Gepasi version 2

© 1989, 1992 Pedro Mendes

Release notes

Simulation with *Gepasi*

Kinetics

Numerical methods

Further reading on metabolic modelling and related subjects

Miscellaneous

Release notes

Aberystwyth, 18 November 1992
This is the third release of *Gepasi* version 2.0 (2.02c)

Gepasi is composed of:

-A MS-Windows front end, with a main program to handle interaction with the simulating engine, **Simulation** (GWSIM.EXE), and a small utility to input reaction schemes and rate-equations, **Topology** (GWTOP.EXE).
-A MS-DOS version of the simulation engine (GEPASI.EXE), which is the program that carries out the simulations.
-This context-sensitive help system (GEPASI.HLP).

The front end runs under MS-Windows version 3.1 in 386 enhance mode.

The simulation engine can be compiled for other operating systems and the source code is available. See Distribution for details.

New in this version:

- arbitrary user-defined rate equations can now be used; these are kept in a database manipulated with **Topology**.
- the installation procedure is now completely carried out with a proper MS-Windows setup program (SETUP.EXE).
- the help file is now complete.

To be realeased are:

-a simple, character based, portable front end, suitable for mainframes and other computers without graphic capabilities.
-a printed user's manual (supplied in postscript format).

Gepasi is a user's program you can help improving it!.

Simulation of metabolic systems with *Gepasi* - basics

Gepasi was built to simulate biochemical reaction systems. Nevertheless, it is as well suited to the simulation of convencional chemical reaction systems. It can even be adapted to other related areas such as pharmacokinetics or population dynamics.

Gepasi is able to simulate models at both the steady-state and transient (dynamic) levels. It produces data for the time-course of reaction and/or steady-state levels of metabolites and magnitude of fluxes, and the associated coefficients of metabolic control analysis (MCA).

In order to use this program for the simulation of such systems the following steps should be taken.

1- Take sufficient time to build a structural model of the metabolic pathway you want to simulate. The detail of this model should be, as a general rule, as simple as possible but still retaining the key properties of the real system. At this stage you should have identified: (i) the steps (reactions *and* relevant transport processes) (ii) the internal metabolites (substrates and products) (iii) external effectors that affect some of the steps and, finally (iv) feedback or feed forward loops within the reaction scheme

2- Identify the kinetics of each step. Take care to check the order of binding of substrates and release of products in steps with more than one substrate or product. Some kinetic mechanisms may also have more than one modifier, in which case they will be numbered in some way. This numbering will determine which to input in **Topology** (GWTOP.EXE) as the first, second, etc. When defining the reaction mechanism in **Topology** it is important to write the substrates and products in the same order they react.

3- You must define the values each kinetic constant will take. Each kinetics has its own associated constants and in order to have a numerical simulation of your model, they must have values. You may have determined these experimentally or have a good estimate or you may even have to make them up. In any case they must be defined. It may be, though, that your interest in making this model is that of finding a good estimate for some kinetic constants. In that case your strategy will be that of testing several values and chosing the one that produces the closest behaviour to that of the real system (see Scanning parameter space).

4- You must define the concentrations of the external metabolites of the system, as these are parameters of your model just like, for example, the kinetic constants. If you will be simulating the dynamics of the model, then you must also decide which values you will assign to the initial concentrations of the internal metaboilites. If your model has the possibility of having more than one steady state or other (unstable or periodic) states, then the starting point of the simulation will decide which state will be reached.

5- If it is the dynamics of the model that will be simulated, you must decide what the final time will be. This will be the (simulation) time after which the simulator will stop. In addition you must also decide upon how many points of the trajectory you want to keep stored in the dynamics data file (`*.dyn`).

6- Enter the reaction mechanism, the kinetic type and modifiers ('loops') with the utility **Topology** (GWTOP.EXE). With **Simulation** (GWSIM.EXE) enter the numerical values of the constants and inital concentraions. Also with **Simulation** activate the simulation engine by pressing the button Run! or the menu with the same title.

Simulation of metabolic systems with *Gepasi* - advanced

Gepasi is able to do more than just the determination of a steady state or trajectory of one model. In most cases you will be interested to know how some properties (variables) of the model change when you change the value of one or more parameters. One way to achieve this is to create as many simulation files as values assigned for these parameters. However, creating a big number of files can be very tedious and, indeed, take more time then the simulations to run! A much better way is to tell *Gepasi* to do these changes automatically: Scanning parameter space.

In most of the times you will want some of the parameters be dependent on others. One frequent arises when you change one kinetic constant but do not want to change the equilibrium constant. This means that for each value of the independent (changing) parameter the dependentmust be adjusted (for example using the Haldane relationships in the case of enzyme kinetics). *Gepasi* can do this automatically for you: Linking parameters

Scanning parameter space

The simulator can be instructed to automatically change a parameter between boundaries. In fact many parameters can be scanned so as to build a sort of box in parameter space where the behaviour of the model is mapped. Of course, if you only chose one parameter to scan, this will be a line, a rectangle if two, a cube if three, or a nD hypercube if more.

There are two ways to scan parameter space: either in *regular intervals* or *randomly*. In both cases it can either be in linear or logarithmic space (so, effectively there are four choices). Some parameters may change in linear space others in log space, it doesn't matter, what cannot be mixed are regular and random scans.

In both cases, for each parameter you must define a lower and an upper limit for the scan. In case the scan is in regular intervals, you must also define a density for each parameter, that being the number of intervals in that dimension. Otherwise, if you choose a random scan, you must define a *global* density, which will then be the *total* number of random points inside the nD-hypercube that the simulator will simulate.

See Scan parameters dialog box and Scan boundaries dialog box.

Linking parameters

When scanning a set of parameters , others can be linked to these. At the present version the only possible link is by product: the linked parameter will be a (sub)multiple of the master parameter. This is enough to adjust one kinetic constant of a chemical reaction to the value of the other in order to maintain the equilibrium constant. It is also useful to have concentrations dependent on each other (for example to keep the same mass action ratio while changing the substrate).

Scanning parameter space with links means that you are changing more than one constant at a time, do not forget this, as it may be very important for the conclusions you make after the results of the simulations.

See Links dialog box to find out how to define links.

