

Gepasi version 2

© 1989, 1992, 1993 Pedro Mendes

NO WARRANTY

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation (see [Copyleft](#)).
02139, USA.

Release notes

Simulation with *Gepasi*

[Basic topics](#)

[Advanced topics](#)

[The simulation engine](#)

[Viewing results](#)

[Examples](#)

Kinetics

[Pre-defined kinetic types](#)

[User-defined kinetic types](#)

Numerical methods

[Differential equations](#)

[Steady-state solution](#)

[Metabolic Control Analysis](#)

Further reading on metabolic modelling and related subjects

[Computer simulation in biochemistry](#)

[Metabolic Control Analysis](#)

[Enzyme kinetics](#)

[Numerical methods and algorithms](#)

Miscellaneous

[Limits](#)

[Distribution](#)

[Mailing list](#)

[Your own changes to *Gepasi*](#)

[Acknowledgements](#)

[Copyleft](#)

Release notes

Aberystwyth, 26 August 1993

This is the release 2.08a of **Gepasi** version 2.0

- Following a period of intense testing with Jon Dean, several memory allocation errors have been corrected in all the modules. This has cured a rare (but annoying) crash of the simulation engine. It has also enabled the front-end to run under MS-Windows 3.0.
- A bug which appeared in version 2.07b related to the possibility of assigning kinetics with higher molecularities than the reactions has been fixed.
- The simulation engine is now compiled with Visual C/C++ version 1, resulting in slightly faster execution of simulations.

Notes from the previous releases

Aberystwyth, 30 June 1993

This is the release 2.07b of **Gepasi** version 2.0

- It is now possible to assign a kinetic type to reactions with higher molecularity than that type (**Topology**). When that is done, only the first n substrates will be used (if n is the molecularity of the kinetic type) and so the order in which they are written in the reaction editor is very important. Such a reaction is then of order zero in terms of the substrates (or products) that do not enter the kinetic equation (meaning that whatever their values are, they do not influence the rate of reaction).
- A bug in memory allocation (**Topology** and **Simulation**) is currently being tackled (solved?). This bug resulted in a message warning of 'not enough memory' when loading files; the fault is not lack of memory.

Aberystwyth, 9 May 1993

This is the release 2.07a of **Gepasi** version 2.0

- A bug in the integration loop was responsible for the production of erroneous time-course data in some cases where there were conservation relations. This only affected time course data, not steady-state solutions. The error has been corrected.
- The kinetic type 'chemical' was renamed to 'mass action'.
- No blank line is produced in data files if the scan is only in one dimension.

Gregynog, 15 March 1993

This is the release 2.06a of **Gepasi** version 2.0

- There is now the option of having contour lines of 3D plots projected on the XY plane and also of hidden line removal. Both features only work for perfect gridded data (that is, when there are no data points missing due to floating-point errors, in which case the outcome of these options is unpredictable).
- When scans are produced, there is now a blank line between each complete variation of the first parameter scanned. This will avoid having lines connecting the last point of a complete scan with the first of the next in GNU PLOT.
- The Hill rate equation has been modified such that K^h is now used instead of K . Also K was wrongly named K_m before this change.
- The editor of user-defined kinetic types (**Topology**) now remembers the type of the symbols used in the rate equation, if it is being modified.
- Metabolites which are activators of some step(s) in a model do not have to be different from zero anymore (although when they are the rate of such steps is zero).
- When **Simulation** (`gwsim.exe`) is called with a filename as argument it will automatically load

that file. This means that the `.sim` extension can be associated with `gwsim.exe` (in File Manager) such that double-clicking `.sim` files will automatically run **Simulation** and load the file.

- The same as above for **Topology** and `.top` files.
- Both in **Simulation** and **Topology** the F1 key now activates the help, as standard in MS-Windows.
- There were corrections on the definitions of some kinetic constants in this help file.

