8-1999

Analysis and parallel implementation of an individually based algae model

Erin M. Miller

To the Graduate Council:

I am submitting herewith a thesis written by Erin M. Miller entitled "Analysis and parallel implementation of an individually based algae model." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Thomas Hallam, Major Professor

We have read this thesis and recommend its acceptance:

Christian Halloy, Louis J. Gross

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:
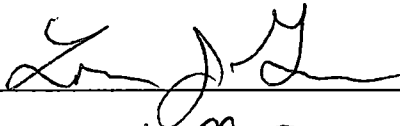
I am submitting herewith a thesis written by Erin M. Miller entitled "Analysis and Parallel Implementation of an Individually Based Algae Model". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

_____

Thomas Hallam, Major Professor

We have read this thesis
and recommend its acceptance:

_____

_____

Accepted for the Council:

_____

Associate Vice Chancellor
and Dean of the Graduate School

Analysis and Parallel Implementation of

an Individually Based Algae Model

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Erin M. Miller

August, 1999

Acknowledgments

I am forever grateful for my family and friends whose love, support and encouragement kept me motivated and optimistic. Thank you for seeing me through!

I am also thankful to Dr. Thomas G. Hallam, Dr. Christian Halloy, and Dr. Louis J. Gross. Dr. Thomas G. Hallam for being my major advisor and providing his guidance, direction and insight. Dr. Christian Halloy for being my boss and providing many helpful suggestions, conversation and support. And, Dr. Louis J. Gross for taking the time to serve on my committe and providing useful and constructive comments and encouragement. To all three I am especially grateful for your patience.

This thesis would not have been possible without financial support from the Joint Institute for Computational Science, and its understanding team of employees.

Finally, I extend my thanks to Bill Wilson and Dr. Bob Smith for lighting the way. Without you, this thesis would not have been possible.

Abstract

The focus of this research is the analysis and parallel implementation of an individually based algae population model. Analysis included examination of the sensitivity of the population's dynamics to increases of 25% and 50% and decreases of 25% and 50% of 58 of the model's parameters from a reference set of values under three different nutrient limiting conditions. Results indicate that the most sensitive parameters are those either directly or indirectly associated with the construction of protein. Analysis of the model also included examination of the influence of fluctuating temperatures on uptake of nutrients. Results indicate that while external nutrients are abundant, temperature influences the system, but when external nutrients become limiting, temperature effects diminish.

Parallel implementation included analysis of a pre–exisiting algae code in order to identify avenues for parallelization. To accomodate identified parallelization avenues, the original code was restructured and subsequently parallelized. Results from the parallel model were then compared with results from the sequential model to determine accuracy, and speed–up issues were addressed. It was determined that the parallel model, in its current form, offers no advantage over the sequential model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The work presented here is part of an ongoing effort towards the construction of a theoretical three species food chain that will ultimately be used to examine the potential impacts of various types of environmental stressors on the model ecosystem. An environmental stressor has been described by Hallam [6] as "any physical, chemical or biological component of an organism's environment that can strain a process associated with [the] ecological entity."

The models eventually comprising the food chain are each individually based systems representing algae, *Daphnia* and fish; the algae model is physiologically and stoichiometrically structured, and the *Daphnia* and fish models are physiologically structured. Extensive analysis has been previously performed on both the *Daphnia* and fish models, including analyses of both the unstressed populations (for *Daphnia* see [10] and for fish see [11]), and the stressed populations under

various types of stressors (for *Daphnia* see [7], [9], and [15] and for fish see [18]). Further analysis on the *Daphnia*–fish predator–prey system has been performed by Henson [11] and Nichols [20]. Collectively, these analyses have contributed valuable insight into the underlying dynamics of these models, insight without which there would be no basis for ultimately understanding the three species food chain dynamics. To further broaden understanding of the individual components comprising this system, an extensive analysis on both an unstressed algae population and a stressed algae population, with temperature playing the role of the stressor, is the focus of this work.

Because the goal of these models is to describe the relevant physiological dynamics of the individuals in their respective ecosystems, the models tend to be extensive, making the models moderately complex, and computationally intensive. These basic facts have led to simplifying assumptions such as the aggregation of individuals with the same biological and environmental parameters into ecotypes for all three models, and the further subdivision of these ecotypes into cohorts for the *Daphnia* and fish models. While serving the purpose of decreasing complexity, the fundamental issue of computational demands still remains. Substantial analysis in light of this issue has been performed on the *Daphnia* model [20], [22], [25], the fish model [20], [22] and the *Daphnia*–fish predator–prey system [20], [22] with parallelization as the natural venue for addressing this problem. It further stands to reason that if computational intensity is of concern for the individual

populations, its importance will only increase for the three species system. Thus, this thesis addresses the issue of parallelizing the algae model in preparation for its coupling with the *Daphnia*-fish predator–prey system.

1.1 Organization

The remainder of this thesis is organized into five chapters. Chapter Two details the sensitivity analysis performed on *Skeletonema costatum* under various nutrient limiting situations. Chapter Three focuses on the inclusion of temperature as a stressor to the system and the results of these effects. Chapter Four describes restructuring of the original algae code and provides several reasons as to why this laborious task was necessary. Chapter Five discusses parallelization of the algae model including numerical schemes tried, speed-up differences between sequential and distributed memory approaches, and recommendations indicating the most time efficient computational approach as a function of the number of ecotypes being examined. And, chapter Six points to potential future directions for this model.

1.2 The Algae Model

The following section describes the algae model as developed by Hurlebaus [12]. The goal of the algae model is to capture the dynamics of the reference algal

populations by describing changes in the internal states of its constituent members through time. This is achieved by examining the flux of nutrients from an individual's external environment to its internal environment and the subsequent flux of these nutrients between components comprising the internal environment. The resulting model describes the internal state of each individual by a set of ordinary differential equations, which are coupled by a McKendrick–Von Foerster [23] equation used to describe the dynamics of the population. (See Hallam [10], [8] for examples of previous applications of this method.)

The individual's internal environment is comprised of four "pools" consisting of a nutrient pool, an energy pool, a storage pool and a structure pool. Fluxes between these pools occur through anabolic and catabolic activities. Anabolic processes create storage and structural materials from available nutrients and energy, thus depleting internal nutrient and energy levels. Catabolic processes, on the other hand, breakdown storage and structural materials, thus releasing nutrients and energy and replenishing internal nutrient and energy pools. Using the internal nutrient or energy pool (INEP) as a reference point, the equation: $INEP = uptake - excretion - anabolism + catabolism$ generically describes this process. Figure 1.1 illustrates the processes and the following discussion details the equations used to model the internal state of an individual algal cell as it moves through time.
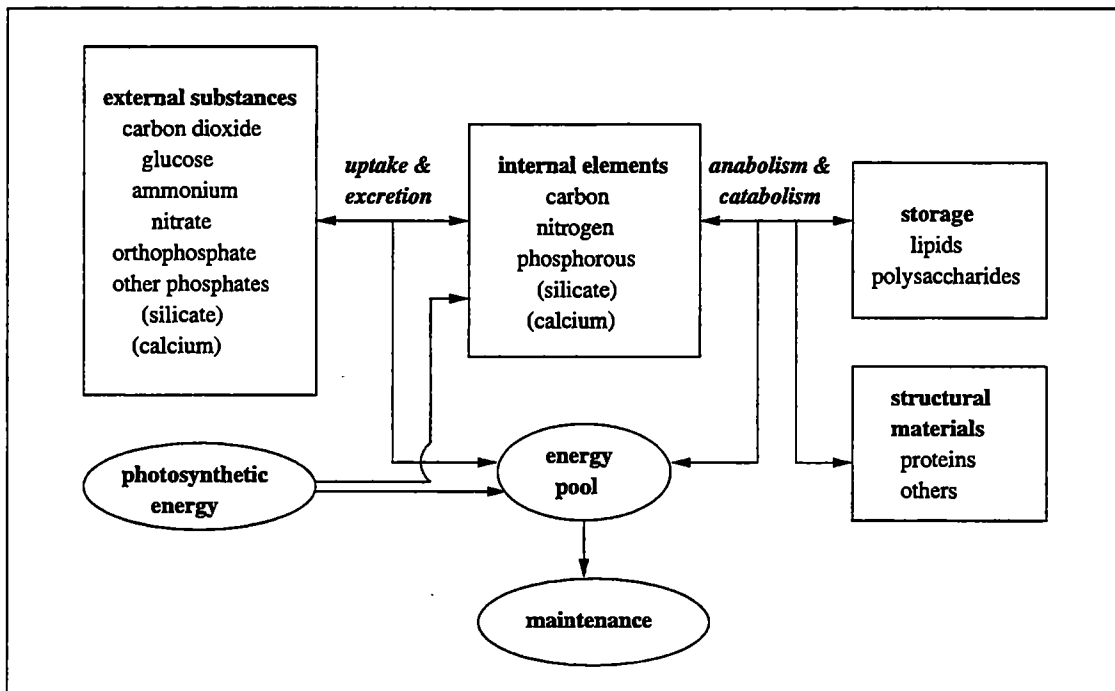
Figure 1.1: Conceptual diagram of the algae model. Modified from Hurlebaus [12]

1.2.1 Transition–Uptake and Transition–Storage Matrices

The driving equations of the algae model are those describing the uptake of nutrients from the external environment and the ultimate fate of those nutrients once in the interior of the cell. Exterior substances include Carbon dioxide (CO_2), Iron (FE), Ammonium (AM), Nitrate (NI), Orthophosphate (OP), other Phosphates (OR), and Silicate (SI). These are converted into the interior substances Carbon (C), Iron (Fe), Nitrogen (N), Phosphorous (P) and Silicate (SI). These processes are described by the following equations (see [17] for a similar application and development of this method):

$$
\begin{array}{cc}
\text{Transition-Uptake} & \text{Transition-Storage} \\
\vec{\rho}_{\mathbf{up},\phi_i} = \mathbf{T}_{\mathbf{up}}\vec{\rho}_{\mathbf{up},\phi_e} & \vec{\rho}_{\mathbf{st},\phi_a} = \mathbf{T}_{\mathbf{st}}\vec{\rho}_{\mathbf{st},\phi_i}
\end{array} \tag{1.1}
$$

where $\vec{\rho}_{\mathbf{up},\phi_i} \left(\frac{\mu mols}{s}\right)$ is the flux of internal nutrients and energy (ATP) due to the uptake of external nutrients and ATP, and it is given by the vector $[\rho_{up,C} \ \ \rho_{up,FE} \ \ \rho_{up,N} \ \ \rho_{up,P} \ \ \rho_{up,ATP} \ \ \rho_{up,SI}]^T$, $\vec{\rho}_{\mathbf{up},\phi_e} \left(\frac{\mu mols}{s}\right)$ is the uptake of substance ϕ_e given by the vector $[\rho_{up,CO_2} \ \ \rho_{up,FE} \ \ \rho_{up,AM} \ \ \rho_{up,NI} \ \ \rho_{up,OP} \ \ \rho_{up,OR} \ \ \rho_{up,SI}]^T$, $\vec{\rho}_{\mathbf{st},i} \left(\frac{\mu mols}{s}\right)$ describes the flux between the internal nutrient pools and storage given by the vector $[\rho_{st,C} \ \ \rho_{st,FE} \ \ \rho_{st,N} \ \ \rho_{st,P} \ \ \rho_{st,ATP} \ \ \rho_{st,SI}]^T$, and $\vec{\rho}_{\mathbf{st},\phi_a} \left(\frac{\mu mols}{s}\right)$ is the anabolism or catabolism of substance ϕ_s to or from storage given by the vector $[\rho_{st,Ps} \ \ \rho_{st,Lp} \ \ \rho_{st,Pr}]^T$. Matrices $\mathbf{T}_{\mathbf{up}}$ and $\mathbf{T}_{\mathbf{st}}$ are described below.

6

$\mathbf{T_{up}}$ is called the Transition–Uptake Matrix. It contains the conversion coefficients $\mu_{\phi_i,\phi_e}(nd)$ representing the number of ϕ_i molecules that can be derived from the uptake of one ϕ_e molecule. It is given by the following:

$$T_{up} = \begin{bmatrix} \mu_{C,CO_2} & \mu_{C,FE} & \mu_{C,AM} & \mu_{C,NI} & \mu_{C,OP} & \mu_{C,OR} & \mu_{C,SI} \\ \mu_{FE,CO_2} & \mu_{FE,FE} & \mu_{FE,AM} & \mu_{FE,NI} & \mu_{FE,OP} & \mu_{FE,OR} & \mu_{FE,SI} \\ \mu_{N,CO_2} & \mu_{N,FE} & \mu_{N,AM} & \mu_{N,NI} & \mu_{N,OP} & \mu_{N,OR} & \mu_{N,SI} \\ \mu_{P,CO_2} & \mu_{P,FE} & \mu_{P,AM} & \mu_{P,NI} & \mu_{P,OP} & \mu_{P,OR} & \mu_{P,SI} \\ \mu_{ATP,CO_2} & \mu_{ATP,FE} & \mu_{ATP,AM} & \mu_{ATP,NI} & \mu_{ATP,OP} & \mu_{ATP,OR} & \mu_{ATP,SI} \\ \mu_{SI,CO_2} & \mu_{SI,FE} & \mu_{SI,AM} & \mu_{SI,NI} & \mu_{SI,OP} & \mu_{SI,OR} & \mu_{SI,SI} \end{bmatrix} \tag{1.2}$$

$\mathbf{T_{st}}$ is called the Transition–Storage Matrix. It contains the conversion coefficients $\mu_{\phi_i,\phi_s}(nd)$ representing the number of ϕ_i molecules needed to make its respective contribution to the composition of a storage or structure molecule, ϕ_s. The matrix is given below.

$$T_{st} = \begin{bmatrix} \mu_{C,Ps} & \mu_{C,Lp} & \mu_{C,Pr} \\ \mu_{Fe,Ps} & \mu_{Fe,Lp} & \mu_{Fe,Pr} \\ \mu_{N,Ps} & \mu_{N,Lp} & \mu_{N,Pr} \\ \mu_{P,Ps} & \mu_{P,Lp} & \mu_{P,Pr} \\ \mu_{ATP,Ps} & \mu_{ATP,Lp} & \mu_{ATP,Pr} \\ \mu_{SI,Ps} & \mu_{SI,Lp} & \mu_{SI,Pr} \end{bmatrix} \tag{1.3}$$

Equations describing the uptake of the individual components of the vectors $\vec{\rho}_{\text{up},\phi_e}$ and $\vec{\rho}_{\text{st},\phi_s}$ are discussed in the following subsections.

1.2.2 Nutrient Pool Equations (Excluding Carbon (C))

Nutrient pools are replenished through catabolism of storage materials (Section 1.2.5) and the uptake of external substances. Depletion occurs through anabolism (Section 1.2.5) and excretion. Carbon is considered separately since its uptake pathway differs from that discussed below.

Nutrient Uptake, ρ_{up,ϕ_e}

Uptake for the external substance ϕ_e is determined by the following equation:

$$\rho_{up,\phi_e} = \begin{cases} \left(\dfrac{1}{\frac{a}{\nu(T)}+\frac{c_{\phi_e}}{\phi_e}}\right)\left(1-\dfrac{\phi_i}{c_{\phi_i}m_{Pr}}\right)n_{\phi_e}SA & \phi_e > 0 \\ \\ 0 & \phi_e = 0 \end{cases} \tag{1.4}$$

where $a(cm)$ is the cell wall thickness, $\nu(T)\left(\frac{\mu mols\ cm}{sec}\right)$ is the transport velocity of substance ϕ_e across the cell wall as a function of temperature, $c_{\phi_e}\left(\frac{s}{cm^3}\right)$ is a proportionality constant associated with substance ϕ_e, $\phi_e\left(\frac{\mu mols}{cm^3}\right)$ is the external concentration of the substance ϕ, $\phi_i\left(\mu mols\right)$ is the internal nutrient ϕ_i derived from the associated external substance ϕ_e, $m_{Pr}\left(\mu mols\right)$ is the reserve protein or structural material in the cell, $c_{\phi_i}\left(nd\right)$ is a proportionality constant associated with the internal nutrient ϕ_i, $n_{\phi_e}\left(\frac{1}{cm^2}\right)$ is the number of processors on the surface

8

of the cell that can uptake substance ϕ_e, and $SA(cm^2)$ is the surface area of the cell.

Transport velocity, $\nu(T)$, is given by the equation:

$$\nu(T) = s(T_1^K)exp\left(\frac{T_A^K}{T_1^K T^K}\left(T^K - T_1^K\right)\right) \qquad (1.5)$$

where $s(T_1^K)\left(\frac{\mu mols\ cm}{s}\right)$ is a constant, $T_A^K(K)$ is the Arrhenius temperature in Kelvins, $T_1^K(K)$ is a chosen reference temperature in Kelvins, and $T^K(K)$ is the absolute temperature in Kelvins. (This equation was first proposed by Arrhenius and is discussed in [4], [16].)

Isomorphism is assumed. Hence, the cell's volume is proportional to the amount of polysaccharides, lipids and proteins within the cell. The equations for volume and surface area are thus:

$$V = \frac{m_{Ps}}{\sigma_{Ps}} + \frac{m_{Lp}}{\sigma_{Lp}} + \frac{m_{Pr}}{\sigma_{Pr}} \qquad (1.6)$$

$$SA = V^{\frac{2}{3}} = \left(\frac{m_{Ps}}{\sigma_{Ps}} + \frac{m_{Lp}}{\sigma_{Lp}} + \frac{m_{Pr}}{\sigma_{Pr}}\right)^{\frac{2}{3}} \qquad (1.7)$$

where m_{Ps}, m_{Lp} and m_{Pr} are the amounts of polysaccharides, lipids and proteins, respectively, in the cell in units of $\mu mols$, and $\sigma_{\phi_s}(\frac{\mu mols}{cm^3})$ is the molecular density of ϕ_s where ϕ_s is polysaccharides, lipids or proteins.

9

Excretion

Excretion of nutrient ϕ_i is given by the following equation:

$$\rho_{ex,\phi_i} = \dot{j}_{\phi_i} \frac{\phi_i}{m_{Pr}} \tag{1.8}$$

where $j_{\phi_i}(\mu mols)$ is a constant associated with ϕ_i.

Phosphorous experiences an additional loss of three molecules per energy molecule residing in the internal energy pool.

1.2.3 Carbon Pool Equations

The internal carbon pool is replenished by catabolism and photosynthetic activity, and is depleted by anabolism and respiration.

Carbon Uptake, $\rho_{up,C}$

Carbon uptake, $\rho_{up,C}$, is assumed to be solely dependent on the photosynthetic process and the size of the cell. It is given by:

$$\rho_{up,C} = L(I)SA \tag{1.9}$$

where $L(I)$ is the photosynthetic activity dependent on light availability.

Photosynthetic activity, $L(I)$ is described by the following set of equations:

$$L(I) = -\frac{n}{k} \sum_{j=0}^{5} \left(-\frac{E_d(0)}{E_k}\right)^j \frac{(e^{kz_0}-1)^j}{j!}$$

$$n = L_{max} \left(1 - \frac{L_\beta}{N_i + L_\alpha}\right)\left(\frac{1}{1+\frac{L_P}{P_i}}\right) \qquad (1.10)$$

$$k = -(K_s + \sigma K_p)$$

where $E_d(0)\left(\frac{\mu mols}{m^2 s}\right)$ is the irradiance just below the surface, E_k is the line where initial photosynthetic increase intersects the maximum photosynthetic line (see [12] for details), $k\left(\frac{1}{m}\right)$ is the absorption of light in water, $K_s\left(\frac{1}{m}\right)$ is the light absorption coefficient due to seawater, $K_p\left(\frac{1}{m}\right)$ is the light absorption coefficient due to phytoplankton, $\sigma(\#)$ is the total number of phytoplankton, $z_0(m)$ is depth, $N_i(\mu mols)$ is the level of internal nitrogen, $P_i(\mu mols)$ is the level of internal phosphorous, and $L_{max}(nd)$, $L_\beta(nd)$, $L_\alpha(nd)$ and $L_P(\mu mols)$ are constants.

Respiration

Respiration depletes internal carbon levels and is as follows:

$$\rho_{ex,C} = j_{ex,C} C_i \qquad (1.11)$$

where $j_{ex,C}(s^{-1})$ is the fraction of free carbon molecules respired per second, and $C_i(\mu mols)$ is the level of internal carbon.

1.2.4 Energy Pool Equations

Internal energy (ATP) levels are replenished by catabolism, photosynthesis and respiration. Depletion occurs through anabolism and cell maintenance where maintenance is given by the following:

$$M = M_0(m_{Ps} + m_{Lp} + m_{Pr}) \tag{1.12}$$

where M is the maintenance demand dependent on internal storage and structure, and $M_0 \left(\frac{1}{\mu mols} \right)$ is the maintenance cost.

1.2.5 Storage and Structure Fluxes, ρ_{st,ϕ_s}

The total flux of substance ϕ_s into or out of storage or structure, where ϕ_s is one of polysaccharides, lipids or proteins, is the difference between the results of the anabolic and catabolic processes. The flux equations are given below:

$$\rho_{st,Ps} = m_{0,Ps} f_{Ps}(\bar{E}, C_i) - g_{Ps}$$
$$\rho_{st,Lp} = m_{0,Lp} f_{Lp}(\bar{E}, C_i, P_i) - g_{Lp} \tag{1.13}$$
$$\rho_{st,Pr} = m_{0,Pr} f_{Pr}(\bar{E}, C_i, P_i, N_i, [SI_i]) - g_{Pr}$$

where $m_{0,\phi_s} \left(\frac{\mu mols}{s} \right)$ is the molecules of type ϕ_s fixed during anabolism per unit time, $f_{\phi_s}(nd)$ is the functional response (anabolism) dependent on availability of internal nutrients ϕ_i needed to comprise one molecule of substance ϕ_s, $g_{\phi_s}(\frac{\mu mols}{s})$

is the catabolic rate of substance ϕ_s, and $\phi_i(\mu mols)$ is available internal nutrients of ATP, C, P, N or SI where the incorporation of SI is species dependent.

Anabolism

Functional responses, f_{ϕ_s}, are given below:

$$f_{Ps}(\bar{E}, C_i) = \frac{1}{max\left(\frac{c'_{C_i}}{C_i}, \kappa_{ATP,Ps}\frac{c'_E}{E}\right) + \frac{1}{I_{m,Ps}}}$$

$$f_{Lp}(\bar{E}, C_i, P_i) = \frac{1}{max\left(\frac{c'_{C_i}}{C_i}, \kappa_{ATP,Lp}\frac{c'_E}{E}, \kappa_{P,Lp}\frac{c'_{P_i}}{P_i}\right) + \frac{1}{I_{m,Lp}}}$$

$$f_{Pr}(\bar{E}, C_i, P_i, N_i, [SI_i]) = \frac{1}{max\left(\frac{c'_{C_i}}{C_i}, \kappa_{ATP,Pr}\frac{c'_E}{E}, \kappa_{P,Pr}\frac{c'_{P_i}}{P_i}, \kappa_{N,Pr}\frac{c'_{N_i}}{N_i}, \left[\kappa_{SI,Pr}\frac{c'_{SI_i}}{SI_i}\right]\right) + \frac{1}{I_{m,Pr}}}$$

$$(1.14)$$

where $c'_{\phi_i}(s)$ is a proportionality constant associated with ϕ_i, $\kappa_{\phi_i,\phi_s}(\frac{\mu mols}{s})$ relates the number of ϕ_i molecules needed to comprise a molecule of ϕ_s, and $I_{m,\phi_s}(nd)$ is a measure for the maximum rate of production of one molecule of ϕ_s.

Catabolism

Catabolic equations, g_{ϕ_s}, are given by the following:

$$g_{Ps} = \begin{cases} \kappa_{Ps,0}\left(1 + \frac{\kappa_{Ps,3}}{P_i}\right)\left(\frac{1}{\kappa_{Ps,1}+\kappa_{Ps,2}\frac{\bar{E}}{m_{Ps}}}\right) & m_{Ps}, P_i \text{ and } \bar{E} > 0 \\ 0 & m_{Ps}, P_i \text{ or } \bar{E} = 0 \end{cases}$$

$$g_{Lp} = \begin{cases} \kappa_{Lp,0}\left(1 + \frac{\kappa_{Lp,3}}{P_i}\right)\left(\frac{1}{\kappa_{Lp,1}+\kappa_{Lp,2}\frac{\bar{E}}{m_{Lp}}}\right) & m_{Lp}, P_i \text{ and } \bar{E} > 0 \\ 0 & m_{Lp}, P_i \text{ or } \bar{E} = 0 \end{cases} \qquad (1.15)$$

$$g_{Pr} = \begin{cases} \kappa_{Pr,0}\dfrac{1}{\kappa_{Pr,1}+max\left(\kappa_{Pr,2}\frac{\bar{E}}{m_{Pr}}, \frac{\kappa_{Pr,3}}{P_i}\right)} & m_{Pr}, P_i \text{ and } \bar{E} > 0 \\ 0 & m_{Pr}, P_i \text{ or } \bar{E} = 0 \end{cases}$$

where $\kappa_{\phi_s,0}\left(\frac{\mu mols}{s}\right)$ is the maximum catabolic rate for substance ϕ_s, $\kappa_{\phi_s,j}(nd), j = 1,2$ is a constant, and $\kappa_{\phi_s,3}(\mu mols)$ is a constant.

1.2.6 Reproduction

Reproduction occurs when the internal protein level passes a minimal threshold before which reproduction could not occur. The result is an equivalent allocation of the parents internal resources to each offspring. This is given by:

$$B(t) = \sum_{ecotypes} 2\sigma(t, m_{Ps}, m_{Lp}, 2m_{Pr,0})$$
$$\sigma(t, m_{Ps}, m_{Lp}, m_{Pr,0}) = 2\sigma(t, 2m_{Ps}, 2m_{Lp}, 2m_{Pr,0}) \qquad (1.16)$$

where $m_{Pr,0}$ is the minimal amount of protein needed for cell division and can be

different for each ecotype, and $\sigma(t, m_{Ps}, m_{Lp}, m_{Pr})$ is the number of cells at time with internal levels of m_{Ps}, m_{Lp} and m_{Pr} .

1.2.7 Population Level Equations

The population equation depends on the levels of m_{Ps}, m_{Lp} and m_{Pr} determined by the individual equations, and is given by the following:

$$\sigma(t + \Delta t, m_{Ps} + \rho_{st,Ps}\Delta t, m_{Lp} + \rho_{st,Lp}\Delta t, m_{Pr} + \rho_{st,Pr}\Delta t) = \sigma(t, m_{Ps}, m_{Lp}, m_{Pr})$$

$$-D(m_{Ps}, m_{Lp}, m_{Pr})\sigma(t, m_{Ps}, m_{Lp}, m_{Pr})\Delta t$$

$$(1.17)$$

where D is the death rate and the birth equation is as given above.

1.2.8 Sinking and Grazing

Sinking was modelled using the following formulation:

$$\frac{x}{z}\sigma(t, x) \qquad (1.18)$$

where $x(cm^3)$ is the volume of the cell, and $z(m)$ is the depth of the surface layer.

1.3 The Reference Populations

The algae model has been parameterized for two reference species. The first is a silicate requiring marine diatom known as *Skeletonema costatum*. This species

15

is often found co-habitating with non-silicate requiring algal species. To produce preliminary results concerning the interaction between silicate and non-silicate requiring species, a theoretical algal species not requiring silicate was created. This species uses the same parameter set as *Skeletonema costatum* except that its maintenance cost, M_0, is higher while its reproduction threshold, Pr_{min}, is lower. Dynamics of *Skeletonema costatum* are explored in the analysis section.

Chapter 2

Sensitivity Results for Reference Species *Skeletonema costatum*

A key objective of this thesis is to explore the sensitivity of the reference populations to parameter perturbations under various limiting external nutrient conditions. Analysis was accomplished by first changing the levels of nutrients pulsed into the systems so that only one of silicate, phosphate or nitrate, limited the system in the steady state; hence a "limited system". A nutrient was considered limiting in the steady state if it was the primary nutrient limiting protein production. Next, each systems' free parameters were perturbed, and the populations simulated under these conditions. Parameters were considered sensitive if there was a resulting change in population numbers in either the transient or steady state behavior of the population. (Parameters not showing sensitivity are not dis-

17

cussed.) All simulations for a particular reference population used the same initial conditions so that the resulting transient behavior was similar. Additionally, all populations, independent of species, experienced an 18 hour day, 6 hour night cycle and constant temperature. Steady states for these populations were determined by finding the maximum and minimum population numbers for the last ten reproductions taking place in a simulated year, and then comparing these values with the maximum and minimum values attained in preceeding reproductions. The point before which the maximum or minimum of two successive reproductions fell beyond the predetermined maximum and minimum values was used as the beginning of the steady state. The results are presented below.

2.1 Perturbation Results for *Skeletonema costatum* in a Silicate Limited System

Silicate levels were chosen so that inputs were low enough for silicate to be growth limiting in the steady state, but high enough for the population to persist. Phosphate and nitrate levels were choosen so that limitations were either short-lived (for phosphorous), or never occurred (for nitrogen). Table 2.1 shows the results of the sensitivity analysis. Parameters were perturbed up 25% and 50% and down 25% and 50% from a reference set of parameters. Dots in the table indicate when the system showed sensitivity to the associated parameter's perturbation under

Table 2.1: Perturbation Results

Parameter	Definition	Up 25%	Up 50%	Dn 25%	Dn 50%
c'_C	functional response proportionality constant for carbon wait time				●
M_0	cost of maintenance		●	●	●
$\mu_{C,Pr}$	number of carbon molecules needed to construct a protein molecule	●	●	●	●
$\mu_{P,Pr}$	number of phosphorous molecules needed to construct protein molecule	●	●	●	●
n_{OP}	number of phophate processors on the surface of the cell				●
Pr_{min}	minimum amount of protein needed for cell division	●	●	●	●
$user3$ ($\mu_{Si,Pr}$)	number of silicate molecules needed to construct a protein molecule	●	●	●	●

these conditions.

2.1.1 Dynamics of the Unperturbed Silicate Limited System

To understand the effects of the perturbations it is important to first understand the underlying dynamics of the unperturbed silicate limited system. This is accomplished by breaking the dynamics down into sections naturally delineated by key events marking transitions in the system.

0–200 Hours

There are several observations in this first interval that set the stage for the progression of events that follow. Initially internal energy levels are high, so that cells are able to meet maintenance demands and grow. Increases in external phosphate,

19

silicate and nitrate levels indicate that internal phosphorous, silicate and nitrogen levels are sufficient to meet individuals' needs. Carbon is the only limiting nutrient throughout this period, thus limits polysaccharide, lipid and protein production. Metabolism is dominated by anabolism even though catabolism occurs simultaneously.

200–500 Hours

This second interval marks significant changes in internal phosphorous and energy levels. At approximately 200 hours the first cell divisions occur in which individuals in the population can no longer support themselves on internal phosphorous levels alone. Thus, the population begins to rely heavily on external phosphorous, and external supplies become virtually depleted during the five reproductions taking place between 200 and 300 hours. The following four reproductions, between 300 and 500 hours, decrease internal energy levels. Recall that each ATP molecule requires three phosphorous molecules, so that the decrease in internal energy alleviates pressure for internal phosphorous, consequently alleviating demands on external phosphorous. This is evidenced by the apparent increase in both internal and external phosphorous levels following the reduction in internal energy. It is important to observe that, although there is a significant decline in internal energy and phosphorous, these pools are not completely depleted since neither of these nutrients limit storage or structure material production. In fact, carbon remains

20

the sole limiting factor for storage and structure production throughout this entire interval.

500–800 Hours

Prior to this period, individual's internal energy levels were continuously high enough to meet maintenance demands. At approximately 500 hours, however, the first signs of the cells' inability to meet these demands appears. This results in the periodic cessation of anabolic activity and the subsequent domination of metabolism by catabolism. Anabolic cessation continues until internal energy levels return to those sufficient to meet maintenance demands.

During intervals of anabolic inactivity, high catabolic activity results in the break down of polysaccharides and lipids, hence the return of previously bound energy, carbon and phosphorous to their respective pools. This results in a marked increase in internal carbon levels by the end of this interval, although not significant enough to prevent carbon from remaining the primary limiting nutrient in polysaccharide, lipid and protein production. Finally, as the population continues to grow, demands on external silicate increase.

Figures

The following four figures show the progression of events from 0 to 800 hours. Figure 2.1 shows cell numbers, internal silicate levels and external silicate levels.
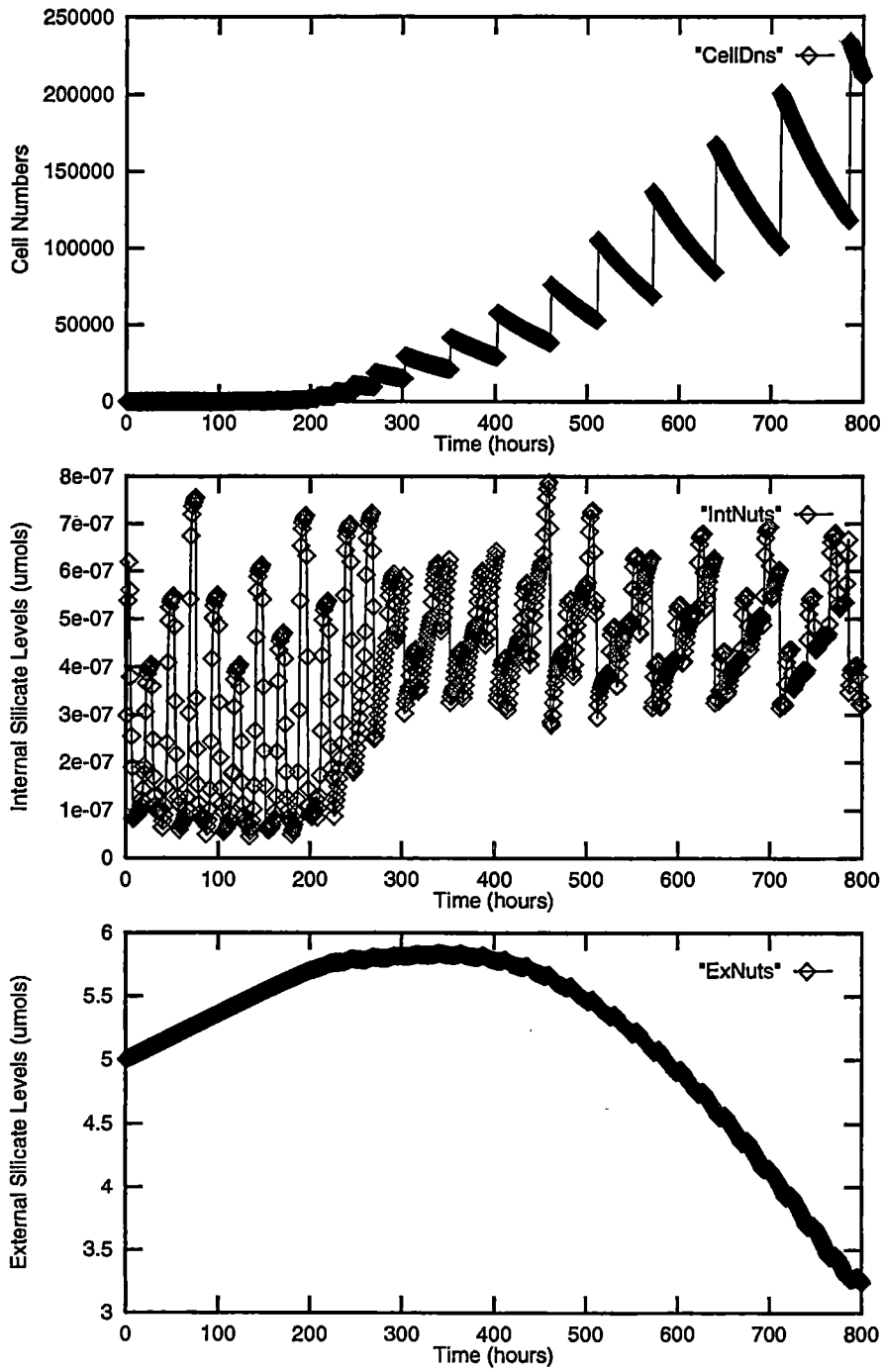
Figure 2.1: Behavior of reference population and internal and external silicate levels for 0–800 hours for silicate limited system.

