SPREADS

Brownian dynamics simulator for diblock copolymers and homopolymers
version 0.1.3
15 September 2020

Takashi Uneyama

Table of Contents

1 Introduction

Block copolymer melts or polymer blends form various phase separation structures [Statistical Physics of Polymers: An Introduction]. Although static properties of phase separation structures are well understood by such as the Flory-Huggins theory or the self consistent field theory [Statistical Physics of Polymers: An Introduction], dynamic properties of these phase separated structures [The Structure and Rheology of Complex Fluids] are not so well understood.

Recently, a highly coarse-grained theory which models a polymer chain as an effective colloid particle is proposed [Louis-Bolhuis-Hansen-Meijer-2000]. The model is generalized to diblock copolymers [Addison-Hansen-Krakoviack-Louis-2005, Pierleoni-Addison-Hansen-Krakoviack-2006,Eurich-Karatchentsev-Baschnagel-Dieterich-Maass-2007] and thus it is possible to perform highly coarse-grained simulations for diblock copolymers or homopolymers. For example, by using the soft colloid model, micellar formation dynamics in diblock copolymer solutions [Cass-Heyes-English-2007,Cass-Heyes-Blanchard-English-2007] or rheology of lamellar structures [Uneyama-2009] can be simulated with small computational costs.

Dynamics simulations can be used to investigate the dynamic properties such as structure formation dynamics or rheology. `spreads` is a Brwonian dynamics simulator for diblock copolymers and homopolymers based on the soft particle model. It can handle blends of symmetric diblock copolymers and homopolymers. Because the soft particle model is a highly coarse-grained particle model, `spreads` enables numerically efficient simulations. Several interesting dynamics such as microphase separation under shear flow or viscoelastic properties can be studied with relatively small computational costs.

2 Installation of spreads

2.1 To Download the Latest Version of spreads

The latest version of spreads is available at the following URL. Access the web page and download the latest version via HTTP (FTP is not supported).

http://polymer-physics.jp/uneyama/spreads.html

2.2 Build and Install from the source

You can build and install spreads if the binary package of your system is not available, or if you want to customize the spreads. The source package of spreads is using GNU Automake and GNU Autoconf, therefore you can build and install spreads just like usual free software. Note that spreads requires zlib (http://www.zlib.net/) and Lua (http://www.lua.org/). You have to install them before building spreads.

The source package is distributed as the gzipped tar archive file, thus first extract it. To extract the archive, do

```
$ zcat spreads-0.1.3.tar.gz | tar xvf -
```

or if you are using GNU tar, do

```
$ tar zxvf spreads-0.1.3.tar.gz
```

Then the source directory will extracted. Move to the directory spreads-0.1.3.

```
$ cd spreads-0.1.3
```

To build spreads, do configure-make-make install just like other free software.

```
$ ./configure
$ make
$ su -
# make install
```

Now spreads will be installed under /usr/local of your system. If you have an error messages and the compilation is aborted, some commands or libraries may be missing. Install the required packages and retry.

If you want to customize or tune spreads,

```
$ ./configure --help
```

will help you.

2.3 Build and Install as the RPM package (for Linux)

The RPM package for your Linux system can be built from the source RPM (SRPM) package.Make sure that the headers and libraries and headers of zlib and lua is already installed to your system. If they are not installed, first you have to install them (zlib, zlib-devel,lua, and lua-devel). Of course you need standard development tools such as C compiler (gcc) or Make (make).

If you are using old system (rpm compatible with RedHat 7.3 or older), use the rpm command to build it.

```
# rpm --rebuild spreads-0.1.3-1.src.rpm
```

If you are using new system (rpm compatible with RedHat 8.0 or newer), use rpmbuild instead of rpm.

```
# rpmbuild --rebuild spreads-0.1.3-1.src.rpm
```

Now the binary RPM package for your system is stored in the directory which is shown in the output message of rpm or rpmbuild. Install it by rpm, for example, if you are using RedHat Linux or Fedora Core on a PC (or i386 compatible computer), like the following.

```
# rpm -Uvh /usr/src/redhat/RPMS/i386/spreads-0.1.3-1.i386.rpm
```

2.4 Compilation with Intel C++ Compiler (icc)

You may want to compile spreads Intel C++ Compiler (icc). icc is mostly compatible GNU C Compiler (gcc) and thus you can compiler spreads with icc easily. But the optimization flag -ipo will cause troubles when compiling spreads. Also note that the flag -ipo is automatically enabled if you set the optimization flag -fast or if you using icc version 9.0 or later.