Viewing results

This version of *Gepasi* does not have any facilities to help visualization of simulation results. Instead it relies on other specialized applications to achieve this.

After the execution of a simulation, the front-end will ask the user if s/he wants to see the report file (if one was produced). When the answer is positive it automatically activates the **Notepad** with the report. It will also load **Notepad** with the debugging output produced by the simulation engine if this option was turned on.

Results from time courses are output in a columnar file, separated by commas, spaces or tabs (at your selection) and with optional titles for each column at the top or at the bottom lines. These files have always the same name of the simulation file with the `.dyn` extension. They contain the time-dependent values of all the concentratoins and all of the fluxes of the model.

Generally, you will be interested in outputing only selected parameters and variables of the model. This is achieved by the definition of a data file (normally with the extention `.dat`) where you select among all possible output elements those of interest, a suitable column width and a separator (comma, space or tab) and other features. These files will be suitable to be imported into spreadsheet programs where the results can be manipulated and graphical output of the simulation results produced.

See Compatible programs for a list of some compatible software and hints on how to achieve the best results with these.

The simulation engine

The simulation engine is the part that does all the numerical processing for the simulation. It is a portable C program that can be compiled for several operating systems. In this MS-Windows package it is an MS-DOS application (GEPASI.EXE) that is activated from the **Simulation** (GWSIM.EXE) window by pressing the **Run!** button or selecting the equivalent menu entry.

Activating the simulator from the MS-DOS prompt
The simulation engine can be run on its own, from the MS-DOS prompt simply by entering the command GEPASI. You will then be asked to enter the names of the simulation files to process (these files must have been produced with **Topology** and **Simulation**). You can process up to 30 simulation files in one run of GEPASI.EXE. After entering a dot to finish the list, the simulator will start to process each simulation in sequence, producing all the output files selected, until it ends the last simulation. Remember that each simulation can actually be itself a sequence of elementary simulations if some parameters are being scanned.

If in any simulation file debug output was selected, GEPASI.EXE writes to screen the operations it is executing. For the numeric integration you will see lines like this:

```
5th gear , hu=1.34e-002, t=2.78e+001, tcur=2.80e+001, s 1586, f 3241, j 28
|      |      |            |              |             |         |        |
order |     step size    time of        lasttime      number function jacobian
      method             solution       evaluated      steps    evaluations
```

The reason why tcur is sometimes greater than t is that the actual value computed by integration was for tcur, t was obtained by interpolation.

For the steady-state solution you will see lines like this:

```
newton() - iteration  6, .......... re= 6.7993e-001
   |           |            |                 |
function   number of     each dot is     residual
           iterations    a dumping step  error
```

Activating the simulator from other operating systems
The simulation engine can also be run on other operating systems. If a front-end was not supplied, than the simulation engine must be run on its own, for which there is no difference from the MS-DOS version as above. Simulation files have to be created with a front-end and so will have to be transfered from a different machine with the front-end (for example a MS-Windows computer). Why bother doing this? Suppose you are scanning 10 parameters (10D hypercube) in 135000 iterations that last 10 seconds each on a MS-Windows machine, this would take 375 hours (over 15 days...), just imagine how faster that would run on a super-computer!

Example topology and simulation files

To illustrate the usage and performance of *Gepasi*, this package includes some example files. Load them into **Topology** and **Simulation** and play around with them. See how the output looks like, adjust whatever options you need to be able load the data into other programs and do several plots.

Brusselator
SEQFB
Performance benchmarks

Example files - Brusselator

This is the famous oscilating model of Prigogine and Lefever (Prigogine, I. & Lefever, R. (1968) *J. Chem. Phys.* **48**, 1695-1700).

There are two files related with this example: `brussel.top` which contains the topology of the model, and `brussel.sim` which has a set of parameters and initial values that leads to oscilations.

This `brussel.top` file when executed will generate `brussel.dyn` which contains the time series of the concentrations and fluxes of the model. This is a columnar file that can be read into spreadsheet or charting programs. You can plot trajectories using the time values for the abcissa and concentrations for the ordinate. Alternatively you can plot one concentration against the other (X and Y are the only internal metabolites of this model) to obtain a phase space graph of the orbit of the system. You will see that to obtain a detailed picture of this orbit you will have to redefine (increase) the number of points in the integration.

If you want to simulate the model with different intial conditions or parameters you can either edit the `brussel.sim` file and then save it with a different name, or build one from scratch with the menu entry File-New, selecting `brussel.top` as the topology. Do all this in the **Simulation** window.

Example files - SEQFB

This is a model of a branched pathway with sequential feedback, described by Hofmeyr (Hofmeyr, J.-H. S. (1986), *Comput. Aplic. Biosci.* **2**, 243-249). The file `seqfb.top` contains the topology, `seqfb_1.sim` defines the parameters and initial concentrations as in the original publication and `seqfb_2.sim` defines a scan of parameter space of this model.

After processing `seqfb_1.sim` read the report file (`seqfb_1.txt`) to see how the information is laid out.

Load `seqfb_2.sim` and select Options-Data output to see how the data file is defined. And also Scan-Parameters and Scan-Boundaries to see how the scanning of parameters is defined. This simulation will change the Ki and n of the first step (respectively the dissociation constant and Hill coefficient of the inhibitor) and the data file will have these two parameters in the first two columns and the steady-state concentration of B (the product of this step) in the third. This file is appropriate to load into GNUPLOT (see Compatible programs) to produce a 3D graph of [B]ss as a function of Ki and n.

