



# ***BlobFlow***

*Vorticity for all*

A high order vortex method for viscous flows  
User's Guide  
Version 1.0 for BlobFlow v3.x.

Louis F. Rossi

September 2009

# Chapter 1

## Formalities

### 1.1 Copyright

Copyright © 2001, 2009 Louis F. Rossi.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being

- Formalities;
- Preliminaries;
- Building BlobFlow;
- Using BlobFlow.

A copy of the license is included in the appendix entitled "GNU Free Documentation License".

### 1.2 History

August	2000	Original draft written by L. F. Rossi for BlobFlow v1.0.
July	2001	Original draft written by L. F. Rossi for BlobFlow v2.01.
August	2001	Revised draft written by L. F. Rossi for BlobFlow v2.01.
August	2009	Revised draft written by L. F. Rossi for BlobFlow v3.x.

## 1.3 Changes since v2.x

Many algorithmic details have changed in BlobFlow since the v2.x release in 2001. Some of these changes do not affect how one sets up the runs the program, but two elements will have a major impact on users:

- Accelerated Biot-Savart evaluation. The earlier version of BlobFlow used an asymptotic Biot-Savart evaluation for elliptical Gaussian basis functions. The current version uses a new spectral interpolation algorithm that offers improvements in speed and accuracy at a cost of requiring look-up tables [2].
- Single-step integration. Earlier versions of BlobFlow used high order Adams family integrators. The current version uses a single step Runge Kutta method. My reason for making the change is that it simplified all other portions of the algorithm to use a single step method so that one would not need to treat trajectories of blobs when removing or adding them to a simulation. This change makes it much easier to add and remove features from the code.
- Splitting and merging has been completely replaced by field interpolation. Field interpolation is a global means of replacing one configuration of particles with another while inducing controllably small errors. While localized splitting and merging was an elegant approach, I have found that using a reverse heat equation solver is far more effective and accurate [1].

# Chapter 2

## Preliminaries

### 2.1 This software is free.

BlobFlow is a computer algorithm known as an elliptical corrected core spreading vortex method (ECCSVM). This software is free in many senses.

- This software is commercially free. Anyone may download and use this code at no cost.
- This software is conceptually free. There is a transparent path between input and output. Anyone may examine the source code to gain a better understanding of how it functions.
- This software is intellectually free. I want others to use my code. I place no obligations on the user of the code beyond the open source license itself.

The name “BlobFlow” was trademarked to represent my particular distribution, but the trademark is no longer maintained.

The code distributed under the name BlobFlow is open source. Anyone is free to copy, modify and use it, as long as they abide by the terms of the attached copyright. If you would like to share modifications and patches with me, I will happily include them in my source code.

### 2.2 What is a vortex method? Will it solve my problem?

A vortex method is a numerical algorithm for calculating fluid flows. BlobFlow is restricted to two-dimensional, incompressible, viscous flows though I hope to develop a

three dimensional version one day. Vortex methods are effective for flows dominated by isolated regions of vorticity. Examples include vorticity shedding from bluff bodies, simulation of coherent vortical structures, boundary layers, jets and so forth. There are situations where vortex methods would not be a good choice, so it is best to analyze the problem for choosing this algorithm.

BlobFlow is based on the elliptical core spreading vortex method. A vortex method is an algorithm that approximates the vorticity field,  $\omega$ , of a fluid flow as a linear combination of moving, localized basis functions, sometimes called *blobs*.

$$\omega(\vec{x}, t) = \sum_{i=1}^N g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i)$$

$$g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i) = \frac{\gamma_i}{4\pi\sigma_i^2} \exp \left\{ -\frac{[c_i(x - x_i) + s_i(y - y_i)]^2/a_i^2 + [-s_i(x - x_i) + c_i(y - y_i)]^2/a_i^2}{4\sigma_i^2} \right\} \quad (2.1)$$

where the parameters  $\vec{x}_i$ ,  $\sigma_i$ ,  $a_i$  and  $\theta_i$  are functions of time. The evolution equations for these quantities which can be found in [3, 4], are related to the flow velocity and the derivatives of the flow velocity. The centroid of the blob is  $\vec{x}_i$ . The width, aspect ratio and orientation of the blob is  $\sigma_i$ ,  $a_i^2$  and  $\theta_i$ , respectively. The total number of blobs,  $N$ , is often referred to as the *problem size* and can be compared to the number of mesh points in a finite difference computation.