There are two way to avoid troubles with the flag -ipo. One way is to add the flag -ipo_obj. This means, to run configure like

```
$ ./configure CC=icc CFLAGS='-O3 -ipo -ipo_obj'
```

(Here note that, this method can be used only for icc version 8. If you are using icc version 9, you should use the following method.) Another way is to use xiar,xild instead of ar,ld. In this case, the additional flag -ipo_obj is not needed.

```
$ ./configure CC=icc CFLAGS='-O3 -ipo'
$ make AR=xiar LD=xild
```

See the manual of Intel C++ Compiler for more information.

3 Invoking `spreads`

The format for running the `spreads` program is:

```
$ spreads option ... input
```

input is the input file for `spreads`. If no input file is specified, `spreads` will read the default input file `spreadsin.lua`.

`spreads` supports the following options:

`--input=`*`input`*
`-i` *`input`* Read the parameters for simulation from the input file *input*. If no input file is specified, spreads will read the input file named `spreadsin.lua`.

`--position=`*`position`*
`-p` *`position`*
Read the initial positions of the particles from the file *position*. The position input file *position* must be the gzipped plain text. You can create one easily by using `gzip`. By default, spreads set the positions randomly.

`--help`
`-h` Show summary of options.

`--version`
`-v` Show version of program.

4 Tutorial

4.1 Simple Example

Here are a simple examples how to use spreads (the input file and some scripts for visualization can be found in the directory examples/). But first you have to install the spreads to your system. If spreads is not installed to your system, see the 'Install spreads' section. We starts from the microphase separation dynamics of a diblock copolymer melt. The input file is distributed with the source code or binary of spreads. To run this example, move to the directory examples/ab_melt_3d/ and just type spreads

```
$ spreads
```

spreads will output some information about the simulation, and starts the simulation. The simulation will end in several hours. You can find output files.

4.2 Visualize Output Data

The output file of spreads is gzipped plain text and the OpenDX (http://www.opendx.org/) data format. If you have OpenDX, you can visualize it directly. The gzipped plain text is more portable and can be handled by most of the plotting / visualizing applications. Most of applications cannot handle the gzipped text directly, we have to decompress it. There are two way to do it. The first way is to use gunzip and then plot the decompressed data.

```
$ gunzip position.dat.gz
```

The second way is to use zcat and pipe.

```
$ zcat position.dat.gz | program
```

where program reads the input data from stdin. The result is just the same as the first way.

4.3 Changing Input File

The input file is the Lua script which sets the parameters needed for the DF simulation. The following is the input file used in the previous section.

```
condition =
{
    save_position_sequential = true,
    save_stress_sequential = true,
    save_energy_sequential = false,
    save_geometry_sequential = false,
    save_dx_sequential = true,
    seed = 11583192,

    apply_shear = false,

    iteration_max = 5000,
    interval = 50,
    dt = 0.02,
    omega = 0.0,
    kappa = 0.0
}

file =
{
    position_output = "position.dat.gz",
    stress_output = "stress.dat",
    energy_output = "energy.dat",
    geometry_output = "geometry.dat",
    dx_output = "spreadsout.dx",

    position_template = "position.%d.dat.gz",
    dx_template = "spreadsout.%d.dx"
}

geometry =
{
    dimension = 3,

    l = {16, 16, 16}
}

blend =
{
    polymer = {"AB_diblock"},
    number = {2048}
}

monomer =
{
    name = {"A", "B"},
    epsilon = {{30, 40},
               {0,  30}}
}

AB_diblock =
{
    monomer = {"A", "B"}
}
```

There are many parameters required by `spreads`. The detail of the input file will be expressed in the section ‘Input File Format’.