Performance benchmarks

Included in this package are a set of files that were designed to test the performance of *Gepasi* in different conditions. Run these tests to see how fast your computer can carry out simulations. Each test is dedicated to a different feature. All files are based on `alpha1.top` and `alpha2.top`. Both of these files define a pathway of 25 reactions that transform the <u>external</u> substrate A into the <u>external</u> product Z, with all the other letters of the alphabet as <u>internal metabolites</u>, appropriately called 'alphabet pathway'. These two topology files differ only in the kinetics of each reaction: in `alpha1.top` all reactions are of the reversible Michaelis-Menten type (a predefined type) while in `alpha2.top` they are of a user-defined type (called Rev M-M) which has exactly the same rate law as the reversible Michaelis-Menten type. This is to compare the speed of predefined to user-defined kinetic types.

test1.sim - This defines the integration of the alphabet pathway with the *predefined* kinetics for 3000 seconds. It is a test for the speed of integrations. Only one point is generated (at 3000 seconds), otherwise the speed of the hard disk would also affect the result.

test2.sim - This defines the steady-state solution of the alphabet pathway. It is a test for the speed of steady-state solutions.

test3.sim - This defines the steady-state solution of the alphabet pathway with *predefined* kinetic types. It has a choice of parameters and initial concentrations that turn it very difficult to solve. In fact, the Newton method does not converge and therefore integration of the model is carried out to find the steady state. This file has the debug output set so that you can see how what operations the simulator did in order to get to the solution.

test4.sim - This defines the integration of the alphabet pathway with the *user-defined* kinetics for 3000 seconds. Like `test1.sim` it is a test for the speed of integrations and can be used to find the amount of extra time spent with user-defined kinetic types by comparing it to `test1.sim`.

test5.sim - This defines the steady-state solution of the alphabet pathway. Like `test2.sim` it is a test for the speed of integrations and can be used to find the amount of extra time spent with user-defined kinetic types by comparing it to `test2.sim`.

Performance of Gepasi on some different computers

```
computer        co-pro  test1 test2 test4 test5
-------------------------------------------------
Cubit 486/50    intern. 16    2     47    2
Dan 486/33      intern. 22    2     49    2
Viglen 386/33   yes     54    5     122   6
ViglenII (286)  no      -     173   -     173
```

Software that might be useful to process data generated with *Gepasi*

To produce graphical output of *Gepasi*'s results, you will need a plotting program or a spreasheet. Here is a list of some software that has been used for this purpose and some hints on how to achieve good results. This list is by no means complete and the author disclaims any connection with the commercial packages here mentioned.

GNUPLOT
This is a <u>free software</u> package included in the GNU project. It can be found on several Internet archives and public BBSs.

Useful features in GNUPLOT are the ability to produce 2D and 3D parametric plots of data (thus 3D graphs from series of (x,y,z) triplets which are missing in most packages); 3D surface and contour plots (that require grided data); and postscript, HPGL, TeX and LaTex output. GNUPLOT cannot manipulate data like spreadsheets do.

GNUPLOT exists for several operating systems, including MS-Windows and can be ported to others (the advantages of being free software!...)

GRAFIT
This is a MS-Windows commercial package written by Robin Leatherbarrow and published in the U.K. by Erithacus Software Ltd., PO Box 35, Staines, Middlesex, TW18 2TG.

GRAFIT has some spreadsheet capabilities like rescaling whole columns and is able to do non-linear regression with user-defined functions, optional weighting methods and adjustable fitting criteria. Virtually all features of scientific graphs can be adjusted.

EXCEL
EXCEL is commercial spreadsheet program with plotting capabilities sold by Microsoft Corp. Version 4 is able to do 3D plots but the data must be grided (so the output from *Gepasi* has to be processed). Being a spreadsheet it will be useful for manipulating the data.

The best way of having data files read by EXCEL is to have columns sparated by tabs.

Pre-defined kinetic types

Both for steady-state or dynamic simulations, each reaction must have assigned a certain kinetic type. A kinetic type has associated a rate equation and, in most cases, a mechanism. For the purposes of simulation it is the rate equation together with it's kinetic constants that are actually needed.

The kinetic types internally defined in *Gepasi* are:

chemical

Michaelis-Menten

Reversible Michaelis-Menten

Specific inhibition

Catalytic inhibition

Mixed inhibition

Specific activation

Catalytic activation

Mixed activation

Hill

Sequential uni-bi

Chemical kinetics

This kinetic tipe covers a wide range of non-enzymatic reactions, with one, two or three substrates and/or one, two or three products. When reversible, these reactions have associated two rate constants, one for the forward and the other for the reverse direction, when irreversible only one.

reversible mechanism

`k1`   - rate constant for the forward direction
`k2`   - rate constant for the reverse direction

irreversible mechanism

`k1`   - rate constant for the forward direction

Michaelis-Menten kinetics

This is the classical enzyme kinetic mechanism. Catalyses one substrate into one product and it is irreversible and has two kinetic constants:

```
        V*S
v  =   --------
       Km  +  S
```

V   - maximum rate of reaction
Km   - substrate concentration such that v = V/2   (Michaelis constant)

Reversible Michaelis-Menten kinetics

This is the reversible equivalent of the Michaelis-Menten rate-law. It has one substrate and one product and has four kinetic constants:

```
        Vf*S/Kms - Vr*P/Kmp
v =   ---------------------
          1 + S/Kms + P/Kmp
```

Vf   - maximum rate of forward reaction
Kms   - substrate concentration such that v= V/2, at saturating P (Michaelis constant)
Vr   - maximum rate of reverse reaction
Kmp   - product concentration such that v= V/2, at saturating S (Michaelis constant)

Reversible Michaelis-Menten kinetics with specific inhibition

This is the Michaelis-Menten mechanism with a specific (competitive) inhibitor that affects Kms(app), and Kmp(app). It has one substrate and one product, it is reversible and has five kinetic constants:

```
        Vf*S/Kms - Vr*P/Kmp
v =  --------------------------
       1 + S/Kms + P/Kmp + I/Ki
```

`Vf`   - maximum rate of forward reaction
`Kms`   - substrate concentration such that v = V/2, at saturating P and zero I (Michaelis constant)
`Vr`   - maximum rate of reverse reaction
`Kmp`   - product concentration such that v = V/2, at saturating S and zero I (Michaelis constant)
`Ki`   - dissociation constant of the enzyme-inhibitor complex

Reversible Michaelis-Menten kinetics with catalytic inhibition

This is the Michaelis-Menten mechanism with a catalytic (non-competitive) inhibitor that affects Vf(app) and Vr(app). It has one substrate and one product, it is reversible and has five kinetic constants:

```
          Vf*S/Kms - Vr*P/Kmp
v =   -----------------------------
       (1 + S/Kms + P/Kmp)(1 + I/Ki)
```

