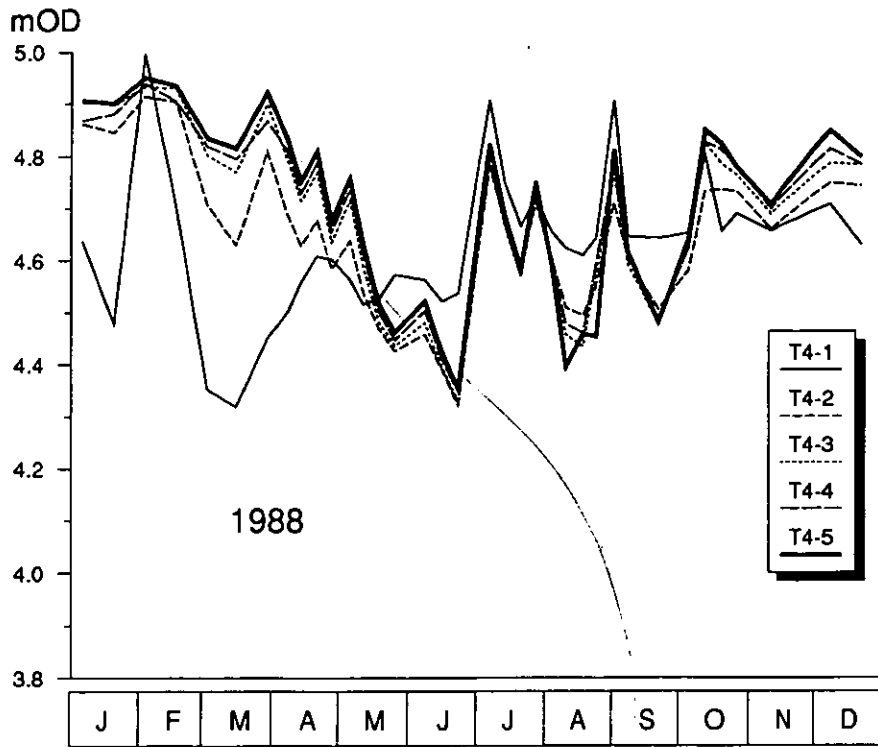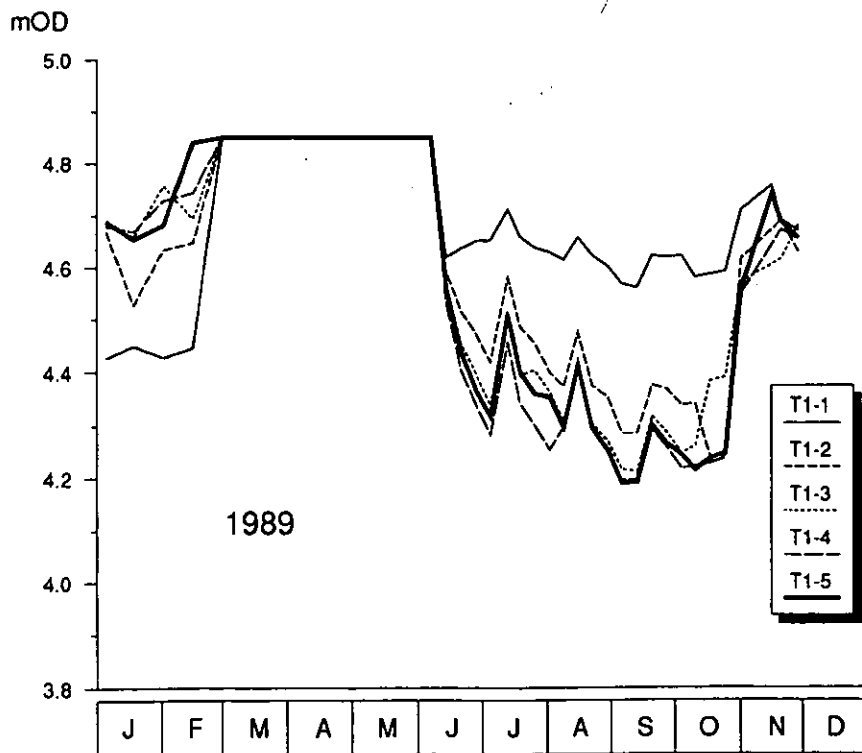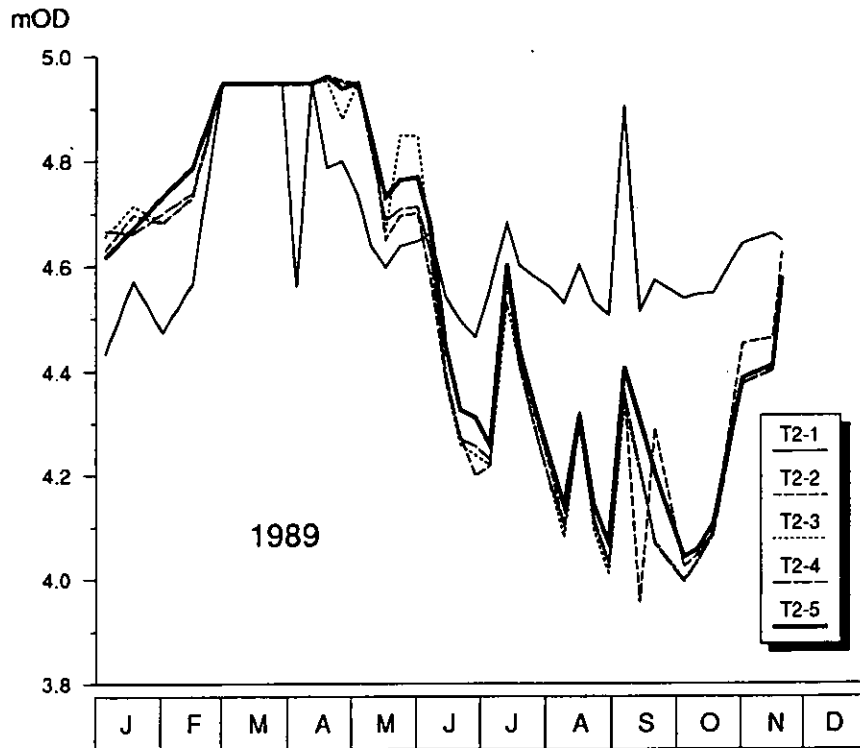
A1.7

Figure A1-14 Dipwell levels in transect **T2**, 1989
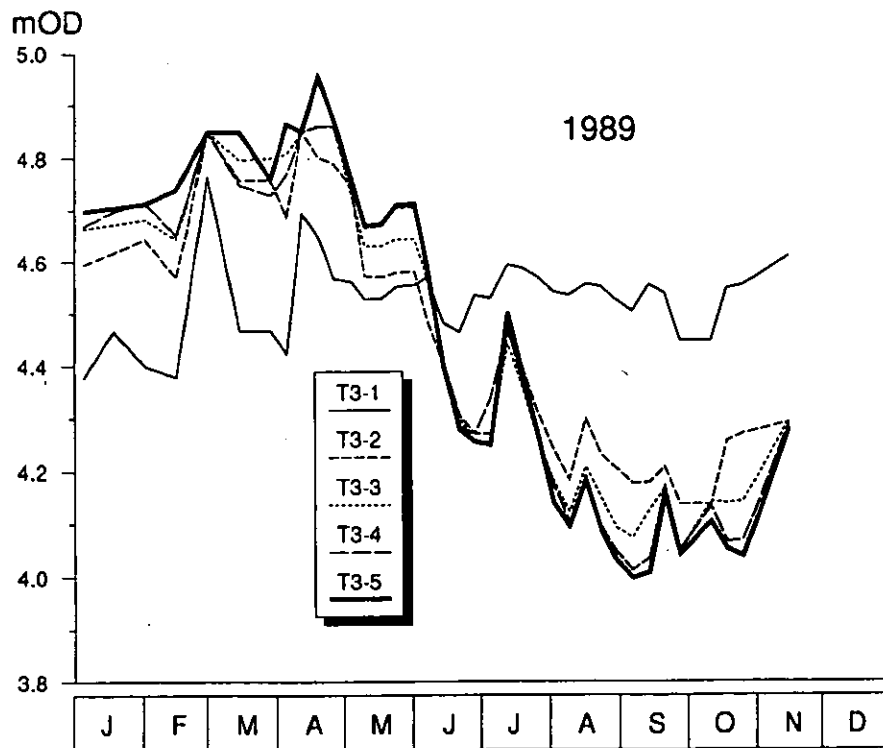
Figure A1-15 Dipwell levels in transect **T3**, 1989

A1.8
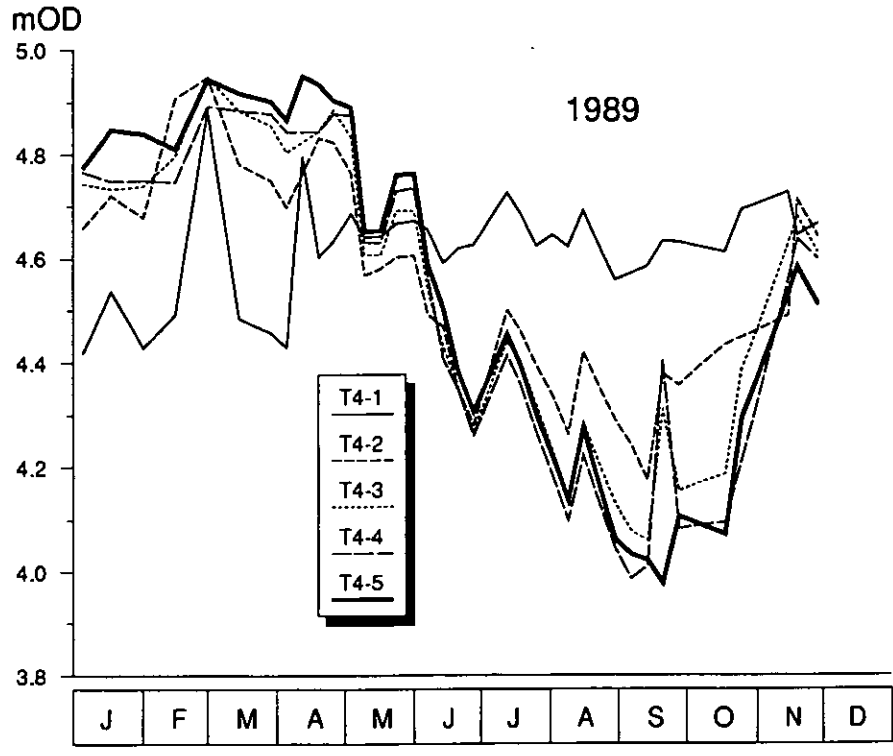
Figure A1-16 Dipwell levels in transect **T4**, 1989

Appendix 2 ONE-DIMENSIONAL MIRE TRANSECT MODEL

The model sets out to simulate the behaviour of groundwater under water table conditions in a strip of land between two parallel open drains or rhynes. The upper horizons, down to a base level of z_b , are assumed to be permeable. Below that level the soil is assumed to be effectively impermeable. Infiltration of water occurs vertically from the ground surface through the unsaturated zone to the water table, and flow in the saturated zone is essentially horizontal towards or away from the rhynes.

Although the concepts on which the groundwater model is based are simple, the complexity of the model increases rapidly as more features are introduced. For instance, it is likely that, because of compaction and humification processes, the hydraulic properties of peat vary according to its depth below the surface. Furthermore, special modifications have to be made to the model equations to account for behaviour at boundaries, for instance the rhyne or the ground surface. As an introduction to the model, a simple case, with uniform hydraulic properties, is tackled first.

2.1 Simple model with uniform permeability

The groundwater flow equation for horizontal flow (the Dupuis-Forscheimer approximation) predicts the water table elevation (hydraulic head) h as a function of time and horizontal distance, for an aquifer with transmissivity T and specific yield S , subject to an infiltration rate q per unit area:

$$-\frac{\partial}{\partial x}\left(T\frac{\partial h}{\partial x}\right) = -S\frac{\partial h}{\partial t} + q$$

which simplifies to

$$-\frac{\partial T}{\partial x}\frac{\partial h}{\partial x} - T\frac{\partial^2 h}{\partial x^2} = -S\frac{\partial h}{\partial t} + q$$

The aquifer is assumed to bounded below by an impermeable horizon at elevation z_b mOD (Figure A2.1). Transmissivity is the integral of the permeability with respect to depth over the saturated thickness of the aquifer: in the case of a vertically uniform permeability in the saturated part of the upper layer of the soil, the transmissivity is the product of the permeability κ and the saturated depth of the aquifer, which is $(h - z_b)$.
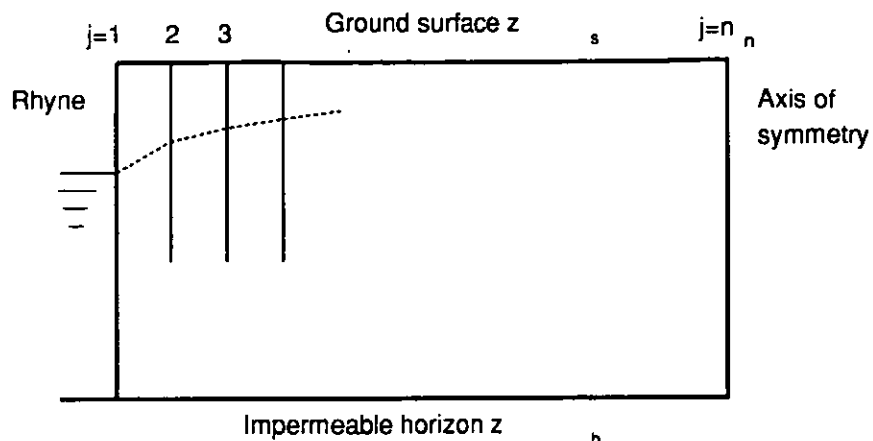
Figure A2.1 Definition sketch for one-dimensional model

The partial differential equation becomes

$$-\kappa \frac{\partial(h-z_b)}{\partial x}\frac{\partial h}{\partial x} - \kappa(h-z_b)\frac{\partial^2 h}{\partial x^2} = -S\frac{\partial h}{\partial t} + q$$

Analytical solutions of this equation exist only for very special conditions, and for most purposes it is necessary to solve it numerically. Finite difference methods, in which the continuous space and time domains are approximated by a network of (usually equally spaced) nodes, are the most commonly used techniques for solution. The equation may be discretised, using the central difference approximations

$$\frac{\partial h}{\partial x}\bigg|_{x=x_j} \sim \frac{(h_{j+1}-h_{j-1})}{(2\Delta x)}$$

and

$$\frac{\partial^2 h}{\partial x^2}\bigg|_{x=x_j} \sim \frac{(h_{j+1}-2h_j+h_{j-1})}{(\Delta x^2)}$$

where Δx is the interval between two nodes of the grid, to give a system of equations relating the head at node j with its neighbours $j-1$ and $j+1$.

$$-\kappa\frac{(h_{j+1}-h_{j-1})^2}{4\Delta x^2} - \kappa(h_j-z_b)\frac{(h_{j+1}-2h_j+h_{j-1})}{\Delta x^2} = -S\frac{\partial h_j}{\partial t} + q$$

Because of the distribution of dipwells in the observation well transects at West Sedgemoor, it was decided to set Δx equal to 2 metres.

The non-linear system of equations for h_j obtained above may be linearised by accepting the head at the preceding timestep h_j^0 as an approximation for h_j in the non-linear terms and approximating the time derivative as

A2.2

$$\left.\frac{\partial h}{\partial t}\right|_{x=x_j} \sim \frac{(h_j - h_j^0)}{\Delta t}$$

Meteorological data are generally available on a daily time scale, whereas groundwater level observations were collected on a longer time interval with some gaps in the record. For this reason it was considered appropriate to adopt 1 day as the time interval Δt of the model.

Powerful methods exist for the solution of systems of linear equations which can be expressed in tridiagonal matrix form (Figure A2.2). The solution, a vector whose elements are the h_j at the end of a finite timestep, is obtained by inverting a matrix of constant coefficients. The simpler the form of this coefficient matrix, i.e. the fewer the off-diagonal elements, the more quickly and efficiently the solution can be reached.
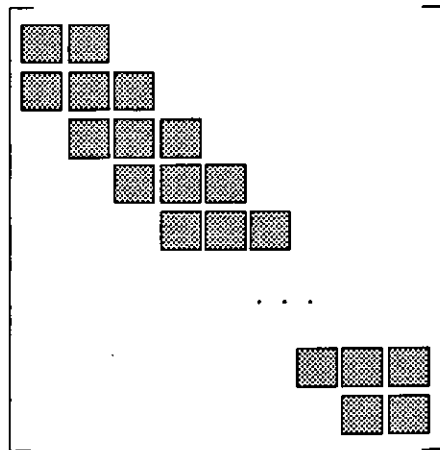
Figure A2.2 The distribution of non-zero elements in the coefficient matrix for a tridiagonal system of equations

Terms are rearranged to yield the system of equations

$$-(h_{j-1}^0 + 4(h_j^0 - z_b) - h_{j+1}^0)h_{j-1} + 8(h_j^0 - z_b)h_j - (-h_{j-1}^0 + 4(h_j^0 - z_b) + h_{j+1}^0)h_{j+1}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(-S\frac{(h_j - h_j^0)}{\Delta t} + q\right)$$

which requires one further manipulation to put it into the tridiagonal form

$$-(h_{j-1}^0 + 4(h_j^0 - z_b) - h_{j+1}^0)h_{j-1}$$

$$+\left(8(h_j^0 - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_j$$

$$-(-h_{j-1}^0 + 4(h_j^0 - z_b) + h_{j+1}^0)h_{j+1}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_j^0}{\Delta t} + q\right)$$

This system of equations may be written:

$$A_j h_{j-1} + B_j h_j + C_j h_{j+1} = D_j$$

where the coefficients A_j etc. are given by

$$A_j = -(h_{j-1}^0 + 4(h_j^0 - z_b) - h_{j+1}^0)$$

$$B_j = 8(h_j^0 - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t})$$

$$C_j = -(-h_{j-1}^0 + 4(h_j^0 - z_b) + h_{j+1}^0)$$

$$D_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_j^0}{\Delta t} + q\right)$$

This system of simultaneous linear algebraic equations may be solved by Thomas' algorithm, which is a form of Gauss elimination, and provides a convenient and robust method for evaluating the h_j from the coefficients A_j etc.

2.1.1 Boundaries

At the rhyne boundary $(j = 1)$ the head h is assumed equal to the rhyne water level

$$h = h_{rhyne}(t)$$

This can be ensured by setting

$$A_1 = 0 \; ; \; B_1 = 1 \; ; \; C_1 = 0 \; ; \; D_1 = h_{rhyne}$$

At the "field" end of the transect (on the centre line of the field midway between the two rhynes), the hydraulic gradient is zero

$$\frac{\partial h}{\partial x}\bigg|_{field} = 0$$

A2.4

This implies that

$$-\kappa(h - z_b)\frac{\partial^2 h}{\partial x^2} = -S\frac{\partial h}{\partial t} + q$$

which discretises to

$$-\frac{\kappa}{\Delta x^2}(h_j^0 - z_b)(h_{j-1} - 2h_j + h_{j+1}) = -S\frac{(h_j - h_j^0)}{\Delta t} + q$$

The zero hydraulic gradient also implies that

$$h_{j+1} = h_{j-1}$$

on the centre line of the field, and the equation for the centre line node can be simplified and linearised to give

$$-8(h_j^0 - z_b)h_{j-1} + 8(h_j^0 - z_b)h_j = 4\frac{\Delta x^2}{\kappa}\left(-S\frac{(h_j - h_j^0)}{\Delta t} + q\right)$$

which in tridiagonal form is

$$-8(h_j^0 - z_b)h_{j-1} + \left(8(h_j^0 - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_j^0}{\Delta t} + q\right)$$

$$A_j = -8(h_j^0 - z_b)$$

$$C_j = 0$$

B_j and D_j are as in the general case.

2.1.2 Ground surface

The possibility of flooding of the ground surface by a very high water table, for instance in winter, is not introduced at this stage.

2.2 Permeability varying with depth

The hypothesis of uniform permeability, though it results in a relatively simple system of equations, is perhaps unjustified. Because of compaction of humified peat at depth, it is likely that the permeability of the saturated soil decreases with depth. In this case the transmissivity is equal to

$$T = \int_{base}^{water\ table} \kappa(h)dh$$

If the permeability is assumed to decrease exponentially with depth

$$\kappa(h) = \kappa_0 e^{-\alpha(z_s - h)}$$

then

$$T(h) = \frac{\kappa_0}{\alpha}\left(e^{-\alpha(z_s - h)} - e^{-\alpha(z_s - z_b)}\right)$$

In the limit as α tends to zero, i.e. a uniform permeability, this reassuringly tends to

$$T = \kappa_0(h - z_b)$$

The expression for $T(h)$ can be generalised to

$$T(h) = f(h) - f(z_b)$$

The adoption of this generalised form improves the appearance of the equations and simplifies the computer program. It also includes as a special case the simple formula

$$T(h) = \kappa(h - z_b)$$

which applies in the case of a vertically uniform permeability.