Here we modify this input file simply. The first example is to change the size and dimension(s) of the simulations box. This can be done by changing the `geometry` in the input file. Change the `geometry` in the input file as follows.

```
geometry =
{
    dimension = 2,

    l = {64, 64}
}
```

`dimension` means the dimension(s) of the system. `l` means the size of the simulation box. Thus the parameters shown above mean the 2 dimensional with the size 16 times 16.

The second example is to change the polymers used in the simulation. This needs more complicated changes. The change will be as follows.

```
blend =
{
    polymer = {"A_homo", "B_homo"},
    number = {2048, 2048}
}

A_homo =
{
    monomer = {"A"}
}

B_homo =
{
    monomer = {"B"}
}
```

`blend` is changed to simulate the blend of A homopolymer / B homopolymer. The polymer species which the blend is consists on are listed in `polymer`. The numbers of each polymer species are specified by `number`. The A homopolymer `A_homo` and the B homopolymer `B_homo` is defined as well (the `AB_diblock` is no longer needed and can be deleted because now it is not used). `monomer` is the monomer species.

4.4 Notes on Input File

4.4.1 Boolean Variables

There are many boolean variables (of which value is `true` or `false`) in the input file for `spreads`. However it may seem verbous to write many boolean values (especially for large adjacency matrices). In such situations one can use `1` and `0` instead of `true` and `false`. `spreads` automatically converts `1` and `0` into boolean values, `true` and `false`, for the boolean variables. (Strictly speaking, number value `0` corresponds to `false` and non-zero numbers, including `1`, correspond to `true`. This is just the same as the standard C mannar.)

4.4.2 Symmetric Matrices

The effective interaction parameter matrix, `monomer.epsilon` are symmteric. Thus we don't need to set all the elements in these matrices. In the input file for `spreads`, the interaction parameter matrice is required to set their upper triangular part. `spreads` automatically fill the lower triangular part by copying the values of upper triangular elements. (See examples in previous sections.)

5 Reporting Bugs

Currently, the error handling routines in `spreads` is not complete and therefore `spreads` may suddenly stops if some input error or calculation error is caused.

If you find a bug in `spreads`, please send electronic mail to uneyama@mp.pse.nagoya-u.ac.jp. Include the version number, which you can find by running `spreads --version`. Also include in your message the output that the program produced and the output you expected.

If you have other questions, comments or suggestions about `spreads`, contact the author via electronic mail to uneyama@mp.pse.nagoya-u.ac.jp. The author will try to help you out, although he may not have time to fix your problems.

6 Input File Format

In this section, the input file format for `spreads` is expressed. The input file is the Lua script which sets the parameters. The parameters are set as the table variables.

6.1 Simulation Condition

The simulation condition is set as the table `condition`. The following elements are required.

`condition.save_position_sequential`

(*boolean* or *number*)

Whether to save the particle positions or not. If `condition.save_position_sequential` is set to `true`, `spreads` saves the positions every `condition.interval` steps. The output file name is generated from `file.positions_template`. If it is set to `false`, `spreads` saves positions to the output file named `file.positions_output` every `condition.interval` steps (in other words, the output file is overwritten). The behavior is the same for following items.

`condition.save_stress_sequential`

(*boolean* or *number*)

Whether to save the stress tensor sequentially or not. See `condition.save_position_sequential` for detail.

`condition.save_energy_sequential`

(*boolean* or *number*)

Whether to save the energy sequentially or not. The energy output routime is not implemented yet and currently this item has no meaning. See `condition.save_position_sequential` for detail.

`condition.save_geometry_sequential`

(*boolean* or *number*)

Whether to save the geometry field sequentially or not. See `condition.save_position_sequential` for detail.

`condition.save_dx_sequential`

(*boolean* or *number*)

Whether to save the DX output sequentially or not. See `condition.save_position_sequential` for detail.

`condition.seed`

(*number*)

Seed for the Mersenne twister random number generator.

`condition.apply_shear`

(*boolean* or *number*)

Whether to apply the external shear flow field or not. The shear flow condition can be set by `condition.kappa` and `condition.omega`.