Vf   - maximum rate of forward reaction
Kms   - substrate concentration such that v = V/2, at saturating P and zero I (Michaelis constant)
Vr   - maximum rate of reverse reaction
Kmp   - product concentration such that v = V/2, at saturating S and zero I (Michaelis constant)
Ki   - inhibitor dissociation constant of the enzyme-substrate-inhibitor complex

Reversible Michaelis-Menten kinetics with mixed inhibition

This is the Michaelis-Menten mechanism with a mixed inhibitor that affects Vf(app), Vr(app), Kms(app), and Kmp(app). It has one substrate and one product, it is reversible and has six kinetic constants:

```
            Vf*S/Kms − Vr*P/Kmp
v =  --------------------------------------
      (1 + I/Ki + S/Kms + P/Kmp)(1 + I/K'i)
```

Vf   - maximum rate of forward reaction
Kms   - substrate concentration such that v = V/2, at saturating P and zero I (Michaelis constant)
Vr   - maximum rate of reverse reaction
Kmp   - product concentration such that v = V/2, at saturating S and zero I (Michaelis constant)
Ki   - dissociation constant of the enzyme-inhibitor complex
K'i   - inhibitor dissociation constant of the enzyme-substrate-inhibitor complex

Reversible Michaelis-Menten kinetics with specific activation

This is the Michaelis-Menten mechanism with a specific activator that affects Kms(app), and Kmp(app).
This type has one substrate and one product, it is reversible and has five kinetic constants:

```
        Vf*S/Kms - Vr*P/Kmp
v =  --------------------------
      1 + S/Kms + P/Kmp + Ka/A
```

Vf   - maximum rate of forward reaction
Kms   - substrate concentration such that v = V/2, at saturating P and zero A (Michaelis constant)
Vr   - maximum rate of reverse reaction
Kmp   - product concentration such that v = V/2, at saturating S and zero A (Michaelis constant)
Ka   - dissociation constant of the enzyme-activator complex

Reversible Michaelis-Menten kinetics with catalytic activation

This is the Michaelis-Menten mechanism with a catalytic activator that affects Vf(app) and Vr(app). It has one substrate and one product, it is reversible and has five kinetic constants:

```
           Vf*S/Kms - Vr*P/Kmp
v =  ----------------------------
       (1 + S/Kms + P/Kmp)(1 + Ka/A)
```

Vf   - maximum rate of forward reaction
Kms   - substrate concentration such that v = V/2, at saturating P and zero A (Michaelis constant)
Vr   - maximum rate of reverse reaction
Kmp   - product concentration such that v = V/2, at saturating S and zero A (Michaelis constant)
Ka   - activator dissociation constant of the enzyme-substrate-activator complex

Reversible Michaelis-Menten kinetics with mixed activation

This is the Michaelis-Menten mechanism with a mixed activator that affects Vf(app), Vr(app), Kms(app), and Kmp(app). It has one substrate and one product, it is reversible and has six kinetic constants:

```
            Vf*S/Kms - Vr*P/Kmp
v =  -------------------------------------
     (1 + Ka/A + S/Kms + P/Kmp)(1 + K'a/A)
```

`Vf`   - maximum rate of forward reaction
`Kms` - substrate concentration such that v = V/2, at saturating P and zero A (Michaelis constant)
`Vr`   - maximum rate of reverse reaction
`Kmp` - product concentration such that v = V/2, at saturating S and zero A (Michaelis constant)
`Ka`   - dissociation constant of the enzyme-activator complex
`K'a` - activator dissociation constant of the enzyme-substrate-activator complex

Hill kinetics

This is the simple empirical cooperative enzyme kinetic mechanism; it has one substrate, one product, it is irreversible and has three kinetic constants:

```
        V*S^n
v =   ----------
       Km + S^n
```

V   - maximum rate of reaction
Km   - concentration of S for which v=V/2
n   - degree of cooperativity of S

Sequential Uni-Bi

This kinetic type has one substrate and two products (or one that is produced in a stoicheiometry of two). The two products are released from the enzyme complexes in a sequential (opposed to random) order. This mechanism is reversible and has six kinetic constants:

```
            Vf*A/KA - Vr*P*Q/KPQ
v =  ------------------------------------
     (1 + A/KA +  + P*Q/KPQ + P/KP + Q/KQ +
      A*P/(KA*KP)*(1 + Vf*(1 - KP*KQ/KPQ)/Vr) )
```

Vf   - maximum rate of forward reaction
Vr   - maximum rate of reverse reaction
KA   - constant associated with the terms in A
KP   - constant associated with the terms in P
KQ   - constant associated with the terms in Q
KPQ   - constant associated with the terms in P*Q

User-defined kinetics

These kinetic types are defined by users and can be virtually any mathematical function, with an arbitrary number of kinetic constants.

The user-defined kinetic types are stored in a database (`user-def.kin`) that can be manitpulated with the **Topology** utility (Edit - Rate Equations). See Kinetic types' editor for details.

This database comes with one type already defined: **substrate inhibition**.

Differential equations

To simulate the dynamical behaviour of pathways the <u>internal metabolite</u> concentrations are described as a set of first order differential equations (ODE) of the type:

$$\partial s_i / \partial t = f(\mathbf{S}; \boldsymbol{K}, t)$$

where s_i is an internal metabolite concentration, part of $\mathbf{S}$, the vector of all internal metabolite concentrations; $\boldsymbol{K}$ is the vector of system parameters (kinetic parameters, <u>external metabolite</u> concentrations, etc.) and f is an appropriate <u>rate equation</u>. The set of differential equations must be solved for several values of time in sequence, until an endtime is reached.

 To simulate the dynamics of a specific model, **_Gepasi_** uses a numerical routine called <u>LSODA</u>. This routine uses a sophisticated strategy to find out if the system of equations is becoming <u>stiff</u> and if so applies the well know BDF (also known as Gear) method, otherwise it uses the Adams method.

Steady-state solutions

To calculate a <u>steady state</u> solution of a certain model *Gepasi* uses a combined strategy. The steady state is, by definition :

$$\partial s_i/\partial t = f(\mathbf{S};\mathbf{\mathit{K}},t) = \mathbf{0}$$

therefore it is a system of (generally non-linear) algebraic equations. This can be solved using Newton's method if one has a "close enough" estimate of the solution.
 A solution can also be obtained by solving the system of (generally non-linear) differential equations at increasing values of the independent variable (time) until a steady state is reached.