The partial differential equation

$$-\frac{\partial T}{\partial x}\frac{\partial h}{\partial x} - T\frac{\partial^2 h}{\partial x^2} = -S\frac{\partial h}{\partial t} + q$$

may be discretised as follows:

$$-(T_{j+1} - T_{j-1})(h_{j+1} - h_{j-1}) - 4T_j(h_{j+1} - 2h_j + h_{j-1})$$

$$= 4\Delta x^2\left(-S\frac{(h_j - h_j^0)}{\Delta t} + q\right)$$

The elements of the coefficient matrix of the tridiagonal system are

$$A_j = -(f(h_{j-1}^0) + 4(f(h_j^0) - f(z_b)) - f(h_{j+1}^0))$$

$$B_j = 8(f(h_j^0) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-f(h_{j-1}^0) + 4(f(h_j^0) - f(z_b)) + f(h_{j+1}^0))$$

$$D_j = 4\Delta x^2\left(S\frac{h_j^0}{\Delta t} + q\right)$$

2.2.1 Boundaries

At the rhyne boundary $(j = 1)$ the head h is assumed equal to the rhyne water level

$$h = h_{rhyne}(t)$$

This can be ensured by setting

$$A_1 = 0 \; ; \; B_1 = 1 \; ; \; C_1 = 0 \; ; \; D_1 = h_{rhyne}$$

At the "field" end of the transect (on the centre line of the field midway between the two rhynes), coefficients are

$$A_j = -8(f(h_j^0) - f(z_b))$$

$$C_j = 0$$

B_j and D_j are as in the general case.

2.2.2 Ground surface

When the water table reaches the ground surface z_s, the specific yield becomes 100%, and the flow pattern is dominated by a thin layer of almost infinite permeability, giving rise to a water table at or above the surface over much of the field. This event, a regular winter occurrence over the Somerset Levels, gives a starting point for the next season's water table decline, so it is important that it be included in the model, though the precise flooding level depends on many other factors, and cannot be simulated accurately. The model embraces surface flooding by modifications to the A etc.

Because the ground surface is not perfectly flat, but is complicated by microrelief, the vegetation mat and the grass sward, the changes in permeability and specific yield are not immediate. It is more realistic to assume a specific yield which increases linearly from its value at the generalised ground surface towards 100% for some given depth of standing water, and a "permeability" for overland flow which increases exponentially with the depth of standing water. The integral of this "permeability" over depth gives an analogue of transmissivity, which in turn determines the flow in response to a given hydraulic gradient. The model remains stable, and plausible results are obtained, if the specific yield reaches 100% for a depth of 100 mm of water on the surface, and the "permeability" is given by

$$\kappa_0 e^{-\gamma(h - z_s)}$$

where κ_0 is the permeability of the saturated soil and γ is a positive constant. A value of $\gamma = 25$ provides a "permeability" which increases by a factor of 12 as the depth of standing water increases to 100 mm.

The above-ground values of permeability are incorporated into the model by replacing the function f with a new function g, which is given by

$$g(h) = \frac{\kappa_0}{\gamma} e^{\gamma(h - z_s)} - \frac{\kappa_0}{\gamma} + f(z_s)$$

for all occasions when $h > z_s$. This expression reduces to $f(z_s)$ for $h = z_s$.

2.3 Specific yield varying with depth

The specific yield of peat, like its permeability, depends critically on the degree of compaction. More compact peat at depth would yield less water on drainage, and the specific yield is a function of the water table elevation. If S is assumed to have an exponential variation with depth

$$S = S_0 e^{-\beta(z_s - h)}$$

As with changes in specific yield due to flooding, the effect of varying specific yield can be taken into account easily by computing S at each time step, for use in the evaluation of B_j and D_j.

2.4 Horizontal variation of permeability

Results from the West Sedgemoor dipwell transects suggest that the hydraulic connection between fields and rhynes is not as good as might be expected. The seasonal pattern of water levels typical of the rhyne, though it is observed in dipwells 2 m from the rhyne, is quickly overwhelmed by the pattern typical of the centre of the field, which is dominated by evaporation and rainfall. Three factors may be responsible:

(i) the rhynes are of modest depth, compared with the depth of the peat.

(ii) there may be sealing of the rhyne beds and banks by fine material, algal growth etc, restricting the flow of water into and out of the channels.

(iii) peat adjacent to the rhyne has been compacted by long-term vehicle traffic and spoil dumping. This effect would be expected to be confined to the four or five metres closest to the rhyne.

In the interests of simplicity, it was decided to simulate the poor hydraulic connection between field and rhyne by defining a lower permeability in a panel 4 m wide. The permeability here is assumed equal to r times that in the field: obviously setting r equal to unity is equivalent to imposing a horizontally uniform permeability (Figure 3).
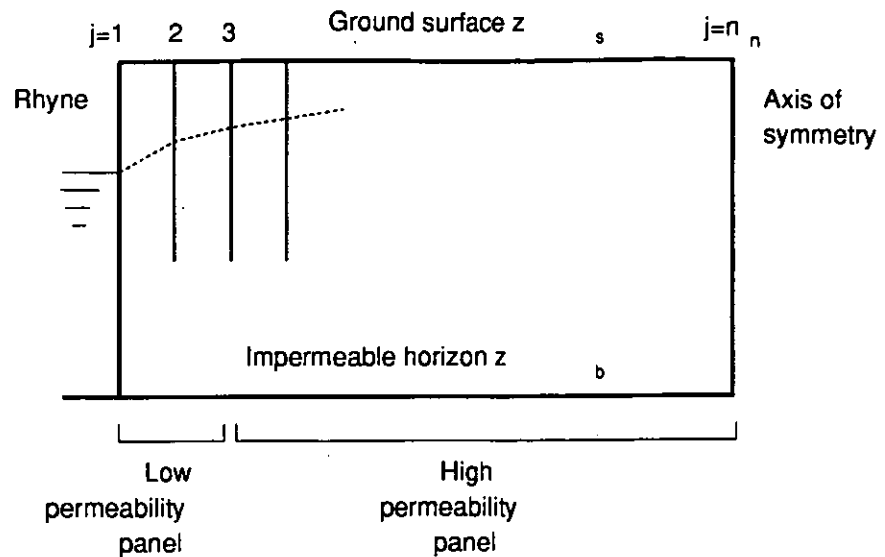
Figure A2.3 Definition sketch for one-dimensional model with low permeability panel

At the rhyne boundary, i.e. at $j = 1$, coefficients are unchanged from the general case:

$$A_1 = 0 \; ; \; B_1 = 1 \; ; \; C_1 = 0 \; ; \; D_1 = h_{rhyne}$$

In the low permeability panel, i.e. at $j = 2$, the substitution of

$$T_j = r(f(h_j) - f(z_b))$$

implies the following changes to the coefficients B_j and D_j .

$$B_j = 8(f(h_j^0) - f(z_b)) + 4 \frac{\Delta x^2 S}{r \Delta t}$$

$$D_j = 4 \frac{\Delta x^2}{r} \left(S \frac{h_j^0}{\Delta t} + q \right)$$

On the boundary between the two panels, i.e. at $j = 3$, modifications are made to A_j, B_j and C_j as follows:

$$A_j = -(r(f(h_{j-1}^0) - f(z_b))$$

$$+2(1+r)(f(h_j^0) - f(z_b))$$

$$-(f(h_{j+1}^0) - f(z_b)))$$

$$B_j = 4(1+r)(f(h_j^0) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-r(f(h_{j-1}^0) - f(z_b))$$

$$+2(1+r)(f(h_j^0) - f(z_b))$$

$$+(f(h_{j+1}^0) - f(z_b)))$$

D_j is as in the general case.

In the higher permeability panel, i.e. for $j > 3$, the formulae for the A_j etc are unchanged from the general case.

2.5 Initial and boundary conditions for running the model

The model program simulates a complete year of groundwater levels, taking as input two data files, one constructed from the climate variables, rainfall and evaporation, and a daily sequence of rhyne levels derived from the results of the IH water level recorders, by infilling from other sites where gaps occurred in the data, and the other consisting of the periodic readings of groundwater level in the dipwell transects. The rhyne levels provide a varying boundary condition for node $j = 1$, while the initial distribution of groundwater levels, necessary to start the model run, is simply obtained by averaging the groundwater levels from dipwells 2 to 5 of the transect for nodes $j > 1$.

Appendix 3 TWO-DIMENSIONAL MODEL

The Somerset Levels are dissected by a network of rhynes, usually following a gridiron pattern, dividing the land up into rectangular fields. Although there are wide variations in the width and depth of the rhynes, and in the quality and frequency of maintenance, in general each rectangular field is bounded, at least in the summer, by water-filled rhynes which provide the "wet fencing" of the Levels. The two-dimensional model seeks to represent a rectangular field bounded by two pairs of efficient, well-maintained rhynes, and the performance of the model is assessed against the observations from the four dipwell transects on West Sedgemoor.

3.1 Simple model with uniform permeability

The groundwater flow equation in two dimensions is

$$-\nabla(T\nabla h) = -\nabla T \cdot \nabla h - T\nabla^2 h = -S\frac{\partial h}{\partial t} + q$$

which is equivalent to

$$-\frac{\partial T}{\partial x}\frac{\partial h}{\partial x} - \frac{\partial T}{\partial y}\frac{\partial h}{\partial y} - T\frac{\partial^2 h}{\partial x^2} - T\frac{\partial^2 h}{\partial y^2} = -S\frac{\partial h}{\partial t} + q$$

Using central difference approximations and a grid interval of Δx in both the x and y directions, the finite difference equation is

$$-\frac{(T_{i+1j}-T_{i-1j})(h_{i+1j}-h_{i-1j})}{4\Delta x^2} - \frac{(T_{ij+1}-T_{ij-1})(h_{ij+1}-h_{ij-1})}{4\Delta x^2}$$

$$-T_{ij}\frac{(h_{i+1j}+h_{i-1j}+h_{ij+1}+h_{ij-1}-4h_{ij})}{\Delta x^2} = -S\frac{(h_{ij}-h_{ij}^0)}{\Delta t} + q$$

for $j = 1$ to n_c and $i = 1$ to n_r.

In the simple model, the transmissivity is assumed to be the product of a uniform permeability κ and the saturated depth of the aquifer $(h - z_b)$.

Reduction of this system of equations to tridiagonal form for solution is not trivial: the equations as stated above lead to a sparse matrix in which each row has up to five non-zero elements, comprising a diagonal element, elements immediately left and right of the diagonal, and two other elements well off the diagonal. Figure A3.1 shows the form of the matrix of coefficients of h_{ij} for a network of 4 x 4 nodes. Rapid inversion of this matrix would not be possible.
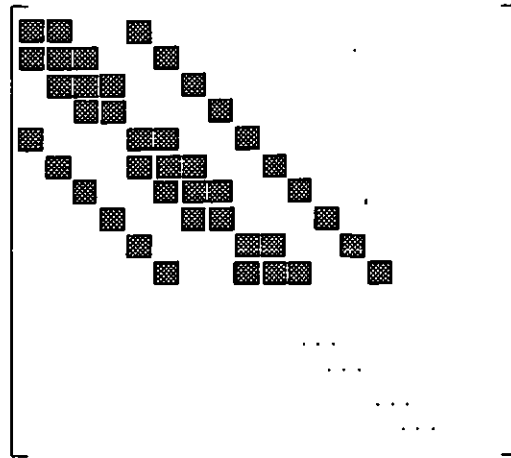
Figure A3.1 The pattern of non-zero coefficients for the equations relating to a two-dimensional model

Peaceman and Rachford (1955)[9] solved this problem by developing the Iterative Alternating Direction Implicit method, in which the calculation proceeds alternately by rows and columns, with values of h in adjacent rows, estimated in advance and improved by iteration, being held constant while a tridiagonal system is solved for the elements in a column, then the values in adjacent columns held constant while the row equations are solved. The discretised equation is re-arranged into two forms for the column and row calculations. For the column calculations the partial derivatives with respect to x are approximated from the estimated values h'_{ij}, and the partial derivatives with respect to y are approximated from the h_{ij}. The equations are solved by an iterative procedure, which produces progressively refined estimates of h_{ij} at the end of the timestep. The method was adapted to groundwater models by Prickett and Lonnquist (1971)[10], who programmed a number of simple applications to regional groundwater supply and aquifer pumping tests in FORTRAN.

For the column calculations

9 Peaceman D W & Rachford H H (1955) The numerical solution of parabolic and elliptic differential equations, *J. Soc. Industrial & Applied Maths 3(1)*, 28-41.

10 Prickett T A & Lonnquist C G (1971) Selected digital computer techniques for groundwater resource evaluation, *Illinois State Water Survey Bull. 53*, Urbana, Illinois.

$$-\frac{(T_{i+1j}-T_{i-1j})(h_{i+1j}-h_{i-1j})}{4\Delta x^2}-T_{ij}\frac{(h_{i+1j}-2h_{ij}+h_{i-1j})}{\Delta x^2}$$

$$=-S\frac{(h_{ij}-h_{ij}^0)}{\Delta t}+q+\frac{(T_{ij+1}-T_{ij-1})(h'_{ij+1}-h'_{ij-1})}{4\Delta x^2}$$

$$+T_{ij}\frac{(h'_{ij+1}-2h'_{ij}+h'_{ij-1})}{\Delta x^2}$$

Substituting for T_{ij}

$$-\kappa\frac{(h'_{i+1j}-h'_{i-1j})(h_{i+1j}-h_{i-1j})}{4\Delta x^2}$$

$$-\kappa(h'_{ij}-z_b)\frac{(h_{i+1j}-2h_{ij}+h_{i-1j})}{\Delta x^2}+S\frac{h_{ij}}{\Delta t}$$

$$=S\frac{h_{ij}^0}{\Delta t}+q+\kappa\frac{(h'_{ij+1}-h'_{ij-1})^2}{4\Delta x^2}$$

$$+\kappa(h'_{ij}-z_b)\frac{(h'_{ij+1}-2h'_{ij}+h'_{ij-1})}{\Delta x^2}$$

In tridiagonal form, these equations are

$$-(h'_{i-1j}+4(h'_{ij}-z_b)-h'_{i+1j})h_{i-1j}$$

$$+\left(8(h'_{ij}-z_b)+4\frac{\Delta x^2 S}{\kappa\Delta t}\right)h_{ij}$$

$$-(-h'_{i-1j}+4(h'_{ij}-z_b)+h'_{i+1j})h_{i+1j}$$

$$=4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t}+q\right)+(h'_{ij+1}-h'_{ij-1})^2+4(h'_{ij}-z_b)(h'_{ij+1}-2h'_{ij}+h'_{ij-1})$$

The elements of the coefficient matrix are

$$A_i = -(h'_{i-1j} + 4(h'_{ij} - z_b) - h'_{i+1j})$$

$$B_i = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_i = -(-h'_{i-1j} + 4(h'_{ij} - z_b) + h'_{i+1j})$$

$$D_i = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h^0_{ij}}{\Delta t} + q\right) + (h'_{ij+1} - h'_{ij-1})^2$$

$$+4(h'_{ij} - z_b)(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$-\frac{(T_{ij+1} - T_{ij-1})(h_{ij+1} - h_{ij-1})}{4\Delta x^2} - T_{ij}\frac{(h_{ij+1} - 2h_{ij} + h_{ij-1})}{\Delta x^2}$$

$$= -S\frac{(h_{ij} - h^0_{ij})}{\Delta t} + q + \frac{(T_{i+1j} - T_{i-1j})(h'_{i+1j} - h'_{i-1j})}{4\Delta x^2}$$

$$+T_{ij}\frac{(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})}{\Delta x^2}$$

Substituting for T_{ij}

$$-\kappa\frac{(h'_{ij+1} - h'_{ij-1})(h_{ij+1} - h_{ij-1})}{4\Delta x^2}$$

$$-\kappa(h'_{ij} - z_b)\frac{(h_{ij+1} - 2h_{ij} + h_{ij-1})}{\Delta x^2} + S\frac{h_{ij}}{\Delta t}$$

$$= S\frac{h^0_{ij}}{\Delta t} + q + \kappa\frac{(h'_{i+1j} - h'_{i-1j})^2}{4\Delta x^2}$$

$$+\kappa(h'_{ij} - z_b)\frac{(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})}{\Delta x^2}$$

A3.4

In tridiagonal form, these equations are

$$-(h'_{ij-1} + 4(h'_{ij} - z_b) - h'_{ij+1})h_{ij-1}$$

$$+\left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$-(-h'_{ij-1} + 4(h'_{ij} - z_b) + h'_{ij+1})h_{ij+1}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + (h'_{i+1j} - h'_{i-1j})^2 + 4(h'_{ij} - z_b)(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

The elements of the coefficient matrix are

$$A_j = -(h'_{ij-1} + 4(h'_{ij} - z_b) - h'_{ij+1})$$

$$B_j = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_j = -(-h'_{ij-1} + 4(h'_{ij} - z_b) + h'_{ij+1})$$

$$D_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + (h'_{i+1j} - h'_{i-1j})^2$$

$$+4(h'_{ij} - z_b)(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

At the end of each iteration, a check is made of the difference between the result of the current iteration and h'_{ij} and depending on the result a further iteration may take place, in which the current result takes the place of h'_{ij}.

3.1.1 Boundaries

The model is intended to simulate a rectangular field bounded by rhynes. It is necessary only to simulate one quarter of the field, i.e. the rectangle between the centre of the field and one vertex. The coefficients have to be defined for the five regions outlined in Figure A3.2.
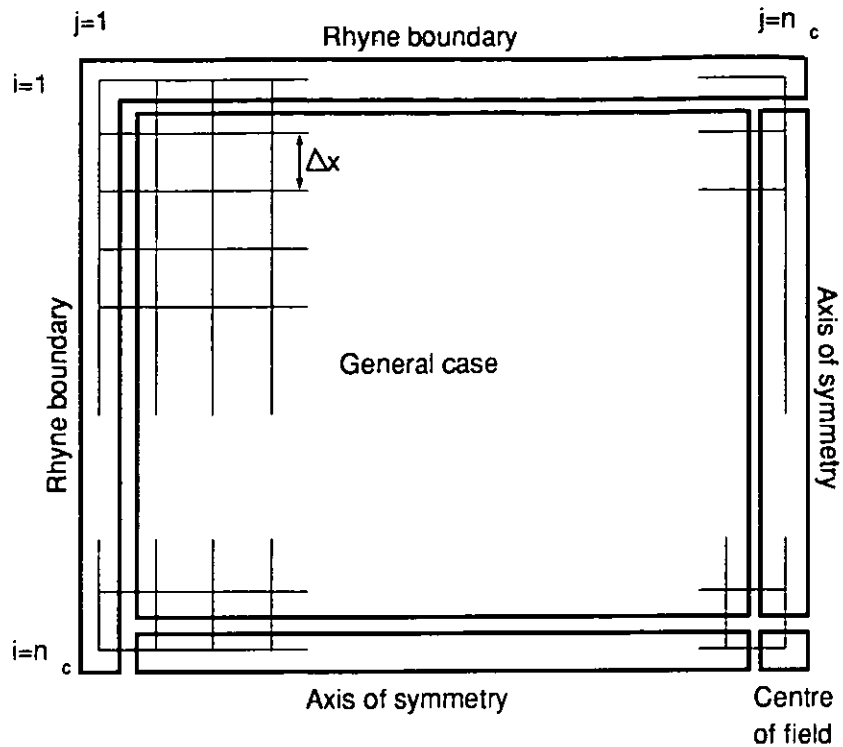
Figure A3.2 Plan of field quadrant, showing regions for evaluation of coefficients

At a rhyne boundary, for $i = 1$ or $j = 1$, assuming that water levels are equal in all the rhynes bounding the field,

$$h = h_{rhyne}(t)$$

For the column calculations

$$A_i = 0 \ ; \ B_i = 1 \ ; \ C_i = 0 \ ; \ D_i = h_{rhyne}$$

For the row calculations

$$A_j = 0 \ ; \ B_j = 1 \ ; \ C_j = 0 \ ; \ D_j = h_{rhyne}$$

Along an axis of symmetry of the field, the hydraulic gradient is zero.