Where finite difference schemes use mesh points as the fundamental computational element, a vortex method uses moving basis functions so that there is no grid imposed upon the problem. Schemes using elements that move with the flow are called Lagrangian schemes because they are formulated in Lagrangian coordinates that move with the fluid rather than Eulerian coordinates which are fixed in some laboratory reference frame. This is both a strength and a weakness. One of the main strengths is that the method is naturally adaptive. Vorticity moves through the domain as dictated by the governing Navier-Stokes equations. The main disadvantage is the lack of a regular grid. A regular grid has more than aesthetic advantages. Having a regular grid means that memory can be allocated in a geometrically relevant way. In a Lagrangian scheme, no such assumptions can be made.

# Chapter 3

## Building BlobFlow

### 3.1 Requirements

I tried to make BlobFlow as simple as possible to use. This is short list of what you will need.

- The source code.
- The lookup table binary files for BlobFlow.
- A C compiler.
- The `lapack` and `blas` libraries. These libraries freely available and easy to install.
- The GNU make utility or a compatible make utility. The make utility is ubiquitous.
- If you choose to use multiple cores with the MPI option, you will need the MPI libraries. Again, these are freely available to download.

### 3.2 For the impatient

BlobFlow is built using the GNU make utility. One can simply unpack the distribution and build it.

```
tar xzf blobflow.tgz
cd BlobFlow_3.x
make eflow
```

You will also need to unpack the look-up tables somewhere. When you run BlobFlow, you will need to tell BlobFlow where to find the look-up tables.

```
# Remember you put these files.  
tar xzf domain.tgz
```

### 3.3 Build options

BlobFlow is written entirely in ANSI C. While it was developed on Unix and Linux platforms, there are no machine-specific functions in the code. The code is distributed with a vanilla Makefile for the Unix *make* utility.

BlobFlow uses conditional compilation so that users can customize the executable for special situations. The most common one can be compiled into the code through the make command, but some require editing the CFLAGS arguments in the makefile.

1. Use the Message Passing Interface (MPI) to parallelize across multiple processors (`mpi=on`): Many aspects of the algorithm are parallelizable, and this feature spreads the work among many processors. To use this capability, you must install and configure a version of MPI (available at <http://www.openmpi.org> for instance) properly on the machines that you wish to use. You should also familiarize yourself running programs with MPI calls. It's easy to learn and worth the time. BlobFlow uses two algorithms under MPI. If only two processors are available, it uses a peer based scheme to evenly spread the work between the two processors. If there are more than two CPU's available, BlobFlow uses a worker-manager algorithm with a receive and dispatch scheme to balance the work amongst all available processors. This means that one process oversees activities without doing any number crunching. In principle, a two processor peer-based system will run about as fast, and perhaps a little faster, than a three processor manager-worker system. However, if one has  $N + 1$  processors, the total computation time should scale like  $1/N$  for small groups of processors.
2. Cache resorting (`CFLAGS = -DCACHERESORT`): Though CPU speeds have increased substantially in recent years, front side bus and memory speeds have not kept pace. In fact, most of the memory is orders of magnitude slower than the CPU. A small reserve, called cache, is fast memory. When instructions and data for calculations are resident in this cache, codes will run substantially faster. Computer architects and compiler authors are pretty crafty and optimizing the use of cache, but there are also ways to write code to take advantage of cache. I have attempted to do this

in BlobFlow but have found that cache awareness has little or no impact on the platforms I have used. However, since cache structure is machine dependent, I have left the feature in place.