22

Figure 2.2 shows internal energy, internal phosphorous and external phosphorous levels. Figure 2.3 shows the most limiting nutrient in polysaccharide, lipid and protein production. And, Figure 2.4 shows internal carbon, and the most anabolized and the most catabolized material. Nitrogen is not shown since it does not play a significant role in determining dynamics for this particular scenario, its values, however, consistently fluctuated between 1.0e-6 and 1.9e-6 throughout the simulation .

Events described by the anabolism and catabolism figures and the figures appearing in 2.3 are discrete. They are shown continuous for purposes of clarity. For example, consider the figure describing anabolism. One stands for polysaccharides, two for lipids and three for proteins. At time zero, anabolism of proteins (three) dominates this activity, and remains dominant until approximately 500 hours when cells begin to anabolize polysaccharides. From that point on, a discrete representation of events would be futile as it would be unclear as to which event came first. Including lines clarifies this since the slope clearly indicates the location of the last event. All figures aforementioned, and those to follow, conform to this convention.

800–1400 Hours

This interval begins the final depletion of the internal phosphorous and energy pools, and the subsequent transition of the system from a carbon limited system

23
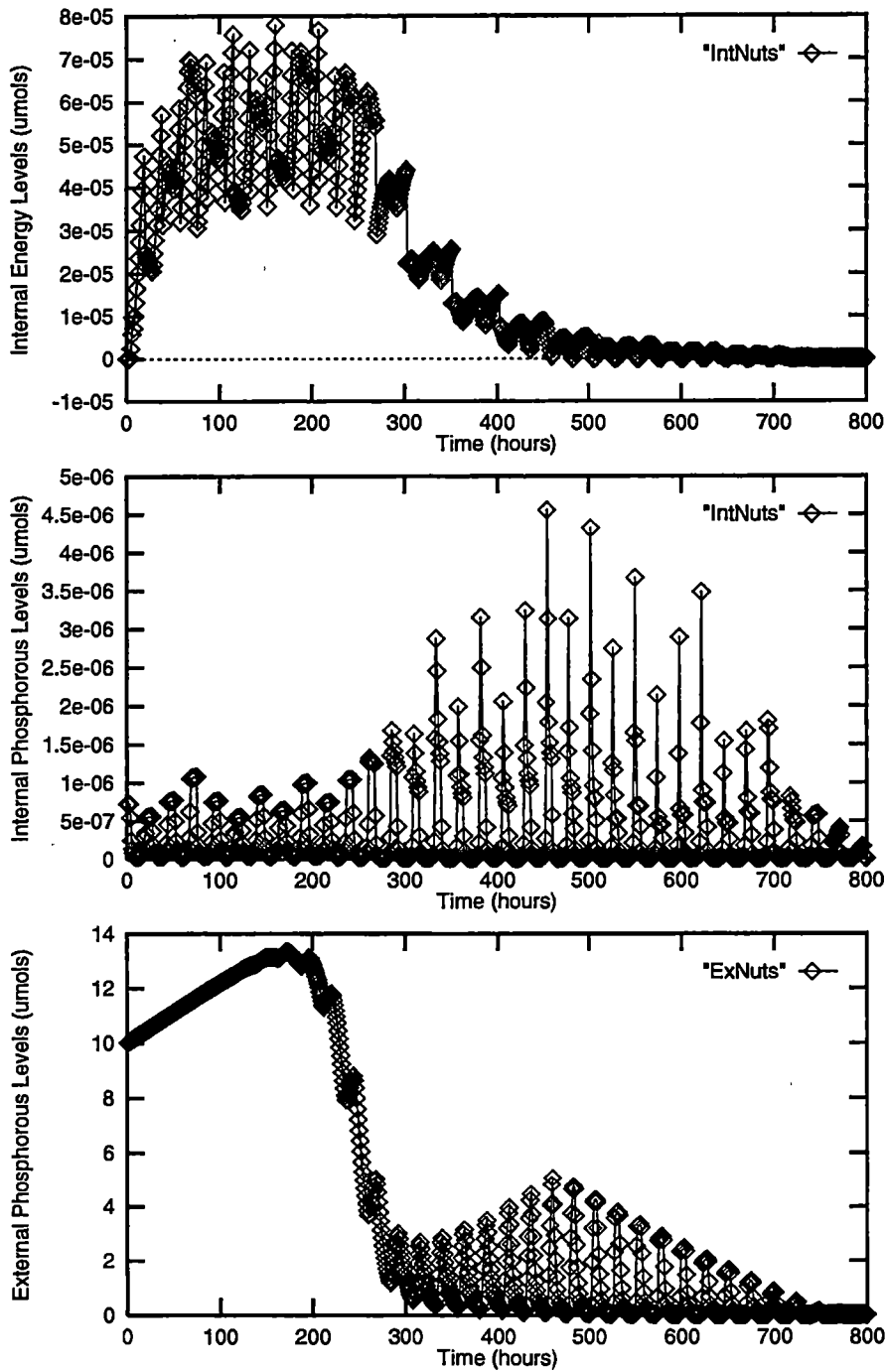
Figure 2.2: Behavior of internal energy and phosphorous and external phosphorous for 0–800 hours for silicate limited system.
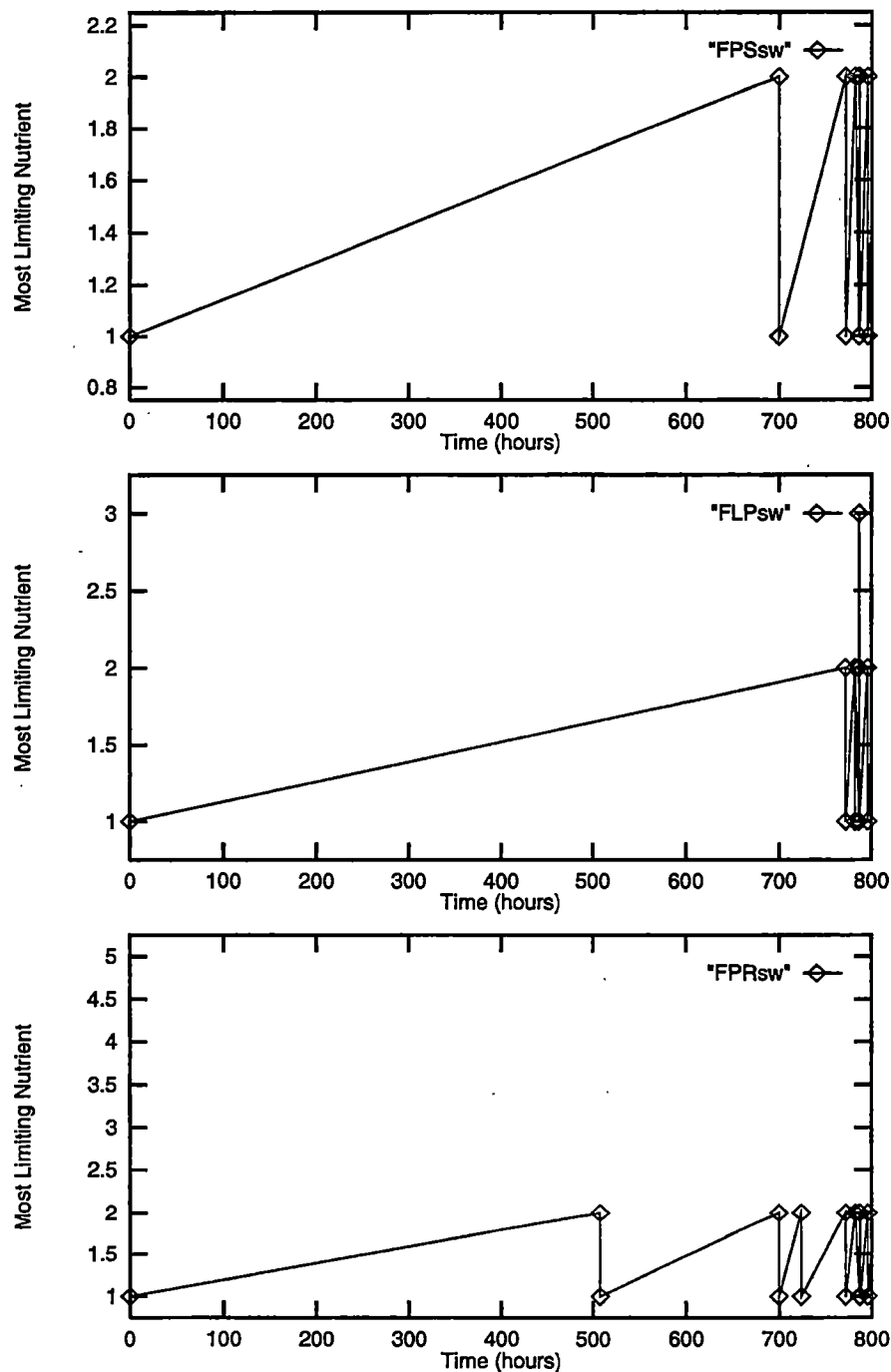
Figure 2.3: Most limiting nutrient for polysaccharides, lipids and proteins, 0–800 hours for silicate limited system. On the y–axis, numbers represent the most limiting nutrient in the following way: 1=carbon, 2=energy, 3=phosphorous, 4=nitrogen, 5=silicate. Lengths of periods where no switch occurs indicates that the last least limiting nutrient was limiting until the next switch.

25
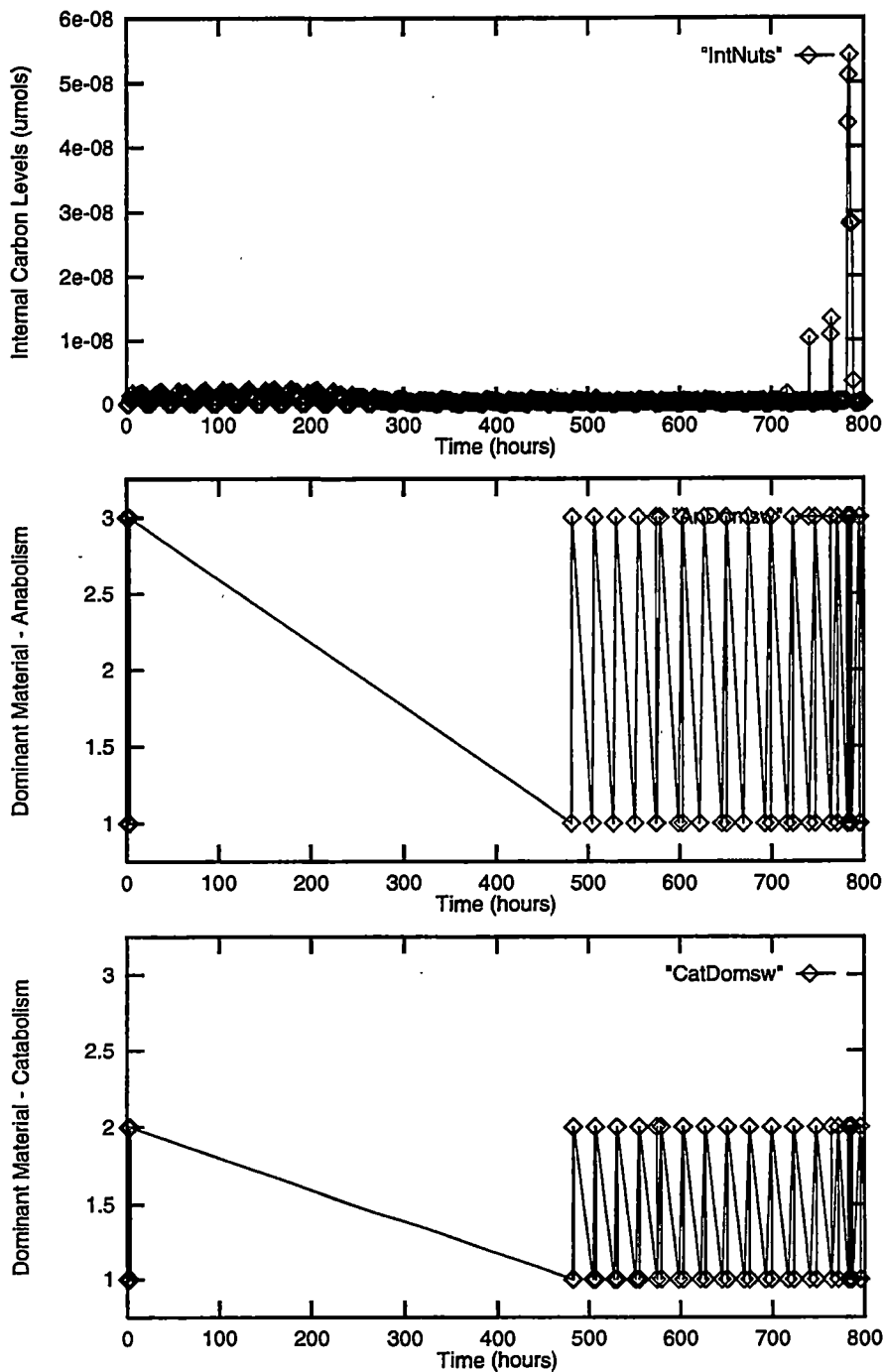
Figure 2.4: Internal Carbon levels and materials dominating anabolism and catabolism for 0–800 hours for silicate limited system. On the y–axis for the dominant anabolism and catabolism materials, numbers correspond to the materials in the following way: 1=polysaccharides, 2=lipids, 3=proteins. Lengths of periods where no switch occurs indicates that the last least limiting material was limiting until the next switch.

to a phosphorous and energy limited system. As individuals remain unable to continuously meet maintenance demands, polysaccharides and lipids are catabolized, hence internal energy, phosphorous and carbon pools are replenished. Albeit there is a positive flux into these pools, cells remain unsupportable by internal phosphorous and energy levels, but, for the first time, are supportable by internal carbon. This is evidenced by the fact that energy begins to play a prominant role in limiting the manufacture of polysaccharides, lipids and proteins, while phosphorous begins to play a prominant role in limiting lipid production. Previously, carbon was the primary limiting factor for all three materials.

In the hours prior to the end of this period, external silicate levels slowly decreased but remained high enough to support cells' demands. By the end of this interval, however, they have decreased below that which can support the population. At approximately 1300 hours, silicate becomes limiting for the production of protein, and by 1400 hours has become the sole limiting factor for the manufacture of this material.

1400-2400 Hours

The beginning of this interval delineates the final transition from a phosphorous and energy limited system to a silicate limited system. The decline in external silicate results in a corresponding decrease in population numbers which fall to levels supportable by available internal phosphorous. Thus, phosphorous no

longer limits the production of lipids, and pressures on external supplies are alleviated. Population decreases have also left individuals with sufficient internal energy so that by the end of this interval cells are once again able to continuously meet maintenance demands, and catabolism ceases to play a dominant role in metabolism. Finally, carbon and energy alternately limit polysaccharide and lipid production, and silicate is the sole limiting factor for protein production, hence is the primary limiting factor for population growth.

Figures

Figures 2.5 through 2.8 depict this sequence of events, showing the hours between 750 and 2400. These figures are a continuation of the previously presented figures which described events between 0 and 800 hours.

2400+ Hours

After approximately 2400 hours the system settles into a steady state. In this state, cells are successfully able to meet maintenance demands, carbon and energy continue to limit polysaccharide and lipid production, and silicate continues to limit protein production hence remains the determining factor for reproduction events.

28
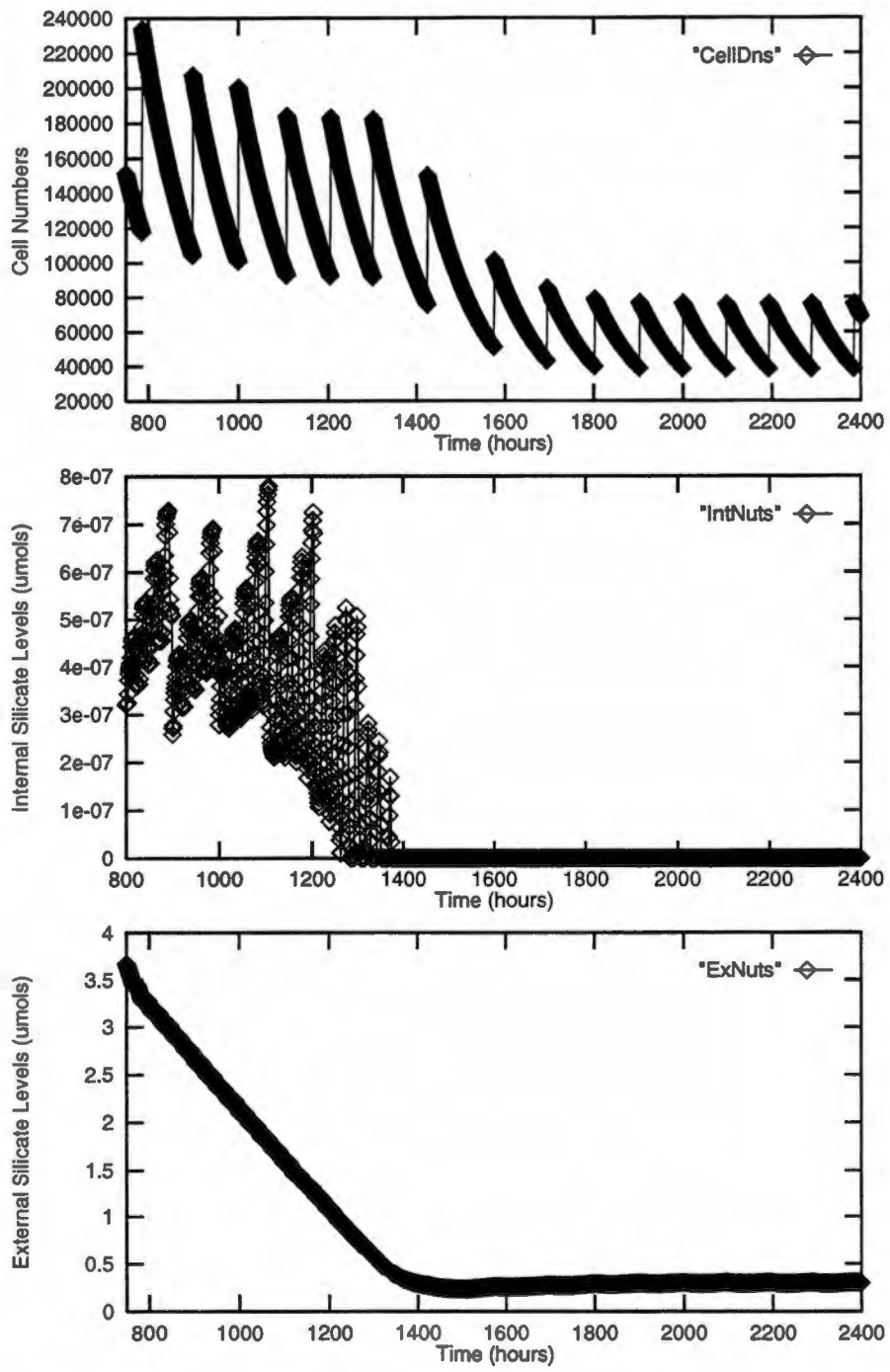
Figure 2.5: Behavior of reference population and internal and external silicate levels for 750–2400 hours for silicate limited system.
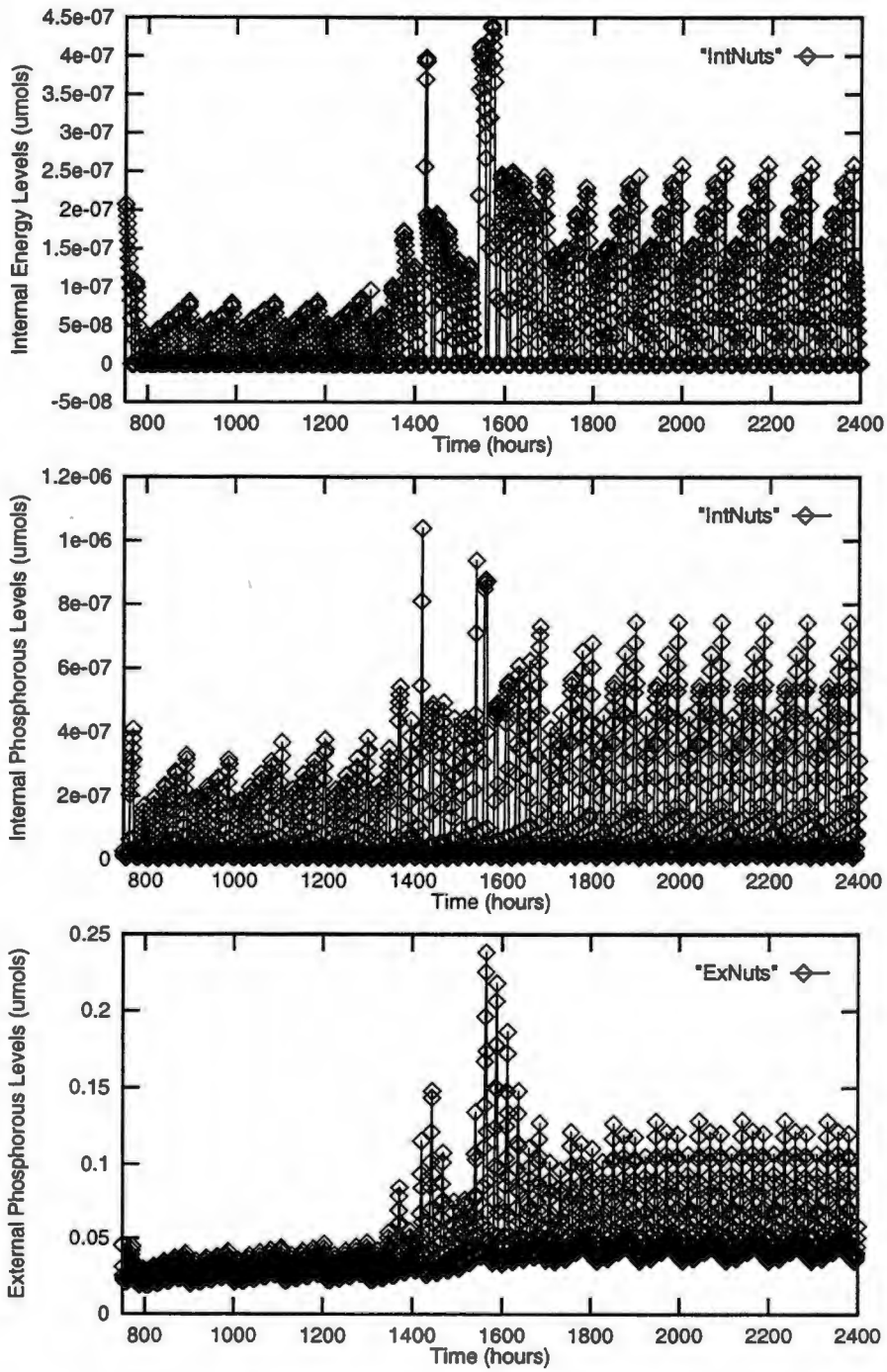
Figure 2.6: Behavior of internal energy and phosphorous and external phosphorous for 750–2400 hours for silicate limited system.
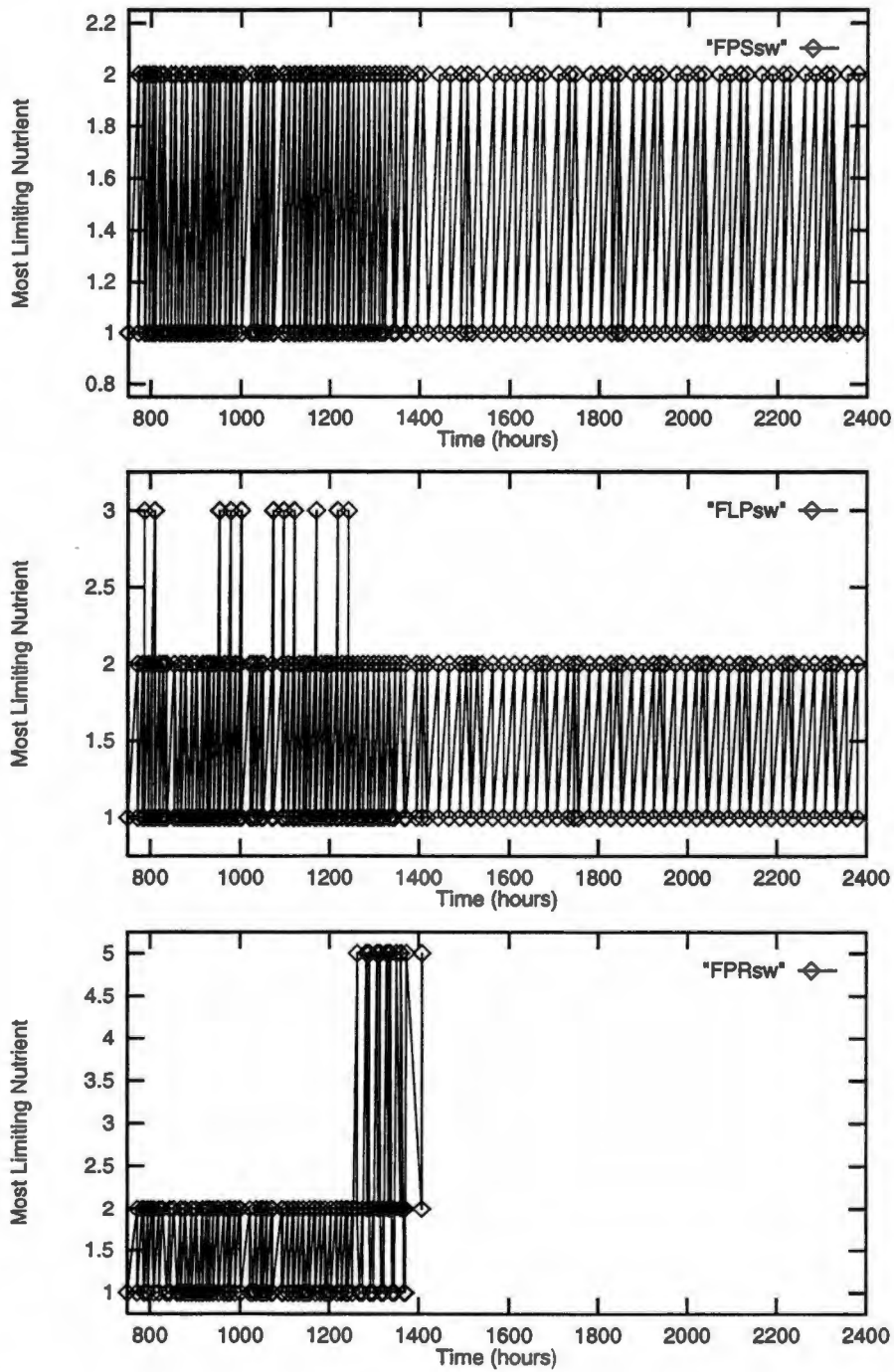
Figure 2.7: Most limiting nutrient for polysaccharides, lipids and proteins, 750–2400 hours for silicate limited system. On the y–axis, numbers represent the most limiting nutrient in the following way: 1=carbon, 2=energy, 3=phosphorous, 4=nitrogen, 5=silicate. Lengths of periods where no switch occurs indicates that the last least limiting nutrient was limiting until the next switch.
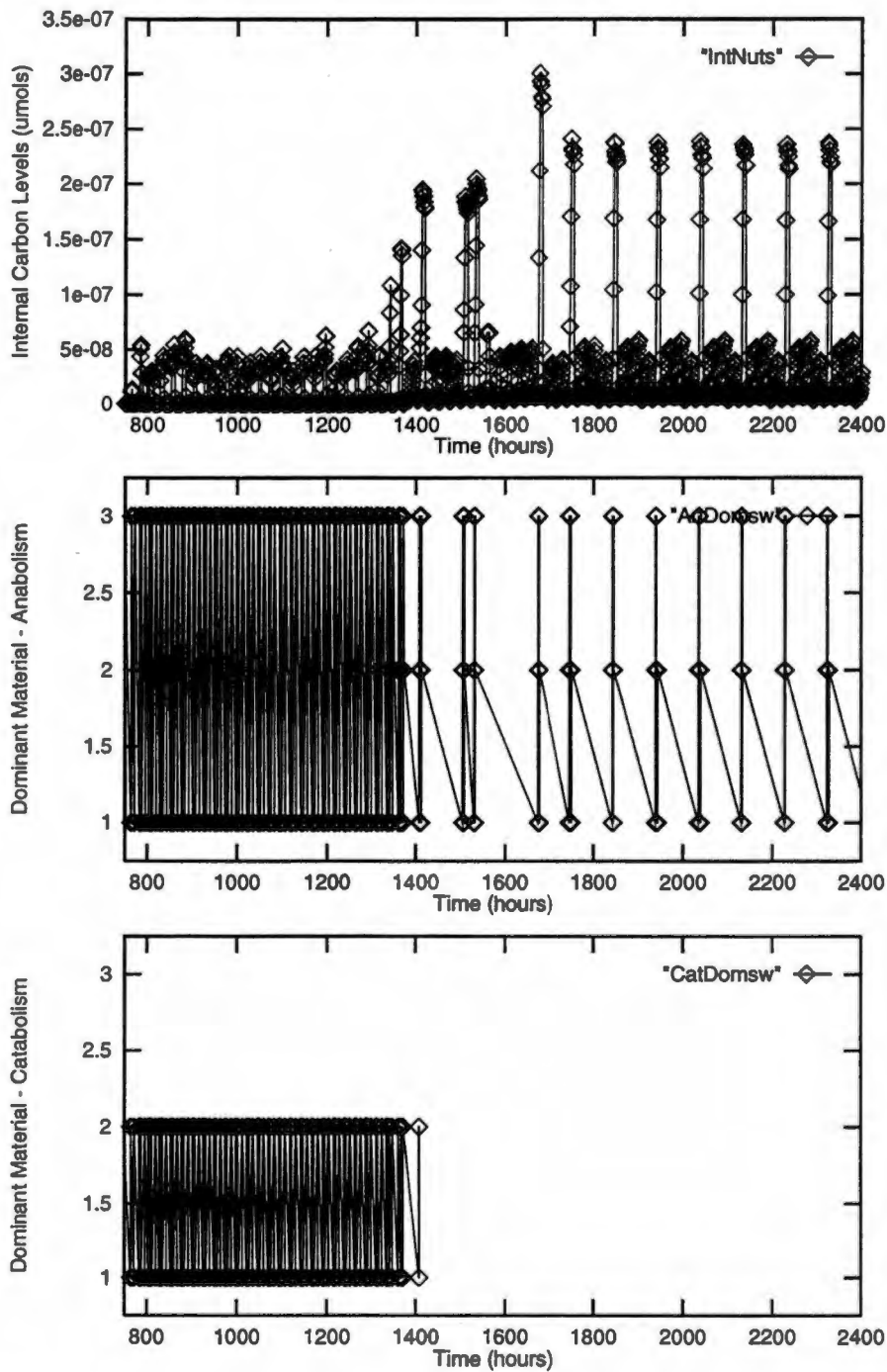
Figure 2.8: Internal Carbon levels and materials dominating anabolism and catabolism for 750–2400 hours for silicate limited system. On the y-axis for the dominant anabolism and catabolism materials, numbers correspond to the materials in the following way: 1=polysaccharides, 2=lipids, 3=proteins. Lengths of periods where no switch occurs indicates that the last least limiting material was limiting until the next switch.

2.1.2 Individual Parameter Discussions

In light of the previous discussion, it is clear that perturbations in parameters that affect processes involving carbon, phosphorous, energy or silicate could have an effect on the system in the transient state (ie. carbon, phosphorous and energy), the steady state (ie. silicate), or both. Parameters affecting the system are discussed in turn.

Parameter c'_C

c'_C is the proportionality constant for the functional response wait time given by the equation: $\frac{c'_C}{C}$, where C is the number of carbon molecules required to construct one protein, polysaccharide or lipid molecule. Increasing c'_C increases the production wait time for one of these molecules, while decreasing c'_C decreases the production wait time. Since this system is carbon limited for the production of polysaccharides, lipids and proteins in the transient state, increasing the wait time should have very little or no effect on population numbers. Similarly, decreasing the wait time should affect the transient state since carbon would be available at an increased rate. Perturbations fell in line with these expectations.

Increasing the wait time by 25% and 50% had no effect on the system (not shown). Decreasing the wait time by 25% had very little effect, shifting the system slightly to the left (not shown). Decreasing the wait time by 50%, however, radically changed the transient behavior of the system. Under these conditions,

33

significantly less carbon was required for storage and structure material production, thus enabling the population to grow larger, faster. Higher cell numbers resulted in an increased rate of depletion of external and internal phosphorous and internal energy. The perturbed system thus by-passed the phosphorous–energy limited transitional phase observed in the reference system and moved straight into a silicate limited state, achieving steady state at approximately 1200 hours. (Table 2.2 gives lengths of transients and maximum and minimum cell numbers found in the steady state. Figure 2.9 shows the effects.)