Along the vertical ("north-south") axis $j = n_c$

$$\frac{\partial h}{\partial x} = 0$$

which implies

$$-\kappa(h - z_b)\frac{\partial^2 h}{\partial x^2} = -S\frac{\partial h}{\partial t} + q$$

The zero hydraulic gradient also implies that

$$h_{ij+1} = h_{ij-1}$$

For the column calculations, the discretised equation is modified to

$$-(h'_{i-1j} + 4(h'_{ij} - z_b) - h'_{i+1j})h_{i-1j}$$

$$+\left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$-(-h'_{i-1j} + 4(h'_{ij} - z_b) + h'_{i+1j})h_{i+1j}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{ij-1} - h'_{ij})$$

The elements of the coefficient matrix are

$$A_i = -(h'_{i-1j} + 4(h'_{ij} - z_b) - h'_{i+1j})$$

$$B_i = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_i = -(-h'_{i-1j} + 4(h'_{ij} - z_b) + h'_{i+1j})$$

$$D_i = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{ij-1} - h'_{ij})$$

For the row calculations, the discretised equation is modified to

$$-8(h'_{ij} - z_b)h_{ij-1} + \left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + (h'_{i+1j} - h'_{i-1j})^2 + 4(h'_{ij} - z_b)(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

The elements of the coefficient matrix are

$$A_j = -8(h'_{ij} - z_b)$$

$$B_j = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_j = 0$$

$$D_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h^0_{ij}}{\Delta t} + q\right) + (h'_{i+1j} - h'_{i-1j})^2$$

$$+ 4(h'_{ij} - z_b)(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

Along the horizontal ("east-west") axis $i = n$,

$$\frac{\partial h}{\partial y} = 0$$

which implies

$$-\kappa(h - z_b)\frac{\partial^2 h}{\partial y^2} = -S\frac{\partial h}{\partial t} + q$$

The zero hydraulic gradient also implies that

$$h_{i+1j} = h_{i-1j}$$

For the column calculations, the discretised equation is modified to

$$-8(h'_{ij} - z_b)h_{i-1j} + \left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h^0_{ij}}{\Delta t} + q\right) + (h'_{ij+1} - h'_{ij-1})^2$$

$$+ 4(h'_{ij} - z_b)(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

The elements of the coefficient matrix are

$$A_i = -8(h'_{ij} - z_b)$$

$$B_i = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_i = 0$$

$$D_i = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + (h'_{ij+1} - h'_{ij-1})^2$$

$$+ 4(h'_{ij} - z_b)(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations, the discretised equation is modified to

$$-(h'_{ij-1} + 4(h'_{ij} - z_b) - h'_{ij+1})h_{ij-1}$$

$$+\left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$-(-h'_{ij-1} + 4(h'_{ij} - z_b) + h'_{ij+1})h_{ij+1}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{i-1j} - h'_{ij})$$

The elements of the coefficient matrix are

$$A_j = -(h'_{ij-1} + 4(h'_{ij} - z_b) - h'_{ij+1})$$

$$B_j = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_j = -(-h'_{ij-1} + 4(h'_{ij} - z_b) + h'_{ij+1})$$

$$D_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{i-1j} - h'_{ij})$$

At the centre of the field, at the intersection of the two axes of symmetry, the equation for the column calculations is

$$-8(h'_{ij} - z_b)h_{i-1j} + \left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{ij-1} - h'_{ij})$$

and for the row calculations

$$-8(h'_{ij} - z_b)h_{ij-1} + \left(8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}\right)h_{ij}$$

$$= 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{i-1j} - h'_{ij})$$

For the column calculations, the elements of the coefficient matrix are

$$A_i = -8(h'_{ij} - z_b)$$

$$B_i = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_i = 0$$

$$D_i = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{ij-1} - h'_{ij})$$

For the row calculations

$$A_j = -8(h'_{ij} - z_b)$$

$$B_j = 8(h'_{ij} - z_b) + 4\frac{\Delta x^2 S}{\kappa \Delta t}$$

$$C_j = 0$$

$$D_j = 4\frac{\Delta x^2}{\kappa}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(h'_{ij} - z_b)(h'_{i-1j} - h'_{ij})$$

3.1.2 Ground surface

Flooding of the ground surface when the water table reaches ground level is simulated in exactly the same way as in the one-dimensional model.

3.2 Permeability varying with depth

The permeability is assumed to decrease exponentially with depth

$$\kappa(h) = \kappa_0 \exp{-\alpha(z_s - h)}$$

The discretised equation is rearranged as before for the purposes of the IADI method, with the transmissivities calculated from the h'_{ij}.

For the column calculations

$$-\frac{(T_{i+1j}-T_{i-1j})(h_{i+1j}-h_{i-1j})}{4\Delta x^2}-T_{ij}\frac{(h_{i+1j}-2h_{ij}+h_{i-1j})}{\Delta x^2}$$

$$=\left(S\frac{(h_{ij}-h_{ij}^0)}{\Delta t}+q\right)+\frac{(T_{ij+1}-T_{ij-1})(h'_{ij+1}-h'_{ij-1})}{4\Delta x^2}$$

$$+T_{ij}\frac{(h'_{ij+1}-2h'_{ij}+h'_{ij-1})}{\Delta x^2}$$

Substituting the generalised form for T_{ij}

$$-\frac{(f(h'_{i+1j})-f(h'_{i-1j}))(h_{i+1j}-h_{i-1j})}{4\Delta x^2}$$

$$-(f(h'_{ij})-f(z_b))\frac{(h_{i+1j}-2h_{ij}+h_{i-1j})}{\Delta x^2}+S\frac{h_{ij}}{\Delta t}$$

$$=S\frac{h_{ij}^0}{\Delta t}+q+\frac{(f(h'_{ij+1})-f(h'_{ij-1}))(h'_{ij+1}-h'_{ij-1})}{4\Delta x^2}$$

$$+(f(h'_{ij})-f(z_b))\frac{(h'_{ij+1}-2h'_{ij}+h'_{ij-1})}{\Delta x^2}$$

$$A_i=-(f(h'_{i-1j})+4(f(h'_{ij})-f(z_b))-f(h'_{i+1j}))$$

$$B_i=8(f(h'_{ij})-f(z_b))+4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i=-(-f(h'_{i-1j})+4(f(h'_{ij})-f(z_b))+f(h'_{i+1j}))$$

$$D_i=4\Delta x^2\left(S\frac{h_{ij}^0}{\Delta t}+q\right)$$

$$+(f(h'_{ij+1})-f(h'_{ij-1}))(h'_{ij+1}-h'_{ij-1})$$

$$+4(f(h'_{ij})-f(z_b))(h'_{ij+1}-2h'_{ij}+h'_{ij-1})$$

For the row calculations

$$-\frac{(T_{ij+1}-T_{ij-1})\,(h_{ij+1}-h_{ij-1})}{4\Delta x^2}-T_{ij}\frac{(h_{ij+1}-2h_{ij}+h_{ij-1})}{\Delta x^2}$$

$$=\left(S\frac{(h_{ij}-h_{ij}^0)}{\Delta t}+q\right)+\frac{(T_{i+1j}-T_{i-1j})\,(h'_{i+1j}-h'_{i-1j})}{4\Delta x^2}$$

$$+T_{ij}\frac{(h'_{i+1j}-2h'_{ij}+h'_{i-1j})}{\Delta x^2}$$

Substituting for T_{ij}

$$-\frac{(f(h'_{ij+1})-f(h'_{ij-1}))\,(h_{ij+1}-h_{ij-1})}{4\Delta x^2}$$

$$-(f(h'_{ij})-f(z_b))\frac{(h_{ij+1}-2h_{ij}+h_{ij-1})}{\Delta x^2}+S\frac{h_{ij}}{\Delta t}$$

$$=S\frac{h_{ij}^0}{\Delta t}+q+\frac{(f(h'_{i+1j})-f(h'_{i-1j}))\,(h'_{i+1j}-h'_{i-1j})}{4\Delta x^2}$$

$$+(f(h'_{ij})-f(z_b))\frac{(h'_{i+1j}-2h'_{ij}+h'_{i-1j})}{\Delta x^2}$$

$$\boxed{\begin{aligned}
A_j &= -(f(h'_{ij-1})+4(f(h'_{ij})-f(z_b))-f(h'_{ij+1}))\\[6pt]
B_j &= 8(f(h'_{ij})-f(z_b))+4\frac{\Delta x^2 S}{\Delta t}\\[6pt]
C_j &= -(-f(h'_{ij-1})+4(f(h'_{ij})-f(z_b))+f(h'_{ij+1}))\\[6pt]
D_j &= 4\Delta x^2\left(S\frac{h_{ij}^0}{\Delta t}+q\right)\\[6pt]
&\quad +(f(h'_{i+1j})-f(h'_{i-1j}))\,(h'_{i+1j}-h'_{i-1j})\\[6pt]
&\quad +4(f(h'_{ij})-f(z_b))\,(h'_{i+1j}-2h'_{ij}+h'_{i-1j})
\end{aligned}}$$

3.2.1 Boundaries

At a rhyne boundary, for $i = 1$ or $j = 1$, assuming that water levels are equal in all the rhynes bounding the field,

$$h = h_{rhyne}(t)$$

For the column calculations

$$A_i = 0 \; ; \; B_i = 1 \; ; \; C_i = 0 \; ; \; D_i = h_{rhyne}$$

For the row calculations

$$A_j = 0 \; ; \; B_j = 1 \; ; \; C_j = 0 \; ; \; D_j = h_{rhyne}$$

Along the vertical ("north-south") axis $j = n_c$

For the column calculations, the elements of the coefficient matrix are

$$A_i = -(f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{i+1j}))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = -(-f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{i+1j}))$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+8(f(h'_{ij}) - f(z_b))(h'_{ij-1} - h'_{ij})$$

For the row calculations, the elements of the coefficient matrix are

$$A_j = -8(f(h'_{ij}) - f(z_b))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = 0$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+(f(h'_{i+1j}) - f(h'_{i-1j}))(h'_{i+1j} - h'_{i-1j})$$

$$+4(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

Along the horizontal ("east-west") axis $i = n$,

For the column calculations, the elements of the coefficient matrix are

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = 0$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ (f(h'_{ij+1}) - f(h'_{ij-1}))(h'_{ij+1} - h'_{ij-1})$$

$$+ 4(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations, the elements of the coefficient matrix are

$$A_j = -(f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{ij+1}))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{ij+1}))$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ 8(f(h'_{ij}) - f(z_b))(h'_{i-1j} - h'_{ij})$$

At the centre of the field, the elements of the coefficient matrix are

For the column calculations

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = 0$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ 8(f(h'_{ij}) - f(z_b))(h'_{ij-1} - h'_{ij})$$

$$A_j = -8(f(h'_{ij}) - f(z_b))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = 0$$

$$D_j = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right)$$

$$+ 8(f(h'_{ij}) - f(z_b))(h'_{i-1j} - h'_{ij})$$

3.3 Specific yield varying with depth

S is assumed to have an exponential variation with depth

$$S = S_0 \exp(-\beta(z_s - h))$$

The effect of varying specific yield is taken into account by computing S at each time step, for use in the evaluation of the B_j and D_j.

3.4 Horizontal variation of permeability

As in the 1-dimensional model, the poor hydraulic connection between field and rhyne was simulated by defining a lower permeability in a panel 4 m wide along the edge of each rhyne. The permeability here is assumed equal to r times that in the field.

The inclusion of the low permeability panel considerably increases the number of regions into which the field quadrant must be divided for evaluation of the coefficients. 13 distinct regions must be considered (Figure A3.3), to take full account of the rhyne boundaries, the boundary between the low and high permeability panels and the axes of symmetry.
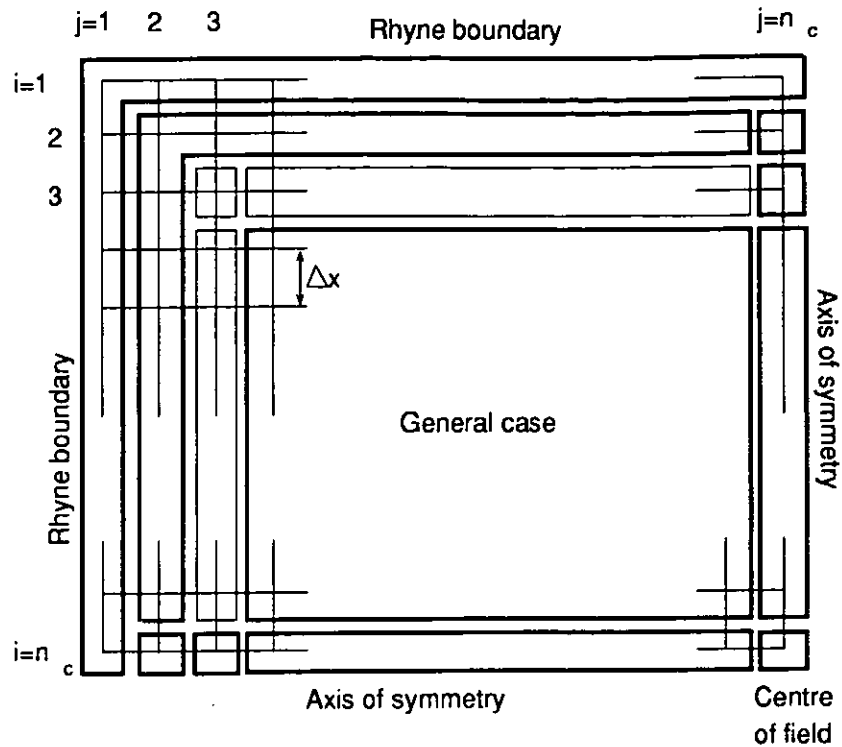
Figure A3.3 Plan of field quadrant with low permeability zone, showing regions for evaluation of coefficients

At the rhyne boundaries $i = 1$ or $j = 1$, the coefficients are unchanged from the general case, i.e.

For the column calculations

$$A_i = 0 \; ; \; B_i = 1 \; ; \; C_i = 0 \; ; \; D_i = h_{rhyne}$$

For the row calculations

$$A_j = 0 \; ; \; B_j = 1 \; ; \; C_j = 0 \; ; \; D_j = h_{rhyne}$$

In the low permeability panel, i.e. for $i = 2$ or $j = 2$, but excluding the nodes on the rhyne boundaries $j = 1$ and $i = 1$ and the axes of symmetry $j = n_c$ and $i = n_r$, the substitution of

$$T_{ij} = r(f(h_{ij}) - f(z_b))$$

implies the following changes to the coefficients B_j and D_j .

For the column calculations

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r\Delta t}$$

$$D_i = 4\frac{\Delta x^2}{r}\left(S\frac{h^0_{ij}}{\Delta t} + q\right)$$

$$+(f(h'_{ij+1}) - f(h'_{ij-1}))(h'_{ij+1} - h'_{ij-1})$$

$$+4(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r\Delta t}$$

$$D_j = 4\frac{\Delta x^2}{r}\left(S\frac{h^0_{ij}}{\Delta t} + q\right)$$

$$+(f(h'_{i+1j}) - f(h'_{i-1j}))(h'_{i+1j} - h'_{i-1j})$$

$$+4(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

At the point $i = 2$, $j = n_c$ on the vertical ("north-south") axis, the coefficients are

For the column calculations

$$A_i = -(f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{i+1j}))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r\Delta t}$$

$$C_i = -(-f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{i+1j}))$$

$$D_i = 4\frac{\Delta x^2}{r}\left(S\frac{h^0_{ij}}{\Delta t} + q\right) + 8(f(h'_{ij}) - f(z_b))(h'_{ij-1} - h'_{ij})$$

For the row calculations

$$A_j = -8(f(h'_{ij}) - f(z_b))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r \Delta t}$$

$$C_j = 0$$

$$D_j = 4\frac{\Delta x^2}{r}\left(S\frac{h_{ij}^0}{\Delta t} + q\right)$$

$$+ (f(h'_{i+1j}) - f(h'_{i-1j}))(h'_{i+1j} - h'_{i-1j})$$

$$+ 4(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

At the point $i = n, , j = 2$ on the horizontal ("east-west") axis, the coefficients are

For the column calculations

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r \Delta t}$$

$$C_i = 0$$

$$D_i = 4\frac{\Delta x^2}{r}\left(S\frac{h_{ij}^0}{\Delta t} + q\right)$$

$$+ (f(h'_{ij+1}) - f(h'_{ij-1}))(h'_{ij+1} - h'_{ij-1})$$

$$+ 4(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations, the elements of the coefficient matrix are

$$A_j = -(f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{ij+1}))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{r \Delta t}$$

$$C_j = -(-f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{ij+1}))$$

$$D_j = 4\frac{\Delta x^2}{r}\left(S\frac{h_{ij}^0}{\Delta t} + q\right) + 8(f(h'_{ij}) - f(z_b))(h'_{i-1j} - h'_{ij})$$

On the boundary $j = 3$ for $i > 3$, but excluding the node on the axis of symmetry $i = n_r$, modifications are made to the A_j, B_j and C_j as follows:

For the column calculations

$$A_i = -(f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{i+1j}))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = -(-f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{i+1j}))$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ ((f(h'_{ij+1}) - f(z_b)) - r(f(h'_{ij-1}) - f(z_b))) (h'_{ij+1} - h'_{ij-1})$$

$$+ 2(1 + r) (f(h'_{ij}) - f(z_b)) (h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$A_j = -(r(f(h'_{ij-1}) - f(z_b))$$

$$+ 2(1 + r) (f(h'_{ij}) - f(z_b))$$

$$- (f(h'_{ij+1}) - f(z_b)))$$

$$B_j = 4(1 + r) (f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-r(f(h'_{ij-1}) - f(z_b))$$

$$+ 2(1 + r) (f(h'_{ij}) - f(z_b))$$

$$+ (f(h'_{ij+1}) - f(z_b)))$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ (f(h'_{i+1j}) - f(h'_{i-1j})) (h'_{i+1j} - h'_{i-1j})$$

$$+ 4(f(h'_{ij}) - f(z_b)) (h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

At the point $i = n_r$, $j = 3$ on the horizontal ("east-west") axis, the coefficients are

For the column calculations

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = 0$$

$$D_i = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right)$$

$$+ ((f(h'_{ij+1}) - f(z_b)) - r(f(h'_{ij-1}) - f(z_b)))(h'_{ij+1} - h'_{ij-1})$$

$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$A_j = -(r(f(h'_{ij-1}) - f(z_b))$$

$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$

$$- (f(h'_{ij+1}) - f(z_b))$$

$$B_j = 4(1 + r)(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-r(f(h'_{ij-1}) - f(z_b))$$

$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$

$$+ (f(h'_{ij+1}) - f(z_b))$$

$$D_j = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right) + 8(f(h'_{ij}) - f(z_b))(h'_{i-1j} - h'_{ij})$$

On the boundary $i = 3$ for $j > 3$, but excluding the node on the vertical ("north-south") axis $j = n_c$, modifications are made to the A_j, B_j and C_j as follows:

For the column calculations

$$A_i = -(r(f(h'_{i-1j}) - f(z_b))$$

$$+2(1+r)(f(h'_{ij}) - f(z_b))$$

$$-(f(h'_{i+1j}) - f(z_b)))$$

$$B_i = 4(1+r)(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = -(-r(f(h'_{i-1j}) - f(z_b))$$

$$+2(1+r)(f(h'_{ij}) - f(z_b))$$

$$+(f(h'_{i+1j}) - f(z_b)))$$

$$D_i = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right)$$

$$+(f(h'_{ij+1}) - f(h'_{ij-1}))(h'_{ij+1} - h'_{ij-1})$$

$$+4(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$A_j = -(f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{ij+1}))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{ij+1}))$$

$$D_j = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right)$$

$$+((f(h'_{i+1j}) - f(z_b)) - r(f(h'_{i-1j}) - f(z_b)))(h'_{i+1j} - h'_{i-1j})$$

$$+2(1+r)(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

At the node $i = 3, j = 3$ the coefficients are

For the column calculations

$$A_i = -(r(f(h'_{i-1j}) - f(z_b))$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$
$$- (f(h'_{i+1j}) - f(z_b)))$$

$$B_i = 4(1 + r)(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = -(-r(f(h'_{i-1j}) - f(z_b))$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$
$$+ (f(h'_{i+1j}) - f(z_b)))$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$
$$+ ((f(h'_{ij+1}) - f(z_b)) - r(f(h'_{ij-1}) - f(z_b)))(h'_{ij+1} - h'_{ij-1})$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))(h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$A_j = -(r(f(h'_{ij-1}) - f(z_b))$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$
$$- (f(h'_{ij+1}) - f(z_b)))$$

$$B_j = 4(1 + r)(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-r(f(h'_{ij-1}) - f(z_b))$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))$$
$$+ (f(h'_{ij+1}) - f(z_b)))$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$
$$+ ((f(h'_{i+1j}) - f(z_b)) - r(f(h'_{i-1j}) - f(z_b)))(h'_{i+1j} - h'_{i-1j})$$
$$+ 2(1 + r)(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

In the higher permeability panel, i.e. for $j > 3$ *and* $i > 3$, but excluding the nodes on the axes of symmetry, $i = n_r$ and $j = n_c$, the formulae for the coefficients are unchanged from the general case.