`condition.iteration_max`

(*number*)

Maximum number of iterations for the simulation. `spreads` ends the simulation if the number of iterations reaches `condition.iteration_max`.

`condition.interval`

(*number*)

Interval for the data output into files.

`condition.dt`

(*number*)

The size of the time step. Too large `condition.dt` will lead unstable simulations or incorrect data.

`condition.omega`

(*number*)

The frequency of the external shear flow field. If this is set to `0`, steady shear is applied.

`condition.kappa`

(*number*)

The maximum shear strain rate.

6.2 Input / Output Files

The input / output file names are set as the `file` table.

`file.position_output`

(*string*)

Output file name for the positions of particles. This is used when `condition.save_position_sequential` is set to `false`. If the `file.position_output` is set to the null string (`""`), no output file will be created and the data will be discarded. The behaviour is the same for the other output files.

`file.stress_output`

(*string*)

Output file name for the stress tensor.

`file.energy_output`

(*string*)

Output file name for the free energy.

`file.geometry_output`

(*string*)

Output file name for the box geometry.

`file.dx_output`

(*string*)

Output file name for the DX output file.

`file.position_template`

(*string*)

Template for the output file of the positions of particles. `file.position_template` must contains `%d` once. `%d` will be replaced by the sequential numbers `1,2,3,...` If `file.positions_template` is set to the null string (`""`), no output file will be created.

`file.dx_template`

(*string*)

Template for the output file of the DX output file.

6.3 Geometry of Simulation Box

6.4 geometry

The simulation box geometry are set as the `geometry` table.

`geometry.dimension`

(*number*)

Number of dimension(s). This must be set to `2` or `3`.

`geometry.l`

(*array of numbers*)

Lengthes of the the simulation box.

6.5 Polymer Blend

The information about the polymer blend is set as the `blend` table. The polymers which is contained in the system is set as the individual tables.

`blend.polymer`

(*array of strings*)

Polymers which is contained in the blend. The polymers used here must be defined as individual tables.

`blend.number`

(*array of numbers*)

Numbers of each polymers

6.6 Monomer Species

The information about monomers is set as the `monomer` table.

`monomer.name`

(*array of strings*)

Names for each monomers.

`monomer.epsilon`

(*array of array of numbers*)

The effective interaction parameters between monomers. Only the diagonal and upper triangular parts are used.

6.7 Polymer Species

The polymers which is used in `blend.polymer` is defined as individual tables of which name is same as the element of `blend.polymer`. For example, if `blend.polymer` is set to `{AB_diblock, C_homo}` the tables `AB_diblock` and `C_homo` must be defined.

`polymer``.monomer`

(*array of strings*)

Monomers for each subchains.

7 Output File Format

In this section, the output file format for `spreads` is described.

7.1 Positions of Particles

An output file of the positions of particles is a gzipped text file. The first line shows the dimensions and number of particle species, and the second line shows the number of particles. After that the position data are stored. Each low corresponds to one particle position and each column corresponds to the x,y, and z component. The output data is like the following data (the output file itself is gzipped).

```
# 3 2
# 2048 2048

1.482591 2.522368 8.503859
14.678206 9.115228 14.011614
6.584831 1.590900 14.042807
10.338461 5.101233 2.578110
3.443631 15.092652 4.027328
8.133353 0.291188 5.869824
14.790443 7.885689 13.788454
6.382475 11.639057 8.738407
    :         :        :
```

You can deflate the output file by using `gunzip` or `zcat`.

7.2 Stress Tensor

An output file of the stress tensor is a text file. It contains 6 stress tensor components (`xx`,`yy`,`zz`, `xy`,`yz`,and `zx`) as follows.

```
-31.587779 -31.638428 -31.601666 0.011065 0.010575 -0.020566
```

7.3 Energy

The energy output routine is not implemented yet.