Parameter M_O

M_0 is the maintenance cost per μmol of structure or storage material in ATP molecules per volume, and is given by the following equation: $M = M_O(m_{P_r} + m_{Lp} + m_{Ps})$. Increasing M_O implies increasing maintenance demands (ATP costs), while decreasing M_O implies decreasing maintenance demands (ATP costs). It would, thus, be expected that increasing M_O would decrease population growth

Table 2.2: Perturbation information for parameter c'_C for silicate limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	–	–	–	–	–
50% increase	–	–	–	–	–
25% decrease	–	–	–	–	–
50% decrease	1200	4.897680e+05	7.732081e+04	3.851027e+04	350
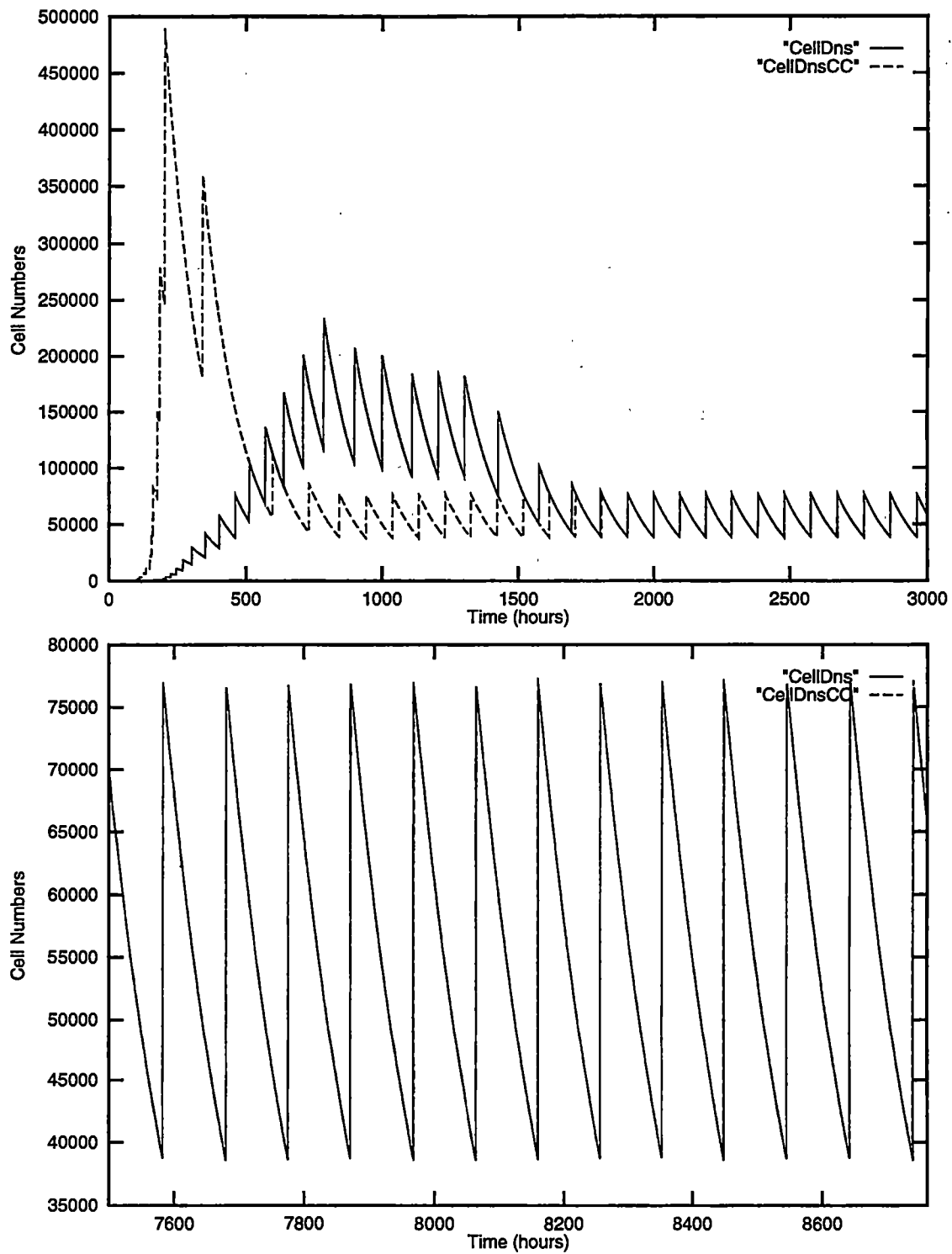
Figure 2.9: Transient and steady state behavior of 50% decrease in original c'_C value in a silicate limited system.

due to increased pressure on internal energy levels, and vice versa for decreasing M_O. The opposite holds true, however.

Increasing M_O by 25% did not have an apparent effect on the system, but increasing M_O by 50% did. Recall that when individuals cannot meet maintenance demands, anabolic activity temporarily ceases and catabolism dominates metabolism. Increasing maintenance costs resulted in early initiation of catabolic activity. This led to the replenishment of internal carbon pools, hence prompting population growth. Growth was short–lived, however, as external and internal phosphorous and internal energy could not meet population demands. The system, thus, by–passed the phosphorous–energy limited transitional phase, and moved into the silicate limited phase, achieving steady state at approximately 2000 hours.

Decreasing maintenance costs by 25% prolonged the initiation of catabolic activity, hence the replenishment of internal carbon pools. This prevented cell numbers from increasing above those supportable by phosphorous and energy. Thus, the system remained in the phosphorous–energy limited transitional phase for a longer period, not becoming silicate limited until approximately 2400 hours. Final steady state did not begin until approximately 3200 hours, however.

Decreasing maintenance costs by 50% enabled the system to continuously meet its maintenance demands. Population numbers, thus, remained carbon limited for the duration of the simulation, increasing slowly so that steady state was not

attained until approximately 4000 hours.

Table 2.3 gives lengths of transients and maximum and minimum cell numbers found in the steady state. Figure 2.10 shows the effects.

Parameter $\mu_{C,Pr}$

$\mu_{C,Pr}$ is the number of carbon molecules needed to comprise a protein molecule. Since internal protein is the sole factor for determining reproductive events, increasing or decreasing the rate of protein production results in a corresponding decrease or increase in population numbers. Since carbon limits the production of protein in the transient state, increasing or decreasing this parameter changes the rate of protein production, hence the growth of the population. 25% and 50% increases in $\mu_{C,Pr}$ lengthened the time the system experienced carbon limitation and cell numbers did not attain heights observed in the reference system. Similarly, 25% and 50% decreases in $\mu_{C,Pr}$ shortened the length of time the system was carbon, phosphorous and energy limited. Since more carbon was available for

Table 2.3: Perturbation information for parameter M_O in a silicate limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	–	–	–	–	–
50% increase	2000	2.997555e+05	7.742632e+04	3.848017e+04	436
25% decrease	3200	1.744088e+05	7.717053e+04	3.860772e+04	53
50% decrease	4000	3.814512e+04	3.746055e+04	1.867213e+04	NA
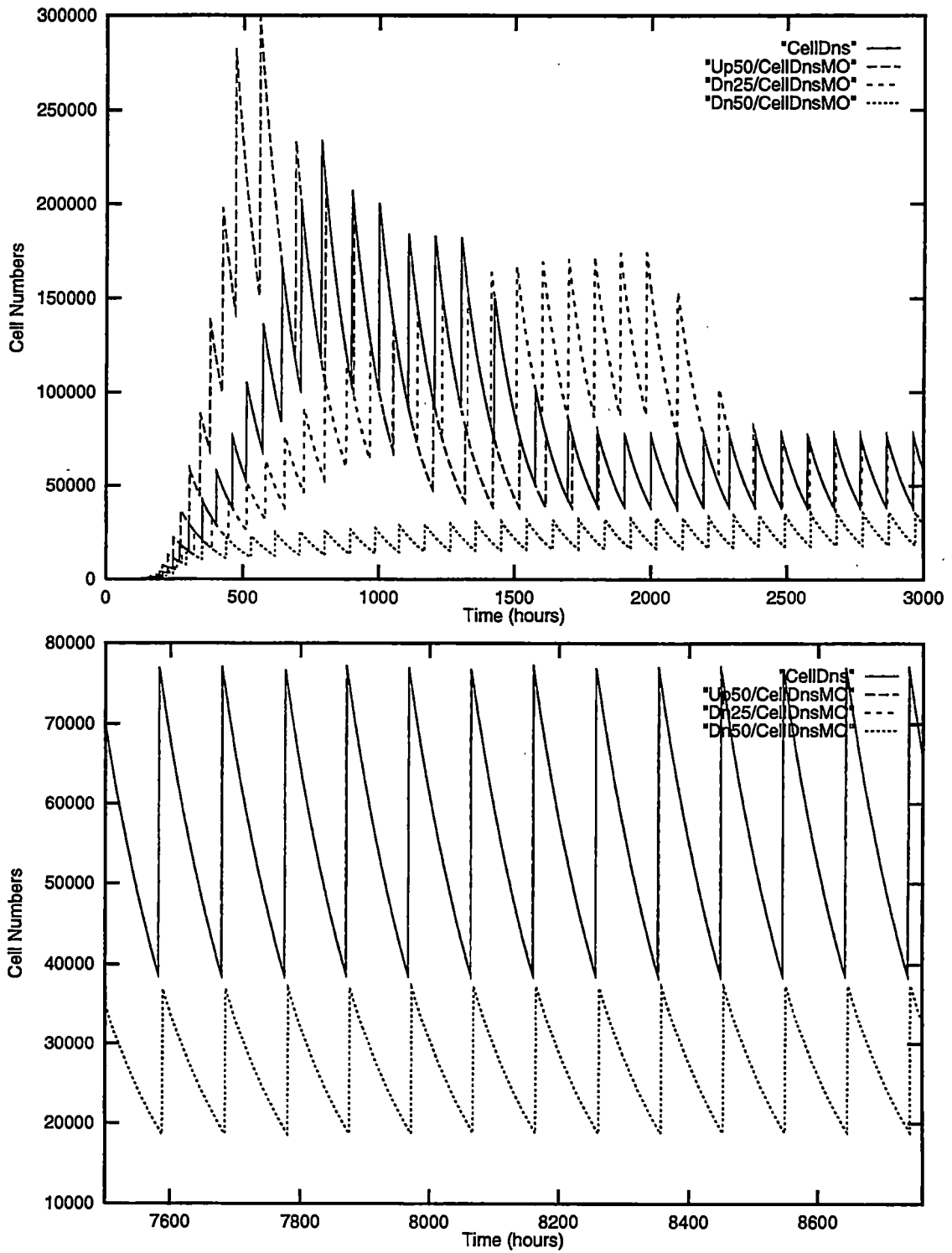
Figure 2.10: Transient and steady state behavior of 50% increase and 25% and 50% decrease in original M_O value in a silicate limited system.

38

protein production, cells were able to meet reproductive needs faster and population numbers exceeded those originally observed. This created an increased demand on external phosphorous, resulting in shortages of phosphorous and energy within individuals. The system, thus, by–passed the phosphorous–energy limiting phase, achieving steady state earlier. (Figure 2.11)

Steady states for increases of 25% and 50% were attained at 2800 and 8000 hours, respectively. Steady states for decreases of 25% and 50% were attained at 2400 and 1600 hours, respectively. (Table 2.4)

Parameter $\mu_{P,Pr}$

$\mu_{P,Pr}$ is the number of phosphorous molecules needed to comprise a protein molecule. The effect of increasing or decreasing this parameter is similar to that described for $\mu_{C,Pr}$, except that the effects do not appear until phosphorous becomes limiting for the production of lipids, hence protein. (How this works is explained in detail in the phosphorous limited section.) Thus, increases resulted in decreasing

Table 2.4: Perturbation information for parameter $\mu_{C,Pr}$ in a silicate limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	2800	1.800998e+05	7.772816e+04	3.818189e+04	1036
50% increase	8000	9.705275e+04	8.380787e+04	3.763601e+04	7662
25% decrease	2400	2.804012e+05	7.721544e+04	3.843542e+04	297
50% decrease	1600	4.188350e+05	7.766741e+04	3.851027e+04	647
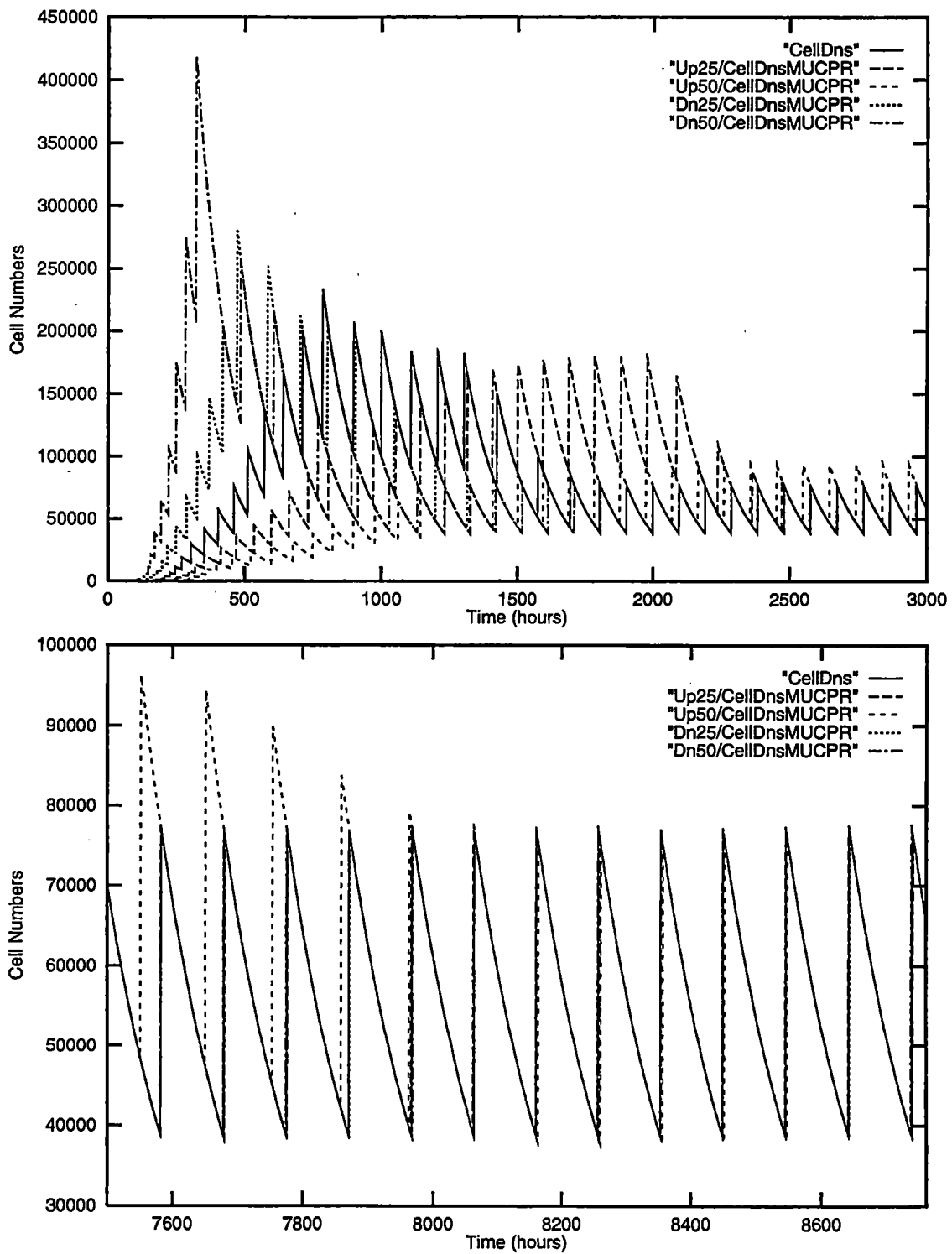
Figure 2.11: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original $\mu_{C,Pr}$ value in a silicate limited system.

40

population numbers and extending the phosphorous–energy limited phase of the system. Similarly, decreases resulted in increased numbers with a corresponding decrease in external and internal phosphorous, thus shortening the phosphorous–energy limited phase of the system. Steady states for increases of 25% and 50% were attained after 2400 and 3000 hours, respectively, while decreases of 25% and 50% were achieved after 1800 hours for both. (Table 2.5 and Figure 2.12)

Parameter n_{OP}

n_{OP} is the number of phosphate processors on the surface of the cell. It directly multiplies the nutrient uptake equation so that increasing the number of processors in effect increases phosphorous availability, while decreasing n_{OP} decreases phosphorous availability. It would thus be expected that corresponding shifts of the population to the left or right with increases or decreases in n_{OP}, respectively, would result. This was not the case, however. Only decreases of 50% affected sys-

Table 2.5: Perturbation information for parameter μ_{P,P_r} in a silicate limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	2400	1.935840e+05	7.762224e+04	3.858526e+04	527
50% increase	3000	1.515493e+05	7.757709e+04	3.860772e+04	459
25% decrease	1800	2.908487e+05	7.706537e+04	3.855511e+04	0
50% decrease	1800	3.669220e+05	7.702054e+04	3.853268e+04	22
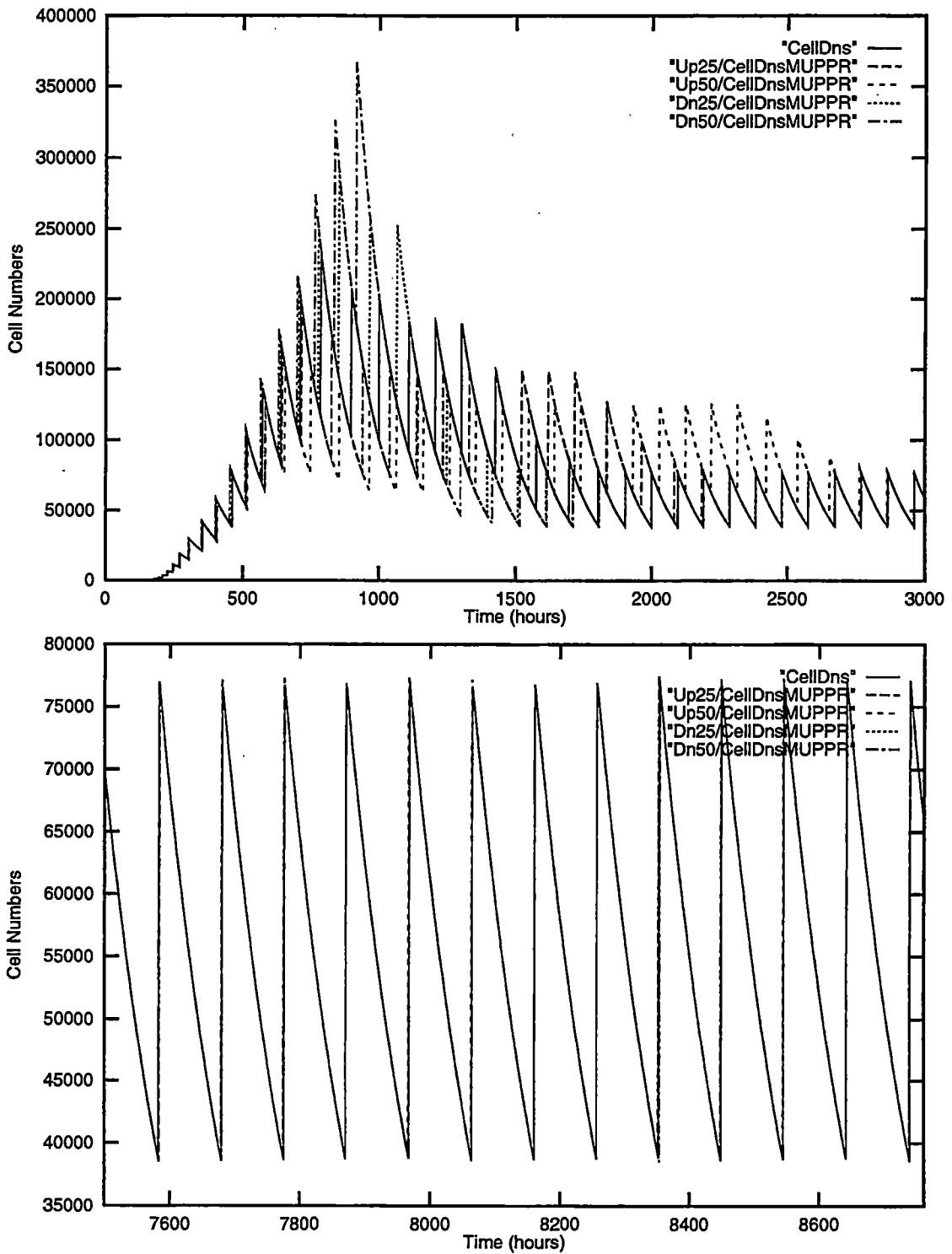
Figure 2.12: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original μ_{P,P_r} value in a silicate limited system.

tem behavior. This was due to sufficient phosphorous avialability in the external environment when individuals became limited for phosphorous. The additional processors, thus had no effect. Decreasing n_{OP} by 50%, however, extended the period of time for which individuals were phosphrous limited since they could not replinish internal levels as quickly. Figure 2.13 shows decreases of 50% in n_{OP}. Steady state was attained at 2400 hours. (Table 2.6)

Parameter Pr_{min}

Pr_{min} is the minimum amount of protein required for reproduction to take place. Increasing Pr_{min} results in decreased population numbers since individuals must accumulate higher levels of protein to divide. Conversely, decreasing Pr_{min} increases population numbers since lower levels of protein are sufficient for division. Perturbations in Pr_{min} support this, and result in changes in both the transient and steady states. (Table 2.7 and Figure 2.14)

Parameter $user3$

$user3$ ($\mu_{Si,Pr}$) is the number of silicate molecules required to construct one protein molecule. Perturbing this parameter has effects similar to those resulting from perturbations of $\mu_{C,Pr}$ and $\mu_{P,Pr}$, except that the resulting influence is not visible until silicate begins to limit the production of protein. In the reference system this occurred at approximately 1400 hours. In the perturbed systems, however,

43

Table 2.6: Perturbation information for parameter n_{OP} in a silicate limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	–	–	–	–	–
50% increase	–	–	–	–	–
25% decrease	–	–	–	–	–
50% decrease	2400	2.411943e+05	7.706537e+04	3.855511e+04	0

Table 2.7: Perturbation information for parameter Pr_{min} in a silicate limited system.

| Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	2000	1.779062e+05	6.152896e+04	3.078238e+04	NA
50% increase	2400	1.475336e+05	5.146341e+04	2.574668e+04	NA
25% decrease	2000	2.944938e+05	1.031873e+05	5.129340e+04	NA
50% decrease	2200	4.229289e+05	1.544309e+05	7.691558e+04	NA
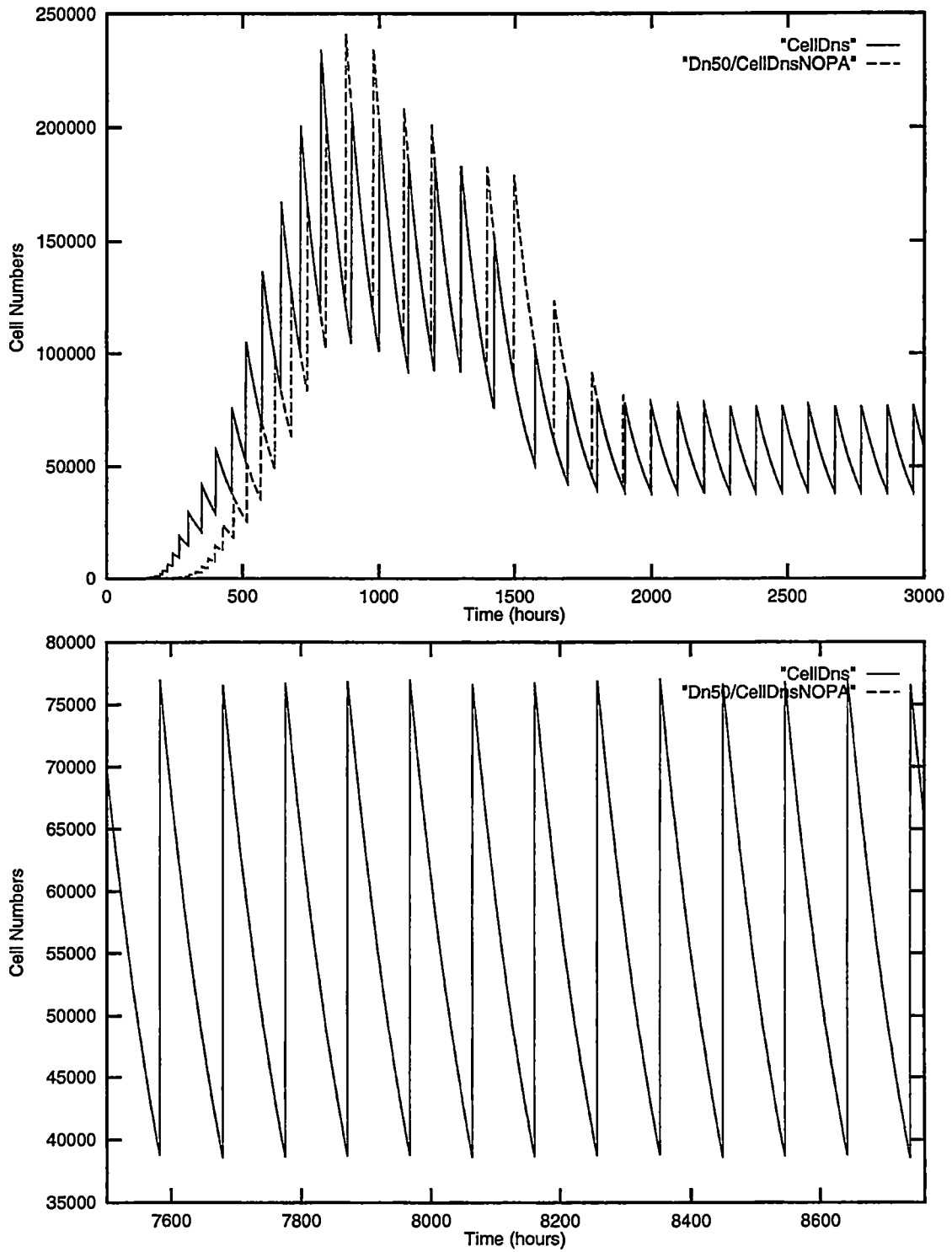
Figure 2.13: Transient and steady state behavior of 50% decrease in original n_{OP} value in a silicate limited system.

45

Figure 2.14: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original Pr_{min} value in a silicate limited system.

silicate became limited earlier or later depending on the perturbation. Increasing *user3* caused silicate limitations to occur earlier hence either shortened (25%) or eliminated (50%) the phosphorous–energy limiting phase. Decreasing *user3*, on the other hand, caused silicate limitations to occur later in the simulation, and, hence, lengthened the phosphorous–energy limited phase. All scenarios perturbed the system to new attractors. (Figure 2.15)

Steady states for 25% and 50% decreases were attained at 2000 and 1600 hours, respectively; and steady states for 25% and 50% increases were attained at 3000 and 5800 hours, respectively. (Table 2.8 and Figure 2.15)

2.2 Perturbation Results for *Skeletonema costatum* in a Phosphorous Limited System

Phosphorous inputs were chosen so that phosphorous limited population numbers in the steady state, while nitrate and silicate inputs were chosen so that neither

Table 2.8: Perturbation information for parameter *user3* in a silicate limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	2400	2.338915e+05	7.706537e+04	3.855511e+04	–
25% increase	2000	2.338915e+05	6.156477e+04	3.080646e+04	NA
50% increase	1600	2.355816e+05	5.163400e+04	2.564670e+04	NA
25% decrease	3000	2.338915e+05	1.029268e+05	5.139328e+04	NA
50% decrease	5800	2.338915e+05	1.547316e+05	7.696035e+04	NA

Figure 2.15: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original *user3* value in a silicate limited system.

48

of these limited cell numbers in the steady state. Perturbation results for increases of 25% and 50% and decreases of 25% and 50% are shown in Table 2.9. Dots in the table indicate population sensitivity to the associated parameters.

2.2.1 Dynamics of the Unperturbed Phosphorous Limited System

Much of the transient dynamics of the unperturbed phosphorous system follow those discussed previously for the silicate limited system. This is a direct result of the previously stated fact that these populations share the same initial conditions. Since this is the case, only the differences between these two systems are discussed.

0–800 Hours

The interval between 0 and 800 hours exhibited the same behavior observed in the silicate limited system; moving from a carbon limited system to a phosphorous–energy limited system. Because phosphorous inputs were reduced, phosphorous limitations occurred earlier in the system, at approximately 600 hours. In all other situations, the analysis is the same.

800–2400 Hours

Under these conditions, phosphorous does not fully limit population growth until approximately 800 hours. This is not directly evident in the system. Recall that limiting the system implies hindering the production of protein. Phosphorous

Table 2.9: Perturbation Results for *Skeletonema costatum* in a Phosphorous Limited System

Parameter	Definition	Up 25%	Up 50%	Dn 25%	Dn 50%
c'_C	functional response proportionality constant for carbon wait time				•
c_{OP}	uptake wait time proportionality constant for phosphorous	•	•		•
M_0	cost of maintenance		•	•	•
$\mu_{C,Pr}$	number of carbon molecules needed to construct a Protein molecule	•	•	•	•
$\mu_{P,Pr}$	number of phosphorous molecules needed to construct Protein molecule	•	•	•	•
n_{OP}	number of uptake processors on the cell suface		•		•
Pr_{min}	minimum amount of protein needed for cell division	•	•	•	•

does not directly affect protein production, because its associated $\kappa_{P,Pr}$ multiplier is zero. Clearly, examining the most limiting nutrient for protein would not show phosphorous as the culprit, but does show both energy and carbon as limiting. This is due to negative feedback as a result of phosphorous limitations. Three phosphorous molecules are required for each energy molecule, and energy is required for maintenance. When phosphorous is limited, energy becomes limited, subsequently cells are unable to meet maintenance demands. A release of carbon from storage to internal pools increases carbon availability. Cells are thus able to grow. But, because they are phosphorous limited, they again cannot meet maintenance demands, and the cycle continues. This interaction with carbon is a direct result of fact that carbon initially limits protein production. The scenario

would be the same with any of the other nutrients if they also initially limited the system.

Note that this process is similar to the one previously described in the silicate limited situations but occurs for a different reason. In the silicate limited situation, excessive phosphorous was available in the environment causing a growth spurt. This spurt resulted in the depletion of energy at which point catabolism became dominant in metabolic activity, and bound carbon was released. The cycle ceased when silicate became the dominant limiting factor for protein production. In this case, however, the cycle never stops. And, although there is an initial growth spurt due to excessive external phosphorous, long-term inputs eventually limit production. The most limiting nutrients for the production of polysaccharides, lipids and proteins are shown in Figure 2.16.

2.2.2 Individual Parameter Discussions

Many of the parameters having affects on the population under the phosphorous limiting conditions are the same as those listed for the population under silicate limiting conditions. Again, this is a direct result of the fact that these systems were simulated under the same initial conditions. Thus, many of the affects of the perturbations observed in the transient behavior of the silicate limited system are also expressed in this system. Since the resulting influence of these parameters is the same in both systems, they are not fully discussed, although the transient

51
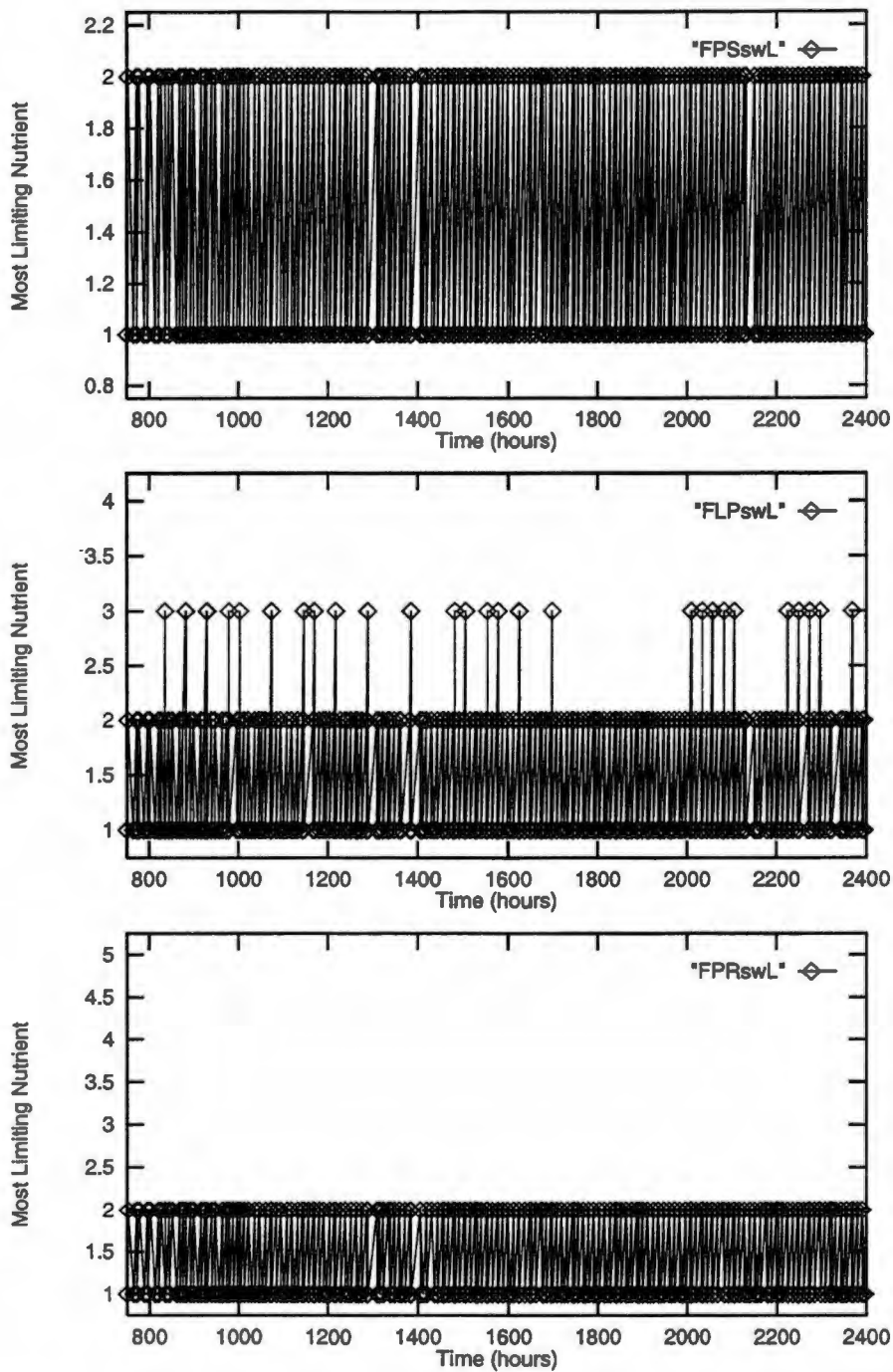
Figure 2.16: Most limiting nutrient for polysaccharides, lipids and proteins, 750–2400 hours in a phosphorous limited system. 1=carbon, 2=energy, 3=phosphorous, 4=nitrogen, 5=silicate. Lengths of periods where no switch occurs indicates that the last least limiting nutrient was limiting until the next switch.

52

results are tabulated and graphically shown. Full discussion of parameters not appearing in the silicate limited set are discussed below.

Parameter c'_C

The resulting effect of perturbing c'_C under phosphorous limiting conditions is the same as that discussed in the silicate limited situation. It took this system slightly longer to reach steady state than in the silicate limited case, attaining it at 1400 hours. The slight increase in time is due to the phosphorous limited conditions. (Table 2.10 gives lengths of transients and maximum and minimum cell numbers found in the steady state. Figure 2.17 shows the effects.)