On the vertical ("north-south") axis $j = n_c$ for $i > 3$, **but excluding the point** $i = n_r$

For the column calculations

$$A_i = -(f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{i+1j}))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = -(-f(h'_{i-1j}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{i+1j}))$$

$$D_i = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right) + 8(f(h'_{ij}) - f(z_b))(h'_{ij-1} - h'_{ij})$$

For the row calculations

$$A_j = -8(f(h'_{ij}) - f(z_b))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = 0$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right)$$

$$+ (f(h'_{i+1j}) - f(h'_{i-1j}))(h'_{i+1j} - h'_{i-1j})$$

$$+ 4(f(h'_{ij}) - f(z_b))(h'_{i+1j} - 2h'_{ij} + h'_{i-1j})$$

On the horizontal ("east-west") axis $i = n_r$ for $j > 3$ but excluding the point $j = n_c$

For the column calculations

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = 0$$

$$D_i = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right)$$

$$+ (f(h'_{ij+1}) - f(h'_{ij-1})) (h'_{ij+1} - h'_{ij-1})$$

$$+ 4(f(h'_{ij}) - f(z_b)) (h'_{ij+1} - 2h'_{ij} + h'_{ij-1})$$

For the row calculations

$$A_j = -(f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) - f(h'_{ij+1}))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = -(-f(h'_{ij-1}) + 4(f(h'_{ij}) - f(z_b)) + f(h'_{ij+1}))$$

$$D_j = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right) + 8(f(h'_{ij}) - f(z_b)) (h'_{i-1j} - h'_{ij})$$

At the intersection of the two axes of symmetry, the point $i = n_r , j = n_c$

For the column calculations

$$A_i = -8(f(h'_{ij}) - f(z_b))$$

$$B_i = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_i = 0$$

$$D_i = 4\Delta x^2 \left(S\frac{h^0_{ij}}{\Delta t} + q \right) + 8(f(h'_{ij}) - f(z_b)) (h'_{ij-1} - h'_{ij})$$

For the row calculations

$$A_j = -8(f(h'_{ij}) - f(z_b))$$

$$B_j = 8(f(h'_{ij}) - f(z_b)) + 4\frac{\Delta x^2 S}{\Delta t}$$

$$C_j = 0$$

$$D_j = 4\Delta x^2 \left(S\frac{h_{ij}^0}{\Delta t} + q \right) + 8(f(h'_{ij}) - f(z_b))(h'_{i-1j} - h'_{ij})$$

Appendix 4   BASIC PROGRAM LISTINGS FOR GROUND-WATER MODEL

4.1 Simple 1-dimensional model - 1d-simp.bas

```
DECLARE SUB Pause ()
DECLARE FUNCTION f (k!, Surface!, z!, Gamma!, RGamma)
DECLARE SUB ScreenBorder ()


'      1-dimensional groundwater simulation program
'      ==============================================

' This version includes an impermeable base at an elevation Floor mOD
' and vertically uniform permeability and specific yield.

' Poor hydraulic connection with rhyne simulated by low permeability
' panel.

' Dipwell levels read in from file and compared with model predictions


'      K Gilman, May 1991


' Model represents vertical slice through horizontal water table aquifer
' discretised with grid of NNodes equally-spaced nodes, Dx% (m) apart.
' This slice is from rhyne at left to centre of field at right. Saturated
' thickness from water table to underlying impermeable boundary
' at a level of Floor (mOD) defines transmissivity.
' Ground surface simulated by
' specific yield rising to 100% and much larger permeability.

' Rhyne at extreme left defines h(1), right bdy (j=NNodes) is zero flow.
' Rhyne level hrhyne (mOD) is forcing function, infiltration q (mm) is
' distributed uniformly in horizontal direction. These inputs are read
' from a data file at each time step in order day number, rainfall,
' evaporation and rhyne water level.

' Time interval Dt = 1 day
' Grid interval Dx% = 2 metres

'      Definition of variables

' h0(j)   ----- Heads at start of time increment (m) for j=1 to NNodes
' h(j)    ----- Heads at end of time increment (m) for j=1 to NNodes
' hMax(j) ----- Greatest head met so far at node j
' hMin(j) ----- Lowest head met so far at node j
' S       ----- Specific yield (entered as %, used as fraction in program)
' q       ----- Net infiltration rate (mm)
' k       ----- Permeability (m/d)
' Grid%   ----- Array for storage of grid for video display

DIM h(200), b(200), g(200), h0(200), hMax(200), hMin(200), Grid%(8000)
      NSteps = 365
      Dt = 1
      Dx% = 2
```

' Setup screen, input of parameters.
' Rhyne factor r is ratio of permeabilities left and right of j=3. Low
' permeability panel between j=0 and j=3 is to take account of sealing
' of rhyne channel, compaction of banks, and shallowness of rhyne.
' Field width is between parallel rhynes.

' First screen for input of parameters

SCREEN 0: COLOR , 1: CLS

CALL ScreenBorder

```
COLOR 14, 1: LOCATE 3, 15: PRINT "COMPASS - COM";
COLOR 12, 1: PRINT "partmented "; : COLOR 14, 1: PRINT "P";
COLOR 12, 1: PRINT "hreatic "; : COLOR 14, 1: PRINT "A";
COLOR 12, 1: PRINT "quifer "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "imulation "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "ystem "
LOCATE 5, 15: PRINT "One-dimensional mire transect model": COLOR 3, 1
LOCATE 7, 15: PRINT "Permeability:  ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT "  m/d"
LOCATE 11, 15: PRINT "Spec yield:    ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT "  %"
LOCATE 15, 15: PRINT "Rhyne factor: ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1
LOCATE 17, 15: PRINT "Base level:   ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT "  mOD  (Base of peat)"
LOCATE 19, 15: PRINT "Surface level: ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT " mOD"
LOCATE 21, 15: PRINT "Field width:   ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT " m    (between rhynes)"
LOCATE 23, 15: PRINT "Filename:   A:";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT ".DAT"
COLOR 1, 3
LOCATE 7, 30: INPUT " ", k
LOCATE 11, 30: INPUT " ", S0
LOCATE 15, 30: INPUT " ", r
LOCATE 17, 30: INPUT " ", Floor
LOCATE 19, 30: INPUT " ", Surface
LOCATE 21, 30: INPUT " ", FieldWidth
LOCATE 23, 30: INPUT " ", File$
```

' Compute NNodes from width of field and grid interval
' Select Gamma, parameter for above-ground flows.

```
NNodes = CINT(.5 * FieldWidth / Dx%) + 1
Gamma = 25
RGamma = 1 / Gamma
```

' Open input and output files - #1 for climate data, #3 for dipwell data

```
OPEN ("a:" + File$ + ".dat") FOR INPUT AS #1
OPEN ("a:" + File$ + "out.dat") FOR OUTPUT AS #2
OPEN ("a:" + File$ + "dw.dat") FOR INPUT AS #3
```

```
' Input coordinates of dipwells from #3

FOR j = 1 TO 5
        INPUT #3, xDw
        jDw(j) = 1 + CINT(xDw / Dx%)
NEXT j
INPUT #3, yDw

' Input first data from dipwell file #3
' Set starting value for h array

INPUT #3, DayDw
        NObs = NObs + 5
        FOR j = 1 TO 5
                INPUT #3, Dw(j)                  \
                Sum = Sum + Dw(j)
                SumSq = SumSq + Dw(j) ^ 2
        NEXT j
        hStart = (Sum - Dw(1)) / 4

' Set up graphics window for plotting. Program tests for VGA or EGA
' Error handling routine CGA at end of program resets colours for
' CGA or mono if necessary.

ON ERROR GOTO CGA
CLS
        col3 = 3            '  cyan
        col4 = 4            '  red
        col6 = 6            '  brown
        col9 = 9            '  light blue
        col11 = 11           '  light cyan
        col15 = 15           '  bright white
        SCREEN 8
        COLOR 3
VIEW (15, 5)-(590, 160), 0, col9
VIEW (50, 20)-(550, 145)
WINDOW SCREEN (0, 0)-(500, 125)
LINE (0, 0)-(500, 60), col9, B
LINE (0, 65)-(365, 125), col9, B
LOCATE 1, 27
PRINT " Distance from rhyne, m "     ·
LOCATE 2, 11
IF NNodes > 1 + 70 / Dx% THEN Npts = 1 + 70 / Dx% ELSE Npts = NNodes

' Labels for x axis along top of section plot

FOR j = 1 TO Npts STEP 10 / Dx%
        PRINT USING " ###    "; (j - 1) * Dx%;
NEXT j

' Grid for section plot

Top = 60 - 50 * (Surface - 3.8)
LINE (50, Top)-(50, 60), col6
FOR i = 2 TO Npts
IF (i * Dx% MOD 10) = Dx% THEN LineStyle = &HFFFF ELSE LineStyle =
&HF0F0
LINE (50 + 200 * (i - 1) * Dx% / 31, Top)-(50 + 200 * (i - 1) * Dx% / 31, 60),
col9, , LineStyle
NEXT i
```

```
LINE (50, Top)-(500, Top), col6
LOCATE 10, 9: PRINT "Section along transect"
GET (0, 0)-(500, 60), Grid%

' Grid for annual plot

LINE (0, 65 + Top)-(365, 65 + Top), col6
LINE (31, 65)-(31, 125), col9, , &HF0F0
LINE (59, 65)-(59, 125), col9, , &HF0F0
LINE (90, 65)-(90, 125), col9, , &HF0F0
LINE (120, 65)-(120, 125), col9, , &HF0F0
LINE (151, 65)-(151, 125), col9, , &HF0F0
LINE (181, 65)-(181, 125), col9, , &HF0F0
LINE (212, 65)-(212, 125), col9, , &HF0F0
LINE (243, 65)-(243, 125), col9, , &HF0F0
LINE (273, 65)-(273, 125), col9, , &HF0F0
LINE (304, 65)-(304, 125), col9, , &HF0F0
LINE (335, 65)-(335, 125), col9, , &HF0F0

' Labels for time axis of annual plot

LOCATE 20, 9: PRINT "J  F  M  A  M  J  J  A  S  O  N  D"

' Information panel

LOCATE 12, 54: PRINT "1-DIM MODEL"
LOCATE 14, 54: PRINT "Permeability (m/d):"
LOCATE 15, 54: PRINT USING "#.##"; k
LOCATE 16, 54: PRINT "Spec yield (%):"
LOCATE 17, 54: PRINT USING "##.#"; S0
LOCATE 18, 54: PRINT "Rhyne factor:"; r
LOCATE 17, 9: PRINT "Annual plot"
LOCATE 18, 9: PRINT "Node"; jDw(5); "("; (jDw(5) - 1) * Dx%; "m )"
LOCATE 19, 54: PRINT "Base level:"; Floor; "mOD"
LOCATE 20, 54: PRINT "Surface:"; Surface; "mOD"

' Fill arrays with default and starting values

FOR j = 1 TO NNodes
    h0(j) = hStart
    h(j) = hStart
    hMin(j) = Surface
    hMax(j) = Floor
NEXT j

' Start of simulation

Time = 0
ObjFun = 0
FOR Istep = 1 TO NSteps
Time = Time + Dt

' Input climate and rhyne level data from #1

INPUT #1, Day, Rain, Evap, hrhyne
q = (Rain - .95 * Evap) * .001
    h(0) = hrhyne
    h0(0) = hrhyne
    h(1) = hrhyne
    h0(1) = hrhyne

' Estimate heads at end of time interval by implicit method

f0 = f(k, Surface, Floor, Gamma, RGamma)
```

```
FOR j = 1 TO NNodes
h0(j) = h(j)
```

' Calculate B & G arrays for each of 5 regions

```
f1 = f(k, Surface, h(j - 1), Gamma, RGamma) - f0
f2 = f(k, Surface, h(j), Gamma, RGamma) - f0
f3 = f(k, Surface, h(j + 1), Gamma, RGamma) - f0

        z = 10 * (h(j) - Surface)
        IF z > 1 THEN
            S = 1
        ELSE
            IF z < 0 THEN
                S = .01 * S0
            ELSE
                S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
            END IF
        END IF
```

' Assign node to one of five regions

```
Region = j
IF j > 3 AND j < NNodes THEN Region = 4
IF j = NNodes THEN Region = 5

Rho = 4 * Dx% * Dx% / Dt

SELECT CASE Region
        CASE 1
            AA = 0: BB = 1: CC = 0: DD = hrhyne
        CASE 2
            AA = -r * (f1 + 4 * f2 - f3)
            BB = 8 * r * f2 + Rho * S
            CC = -AA - 8 * r * f2
            DD = Rho * (S * h0(j) + q * Dt)
        CASE 3
            AA = -(r * f1 + 2 * (1 + r) * f2 - f3)
            BB = 4 * (1 + r) * f2 + Rho * S
            CC = -AA - 4 * (1 + r) * f2
            DD = Rho * (S * h0(j) + q * Dt)
        CASE 4
            AA = -(f1 + 4 * f2 - f3)
            BB = 8 * f2 + Rho * S
            CC = -AA - 8 * f2
            DD = Rho * (S * h0(j) + q * Dt)
        CASE 5
            AA = -8 * f2
            BB = 8 * f2 + Rho * S
            CC = 0
            DD = Rho * (S * h0(j) + q * Dt)
END SELECT

W = BB - AA * b(j - 1)
b(j) = CC / W
g(j) = (DD - AA * g(j - 1)) / W
NEXT j
```

' Solve linear system of equations for heads

```
h(NNodes) = g(NNodes)
FOR n = NNodes - 1 TO 1 STEP -1
     h(n) = g(n) - b(n) * h(n + 1)
NEXT n

' Print results to screen and file
' Calculate objective function ObjFun

IF Day = DayDw THEN
     FOR i = 1 TO 5
          Deviation = h(jDw(i)) - Dw(i)
          ObjFun = ObjFun + Deviation * Deviation
     NEXT i
END IF

' Re-plot grid for section plot

PUT (0, 0), Grid%, PSET

' Plot rhyne water level

LINE (30, 60 - (h(1) - 3.8) * 50)-(50, 60 - (h(1) - 3.8) * 50)

' Plot water table

FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(j) - 3.8) * 50), col11, , &HCCCC
NEXT j

' Plot observed dipwell levels on section plot

IF Day = DayDw THEN
  FOR i = 1 TO 5
    CIRCLE (50 + 200 * (jDw(i) - 1) * Dx% / 31, 60 - (Dw(i) - 3.8) * 50), 4, col4
  NEXT i
END IF

' Update maximum and minimum levels

FOR j = 1 TO NNodes
     IF h(j) > hMax(j) THEN
          hMax(j) = h(j)
     ELSE
          IF h(j) < hMin(j) THEN
               hMin(j) = h(j)
          END IF
     END IF
NEXT j

' Plot water level at field end dipwell on annual plot

LINE (Istep, 125 - (h0(jDw(5)) - 3.8) * 50)-(Istep, 125 - (h(jDw(5)) - 3.8) * 50),
col15

' Plot observed levels at field end dipwell on annual plot
' Update sum of squares for computation of objective function

IF Day = DayDw THEN
     CIRCLE (Istep, 125 - (Dw(5) - 3.8) * 50), 4, col4
     IF NOT EOF(3) THEN
          INPUT #3, DayDw
          NObs = NObs + 5
          FOR j = 1 TO 5
               INPUT #3, Dw(j)
               Sum = Sum + Dw(j)
```

A4.6

```
                    SumSq = SumSq + Dw(j) ^ 2
          NEXT j
        END IF
END IF

' Print footer information panel and output predicted dipwell levels to file

LOCATE 22, 2
PRINT USING " & ### & #.### & ##.# & ##.# &"; "Day:"; Time; "  Rhyne
level"; hrhyne; "mOD.    Rainfall "; Rain; "mm.  Evapn "; Evap; "mm."
LOCATE 23, 2
PRINT USING " & #####"; "Output to file A:" + File$ + "OUT.DAT: "; Day;
FOR i = 1 TO 5
        PRINT USING "  #.###"; h(jDw(i));
NEXT i: PRINT
PRINT #2, USING "#####"; Day;
FOR i = 1 TO 5
        PRINT #2, USING "  #.###"; h(jDw(i));
NEXT i
PRINT #2,
NEXT Istep

' Plot maximum and minimum water table levels on section plot

LINE (50, Top)-(50, 50), col6
LINE (50, Top)-(500, Top), col6
LINE (30, 60 - (hMax(1) - 3.8) * 50)-(50, 60 - (hMax(1) - 3.8) * 50), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (hMax(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - (hMin(1) - 3.8) * 50)-(50, 60 - (hMin(1) - 3.8) * 50), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (hMin(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - (h(1) - 3.8) * 50)-(50, 60 - (h(1) - 3.8) * 50), col11
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(j) - 3.8) * 50), col11, , &HCCCC
NEXT j
```

' Pause to allow PrtSc command. Resume on keypress.

```
CALL Pause
```

' Final screen displays objective function

```
SCREEN 0: COLOR , 1: CLS : COLOR 11, 1
CALL ScreenBorder
LOCATE 4, 15: PRINT "1-dimensional mire transect model": COLOR 3, 1
LOCATE 10, 15: PRINT "Input file  A:"; File$; ".DAT"
SumSq = SumSq - Sum * Sum / NObs
LOCATE 15, 15: PRINT "Objective function"; 100 * ObjFun / SumSq; "%"
LOCATE 17, 20
PRINT "i.e."; 100 * (1 - ObjFun / SumSq); "% of variance accounted for by
model"
```

' Pause to allow PrtSc command. Return to DOS on keypress.

```
CALL Pause
END
```

' Error handling routine to cope with CGA screen
' Called when SCREEN 8 executed

```
CGA:
    SCREEN 2
    col3 = 1
    col4 = 1
    col6 = 1
    col9 = 1
    col11 = 1
    col15 = 1
    RESUME NEXT

FUNCTION f (k, Surface, z, Gamma, RGamma)
IF z <= Surface THEN
    f = k * z
ELSE
    f = (k * RGamma) * (EXP(Gamma * (z - Surface)) - 1) + k * Surface
END IF
END FUNCTION

SUB Pause
'
' Awaits key input before end of execution. This allows screen dump on PC
' using Prtsc.
'

WHILE INKEY$ = ""
WEND
END SUB

SUB ScreenBorder
COLOR 4
LOCATE 1, 1: PRINT CHR$(201);
FOR l = 2 TO 78: PRINT CHR$(205); : NEXT l
PRINT CHR$(187);
FOR l = 2 TO 24: LOCATE l, 79: PRINT CHR$(186); : NEXT l
LOCATE 25, 79: PRINT CHR$(188);
FOR l = 78 TO 2 STEP -1: LOCATE 25, l: PRINT CHR$(205); : NEXT l
LOCATE 25, 1: PRINT CHR$(200);
FOR l = 24 TO 2 STEP -1: LOCATE l, 1: PRINT CHR$(186); : NEXT l
END SUB
```

4.2 Complex 1-dimensional model - 1d-comp.bas

```
DECLARE SUB Pause ()
DECLARE FUNCTION f (k!, Alpha!, RAlpha, Surface!, z!, Gamma!, RGamma)
DECLARE SUB ScreenBorder ()


'       1-dimensional groundwater simulation program
'       ==============================================

' This version includes an impermeable base at an elevation Floor mOD
' and exponential variation in both permeability and specific yield.

' Poor hydraulic connection with rhyne simulated by low permeability
' panel.

' Dipwell levels read in from file and compared with model predictions


'       K Gilman, May 1991
```

' Model represents vertical slice through horizontal water table aquifer
' discretised with grid of NNodes equally-spaced nodes, Dx% (m) apart.
' This slice is from rhyne at left to centre of field at right. Saturated
' thickness from water table to underlying impermeable boundary
' at a level of Floor (mOD) defines transmissivity. Permeability assumed
' to vary exponentially with depth. Above ground flows simulated by
' specific yield rising to 100% and much larger permeability.

' Rhyne at extreme left defines h(1), right bdy (j=NNodes) is zero flow.
' Rhyne level hrhyne (mOD) is forcing function, infiltration q (mm) is
' distributed uniformly in horizontal direction. These inputs are read
' from a data file at each time step in order day number, rainfall,
' evaporation and rhyne water level.

' Time interval Dt = 1 day
' Grid interval Dx% = 2 metres

'   Definition of variables
,

' h0(j) ----- Heads at start of time increment (m) for j=1 to NNodes
' h(j)   ----- Heads at end of time increment (m) for j=1 to NNodes
' hMax(j) ----- Greatest head met so far at node j
' hMin(j) ----- Lowest head met so far at node j
' S   ----- Specific yield (entered as %, used as fraction in program)
' q   ----- Net infiltration rate (mm)
' k   ----- Permeability (m/d)
' Grid% ----- Array for storage of grid for video display

DIM h(200), B(200), g(200), h0(200), hMax(200), hMin(200), Grid%(8000)
        Nsteps = 365
        Dt = 1
        Dx% = 2

' Setup screen, input of parameters. k and S defined as product of
' value at surface and exponential function of depth below surface.
' Rhyne factor r is ratio of permeabilities left and right of j=3. Low
' permeability panel between j=0 and j=3 is to take account of sealing
' of rhyne channel, compaction of banks, and shallowness of rhyne.
' Field width is between parallel rhynes.