Aberystwyth, 22 February 1993

This is the release 2.05c of **Gepasi** version 2.0

- When external metabolites were scanned, the simulation engine failed to actually change these parameters so that no scan occurred. This is now working as expected (in fact the problem only existed in version 2.05a and 2.05b).
- GNUPLOT is now called iconized, but the graph window is still visible. This just speeds up the process of producing a graph directly from the **Plot** menu.

Aberystwyth, 10 February 1993

This is the release 2.05b of **Gepasi** version 2.0

- A bug in `.sim` file saving was corrected.
- Flux resolution is now a parameter that can be scanned and output in `.dat` files.
- A new set of examples was added.

Aberystwyth, 5 February 1993

This is the release 2.05a of **Gepasi** version 2.0

New in this version:

- **Gepasi** now accepts up to 45 metabolites and 45 steps.
- Initial concentrations are now available for output in data files. Concentrations are now labeled `[]i` (initial) and `[]f` (final).
- 2D plots can display multiple data series. Just select the several items to display on the Y list box.
- The integration routine LSODA wasn't freeing all the memory that it requested which resulted in a crash of the simulator after about 89 iterations. This is now fixed.

Oxnard, 18 January 1993

This is the release 2.04b of **Gepasi** version 2.0

New in this version:

- **Gepasi's** source code is now available.
- A minor bug in the data filename generation was corrected

Canterbury, 6 January 1993

This is release 2.04a of **Gepasi** version 2.0

New in this version:

- **Plot...** menu entry. One can now, from **Gepasi**, quickly generate plots of the results of simulations with the first class **GNUPLOT** program. These plots can be in 2 or 3 dimensions.
- the **Simulation** and **Topology** windows now have a fixed size.
- one bug associated with user-defined rate equations caused some models to be wrongly processed by the front-end (both **Simulation** and **Topology**). This has been fixed.
- other minor bugs have been fixed.

Aberystwyth, 25 November 1992

This is release 2.03a of **Gepasi** version 2.0

New in this version:

- one missing page in the help file was added and some spelling mistakes corrected
- one bug was affecting steady-state solutions in some cases where there were mass conservation cycles, this has been corrected.

BUG:

- unfortunately **Gepasi** is failing to run properly under MS-Windows 3.0. The cause has not yet been identified and so claims of compatibility with this version of Windows have been withdrawn (suspended...)

Aberystwyth, 18 November 1992

This is the third release of **Gepasi** version 2.0 (2.02c)

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.

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.

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 conventional 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 choosing 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 metabolites. 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 initial concentrations. 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 than the simulations to run! A much better way is to tell *Gepasi* to do these changes automatically: [Scanning parameter space](#).

Most of the times you will want some of the parameters to be dependent on others. One frequent requirement is that of changing one kinetic constant without changing the equilibrium constant. This means that for each value of the independent (changing) parameter the dependent must 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 mass action 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* is capable of producing plots of results by sending the appropriate commands to the program **GNU PLOT**. For this to work, you obviously have to obtain **GNU PLOT** (the MS-Windows version will be the most convenient, but the MS-DOS version will also work), install it in one directory and let *Gepasi* know where it is. This is done by selecting the **Plot...** menu entry which will setup the link between both programs (and after this work as expected: to plot graphs). **GNU PLOT** is available free of charge from several archives in the Internet or other networks, at least in all those where you can find *Gepasi* (see [Distribution](#)). The floppy disk distribution versions of *Gepasi* also include a version of **GNU PLOT**. See [Plot](#) for more details.

Other plotting programs may be needed for specific requirements and *Gepasi* allows for the generation of data in formats that should enable the use of most popular programs for this purpose.

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 concentrations and all of the fluxes of the model.

Generally, you will be interested in outputting only selected parameters and variables of the model. This is achieved by the definition of a data file (normally with the extension **.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 programs and hints on how to achieve the best results with them.

The simulation engine

The simulation engine is the module 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 |   |   |   |   |   |   |   |   |   |   |
      |   |   |   |   |   |   |   |   |   |   |
      method   step size   time of   lasttime   number function jacobian
                           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 transferred from a different machine with the front-end (for example a MS-Windows computer).