Parameter c_{OP}

c_{OP} is the wait time proportionality constant for the uptake of phosphate. It is included in the nutrient uptake equation and its equation is given by: $\frac{c_{OP}}{\phi_{OP}}$ where ϕ_{OP} is the external concentration of phosphate. Recall from the analysis of the

Table 2.10: Perturbation information for parameter c'_C in a phosphorous limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	–	–	–	–	–
50% increase	–	–	–	–	–
25% decrease	–	–	–	–	–
50% decrease	1400	4.758639e+05	6.200963e+04	3.060325e+04	887
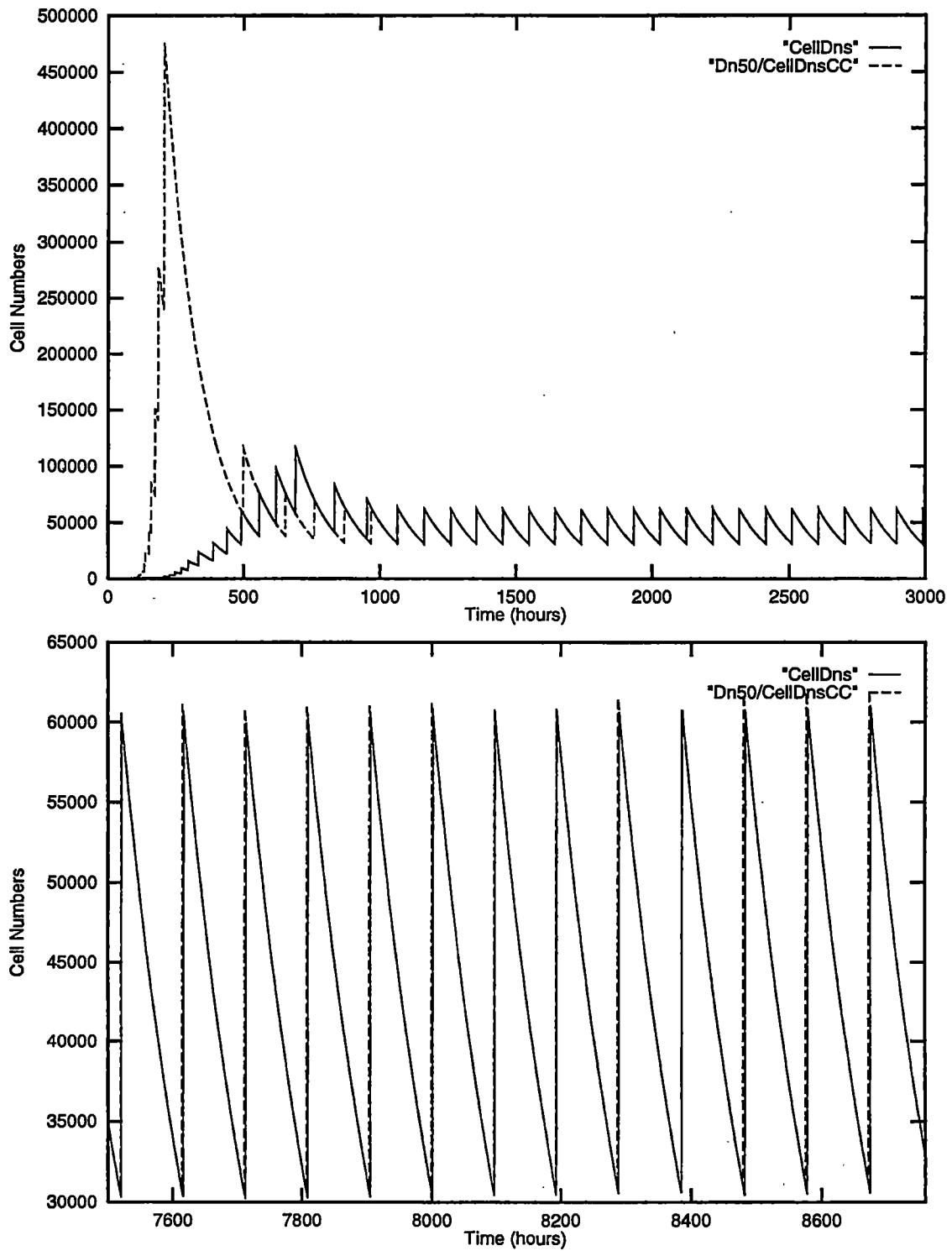
Figure 2.17: Transient and steady state behavior of 50% increase in original c_C' value in a phosphorous limited system.

silicate limited system that at approximately 200 hours, internal levels of phosphorous are no longer sufficient to meet cell demands and the population almost immediately depletes external phosphate levels. Observe also, that under the same population pressures in the other two cases, c_{OP} does not have an apparent affect on the population when perturbed. This is due to the fact that the cumulative levels of phosphorous over the preceeding 200 hours are much higher in those systems than in this one. Thus, even though there is an immediate decrease in external phosphates, the external levels do not fall below the level for which changes in the wait time are affected. In this case, however, they do, so that increasing or decreasing c_{OP} causes a shift to the right or left in the transient behavior, respectively. The effect of perturbing c_{OP} is not evident in the steady states. During this phase, external levels are so low that external concentrations dominate this part of the uptake equation.

Steady states for all perturbations occurred at approximately 1600 hours. (Table 2.11 and Figure 2.18)

Parameter M_O

The resulting effect of perturbing M_O under phosphorous limiting conditions is the same as that discussed in the silicate limited situation. For increases in 25% the system again did not respond. For increases in 50% steady state was achieved at approximately 1600 hours, as compared to 2000 hours in the silicate limited

Table 2.11: Perturbation information for parameter c_{OP} in a phosphorous limited system.

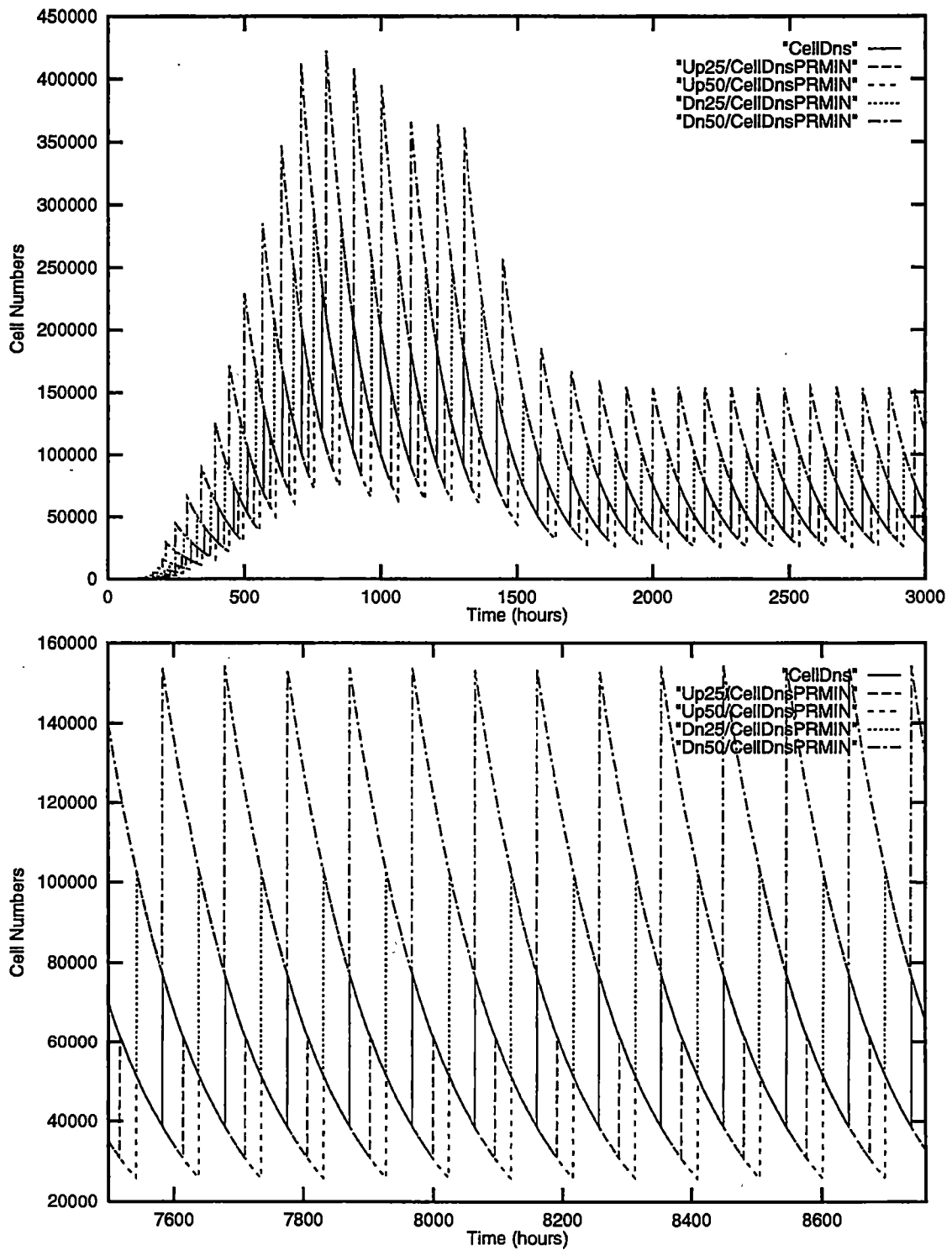|	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	1600	1.070573e+05	6.068458e+04	3.024810e+04	83
50% increase	1600	1.221085e+05	6.068458e+04	3.024810e+04	83
25% decrease	–	–	–	–	–
50% decrease	1600	1.045660e+05	6.068458e+04	3.024810e+04	83

Figure 2.18: Transient and steady state behavior of 25% and 50% increase and 50% decrease in original c_{OP} value in a phosphorous limited system.

system. The earlier attainment of steady state in this case was due to the reduction in phosphate availability, thus the population by-passed the short-lived phosphorous-energy limiting period observed in the silicate limited system. Decreases of 25% resulted in earlier attainment of steady state as well, achieving it at approximately 2200 hours, as compared to the 3200 hours observed in the silicate limited system. Again, high phosphorous availability in the silicate limited state prolonged the phosphorous-energy limited period, while in this case, lower phosphorous availability shortened it. Finally, decreasing M_O by 50% resulted in the achievement of steady state at approximately 4000 hours, the same time as that observed in the silicate limited state. This is to be expected in all cases, however, since the dynamics of the system under these conditions are dictated by limitations in carbon. (Table 2.12 and Figure 2.19)

Table 2.12: Perturbation information for parameter M_O in a phosphorous limited system.

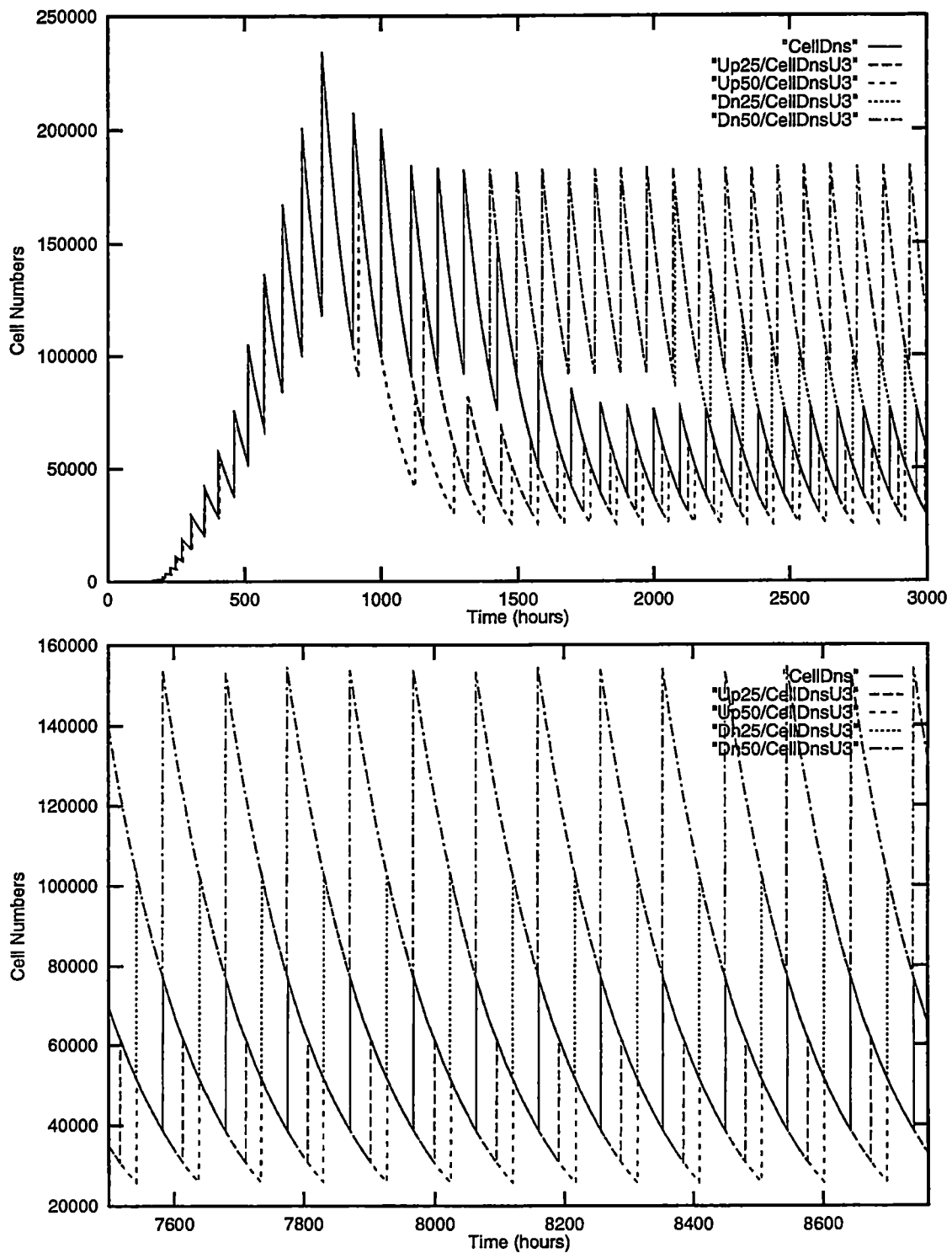| Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	–	–	–	–	–
50% increase	1600	1.781490e+05	6.156477e+04	3.068683e+04	359
25% decrease	2200	6.366274e+04	5.789313e+04	2.892973e+04	2896
50% decrease	4000	1.495779e+04	1.235306e+04	6.172938e+03	NA
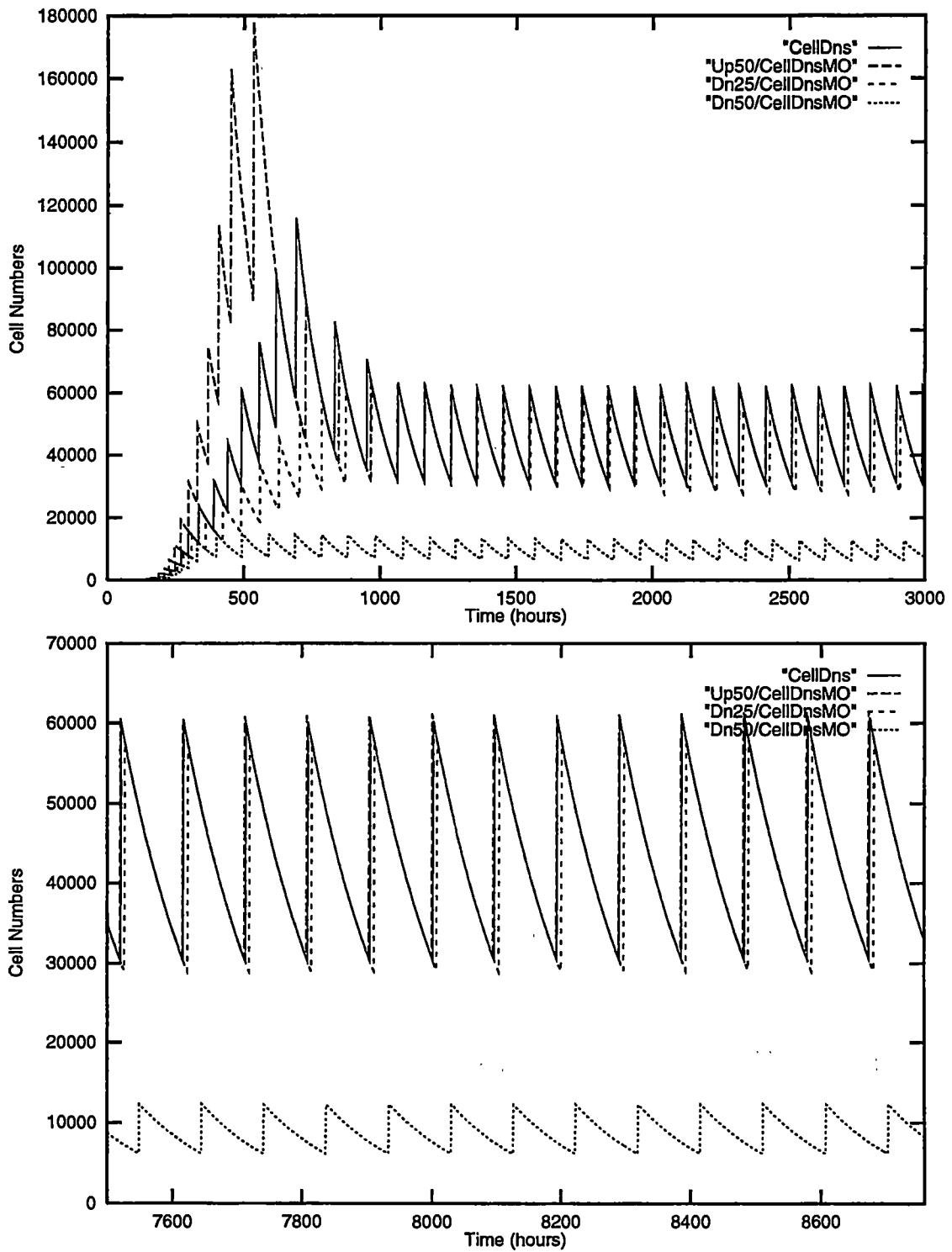
Figure 2.19: Transient and steady state behavior of 50% increase and 25% and 50% decrease in original M_O value in a phosphorous limited system.

59

Parameter $\mu_{C,Pr}$

The resulting effect of perturbing $\mu_{C,Pr}$ under phosphorous limiting conditions is the same as that described in the silicate limited situation for increases of 25% and decreases of both 25% and 50%. It is not the same for increases of 50%, however, because of the role carbon plays in this system. Recall that phosphorous limitations result in energy limitations, which results in catabolism. This releases carbon and the cells grow. They again become phosphorous limited, and the cycle continues. Under these conditions, carbon plays a dominant role in limiting protein production. Increasing the number of carbon molecules to produce a protein molecule inhibits reproduction thus population numbers. This is evident in increases of both 25% and 50%. In both cases, population numbers grow slowly. In the 25% increase case, the population eventually reaches the levels exhibited in the reference system. In the 50% increase case, however, the population remains carbon limited, attaining maximum heights below that observed in the reference system. (Table 2.13 and Figure 2.20) Steady states for increases of 25% and 50% were 3200 and 1200, respectively. Steady states for decreases of 25% and 50% were 1200 and 2200, respectively. These differed from that reported for the silicate limited system because of the role phosphorous plays in both.

Table 2.13: Perturbation information for parameter $\mu_{C,Pr}$ in a phosphorous limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	3200	6.954299e+04	5.915723e+04	2.956141e+04	923
50% increase	1200	3.555141e+04	3.367150e+04	1.682596e+04	NA
25% decrease	1200	1.666465e+05	6.156477e+04	3.052601e+04	520
50% decrease	2200	2.471339e+05	6.200963e+04	3.090857e+04	582
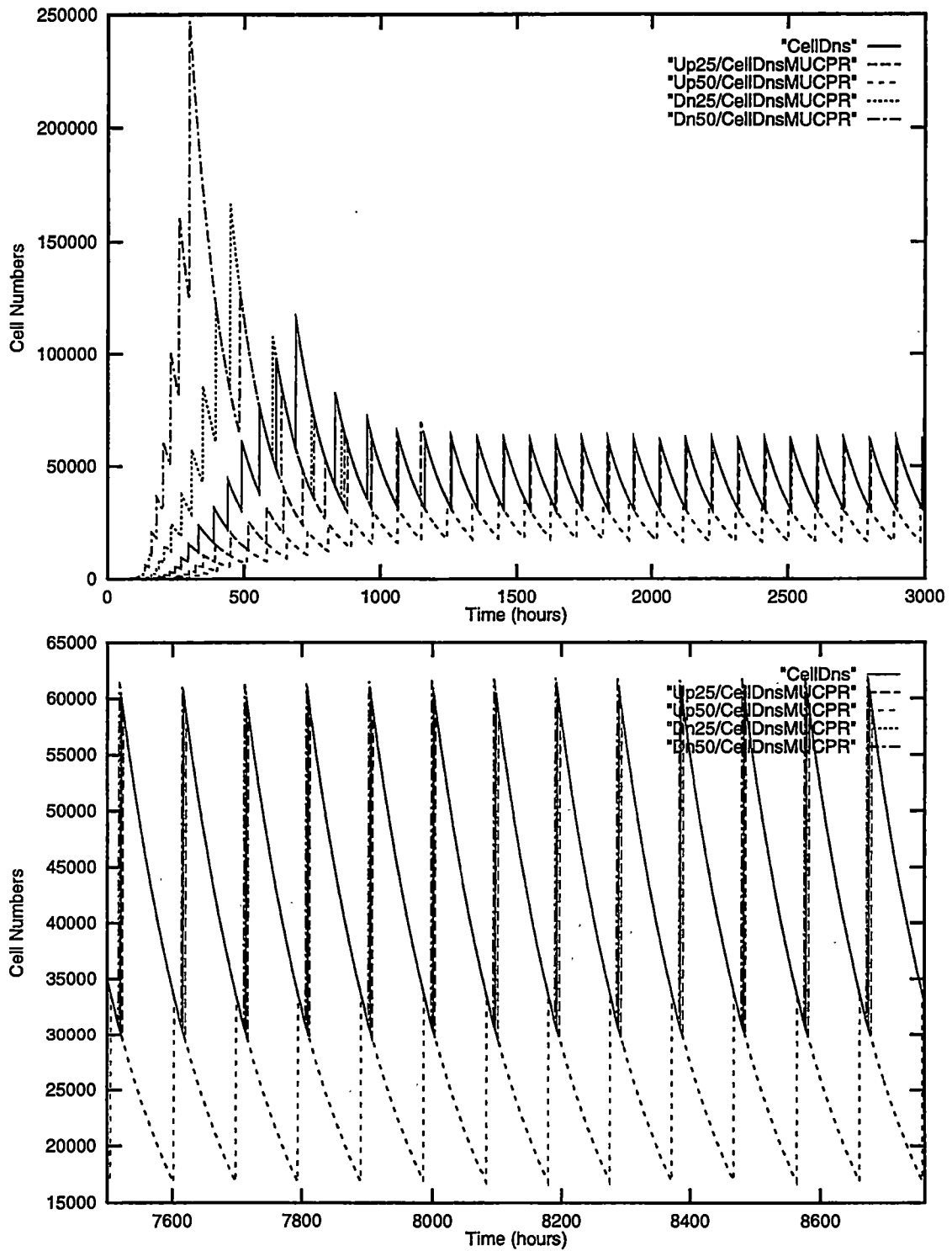
Figure 2.20: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original $\mu_{C,Pr}$ value in a phosphorous limited system.

Parameter μ_{P,P_r}

The resulting effect of perturbing μ_{P,P_r} is the same as that described in the silicate limited situation. The resulting behavior is different, however, since phosphorous limits this system. Observe that phosphorous can be limiting in both the internal and external environments but that it is not until phosphorous begins to affect the production of protein that perturbations in this parameter manifest themselves in the population. Hence, perturbations in either direction show small changes in the transient but all perturb the system to new steady state levels, evidenced by maximum and minimum figures presented in Table 2.14 and seen in Figure 2.21. Steady states are attained at 2200 and 2000 for increases in 25% and 50%, respectively, and 1400 and 4500 for decreases in 25% and 50%, respectively.

Table 2.14: Perturbation information for parameter μ_{P,P_r} in a phosphorous limited system.

	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	2200	9.054243e+04	4.976930e+04	2.477360e+04	NA
50% increase	2000	8.797199e+04	4.157079e+04	2.080166e+04	NA
25% decrease	1400	1.265845e+05	8.298000e+04	4.130484e+04	NA
50% decrease	4500	1.741046e+05	9.262373e+04	4.634809e+04	NA

Figure 2.21: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original μ_{P,P_r} value in a phosphorous limited system.

Parameter n_{OP}

The resulting effect of perturbing n_{OP} in this system is the expected result described in the silicate limited case. Perturbations of 25% either up or down were insufficient to make a difference in the dynamics of the system, but perturbations of 50% both up and down were. Increasing n_{OP} by 50% exhibited affects on this population and not on the silicate population because the accumulated phosphorous in this system over the prior 200 hours was less in this system than in the silicate limited system due to the decreased level of inputs. Decreasing n_{OP} by 50% had the same affect as in the silicate limited case. Steady states were attained at 1400 hours for increases of 50% and 2000 hours for decreases of 50%. (Table 2.15 and Figure 2.22)

Parameter Pr_{min}

The resulting effect of perturbing Pr_{min} is the same as that discussed in the silicate limited situations. Although this is true, the transient and steady state behavior of the systems cannot be compared since these dynamics are intimately tied with the limiting nutrient. Steady states are listed in Table 2.16 and illustrated in Figure 2.23.

Table 2.15: Perturbation information for parameter n_{OP} in a phosphorous limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	–	–	–	–	–
50% increase	1400	1.038159e+05	6.068458e+04	3.024810e+04	83
25% decrease	–	–	–	–	–
50% decrease	2000	1.268310e+05	6.068458e+04	3.024810e+04	83

Table 2.16: Perturbation information for parameter Pr_{min} in a phosphorous limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1600	1.158811e+05	6.076739e+04	3.024810e+04	–
25% increase	1600	9.006791e+04	4.967257e+04	2.477360e+04	NA
50% increase	1800	7.415545e+04	4.157079e+04	2.073293e+04	NA
25% decrease	1800	1.448596e+05	8.314159e+04	4.146587e+04	NA
50% decrease	1600	2.031950e+05	1.213692e+05	6.049621e+04	NA
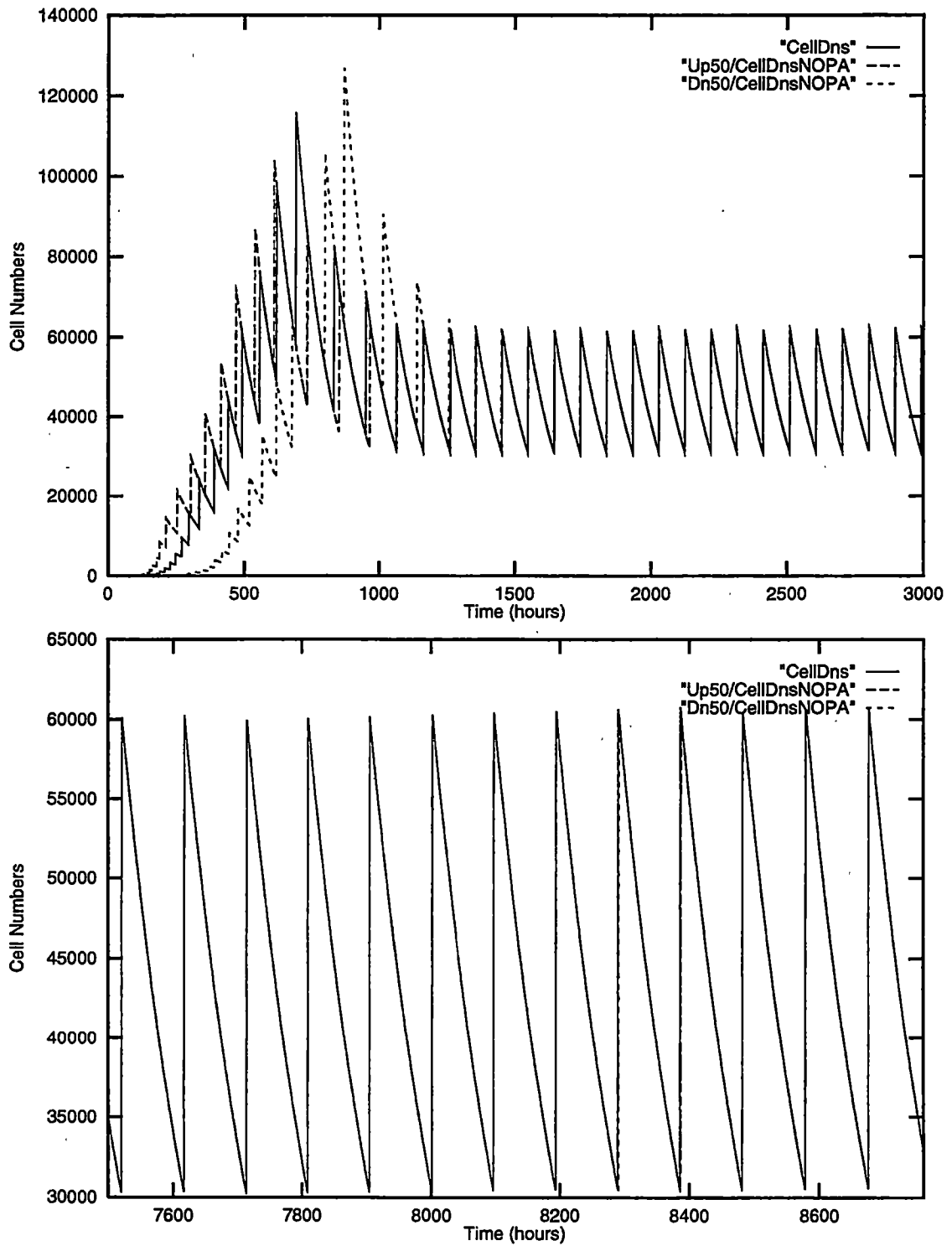
Figure 2.22: Transient and steady state behavior of 50% increase and 50% decrease in original n_{OP} value in a phosphorous limited system.

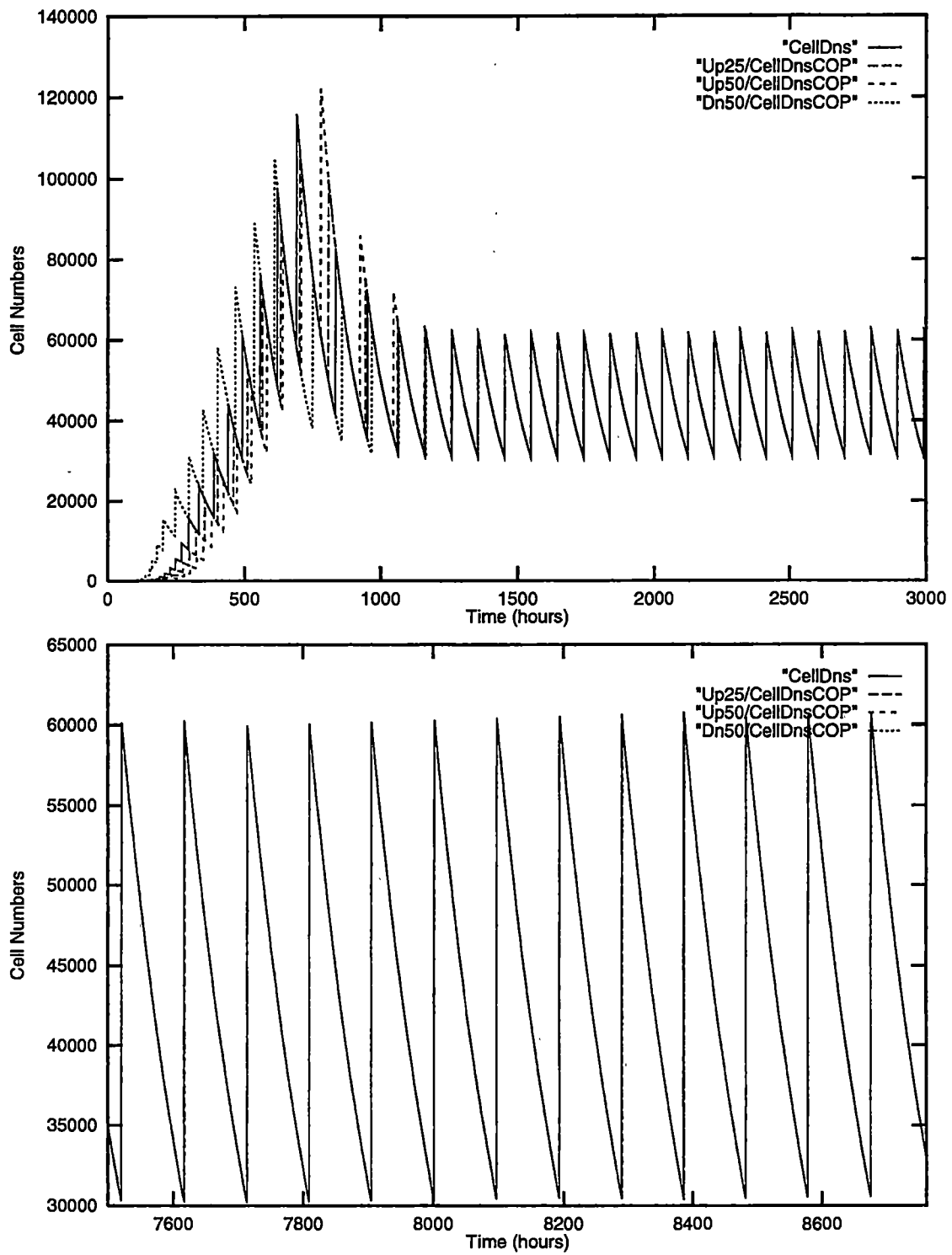Figure 2.23: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original Pr_{min} value in a phosphorous limited system.

2.3 Perturbation Results for *Skeletonema costatum* in a Nitrogen Limited System

Nitrogen inputs were chosen so that nitrogen limited population numbers in the steady state, while phosphate and silicate inputs were chosen so that neither of these limited cell numbers in the steady state. Perturbation results for increases of 25% and 50% and decreases of 25% and 50% are shown in Table 2.17. Dots in the table indicate population sensitivity to the associated parameters.

2.3.1 Dynamics of the Unperturbed Nitrogen Limited System

0–800 Hours

Like the phosphorous system, the behavior of this system for the first 800 hours is the same as that for the silicate limited system, so it is not discussed here.