 Gepasi tries to find a solution with the <u>damped Newton's method</u> using the set of concentrations of the last time value of the integration as the guess. If this is estimate is not good enough for the damped Newton method to converge then a solution is attempted by means of solving the <u>differential equations</u> for increasing values of time. If a steady state is not reached at 1010 units of time *Gepasi* will leave the problem unsolved and notify the user. (Note that if the unit of time is second, 1010 corresponds approximately to 317 years!)

Metabolic Control Analysis algorithms

Elasticity coefficients are obtained from the partial derivatives of the predefined   rate equations or by finite differences (user-defined rate equations). Concentration control coefficients and combined response coefficients are obtained by the method described in Reder, C. (1988) Metabolic control theory: a structural aproach. *J. Theor. Biol.*, **135**, 175-201. All coefficients are obtained unscaled by this method and are scaled with the appropriate steady state concentrations and fluxes when necessary.
Matrix inversion is done by triangular factorization as described in Press, W. H., Flannery, B. P., Teukolsky, W. H., Vetterling, W. T. (1989), *Numerical recipes. The art of scientific computing.* Cambridge University Press, Cambridge, U.K.. The reduction of the stoicheiometric matrix is done by Gauss elimination with row and column switch as described in Comte, S. D. and de Boor, C. (1980), *Elementary numerical analysis. An algorithmic approach.*, 151-152, McGraw Hill International Book Company, Singapore. The mass conservation relations are obtained in the process of reducing the stoicheiometry matrix and the total amount of each conserved moiety is computed as the sum of the initial concentrations of its components.

Damped Newton method of solving systems of nonlinear algebraic equations

This is a variant of the famous Newton-Raphson method. The solution is obtained from a first guess and then an iterative procedure is applied where the guess is refined until the residual error is smaller than required. If a limit number of iterations is reached without an acceptable solution, the routine exits signaling no conversion.

The iteration of the plain Newton-Raphson method is:

```
h = f(old_x)/f'(old_x)
new_x = old_x - h
```

in the *damped* method if the `new_x` has a larger residual error than `old_x` one looks at `old_x - h/2^i` for 0<=i<=32 and accepts the first such value that has a smaller residual error than `old_x`. If none is found, the procedure exits signaling a local minimum.

LSODA - "Livermore Solver of Ordinary Differential Equations, with automatic method switching for stiff and non stiff problems"

This solver of ordinary differential equations(ODE), part of the ODEPACK package of numerical methods for ODEs was written by:

> Linda R. Petzold and Alan C. Hindmarsh,
> Computing and Mathematics Research Division,
> Lawrence Livermore National Laboratory,
> Livermore, CA 94550, U.S.A.

The C version of LSODA was converted from the FORTRAN original by Hon Wah Tam, Wolfram Research, Inc., and enhanced by Pedro Mendes and Herbert M. Sauro.

More information on LSODA can be found in the following references:

> Alan C. Hindmarsh (1983),   ODEPACK, A Systematised Collection of ODE Solvers, *in Scientific Computing*, R. S. Stepleman et al. (eds.), North-Holland, Amsterdam, pp. 55-64.

> Linda R. Petzold (1983), Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations *SIAM J. Sci. Stat. Comput.* **4** , 36-148.

Reaction editor

This is the dialog box where chemical reactions of a pathway are edited.

Add - parses the reaction typed in the edit box and if it is syntactically correct adds it to the pathway scheme and displays it in the list box. If any errors are found a message box is displayed.
Delete - deletes the highlighted reaction in the list box from the reaction pathway.
Help - calls this help page.
Ok - finishes the edition of chemical reactions of the pathway.
Cancel - closes the dialog box discarding all changes.

Double click on a reaction on the list box to pull it down for editing.

See also Syntax of chemical reactions for an explanation of how to write the reactions.

Syntax of chemical reactions

Chemical reactions are written in the widely accepted way of chemists and biochemists:

 - The reagents on the left side of the equation and the products on the right.
 - To connect both sides a = or a -> ('-' followed by '>') are used. The first for (thermodynamically) reversible and the second for irreversible reactions.
 - Identifiers for chemical species are separated by +.
 - Whenever a chemical species enters a reaction in a stoicheiometry bigger than 1 the stoicheiometry coefficient must be written before the identifier of the species together with a *.
 - To specify that a certain reactant is an external metabolite the user can precede it with a $.
 - Identifiers for chemical species can include and start with any character except '+', '$', '*', or the sequence "->".
 - Identifiers for chemical species are truncated to 20 characters.
 - Reactions may have only reagents or products in which case the missing ones are considered to be external but must not be parameters of any rate equation. External metabolites that are parameters of rate equations *must* be explicitly included in the model

Examples of valid reactions:

```
pyruvate + NADH = lactate + NAD
ADP + $Pi -> ATP
-> glucose
```

Examples of invalid reactions:

```
pyruvate + NADH = lactate + NAD+
```
("NAD+" is an invalid identifier)
```
S1 + S2 - S3
```
('-' is not a legal character to separate reagents and products)
```
A B = P Q
```
('+' missing between A and B and P and Q)

Kinetic type editor

This is where <u>kinetic mechanisms</u> are assigned to each reaction.

For each reaction of the list box on the left select one kinetic type on the list box on the right.

Press **Ok** to exit the dialog box keeping all alterations made.
Press **Help** to calls this help page.
Press **Cancel** to exit the dialog box discarding all changes.

Metabolite editor

In this dialog box, users can change names and the *status* of the metabolites. This dialog box can, and indeed will, be avoided by users: both the metabolite names and the *status* of metabolites can be specified in the reaction editor. However it will be rather useful to change the name of metabolites as doing so in the reaction editor would require changes of all reactions where the target metabolites enter. It is also useful for checking if the metabolites have the required *status* as the reaction editor does not show wether they are internal or external.

To toggle between internal or external click on the required radio button.

Press **Ok** to exit the dialog box and keep the changes.
Press **Cancel** to exit the dialog box and discard all changes.
Press **More** to view more metabolites (invisible if less than 10 metabolites).
Press **Help** to view this help page.

Loop editor

This is where metabolites are assigned to be modifiers of certain reactions of the pathway.