Presently the simulation engine has been compiled under UNIX (ULTRIX 4.2 and 4.3)

Limits of *Gepasi*

This version of *Gepasi* has the following limits:

- only handles models with 45 reactions.
- only handles models with 45 metabolites (internal and external).
- metabolite names are truncated to 20 characters.
- step (reaction) names are truncated to 20 characters.
- multi-compartment pathways are presently not supported.

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](#)

[Decroly-Goldbeter](#)

[Performance benchmarks](#)

Example files - Brusselator

This is the famous oscillating 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 oscillations.

This `brussel.sim` 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 abscissa 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 initial 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. Applic. 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 K_i and h 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 GNU PLOT (see Compatible programs) to produce a 3D graph of $[B]_{ss}$ as a function of K_i and h .

Example files - Decroly-Goldbeter

This is a simplified model of glycolysis studied by Decroly and Goldbeter (Decroly, O. & Goldbeter, A. (1982) *Proc. Natl. Acad. Sci. USA* **79**, 6917-6921). What makes this model interesting is that it displays a variety of modes of dynamic behaviour.

This example is constituted by 4 files:

`decroly.top` is the topology file;
`decroly1.sim` converges to a steady state;
`decroly2.sim` displays complex oscillations;
`decroly3.sim` displays chaotic behaviour (in the amplitude of the oscillations).

To analyse these examples, load and run the various `.sim` files and plot [B] and [C] versus time to get the trajectories or [A] versus [B] versus [C] in 3D to get the phase space orbit.

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 external substrate A into the external product Z, with all the other letters of the alphabet as internal metabolites, 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. The purpose of 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 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 computers

computer	co-pro	test1	test2	test4	test5
Cubit 486/50	intern.	16	2	47	2
Dan 486/33	intern.	22	2	49	2
Tandon SV1486	intern.	29	2	67	3
Viglen 386/33	yes	54	5	122	6
Viglen II(286)	no	-	173	-	173

results are in seconds of execution time; lower values indicate better performances

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 spreadsheet. 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

Free software package. It can be found on several Internet archives and public BBSs and included with *Gepasi* on the disk version.

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 many commercial packages can't do); 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!...)

From release 2.04a onwards, the MS-Windows port of GNUPLOT is being distributed with *Gepasi* (only in floppy disks, not in archive versions; the archives contain GNUPLOT as a separate package).

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 its kinetic constants that are actually needed.

The kinetic types internally defined in *Gepasi* are:

constant rate

mass action

Michaelis-Menten

Reversible Michaelis-Menten

Specific inhibition

Catalytic inhibition

Mixed inhibition

Specific activation

Catalytic activation

Mixed activation

Hill

Ordered Uni Bi (two products)

Ordered Uni Bi (one product)

Constant rate kinetics

This kinetic type is that of zero-order reactions, i.e. those that proceed at a constant rate without any influence from the concentration of its substrates or products.

Reactions of this type have only one parameter (rate constant) that has dimensions of flux. This type applies only to irreversible reactions.

$$v = k$$

k - constant rate of reaction

Mass action kinetics

This kinetic type 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

$$v = k_1 \cdot s_i - k_2 \cdot p_i$$

k_1 - rate constant for the forward direction

k_2 - rate constant for the reverse direction

irreversible mechanism

$$v = k_1 \cdot s_i$$

k_1 - rate constant for the forward direction

where s_i is the product of all substrate concentrations and p_i is the product of all product concentrations.

Michaelis-Menten kinetics

This is the classical enzyme kinetic mechanism. There is one substrate which is converted to one product, the process is irreversible and has two kinetic constants:

$$v = \frac{V \cdot S}{K_m + S}$$

V - limiting rate of reaction

K_m - 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:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{1 + S / K_{ms} + P / K_{mp}}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ (Michaelis constant)