800–2400 Hours

At approximately 800 hours, the population experiences a growth spurt that not only depletes internal phosphorous and energy, but also depletes both external and internal nitrogen levels. The population, thus, by–passes the phosphorous–energy limiting phase, immediately becoming nitrogen limited. Figure 2.24 shows the most limiting factors for polysaccharide, lipid and protein production, where only the hours of 750–1000 are shown for protein. The system becomes limited

Table 2.17: Perturbation Results for *Skeletonema costatum* in a Nitrogen Limited System

Parameter	Definition	Up 25%	Up 50%	Dn 25%	Dn 50%
c'_C	functional response proportionality constant for carbon wait time				●
M_0	cost of maintenance		●	●	●
$\mu_{C,Pr}$	number of carbon molecules needed to construct a Protein molecule	●	●	●	●
$\mu_{N,Pr}$	number of nitrogen molecules needed to construct a Protein molecule	●	●	●	●
n_{OP}	number of uptake processors on the cell suface				●
Pr_{min}	minimum amount of protein needed for cell division	●	●	●	●
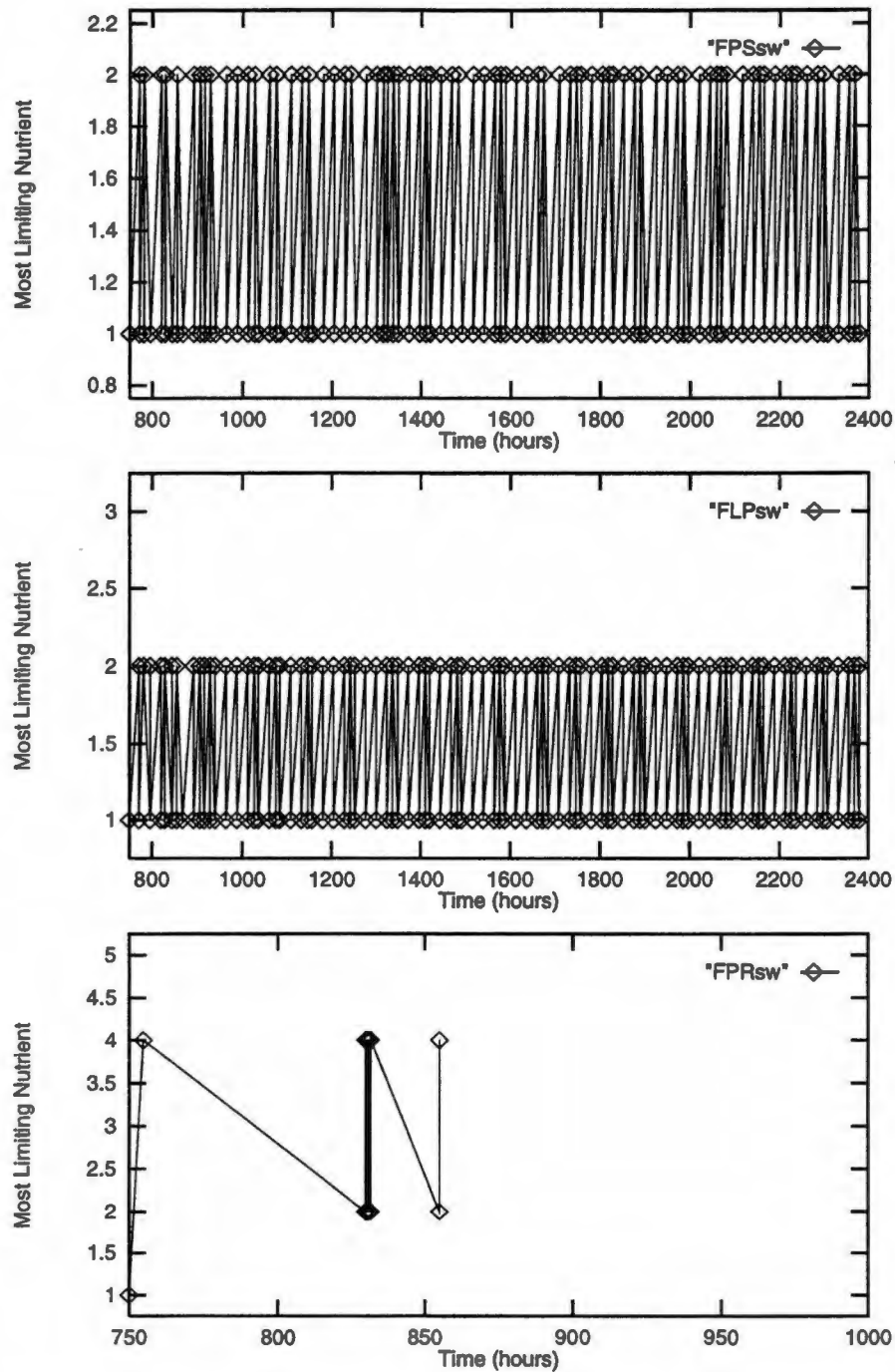
Figure 2.24: Most limiting nutrient for polysaccharides, lipids and proteins, 750–2400 hours for a nitrogen limited system. 1=carbon, 2=energy, 3=phosphorous, 4=nitrogen, 5=silicate. Lengths of periods where no switch occurs indicates that the last least limiting nutrient was limiting until the next switch.

71

for nitrogen at approximately 850 hours and remains there for the duration of the simulation, hence is the primary limiting factor for population growth.

2.3.2 Individual Parameter Discussions

Many of the parameters having effects on the population under the nitrogen limiting conditions are the same as those listed for the population under both the silicate and phosphate limiting conditions. Thus, many of the effects of the perturbations observed in the transient behavior of the silicate limited system are also expressed in this system. Since the resulting influence of these parameters is the same in both systems, they are not fully discussed, although the transient results are tabulated and graphically shown. Full discussion of parameters not appearing in the silicate limited set are discussed below.

Parameter c'_C

The resulting effect of perturbing c'_C under nitrogen limiting conditions is the same as that discussed in both the phosphate and silicate limited situations. This system attains steady state at approximately 1400 hours. (Table 2.18 and Figure 2.25)

Table 2.18: Perturbation information for parameter c'_C in a nitrogen limited system.

| Perturbation | Transient | | Steady State | | |
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase		–	–	–	–
50% increase	–	–	–	–	–
25% decrease	–	–	–	–	–
50% decrease	1400	2.531453e+04	4.054809e+04	1.884368e+04	276
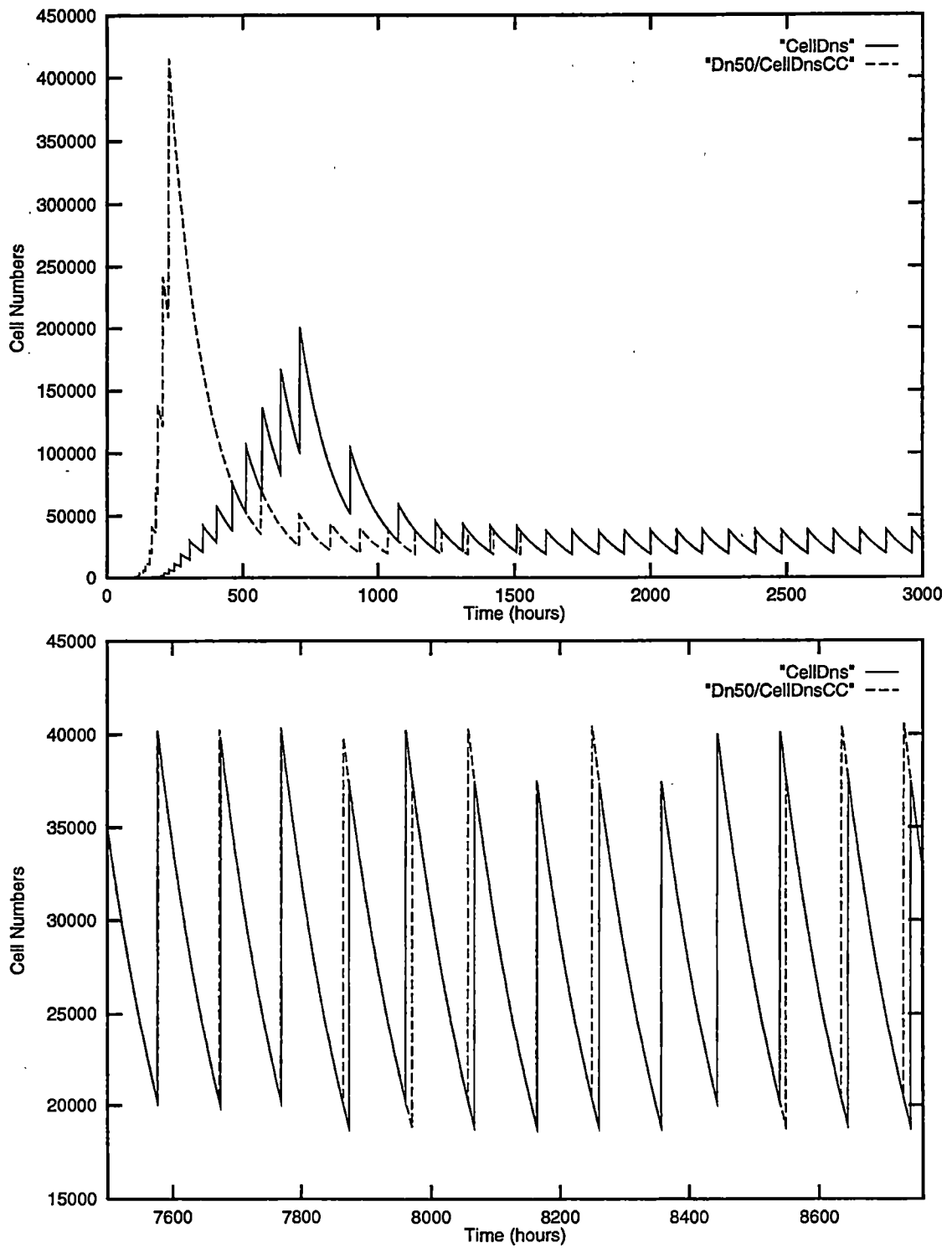
Figure 2.25: Transient and steady state behavior of 50% increase in original c'_C value in a nitrogen limited system.

74

Parameter M_O

The resulting effect of perturbing M_O under nitrogen limiting conditions is the same as that discussed in the silicate and phosphate limited situations. For increases in 25% the system did not respond. For increases in 50% steady state was achieved at approximately 1400 hours, as compared to 2000 hours for the silicate limited system and 1600 hours for the phosphate limited system. The earlier attainment of steady state in this case was due to the fact that this system by–passes the phosphorous–energy limiting periods observed in both the other systems. Decreases of 25% resulted in later attainment of steady state, achieving it at approximately 5600 hours, as compared to the 3200 hours observed in the silicate limited system and 2200 hours observed in the phosphate limited system. Finally, decreasing M_O by 50% resulted in the achievement of steady state at approximately 4000 hours, the same time as that observed in both the silicate limited and phosphate limited states. This is to be expected since the dynamics of the system under these conditions is dictated by limitations in carbon. Observe that in this case the population settles to a steady state at approximately the same levels as that of the reference system. It appears that the system is nitrogen limited. In fact both the reference nitrogen limited system and the carbon limited system (resulting from the 50% decrease in M_O) share the same steady state but for different reasons. (Table 2.19 and Figure 2.26)

Table 2.19: Perturbation information for parameter M_O in a nitrogen limited system.

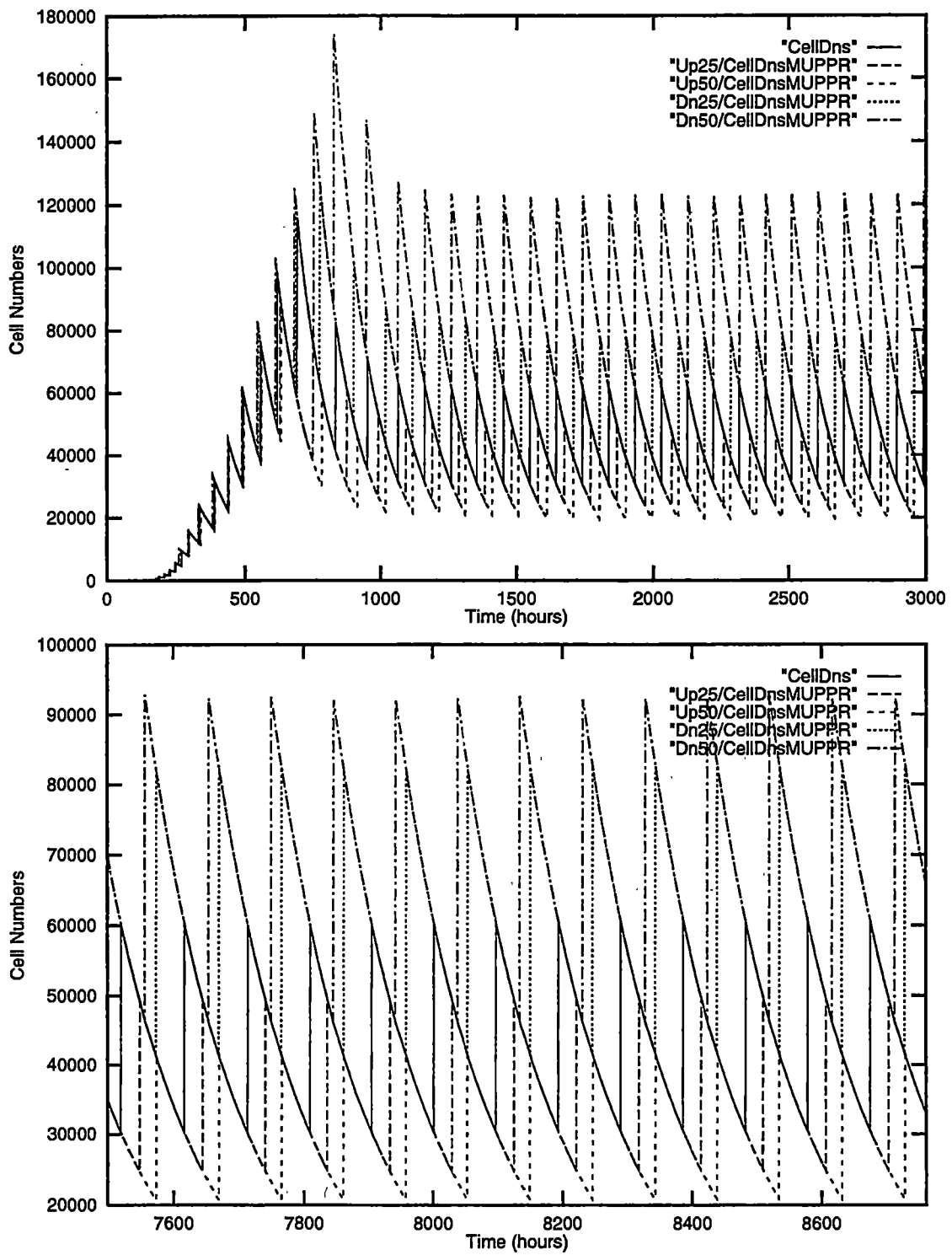| Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase	–	–	–	–	–
50% increase	1400	2.824274e+05	4.060342e+04	1.886562e+04	309
25% decrease	5600	1.441843e+05	4.044575e+04	1.977923e+04	763
50% decrease	4000	3.814512e+04	3.746055e+04	1.867213e+04	NA
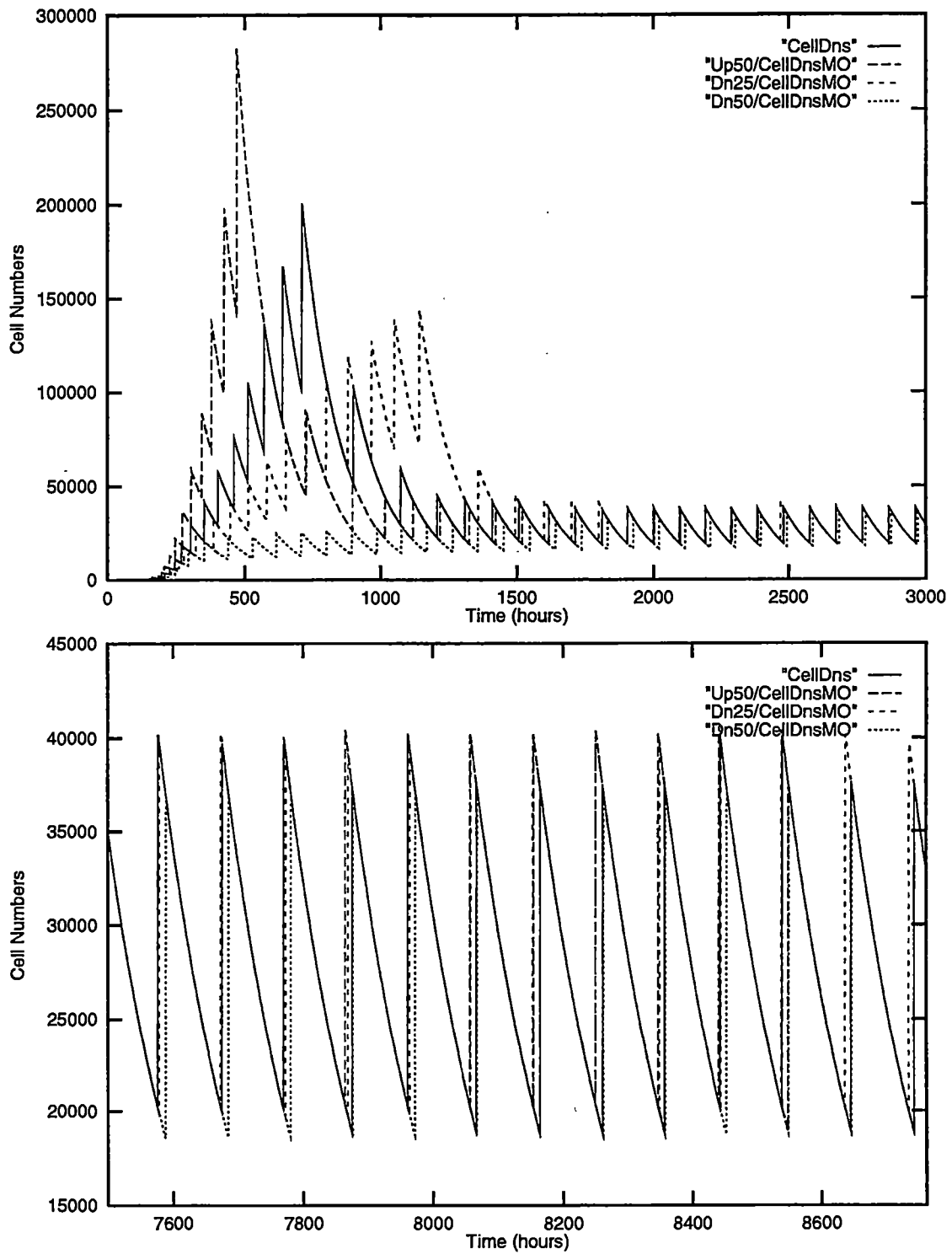
Figure 2.26: Transient and steady state behavior of 50% increase and 25% and 50% decrease in original M_O value in a nitrogen limited system.

Parameter $\mu_{C,Pr}$

The resulting effect of perturbing $\mu_{C,Pr}$ under nitrogen limiting conditions is the same as that discussed for the silicate limited conditions. Steady states are attained at 1800 and 2800 hours for increases of 25% and 50%, and 6500 and 7000 hours for decreases of 25% and 50%. These differ from those observed in the silicate limited system due to the fact that this system does not experience the phosphorous–energy limiting state observed in the silicate limited system. Thus, decreases in $\mu_{C,Pr}$ allow the system to achieve steady state earlier, while increases in $\mu_{C,Pr}$ prolong carbon limitations and cause the system to attain steady state later. (Table 2.20 and Figure 2.27)

Parameter $\mu_{N,Pr}$

$\mu_{N,Pr}$ is the number of nitrogen molecules needed to construct one protein molecule. Increasing $\mu_{N,Pr}$ decreases nitrogen availability for the construction of protein, thus hinders population growth. Decreasing $\mu_{N,Pr}$ increases nitrogen availability

Table 2.20: Perturbation information for parameter $\mu_{C,Pr}$ in a nitrogen limited system.

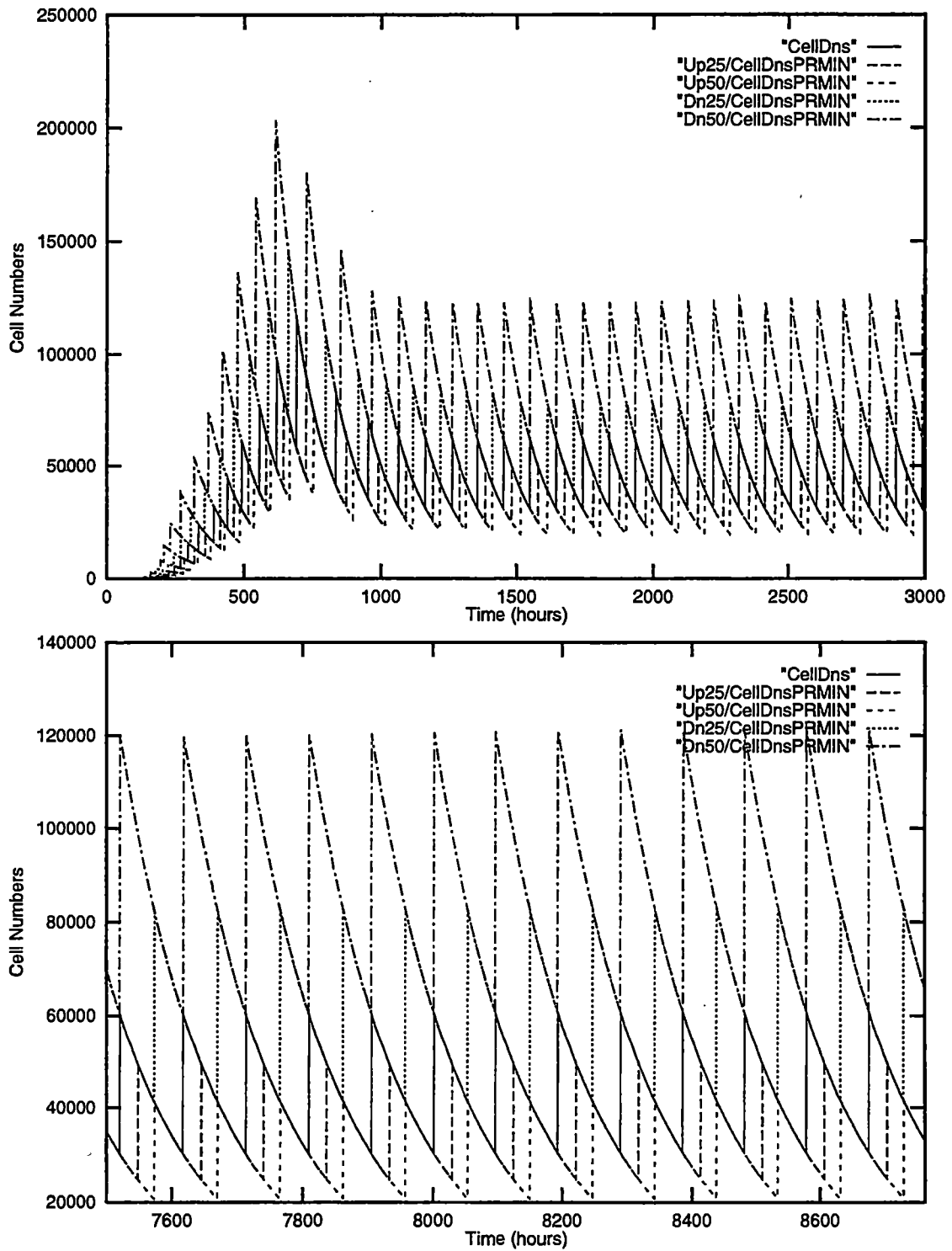|	Transient		Steady State		
Perturbation	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase	1800	1.441843e+05	4.039063e+04	1.881800e+04	144
50% increase	2800	9.171215e+04	4.046928e+04	1.880706e+04	233
25% decrease	6500	2.804012e+05	4.054809e+04	1.881800e+04	301
50% decrease	7000	2.753190e+05	4.028868e+04	1.880706e+04	53
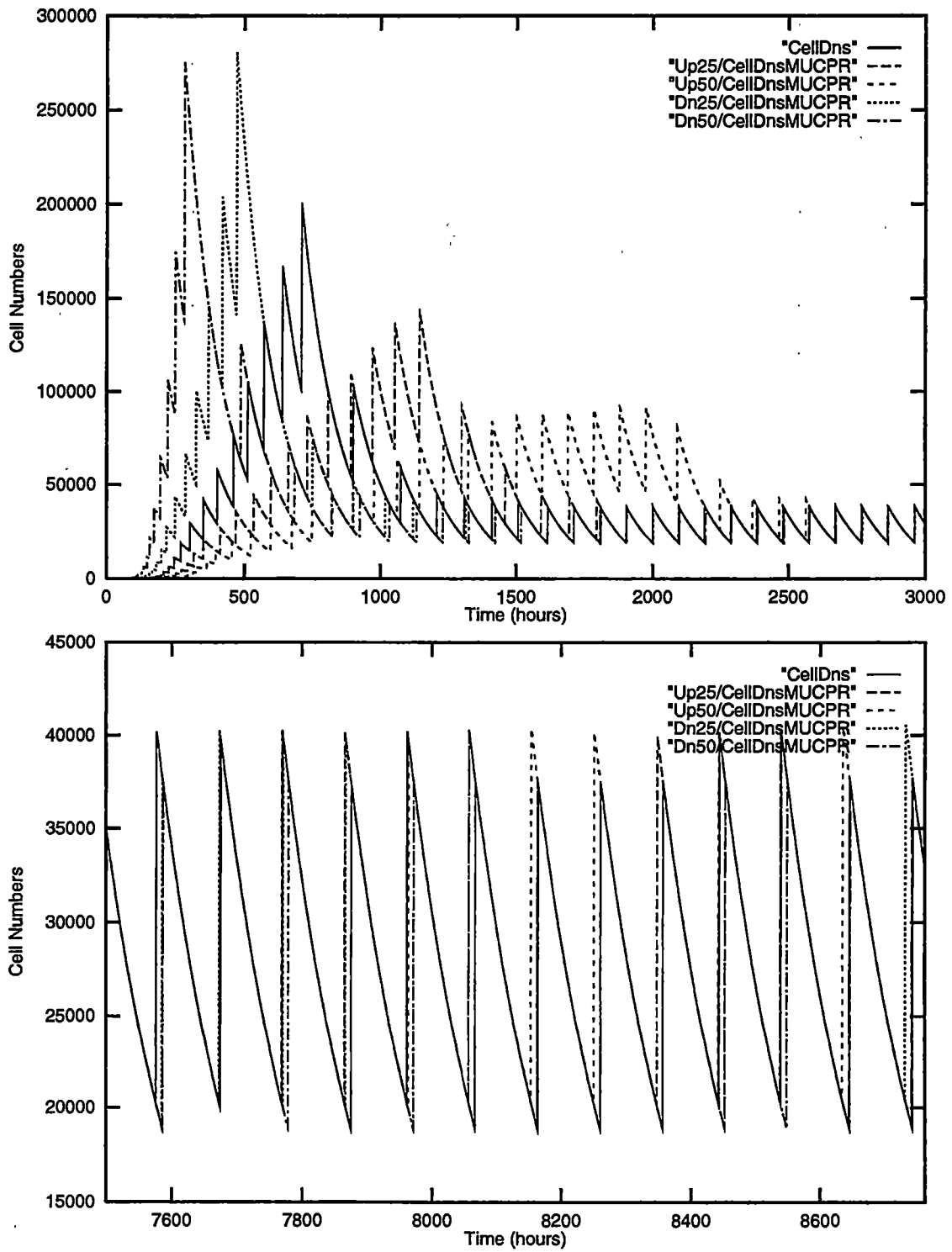
Figure 2.27: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original $\mu_{C,Pr}$ value in a nitrogen limited system.

for protein production hence promotes population growth. Observe that perturbations do not manifest themselves in the dynamics until nitrogen limits the production of protein, around 800 hours. (Figure 2.28) Transients are thus unaffected, while steady states are perturbed to new levels. Steady states for increases of 25% and 50% are attained at approximately 1800 and 1600 hours, respectively, and steady states for decreases of 25% and 50% are attained at approximately 2000 and 2400 hours, respectively. Upper and lower bounds for population numbers reached in the steady state are given in Table 2.21.

Parameter n_{OP}

The resulting effect of perturbing n_{OP} was the same as that for the silicate limted case. Steady state was attained at approximately 1800 hours. (Table 2.22 and Figure 2.29)

Parameter Pr_{min}

The resulting effect of perturbing Pr_{min} is the same as that discussed in the silicate limited situations. Although this is true, the transient and steady state behavior of the systems cannot be compared since these dynamics are intimately tied with the limiting nutrient. Steady states are listed in Table 2.23 and shown in Figure 2.30.

Table 2.21: Perturbation information for parameter $\mu_{N,Pr}$ in a nitrogen limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase	1600	1.672961e+05	3.082439e+04	1.526300e+04	NA
50% increase	1800	1.672961e+05	2.574668e+04	1.255664e+04	NA
25% decrease	2000	2.338915e+05	5.022655e+04	2.508407e+04	NA
50% decrease	2400	2.338915e+05	7.531582e+04	3.768737e+04	NA

Table 2.22: Perturbation information for parameter n_{OP} in a nitrogen limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase	–	–	–	–	–
50% increase	–	–	–	–	–
25% decrease	–	–	–	–	–
50% decrease	1800	2.411943e+05	4.054809e+04	1.884368e+04	275

Table 2.23: Perturbation information for parameter Pr_{min} in a nitrogen limited system.

Perturbation	Transient		Steady State		
	Length of Transient	Max Trans Height	Max Cell Numbers	Min Cell Numbers	Max Diff
Reference	1800	2.006800e+05	4.021038e+04	1.878143e+04	–
25% increase	6200	1.676219e+05	3.104713e+04	1.514166e+04	NA
50% increase	1600	1.441004e+05	2.605419e+04	1.258109e+04	NA
25% decrease	2000	2.508643e+05	5.159364e+04	2.531453e+04	NA
50% decrease	1600	4.130872e+05	8.051439e+04	3.763601e+04	NA
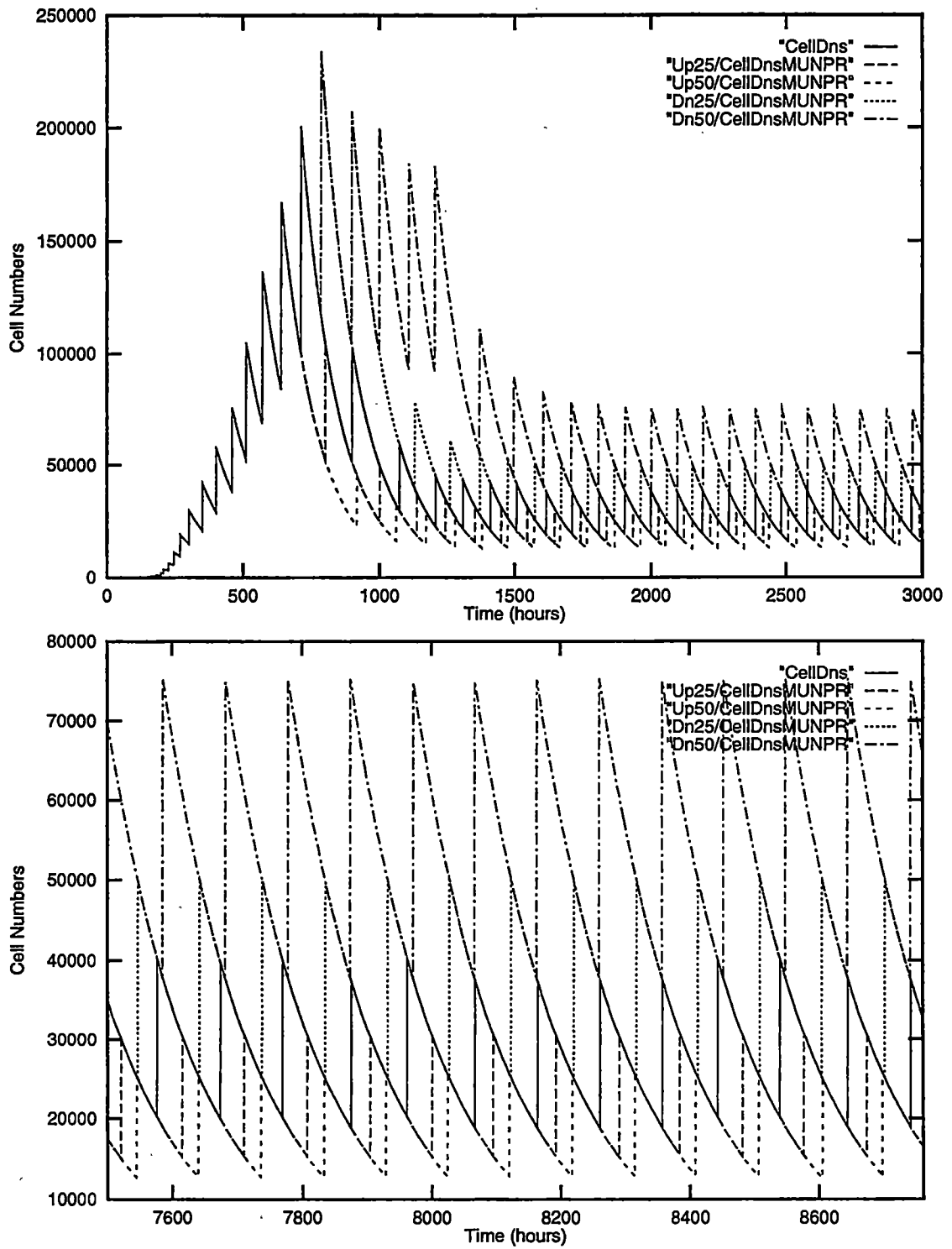
Figure 2.28: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original $\mu_{N,Pr}$ value in a nitrogen limited system.
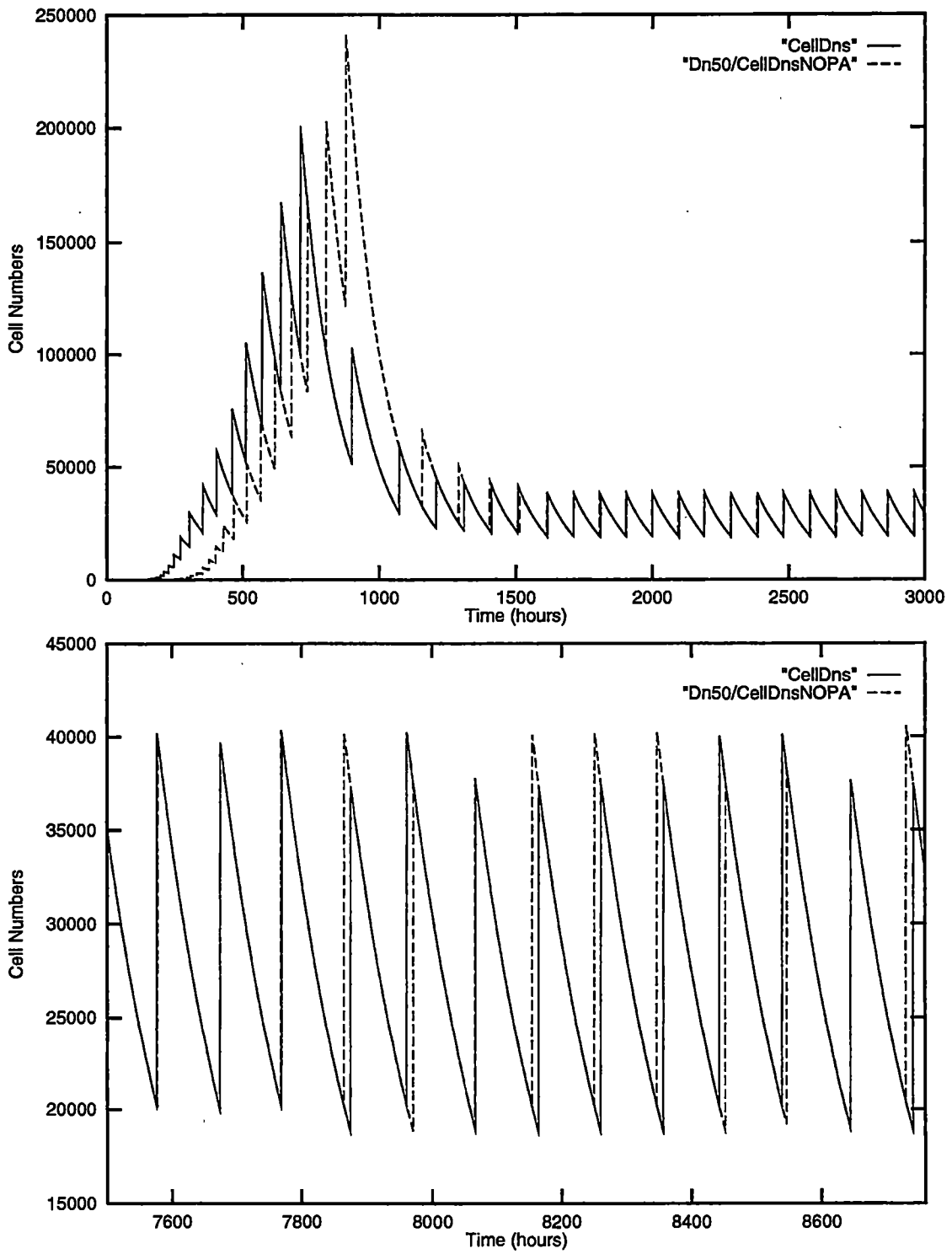
Figure 2.29: Transient and steady state behavior of 50% increase and 50% decrease in original n_{OP} value in a nitrogen limited system.
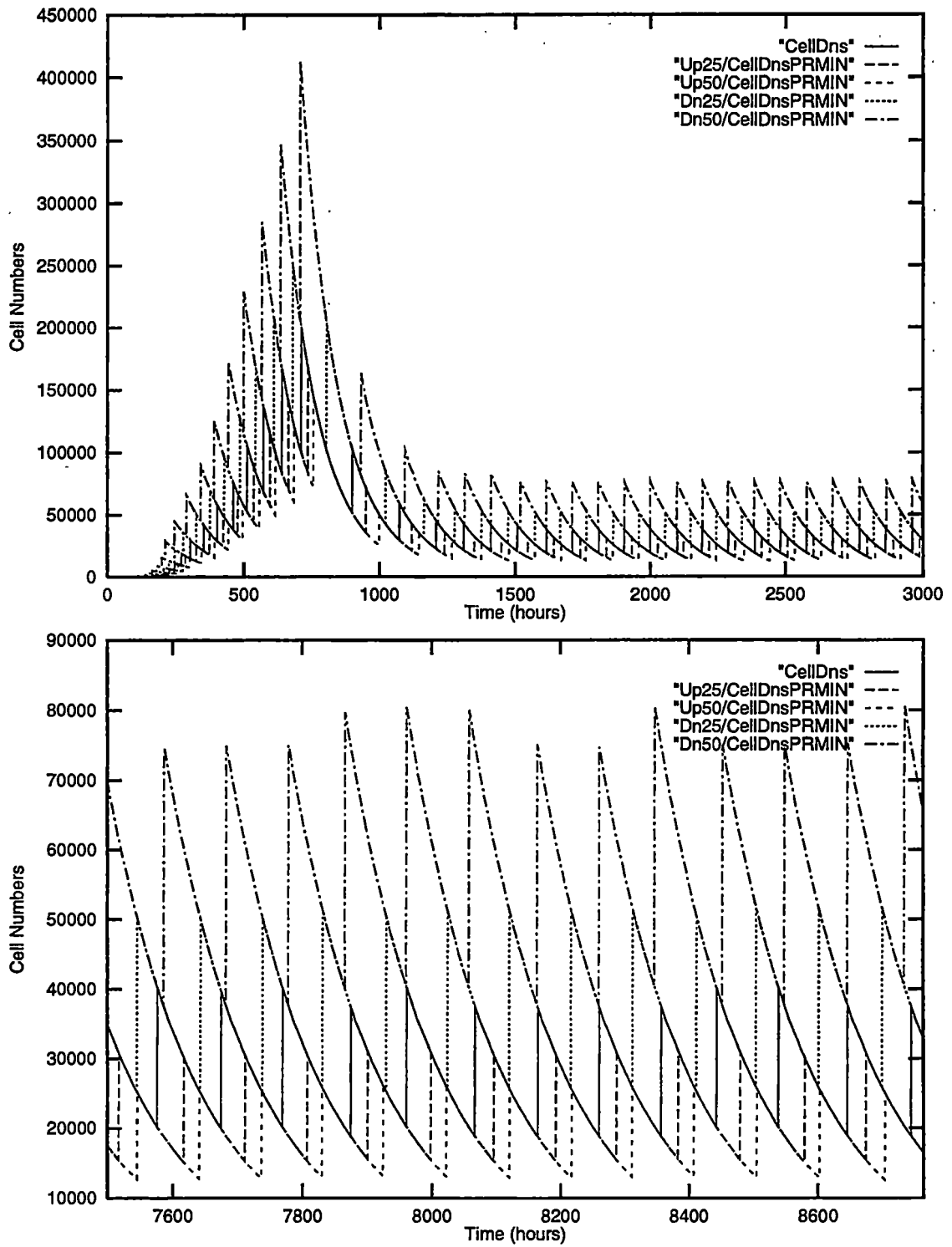
Figure 2.30: Transient and steady state behavior of 25% and 50% increase and 25% and 50% decrease in original Pr_{min} value in a nitrogen limited system.