7.4 Box Geometry

An output file of the box geometry is a text file. The first line shows dimension(s) of the system, the second line shows the box size, and the third line shows the gap size between periodic boxes caused by the shear strain. For example, an output file for a three dimensional system becomes as follows.

```
3
16.000000 16.000000 16.000000
0.000000
```

7.5 DX Output File

The DX output file is the data format for OpenDX (visualization software). It contains the particle positions (`position0,position1,...`), the box geometry (`box`), and the gap size (`delta_l`). You can visualize it by using OpenDX. The sample OpenDX program to visualize the OpenDX format `spreads` output data will be found in the `example/` directory.

8 Utility Programs

8.1 Converter for DX Output Files

The utility programs spreads-dx2position and spreads-dx2geometry convert an OpenDX format output file generated by spreads into gzipped text files. spreads-dx2position or spreads-dx2geometry convert the positions of particles or the box geometry in the DX file into text files.

To convert particle positions in a DX output file spreadsout.dx into a gzipped text file position.dat.gz, execute spreads-dx2position as follows.

```
$ spreads-dx2position spreadsout.dx position.dat.gz
```

spreads-dx2geometry can be used in the same way.

```
$ spreads-dx2geometry spreadsout.dx geometry.dat
```

8.2 Converter for Gzipped Text Files

The utility program spreads-position2bond converts a gzipped text file of particle positions into another gzipped text file of bond vectors. This program is experimental one and thus you may encounter errors if you specify incorrect input files. Please use the only the position files of diblock copolymer melts.

To convert position file position.dat.gz into a bond vector file bond.dat.gz, execute spreads-position2bond as

```
$ spreads-dx2phi position.dat.gz geometry.dat bond.dat.gz
```

Notice that a box geometry file is needed for this program.

8.3 Utility to Calculate Shear Relaxation Modulus

The utility program spreads-stress2relaxation calculates shear relaxation modulus from sequentially saved output stress tensor files.

To calculate shear relaxation modulus from stress tensor files, execute spreads-stress2relaxation as follows.

```
$ spreads-stress2relaxation spreadsin.lua modulus.dat
```

All the parameters will be read from spreadsin.lua automatically.

9 References

[Addison-Hansen-Krakoviack-Louis-2005] C. I. Addison, J. P. Hansen, V. Krakoviack, A. A. Louis, *Mol. Phys.* **103**, 3045 (2005).

[Cass-Heyes-Blanchard-English-2007] M. J. Cass, D. M. Heyes, R. L. Blanchard, and R. J. English, *J. Phys.: Cond. Mat.* **20**, 335103 (2008).

[Cass-Heyes-English-2007] M. J. Cass, D. M. Heyes, and R. J. English, *Langmuir* **23**, 6576 (2007).

[Eurich-Karatchentsev-Baschnagel-Dieterich-Maass-2007] F. Eurich, A. Karatchentsev, J. Baschnagel, W. Dieterich, P. Maass, *J. Chem. Phys.* **127**, 134905 (2007).

[Louis-Bolhuis-Hansen-Meijer-2000] A. A. Louise, P. G. Bolhuis, J.-P. Hansen, E. J. Meijer, *Phys. Rev. Lett.* **85**, 2522 (2000).

[Pierleoni-Addison-Hansen-Krakoviack-2006] C. Pierlenoi, C. Addison, J. P. Hansen, V. Krakoviack, *Phys. Rev. Lett* **96**, 128302 (2006).

[Uneyama-2009] T. Uneyama *Nihon Reorogi Gakk. (J. Soc. Rheol. Jpn.)* **39**, 135 (2011).

[Statistical Physics of Polymers: An Introduction] T. Kawakatsu, *Statistical Physics of Polymers : An Introduction*, Springer Verlag (2004).

[The Structure and Rheology of Complex Fluids] R. G. Larson, *The Structure and Rheology of Complex Fluids*, Oxford University Press (1994).

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU 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. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), 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 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 show them these terms so they know 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.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

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

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License 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 derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

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 License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   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.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary 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 License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 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 Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

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

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing 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 for copying, distributing or modifying the Program or works based on it.

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. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. 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 this 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 this License, you may choose any version ever published by the Free Software Foundation.
10. 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.

11. 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.
12. 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 the public, 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) yyyy  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 2 of the License, 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., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, 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) year 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 is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Concept Index