Reversible Michaelis-Menten kinetics with specific inhibition

This is the Michaelis-Menten mechanism with a specific (competitive) inhibitor that affects $K_{ms}(\text{app})$, and $K_{mp}(\text{app})$. It has one substrate and one product, it is reversible and has five kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{1 + S / K_{ms} + P / K_{mp} + I / K_i}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and $I=0$ (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and $I=0$ (Michaelis constant)
- K_i - 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 $V_f(\text{app})$ and $V_r(\text{app})$. It has one substrate and one product, it is reversible and has five kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{(1 + S / K_{ms} + P / K_{mp}) (1 + I / K_i)}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and $I=0$ (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and $I=0$ (Michaelis constant)
- K_i - 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 $V_f(\text{app})$, $V_r(\text{app})$, $K_{ms}(\text{app})$, and $K_{mp}(\text{app})$. It has one substrate and one product, it is reversible and has six kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{(1 + I / K_i + S / K_{ms} + P / K_{mp}) (1 + I / K' i)}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and $I=0$ (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and $I=0$ (Michaelis constant)
- K_i - 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 $K_{ms}(\text{app})$, and $K_{mp}(\text{app})$. This type has one substrate and one product, it is reversible and has five kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{1 + S / K_{ms} + P / K_{mp} + K_a / A}$$

V_f - limiting rate of forward reaction

K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and A is saturating (Michaelis constant)

V_r - limiting rate of reverse reaction

K_{mp} - product concentration such that $v = V/2$, when $S=0$ and A is saturating (Michaelis constant)

K_a - 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 $V_f(\text{app})$ and $V_r(\text{app})$. It has one substrate and one product, it is reversible and has five kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{(1 + S / K_{ms} + P / K_{mp}) (1 + K_a / A)}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and A is saturating (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and A is saturating (Michaelis constant)
- K_a - 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 $V_f(\text{app})$, $V_r(\text{app})$, $K_{ms}(\text{app})$, and $K_{mp}(\text{app})$. It has one substrate and one product, it is reversible and has six kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{(1 + K_a / A + S / K_{ms} + P / K_{mp}) (1 + K'_a / A)}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and A is saturating (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and A is saturating (Michaelis constant)
- K_a - dissociation constant of the enzyme-activator complex
- K'_a - activator dissociation constant of the enzyme-substrate-activator complex

Allosteric inhibition

This is an empirical equation with an inhibitor that affects $K_{ms}(\text{app})$, and $K_{mp}(\text{app})$. It has one substrate and one product, it is reversible and has six kinetic constants:

$$v = \frac{V_f \cdot S / K_{ms} - V_r \cdot P / K_{mp}}{(1 + S / K_{ms} + P / K_{mp} + I^h / K_i)}$$

- V_f - limiting rate of forward reaction
- K_{ms} - substrate concentration such that $v = V/2$, when $P=0$ and $I=0$ (Michaelis constant)
- V_r - limiting rate of reverse reaction
- K_{mp} - product concentration such that $v = V/2$, when $S=0$ and $I=0$ (Michaelis constant)
- K_i - inhibition constant
- h - Hill coefficient for the inhibition

Note that when $h=1$ this rate equation becomes the same as that for Specific inhibition.

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:

$$v = \frac{V \cdot S^h}{K^h + S^h}$$

- V - limiting rate of reaction
- K - K is the concentration of S for which $v=V/2$
- h - degree of cooperativity of S (Hill coefficient)

Ordered Uni Bi (two different products)

This kinetic type has one substrate and two products. The two molecules of products are released from the enzyme complexes in an ordered fashion (opposed to random). This mechanism is reversible and has six kinetic constants:

$$v = \frac{V_f \cdot A / K_A - V_r \cdot P \cdot Q / K_{PQ}}{(1 + A/K_A + P \cdot Q / K_{PQ} + P/K_P + Q/K_Q + A \cdot P / (K_A \cdot K_P) \cdot (1 + V_f \cdot (1 - K_P \cdot K_Q / K_{PQ}) / V_r))}$$

- V_f - limiting rate of forward reaction
- V_r - limiting rate of reverse reaction
- K_A - constant associated with the terms in A
- K_P - constant associated with the terms in P
- K_Q - constant associated with the terms in Q
- K_{PQ} - constant associated with the terms in P*Q