3. Direct summation (CFLAGS = -DNOFASTMP): To accelerate computations, the program uses a modified fast multipole summation for the velocity and velocity derivative calculations. While this is accepted practice, it can induce small but quantifiable errors. This can be disabled with the NOFASTMP flag.

To use these options either together or separately, under the GNU make utility, you just set the switches on the command line. For instance,

```
make mpi=on xantisymm=on
```

will build an executable with both options.

If you do not have the GNU make utility on your system, you must edit the vanilla Makefile. It is easy to do. Just follow the instructions and comment/uncomment the appropriate sections of flags.

# Chapter 4

## Using BlobFlow

### 4.1 For the impatient.

A fluid simulation is not something for the impatient. There are some sample files included with the distribution. These should all work if you set your environment variables to point at the correct simulation files.

Even with reasonable initial conditions and physical parameters, BlobFlow might crash. If the computational parameters are too coarse to capture the relevant physical effects, the computational representation may overflow, divide by zero or use up all the memory. Strong numerical tools are fast and accurate but they are not necessarily bullet-proof.

### 4.2 Command line options

BlobFlow is run from the command line with three mandatory parameters.

```
e-flow -inputdir <INPUT_DIR> -config <CONFIG_FILE> -domdir <DOM_DIR>
```

BlobFlow will look in the directory INPUT\_DIR for the simulation and control files, with names CONFIG\_FILE.sim and CONFIG\_FILE.ctl, respectively. BlobFlow also uses a lookup table when it evaluates the velocity field contributions from the deformed blobs [2]. It expects to find these files in the directory DOM\_DIR. (This is where you unpacked the domain.tgz archive.)

### 4.3 The simulation and control files

There are two groups of parameters for BlobFlow. The first group is the physical description of the problem such as the fluid viscosity and the initial conditions. The information in this file only describes the fluid experiment, and has nothing to do with BlobFlow, which is one means of approximating the outcome of the experiment. BlobFlow performs some elementary tests to make sure that the essential parameters are set properly before the simulation begins, but this is by no means bulletproof, and it is not recommended that this feature be used to assure that control or simulation parameters are self-consistent. This simulation description file has the general basic format

<paramname>: <paramvalue>

The required fields in the simulation description file are

- Viscosity: [nonnegative number] - The kinematic viscosity of the fluid.
- FrameStep: [positive number] - This is the time interval over which one wishes observe the fluid flow. BlobFlow will write out the state of the system at increments of this time interval.
- EndTime: [positive number] - The time at which the simulation ends. BlobFlow always assumes that the initial conditions refer to  $t = 0$ .
- The initial condition  $\omega(x, y, 0)$  can be specified two ways.
  - VtxInit: [string] - A file containing the initial blobs for the system. This file is a list of parameters for elliptical Gaussian basis functions that describe the vorticity field when  $t = 0$ . Each basis function entry consists six numbers:  $x, y, \gamma, \sigma, a, \theta$ . Together, this list of blobs represents  $\omega(\vec{x}, t = 0)$  in the form described in (2.1).
  - GrdInit: [string]; GrdX0: [number]; GrdX1: [number]; GrdY0: [number]; GrdY1: [number]; GrdNumPts: [integer] - A description of the initial vorticity field on a  $\text{GrdNumPts} \times \text{GrdNumPts}$  square lattice. The lattice spacing must be uniform and square which means  $\text{GrdX1} - \text{GrdX0} = \text{GrdY1} - \text{GrdY0}$ . BlobFlow will use reverse heat equation field interpolation to project the initial vorticity onto a collection of basis functions to be used as initial conditions for the vortex method. This initialization routine uses `InterpPopulationControl` from the control file.

There is one optional simulation parameter.