' First screen for input of parameters

SCREEN 0: COLOR , 1: CLS

CALL ScreenBorder

COLOR 14, 1: LOCATE 3, 15: PRINT "COMPASS - COM";
COLOR 12, 1: PRINT "partmented "; : COLOR 14, 1: PRINT "P";
COLOR 12, 1: PRINT "hreatic "; : COLOR 14, 1: PRINT "A";
COLOR 12, 1: PRINT "quifer "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "imulation "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "ystem "
LOCATE 5, 15: PRINT "One-dimensional mire transect model": COLOR 3, 1
LOCATE 7, 15: PRINT "Permeability: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT " m/d"
LOCATE 9, 25: PRINT "Exp coeff: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT
LOCATE 11, 15: PRINT "Spec yield: ";
COLOR 3, 3: PRINT "   "; .

```
COLOR 3, 1: PRINT "  %"
LOCATE 13, 25: PRINT "Exp coeff:     ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT
LOCATE 15, 15: PRINT "Rhyne factor: ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1
LOCATE 17, 15: PRINT "Base level:    ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT " mOD  (Base of peat)"
LOCATE 19, 15: PRINT "Surface level: ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT " mOD"
LOCATE 21, 15: PRINT "Field width:   ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT " m    (between rhynes)"
LOCATE 23, 15: PRINT "Filename:    A:";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT ".DAT"
COLOR 1, 3
LOCATE 7, 30: INPUT " ", k
LOCATE 9, 40: INPUT " ", Alpha
LOCATE 11, 30: INPUT " ", S0
LOCATE 13, 40: INPUT " ", Beta
LOCATE 15, 30: INPUT " ", r
LOCATE 17, 30: INPUT " ", Floor
LOCATE 19, 30: INPUT " ", Surface
LOCATE 21, 30: INPUT " ", FieldWidth
LOCATE 23, 30: INPUT " ", File$

' Compute NNodes from width of field and grid interval
' Select Gamma, parameter for above-ground flows.

NNodes = CINT(.5 * FieldWidth / Dx%) + 1
RAlpha = 1 / Alpha
Gamma = 25
RGamma = 1 / Gamma

' Open input and output files - #1 for climate data, #3 for dipwell data

OPEN ("a:" + File$ + ".dat") FOR INPUT AS #1
OPEN ("a:" + File$ + "out.dat") FOR OUTPUT AS #2
OPEN ("a:" + File$ + "dw.dat") FOR INPUT AS #3

' Input coordinates of dipwells from #3

FOR j = 1 TO 5
    INPUT #3, xDw
    jDw(j) = 1 + CINT(xDw / Dx%)
NEXT j
INPUT #3, yDw

' Input first data from dipwell file #3
' Set starting values for h array

INPUT #3, DayDw
    NObs = NObs + 5
    FOR j = 1 TO 5
        INPUT #3, Dw(j)
        Sum = Sum + Dw(j)
```

```
            SumSq = SumSq + Dw(j) ^ 2
        NEXT j
        hStart = (Sum - Dw(1)) / 4

' Set up graphics window for plotting. Program tests for VGA or EGA
' Error handling routine CGA at end of program resets colours for
' CGA or mono if necessary.

ON ERROR GOTO CGA
CLS
        col3 = 3            ' cyan
        col4 = 4            ' red
        col6 = 6            ' brown
        col9 = 9            ' light blue
        col11 = 11          ' light cyan
        col15 = 15          ' bright white
        SCREEN 8
        COLOR 3
VIEW (15, 5)-(590, 160), 0, col9
VIEW (50, 20)-(550, 145)
WINDOW SCREEN (0, 0)-(500, 125)
LINE (0, 0)-(500, 60), col9, B
LINE (0, 65)-(365, 125), col9, B
LOCATE 1, 27
PRINT " Distance from rhyne, m "
LOCATE 2, 11
IF NNodes > 1 + 70 / Dx% THEN Npts = 1 + 70 / Dx% ELSE Npts = NNodes

' Labels for x axis along top of section plot

FOR j = 1 TO Npts STEP 10 / Dx%
    PRINT USING " ###    "; (j - 1) * Dx%;
NEXT j

' Grid for section plot

Top = 60 - 50 * (Surface - 3.8)
LINE (50, Top)-(50, 60), col6
FOR i = 2 TO Npts
IF (i * Dx% MOD 10) = Dx% THEN LineStyle = &HFFFF ELSE LineStyle =
&HF0F0
LINE (50 + 200 * (i - 1) * Dx% / 31, Top)-(50 + 200 * (i - 1) * Dx% / 31, 60),
col9, , LineStyle
NEXT i
LINE (50, Top)-(500, Top), col6
LOCATE 10, 9: PRINT "Section along transect"
GET (0, 0)-(500, 60), Grid%

' Grid for annual plot

LINE (0, 65 + Top)-(365, 65 + Top), col6
LINE (31, 65)-(31, 125), col9, , &HF0F0
LINE (59, 65)-(59, 125), col9, , &HF0F0
LINE (90, 65)-(90, 125), col9, , &HF0F0
LINE (120, 65)-(120, 125), col9, , &HF0F0
LINE (151, 65)-(151, 125), col9, , &HF0F0
LINE (181, 65)-(181, 125), col9, , &HF0F0
LINE (212, 65)-(212, 125), col9, , &HF0F0
LINE (243, 65)-(243, 125), col9, , &HF0F0
```

```
LINE (273, 65)-(273, 125), col9, , &HF0F0
LINE (304, 65)-(304, 125), col9, , &HF0F0
LINE (335, 65)-(335, 125), col9, , &HF0F0
```

' Labels for time axis of annual plot

```
LOCATE 20, 9: PRINT "J  F  M  A  M  J  J  A  S  O  N  D"
```

' Information panel

```
LOCATE 12, 54: PRINT "1-DIM MODEL"
LOCATE 14, 54: PRINT "Permeability (m/d):"
LOCATE 15, 54: PRINT USING "#.## & #.## &"; k; "exp -"; Alpha; "z"
LOCATE 16, 54: PRINT "Spec yield (%):"
LOCATE 17, 54: PRINT USING "##.# & #.## &"; S0; "exp -"; Beta; "z"
LOCATE 18, 54: PRINT "Rhyne factor:"; r
LOCATE 17, 9: PRINT "Annual plot"
LOCATE 18, 9: PRINT "Node"; jDw(5); "("; (jDw(5) - 1) * Dx%; "m )"
LOCATE 19, 54: PRINT "Base level:"; Floor; "mOD"
LOCATE 20, 54: PRINT "Surface:"; Surface; "mOD"
```

' Fill arrays with default and starting values

```
FOR j = 1 TO NNodes
     h0(j) = hStart
     h(j) = hStart
     hMin(j) = Surface
     hMax(j) = Floor
NEXT j
```

' Start of simulation

```
Time = 0
ObjFun = 0
FOR Istep = 1 TO Nsteps
Time = Time + Dt
```

' Input climate and rhyne level data from #1

```
INPUT #1, Day, Rain, Evap, hrhyne
q = (Rain - .95 * Evap) * .001
     h(0) = hrhyne
     h0(0) = hrhyne
     h(1) = hrhyne
     h0(1) = hrhyne
```

' Estimate heads at end of time interval by implicit method

```
f0 = f(k, Alpha, RAlpha, Surface, Floor, Gamma, RGamma)
```

```
FOR j = 1 TO NNodes
h0(j) = h(j)
```

' Calculate B & G arrays for each of 5 regions

```
f1 = f(k, Alpha, RAlpha, Surface, h(j - 1), Gamma, RGamma) - f0
f2 = f(k, Alpha, RAlpha, Surface, h(j), Gamma, RGamma) - f0
f3 = f(k, Alpha, RAlpha, Surface, h(j + 1), Gamma, RGamma) - f0

     z = 10 * (h0(j) - Surface)
     IF z > 1 THEN
          z = 1
          S = 1
     ELSE
          IF z < 0 THEN
```

```
                    S = .01 * S0 * EXP(-Beta * (Surface - h0(j)))
                ELSE
                    S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
                END IF
            END IF

' Assign node to one of five regions

Region = j
IF j > 3 AND j < NNodes THEN Region = 4
IF j = NNodes THEN Region = 5

Rho = 4 * Dx% * Dx% / Dt

SELECT CASE Region
        CASE 1
                AA = 0: BB = 1: CC = 0: DD = hrhyne
        CASE 2
                AA = -r * (f1 + 4 * f2 - f3)
                BB = 8 * r * f2 + Rho * S
                CC = -r * (-f1 + 4 * f2 + f3)
                DD = Rho * (S * h0(j) + q * Dt)
        CASE 3
                AA = -(r * f1 + 2 * (1 + r) * f2 - f3)
                BB = 4 * (1 + r) * f2 + Rho * S
                CC = -(-r * f1 + 2 * (1 + r) * f2 + f3)
                DD = Rho * (S * h0(j) + q * Dt)
        CASE 4
                AA = -(f1 + 4 * f2 - f3)
                BB = 8 * f2 + Rho * S
                CC = -(-f1 + 4 * f2 + f3)
                DD = Rho * (S * h0(j) + q * Dt)
        CASE 5
                AA = -8 * f2
                BB = 8 * f2 + Rho * S
                CC = 0
                DD = Rho * (S * h0(j) + q * Dt)
END SELECT

W = BB - AA * B(j - 1)
B(j) = CC / W
g(j) = (DD - AA * g(j - 1)) / W
NEXT j

' Solve linear system of equations for heads

h(NNodes) = g(NNodes)
FOR n = NNodes - 1 TO 1 STEP -1
        h(n) = g(n) - B(n) * h(n + 1)
NEXT n

' Print results to screen and file
' Calculate objective function ObjFun

IF Day = DayDw THEN
        FOR i = 1 TO 5
                Deviation = h(jDw(i)) - Dw(i)
                ObjFun = ObjFun + Deviation * Deviation
        NEXT i
END IF

' Re-plot grid for section plot
```

```
PUT (0, 0), Grid%, PSET

'Plot rhyne water level

LINE (30, 60 - (h(1) - 3.8) * 50)-(50, 60 - (h(1) - 3.8) * 50)

' Plot water table

FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(j) - 3.8) * 50), col11, , &HCCCC
NEXT j

' Plot observed dipwell levels on section plot

IF Day = DayDw THEN
  FOR i = 1 TO 5
    CIRCLE (50 + 200 * (jDw(i) - 1) * Dx% / 31, 60 - (Dw(i) - 3.8) * 50), 4, col4
  NEXT i
END IF

' Update maximum and minimum values

FOR j = 1 TO NNodes
    IF h(j) > hMax(j) THEN
        hMax(j) = h(j)
    ELSE
        IF h(j) < hMin(j) THEN
            hMin(j) = h(j)
        END IF
    END IF
NEXT j

' Plot water level at field end dipwell on annual plot

LINE (Istep, 125 - (h0(jDw(5)) - 3.8) * 50)-(Istep, 125 - (h(jDw(5)) - 3.8) * 50),
col15

' Plot observed levels at field end dipwell on annual plot
' Update sum of squares for computation of objective function

IF Day = DayDw THEN
    CIRCLE (Istep, 125 - (Dw(5) - 3.8) * 50), 4, col4
    IF NOT EOF(3) THEN
        INPUT #3, DayDw
        NObs = NObs + 5
        FOR j = 1 TO 5
            INPUT #3, Dw(j)
            Sum = Sum + Dw(j)
            SumSq = SumSq + Dw(j) ^ 2
        NEXT j
    END IF
END IF

' Print footer information panel and output predicted dipwell levels to file

LOCATE 22, 2
PRINT USING " & ### & #.### & ##.# & ##.# &"; "Day:"; Time; "  Rhyne
level"; hrhyne; "mOD.   Rainfall "; Rain; "mm. Evapn "; Evap; "mm."
LOCATE 23, 2
PRINT USING " & #####"; "Output to file A:" + File$ + "OUT.DAT: "; Day;
FOR i = 1 TO 5
    PRINT USING " #.###"; h(jDw(i));
NEXT i: PRINT
PRINT #2, USING "#####"; Day;
```

```
FOR i = 1 TO 5
    PRINT #2, USING "  #.###"; h(jDw(i));
NEXT i: PRINT #2,
NEXT Istep
```

' Plot maximum and minimum water table levels on section plot

```
LINE (50, Top)-(50, 50), col6
LINE (50, Top)-(500, Top), col6
LINE (30, 60 - (hMax(1) - 3.8) * 50)-(50, 60 - (hMax(1) - 3.8) * 50), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (hMax(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - (hMin(1) - 3.8) * 50)-(50, 60 - (hMin(1) - 3.8) * 50), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (hMin(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - (h(1) - 3.8) * 50)-(50, 60 - (h(1) - 3.8) * 50), col11
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(j) - 3.8) * 50), col11, , &HCCCC
NEXT j
```

' Pause to allow PrtSc command. Resume on keypress.

CALL Pause

' Final screen displays objective function

```
SCREEN 0: COLOR , 1: CLS : COLOR 11, 1
CALL ScreenBorder
LOCATE 4, 15: PRINT "1-dimensional mire transect model": COLOR 3, 1
LOCATE 10, 15: PRINT "Input file  A:"; File$; ".DAT"
SumSq = SumSq - Sum * Sum / NObs
LOCATE 15, 15: PRINT "Objective function"; 100 * ObjFun / SumSq; "%"
LOCATE 17, 20
PRINT "i.e."; 100 * (1 - ObjFun / SumSq); "% of variance accounted for by
model"
```

' Pause to allow PrtSc command. Return to DOS on keypress.

```
CALL Pause
END
```

' Error handling routine to cope with CGA or mono screen
' Called when Screen 8 executed

```
CGA:
    SCREEN 2
    col3 = 1
    col4 = 1
    col6 = 1
    col9 = 1
    col11 = 1
    col15 = 1
    RESUME NEXT

FUNCTION f (k, Alpha, RAlpha, Surface, z, Gamma, RGamma)
IF z <= Surface THEN
    f = k * EXP(-Alpha * (Surface - z)) * RAlpha
ELSE
```

```
        f = ((k * RGamma) * (EXP(Gamma * (z - Surface)) - 1) + k * RAlpha)
END IF
END FUNCTION

SUB Pause

' Awaits key input before end of execution. This allows screen dump on PC
' using Prtsc.

WHILE INKEY$ = ""
WEND

END SUB

SUB ScreenBorder
COLOR 4
LOCATE 1, 1: PRINT CHR$(201);
FOR l = 2 TO 78: PRINT CHR$(205); : NEXT l
PRINT CHR$(187);
FOR l = 2 TO 24: LOCATE l, 79: PRINT CHR$(186); : NEXT l
LOCATE 25, 79: PRINT CHR$(188);
FOR l = 78 TO 2 STEP -1: LOCATE 25, l: PRINT CHR$(205); : NEXT l
LOCATE 25, 1: PRINT CHR$(200);
FOR l = 24 TO 2 STEP -1: LOCATE l, 1: PRINT CHR$(186); : NEXT l
END SUB
```

4.3 Simple 2-dimensional model - 2d-simp.bas

```
DECLARE SUB pause ()
DECLARE SUB ScreenBorder ()
DECLARE FUNCTION f! (k!, Surface!, z!, Gamma!, RGamma)
DECLARE SUB FindRegion (i, j, Nr, Nc, Region)


'     2-dimensional groundwater simulation program
'     ============================================

' This version includes an impermeable base at an elevation Floor mOD
' and vertically uniform permeability and specific yield.

' Poor hydraulic connection with rhynes simulated by low permeability
' panel.

'     K Gilman, May 1991


' The field is represented as a rectangular area bounded by rhynes.
' There are Nc columns and Nr rows. First column (j = 1) and first row
' (i = 1) represent rhyne boundaries, last column (j = Nc) and last row
' (i = Nr) represent axes of symmetry of field.
' Saturated thickness from water table to underlying impermeable boundary
' at a level of Floor (mOD) defines transmissivity.
' Ground surface simulated by
' specific yield rising to 100% and much larger permeability.

' Rhyne level hrhyne (mOD) is forcing function, infiltration q (mm) is
' distributed uniformly in horizontal direction. These inputs are read
' from a data file at each time step in order day number, rainfall,
' evaporation and rhyne water level.
```

' Time interval 1 day. Sub-increment Dt=0.5 day.
' Shorter sub-increment of 0.25 day when water level above surface.
' Grid interval Dx% = 5 metres.
' x direction is along transect, y direction perpendicular to transect.

' Definition of variables

' h0(i,j) ----- Heads at start of time increment (m)
' h1(i,j) ----- Heads at end of sub-increment (m)
' hprime(i,j) -- Heads computed for end of sub-increment
' h(i,j) ----- Heads at end of time increment (m)
' hMax(j) ----- Greatest head met so far at node (iDw,j)
' hMin(j) ----- Lowest head met so far at node (iDw,j)
' S   ----- Specific yield (entered as %, used as fraction in program)
' q   ----- Net infiltration rate (mm)
' k   ----- Permeability (m/d)
' Grid% ----- Array for storage of grid for video display

DIM h(45, 45), B(45), g(45), h0(45, 45), h1(45, 45), Grid%(8000)
DIM hprime(45, 45), hMax(45), hMin(45)
     Nsteps = 365
     Dt = .5
     Nsubs = 1
     Dx% = 5

' Setup screen, input of parameters.
' Rhyne factor r is ratio of permeabilities left and right of j=3, above
' and below i=3. Low permeability panel between j=1 and j=3, i=1 and i=3
' permeability panel between j=0 and j=3 is to take account of sealing
' takes account of sealing, compaction of banks, and shallowness of rhyne.
' Field width and length are between parallel rhynes.