Ordered Uni Bi (one product)

This kinetic type has one substrate and one product, which is produced in a stoichiometry of two. The release of the second molecule of product has different time constants than the release of the first, such that the mechanism can be considered 'ordered' ('random' otherwise). Due to this there are two constants related with terms in P. This mechanism is reversible and has six kinetic constants:

$$v = \frac{V_f \cdot A / K_A - V_r \cdot P^2 / K_{PP}}{(1 + A/K_A + P^2/K_{PP} + P/K_P + P/K_{P'} + A \cdot P / (K_A \cdot K_P)) \cdot (1 + V_f \cdot (1 - K_P \cdot K_{P'} / K_{PP}) / V_r)}$$

- V_f - limiting rate of forward reaction
- V_r - limiting rate of reverse reaction
- K_A - constant associated with the terms in A
- K_P - constant associated with the terms in P
- $K_{P'}$ - second constant associated with the terms in P
- K_{PP} - constant associated with the terms in P^2

User-defined kinetics

These kinetic types are defined by users and its rate equations can be any mathematical expressions, with an arbitrary number of kinetic constants. The complexity of these equations is limited by memory constraints.

Since user-defined rate equations are rather slower to compute than the predefined ones, users should try to use as few of the former as possible in a single simulation. To achieve this users might have to redefine some reactions in their elementary steps (*i.e.* with an explicit mechanism which is only composed of mass action steps).

Depending on the size of each user-defined equation and their number (in a single simulation) there might not be enough memory for the simulation engine to work. In such cases try redefining the simulation by:

- i)* using fewer user-defined kinetic types,
- ii)* simplifying the mathematical expressions of the rate equations and using as less symbols as possible, and
- iii)* select less items on the output list as these also consume memory

The user-defined kinetic types are stored in a database (`user-def.kin`) that can be manipulated 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 internal metabolite concentrations are described as a set of first order differential equations (ODE) of the type:

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

where s_i is an internal metabolite concentration, part of \mathbf{S} , the vector of all internal metabolite concentrations; \mathbf{K} is the vector of system parameters (kinetic parameters, external metabolite concentrations, etc.) and f is an appropriate rate equation. 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 LSODA. This routine uses a sophisticated strategy to find out if the system of equations is becoming stiff 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 steady state solution of a certain model *Gepasi* uses a combined strategy. The steady state is, by definition :

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

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 damped Newton's method using the set of concentrations of the last time value of the integration as the guess. If this estimate is not good enough for the damped Newton method to converge then a solution is attempted by means of solving the differential equations for increasing values of time. If a steady state is not reached at 10^{10} units of time *Gepasi* will leave the problem unsolved and notify the user. (Note that if the unit of time is second, 10^{10} 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).

Control coefficients are obtained by the method described in Reder, C. (1988) Metabolic control theory: a structural approach. *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 stoichiometric 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 stoichiometry matrix and the **total amount of each conserved moiety** is computed as the sum of the initial concentrations of its components. When there are mass conservation relations, the number of ODEs is reduced.

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:

$$\mathbf{h} = \mathbf{f}(\mathbf{old_x}) / \mathbf{f}'(\mathbf{old_x})$$
$$\mathbf{new_x} = \mathbf{old_x} - \mathbf{h}$$

in the *damped* method if the $\mathbf{new_x}$ has a larger residual error than $\mathbf{old_x}$ one looks at $\mathbf{old_x} - \mathbf{h}/2^i$ for $0 \leq i \leq 32$ and accepts the first such value that has a smaller residual error than $\mathbf{old_x}$. 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.

See [Syntax of chemical reactions](#) for an explanation of how to write the reactions.

Bear in mind that the order in which the substrates and products are entered is significant: the first substrate entered will be assigned to the first substrate in the rate equation and so on; the same for the products.

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.

Syntax of chemical reactions

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

- The substrates 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 stoichiometry bigger than 1 the stoichiometry 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 no substrates 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 kinetic mechanisms 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.