Certain kinetic types may have more than one modifier and you will have to select an equal number of metabolites to have these functions. Please note that the numbering of the modifiers is not random: modifier number one may be a competitive inhibitor and modifier number two a catalytic activator. This is specific to each mechanism. See Kinetic types for details on the predefined mechanisms. Take this into account when you add your own mechanisms.

Select each reaction listed and assign its modifiers by selecting them from the combo-box on the right. You can also type in a metabolite that is not part of the list (one that does not take part in the reaction mechanism) and it will be automatically added to the model.

For reactions that take only one modifier, the **Next modifier** button is not visible.

Press **Next modifier** to select the next modifier for the highlighted reaction.
Press **Ok** to exit the dialog box keeping all alterations made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Pathway title

This is where you can define and alter the title of the pathway.

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box discarding all changes.

Kinetic type database

This is where you can add, change or delete kinetic types from the database (`user-def.kin`). In order not to lose any changes to this database, you should only edit kinetic types when there is <u>only one instance of the **Topology** utility running</u>. This is because each running instance of **Topology** reads the database when loading and stores it in memory, if any changes were made it writes the database back to the file when closing. In future versions this inconvenient restriction will probably disappear.

The list box on the left contains the existing kinetic types on the database. Double click on a kinetic type to edit it (the same as the **<u>C</u>hange** button).

Close - finishes the edition of kinetic types.
<u>H</u>elp - calls this help page.
<u>C</u>hange - brings up a dialog box with the rate equation of the selected kinetic type to be edited. No action if there is no selected kinetic type.
New - brings up a dialog box to edit and parse the rate equation for a new kinetic type.
Delete - deletes the selected kinetic type in the list box from the database.

Scan parameters dialog box

This is where you can select the parameters that you want to be scanned.

The **Available** list box on the left contains all the parameters that can be selected for this purpose. Double-clicking on any of these will add them to the list box with the **Selected** parameters. To remove parameters from the **Selected** list just double-click on their entries on this list. Below each list box is a count of its items.

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box discarding all changes.

Scan boundaries dialog box

This is where you define the boundaries and scan density of the parameters that will be scanned.

The list box on the left (**Parameters**) contains a list of the scan parameters defined in the Scan parameters dialog box. Select each of these parameters at a time to define the values of the **Lower** and **Upper** boundaries as well as the **Density** for the scan of this parameter. When the scanning method is random, the density will be the total density of the scan and not for each variable. When scanning with regular intervals, this density is for the relevant parameter only.

 Be careful to always check the number of iterations (displayed in **Simulation**'s main window) before activating the simulation engine. Take into account the fact that each iteration may well take several minutes (depending on you hardware and on the complexity of the model). When you are not sure how long one iteration takes, deselect scanning and try doing just one simulation with the debugging log turned on (see Data output). At the end of the simulation you will be able to see in the log file how long it took to do just one iteration.

Tick the **Log** box to have this parameter changing in logarithmic space.

Don't forget that you must define upper and lower boundaries for all the parameters on the list box. Failing to do so will eventually result in a floating-point error while executing the simulation!

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box discarding all changes.

Rate equation editor

This is where rate equations are entered. Write them as a mathematical expression using the operators and functions listed below together with brackets, numeric constants and simbols (identifiers) that will later be characterised as substrates, products, modifiers or constants. There is no restriction on the complexity of the rate equations apart from memory requirements. After pressing **Ok**, the program parses the expression entered and if there are errors it will warn the user and wait for the correction.

+	addition
-	subtraction
*	multiplication
/	division
^	power
log()	logarithm base 10
ln()	logarithm base e
exp()	exponential base e
sin()	sine
cos()	cosine

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Number of substrates and products

In this dialog box you must input the number of <u>different</u> substrates and products of the kinetic type being defined.

In **No. substrates** write the number of substrate species.
In **No. products** write the number of product species.

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Rate equation details

In this dialog box you define some details of the kinetic type.

Select each identifier of the rate equation in the list-box and select its type, **Constant**, **Substrate**, **Product** or **Modifier**.

Enter a **Title** under in the edit-box and check the **Reversible reaction** box if this reaction is thermodinamically reversible.

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all

Links dialog box

This is where you can add and change <u>links</u> between parameters of the simulation.
The list box contains the parameters that are linked to others.

Press **Ok** to exit the dialog box keeping the changes made.
Press **<u>Help</u>** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.
Press **<u>Edit</u>** to change the selected <u>link</u>. This will bring up the <u>Edit links</u> dialog box.
Press **<u>Add new</u>** to add a new <u>link</u>. This will bring up the <u>Edit links</u> dialog box.
Press **<u>Delete</u>** to delete the selected <u>link</u>.

Edit link dialog box

This is where you actually edit a link.

Select one parameter from the **Parameter:** list to be linked with a parameter selected from the **Linked with:** list. Change the scaling constant appropriately.

The equation below the two lists reflects the values of the linked parameter (on the left). The simulating engine adjusts the linked parameter with this expression every time it changes the value of any other parameter being scanned.

Press **Ok** to exit the dialog box keeping the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Metabolite initial concentrations

This is where the initial concentration of metabolites is assigned. Take notice that there are two kinds of metabolites: external metabolites which are parameters of the system and therefore have to be set (some may be zero but in general will have finite values); and the internal metabolites which are variables of the system and can therefore start from any value. However in many cases different sets of initial concentrations of these metabolites will lead to different steady-states (or even to other less stable states). This is why when lists of parameters are shown, these metabolite concentrations are also included.

Enter the appropriate values in each edit box. The names of the metabolites are displayed on the left.

For models with less than ten metabolites, **More** is not visible.

Press **More** to view the following ten metabolites. At the end, it circles back to the first ten.
Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Kinetic constants

This is where the magnitudes of the kinetic constants are defined.

The list on the right contains all the steps of the model. Select each one at a time in order to assign values to all constants. The name of the currently selected step is displayed above the list box.

The wide button on the top centre of the dialog has the name of the kinetic type of the currently selected step. Press it to obtain information on this kinetic type.

When the kinetic type of the currently selected step has less than six kinetic constants, **More** is not visible.