' First screen for input of parameters

SCREEN 0: COLOR , 1: CLS

CALL ScreenBorder

COLOR 14, 1: LOCATE 3, 15: PRINT "COMPASS - COM";
COLOR 12, 1: PRINT "partmented "; : COLOR 14, 1: PRINT "P";
COLOR 12, 1: PRINT "hreatic "; : COLOR 14, 1: PRINT "A";
COLOR 12, 1: PRINT "quifer "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "imulation "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "ystem "
LOCATE 5, 15: PRINT "Two-dimensional mire transect model": COLOR 3, 1
LOCATE 7, 15: PRINT "Permeability: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT " m/d"
LOCATE 11, 15: PRINT "Spec yield: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT " %"
LOCATE 15, 15: PRINT "Rhyne factor: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1
LOCATE 17, 15: PRINT "Base level: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT "mOD (Base of peat)"
LOCATE 19, 15: PRINT "Surface level: ";
COLOR 3, 3: PRINT "   ";
COLOR 3, 1: PRINT "mOD"
LOCATE 21, 15: PRINT "Field width: ";

```
COLOR 3, 3: PRINT "    ";
COLOR 3, 1: PRINT "m"
LOCATE 21, 40: PRINT "Length:       ";
COLOR 3, 3: PRINT "    ";
COLOR 3, 1: PRINT "m"
LOCATE 23, 15: PRINT "Filename:   A:";
COLOR 3, 3: PRINT "    ";
COLOR 3, 1: PRINT ".DAT"
COLOR 1, 3
LOCATE 7, 30: INPUT " ", k
LOCATE 11, 30: INPUT " ", S0
LOCATE 15, 30: INPUT " ", r
LOCATE 17, 30: INPUT " ", Floor
LOCATE 19, 30: INPUT " ", Surface
LOCATE 21, 30: INPUT " ", FieldWidth
LOCATE 21, 54: INPUT " ", FieldLength
LOCATE 23, 30: INPUT " ", File$
```

' Compute Nc and Nr from width and length of field and grid interval
' Select Gamma, parameter for above-ground flows.

```
Nc = CINT(.5 * FieldWidth / Dx%) + 1
Nr = CINT(.5 * FieldLength / Dx%) + 1
Er = .00005 * Nc * Nr
Gamma = 25
RGamma = 1 / Gamma
```

' Open input and output files - #1 for climate data, #3 for dipwell data

```
OPEN ("a:" + File$ + ".dat") FOR INPUT AS #1
OPEN ("a:" + File$ + "out.dat") FOR OUTPUT AS #2
OPEN ("a:" + File$ + "dw.dat") FOR INPUT AS #3
```

'Input coordinates of dipwells from #3

```
FOR j = 1 TO 5
    INPUT #3, xDw(j)
    jdw(j) = 1 + CINT(xDw(j) / Dx%)
NEXT j
INPUT #3, yDw
iDw = 1 + CINT(yDw / Dx%)
```

' Input first data from dipwell file #3
' Set up starting value for h array

```
INPUT #3, DayDw
    NObs = NObs + 5
    FOR j = 1 TO 5
        INPUT #3, Dw(j)
        Sum = Sum + Dw(j)
        SumSq = SumSq + Dw(j) ^ 2
    NEXT j
    hStart = (Sum - Dw(1)) / 4
```

' Set up graphics window for plotting. Program tests for VGA or CGA
' Error handling routine CGA at end of program resets colours for
' CGA or mono if necessary.

```
ON ERROR GOTO CGA
CLS
        col3 = 3            ' cyan
        col4 = 4            ' red
        col6 = 6            ' brown
        col9 = 9            ' light blue
        col11 = 11          ' light cyan
        col15 = 15          ' bright white
        SCREEN 8
        COLOR 3
VIEW (15, 5)-(590, 160), 0, col9
VIEW (50, 20)-(550, 145)
WINDOW SCREEN (0, 0)-(500, 125)
LINE (0, 0)-(500, 60), col9, B
LINE (0, 65)-(365, 125), col9, B
LOCATE 1, 27
PRINT " Distance from rhyne, m "
LOCATE 2, 11
IF Nc > 1 + 70 / Dx% THEN Npts = 1 + 70 / Dx% ELSE Npts = Nc

' Labels for x axis along top of section plot

FOR j = 1 TO Npts STEP 10 / Dx%
        PRINT USING " ###    "; (j - 1) * Dx%;
NEXT j

' Grid for section plot

Top = 60 - 50 * (Surface - 3.8)
LINE (50, Top)-(50, 60), col6
FOR j = 2 TO Npts
IF (j * Dx% MOD 10) = Dx% THEN LineStyle = &HFFFF ELSE LineStyle =
&HF0F0
LINE (50 + 200 * (j - 1) * Dx% / 31, Top)-(50 + 200 * (j - 1) * Dx% / 31, 60),
col9, , LineStyle
NEXT j
LINE (50, Top)-(500, Top), col6
LOCATE 10, 9: PRINT "Section along transect"
GET (0, 0)-(500, 60), Grid%

' Grid for annual plot

LINE (0, 65 + Top)-(365, 65 + Top), col6
LINE (31, 65)-(31, 125), col9, , &HF0F0
LINE (59, 65)-(59, 125), col9, , &HF0F0
LINE (90, 65)-(90, 125), col9, , &HF0F0
LINE (120, 65)-(120, 125), col9, , &HF0F0
LINE (151, 65)-(151, 125), col9, , &HF0F0
LINE (181, 65)-(181, 125), col9, , &HF0F0
LINE (212, 65)-(212, 125), col9, , &HF0F0
LINE (243, 65)-(243, 125), col9, , &HF0F0
LINE (273, 65)-(273, 125), col9, , &HF0F0
LINE (304, 65)-(304, 125), col9, , &HF0F0
LINE (335, 65)-(335, 125), col9, , &HF0F0

' Labels for time axis of annual plot

LOCATE 20, 9: PRINT "J  F  M  A  M  J  J  A  S  O  N  D"

' Information panel
```

```
LOCATE 12, 54: PRINT "2-DIM MODEL"
LOCATE 14, 54: PRINT "Permeability (m/d):"
LOCATE 15, 54: PRINT USING "#.## "; k
LOCATE 16, 54: PRINT "Spec yield (%):"
LOCATE 17, 54: PRINT USING "##.# "; S0
LOCATE 18, 54: PRINT "Rhyne factor:"; r
LOCATE 17, 9: PRINT "Annual plot"
LOCATE 18, 9: PRINT "Node ("; iDw; ","; jdw(5); ")"
LOCATE 19, 54: PRINT "Base level:"; Floor; "mOD"
LOCATE 20, 54: PRINT "Surface:"; Surface; "mOD"

' Fill h array with starting value

FOR i = 1 TO Nr
     FOR j = 1 TO Nc
          h(i, j) = hStart
          hMax(j) = Floor
          hMin(j) = Surface
     NEXT j
NEXT i

' Start of simulation

Time = 0
ObjFun = 0
f0 = f(k, Surface, Floor, Gamma, RGamma)
FOR Istep = 1 TO Nsteps

' Input climate and rhyne level data from #1

INPUT #1, Day, Rain, Evap, hrhyne
q = (Rain - .95 * Evap) * .001
     FOR i = 1 TO Nr
          h(i, 1) = hrhyne
          h0(i, 1) = hrhyne
          h1(i, 1) = hrhyne
     NEXT i

     FOR j = 1 TO Nc
          h(1, j) = hrhyne
          h0(1, j) = hrhyne
          h1(1, j) = hrhyne
     NEXT j

' Refine estimates of heads by IADI method
' Use 2 sub-increments to improve convergence if water above surface

IF h0(Nr, Nc) >= Surface THEN
     Dt = .25
     Nsubs = 2
ELSE
     Dt = .5
     Nsubs = 1
END IF
Time = Time + 2 * Nsubs * Dt
FOR Isub = 1 TO Nsubs
Iter = 0
DO
E = 0
Iter = Iter + 1

' Predict future heads by simple SMD model for 1st iteration only
```

```
IF Iter = 1 THEN
    FOR i = 2 TO Nr
        FOR j = 2 TO Nc
            h0(i, j) = h(i, j)
            h1(i, j) = h0(i, j) + q * Dt * 100 / S0
        NEXT j
    NEXT i
END IF

' Column calculations

FOR j = Nc TO 1 STEP -1
FOR i = 1 TO Nr

' Calculate B & G arrays for each of 13 regions

f1 = f(k, Surface, h0(i, j - 1), Gamma, RGamma) - f0
f2 = f(k, Surface, h1(i - 1, j), Gamma, RGamma) - f0
f3 = f(k, Surface, h0(i, j + 1), Gamma, RGamma) - f0
f4 = f(k, Surface, h1(i + 1, j), Gamma, RGamma) - f0
f5 = f(k, Surface, h0(i, j), Gamma, RGamma) - f0
f6 = f(k, Surface, h1(i, j), Gamma, RGamma) - f0

    z = 10 * (h0(i, j) - Surface)
    IF z > 1 THEN
        S = 1
    ELSE
        IF z < 0 THEN
            S = .01 * S0
        ELSE
            S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
        END IF
    END IF

' Assign node to one of 13 regions

CALL FindRegion(i, j, Nr, Nc, Region)

Rho = 4 * Dx% * Dx% / Dt

SELECT CASE Region
    CASE 1
        AA = 0: BB = 1: CC = 0: DD = hrhyne
    CASE 2
        AA = -r * (f2 + 4 * f6 - f4)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h0(i, j) + q * Dt)
        DD = DD + r * (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))
        DD = DD + 4 * r * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))
    CASE 3
        AA = -r * (f2 + 4 * f6 - f4)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h0(i, j) + q * Dt)
        DD = DD + 8 * r * f5 * (h0(i, j - 1) - h0(i, j))
    CASE 4
        AA = -8 * r * f6
        BB = 8 * r * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h0(i, j) + q * Dt)
```

$$DD = DD + r * (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * r * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 5
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 6
$$AA = -8 * f6$$
$$BB = 8 * f6 + Rho * S$$
$$CC = 0$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 7
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))$$
CASE 8
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 9
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 10
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 11
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))$$
CASE 12
$$AA = -8 * f6$$
$$BB = 8 * f6 + Rho * S$$
$$CC = 0$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 13

```
                AA = -8 * f6
                BB = 8 * f6 + Rho * S
                CC = 0
                DD = Rho * (S * h0(i, j) + q * Dt)
                DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))
        END SELECT

        W = BB - AA * B(i - 1)
        B(i) = CC / W
        g(i) = (DD - AA * g(i - 1)) / W
        NEXT i
```

' Solve linear system of equations for heads

```
        hprime(Nr, j) = g(Nr)
        FOR n = Nr - 1 TO 1 STEP -1
                Ha = g(n) - B(n) * hprime(n + 1, j)
                hprime(n, j) = Ha
        NEXT n
        NEXT j
```

' Modify prediction of h(i,j) using results of first sub-increment
' for first iteration step only.

```
        FOR i = 2 TO Nr
                FOR j = 2 TO Nc
                        h1(i, j) = hprime(i, j)
                        IF Iter = 1 THEN
                                h(i, j) = 2 * h1(i, j) - h0(i, j)
                        END IF
                NEXT j
        NEXT i
```

' Row calculations

```
        FOR i = Nr TO 1 STEP -1
        FOR j = 1 TO Nc
```

' Calculate B & G arrays for each of 13 regions

```
        f1 = f(k, Surface, h(i, j - 1), Gamma, RGamma) - f0
        f2 = f(k, Surface, h1(i - 1, j), Gamma, RGamma) - f0
        f3 = f(k, Surface, h(i, j + 1), Gamma, RGamma) - f0
        f4 = f(k, Surface, h1(i + 1, j), Gamma, RGamma) - f0
        f5 = f(k, Surface, h1(i, j), Gamma, RGamma) - f0
        f6 = f(k, Surface, h(i, j), Gamma, RGamma) - f0

                z = 10 * (h1(i, j) - Surface)
                IF z > 1 THEN
                        S = 1
                ELSE
                        IF z < 0 THEN
                                S = .01 * S0
                        ELSE
                                S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
                        END IF
                END IF
```

' Assign node to one of 13 regions

```
        CALL FindRegion(i, j, Nr, Nc, Region)
```

```
SELECT CASE Region
    CASE 1
        AA = 0: BB = 1: CC = 0: DD = hrhyne
    CASE 2
        AA = -r * (f1 + 4 * f6 - f3)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + r * (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * r * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 3
        AA = -8 * r * f6
        BB = 8 * r * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + r * (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * r * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 4
        AA = -r * (f1 + 4 * f6 - f3)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + 8 * r * f5 * (h1(i - 1, j) - h1(i, j))
    CASE 5
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 6
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
    CASE 7
        AA = -8 * f6
        BB = 8 * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 8
        AA = -(f1 + 4 * f6 - f3)
        BB = 8 * f6 + Rho * S
        CC = -AA - 8 * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 9
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
```

```
        CASE 10
            AA = -(f1 + 4 * f6 - f3)
            BB = 8 * f6 + Rho * S
            CC = -AA - 8 * f6
            DD = Rho * (S * h1(i, j) + q * Dt)
            DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
            DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
        CASE 11
            AA = -8 * f6
            BB = 8 * f6 + Rho * S
            CC = 0
            DD = Rho * (S * h1(i, j) + q * Dt)
            DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
            DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
        CASE 12
            AA = -(f1 + 4 * f6 - f3)
            BB = 8 * f6 + Rho * S
            CC = -AA - 8 * f6
            DD = Rho * (S * h1(i, j) + q * Dt)
            DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
        CASE 13
            AA = -8 * f6
            BB = 8 * f6 + Rho * S
            CC = 0
            DD = Rho * (S * h1(i, j) + q * Dt)
            DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
END SELECT

W = BB - AA * B(j - 1)
B(j) = CC / W
g(j) = (DD - AA * g(j - 1)) / W
NEXT j

' Solve linear system of equations for heads

E = E + ABS(h(i, Nc) - g(Nc))
hprime(i, Nc) = g(Nc)
FOR n = Nc - 1 TO 1 STEP -1
    Ha = g(n) - B(n) * hprime(i, n + 1)
    E = E + ABS(Ha - h(i, n))
    hprime(i, n) = Ha
NEXT n
NEXT i
FOR i = 2 TO Nr
    FOR j = 2 TO Nc
        h(i, j) = hprime(i, j)
    NEXT j
NEXT i

' Print iteration number and error as check on convergence

LOCATE 24, 15
PRINT USING " & ## & ##.####"; "Iteration:"; Iter; "   Mean iteration error:";
E / (Nc * Nr);
LOOP WHILE E > Er AND Iter < 10
NEXT Isub

' Print results to screen and file
' Calculate objective function ObjFun
```

```
IF Day = DayDw THEN
     FOR i = 1 TO 5
          Deviation = h(iDw, jdw(i)) - Dw(i)
          ObjFun = ObjFun + Deviation * Deviation
     NEXT i
END IF
```

' Re-plot grid for section plot

```
PUT (0, 0), Grid%, PSET
```

' Plot rhyne water level

```
LINE (30, 60 - (h(iDw, 1) - 3.8) * 50)-(50, 60 - (h(iDw, 1) - 3.8) * 50)
```

' Plot water table

```
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(iDw, j) - 3.8) * 50), col11, , &HCCCC
NEXT j
```

' Plot observed dipwell levels on section plot

```
IF Day = DayDw THEN
     FOR i = 1 TO 5
       CIRCLE (50 + 200 * xDw(i) / 31, 60 - (Dw(i) - 3.8) * 50), 4, col4
     NEXT i
END IF
```

' Update maximum and minimum values

```
FOR j = 1 TO Npts
     IF h(iDw, j) > hMax(j) THEN
          hMax(j) = h(iDw, j)
     ELSE
          IF h(iDw, j) < hMin(j) THEN
               hMin(j) = h(iDw, j)
          END IF
     END IF
NEXT j
```

' Plot water level at field end dipwell on annual plot

```
LINE (Istep - 1, 125 - (h0(iDw, jdw(5)) - 3.8) * 50)-(Istep, 125 - (h(iDw, jdw(5))
- 3.8) * 50), col15
```

' Plot observed levels at field end dipwell on annual plot
' Update sum of squares for computation of objective function

```
IF Day = DayDw THEN
     CIRCLE (Istep, 125 - (Dw(5) - 3.8) * 50), 4, col4
     IF NOT EOF(3) THEN
          INPUT #3, DayDw
          NObs = NObs + 5
          FOR j = 1 TO 5
               INPUT #3, Dw(j)
               Sum = Sum + Dw(j)
               SumSq = SumSq + Dw(j) ^ 2
          NEXT j
     END IF
END IF
```

' Print footer information panel and output predicted dipwell levels to file

```
LOCATE 22, 1
PRINT USING " & ### & #.### & ##.# & ##.# &"; "Day:"; Time; "  Rhyne
level"; hrhyne; "mOD.   Rainfall "; Rain; "mm. Evapn "; Evap; "mm."
PRINT USING " & #####"; "Output to file A:" + File$ + "OUT.DAT: "; Day;
FOR i = 1 TO 5
    PRINT USING " #.###"; h(iDw, jdw(i));
NEXT i
PRINT
PRINT #2, USING "#####"; Day;
FOR i = 1 TO 5
PRINT #2, USING " #.###"; h(iDw, jdw(i));
NEXT i
PRINT #2,

NEXT Istep

' Plot maximum and minimum water table levels on section plot

LINE (50, Top)-(50, 50), col6
LINE (50, Top)-(500, Top), col6
LINE (30, 60 - 50 * (hMax(1) - 3.8))-(50, 60 - 50 * (hMax(1) - 3.8)), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - 50 * (hMax(j) - 3.8)), col15
NEXT j
LINE (30, 60 - 50 * (hMin(1) - 3.8))-(50, 60 - 50 * (hMin(1) - 3.8)), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - 50 * (hMin(j) - 3.8)), col15
NEXT j
LINE (30, 60 - 50 * (h(iDw, 1) - 3.8))-(50, 60 - 50 * (h(iDw, 1) - 3.8)), col11
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - 50 * (h(iDw, j) - 3.8)), col15, , &HCCCC
NEXT j

' Pause to allow PrtSc command. Resume on keypress.

CALL pause

' Final screen displays objective function

SCREEN 0: COLOR , 1: CLS : COLOR 11, 1
CALL ScreenBorder
LOCATE 4, 15: PRINT "2-dimensional mire transect model": COLOR 3, 1
LOCATE 10, 15: PRINT "Input file a:"; File$; ".dat"
SumSq = SumSq - Sum * Sum / NObs
LOCATE 15, 15: PRINT "Objective function"; 100 * ObjFun / SumSq; "%"
LOCATE 17, 20
PRINT "i.e."; 100 * (1 - ObjFun / SumSq); "% of variance accounted for by
model"

' Pause to allow PrtSc command. Return to DOS on keypress.

CALL pause
END

' Error handling routine to cope with CGA screen
' Called when SCREEN 8 executed

CGA:
    SCREEN 2
    col3 = 1
    col4 = 1
    col6 = 1
```

```
                    col9 = 1
                    col11 = 1
                    col15 = 1
                    RESUME NEXT
      FUNCTION f (k, Surface, z, Gamma, RGamma)
          IF z <= Surface THEN
            f = k * z
          ELSE
            f = (k * RGamma) * (EXP(Gamma * (z - Surface)) - 1) + k * Surface
          END IF
      END FUNCTION

      SUB FindRegion (i, j, Nr, Nc, Region)
      IF i = 1 OR j = 1 THEN
          Region = 1
      ELSEIF i = 2 THEN
          IF j = Nc THEN
              Region = 3
          ELSE
              Region = 2
          END IF
      ELSEIF i = 3 THEN
          IF j = 2 THEN
              Region = 2
          ELSEIF j = 3 THEN
              Region = 9
          ELSEIF j = Nc THEN
              Region = 7
          ELSE
              Region = 8
          END IF
      ELSEIF i = Nr THEN
          IF j = 2 THEN
              Region = 4
          ELSEIF j = 3 THEN
              Region = 6
          ELSEIF j = Nc THEN
              Region = 13
          ELSE
              Region = 12
          END IF
      ELSE
          IF j = Nc THEN
              Region = 11
          ELSEIF j = 2 THEN
              Region = 2
          ELSEIF j = 3 THEN
              Region = 5
          ELSE
              Region = 10
          END IF
      END IF
      END SUB

      SUB pause

      ' Awaits key input before end of execution. This allows screen dump on PC
      ' using Prtsc.
```

```
WHILE INKEY$ = ""
WEND
END SUB

SUB ScreenBorder
COLOR 4
LOCATE 1, 1: PRINT CHR$(201);
FOR l = 2 TO 78: PRINT CHR$(205); : NEXT l
PRINT CHR$(187);
FOR l = 2 TO 24: LOCATE l, 79: PRINT CHR$(186); : NEXT l
LOCATE 25, 79: PRINT CHR$(188);
FOR l = 78 TO 2 STEP -1: LOCATE 25, l: PRINT CHR$(205); : NEXT l
LOCATE 25, 1: PRINT CHR$(200);
FOR l = 24 TO 2 STEP -1: LOCATE l, 1: PRINT CHR$(186); : NEXT l
END SUB
```