2.4 Discussion

It is clear from the foregoing discussions that the behavior of the system is rather complex and depends on the growth limiting nutrient. It is also clear from the foregoing discussions that there are three groups of parameters exhibiting sensitivity. These include those that are sensitive during the transient state, those that are sensitive during the steady state, and those that exhibit sensitivity during both the transient state and the steady state. Of course, classification of these parameters depends on the system under investigation since the conditions from system to system change.

Parameters exhibiting sensitivity during the transient state include c'_C for all three scenarios, $\mu_{C,Pr}$ for the silicate and nitrogen limited cases, $\mu_{P,Pr}$ for the silicate limted case, and n_{OP} for the nitrogen limited case. Parameters exhibiting sensitivity during the steady state include $\mu_{Si,Pr}$ for the silicate limited case, c_{OP}, $\mu_{C,Pr}$, $\mu_{P,Pr}$ and n_{OP} for the phosphorous limited case, and $\mu_{N,Pr}$ for the nitrogen limted case. And, parameters exhibiting sensitivity during both the transient state and steady state M_O and Pr_{min}. Table 2.24 shows these classifications.

Fifty–eight parameters were perturbed for this analysis. For the silicate and phosphorous limited systems, 12.07% of the perturbed parameters showed sensitivity in the transient state, the steady state or both, and in the nitrogen limited system, 10.34% of the perturbed parameters showed sensitivity.

Table 2.24: Parameter classifications. TS = exhibits sensitivity during transient states, SS = exhibits sensitivity during steady states, BTH = exhibits sensitivity during both the transient and steady states, – = parameter does not exhibit changes in this system.

Parameter	Silicate Limited System	Phosphorous Limited System	Nitrogen Limited System
c'_C	TS	TS	TS
c_{OP}	–	SS	–
M_O	BTH	BTH	BTH
$\mu_{C,Pr}$	TS	SS	TS
$\mu_{N,Pr}$	–	–	SS
$\mu_{P,Pr}$	TS	SS	–
$user3(\mu_{S_i,Pr})$	SS	–	–
n_{OP}	TS	SS	TS
Pr_{min}	BTH	BTH	BTH

In all three cases, parameters consistently showing sensitivity during the steady states were those associated with the construction of protein and the most limiting nutrient for that system, thus indicating that the most limiting nutrient plays an important role in determining the steady state level of the population. Since these parameters affect reproduction and population growth and since population growth was used to determine parameter sensitivity, it is expected that these parameters will exhibit sensitivity in the steady state under differering initial conditions.

Parameters sensitive during the transient states, c'_C and n_{OP} (for the silicate and nitrogen limited systems), showed sensitivity because the initial conditions are such that carbon and phosphorous limit population growth at various points during the transient state. Thus, it is not expected that these parameters will

show sensitivity given another set of initial conditions.

Finally, parameters exhibiting sensitivity in both the transient and steady states include the maintenance parameter, M_O, and the reproduction parameter, Pr_{min}. M_O is categorized here because it exhibits changes in both the transient and steady state behaviors for some or all of its peturbations. Pr_{min} is classified here because it exhibits changes in both the transient and steady state behavior for all of its perturbations in all systems. It is expected that since both these parameters affect reproduction and population growth, that they will be sensitive given any set of initial conditions and under any set of limiting conditions.

Chapter 3

Temperature Effects on the Marine Diatom *Skeletonema costatum*

A second objective of this thesis is to examine the effects of various temperature scenarios on the reference algae model to determine the validity of the model's assumptions concerning temperature effects on *Skeletonema costatum.*

3.1   Temperature and *Skeletonema costatum*

It is well known that temperature is a determining factor in the distribution of species. Some species survive in a narrow temperature range, unable to withstand

long periods of exposure beyond this range, while other species survive in a wide range of temperatures being able to tolerate, and in some cases adapt to, temperatures beyond their usual interval. *Skeletonema costatum* is of this second type. It is a cosmopolitan species, enjoying distributions in both tropical and temperate waters. Duration of appearance and level of abundance differ from location to location, however. For example, *Skeletonema costatum* will occur year–round in the eastern Gulf of Mexico [2], while peaking in appearance and abundance in August in Northern Atlantic waters [24]. Thus, it is expected that if the model accurately reflects the role of temperature on the physiological and chemical processes of the cell, levels of abundance and duration of appearance will also be accurately reflected under these temperature regimes when they are incorporated into the model.

Temperature information was obtained from the National Data Buoy Center's web site [1] for three locations. The National Data Buoy Center maintains moored buoys (stations) in several locations in U.S. coastal waters. Various types of information are collected and compiled from these locations. Among data collected and compiled are average water temperature readings over several years. The first station, 42007, is located in the Gulf of Mexico near Biloxi, Mississippi. Temperature data was collected from a depth of 0.6 meters from 1/81 through 12/93. The second station, 41009, is located off the coast of Florida near Cape Canaveral in a tropical location. Temperature data was collected from a depth of

1.0 meters from 8/88 through 12/93. This station was choosen for comparative purposes since water temperatures in this area tend to fluctuate less than in the first and third cases. The third station, 44013, is located off the coast of Massachusetts near Boston in a temperate locale. Temperature data was collected from a depth of 0.6 meters from 8/84 through 12/93.

3.2 Temperature Plots and Curves

Data obtained from the National Data Buoy Center is displayed in Appendix B for the three choosen sites. Sine curves were developed for all three data sets and are shown in Figure 3.1.
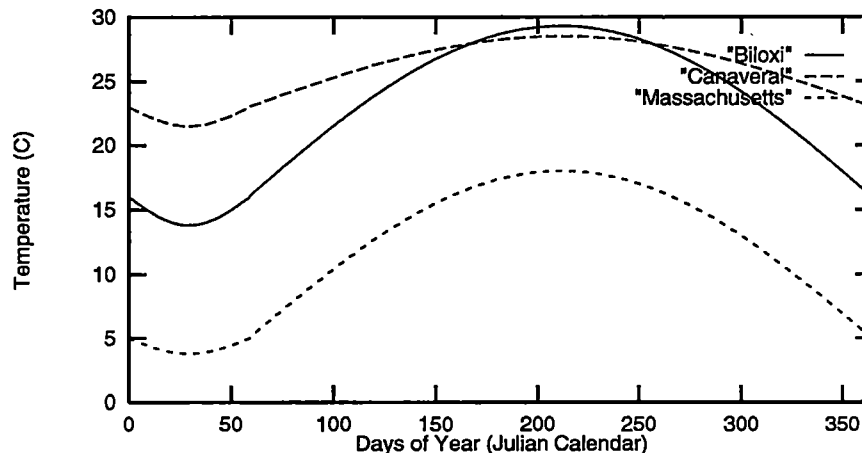
Figure 3.1: Derived data curves for stations located near Biloxi, Mississippi, Cape Canaveral, Florida, and Boston, Massachusetts

3.2.1 Temperature Curve for the Gulf of Mexico

To create a curve that captured temperature fluctuations for the entire year, the data was divided into two subsections, one starting at the beginning of January and going through the end of February and the second starting at the beginning of March and going through the end of December. December and March were choosen as anchor months because they share similar temperature averages, $16.0°C$ and $16.3°C$, respectively. It was thus assumed that the beginning of January, the beginning of March, and the end of December all share the same temperature, $16.0°C$, hence enabling the establishment of periodicity for these intervals.

The final equations are as follows:

$$T_C = \begin{cases} -2.2 sin\left(\frac{2\pi}{118.0}d\right) + 16.0 & d <= 59.0 \\ 13.3 sin\left(\frac{2\pi}{612.0}d - 0.6\right) + 16.0 & 59.0 < d <= 365.0 \end{cases} \qquad (3.1)$$

where $T_C(°C)$ is the temperature in degrees Celsius, and $d(day)$ is the day of the year in Julian calendar days.

3.2.2 Temperature Curves for Boston, Massachusetts and Cape Canaveral, Florida

Aside from slight adjustments in the numbers, curves created for the remaining two locations followed the same line of reasoning as that for the development of

the Gulf of Mexico. The differences being that 5.0°C was used as the reference temperature for the Massachusetts curve, and 23.3°C was used for the Florida curve. The equations are shown below:

Cape Canaveral, Florida:

$$T_C = \begin{cases} -1.5sin\left(\frac{2\pi}{118.0}d\right) + 23.3 & d <= 59.0 \\ 5.5sin\left(\frac{2\pi}{612.0}d - 0.6\right) + 23.3 & 59.0 < d <= 365.0 \end{cases} \qquad (3.2)$$

Boston, Massachusetts:

$$T_C = \begin{cases} -1.2sin\left(\frac{2\pi}{118.0}d\right) + 5.0 & d <= 59.0 \\ 13.0sin\left(\frac{2\pi}{612.0}d - 0.6\right) + 5.0 & 59.0 < d <= 365.0 \end{cases} \qquad (3.3)$$

3.3 Model Incorporation

The original temperature equation 1.5 was modified to accomodate the preceeding scenarios in the following manner. $s(T_1^K)$ remained the same. The Arrhenius temperature, T_A^K, was modified to reflect the value associated with algae, 6842.0 [16]. T_1^K, was calculated using the reference temperature associated with the area being simulated adjusted to Kelvins. And, T_K was modified to reflect T_C for the respective area adjusted to Kelvins. The modified uptake fluxes due to temperature are shown in Figure 3.2.
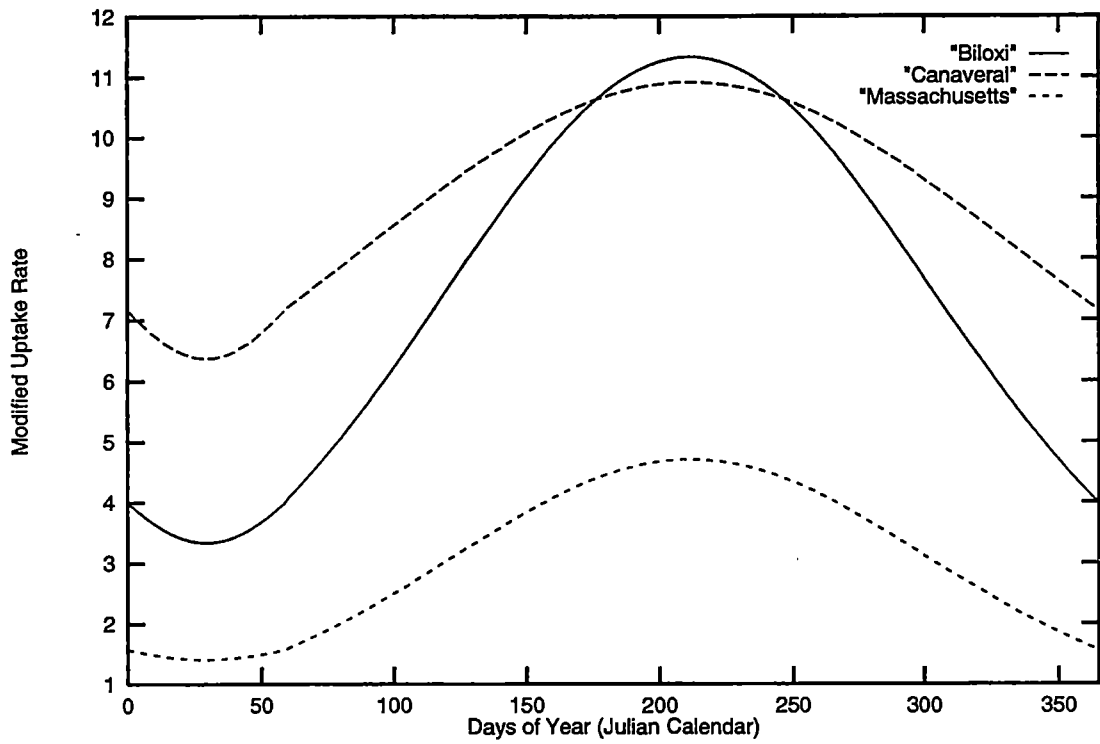
Figure 3.2: Modified uptake rates resulting from incorporation of temperatures from the Gulf of Mexico (Biloxi), Cape Canaveral, Florida (Canaveral), and Boston, Massachusetts (Massachusetts)

93

3.4 Results and Analysis

The above three scenarios were simulated with the three nutrient limiting situations discussed in Chapter 2. Between the three temperture scenarios, there was no difference in the resulting effects on any of the systems. (For example, the effect of the "Biloxi" temperature regime was exactly the same as that produced by the "Canaveral" and "Massachusetts" temperature regimes for all three nutrient limiting situations.) This is a result of the manner in which the uptake term was designed. Recall that the form of the uptake term is as follows:

$$\rho_{up,\phi_e} = \begin{cases} \left(\frac{1}{\frac{a}{\nu(T)} + \frac{c_{\phi_e}}{\phi_e}} \right) \left(1 - \frac{\phi_i}{c_{\phi_i} m_{Pr}} \right) n_{\phi_e} SA & \phi_e > 0 \\ 0 & \phi_e = 0 \end{cases} \tag{3.4}$$

Observe that the following parameters a, c_{ϕ_e}, c_{ϕ_i} and n_{ϕ_e} are all constant and have the values: $a = 0.5$ for all three systems; c_{ϕ_e} is 27 for silicate, 30 for phoshate, and 18.75 for nitrate; c_{ϕ_i} is 5.0e-3 for silicate, 8.0e-3 for phosphorous, and 5.5e-3 for nitrogen; and n_{ϕ_e} is 1.50e-7 for silicate, 7.93e-6 for phosphate, and 2.0e-6 for nitrate. Additionally, the surface area of the cell is always greater than 5.0e-3. For all three systems, the term $\left(1 - \frac{\phi_i}{c_{\phi_i} m_{Pr}} \right)$ is near one when internal nutrients are limiting, and is around 0.90 when nutrients are sufficient. When external nutrients are abundant, the term $\frac{a}{\nu(T)}$ will influence the term $\frac{c_{\phi_e}}{\phi_e}$ since ϕ_e is large. When external nutrients become limiting ϕ_e is decreased significantly (being 0.35 or less

for all three systems) and the term $\frac{c\phi_e}{\phi_e}$ becomes large. Simultaneously, the term $\nu(T)$ is increasing so that the term $\frac{a}{\nu(T)}$ becomes small and is thus insignificant in comparison to $\frac{c\phi_e}{\phi_e}$. Since internal nutrient levels parallel external nutrient levels, decreases in external nutrients means decreases in internal nutrients. Thus, when external nutrients are limiting, external nutrients dominate uptake, hence internal nutrient levels.

This can be seen in the transient and steady state behavior of the systems. The results are shown in Figures 3.3, 3.4, and 3.5, for the silicate limited system, the phosphorous limited system, and the nitrogen limited system, respectively. "CellDns" in all three cases is the reference system, "Biloxi" is the effects on the population with the Gulf of Mexico temperature equations incorporated, "Canaveral" is the effects on the population with the Cape Canaveral temperature equations incorporated, and "Massachusetts" is the effects on the population with the Massachusetts temperature equations incorporated.

3.5 Conclusions

From the preceeding discussion it is clear that the model responds to fluctuating temperatures when nutrients are sufficient. When nutrients are limiting, however, no response is observed even though temperature fluctuations are highest during these times. In all three cases, population numbers are set by external factors – the
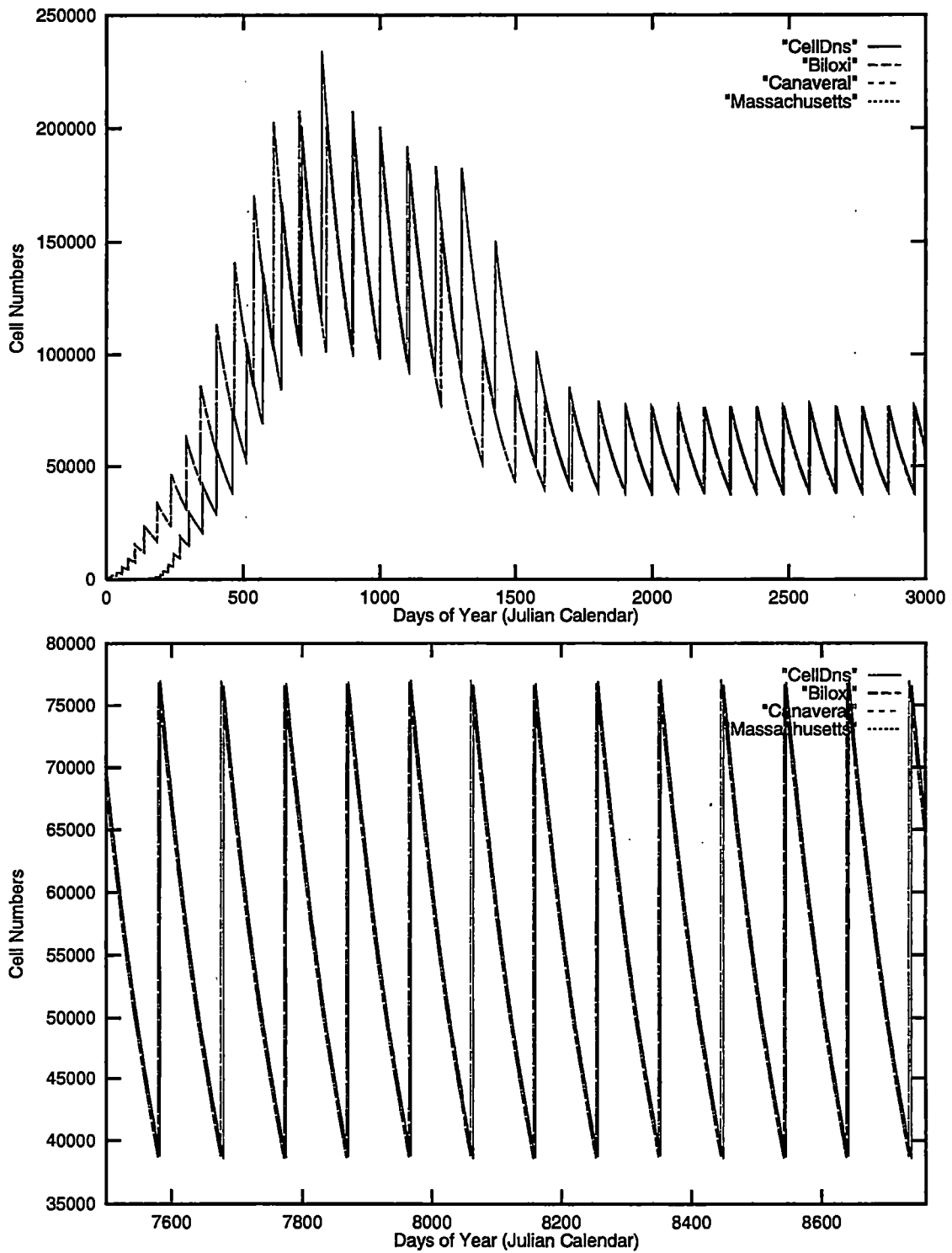
Figure 3.3: Silicate limited system (CellDns) simulated with temperature curves for Biloxi, Mississippi (Biloxi), Cape Canaveral, Florida (Canaveral), and Boston, Massachusetts (Massachusetts)
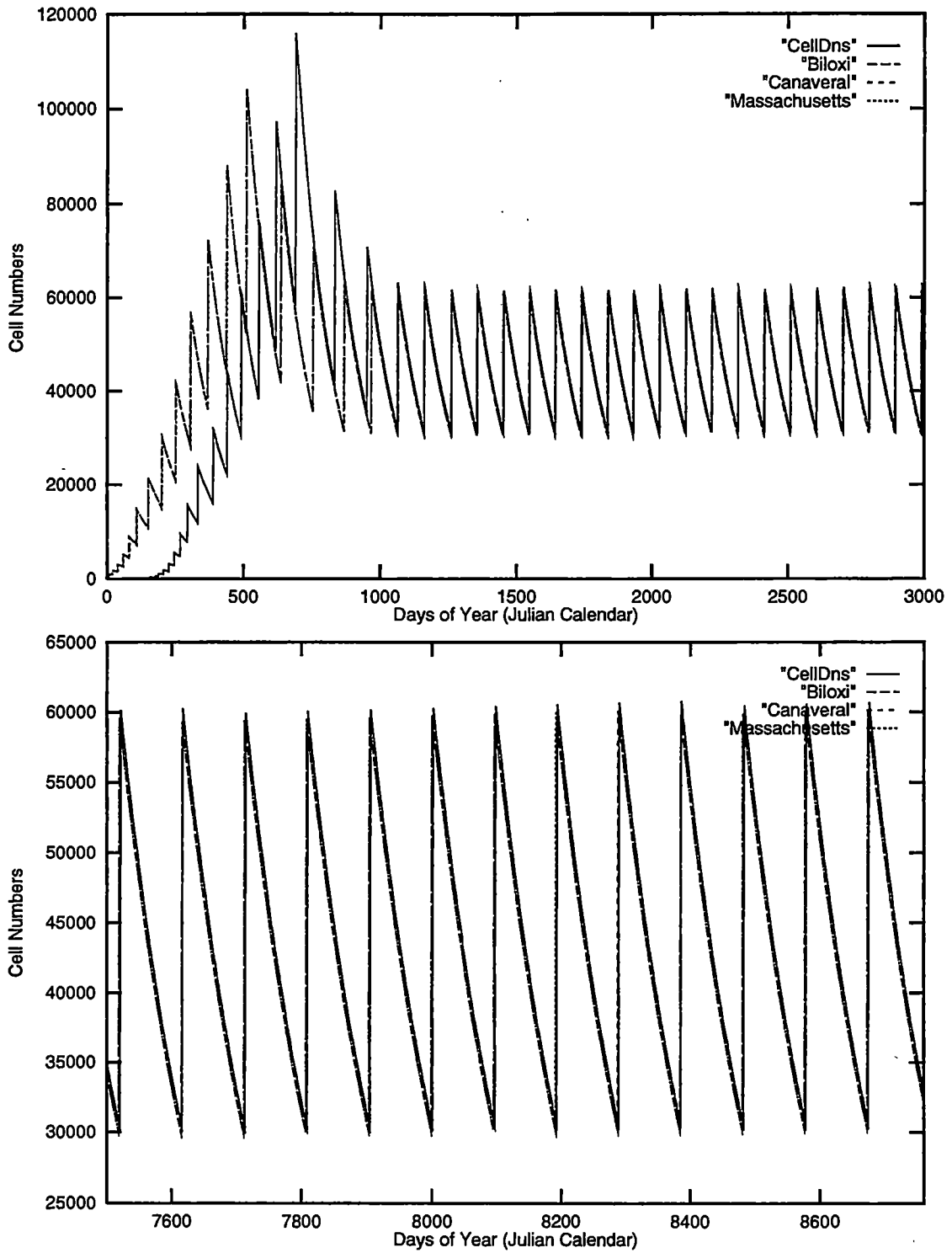
Figure 3.4: Phosphorous limited system (CellDns) simulated with temperature curves for Biloxi, Mississippi (Biloxi), Cape Canaveral, Florida (Canaveral), and Boston, Massachusetts (Massachusetts)

97
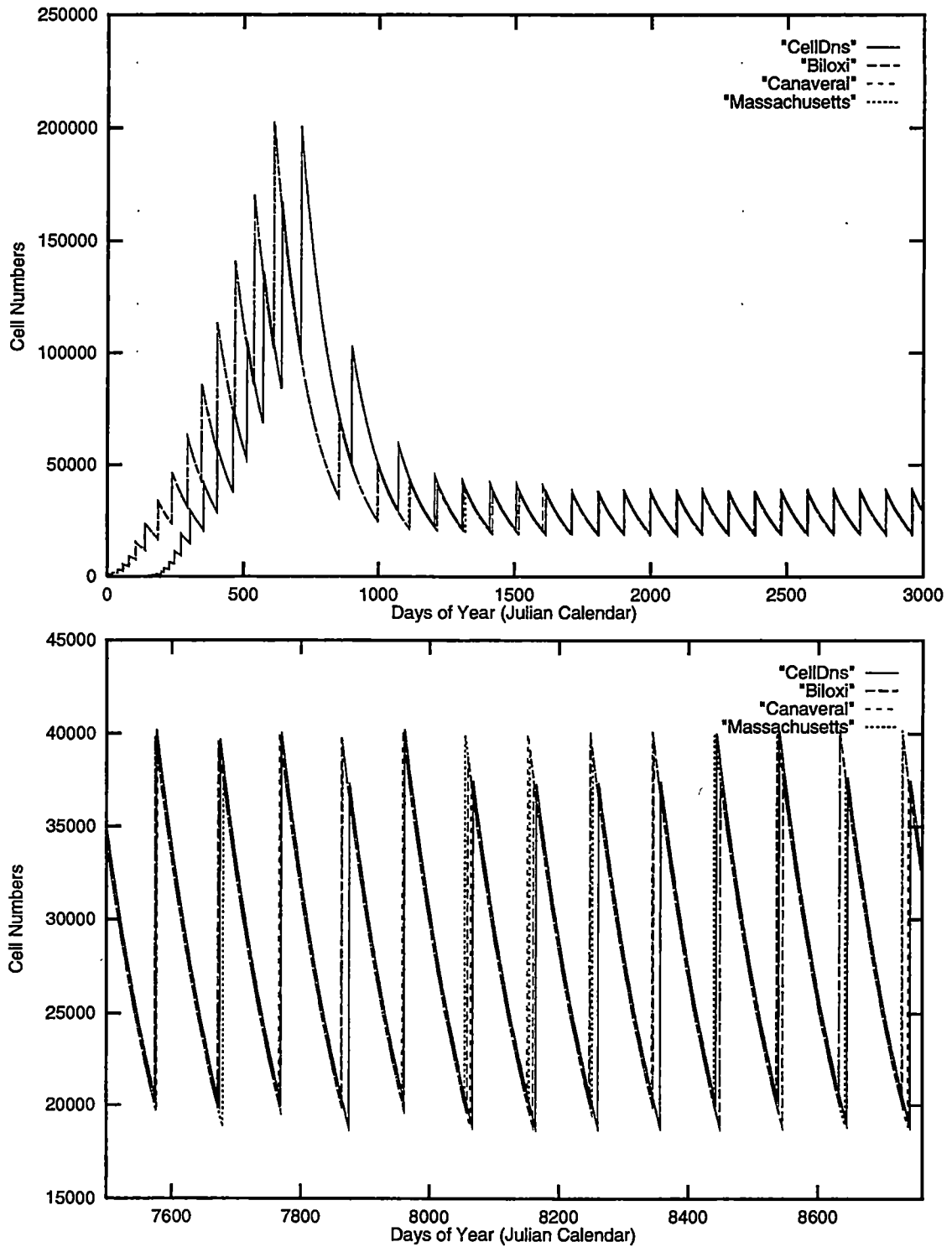
Figure 3.5: Nitrogen limited system (CellDns) simulated with temperature curves for Biloxi, Mississippi (Biloxi), Cape Canaveral, Florida (Canaveral), and Boston, Massachusetts (Massachusetts)

98

carrying capacity of the most limiting nutrient in the environment. For organisms that are able to regulate their own temperatures, these results are expected. But, for organisms that are unable to regulate their own temperatures, temperature plays a significant role in metabolism, growth, reproduction, photosynthesis and respiration [2]. In the model's current formulation, these are partially accounted for.

Via underlying assumptions of the model, temperature's only direct effect is the rate of nutrient availability on the interior of the cell. This in turn affects anabolic processes hence growth and reproduction. According to Dawes [2], the rate of nutrient availability is only one aspect of the metabolic process and temperature ranges in which these reactions take place must also be considered. Jorgensen [14] has observed that growth rates of *Skeletonema costatum* decrease with decreasing temperatures, while organic matter, protein and the height of cells increases. This is apparently attributable to increased protein manufacture and carbon incorporation in cooler temperatures. Mortain– Bertrand et al [19] have confirmed this and in addition have found that the photosynthesis vs. irradiance curve is controlled by light history, while L_{max} is controlled by temperature. It would thus be expected that populations experiencing the Massachusetts temperature regime would have decreased population numbers and increased cell sizes, while populations experiencing the Cape Canaveral or Gulf of Mexico temperature regimes would have increased population numbers and smaller cell sizes.

99

It is my opinion that the current formulation of the model is exceptional in its ability to express the physiological dynamics of the population due to nutrient availability, but that further refinement of the model is necessary. In it present formulation, temperature affects the rate of availability of nutrients for metabolism, but not the actual rate of the process. It is known that temperature affects both these processes Dawes [2], as well as photosynthesis and other proceses. A natural next step in accounting for temperature affects would be to incorporate temperature into these processes.

Chapter 4

Algae Code Modifications

This chapter describes the design and structure of the original algae code, and discusses subsequent modifications. Modifications were prompted by the avenue choosen for parallelization efforts, discussed in Section 5.3, and were done to increase usability so that future users can concentrate more on using the code and obtaining results and less on changing the code so that it runs properly with new data sets.

4.1   Description of the Original Algae Code

The original algae code consisted of the files *main.c*, *fun1r.c*, *fun2r.c*, *fun3r.c*, *globalr.h*, and *global2r.h*, and required four types of input files. The input files consisted of a file containing the initial system variables, *globr.in*, a file containing global parameter values, *globr.par*, files containing species specific intialization

values, *spec1r.in*, *spec2r.in*, and *spec3r.in*, and files containing species specific parameter values, *spec1r.par*, *spec2.par* and *spec3r.par*.

4.1.1 Data Input

Data and parameter initialization was performed by the *initial()* and *pararead()* functions called by *main()* and contained within *main.c*. The input filenames were stored in various arrays of strings and looped through as data was read from each. All data read into the code was stored in a large structure defined in *globalr.h* with type name *ecovars*. This structure contained 94 parameter arrays of type float, 11 global variables of type float, two two–dimensional arrays of type double containing either initial value data (called x) or intermediate value data (called z), one array of type double containing system variables (called y), and one integer used as a counter. Each parameter array and both the initial value data array, x, and the intermediate value data array, z, were of length ECO where ECO was the total number of ecotypes for all species and was defined in *globalr.h*. The structure *ecovars* is shown in Figure 4.1.