The kinetic types listed are those that have the same reversibility status and the same or smaller molecularities (number of substrates or products) than the reaction selected. For example, the Michaelis-Menten type appears for any irreversible reaction with one substrate and one product or more; the Ordered Uni Bi type does not appear for reactions with less than two products. One must be careful in the selection of kinetic types. Whenever a kinetic type with less substrates or products is selected, the reaction will be of order zero in some of the substrates or products. The ones that are left off are the last (the order in which they were written in the reaction editor is very important and determines which reactants are which in the kinetic equation).

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 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 whether 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 as 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

Here you can define and alter the title of the model.

Press **OK** to exit the dialog box keeping the changes made.

Press **H**elp to call this help page.

Press **C**ancel to exit the dialog box discarding all changes.

User-defined kinetic type editor

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 only one instance of the **Topology** utility running. 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 **C**hange button).

Close - finishes the edition of kinetic types.

Help - calls this help page.

Change - 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. See Key to items for the meaning of the symbols used.

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 your 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 symbols (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 different 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, **C**onstant, **S**ubstrate, **P**roduct or **M**odifier.

Enter a **T**itle under in the edit-box and check the **R**eversible reaction box if this reaction is thermodynamically reversible.

Press **O**K to exit the dialog box keeping the changes made.

Press **H**elp to call this help page.

Press **C**ancel to exit the dialog box and discard all

Links dialog box

This is where you can add and change links 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 **H**elp to call this help page.

Press **C**ancel to exit the dialog box and discard all changes.

Press **E**dit to change the selected link. This will bring up the Edit links dialog box.

Press **A**dd new to add a new link. This will bring up the Edit links dialog box.

Press **D**elete to delete the selected link.

Edit link dialog box

This is where links are edited.

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 simulation 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 are defined. 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.

WARNING: in some kinetic types, defining constants to be zero may generate floating-point errors (division by zero).

The list on the left 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 differential equations. 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 steady-state solution, 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 Report options).

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

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.

Key to items (scan and output)

The following symbols are used to specify both scan and output items:

$[x]_i$	- initial concentration of metabolite x.
$[x]_f$	- final concentration of metabolite x (either at the end time or steady state).
$J(r_1)$	- flux through step r1 at the end time.
$e(r_1, [x])$	- elasticity of step r1 in respect to metabolite x.
$C([x], r_1)$	- concentration control coefficient of metabolite x in respect to step r1.
$C(J(r_1), r_2)$	- flux control coefficient of step r1 in respect to step r2.
time	- time at which the simulation stopped.
integration steps	- number of steps taken by the <u>ODE solver</u> to reach the end time.
function evaluations	- number of times the <u>ODE solver</u> had to evaluate the functions (the ODEs).
jacobian evaluations	- number of times the <u>ODE solver</u> had to evaluate the jacobian matrix.
last step size	- size (in time units) of the last step taken by the <u>ODE solver</u> .
relative tolerance	- relative tolerance used by the <u>ODE solver</u> .
absolute tolerance	- absolute tolerance used by the <u>ODE solver</u> .
flux resolution	- minimum value of flux to be considered significantly different from zero.

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 items that can be written in the results data file (see key to items). 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 column 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 (required by 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.

Plot

In this dialog box you select variables and/or parameters to be plotted with GNUPLOT. Data from dynamic or steady states are available if selected for the simulation.

To have steady state data available you must have selected output to a columnar data file, with the required items selected and without titles.

Select **Dynamic** or **Steady state** radio buttons and the items available for plotting will be listed (see key to items for their meaning).

Select **2D** or **3D** according to the type of plot you want. Check the box **Lines** if you want the points connected by straight lines, otherwise the data will be plotted as dots only. For 3D plots you can select **Hidden** line removal and **Contour** plots in the XY plane. However, if some data point is missing (because there was a floating-point error in its calculation) these features will not work.

In the lists **X**, **Y** and **Z** you select the items to be plotted on each axis. Above the lists you can check boxes to indicate logarithmic scales on each axis.