4.4 Complex 2-dimensional model - 2d-comp.bas

```
DECLARE SUB pause ()
DECLARE SUB ScreenBorder ()
DECLARE FUNCTION f! (k!, Alpha!, RAlpha, Surface!, z!, Gamma!, RGamma)
DECLARE SUB FindRegion (i, j, Nr, Nc, Region)


'       2-dimensional groundwater simulation program
'       ================================================

' This version includes an impermeable base at an elevation Floor mOD
' and exponential variation in both permeability and specific yield.

' Poor hydraulic connection with rhynes simulated by low permeability
' panel.

'       K Gilman, May 1991


' The field is represented as a rectangular area bounded by rhynes.
' There are Nc columns and Nr rows. First column (j = 1) and first row
' (i = 1) represent rhyne boundaries, last column (j = Nc) and last row
' (i = Nr) represent axes of symmetry of field.
' Saturated thickness from water table to underlying impermeable boundary
' at a level of Floor (mOD) defines transmissivity. Permeability and
' spec. yield may vary exponentially with depth. Ground surface simulated by
' specific yield rising to 100% and much larger permeability.

' Rhyne level hrhyne (mOD) is forcing function, infiltration q (mm) is
' distributed uniformly in horizontal direction. These inputs are read
' from a data file at each time step in order day number, rainfall,
' evaporation and rhyne water level.

' Time interval 1 day. Sub-increment Dt=0.5 day.
' Shorter sub-increment of 0.25 day when water level above surface
' Grid interval Dx% = 5 metres.
' x direction is along transect, y direction perpendicular to transect.

' Definition of variables
```

```
' h0(i,j)  ----- Heads at start of time increment (m)
' h1(i,j)  ----- Heads at end of sub-increment (m)
' hprime(i,j)  -- Dummy array of heads computed for end of sub-increment
' h(i,j)  ----- Heads at end of time increment (m)
' hMax(j)  ----- Greatest head met so far at node (iDw,j)
' hMin(j)  ----- Lowest head met so far at node (iDw,j)
' S       ----- Specific yield (entered as %, used as fraction in program)
' q       ----- Net infiltration rate (mm)
' k       ----- Permeability (m/d)
' Grid%   ----- Array for storage of grid for video display

DIM h(45, 45), B(45), g(45), h0(45, 45), h1(45, 45), Grid%(8000)
DIM hprime(45, 45), hMax(45), hMin(45)
      Nsteps = 365
      Dt = .5
      Nsubs = 1
      Dx% = 5

' Setup screen, input of parameters. k and S defined as product of
' value at surface and exponential function of depth below surface.
' Rhyne factor r is ratio of permeabilities left and right of j=3, above
' and below i=3. Low permeability panel between j=1 and j=3, i=1 and i=3
' permeability panel between j=0 and j=3 is to take account of sealing
' takes account of sealing, compaction of banks, and shallowness of rhyne.
' Field width and length are between parallel rhynes.

' First screen for input of parameters

SCREEN 0: COLOR , 1: CLS

CALL ScreenBorder

COLOR 14, 1: LOCATE 3, 15: PRINT "COMPASS - COM";
COLOR 12, 1: PRINT "partmented "; : COLOR 14, 1: PRINT "P";
COLOR 12, 1: PRINT "hreatic "; : COLOR 14, 1: PRINT "A";
COLOR 12, 1: PRINT "quifer "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "imulation "; : COLOR 14, 1: PRINT "S";
COLOR 12, 1: PRINT "ystem "
LOCATE 5, 15: PRINT "Two-dimensional mire transect model": COLOR 3, 1
LOCATE 7, 15: PRINT "Permeability: ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT " m/d"
LOCATE 9, 25: PRINT "Exp coeff:    ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT
LOCATE 11, 15: PRINT "Spec yield:   ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT " %"
LOCATE 13, 25: PRINT "Exp coeff:    ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT
LOCATE 15, 15: PRINT "Rhyne factor: ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1
LOCATE 17, 15: PRINT "Base level:   ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT "mOD  (Base of peat)"
LOCATE 19, 15: PRINT "Surface level: ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT "mOD"
```

```
LOCATE 21, 15: PRINT "Field width:   ";
COLOR 3, 3: PRINT "      ";
COLOR 3, 1: PRINT "m"
LOCATE 21, 40: PRINT "Length:    ` ";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT "m"
LOCATE 23, 15: PRINT "Filename:   A:";
COLOR 3, 3: PRINT "     ";
COLOR 3, 1: PRINT ".DAT"
COLOR 1, 3
LOCATE 7, 30: INPUT " ", k
LOCATE 9, 40: INPUT " ", Alpha
LOCATE 11, 30: INPUT " ", S0
LOCATE 13, 40: INPUT " ", Beta
LOCATE 15, 30: INPUT " ", r
LOCATE 17, 30: INPUT " ", Floor
LOCATE 19, 30: INPUT " ", Surface
LOCATE 21, 30: INPUT " ", FieldWidth
LOCATE 21, 54: INPUT " ", FieldLength
LOCATE 23, 30: INPUT " ", File$

' Compute Nc and Nr from width and length of field and grid interval
' Select Gamma, parameter for above-ground flows.

Nc = CINT(.5 * FieldWidth / Dx%) + 1
Nr = CINT(.5 * FieldLength / Dx%) + 1
Er = .00005 * Nc * Nr
RAlpha = 1 / Alpha
Gamma = 25
RGamma = 1 / Gamma

' Open input and output files - #1 for climate data, #3 for dipwell data

OPEN ("a:" + File$ + ".dat") FOR INPUT AS #1
OPEN ("a:" + File$ + "out.dat") FOR OUTPUT AS #2
OPEN ("a:" + File$ + "dw.dat") FOR INPUT AS #3

' Input coordinates of dipwells from #3

FOR j = 1 TO 5
    INPUT #3, xDw(j)
    jDw(j) = 1 + CINT(xDw(j) / Dx%)
NEXT j
INPUT #3, yDw
iDw = 1 + CINT(yDw / Dx%)

' Input first data from dipwell file#3
' Set up starting value of h array

INPUT #3, DayDw
    NObs = NObs + 5
    FOR j = 1 TO 5
        INPUT #3, Dw(j)
        Sum = Sum + Dw(j)
        SumSq = SumSq + Dw(j) ^ 2
    NEXT j
    hStart = (Sum - Dw(1)) / 4

' Set up graphics window for plotting. Program tests for VGA or EGA
' Error handling routine CGA at end of program resets colours for
' CGA or mono if necessary.
```

```
ON ERROR GOTO CGA
CLS
        col3 = 3            ' cyan
        col4 = 4            ' red
        col6 = 6            ' brown
        col9 = 9            ' light blue
        col11 = 11          ' light cyan
        col15 = 15          ' bright white
        SCREEN 8
        COLOR 3
VIEW (15, 5)-(590, 160), 0, col9
VIEW (50, 20)-(550, 145)
WINDOW SCREEN (0, 0)-(500, 125)
LINE (0, 0)-(500, 60), col9, B
LINE (0, 65)-(365, 125), col9, B
LOCATE 1, 27
PRINT " Distance from rhyne, m "
LOCATE 2, 11
IF Nc > 1 + 70 / Dx% THEN Npts = 1 + 70 / Dx% ELSE Npts = Nc

' Labels for x axis along top of section plot

FOR j = 1 TO Npts STEP 10 / Dx%
        PRINT USING " ###    "; (j - 1) * Dx%;
NEXT j

' Grid for section plot

Top = 60 - 50 * (Surface - 3.8)
LINE (50, Top)-(50, 60), col6
FOR j = 2 TO Npts
IF (j * Dx% MOD 10) = Dx% THEN LineStyle = &HFFFF ELSE LineStyle =
&HF0F0
LINE (50 + 200 * (j - 1) * Dx% / 31, Top)-(50 + 200 * (j - 1) * Dx% / 31, 60),
col9, , LineStyle
NEXT j
LINE (50, Top)-(500, Top), col6
LOCATE 10, 9: PRINT "Section along transect"
GET (0, 0)-(500, 60), Grid%

' Grid for annual plot

LINE (0, 65 + Top)-(365, 65 + Top), col6
LINE (31, 65)-(31, 125), col9, , &HF0F0
LINE (59, 65)-(59, 125), col9, , &HF0F0
LINE (90, 65)-(90, 125), col9, , &HF0F0
LINE (120, 65)-(120, 125), col9, , &HF0F0
LINE (151, 65)-(151, 125), col9, , &HF0F0
LINE (181, 65)-(181, 125), col9, , &HF0F0
LINE (212, 65)-(212, 125), col9, , &HF0F0
LINE (243, 65)-(243, 125), col9, , &HF0F0
LINE (273, 65)-(273, 125), col9, , &HF0F0
LINE (304, 65)-(304, 125), col9, , &HF0F0
LINE (335, 65)-(335, 125), col9, , &HF0F0

' Labels for time axis of annual plot

LOCATE 20, 9: PRINT "J  F  M  A  M  J  J  A  S  O  N  D"

' Information panel
```

```
LOCATE 12, 54: PRINT "2-DIM MODEL"
LOCATE 14, 54: PRINT "Permeability (m/d):"
LOCATE 15, 54: PRINT USING "#.## & #.## &"; k; "exp -"; Alpha; "z"
LOCATE 16, 54: PRINT "Spec yield (%):"
LOCATE 17, 54: PRINT USING "##.# & #.## &"; S0; "exp -"; Beta; "z"
LOCATE 18, 54: PRINT "Rhyne factor:"; r
LOCATE 17, 9: PRINT "Annual plot"
LOCATE 18, 9: PRINT "Node ("; iDw; ","; jDw(5); ")"
LOCATE 19, 54: PRINT "Base level:"; Floor; "mOD"
LOCATE 20, 54: PRINT "Surface:"; Surface; "mOD"

' Fill h array with starting value

FOR i = 1 TO Nr
    FOR j = 1 TO Nc
        h(i, j) = hStart
        hMax(j) = Floor
        hMin(j) = Surface
    NEXT j
NEXT i

' Start of simulation

Time = 0
ObjFun = 0
f0 = f(k, Alpha, RAlpha, Surface, Floor, Gamma, RGamma)
FOR Istep = 1 TO Nsteps

' Input climate and rhyne level data from #1

INPUT #1, Day, Rain, Evap, hrhyne
q = (Rain - .95 * Evap) * .001
    FOR i = 1 TO Nr
        h(i, 1) = hrhyne
        h0(i, 1) = hrhyne
        h1(i, 1) = hrhyne
    NEXT i

    FOR j = 1 TO Nc
        h(1, j) = hrhyne
        h0(1, j) = hrhyne
        h1(1, j) = hrhyne
    NEXT j

' Refine estimates of heads by IADI method
' Use 2 sub-increments to improve convergence if water above surface

IF h0(Nr, Nc) >= Surface THEN
    Dt = .25
    Nsubs = 2
ELSE
    Dt = .5
    Nsubs = 1
END IF
Time = Time + 2 * Nsubs * Dt
FOR Isub = 1 TO Nsubs
Iter = 0
DO
E = 0
Iter = Iter + 1

' Predict future heads by simple SMD model for 1st iteration only
```

```
IF Iter = 1 THEN
    FOR i = 2 TO Nr
        FOR j = 2 TO Nc
            h0(i, j) = h(i, j)
            h1(i, j) = h0(i, j) + q * Dt * 100 / S0
        NEXT j
    NEXT i
END IF

' Column calculations

FOR j = Nc TO 1 STEP -1
FOR i = 1 TO Nr

' Calculate B & G arrays for each of 13 regions

f1 = f(k, Alpha, RAlpha, Surface, h0(i, j - 1), Gamma, RGamma) - f0
f2 = f(k, Alpha, RAlpha, Surface, h1(i - 1, j), Gamma, RGamma) - f0
f3 = f(k, Alpha, RAlpha, Surface, h0(i, j + 1), Gamma, RGamma) - f0
f4 = f(k, Alpha, RAlpha, Surface, h1(i + 1, j), Gamma, RGamma) - f0
f5 = f(k, Alpha, RAlpha, Surface, h0(i, j), Gamma, RGamma) - f0
f6 = f(k, Alpha, RAlpha, Surface, h1(i, j), Gamma, RGamma) - f0

    z = 10 * (h0(i, j) - Surface)
    IF z > 1 THEN
        S = 1
    ELSE
        IF z < 0 THEN
            S = .01 * S0 * EXP(-Beta * (Surface - h0(i, j)))
        ELSE
            S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
        END IF
    END IF

' Assign node to one of 13 regions

CALL FindRegion(i, j, Nr, Nc, Region)

Rho = 4 * Dx% * Dx% / Dt

SELECT CASE Region
    CASE 1
        AA = 0: BB = 1: CC = 0: DD = hrhyne
    CASE 2
        AA = -r * (f2 + 4 * f6 - f4)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h0(i, j) + q * Dt)
        DD = DD + r * (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))
        DD = DD + 4 * r * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))
    CASE 3
        AA = -r * (f2 + 4 * f6 - f4)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        'DD = Rho * (S * h0(i, j) + q * Dt)
        DD = DD + 8 * r * f5 * (h0(i, j - 1) - h0(i, j))
    CASE 4
        AA = -8 * r * f6
        BB = 8 * r * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h0(i, j) + q * Dt)
```

A4.34

$$DD = DD + r * (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * r * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 5
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 6
$$AA = -8 * f6$$
$$BB = 8 * f6 + Rho * S$$
$$CC = 0$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 7
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))$$
CASE 8
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 9
$$AA = -(r * f2 + 2 * (1 + r) * f6 - f4)$$
$$BB = 4 * (1 + r) * f6 + Rho * S$$
$$CC = -AA - 4 * (1 + r) * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - r * f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 2 * (1 + r) * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 10
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 11
$$AA = -(f2 + 4 * f6 - f4)$$
$$BB = 8 * f6 + Rho * S$$
$$CC = -AA - 8 * f6$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))$$
CASE 12
$$AA = -8 * f6$$
$$BB = 8 * f6 + Rho * S$$
$$CC = 0$$
$$DD = Rho * (S * h0(i, j) + q * Dt)$$
$$DD = DD + (f3 - f1) * (h0(i, j + 1) - h0(i, j - 1))$$
$$DD = DD + 4 * f5 * (h0(i, j + 1) - 2 * h0(i, j) + h0(i, j - 1))$$
CASE 13

```
                    AA = -8 * f6
                    BB = 8 * f6 + Rho * S
                    CC = 0
                    DD = Rho * (S * h0(i, j) + q * Dt)
                    DD = DD + 8 * f5 * (h0(i, j - 1) - h0(i, j))
END SELECT

W = BB - AA * B(i - 1)
B(i) = CC / W
g(i) = (DD - AA * g(i - 1)) / W
NEXT i
```

' Solve linear system of equations for heads

```
hprime(Nr, j) = g(Nr)
FOR n = Nr - 1 TO 1 STEP -1
        Ha = g(n) - B(n) * hprime(n + 1, j)
        hprime(n, j) = Ha
NEXT n
NEXT j
```

' Modify prediction of h(i,j) using results of first sub-increment
' for first iteration step only.

```
FOR i = 2 TO Nr
        FOR j = 2 TO Nc
                h1(i, j) = hprime(i, j)
                IF Iter = 1 THEN
                        h(i, j) = 2 * h1(i, j) - h0(i, j)
                END IF
        NEXT j
NEXT i
```

' Row calculations

```
FOR i = Nr TO 1 STEP -1
FOR j = 1 TO Nc
```

' Calculate B & G arrays for each of 13 regions

```
f1 = f(k, Alpha, RAlpha, Surface, h(i, j - 1), Gamma, RGamma) - f0
f2 = f(k, Alpha, RAlpha, Surface, h1(i - 1, j), Gamma, RGamma) - f0
f3 = f(k, Alpha, RAlpha, Surface, h(i, j + 1), Gamma, RGamma) - f0
f4 = f(k, Alpha, RAlpha, Surface, h1(i + 1, j), Gamma, RGamma) - f0
f5 = f(k, Alpha, RAlpha, Surface, h1(i, j), Gamma, RGamma) - f0
f6 = f(k, Alpha, RAlpha, Surface, h(i, j), Gamma, RGamma) - f0

        z = 10 * (h1(i, j) - Surface)
        IF z > 1 THEN
            S = 1
        ELSE
            IF z < 0 THEN
                    S = .01 * S0 * EXP(-Beta * (Surface - h1(i, j)))
            ELSE
                    S = .01 * (S0 + (100 - S0) * z * z * (3 - 2 * z))
            END IF
        END IF
```

' Assign node to one of 13 regions

```
CALL FindRegion(i, j, Nr, Nc, Region)
```

SELECT CASE Region
    CASE 1
        AA = 0: BB = 1: CC = 0: DD = hrhyne
    CASE 2
        AA = -r * (f1 + 4 * f6 - f3)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + r * (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * r * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 3
        AA = -8 * r * f6
        BB = 8 * r * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + r * (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * r * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 4
        AA = -r * (f1 + 4 * f6 - f3)
        BB = 8 * r * f6 + Rho * S
        CC = -AA - 8 * r * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + 8 * r * f5 * (h1(i - 1, j) - h1(i, j))
    CASE 5
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 6
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
    CASE 7
        AA = -8 * f6
        BB = 8 * f6 + Rho * S
        CC = 0
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 8
        AA = -(f1 + 4 * f6 - f3)
        BB = 8 * f6 + Rho * S
        CC = -AA - 8 * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
    CASE 9
        AA = -(r * f1 + 2 * (1 + r) * f6 - f3)
        BB = 4 * (1 + r) * f6 + Rho * S
        CC = -AA - 4 * (1 + r) * f6
        DD = Rho * (S * h1(i, j) + q * Dt)
        DD = DD + (f4 - r * f2) * (h1(i + 1, j) - h1(i - 1, j))
        DD = DD + 2 * (1 + r) * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))

```
        CASE 10
             AA = -(f1 + 4 * f6 - f3)
             BB = 8 * f6 + Rho * S
             CC = -AA - 8 * f6
             DD = Rho * (S * h1(i, j) + q * Dt)
             DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
             DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
        CASE 11
             AA = -8 * f6
             BB = 8 * f6 + Rho * S
             CC = 0
             DD = Rho * (S * h1(i, j) + q * Dt)
             DD = DD + (f4 - f2) * (h1(i + 1, j) - h1(i - 1, j))
             DD = DD + 4 * f5 * (h1(i + 1, j) - 2 * h1(i, j) + h1(i - 1, j))
        CASE 12
             AA = -(f1 + 4 * f6 - f3)
             BB = 8 * f6 + Rho * S
             CC = -AA - 8 * f6
             DD = Rho * (S * h1(i, j) + q * Dt)
             DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
        CASE 13
             AA = -8 * f6
             BB = 8 * f6 + Rho * S
             CC = 0
             DD = Rho * (S * h1(i, j) + q * Dt)
             DD = DD + 8 * f5 * (h1(i - 1, j) - h1(i, j))
END SELECT

W = BB - AA * B(j - 1)
B(j) = CC / W
g(j) = (DD - AA * g(j - 1)) / W
NEXT j

' Solve linear system of equations for heads

E = E + ABS(h(i, Nc) - g(Nc))
hprime(i, Nc) = g(Nc)
FOR n = Nc - 1 TO 1 STEP -1
     Ha = g(n) - B(n) * hprime(i, n + 1)
     E = E + ABS(Ha - h(i, n))
     hprime(i, n) = Ha
NEXT n
NEXT i
FOR i = 2 TO Nr
     FOR j = 2 TO Nc
          h(i, j) = hprime(i, j)
     NEXT j
NEXT i

' Print iteration number and error as check on convergence

LOCATE 24, 15
PRINT USING " & ## & ##.####"; "Iteration:"; Iter; "   Mean iteration error:";
E / (Nc * Nr);
LOOP WHILE E > Er AND Iter < 10
NEXT Isub

' Print results to screen and file
' Calculate objective function ObjFun
```

A4.38

```
IF Day = DayDw THEN
        FOR i = 1 TO 5
                Deviation = h(iDw, jDw(i)) - Dw(i)
                ObjFun = ObjFun + Deviation * Deviation
        NEXT i
END IF
```

' Re-plot grid for section plot

```
PUT (0, 0), Grid%, PSET
```

' Plot rhyne water level

```
LINE (30, 60 - (h(iDw, 1) - 3.8) * 50)-(50, 60 - (h(iDw, 1) - 3.8) * 50)
```

' Plot water table

```
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(iDw, j) - 3.8) * 50), col11, , &HCCCC
NEXT j
```

' Plot observed dipwell levels on section plot

```
IF Day = DayDw THEN
        FOR i = 1 TO 5
          CIRCLE (50 + 200 * xDw(i) / 31, 60 - (Dw(i) - 3.8) * 50), 4, col4
        NEXT i
END IF
```

' Update maximum and minimum values

```
FOR j = 1 TO Npts
        IF h(iDw, j) > hMax(j) THEN
                hMax(j) = h(iDw, j)
        ELSE
                IF h(iDw, j) < hMin(j) THEN
                        hMin(j) = h(iDw, j)
                END IF
        END IF
NEXT j
```

' Plot water level at field end dipwell on annual plot

```
LINE (Istep, 125 - (h0(iDw, jDw(5)) - 3.8) * 50)-(Istep, 125 - (h(iDw, jDw(5)) -
3.8) * 50), col15
```

' Plot observed levels at field end dipwell on annual plot
' Update sum of squares for computation of objective function

```
IF Day = DayDw THEN
        CIRCLE (Istep, 125 - (Dw(5) - 3.8) * 50), 4, col4
        IF NOT EOF(3) THEN
                INPUT #3, DayDw
                NObs = NObs + 5
                FOR j = 1 TO 5
                        INPUT #3, Dw(j)
                        Sum = Sum + Dw(j)
                        SumSq = SumSq + Dw(j) ^ 2
                NEXT j
        END IF
END IF
```

' Print footer information panel and output predicted dipwell levels to file

```
LOCATE 22, 1
PRINT USING " & ### & #.### & ##.# & ##.# &"; "Day:"; Time; "  Rhyne
level"; hrhyne; "mOD.   Rainfall "; Rain; "mm. Evapn "; Evap; "mm."
PRINT USING " & #####"; "Output to file A:" + File$ + "OUT.DAT: "; Day;
FOR i = 1 TO 5
    PRINT USING "  #.###"; h(iDw, jDw(i));
NEXT i
PRINT
PRINT #2, USING "#####"; Day;
FOR i = 1 TO 5
PRINT #2, USING "  #.###"; h(iDw, jDw(i));
NEXT i
PRINT #2,

NEXT Istep
```

'Plot maximum and minimum water table levels on section plot

```
LINE (50, Top)-(50, 50), col6
LINE (50, Top)-(500, Top), col6
LINE (30, 60 - 50 * (hMax(1) - 3.8))-(50, 60 - 50 * (hMax(1) - 3.8)), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) *.Dx% / 31, 60 - (hMax(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - 50 * (hMin(1) - 3.8))-(50, 60 - 50 * (hMin(1) - 3.8)), col15
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (hMin(j) - 3.8) * 50), col15
NEXT j
LINE (30, 60 - 50 * (h(iDw, 1) - 3.8))-(50, 60 - 50 * (h(iDw, 1) - 3.8)), col11
FOR j = 1 TO Npts
LINE -(50 + 200 * (j - 1) * Dx% / 31, 60 - (h(iDw, j) - 3.8) * 50), col11, , &HCCCC
NEXT j
```