The two reading routines *initial()* and *pararead()* were set up to read initial values and paremeter values, respectively. Thus, *initial()* read the files *spec1r.in*, *spec2r.in*, *spec3r.in* and *globarlr.in*. And, *pararead()* read in the files *spec1r.par*, *spec2r.par*, *spec3r.par*, and *globr.par*.

```
struct ecovars {
 double x[11][ECO];  /* x values for each ecotype */
 double y[18];  /* y values not dependent on ecotype */
 double z[16][ECO];  /* z values for each ecotype */
 int spcount;  /* counter used to loop over species */
 /* Active Transport Parameters */
 float  a[ECO], c_FE[ECO], c_AM[ECO], c_NI[ECO], c_OP[ECO], c_OR[ECO],
   n_FE[ECO], n_AM[ECO], n_NI[ECO], n_OP[ECO], n_OR[ECO];
 /* Anabolism Parameters */
 float  m0Ps[ECO], m0Pr[ECO], m0Lp[ECO];
 /* Functional Response Parameters */
 float  tPsb[ECO], tLpb[ECO], tPrb[ECO], c_C[ECO], c_N[ECO], c_P[ECO],
   c_E[ECO], k_CPs[ECO], k_NPs[ECO], k_CLp[ECO], k_PLp[ECO],
   k_CPr[ECO], k_NPr[ECO], k_PPr[ECO], k_EPs[ECO], k_ELp[ECO],
   k_EPr[ECO], L_mPs[ECO], L_mPr[ECO], L_mLp[ECO];
 /* Catabolism Parameters */
 float  ka_Ps0[ECO], ka_Ps1[ECO], ka_Ps2[ECO], ka_Ps3[ECO], ka_Pr0[ECO],
   ka_Pr1[ECO], ka_Pr2[ECO], ka_Pr3[ECO], ka_Lp0[ECO], ka_Lp1[ECO],
   ka_Lp2[ECO], ka_Lp3[ECO];
 /* Photosynthesis, Maintenance, and Temperature Dependent Parameters */
 float  E_k[ECO], L_max[ECO], L_alp[ECO], L_beta[ECO], L_P[ECO],
   mu_COATP[ECO], MO[ECO], sT1K[ECO], T1K[ECO], TAK[ECO];
 /* Flux Matrices Parameters */
 float  mu_CCO[ECO], mu_CFE[ECO], mu_COR[ECO], mu_NAM[ECO],
   mu_NNI[ECO], mu_POP[ECO], mu_POR[ECO], mu_ATPCO[ECO],
   mu_ATPFE[ECO], mu_ATPNI[ECO], mu_ATPOP[ECO], mu_CPs[ECO],
   mu_CLp[ECO], mu_CPr[ECO], mu_NPs[ECO], mu_NPr[ECO], mu_PLp[ECO],
   mu_PPr[ECO], mu_ATPPs[ECO], mu_ATPLp[ECO], mu_ATPPr[ECO];
 /* Surface & Volume, Reprodction and Excretion Parameters */
 float  sigma_Ps[ECO], sigma_Pr[ECO], sigma_Lp[ECO], Prmin[ECO],
   zeta_GL[ECO], zeta_AM[ECO], zeta_OR[ECO];
 /* User Defined Parameters */
 float  user1[ECO], user2[ECO], user3[ECO], user4[ECO], user5[ECO],
   user6[ECO], user7[ECO], user8[ECO], user9[ECO], user10[ECO];
 /* Sinking, Grazing and Photosynthesis Parameters */
 float  sz, qu, r, MI, AE, varphi, teta, ED0, D, K_d, K_p;
}
```

Figure 4.1: *ecovars* data structure from original algae code.

Files prefixed with *spec* contained species' specific information. In these files, data for the ecotypes were associated by parameter so that for the same parameter, values for each ecotype appeared on the same line separated by a space. For example, the listing for the cell wall thickness would look something like: 0.5 0.5 for two ecotypes of the same species.

initial() read species specific and global initial values. Species specific information was read utilizing one outer "for" loop and three inner "for" loops. The outer loop looped over the total number of species being modelled, opening the associated species files in turn. The first inner loop read the first four lines in the file. The second inner loop read the next line in the file, looping over the number of ecotypes for the current species. The third inner loop performed two functions. It first read the next line in the data file which had to be a comment. It then read the following line by looping over the total number of ecotypes for the current species; this line had to be data. Thus, the reading routine was designed to read a file that alternated between a line of information and a line of data; the first four lines being comments. Global initial values were read utilizing one loop which looped over the total number of initial values contained in the file. It then read additional miscellaneous data with appropriately placed C *fgets()* and *fscanf()* functions.

The *pararead()* routine read in species specific and global parameter values and followed similar convention as those used above.

4.1.2 Calculations and Output

Code calculations were directed by a simple loop contained within *main()*. This loop's execution was controlled by the boolean result of the comparison between the two variables *ttime* and *maxtime*. *ttime* served as a cumulative counter for passed time steps and was initialized to zero prior to the loop's execution. *maxtime* served as the terminal value and was initialized by the product of the maximum simulation time in hours (defined in the *globr.in* file) multiplied by 3600.0, the number of seconds per hour. For each iteration of the loop the function *runge()* was called, *ttime* was incremented by the time step (0.5), and the print control counter, *co*, was incremented by one. Data values were printed to the screen every simulated second (3600.0/time step).

Actual model calculations were performed in the functions *runge()*, *fun1r()*, *fun2r()* and *fun3r()*, *runge()* being in *runge.c* and *fun*r()* being in *fun*r.c* (* = 1, 2, 3). The function *runge()* utilized a second order Runge Kutta numerical scheme, calculated the cell density and irradiance at the current time step, and called the functions *fun1r()*, *fun2r()* and *fun3r()* in turn. The functions *fun1r()*, *fun2r()* and *fun3r()* were associated specifically with the species information contained in the files prefixed with *spec*. The *fun* functions all calculated temperature, active transport, volume and surface values, functional response, catabolism, photosynthetic rate, and internal pool levels. *fun1r()* and *fun3r()* were identical and

were designed to model *Skeletonema costatum* populations, hence, contain equations modified to incorporate silicate. *fun2r()*, on the other hand, was designed to model theoretical algal populations thus did not incorporate silicate. The general algorithm is shown in Figure 4.2.

4.2 Changes to the Algae Code

Analysis of potential avenues for parallelization indicated data parallelization as having the highest probability for success. (Refer to Chapter 5 for details.) Data parallelization entails each processor working on a subset of the entire data set. Recall that the original code was structured so that data was read into the large *ecovars* structure; the main reading routines were such that adding or deleting lines in the files would cause the routines to fail; the main calculation routines were separated according to the species they operated on; and output was printed to the screen. To accomodate data parallelization, the *ecovars* structure was disassembled and replaced with several independent arrays, the reading routines were redesigned so that comments, species, and ecotypes could be added or deleted freely, the main calculation routines *fun1r()*, *fun2r()* and *fun3r()* were collapsed into one, and data output was redirected to four predesignated files. These modifications are illuminated in the following subsections.
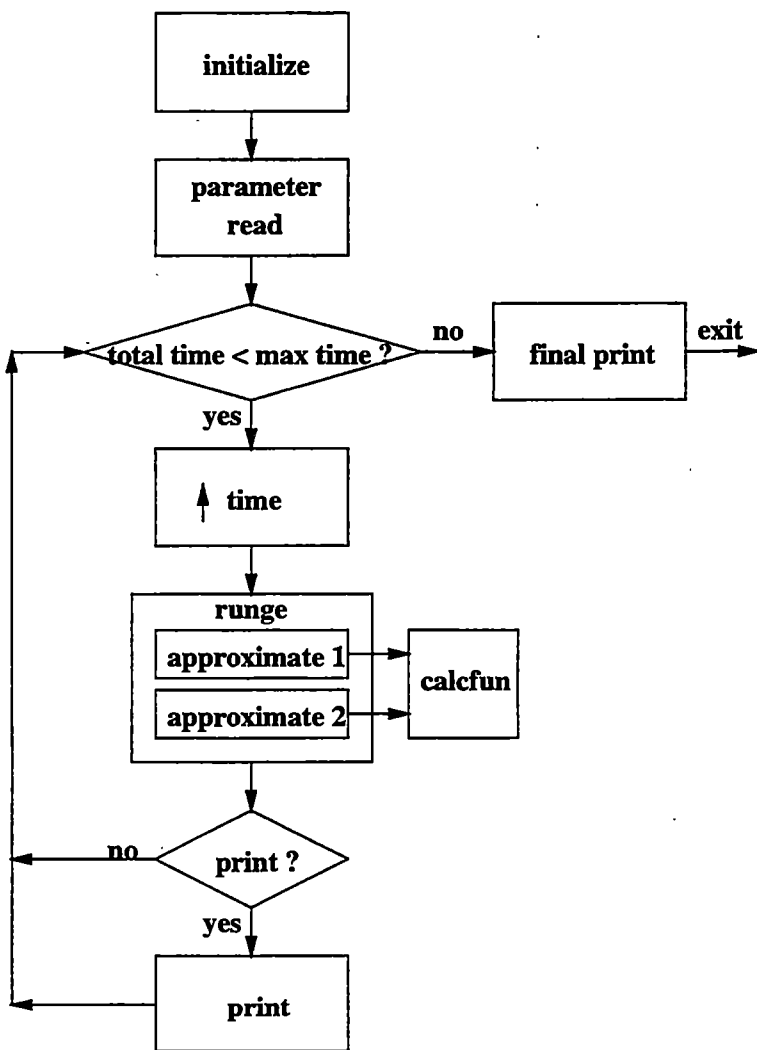
Figure 4.2: Flow diagram for the algae code.

4.2.1 Data Input

The *ecovars* Structure

As discussed, data was read into a large structure called *ecovars* (Figure 4.1). Parameters and variables in this structure were associated by type and were contained in arrays of length ECO for parameters, and longer for variables. For parameters alone, there were 94 of these arrays. In light of parallelization, a structure of this type is unnecessary. One benefit of parallelization is the ability to distribute data on an as needed basis. With the data being arranged in such a manner, dispersal of the entire structure would have been required not only during initialization, but also at each time step when system information had to be shared. Although, this places no additional demands on communication resources, the task of packing and unpacking this structure into or out of a buffer would have been onerous, clumsy and redundant.

To remedy this situation the *ecovars* structure was dismantled and its individual components renamed and reassociated. The initial value data array, x, was renamed *ecoinit*, the system variables array, y, was renamed *sysvars*, and the intermediate value data array was renamed *ecovals*. The original arrays x and y had row dimension corresponding to the number of respective parameters and column dimension corresponding to the number of ecotypes; this was reversed. *spcount* was no longer needed in the revised version of the code so was eliminated. The

94 parameter arrays of length ECO were converted into a single two–dimensional array with row dimension equal to the total number of ecotypes and column dimension equal to 94. This array was called *ecoprms* and resulted in parameters being associated by ecotype instead of by parameter type. The remaining values in *ecovars* were gathered into an array called *glbprms*.

In the new version of the code, all arrays were dynamically allocated at run time. For the one–dimensional arrays, *sysvars* and *glbprms*, allocation was determined by the number of respective values each had to hold. For the two–dimensional arrays, *ecoinit*, *ecovals* and *ecoprms*, however, allocation was quite a bit different. First, an array of pointers was allocated corresponding to the total number of ecotypes. Indices into this array corresponded directly with the ecotypes as they were read in. For example, if species one had two ecotypes and species two had two ecotypes and they were read into the code in that order, then the first two indices of the pointers array would correspond to the two ecotypes of the first species, and last two indices of the pointers array would correspond to the two ecotypes of the second species. Next, each pointer in the array of pointers was assigned to point to the beginning of a one–dimensional array dynamically allocated for the required length. This approach is used quite often and a similar one can be found in [21].

Reading Routines

Recall that the data reading routines *initread()* and *pararead()* were originally designed in such a manner that adding or deleting lines in the input files, be it data or a comment, would cause the code to fail unless the code itself was modified to directly accomodate these changes. Two objectives of this thesis, besides the aforementioned one surrounding parallelization, were to perform a sensitivity analysis on the system and examine the effects of temperature on the algae. Accomodating these objectives demanded flexibility in being able to add, delete and change information in the data files without having to modify the code. In addition, future directions mandate system enlargement via the additon of ecotypes and species, and the coupling of the model. At these points, interaction time with the code should be minimal as the focus will need to be directed toward obtaining and understanding simulation results.

In addressing these issues, the two reading routines *initread()* and *pararead()* were replaced with the five routines *popspecinfo()*, *initspeciesparams()*, *initglobalparams()*, *speciesinit()*, and *sysvarsinit()*, all contained in the file *initalgae.c* and all of which were designed to follow four simple rules. One, all lines in the input files not containing data begin with a # sign and are skipped during file processing. Two, the end of file marker is a right brace }. Three, all species information is contained in the same file, with the end of species delimiter also being the right

brace }. Four, following the convention used in the original model, ecotypes belonging to the same species are grouped together in the file so that associated parameter values appear on the same line. These routines were modelled after those developed for the *Daphnia* and fish models. The reading routines and their associated files are as follows.

The routine *popspecinfo()* reads general population specifications from the data file PopSpecInfo. This includes the total number of species being modelled, the total number of ecotypes being modelled, and the number of ecotypes for each individual species. It is important to observe that the number of ectotypes read in from this file is used to allocate the arrays discussed earlier and should therefore accurately reflect the contents of the species specific data files. *initspeciesparams()* and *speciesinit()* read in the SpeciesFile and SpeciesIn files, respectively. Species-File contains ecotype specific parameter values, and SpeciesIn contains ecotype specific initial conditions. The routines *initglobalparams()* and *sysvars()* read in the files GlobalParams and SysVars, respectively. GlobalParams contains species independent global parameter values and SysVars contains species independent values for the system variables. The data input files are passed into the code on the command line in the following order: PopSpecInfo, SpeciesFile, Global-Params, SpeciesIn and SysVars. (Of course these can be called anything, but the order must be the same.)

4.2.2 Calculations

As described previously, the main calculation routines, *fun1r()*, *fun2r()* and *fun3r()*, were designed to be associated with specific types of species; *fun1r()* and *fun3r()* calculated results for diatom species, and *fun2r()* calculated results for non–diatom species. While this feature is necessary to obtain answers to some of the questions under consideration in this thesis, the design does not yield the flexibility needed to obtain these answers. One aspect of the analysis objective was to compare the dynamics of *Skeletonema costatum* with the dynamics of the theoretical algal species both with and without temperature affects. To do this with the original code meant changing the predefined value of SPEC, changing the string arrays containing the names of the files prefixed with *spec*, modifying the pointer array containing pointers to the respective *fun* functions, and, finally, recompiling with the appropriate *fun* function. Although individually these tasks are not difficult nor laborious, collectively there are enough of them that forgetting to perform one or more could lead to erroneous results and increased preparation time.

To reduce the number of potential sources for error, the files *fun1r.c*, *fun2r.c* and *fun3r.c* were collapsed into one file called *calcfun.c*. Calculations in the individual *fun* functions were then consolidated to eliminate species–type dependence in the functions themselves. As stated before, the primary difference between the functions *fun1r()* and *fun3r()* and the function *fun2r()* was the incorpora-

tion of silicate for diatom species. Since this feature had to be maintained, a diatom indicator flag was declared and added for each ecotype to the end of each species' initial value data set. For diatom species, the flag was set to one, and for non–diatom species, the flag was set to zero. Silicate equations were then wrapped in an "if" statement and executed only when the flag was set. To further modularize the *calcfun.c* file, each independent computation was placed in it's own function. Thus, the calculations for temperature, active transport, volume and surface values, functional response, catabolism, photosynthetic rate, and internal pool levels, became encapsulated in the functions *calctemp()*, *acttrans()*, *calcvolsur()*, *funcresp()*, *catabolism()*, *photosyn()* and *calcfs()*, respectively, with an addition of the function *sysloss()* which calculates system losses. These functions are called only by the function *calcfun()* which was established primarily for this purpose.

4.2.3 Output

Output from the original code was directed to the screen. While this is acceptable for small amounts of data generated from a sequential processor, this approach is not practical for large amounts of data or for parallel processors. Therefore, the *openit()* and *report()* functions were created to open four predesignated files in which to write results. These four files are called CellDns, StrucResv, IntNuts, and ExNuts and contain each ecotype's cell numbers, internal storage and structure

113

levels, internal nutrient levels, and external nutrient levels, respectively.

4.2.4 Miscellaneous Changes

Once the revised version of the code was completed, it was discovered that for long simulation periods some values in the code would become smaller than a float variable was able to hold, hence causing population numbers to become infinite while simultaneously resulting in large negative internal nutrient values. This behavior was observed only when simulations were run with several ecotypes and is clearly not a biologically possible situation. All floats were changed to doubles, thus correcting the problem.

It was also found that uptake, death and all other processes occurred when populations were extinct; another biologically impossibility. This was solved by checking that cells existed before values for these processes were calculated.

4.2.5 Final Version

The final version of the sequential code consists of the files: *main.c, initalgae.c, runge.c, calcfun.c, report.c, error.c, mem_alloc.c*, their associated header files, and the two additional header files *cons.h* and *globalr.h*. The input and output files are as described above. A makefile, called *MakeSerAlg*, which allows easy compilation of the entire program, has also been created for use with these files.

Chapter 5

Parallelizing the Algae Model

The final objective of this thesis was to explore the feasibility of parallelizing the algae model in preparation for its coupling with the *Daphnia*-fish predator–prey system. The IBM SP2 at the University of Tennesse, Knoxville was used for this effort, and for two reasons, parallelization was not as successful as was hoped. This chapter discusses various parallelization approaches and numerical schemes tried, and presents conclusions based on these findings.

5.1   Brief Note to the Reader

It is assumed that the reader is familiar with issues in parallel computing. If not, an excellent introduction can be found online at JICS [13].

5.2 Parallel Approaches for the *Daphnia* Model, the *Daphnia–fish* Predator–prey System, and the Algae Model

The primary goal of parallelization is to reduce the amount of time it takes for a single task to execute. This is accomplished by dividing the task into several smaller tasks and then distributing these tasks across individual processors. Division of the work is determined by the design and structure of the code. A single task will usually have a few potential avenues for parallelization, which may include various combinations of dispersing subsets of the data or parts of the code itself. Once these options are identified, the one that maximizes the $\frac{computation}{communication}$ ratio and yields the most efficient load balance is the one that should be selected.

For the *Daphnia* model, two parallelization approaches were considered; a coarse–grained approach in which ecotypes were distributed across the processors, and a fine–grained approach in which cohorts were distributed across the processors. For the first approach, it was argued that while dispersing ecotypes across processors had the advantage of data remaining stationary during the simulation, it had the greater disadvantage of severe load imbalances across the processors as the simulation progressed because of the survival of the fittest phenomenon. For the second approach, it was argued that cohorts could be distributed across processors without regard to their associated ecotypes. In this approach a global combine routine used to combine births across processors was employed as well as

116

a load balancing routine to avoid situations similar to the ones described for the coarse–grained approach. These arguments are presented by Ramachandramurthi et al. [22].

The fish model in the predator–prey system has not actually been parallelized in the sense that distinct subsets of data are dispersed across processors. Instead, all fish are distributed to each processor and fish activity on the individual processors is communicated. This is a direct result of underlying assumptions concerning the feeding behavior of the fish [22]. (The feeding behavior is fully described in [11], and a full description of the parallelization of the *Daphnia*–fish, predator–prey system can be found in [20].)

Considering the algae model, two potential avenues for parallelization emerged. Due to the underlying assumptions surrounding reproduction in the algae model, cohorts were non–existent so that the two parallelization options concentrated on were ways in which to distribute ecotypes. The first possibility was to utilize the original form of the algae code in which each processor would execute a particular function (*fun1()*, *fun2()* . . . *funN()*), operating on the associated population's ecotypes. This approach had the disadvantages that the number of processors was set by the number of populations and that the number of ecotypes calculated per function could vary considerably thus resulting in load imbalance across processors. The second approach was to disperse ecotypes regardless of population association. This had the advantages of being able to have a variable number

117
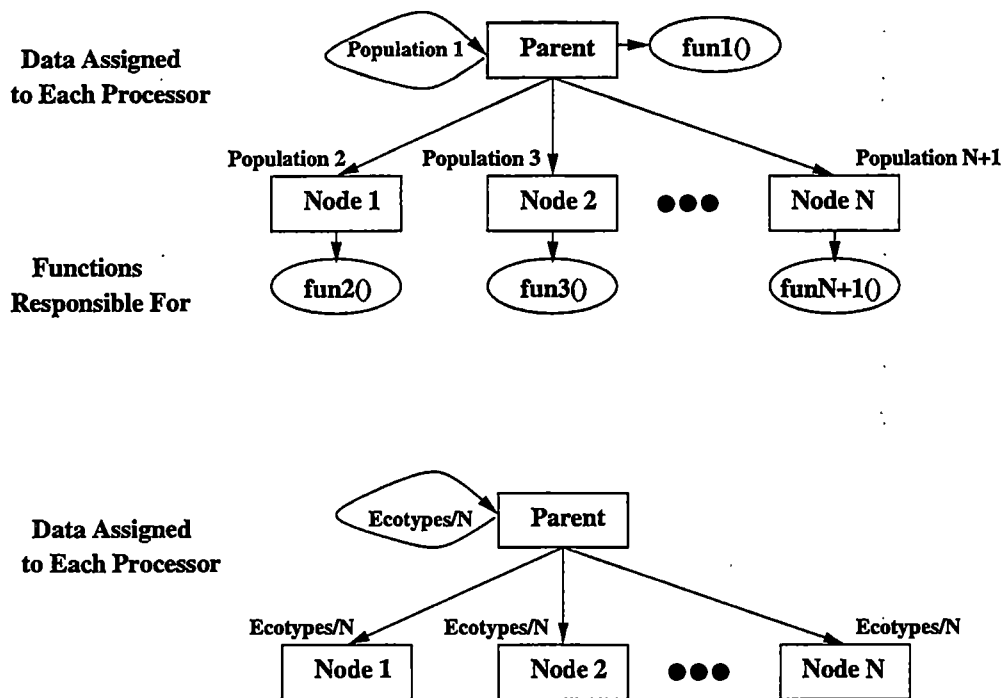
Figure 5.1: Parallel data distribution options for the algae model.

of processors and being able to ensure even dispersal of ecotypes across the processors depending on processor availability. Since this approach was choosen as the most attractive in light of coupling it with the *Daphnia*-fish predator–prey system, the original sequential code was restructured as described in Chapter 4 to accomodate this. These two approaches are shown in Figure 5.1.

5.3 Parallel Implementation

Target Architecture

The target platform for the algae model was the SP2's thin nodes. The thin nodes consist of a total of 32 nodes, 24 nodes each with 120MHz processors (Pool 0) and 8 nodes each with 160 MHz processors (Pool 2). The interconnection network consists of a TB3 switch and adapter with a peak bandwidth of 150 MBytes/second. All processors have their own memory so that collectively, the thin nodes are classified as a MIMD (mulitple instruction, multiple data) architecture. Communication in this type of architecture is generally through a message passing library of some sort. MPI (Message Passing Library) was used to parallelize the algae code.

Communication

Communication occurs at every time step, in the algae model. This is an unavoidable inconvenience in light of parallelization and is a direct result of the underlying assumptions that all individuals share all available resources and that all individuals affect photosynthetic activity for each individual. Thus, in order for individuals on each processor to know what was extracted from the environment by individuals on other processors, and for individuals to correctly calculate photosynthesis, nutrients extracted by individuals on each processors and total

119
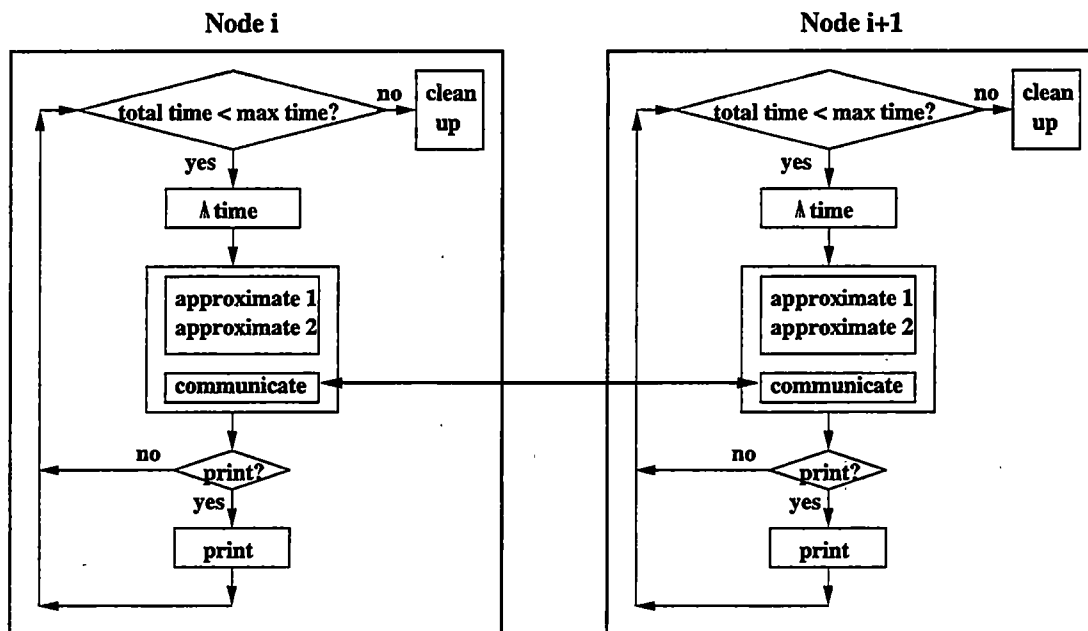
Figure 5.2: Flow representation of communication between processors.

cell numbers for each ecotype on each processor must be communicated at each time step. Then, all individuals can uptake from the same resource level and accurately determine photosynthesis at the next time step. The flow diagram for communication is shown in Figure 5.2.

Data Distribution

All data is read in and distributed by the parent node. The first file read contains the total number of ecotypes, the total number of populations and the total number of ecotypes per population (see Chapter 4, Section 4.2 for details). This information, in conjunction with the number of processors, is used by the parent to evenly divide the ecotypes across the processors. Extra ecotypes are distributed

one at a time until gone starting with the parent and successively distributing making sure that the data distributed remains contiguous. During this process an index array (*indary*) is created and stores the ending indices of the ecotypes each processor is assigned. This array is used many places throughout the code, but is particularly useful for ensuring that the processors pack and unpack data into the correct position in the array globally shared at each time step. Once the number of ecotypes per processor and the *indary* are determined, the parent broadcasts this information to each node. The nodes in turn use this information to determine how much data they will be receiving and allocate memory accordingly and where appropriate.

All initialization information is either broadcast or sent by the parent and either broadcast or received by the nodes. The sending functions invoked by the parent reside in the previously discussed *initalgae.c* file. The matching receiving calls posted by the nodes are contained within the *initalgaenode.c* file. Note that all parallel invocation in the code is performed only when the PARALLELALGE flag is set in the *cons.h* file.

Times for this process to complete for various numbers of ecotypes distributed on 2 and 4 nodes is shown in Table 5.1.

Table 5.1: Initialization times in seconds for a variable number of ecotypes run on 2 and 4 nodes of the SP2. (Time simulated is 5 days, but this does not affect the initialization process.)

| Number of Ecotypes | Number of Nodes | | | | | |
| | 1 Node | | 2 Nodes | | 4 Nodes | |
	Initialize Time	% Total Time	Initialize Time	% Total Time	Initialize Time	% Total Time
32	0.51	0.55%	0.58	0.4%	0.37	0.2%
128	0.78	0.2%	0.64	0.2%	0.61	0.2%
512	1.40	0.03%	1.7	0.07%	2.0	0.1%

Parallel I/O

On the SP2, writing to a file in a home area while a code is executing requires information to be sent across the interconnection network for each write process. For one or two messages, this is not a threat to the performance of the code, but for frequent writes, performance could be jeopardized. Writes in the algae code take place every simulated hour, or 7200 iterations for a time step of a half of a second. To avoid threatening performance, four functions were created, *createdirstrng()*, *openit()*, *merge()* and *cleanup()*. Their purpose and function are discussed below.

Seven arguments are required on the command line for the parallel code. The first five are the data files that were discussed in Section 4.2. The remaining two are the name of the directory on the node in which to write to and the full path name of the directory in which the files will ultimately end up. The most convenient choices for these two directories are the local /tmp/ directory, which avoids sending data across the intercommunication network during execution, and

the directory from which the executing code resides. The four functions and the
two directories are used in the following way.

The parent node, after reading the command line information, calls *create-dirstrng()* which creates a directory string using the sixth argument passed on
the command line (/tmp/ usually). This string and the final directory string are
broadcast to the nodes. Each node, including the parent, then calls *openit()*.
The function *openit()* enables each node to create its own specialized output file
whose name consists of the predesignated filenames discussed in Section 4.2 with
its processor rank appended as a suffix. *openit()* then uses the string created in
createdirstrng() to open files on the respective nodes in the designated directory.
The reporting function has been designed so that only the data generated by the
node is written to its local file. Upon completion of the main execution loop, the
remaining two functions, *cleanup()* and *merge()*, are invoked. Every node calls
cleanup(). *cleanup()* enables each node to copy its specialized local output file to
the final directory specified as the seventh argument on the command line. If the
node invoking *cleanup()* is the parent, then *merge()* is called. *merge()* merges all
the individually created files into one file in the final directory, and removes the
individual files.

The time interval beginning after completion of the main execution loop and
ending after completion of the merge process is considered the finalization time.
Finalization times are shown in Table 5.2. The times given are for a simulated

five days. Unlike initialization, both the number of ecotypes and the length of the run will affect the time spent in clean up. The percentage of time decrease observed for 512 ecotypes is part of a larger picture discussed in later sections.

Table 5.2: Finalization times in seconds for a variable number of ecotypes run on 2 and 4 nodes of the SP2 for a simulated 5 days.

Number of Ecotypes	Number of Nodes					
	1 Node		2 Nodes		4 Nodes	
	Finalize Time	% Total Time	Finalize Time	% Total Time	Finalize Time	% Total Time
32	0.001	0.002%	4.57	3.0%	11.43	5.5%
128	0.005	0.001%	8.05	2.4%	15.84	5.2%
512	0.023	0.0005%	26.3	1.1%	40.5	2.7%

5.4 Parallelization Analysis for the Algae Code

Parallelization was successfully implemented with the original six ecotypes. A comparative illustration for one ecotype from an 80 day simulation run in parallel on 2 nodes from Pool 2 and one node from Pool 2 are displayed in Figure 5.3. The purpose of Figure 5.3 is to illustrate numerical accuracy of both the parallel and the sequential codes. Times for the 80 day simulation were 3.25 hours for the parallel code and 41 minutes for the sequential code. The time step in each case was a half of a second, thus requiring over 13 million communication events. The number of floating point operations for one ecotype per time step is approximately 650.
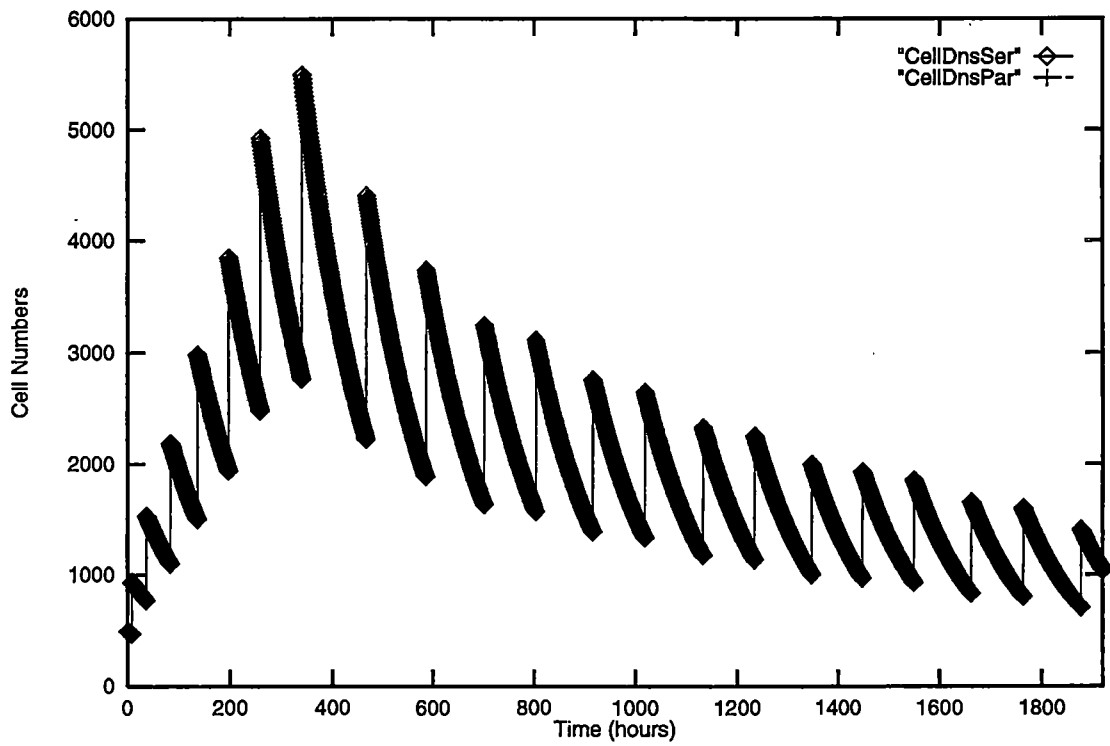
Figure 5.3: Results comparison for one ecotypes produced in parallel on two nodes from Pool 2 on the SP2 and one node from Pool 2. The absence of differences indicates successful parallel implementation of the algae model.

Glancing at the time for execution stated above, it was not too hard to deduce that communication much greater than computation. To reduce the amount of communication and increase the amount of computation, several avenues were explored. These included increasing the time step, increasing the number of ecotypes dispersed across nodes, changing the numerical scheme, and implementing various communication calls provided by MPI. The two most successful approaches were increasing the time step and including more ecotypes per node. All approaches are discussed below.

Increasing the Time Step

Perhaps the simplest way to decrease the amount of communication is to increase the time step as much as possible without changing the system's dynamics. Recall from Chapter 4 that the numerical scheme is a second order Runge Kutta method and that the time step is a half of a second. This time step was pushed to 2.5 seconds before any noticable changes manifested themselves in the population dynamics, thus reducing communication events approximately five–fold.

Increasing the Ecotypes

The most obvious way to increase the $\frac{computation}{communication}$ ratio is to increase the number of computations performed by each node. This is easily accomplished by increasing the number of ecotypes dealt with by each node. To determine at what point the

benefits of parallelization are realized, data sets $2^n, n = 3-9$ were created and five day simulations run on 1, 2, and 4 nodes of Pool 0 on the SP2. Figure 5.4 shows ecotypes vs. seconds/ecotype. For sequential runs, it is expected that because of the way data was added to the data sets, that each addition of new data doubles time to completion. This is evidenced by the straight line from 8 until 256 ecotypes. Parallelization benefits for 2 and 4 nodes appear between 64 and 128 ecotypes per node. Table 5.3 and Figure 5.4 gives the timings for these runs.

At 512 ecotypes an anomolous event occurs that affects all simulations, regardless of the number of nodes. It's not clear why performance diminishes at 512 ecotypes. Initially it was thought that at this size the cache was too small to accomodate the data. (Cache size for each processor on Pool 0 is 128 KB). This is a reasonable assumption for the sequential simulations, but fails in logic for simulations with two and four nodes since the total number of ecotypes is distributed evenly across the processors. In these cases, it would be expected that the processes run before 1024 and 2048, respectively, would show a continued increase in performance, and when these numbers were reached performance would decrease significantly as in the serial case. To be sure, however, the code was run on 2 and 4 processors on the High Nodes and one machine in the Computer Science Cetus Lab, where cache is 1MB and 16KB, respectively. The trend remained the same, however. Figure 5.5 shows the results from Pool 0 and the High Nodes. Table 5.4

Parallel Results for the Algae Model
Run on 1, 2 and 4 Nodes (120MHz each) on the IBM SP2

Parallel Results for Algae Model
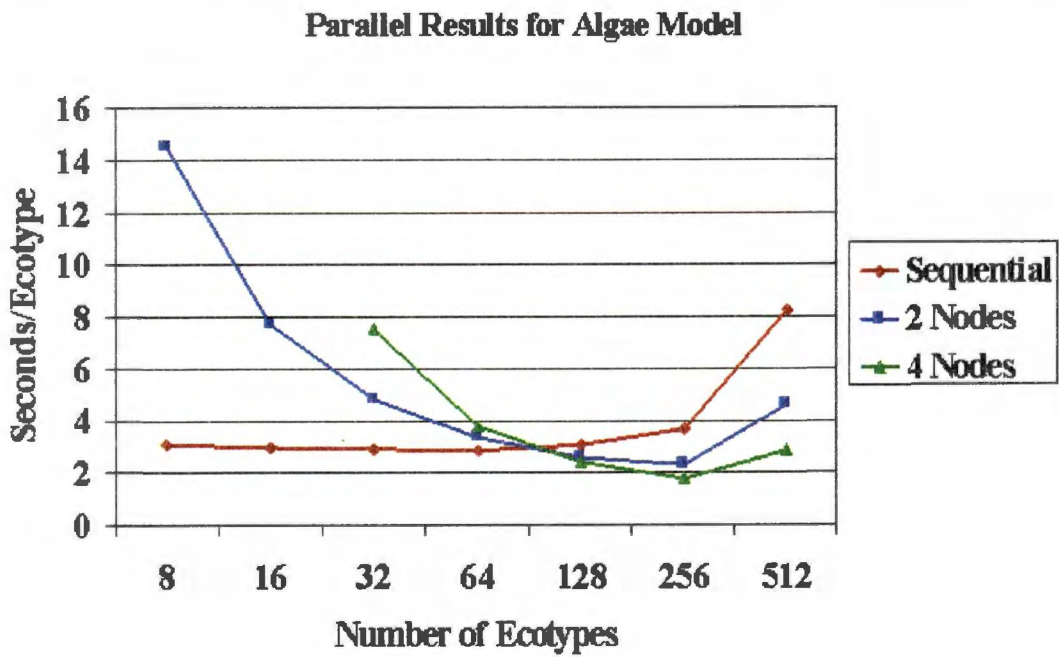
Figure 5.4: Parallel results for the algae model run on 1, 2 and 4 nodes from Pool 0 of the SP2.