- XAntiSymmetry: [string=y/Y/yes/Yes] - Set this option if you want the code to assume that  $\omega(x, y, t) = -\omega(x, -y, t)$ . When using this feature, the code only requires half of the usual amount of information, and there is a savings of a factor of two in CPU time for a calculation. When this flag is set, one should only initialize the code with  $\omega$  defined on half the domain. The code will automatically produce the full vorticity field using the antisymmetric extension of the initial conditions.

The control file includes parameters that govern the performance and accuracy of the BlobFlow simulation. The accuracy of any fluid simulation depends up the exact solution. Of course, computational simulation is used to approximate solutions when the exact solution is not available. If this process sounds circular, it is! In practice, no one calculates a flow once. Investigators calculate the flow, observe the results, identify relevant features, try to recalculate the flow after adjusting the parameters to resolve features of interest.

The required and optional parameters in BlobFlow have changed significantly from version 2.x to version 3.x. The changes are all for the better. The algorithms are better and the algorithms require fewer parameters to control their performance. The diagnostic log file will contain annotations if your control file assigns values to parameters from version 2.x for algorithms that are no longer in BlobFlow,

The control file has the following required fields:

- TimeStep - This is the fundamental time integration interval as the vortex blobs move and evolve in the flow.
- InterpStep - BlobFlow uses reverse heat equation field interpolation to replace collections of inaccurate blobs<sup>1</sup> with a collection of accurate blobs[1]. The parameter controls when field interpolation will be applied.
- InterpVar - The field interpolation algorithm used in BlobFlow replaces inaccurate blobs with axisymmetric blobs. This parameter is the core size,  $\sigma^2$  in (2.1), of the new blobs.
- InterpPopulationControl - The field interpolation algorithm used in BlobFlow replaces inaccurate blobs with new blobs on a regular grid covering the entire domain. It is likely that many of these blobs have almost no circulation. This parameter is the *vorticity threshold* for blobs to be part of the new configuration. If the absolute value of the computed vorticity  $\omega_i = \gamma_i/h^2$  where  $h$  is the mesh width is lower than this parameter, the blob will be discarded from the simulation.

---

<sup>1</sup>A blob becomes less accurate when the core grows large or too elongated.

## 4.4 Running BlobFlow

BlobFlow generates a number of output files. A computational log file, called `INPUT_DIR/CONFIG.com`, contains ordinary information about BlobFlow as it progresses through the computation. The amount of CPU required to perform various portions of the velocity computation is stored in a file called `INPUT_DIR/CONFIG.cpu.log`. If one is running BlobFlow across multiple processors with MPI, each process generates a computational log with the name `INPUT_DIR)/CONFIG.comp.n.log` where  $n$  is the process rank beginning with 0. There is a diagnostic log file called `INPUT_DIR/CONFIG.diag.log` for debugging purposes. Under ordinary circumstances, this file should remain empty. If one is running the code with a multiprocessor, message passing log files are generated with the file names `INPUT_DIR/CONFIG.`

The vorticity field is written to files in the form `INPUT_DIR/CONFIGnnnn.vtx` where  $nnnn$  a four digit index for the state being dumped. If the `FrameStep` parameter is set to 0.1, the file `INPUT_DIR/CONFIG1234.vtx` contains basis functions representing the vorticity field at time  $t = 123.4$ . Each line of the file contains six parameters describing a basis function as in the definition of `VtxInit`. If one is running across multiple processors, only the process of rank 0 will write data files.

## 4.5 Postprocessing

BlobFlow basis functions are seldom a desirable form of output for interpretation or analysis. Most graphics packages like to read data on some form of regular grid. The program `egrid` projects vortex basis functions onto a regular grid. As a tool, `egrid` is fairly unfriendly. It will look for a file called `egrid.default` which should have five numbers in it:  $(x_0, y_0)$ ,  $(x_1, y_1)$  and  $n$ , where the first coordinate pair specifies the lower left corner of the domain onto which you wish to project your data. The second coordinate is the upper right corner, and  $n$  is the desired number of mesh points. If it cannot find `egrid.default`, it will query the user for this information.