' Pause to allow PrtSc command. Resume on keypress.

```
CALL pause
```

' Final screen displays objective function

```
SCREEN 0: COLOR , 1: CLS : COLOR 11, 1
CALL ScreenBorder
LOCATE 4, 15: PRINT "2-dimensional mire transect model": COLOR 3, 1
LOCATE 10, 15: PRINT "Input file  a:"; File$; ".dat"
SumSq = SumSq - Sum * Sum / NObs
LOCATE 15, 15: PRINT "Objective function"; 100 * ObjFun / SumSq; "%"
LOCATE 17, 20
PRINT "i.e."; 100 * (1 - ObjFun / SumSq); "% of variance accounted for by
model"
```

' Pause to allow PrtSc command. Return to DOS on keypress.

```
CALL pause
END
```

' Error handling routine to cope with CGA screen
' Called when SCREEN 8 executed

```
CGA:
    SCREEN 2
    col3 = 1
    col4 = 1
    col6 = 1
```

```
            col9 = 1
            col11 = 1
            col15 = 1
            RESUME NEXT

FUNCTION f (k, Alpha, RAlpha, Surface, z, Gamma, RGamma)
        IF z <= Surface THEN
                f = k * EXP(-Alpha * (Surface - z)) * RAlpha
        ELSE
                f = (k * RGamma) * (EXP(Gamma * (z - Surface)) - 1) + k * RAlpha
        END IF
END FUNCTION

SUB FindRegion (i, j, Nr, Nc, Region)
IF i = 1 OR j = 1 THEN
        Region = 1
ELSEIF i = 2 THEN
        IF j = Nc THEN
                Region = 3
        ELSE
                Region = 2
        END IF
ELSEIF i = 3 THEN
        IF j = 2 THEN
                Region = 2
        ELSEIF j = 3 THEN
                Region = 9
        ELSEIF j = Nc THEN
                Region = 7
        ELSE
                Region = 8
        END IF
ELSEIF i = Nr THEN
        IF j = 2 THEN
                Region = 4
        ELSEIF j = 3 THEN
                Region = 6
        ELSEIF j = Nc THEN
                Region = 13
        ELSE
                Region = 12
        END IF
ELSE
        IF j = Nc THEN
                Region = 11
        ELSEIF j = 2 THEN
                Region = 2
        ELSEIF j = 3 THEN
                Region = 5
        ELSE
                Region = 10
        END IF
END IF
END SUB
```

```
SUB pause
'

' Awaits key input before end of execution. This allows screen dump on PC
' using Prtsc.
'

WHILE INKEY$ = ""
WEND
END SUB

SUB ScreenBorder
COLOR 4
LOCATE 1, 1: PRINT CHR$(201);
FOR l = 2 TO 78: PRINT CHR$(205); : NEXT l
PRINT CHR$(187);
FOR l = 2 TO 24: LOCATE l, 79: PRINT CHR$(186); : NEXT l
LOCATE 25, 79: PRINT CHR$(188);
FOR l = 78 TO 2 STEP -1: LOCATE 25, l: PRINT CHR$(205); : NEXT l
LOCATE 25, 1: PRINT CHR$(200);
FOR l = 24 TO 2 STEP -1: LOCATE l, 1: PRINT CHR$(186); : NEXT l
END SUB
```

Appendix 5 GROUNDWATER MODEL OPERATION MANUAL

The model programs are designed to run on an IBM PC compatible computer. They have been developed on an Epson PC with 640 Kb of memory and a CGA monitor, but will take advantage of a numerical coprocessor (e.g. 8087) chip if fitted, or of VGA graphics.

Three 360Kb 5.25 inch discs are provided: two discs hold the programs in executable (.EXE) form, and the other holds the data files which were used in runs of the model for this report. The programs may be loaded on to, and run from, a hard disc if available, but the data files must be used from a flexible disc on Drive A: of the machine. The output file is also sent to Drive A:, so the disc in that drive must be write enabled and have sufficient space for the output file. The name of the output file is derived from that of the input file named in the initial dialogue: it may be renamed or copied after running the program.

BASIC listings are provided as Appendix 4 of the report. The language used is Microsoft QuickBasic Version 4.5.

5.1 Installation on floppy disc system

Start the computer as normal, and change the active drive to A:

Make a backup copy of each program disc (refer to the operating system manual).

Make a backup copy of the data disc.

These backup copies should be used in all subsequent work, and the original discs should be retained in a safe place in case of loss or damage of the original discs.

5.2 Running the program on floppy disc system

5.2.1 Simple 1-dimensional model

With a copy of Program Disc 1 in Drive A:, type **1D-SIMP** (for simple 1-dimensional model) followed by [Return] or [Enter]. For convenience the [Return] or [Enter] key will be referred to as [Return] from now on.

Remove the Program Disc and insert a copy of the Data Disc.

The first screen is for data entry. Each entry is followed by [Return]. The first entries are the parameters permeability, specific yield and rhyne factor: all must be positive numbers, and the model has not been tested for rhyne factors in excess of unity. At any stage the program can be terminated by [CTRL]/[Break]. The base level and surface level are the lower and upper boundaries of the peat in mOD, and have taken the following values in the work described in the report:

Table A5.1 Values of base level and surface level

Transect	Base level mOD	Surface level mOD
T1	-1.13	4.77
T2	-0.40	4.84
T3	-0.89	4.86
T4	-0.49	4.91

The width of the field is always measured in the direction of the dipwell transect: this convention results in three cases in the somewhat paradoxical situation where the field width exceeds its length. Field dimensions for the four transects (see Section 2.1) are:

Table A5.2 Field dimensions for the four dipwell transects

Transect	Field width m	Field length m
T1	123	225
T2	418	118
T3	253	150
T4	170	144

The last entry on the page is a file name. The data files supplied on the Data Disc are named **T1-87.DAT** etc for the climate and rhyne level files, and **T1-87DW.DAT** etc for the dipwell data files. The climate and rhyne level files have the following structure:

```
31778    2.7    0.1    4.642
31779    0.0    0.1    4.684
31780    0.0    0.2    4.605
31781    0.6    0.1    4.571

..      ..     ..     ..

32142    0.4    0.7    4.445
```

where the first column is a day number (counting from day 0 on 1 Jan 1900, the second column is the daily rainfall at the West Sedgemoor Pumping Station in mm, the third column is the MORECS estimate of potential evaporation obtained from the Meteorological Office, and the last column is a synthesised rhyne level, derived from the records of the nearest IH rhyne level recorder. It was found necessary for the stability of the model to restrict rhyne levels to less than about 5.2 mOD: this is not considered to be a serious restriction on the use of the model as it applies to only a few days' data from certain transects. The data files are stored in ASCII format, and may be edited, typed or printed using conventional PC operating system programs, but it was found helpful to use a spreadsheet program (Lotus 123) to carry out operations such as adding a constant 0.05 m for the simulations of changed water level management (Chapter 6).

The dipwell data files have the following structure:

```
2 12 22 32 52 51.5
31784    4.608    4.734    4.791    4.801    4.826
31812    4.557    4.785    4.789    4.803    4.825
31826    4.505    4.744    4.802    4.811    4.808

..       ..       ..       ..       ..       ..

32127    4.749    4.768    4.760    4.773    4.802
```

In this case the first line of the file carries data on the locations of the dipwells in the transect: the first five numbers are the distances from the rhyne in metres, and the fifth number is the distance in metres from the nearest lateral rhyne. Succeeding lines consist of a day number (see above) and five dipwell water levels, corrected for ground level movement as described in Chapter 2.

The program assumes that dipwell file names are derived from climate data files by adding the suffix "DW".

The output file, carrying predicted daily water levels at the dipwell locations, will be named by adding the suffix "OUT", and saved on the disc in Drive A:. Of course, succeeding runs of the model with the same climate file will overwrite this file, so if it is to be saved in the long term it must be renamed or moved to another disc. The contents of the disc should be checked frequently, to ensure that there is sufficient room on the disc for the output files.

On completion of the data entry screen, the graphical output from the program will appear on the screen. Computers with VGA graphics output will display this second screen in colour, but CGA machines will display it in monochrome. The graphical screen is shown in Figure A5.1.
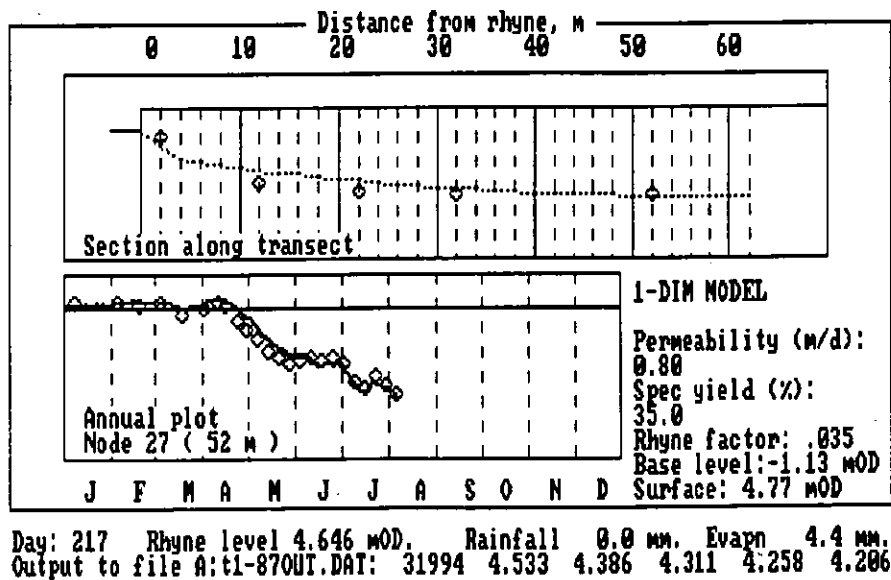

Figure A5.1 Example of graphical output from model

The upper half of the screen is taken up by a section plot along the dipwell transect. If the field width exceeds 140 m, only the 70 m nearest the rhyne is displayed; otherwise half the field width is shown. The upper heavy line (brown on a VGA screen) represents the ground surface, and the bottom line of the section (immediately below the label) is 3.8 mOD. On days for which dipwell data is contained in the data input file, circles (in red on VGA) are used to show the dipwell observations. The line representing the water table is deleted and updated for each day of the simulation.

The lower half is a plot of water table elevation versus time, at the position of the "field end" dipwell. This plot is updated daily, and circles are used to show the "field end" dipwell observations.

A5.3

The information panel at the foot of the screen shows the day number referred to the beginning of the year, and the climate and rhyne level data input for that day. The last line of the display shows the output file name and each line of output sent to it.

At any stage in the simulation, it is possible to print the screen using the (PrtSc) key, which will freeze operation temporarily, but at the end of the program a deliberate pause is inserted to allow the printing of the day 365 output. Following printing of the screen using the (PrtSc) key, pressing the (Return) key will step the program on to the final screen, which displays the objective function and goodness of fit. There is an opportunity to print this final screen before pressing the (Return) key to terminate the program and return control to the operating system.

Using the operating system installed with the computer, the (PrtSc) key cannot produce graphics screen dumps for VGA output: to obtain hard copy graphical output from a VGA monitor, an additional program such as Grafplus must be installed.

5.2.2 Complex 1-dimensional model

With a copy of Program Disc 1 in Drive A:, type **1D-COMP** (for complex 1-dimensional model) followed by (Return).

Remove the Program Disc and insert a copy of the Data Disc.

The procedure for running the complex 1-dimensional mode is similar, but it is necessary to enter two additional parameters, the exponential coefficients for the vertical variation of permeability and specific yield. Permeability and specific yield are allowed to vary exponentially with depth below the ground surface: the sign of the exponential coefficient is positive for the case of an exponential decrease with depth. Negative values may be used, but a zero coefficient (equivalent to a vertically uniform permeability or specific yield) will result in a division by zero. It is recommended that a small positive value (e.g. 0.001) is used in this case. If both parameters are vertically uniform, the simple model should be used.

5.2.3 Simple 2-dimensional model

With a copy of Program Disc 2 in Drive A:, type **2D-SIMP** (for simple 2-dimensional model) followed by (Return).

Remove the Program Disc and insert a copy of the Data Disc.

Parameter entry is identical to that for the 1-dimensional model, but it will be found that rhyne factors must be larger (see Section 1.2.2) by a factor of 6.25 to take account of the increased grid interval. The 2-dimensional model requires one additional entry, the field length, which is given in Table A5.2.

It will be found that the program runs very slowly in comparison with the 1-dimensional model: this is an inevitable consequence of the increased number of nodes, e.g. from 32 to 299 for transect T1 despite the change in grid interval.

5.2.4 Complex 2-dimensional model

With a copy of Program Disc 2 in Drive A:, type **2D-COMP** (for simple 2-dimensional model) followed by (Return).

Remove the Program Disc and insert a copy of the Data Disc.

Parameter entry is as for the simple 2-dimensional model, with the addition of the two exponential coefficients (see Section A5.2.2).

5.3 Installation on hard disc system

Start the computer as normal, and change active drive to A:

Make a backup copy of each program disc (refer to operating system manual).

Make a backup copy of the data disc.

These backup copies should be used in all subsequent work, and the original discs should be retained in a safe place in case of loss or damage of the original discs.

Change the active drive to C: (the hard disc).

Make a directory for the model programs, using the **MD** instruction (the directory name **MODEL** is a suggestion only, and the user may choose a suitable directory name):

MD MODEL

Change the active directory:

CD MODEL

Copy the program files from Program Disc 1 to the hard disc:

COPY A:*.EXE C:

Repeat the procedure with Program Disc 2.

5.4 Running the model on a hard disc system

The procedure of running the model from the hard disc is slightly simpler than from a floppy disc.

Start the computer as normal.

Change the active directory:

CD MODEL

Insert a copy of the Data Disc in Drive A:

5.4.1 Simple 1-dimensional model

Type **1D-SIMP** and proceed as in Section 1.2.1 with the entry of parameters.

5.4.2 Complex 1-dimensional model

Type **1D-COMP** and proceed as in Section 1.2.2 with the entry of parameters.

5.4.3 Simple 2-dimensional model

Type **2D-SIMP** and proceed as in Section 1.2.3 with the entry of parameters.

5.4.4 Complex 2-dimensional model

Type **2D-COMP** and proceed as in Section 1.2.4 with the entry of parameters.

A consortium of institutes within NERC conducted a commissioned research project for the NRA under the coordination of Dr Stephen Briggs of the Remote Sensing Applications Development Unit. The review was conducted between May and October, 1991 when a final report titled "A review of remote sensing techniques" was presented. Contributions were made from staff of IH, ITE, IFE, POL, NUTIS, RSADU, and PML.

The report is in two parts: a review of the NRA's data/information requirements and how remote sensing can contribute to satisfying these requirements, and a case studies pack to demonstrate what may be achieved using remote sensing techniques. Reference is made in part 1 of the report to the various case studies.

As part of the terms of reference of the contract, it was stipulated that the review be conducted according to the various NRA Commissions:-

Commission A   Water Quality
           B   Water Resources
           C   Flood Defence
           D   Fisheries
           E   Recreation and Navigation
           F   Conservation
           G   General Operations

Of these seven commissions, IH was given overall responsibility for Commission B - Water Resources and Commission C - Flood Defence, the latter with some input from other institutes, notably POL.

Part 1 of the report begins with a review of the information requirements of the NRA. This is done by Commission as indicated above, and includes all the information/data requirements of the NRA, indicating where remote sensing may be used in addition or as an alternative means of data collection. Following this is a review of remote sensing techniques. This describes current and future remote sensing platform/sensor combinations and indicates what applications are best suited for the various combinations. This is followed by a literature survey of the remote sensing techniques that had been identified as being of potential benefit to NRA's information requirements. Each review includes an extensive reference list. In the case of IH, this literature survey involved soil moisture, snow monitoring, land surface temperatures, areal rainfall measurement, and the identifications of urban (impervious) areas. The acquisition and processing of remotely sensed images are then described, together with estimates of costs. This part of the report ends with a chapter of conclusions and recommendations. For IH, it is concluded that remote sensing can provide the data requirements of the NRA in the areas reviewed in the literature, outlined above.

An extensive bibliography is given in an Appendix. This is structured in accordance with the main operational areas which are the responsibility of the NRA.

I have one copy of the final report in my office.   Gareth Roberts. 23.11.91

DCWM   West Sedgemoor job for Wessex Rivers   14/3/90

I was glad to find out from A/cs that this job has been fully invoiced and paid for.

Can you confirm that

(a) BSP has received a copy of the final report for the IH CR archive

(b) WBW has received a copy of the contents list / summary pages.

                    Thanks

                      FML

TERRESTRIAL AND FRESHWATER SCIENCES DIRECTORATE

Natural Environment Research Council
Polaris House
North Star Avenue
Swindon SN2 1EU
Tel: 0793 411500 Fax: 411502
Telex: 444293 ENVRE G

INSTITUTE OF FRESHWATER ECOLOGY

Windermere Laboratory
Far Sawrey, Ambleside
Cumbria LA22 0LP
Tel: 09662 2468 Fax: 6914
Telex: 94070416 WIND G

River Laboratory
East Stoke, Wareham
Dorset BH20 6BB
Tel: 0929 462314 Fax: 462180
Telex: 94070672 WARE G

Edinburgh Laboratory
Bush Estate, Penicuik
Midlothian, EH26 0QB
Tel: 031 445 4343 Fax: 3943
Telex: 72579 BUSITE G

Eastern Rivers Laboratory
Monks Wood Experimental Station
Abbots Ripton, Huntingdon
Cambs PE17 2LS
Tel: 04873 381 Fax: 467
Telex: 32416 MONITE G

Teesdale Laboratory
c/o Northumbrian Water
Lartington Treatment Works
Lartington, Barnard Castle
Co Durham DL12 9DW
Tel: 0833 50600 Fax: 50827

INSTITUTE OF HYDROLOGY

Maclean Building
Crowmarsh Gifford
Wallingford
Oxon OX10 8BB
Tel: 0491 38800 Fax: 32256
Telex: 849365 HYDROL G

Plynlimon Office
Staylittle, Llanbrynmair
Powys SY19 7DB
Tel: 05516 652 Fax: 441

Balquhidder Office
Tulloch Lodge
Balquhidder, Lochearnhead
Perthshire FK19 8PQ
Tel: 08774 257

INSTITUTE OF TERRESTRIAL ECOLOGY (NORTH)

Edinburgh Research Station
Bush Estate, Penicuik
Midlothian EH26 0QB
Tel: 031 445 4343 Fax: 3943
Telex: 72579 BUSITE G

Banchory Research Station
Hill of Brathens, Glassel
Banchory, Kincardineshire, AB31 4BY
Tel: 03302 3434 Fax: 3303

Merlewood Research Station
Grange-over-Sands, Cumbria LA11 6JU
Tel: 05395 32264 Fax: 34705
Telex: 65102 MERITE G

INSTITUTE OF TERRESTRIAL ECOLOGY (SOUTH)

Monks Wood Experimental Station
Abbots Ripton, Huntingdon
Cambs PE17 2LS
Tel: 04873 381 Fax: 467
Telex: 32416 MONITE G

Bangor Research Unit
University College of North Wales
Deinol Road, Bangor LL57 2UW
Tel: 0248 370045 Fax: 355365
Telex: 61224 BANITE G

Furzebrook Research Station
Wareham, Dorset BH20 5AS
Tel: 0929 551518/9 Fax: 551087

INSTITUTE OF VIROLOGY AND ENVIRONMENTAL MICROBIOLOGY

Mansfield Road, Oxford OX1 3SR
Tel: 0865 512361 Fax: 59962

UNIT OF COMPARATIVE PLANT ECOLOGY
Department of Animal and Plant Sciences
University of Sheffield, Sheffield S10 2TN
Tel: 0742 768555 Fax: 760159
Telex: 547216 UGSHEF G

CENTRE FOR POPULATION BIOLOGY
Imperial College, Silwood Park
Ascot, Berks SL5 7PY
Tel: 0344 23911 Fax: 294339

WATER RESOURCE SYSTEMS RESEARCH UNIT
Department of Civil Engineering, Newcastle University
Newcastle-upon-Tyne NE1 7RU
Tel: 091 232 8511 Fax: 091 261 1182
Telex: 53654 UNINEW G

UNIT OF BEHAVIOURAL ECOLOGY
Department of Zoology, University of Oxford
South Parks Road, Oxford OX1 3PS
Tel: 0865 271165 Fax: 310447

Natural Environment Research Council

Terrestrial and Freshwater Sciences