Press **More** to view the following six constants of this step. At the end, it circles back to the first six.
Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Execution tasks

In this dialog box the tasks to be executed in the simulation are selected. These tasks are divided in two groups: the simulation tasks and the output tasks.

Simulation

Check the **Dynamics** box for the simulation of the dynamics of the model by solution of the <u>differential equations</u>. In this case you must enter values for an **End time** at which the trajectory stops and a number of **Points** for the output file (`*.dyn`) containing the values of the trajectory. This `*.dyn` file is always generated when **Dynamics** is selected.

Check the **Steady state** box to have the simulation engine determine a <u>steady state solution</u>, using as a first guess the values of concentrations specified or *if **Dynamics** is selected* the concentrations at the endtime.

Output

Check the **Report** box to have a report file produced (`*.txt`) with the results of the simulation (see also <u>Report options</u>).

Check the **Data file** box to have a columnar data file produced with the results of the simulation (see also <u>Data file format</u>).

Check the **Debug log** box to have a debug log file produced (`*.log`) with information on the actions taken by the simulator.

Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Report options

In this dialog box one can chose optional data to be written to the report file.

Check the **Structural analysis** box to have the reduced stoicheiometry and reduction matrices included in the report.

The **Stability analysis** box cannot be selected as this feature is not yet part of *Gepasi*.

MCA
Check the boxes of the MCA coefficients that you want to have in the report file. The elasticity coefficients that will be written in the report are only those different from zero. All the control coefficients will be written to the file if selected here, note that this includes *all* possible reference fluxes and metabolite concentrations (see further reading on MCA)

Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Numerical parameters

In this dialog box some parameters of the numerical methods can be altered.

Relative tolerance and **Absolute tolerance** are the tolerances of the integration methods used by LSODA.

Adams max order is the maximum order that LSODA will use for the Adams method in non-stiff regions of the integration.

BDF max order is the maximum order that LSODA will use for the BDF method in stiff regions of the integration.

Flux resolution is the criterium used by the damped Newton method to identify a steady state. If the sum of the squares of the rates of change of all metabolites is smaller than this value then a steady state is recognised.

Press **Defaults** to reset all the parameters to the preset default values.
Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Columnar data file format

In this dialog box several options can be set to tailor the format of the columnar output data file (these options also apply to the trajectory files `*.dyn`).

Under **Available** is a list of all the quantities that can be written in the results data file. Select them to be columns of this file by double-clicking.

Under **Output (columns)** is a list of the already selected items. Each of these will become a colums of the data file, the topmost on the list will be the first (left) column and so on. To delete items from this list select them and double-click the left mouse button. Press **Clear** to delete all items from this list. Pressing **Up** will push the selected item on the list one position up (one column to the left), if no item is selected, nothing happens.

Press **File** to change the name for the data file and check **Append** if you want the file to be appended, rather than overwritten (if it already existed before the simulation). Append will be useful to have the results of different simulations appended on the same file.

Select the **Separator** of columns (the character that separates adjacent columns) from the combo-box. Select the **Column width** (the number of characters the column will have) from the listed values in the combo-box.

In **Titles** select where you want the column titles to be: at the **Header** (top), **Footer** (bottom) or if **None** should be written to the file. Also check the **Quotes** box if you need the titles to be bounded by quotes (needed on some spreadsheets, like Borland's Quattro).

In **MCA** select how you want the coefficients of metabolic control analysis to be output: **Scaled** or **Unscaled**, that is the logarithmic derivatives or the simple ones.

In **Apply to** you select what data is the data file going to have: that from the **Dynamics** (at the end of the trajectory) or the **Steady state**.

Press **Ok** to exit the dialog box and keep the changes made.
Press **Help** to call this help page.
Press **Cancel** to exit the dialog box and discard all changes.

Distribution of *Gepasi*

Gepasi is available from it's author by post or eMail:

post   Pedro Mendes
        Dept. of Biological Sciences
        University of Wales, Aberystwyth,
        Dyfed, SY23 3DA, United Kingdom
eMail   `prm@aber.ac.uk`
fax   ++ 44 970 622350
phone   ++ 44 970 622353

Gepasi is also available from the Internet. It is currently stored in the archive SIMTEL20 at `wsmr-simtel20.army.mil` in the directory `<MSDOS.BIOLOGY>` and therefore in `wuarchive.wustl.edu`, `src.doc.ic.ac.uk`, `garbo.uwasa.fi` and other SIMTEL20 mirrors.

If you want to receive bug-reports or new version announcements, subscribe to the BTK-MCA mailing list.

BTK-MCA mailing list

There is a mailing list on the Internet and connected networks where metabolic control analysis, metabolic modelling and related subjects are discussed. If you have access to email this might be the ideal place to ask questions about *Gepasi*. Bug reports and general notices about *Gepasi* will be posted in this list.

To subscribe to this list send an email message to `biosci@net.bio.net` asking to be added to the list. To post your messages to the list send email to `btk-mca@net.bio.net`. For more details about this see the file `btk-mca.txt` in the `docs` subdirectory of your *Gepasi* directory.

Tailoring *Gepasi* to your needs

Gepasi can be altered by you! The source code is available from the author (see   <u>Distribution</u>). As your changes will probably be useful to the rest of the users, you are kindly requested to send them to the author (see addresses in <u>Distribution</u>) so that it can be included in the next version, together with other people's alterations. You will be included in the list of contributors or, if the changes or additions are major, in the list of authors.

Even if you do not feel comfortable with programming in the C language or with the MS-Windows system calls and find bugs or think *Gepasi* should be altered in any way, the author would like to hear your opinion. As implied in the GNU license, the author is by no means obliged to provide any assistance but will probably be interested in pertinent suggestions (given time to work on them).

Acknowledgements

The following people and organizations contributed in a way or another to the making of *Gepasi*:

Jeremy Bennett, Athel Cornish-Bowden, Karel van Dam, Alan C. Hindmarsh, Jannie Hofmeyr, Horst Holstein, J.N.I.C.T., Douglas B. Kell, Pedro Moniz Barreto, Linda Petzold, Herbert M. Sauro, Hon W. Tam, The University of Wales Aberystwyth, Ana Vaz Gomes, and   Hans V. Westerhoff.

I thank all of them.