Parallel Results for Algae Model Run with 2 and 4 Nodes on SP2 Thin Nodes (120 MHz) and SP2 High Nodes (112MHz)
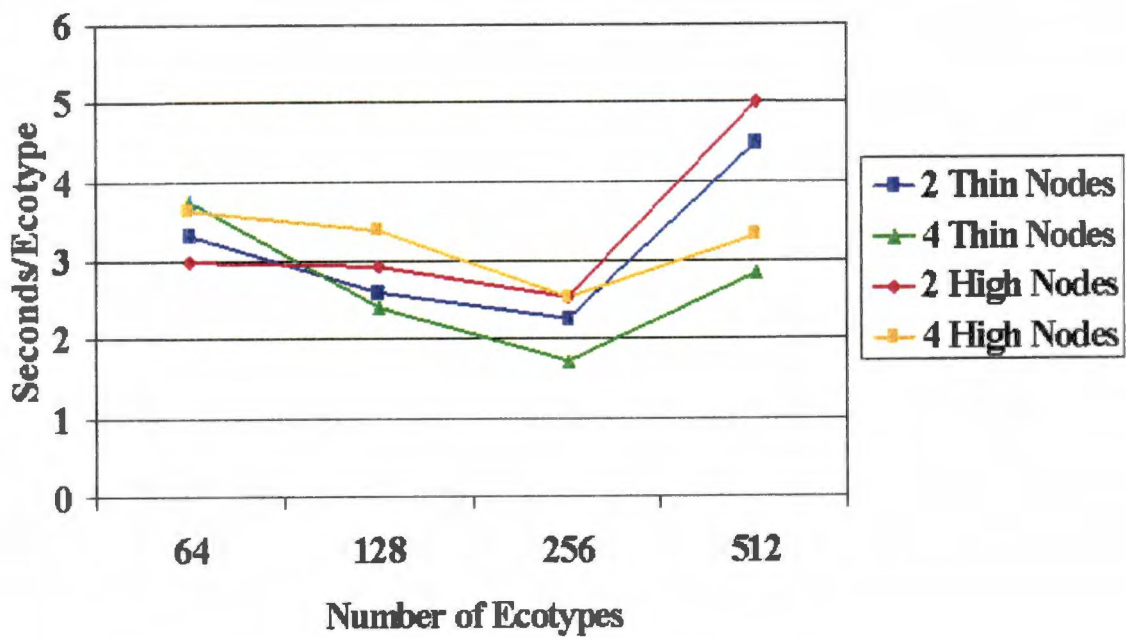
Figure 5.5: Parallel results for the algae model run on 2 and 4 nodes from Pool 0 and the High Nodes.

Table 5.3: Timings for the algae code run for a 5 day simulation on 1, 2 and 4 nodes from Pool 0 on the SP2 in seconds.

Number of Ecotypes	Number of Nodes					
	Sequential		2 Nodes		4 Nodes	
	Time	Sec/Eco	Time	Sec/Eco	Time	Sec/Eco
8	24.26	3.03	115.8	14.5	–	–
16	46.62	2.91	122.71	7.67	–	–
32	91.64	2.86	152.05	4.78	204.03	7.50
64	181.80	2.84	212.58	3.32	239.4	3.74
128	390.60	3.05	330.61	2.58	307.1	2.40
256	937.62	3.66	577.51	2.26	440.82	1.72
512	4214.64	8.22	2353.1	4.59	1458.42	2.84

Table 5.4: Seconds/ecotype for the algae code run for a 2.5 day simulation on 2 and 4 nodes from the High Nodes on the SP2 and sequentially on Cetus2c, and for a 5 day simulation on 2 and 4 nodes from Pool 0 on the SP2. All time in seconds.

Number of Ecotypes	Number of Nodes				
	Cetus2c	Pool 0		High Nodes	
		2 Nodes	4 Nodes	2 Nodes	4 Nodes
64	2.64	3.32	3.74	2.97	3.63
128	2.69	2.58	2.40	2.91	3.38
256	2.69	2.26	1.72	2.52	2.52
512	4.98	4.59	2.84	5.01	3.31

gives the corresponding times.

A second consideration concerned the times for data initialization and finalization, but figures presented in Tables 5.1 and 5.2 do not support this.

A third idea was that it was an ouput buffering problem during simulation reporting. A short experiment on a quiet node of the High Nodes revealed that the times for execution without printing results during code exectution for a 60 day simulation were similar, 41.82 minutes without reporting and 42.89 minutes

130

with reporting.

The observation that the problem occurred with a large number of ecotypes, led to the exploration of the set up of the data files themselves, which it turns out, has an effect. When the ecotypes were spread across four populations (as opposed to two in the previous runs), the code was able to execute faster and the increase at 512 ecotypes was alleviated somewhat. This is observed in Figure 5.6. The simulations in this case were run on Pool 2 for 5 days, where it can be seen that the trend has disappeared. It is clear that there is a correlation between the input file set ups and the speed of the code's execution, but the reason for this correlation is still unclear.

Numerical Schemes

Two alternative numerical schemes were explored in a continued effort to alter the $\frac{computation}{communication}$ ratio. These included a fourth–order Runge Kutta and a devised method that seemingly would allow the time step to increase.

It was hoped that, because the fourth–order Runge Kutta method provides greater accuracy than the second–order Runge Kutta method, the time step could be pushed much further than the 2.5 second limit found above. It was also hoped that because it requires twice as many calculations that in one effort the $\frac{computation}{communication}$ ratio would change in favor of computation. Because of the discrete birthing process, however, underlying assumptions for this method were

Parallel Results for the Algae Model
Run on 2 Nodes (160MHz each) on the IBM SP2

Parallel Results for Algae Model
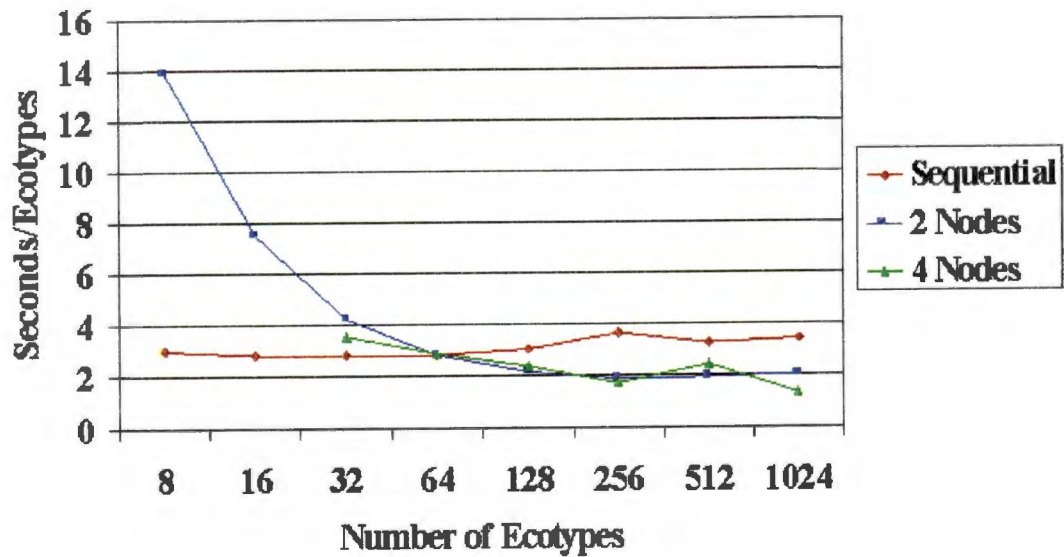
Figure 5.6: Parallel results for the algae model run on 1, 2 and 4 nodes from Pool 2 of the SP2.

violated, and, although, implementation was successful, the time step could only be increased to 1.0 seconds before the dynamics of the population were altered. Since the objective was not attained via this route, the second–order Runge Kutta method with the 2.5 second time step was retained.

The second numerical method explored was entrenched in the fact that since the function is continuous prior to a birthing event, the time step could be greatly increased during this period, and then decreased when any of the ecotypes fell within 10% (for example) of its reproduction threshold. The problem with this approach was that when one ecotype was within its reproductive threshold, all ecotypes were required to compute at the smaller time step because the assumptions concerning shared resources and photosynthesis require communication at each time step. This would be acceptable if the length of time it took for all ecotypes to reproduce was similar. But, ecotypes have varying reproductive thresholds so that for some it takes longer to reproduce than for others. Thus, synchronization between ecotypes could not be achieved. Unfortunately, more time was spent calculating with the smaller time step than with the large. Since nothing was gained from this approach and it was much more complicated than the second–order Runge Kutta method, the second–order Runge Kutta method with time step 2.5 was retained.

Using Different MPI Communication Schemes

Generally message passing libraries have two types of communication options – point–to–point and collective. In an attempt to reduce the amount of communication time, two point–to–point options and one collective option were implemented. The two point–to–point options included various combinations of both blocking and non–blocking sends and receives. The difference between the results of employing these options showed no difference. The advantage of utilizing non–blocking calls is that the code can continue execution while a message is in transit, checking for its completion later. This approach was implemented but, for a small number of ecotypes, showed no gain because the execution time between the send and when the data was required was not enough to affect communication times. Blocking sends and receives were also employed, but they too showed no signs of decreasing communication time as compared to the non–blocking operations. Of the collective communication options, a reduction function, MPI_Allreduce() was used. Communication times between the three methods were similar so that in the end the MPI_Allreduce() function was retained. Times for a simulated 5 days on 2 nodes from Pool 0 are shown in Table 5.5 for the MPI_Allreduce() and the blocking MPI_Send()/MPI_Recv(). Tests were not run for the non–blocking calls when more ecotypes were added to the populations.

Table 5.5: Time comparison between MPI communication options.

Number of Ecotypes	MPI Call	
	MPI_Allreduce()	MPI_Send()/MPI_Recv()
16	124.0	122.71
64	242.0	212.58
256	725.0	577.51

5.5 Conclusions

Current work indicates that the parallel potential of this model is feasible. Currently the reproductive process is being modified by Maria Siopsis for her doctoral thesis. The ensuing changes will result in the further subdivision of ecotypes into cohorts, hence increasing the amount of data to be calculated and the amount of parallelization that can be realized.

Along a different vein, shared memory explorations may prove successful. As of this writing, Jeff Nichols [20] has modified parts of the MPICH shared memory distribution to interact specifically with the *Daphnia*-fish predator–prey system. Further modification for use with the algae model is certainly a viable option.

A final option, would be to relax the assumptions concerning shared resources and photosynthesis. Instead of assuming that all algae uptake evenly from all resources, and that all individuals affect other individuals in the photosynthetic process, local sharing of resources and local interference for photosynthesis might be implemented, with occasional sharing or mixing of available nutrients.

One thing is clear, however, in its current formulation, with the time step and

the aforementioned assumptions, the algae model will be the slowest of the three models. For example, the *Daphnia* model's time step is $\frac{1}{20}$ of a day, so that for any simulated period of time, the communication events required by the algae model are much more frequent than that for the *Daphnia* model. More optimization strategies need to be applied to this model if it is to remain in its current state and if the benefits of parallelization are to be realized not only with this model but with the algae–*Daphnia* system and the three–species food chain.

Chapter 6

Future Directions

There are several mathematical and computational future directions for this model. Of course, there is the previously alluded to coupling of this model with the *Daphnia*–fish predator–prey system, and the ensuing analyses of the algae–*Daphnia* system and the three species food chain in both the unstressed and stressed situations. An alternative includes using this model to examine both the direct and indirect effects of algal blooms on fish and shellfish mortality. Direct effects observed include clogging of the gills of shellfish during blooms of several species of diatom of the genus *Thalassiosira* [26]. Indirect effects observed include oxygen depletion from algal blooms caused by increased nitrogen in the systems [3].

Further refinement of the model holds many options as well. Chapter 3 demonstrated that temperature effects are insignificant when nutrients become limiting and discussed two additional avenues in which temperature plays a role – photo-

synthesis and rate of metabolic reactions [2], [14]. Like temperatures, nutrients also fluctuate so that refining the nutrient pulsing functions are also an option.

Additional exploration of competition between species and seasonal shifts of phytoplankton assemblages is also a feasible alternative. Work by Goldman and Ryther [5] indicate that temperature plays a large role in determining the outcome of competition. Preliminary work in the area of competiton has been done by Hurlebaus [12] in looking at interactions between *Skeletonema costatum* and the theoretical algal species in the transient state. The analysis performed in this thesis coupled with his work is a solid foundation for further investigations into these areas.

The addition of space is always a viable alternative. If space were included in the model, the assumptions blocking parallelization – shared resources and shading – could potentially be relaxed hence increasing the potential for successful parallelization.

Along the vein of successful parallelization, exploring shared memory possibilities would be challenging to the advanced programmer. Currently, Jeff Nichols [20] has code that he has adapted from the MPICH distribution to work with the *Daphnia*–fish predator–prey system. Further refinement would allow its use with the algae code. Progress in this area has actually been made by myself, but time constraints prevented completion of this venture.

Bibliography

Bibliography

[1] National Data Buoy Center. National Data Buoy Center – Station Information. World Wide Web at http://www.ndbc.noaa.gov/stations.shtml, December 1998.

[2] Clinton J. Dawes. *Marine Botany*. John Wiley and Sons, Inc., 1998.

[3] Environmental Defense Fund. Edf Letter. World Wide Web at http://www.edf.org/pubs/EDF-Letter/1988/Sep/j_fishkill.html, 1998.

[4] S. Glasstone, K.J. Laidler, and H. Eyring. *The Theory of Rate Processes*. McGraw-Hill Book Company, Inc., 1941.

[5] J. C. Goldman and J. H. Ryther. Temperature–influenced species competition in mass cultures of marine phytoplankton. *Biotechnology and Bioengineering*, XVIII:1125–1144, 1976.

[6] T. G. Hallam. Physiological stress ecology: Theory, simulations and ecotoxilogical applications in aquatic systems. Lecture Notes, Fall 1997 - Spring

1998.

[7] T. G. Hallam, G. A. Canziani, and R. R. Lassiter. Sublethal narcosis and population persistence: A modeling study of growth effects. *Environmental Toxicology and Chemistry*, 12:947–954, 1993.

[8] T. G. Hallam, R. R. Lassiter, and S. M. Henson. Modelling fish population dynamics: A physiologically–based perspective, 1993. manuscript.

[9] T. G. Hallam, R. R. Lassiter, J. Li, and W. McKinney. Toxicant–induced mortality in models of D*aphnia* populations. *Environmental Toxicology and Chemistry*, 9:597–621, 1990.

[10] T. G. Hallam, R. R. Lassiter, J. Li, and L. A. Suarez. Modelling individuals employing an integerated energy response: Application to D*aphnia*. *Ecology*, 71(3):938–954, 1990.

[11] Shandelle M. Henson. *Individual-Based Physiologically Structured Population and Community Models*. PhD thesis, University of Tennessee and Knoxville, August 1994.

[12] Jochen Hurlebaus. An individual-based model for phytoplankton assemblages with application to S*keltonema costatum*. Master's thesis, University of Tennessee and Knoxville, August 1997.

[13] Joint Institute for Computational Science. Joint Institute for Computational Science – Home Page. World Wide Web at http://www-jics.cs.utk.edu, 1998.

[14] Erik G. Jorgensen. The adaptation of plankton algae: II. Aspects of the temperature adaptation of *Skeletonema costatum*. *Physiologia Plantarum*, 21:423–427, 1968.

[15] Hock Lye Koh, Thomas G. Hallam, and Hooi Ling Lee. Combined effects of environmental and chemical stressors on a model *Daphnia* population. *Ecological Modelling*, 103:19–32, 1997.

[16] S.A.L.M. Kooijman. *Dynamic Energy Budgets in Biological Systems*. Cambridge University Press, 1993.

[17] S.A.L.M. Kooijman, B.W. Kooi, and T.G. Hallam. The stoichiometry of population energetics. Unpublished manuscript submitted June 17, 1997.

[18] Cynthia M. Lovelock. The effects of temperature and dissolved oxygen on a model fish population. Master's thesis, University of Tennessee and Knoxville, August 1996.

[19] A. Mortain-Bertrand, C. Descolas-Gros, and H. Jupin. Growth, photosynthesis and carbon metabolism in the temperate marine diatom *Skeletonema costatum* adapted to low temperature and low photon–flux density. *Marine Biology*, 100:135–141, 1988.

[20] Jeffery Nichols. *Design and Analysis of a Predator–Prey Model.* PhD thesis, University of Tennessee and Knoxville, August 1999.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterline, and B. P. Flanney. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.

[22] Siddharthan Ramachandramurthi, Jeffrey A. Nichols, and Thomas G. Hallam. Ecological assessment in watersheds: Individual-based modeling and parallel simulation. In *Mission Earth Symposium.* SCS Western Multiconference, 1996.

[23] James W. Sinko and William Streifer. A new model for age-size structure of a population. *Ecology,* 48(6):910–918, 1967.

[24] H. U. Sverdrup, M. W. Johnson, and R. H. Fleming. *The Oceans, Their Physics, Chemistry and General Biology.* Prentice–Hall, 1964.

[25] Scott M. Sylvester. Design and analysis of a parallel D*aphnia* model. Master's thesis, University of Tennessee and Knoxville, August 1995.

[26] Fisheries Western Australia. Fish Health Info. – Marine Biotoxins. World Wide Web at http://www.wa.gov.au/westfish/sf/broc/fhinfo/fhinfo04.html, 1998.

Appendix

Appendix A

Model Parameters, Equations and Variables

A.1 Model Parameters – Units and Definitions

The following is a listing of most of the model parameters as found in Hurlebaus [12] along with their associated units, definitions, equation citations as found in this thesis, old code names and new code names.

Table A.1: Nutrient Uptake Parameters.

Nutrient Uptake Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
a	cm	cell wall thickness	1.4	A1.a[k]	ecoprms[i][CWT]
$\nu(T)$	$\frac{\mu mols\ cm}{s}$	transport velocity across cell wall	1.4	A1.z[14][k]	ecovals[i][TEMP]
c_{ϕ_e}	$\left(\frac{s}{cm^3}\right)$	proportionality constant for substance ϕ_e	1.4	A1.c$_{\phi_e}$[k]	ecoprms[i][ϕ_e]
ϕ_e	$\left(\frac{\mu mols}{cm^3}\right)$	external concentration of substance ϕ	1.4	A1.z[i]	sysvars[ϕ_e]
ϕ_i	$(\mu mols)$	internal nutrient ϕ	1.4	A1.z[i][k]	ecovals[i][ϕ_i]
m_{Pr}	$(\mu mols)$	protein reserve or structural material	1.4	A1.x[8][k]	ecoinit[i][MPR]
c_{ϕ_i}	$(nd.)$	proportionality constant for substance ϕ_i	1.4	constant in code	constant in code
n_{ϕ_e}	$\left(\frac{1}{cm^2}\right)$	number of processors on surface of cell	1.4	A1.n$_{\phi_e}$[k]	ecoprms[i][ϕ_e]

Table A.2: Temperature Parameters.

Temperature Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
$s(T_1^K)$	$\left(\frac{\mu mols\ cm}{s}\right)$	constant	1.5	A1.sT1K[k]	ecoprms[i][ST1K]
(T_1^K)	K	choosen reference temperature in K	1.5	A1.T1K[k]	ecoprms[i][T1K0]
(T_A^K)	K	Arrhenius temp. in Kelvins	1.5	A1.TAK[k]	ecoprms[i][TAK0]
(T^K)	K	absolute temperature in Kelvins	1.5	A1.y[7]	sysvars[TEMPINC]

<div align="center">Table A.3: Volume and Surface Parameters.</div>

Volume and Surface Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
V	(cm^3)	volume of cell	1.6	A1.z[5][k]	ecovals[i][VOLUME]
SA	(cm^2)	surface area of cell	1.7	A1.z[6][k]	ecovals[i][SURFACE]
ϕ_s	$(\mu mols)$	level of reserve polys. lipid or protein	1.6, 1.7	A1.x[6-8][k]	ecoinit[i][ϕ_s]
σ_{ϕ_s}	$(\mu mols)$	molecular density of ϕ_s	1.6, 1.7	A1.sigma$_{\phi_s}$[k]	ecoinit[i][SIGϕ_s]

<div align="center">Table A.4: Carbon Pool and Energy Pool Parameters.</div>

Carbon Pool and Energy Pool Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
$L(I)$	$(nd.)$	photosynth. activity	1.10	A1.z[13][k]	ecoprms[i][PHOTO]
$E_d(0)$	$\left(\frac{\mu mols}{m^2 s}\right)$	irradiance just below	1.10	A1.y[17]	sysvars[IRATT]
E_K	$(nd.)$	intersection of irrad.-photosynthesis curve	1.10	A1.E_k[ECO]	ecoprms[i][EK]
K_s	$\left(\frac{1}{m}\right)$	light absorbtion coeff. due to seawater	1.10	A1.K_d	glbprms[KS]
K_p	$\left(\frac{1}{m}\right)$	light absorbtion coeff. due to phytoplankton	1.10	A1.K_p	glbprms[KP]
z_0	(m)	depth	1.10	A1.sz	glbprms[SZ]
L_{max}	$(nd.)$	constant	1.10	A1.L_max[ECO]	ecoprms[i][LMAX]
L_α	$(nd.)$	constant	1.10	A1.L_alp[ECO]	ecoprms[i][LALP]
L_β	$(nd.)$	constant	1.10	A1.L_bet[ECO]	ecoprms[i][LBET]
L_P	$(nd.)$	constant	1.10	A1.L_P[ECO]	ecoprms[i][LP]
M_0	$\left(\frac{1}{\mu mols}\right)$	maintenance cost	1.12	A1.MO[ECO]	ecoprms[i][MO]

<div align="center">Table A.5: Excretion and Respiration Parameters.</div>

Excretion and Respiration Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
j_{ϕ_i}	$(\mu mols)$	constant associated with ϕ_s	1.8	A1.zeta_ϕ_i	ecoprms[i][ZETAϕ_i]
$j_{ex,C}$	(s^{-1})	free Carbon molecules respired per second	1.11	A1.zeta_GL[k]	ecoprms[ZETAGL]

Table A.6: Storage and Structure Flux Parameters.

Storage and Structure Flux Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
m_{0,ϕ_s}	$\left(\frac{\mu mols}{s}\right)$	molecules of ϕ_s fixed during anabolism	1.13	A1.m0ϕ_s[ECO]	ecoprms[i][ϕ_s]
c'_{ϕ_i}	(s)	proportionality const.	1.14	A1.ϕ_i	ecoprms[i][ϕ_i]
κ_{ϕ_i,ϕ_s}	$\left(\frac{\mu mols}{s}\right)$	number of ϕ_i molecules needed to comprise one ϕ_s molecule	1.14	A1.k_ϕ_s	ecoprms[i][ϕ_s]
I_{m,ϕ_s}	$(nd.)$	max rate of production of one molecule of ϕ_s	1.14	A1.I_mϕ_s[ECO]	ecoprms[i][IMϕ_s]
$\kappa_{\phi_s,0}$	$\left(\frac{\mu mols}{s}\right)$	max catabolic rate for substance ϕ_s	1.15	A1.ka_ϕ_s0	ecoprms[i][ϕ_s0]
$\kappa_{\phi_s,1}$	$(nd.)$	constant for substance ϕ_s	1.15	A1.ka_ϕ_s1	ecoprms[i][ϕ_s1]
$\kappa_{\phi_s,2}$	$(nd.)$	constant for substance ϕ_s	1.15	A1.ka_ϕ_s2	ecoprms[i][ϕ_s2]
$\kappa_{\phi_s,3}$	$(nd.)$	constant for substance ϕ_s	1.15	A1.ka_ϕ_s3	ecoprms[i][ϕ_s3]

Table A.7: Miscellaneous Parameters.

Miscellaneous Parameters					
Param.	Units	Definition	Eq.	Orig. ID	New ID
μ_{ϕ_i,ϕ_e}	(nd.)	number of ϕ_i derived from one ϕ_e molecule	1.1	A1.mu_ϕ_i, ϕ_e	ecoprms[i][ϕ_i, ϕ_e]
μ_{ϕ_i,ϕ_s}	(nd.)	number of ϕ_i needed to make one ϕ_e molecule	1.1	A1.mu_ϕ_i, ϕ_s	ecoprms[i][ϕ_i, ϕ_s]
Pr_{min}	$(\mu mols)$	minimal protein required for reprod.	1.16	A1.Prmin[ECO]	ecoprms[i][PRMIN]

Appendix B

Original Temperature Data

The following three tables show the original temperature data for the Gulf of Mexico (B.1), Florida (B.2) and Massachusetts (B.3), respectively.
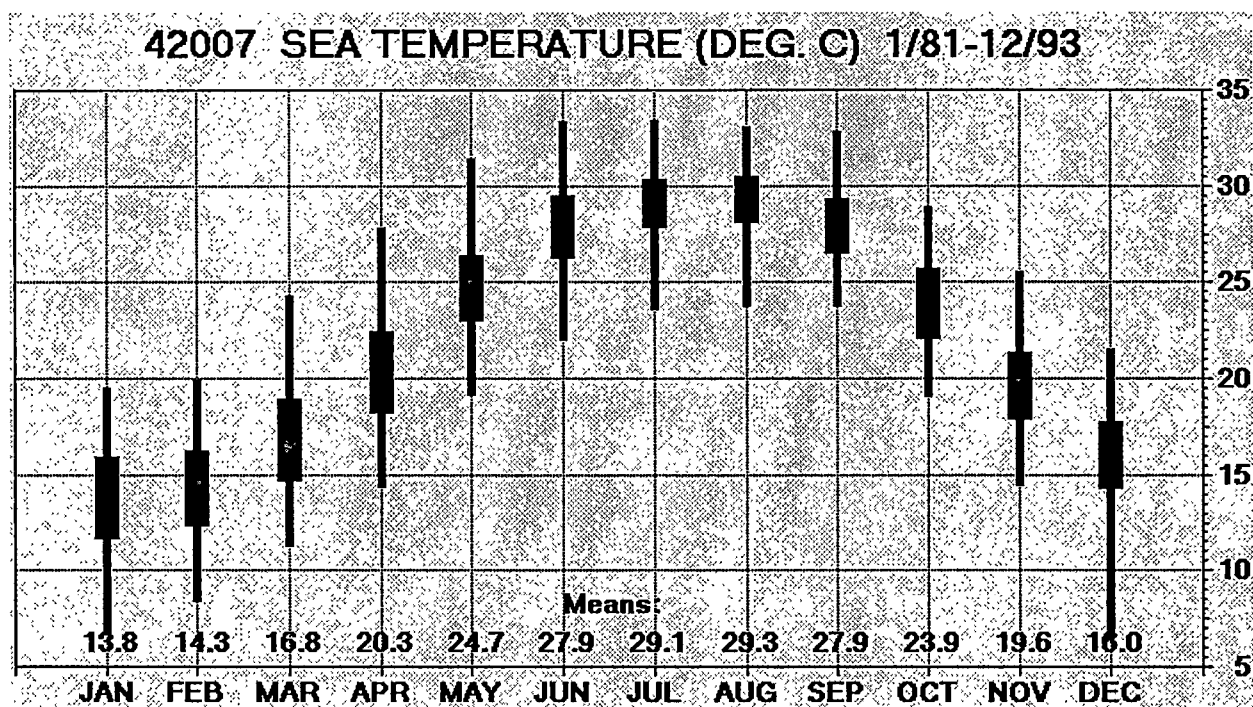
Figure B.1: Data from anchored buoy, south–southeast of Biloxi, Mississippi
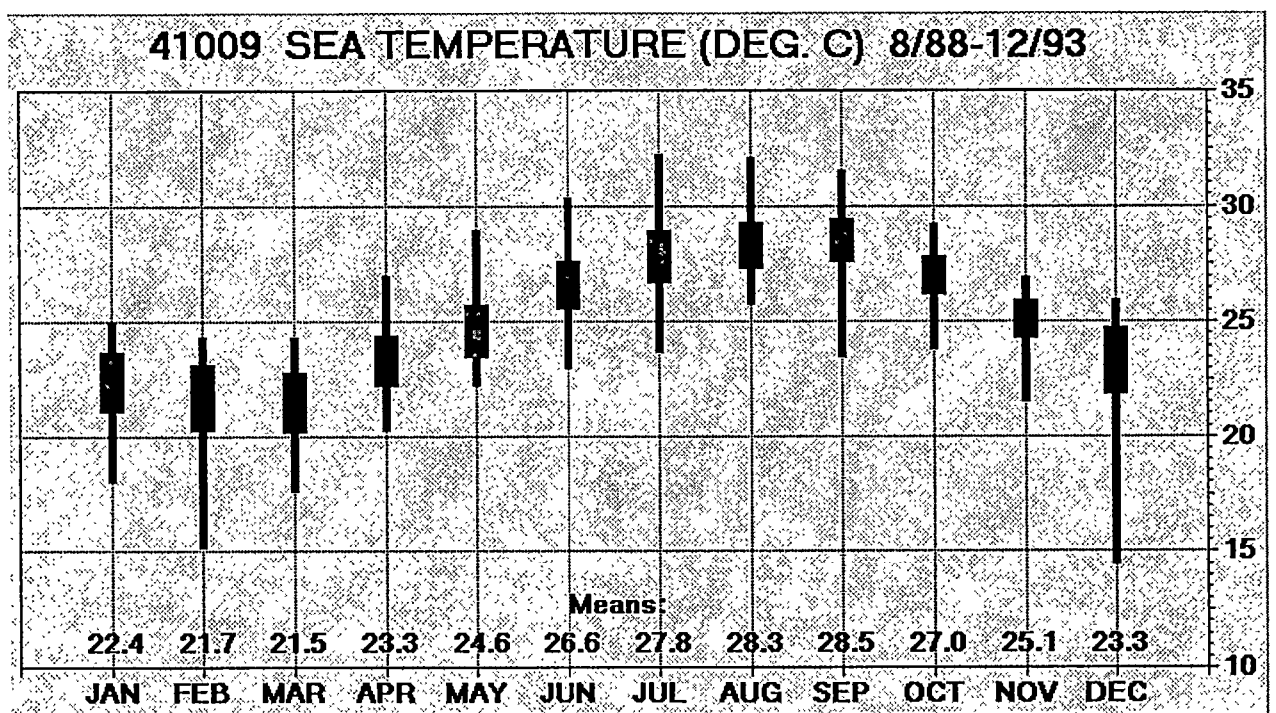
150

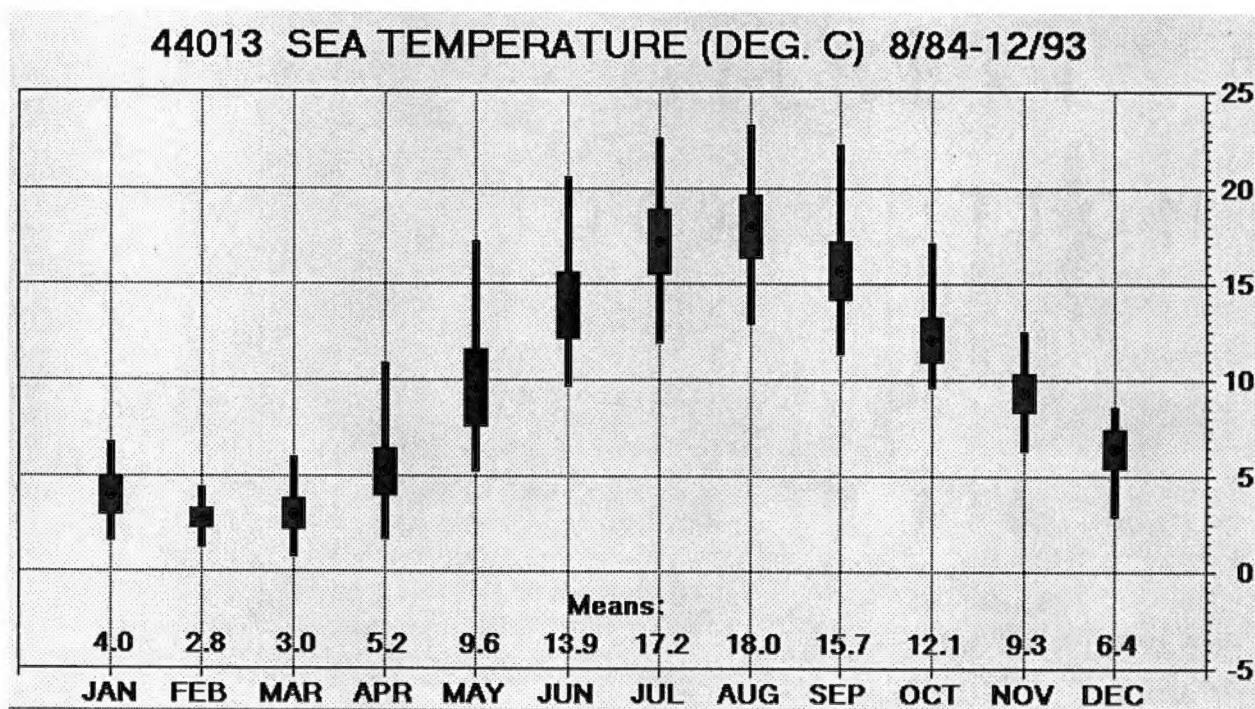Figure B.2: Data from anchored buoy, Cape Canaveral, Florida

151

Figure B.3: Data from anchored buoy, Boston, Massachusetts

Vita

Erin M. Miller was born on July 13, 1969 in San Diego, California. She grew up in Fresno, California and graduated Fresno High School in June 1987. In August of 1987 she entered Fresno State University undeclared. By May of 1988, she felt academically unchallenged and dropped out to establish a course of action. In August 1988 she entered Fresno City College where she planned to fulfill core requirements and transfer to the University of California, Davis. In August 1990 she graduated from Fresno City College with an Associate of Arts degree and subsequently entered the University of California, Davis as a Mathematics major. During her first year, her interests grew in environmental issues so that in September 1991 she switched to a major in Environmental Policy Analysis and Planning. In June 1994 she graduated with a Bachelor of Science in Environmental Policy Analysis and Planning with a Water Quality Emphasis and a Mathematics minor. She took a year off to participate in field research and enjoy cooperative living. In August 1995 she entered the graduate program in Ecology at the University of Tennessee, Knoxville but later switched to the Mathematics department. In August 1999, she received a Master of Science degree in Mathematics. Areas of future exploration include Computer Based Modelling and Computer Science.