A few seconds after pressing **OK** a window with the plot will appear on the screen, together with a message box saying 'press key to exit'. Drag the message box to a region of the screen where it will not interfere with the actual plot. You can print the plot or copy it into the clipboard.; for this select the system menu of the plot window and go to the Options sub-menu. There are also other options that can be manipulated. Press one of the buttons of the message box to close the plot. Note that you can not do another plot before closing the current one.

Press **OK** to exit the dialog box and activate GNUPLOT to generate the plot.

Press **Help** to call this help page.

Press **Cancel** to exit the dialog box, discarding all changes and not producing the plot.

See also Installing GNUPLOT to learn how to install GNUPLOT and link it to *Gepasi*.

Installing GNUPLOT to work with *Gepasi*

If you have obtained *Gepasi* without a version of GNUPLOT, you may find very convenient to have it installed in your system so that you can quickly produce plots of your simulations (even if you are later going to use another package). GNUPLOT is free and is also very good, so why not give it a try?

The best version of GNUPLOT to work with *Gepasi* is the MS-Windows port, version 3.3 (the most recent to this date). You can find this in Internet archives like SIMTEL-20 (see [Distribution](#)) or public access BBSs under the name of `wgplt339.zip`, `wgnuplot.zip` or similar. Previous versions of GNUPLOT for MS-Windows had a limit of 64 k on data but versions later than 3.38 can use all the available memory (including that of a swap file) which makes it very useful as it can display plots with large data sets. This is normally required for dynamics with very sharp peaks and long periods (like in quasi-periodic and chaotic systems, but not only in these). If you do not have access to the MS-Windows version, you can still link the MS-DOS GNUPLOT to the front end (but this is also unable to use all the available memory). If you use a version of the simulation engine in unix or other operating system you will be glad to know that there are ports of GNUPLOT for a large number of these. Send email queries to `info-gnuplot@dartmouth.edu` send GNUPLOT bug reports to `bug-gnuplot@dartmouth.edu`.

After downloading GNUPLOT, unarchive it to a directory of your choice (using unzip or other appropriate uncompressing program). Then go into **Simulation** select **P**lot... and answer the message-box with **OK**. You will then have to point the GNUPLOT executable file (in the windows version `wgnuplot.exe`, `gnuplot.exe` for the MS-DOS version) in its path. Press **OK** and you have done it all! Subsequent selections of **P**lot... will then allow you to generate plots.

Distribution of *Gepasi*

Gepasi is available from the author by post (send a diskette with at least 1.2 Mb for the most recent version):

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 available from the Internet. It is currently stored in the archive SIMTEL20 at `wsmr-simtel20.army.mil` in the directory `<MSDOS.BIOLOGY>`. Since SIMTEL20 is mirrored all over the world, you will probably be able to find *Gepasi* very near you. Particularly, the program is known to be stored in the following FTP servers (choose the closest to your location):

ADDRESS	DIRECTORY
Europe	
<code>src.doc.ic.ac.uk</code>	<code>/packages/simtel20/biology</code>
<code>nic.switch.ch</code>	<code>/software/msdos/simtel20/biology</code>
<code>ftp.urz.uni-heidelberg.de</code>	<code>/pub/simtel/biology</code>
<code>ftp.hrz.tu-chemnitz.de</code>	<code>/fs12/pub/simtel/msdos/biology</code>
<code>swdsrv.edvz.univie.ac.at</code>	<code>/pc/dos/biology</code>
<code>nic.funet.fi</code>	<code>/pub/msdos/SIMTEL20-mirror/biology</code>
<code>ftp.vse.cs</code>	<code>/pub/msdos/simtel/biology</code>
North America	
<code>wsmr.simtel20.army.mil</code>	<code>pdl:<MSDOS.BIOLOGY></code>
<code>rigel.acs.oakland.edu</code>	<code>/pub/msdos/biology</code>
<code>ftp.wustl.edu</code>	<code>/systems/ibmpc/msdos/biology</code>
<code>ftp.uu.net</code>	<code>/systems/ibmpc/msdos/simtel20/biology</code>
<code>knot.queensu.ca</code>	<code>/warchive/systems/ibmpc/msdos/biology</code>
<code>cs.ubc.ca</code>	<code>/mirror4/msdos/biology</code>
Australia	
<code>plaza.aarnet.edu.au</code>	<code>/micros/pc/oak/biology</code>
Africa	
<code>itu2.sun.ac.za</code>	<code>/simtel/biology</code>
Far-east Asia	
<code>nctuccca.edu.tw</code>	<code>/PC-MsDos/SIMTEL-msdos/biology</code>
<code>isfs.kuis.kyoto-u.ac.jp</code>	(unknown directory)