Suggested reading on enzyme kinetics

Cleland, W. W. (1963), The kinetics of enzyme-catalyzed reactions with two or more substrates or products. I. Nomeclature and rate eqautions. *Biochim. Biophys.Acta* **67**, 104-137.

Cornish-Bowden, A. (1979), *Fundamentals of enzyme kinetics*, Butterworths, London & Boston.

Fersht, A. (1984), *Enzyme structure and mechanism*, 2nd edition, Freeman, New York.

Keleti, T. (1986), *Basic Enzyme Kinetics*, Akadémiai Kiadó, Budapest.

Wharton, C.W. & Eisenthal, R. (1981), *Molecular Enzymology*, Blackie & Son, Glasgow.

Suggested reading on metabolic modelling and computer simulation

Garfinkel, D., Garfinkel, L., Pring, M., Green, S. B. & Chance, B. (1970), Computer aplications to biochemical kinetics. *Ann. Rev Biochem.* **39**, 473-498.

Garfinkel, D. (1981), Computer modeling of metabolic pathways *Trends Biochem. Sci.*   --

Hayashi, K. & Sakamoto, N. (1986) *Dynamic analysis of enzyme systems*, Springer-Verlag, New York.

Heinrich, R., Rapoport, S. M. & Rapoport, T. A. (1977), Metabolic regulation and mathematical models. *Prog. Biophys. Molec. Biol.* **32**, 1-82.

Hofmeyr, J.-H. S. (1986), Steady-state modelling of metabolic pathways: A guide for the prospective simulator. *Comput. Aplic. Biosci.* **2**, 5-11.

Reich, J. & Sel'kov, E. E. (1981) *Energy metabolism of the cell.* Academic Press, New York.

Suggested reading on metabolic control analysis

Cornish-Bowden, A. & Cárdenas, M. L. eds.(1990) *Control of metabolic processes*, Plenum Press, New York.

Fell, D. A. (1992), Metabolic control analysis: a survey of its theoretical and experimental development. *Biochem. J.* **286**, 313-330.

Kakser, H. and Burns, J. A. (1973), The control of flux. *Symp. Soc. Exp. Biol.* **27**, 65-104.

Heinrich, R. and Rapoport, T. A. (1974), A linear steady-state treatment of enzymatic chains. General properties, control and effector strength., *Eur. J. Biochem.* **42**, 89-95.

Reder, C. (1988), Metabolic control theory: a structural aproach. *J. Theor. Biol.* **135**, 175-201.

Suggested reading on numerical analysis

Conte, S. D. & de Boor, C. (1980), *Elementeary numerical analysis. An algorithmic approach.* McGraw Hill , Singapore.

Gear, C. W. (1971), *Numerical initial value problems in ordinary differential equations.* Prentice-Hall, Englewood Cliffs, N.J.

Press, W. H., Flannery, B. P., Teukolsky, W. H., Vetterling, W. T. (1989), *Numerical recipes. The art of scientific computing.* Cambridge University Press, Cambridge, U.K..

Stoer, J. & Bulirsch, R. (1980) *Introduction to numerical analysis*. New York: Springer-Verlag.

A **steady state** of a metabolic pathway is a state where the internal metabolite concentrations are constant in time. If all fluxes are zero then the state is referred to as equilibrium. A **steady state** can be stable or unstable. It is stable if after small perturbations it relaxes to the original (steady) state. It is unstable if it relaxes to a different state or enters a periodic or chaotic motion.

A **step** of a pathway can be a chemical reaction, a transport event or indeed any event that connect individual pools of metabolites (one chemical species can have several individual pools)

A **pool** is an ensamble of molecules with the same chemical and physical properties. Thus in a well stirred solution with two chemical species there are two **pools**. If there is an interface separating two regions of this solution (effectively creating two solutions) then there are four **pools**. The molecules that form a **pool** are effectively in equilibrium, and so **pool** refers to the amount of a chemical substance within one phase.

Feedback loop is a term that is used to describe the mechanism by which a metabolite downstream (towards the product) in a pathway inhibits or activates a step upstram (towards the substrate). See also feedforward.

Feedforward loop is a term that is used to describe the mechanism by which a metabolite upstream (towards the substrate) in a pathway activates or inhibits a step downstream (towards the product). See also feedback.

The **flux** of a metabolic <u>step</u> is the rate at which its product is being formed. In a pathway at <u>steady-state</u> there will be as many different **fluxes** as branches in that pathway. In non-steady state conditions, there are as many different **fluxes** as steps.

The **status** of a metabolite is the property of being an <u>internal</u> or <u>external</u> metabolite.

Internal metabolites of a pathway are those that are dependent on the system's parameters and initial values, opposed to the external metabolites. Internal metabolites are variables of the system.

External metabolites of a pathway are those that are set by the modeller (or experimentalist)   and are not dependent on the system's parameters and initial values, opposed to the internal metabolites. The concentration of **external metabolites** is usually constant in time but can vary according to any arbitrary explicit function.

Rate equations are equations that relate the rate of reactions to the concentration of certain metabolites called effectors. Effectors can be substrates, products, inhibitors, activators or allosteric modulators. Substrates are the metabolites being transformed into the products. Inhibitors decrease and activators increase the rate of reaction, but are not transformed. Allosteric modulators change the shape of the rate curve (rate *v.* substrate concnetration). See Kinetic types for details on available **rate equations**.

A **modifier** is a substance that changes the rate of a reaction but that does not enter the reaction. **Modifiers** are usually either inhibitors or an activators, although for some mechanisms they could take both roles, perhaps at different concentration ranges.

A system of ordinary differential equations (ODE) is **stiff** when some components of the solution differ from the others by several orders of magnitude. A set of ODEs is not necessarily **stiff** in the whole domain of time. **Stiff** systems of ODEs cannot be solved by ordinary methods because of instability, methods to deal with stiff problems are generally implicit.

To **scan** a parameter of a model is to run several simulations of the model with that parameter taking different values while keeping the others constant. Alternatively, some of the other parameters can be linked to the later.

A **link** is a methematical expression that defines the value of one parameter as a function of others. **Links** are useful to allow some parameters to shadow others being scanned. For example, to keep an equilibrium constant fixed while changing a kinetic constant, one would link other constants to that, such that changes in the former would be compensated.