There are two switches for `egrid`. Ordinarily, `egrid` will keep the user informed of its progress row by row. However, if the `-q` switch is set, `egrid` will run silently. The `-x` switch generates the x-antisymmetric projection of the vorticity field corresponding to the `XAntiSymmetry` simulation parameter.

Since users typically want to project an whole group of files onto a regular grid, there is a Python front end for `egrid` which presents all this information in one place, and once the user sets all the parameters, will repeatedly run `egrid`.

## 4.6 Credits

Vortex methods and BlobFlow have been a passion of mine for more than a decade. While it is primarily a solitary hobby, I doubt I would be doing this without the support and encouragement of groups and individuals. A National Science Foundation grant (DMS-9971800) supported most of the fundamental research into deforming blobs for vortex methods. Most of the parallel computing was developed and tested on our NSF Scientific Computing Research Environments for the Mathematical Sciences (SCREMS) cluster (DMS-0322583). Lorena Barba and I have enjoyed working on field interpolation together. Prof. Stephen Siegel and his graduate student, Sam Moleus **check the spelling!**, verified that the serial and parallel version of BlobFlow are mathematically equivalent using their MPI-Spin tool. Rodrigo Platte was a postdoctoral fellow at the University of Delaware in 2006 when we worked on the improved Biot-Savart algorithm. The parallelization through message passing was accomplished with contributions from Ken Valentine, a math major at UMass Lowell during the Spring of 2000.

# Bibliography

- [1] L. A. Barba and L. F. Rossi. Global field interpolation for particle methods. *J. Comput. Phys.*, To appear, 2009.
- [2] R. B. Platte, L. F. Rossi, and T. B. Mitchell. Using global interpolation to evaluate the biot-savart integral for deformable elliptical gaussian vortex elements. *SIAM Journal on Scientific Computing*, 31(3):2342–2360, 2009.
- [3] L. F. Rossi. Achieving high-order convergence rates with deforming basis functions. *SIAM J. Sci. Comput.*, 26(3):885–906, 2005.
- [4] L. F. Rossi. A comparative study of Lagrangian methods using axisymmetric and deforming blobs. deforming blobs. *SIAM J. Sci. Comput.*, 27(4):1168–1180, 2006.

# Appendix A

## Message passing and parallelization.

When using BlobFlow on many processors whether it is one SMP machine or several networked processors, there are three things to remember. Every multicomputer is different. Every multicomputer is different. Every multicomputer is different.

While it is more and more common to find large numbers of unused computers to run your calculations, using BlobFlow with MPI may require some tuning. The most important of these is communication overhead. Scaling across multiple processors requires that the individual processors communicate from time to time. Naturally, it is best to keep this to a minimum because message passing is the slowest operation in this algorithm. Both the peer and the master-slave scheme do their best to keep all processors busy, but network latency can spoil things.

For the master-slave algorithm, `WORKSIZE` is a key parameter that can be tuned. This integer controls the size, in blobs or computational elements, of the work to be performed by individual processes. Larger worksizes mean larger packets will be exchanged less frequently. Smaller worksizes mean smaller packets will be exchanged more often. Tuning `WORKSIZE` for your network can dramatically improve (or diminish) the performance of BlobFlow.

For the peer algorithm, a similar parameter called `SMALLWORK` governs whether or not a process will share work with its companion if the companion has finished all of its tasks. If the processor has `SMALLWORK` or less to do, it does not share the work because it would not be worth the transmission time. However, if the processor has more than `SMALLWORK`, it will share half of it with the companion.

# Appendix B

## GNU Free Documentation License

GNU Free Documentation License  
Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free

software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use

technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location

until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections

entitled "Endorsements."

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not

as a draft) by the Free Software Foundation.