For those that do not have easy access to ftp, *Gepasi* can be retrieved by electronic mail from `NETSERV@EMBL-HEIDELBERG.DE`. Send a message to this address with `HELP` on the first line of text to receive instructions on how to use this service, send a message with `GETDOS_SOFTWARE:GEPASI.UAA` on the first line of text to receive *Gepasi* (you will have to decode the ASCII messages into binary and transfer it to a PC).

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`.

The `btk-mca` mailing list is now also mirrored in the Usenet newsgroup `bionet.metabolic-reg`.

Tailoring *Gepasi* to your needs

Gepasi can be altered by you! The source code is available from the author (see [Distribution](#)). 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 [Distribution](#)) 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 GPL](#) 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).

The source code can be found in the same location where you got this package in a file named gsrcXXXX.zip where the XXXX is a version tag (for example gsrc205b.zip). This is a compressed archive in the popular ZIP file format.

Acknowledgements

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

Jeremy Bennett, Athel Cornish-Bowden, Jon Dean, George Dervakos, The Free Software Foundation, Alan C. Hindmarsh, Jannie Hofmeyr, Horst Holstein, J.N.I.C.T., Douglas Kell, Pedro Moniz Barreto, Linda Petzold, Herbert Sauro, Hon Tam, The University of Wales Aberystwyth, Karel van Dam, Ana Vaz Gomes, Hans Westerhoff and three anonymous referees from the journal *Computer Applications in the Biosciences*.

Thanks!

Suggested reading on enzyme kinetics

Cleland, W. W. (1963). The kinetics of enzyme-catalyzed reactions with two or more substrates or products. I. Nomenclature and rate equations. *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 applications to biochemical kinetics. *Ann. Rev Biochem.* **39**, 473-498.

Garfinkel, D. (1981). Computer modeling of metabolic pathways. *Trends Biochem. Sci.* **6**(3), 69-71.

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.

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

Kell, D. B., van Dam, K. & Westerhoff, H. V. (1989). Control analysis of microbial growth and productivity. *Symp. Soc. Gen Microbiol.* **44**, 61-93.

Heinrich, R. & 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 approach. *J. Theor. Biol.* **135**, 175-201.

Schuster, S. & Heinrich, R. (1992). The definitions of metabolic control analysis revisited. *BioSystems* **27**, 1-15.

Suggested reading on numerical analysis

Conte, S. D. & de Boor, C. (1980). *Elementary 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.

Gear, C. W. (1980). Numerical software: science or alchemy? *Adv. Comput.* **19**, 229-248

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. (1992). *Numerical recipes in C. The art of scientific computing*. second edition, Cambridge University Press, Cambridge, U.K.

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

A **scan item** is a parameter of the model that can be scanned, that is varied within a range. This includes the formal parameters of the model (kinetic constants and initial concentrations) and parameters of the numerical methods.

A **output item** is either a parameter or a variable of the model that can be written in the output data file. This includes the formal parameters of the model (kinetic constants and initial concentrations), parameters of the numerical methods, the variables (concentrations and fluxes) and the coefficients of metabolic control analysis.

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). Diffusion is a gradual process and cannot be considered a step.

A **pool** is an ensemble 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 which is not permeable (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 upstream (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 step is the rate at which its product is being formed. In a pathway at steady-state 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 internal or external 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 concentration). 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 mathematical 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.

Copyleft

Gepasi is covered by the GNU General Public License. Below is the full text of this license

GNU GENERAL PUBLIC LICENSE Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes

make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!