GSLSP

Extensions to the GNU Scientific Library

Version 1.0

13 January 2014

Patrick Alken

Table of Contents

1 Introduction

These routines provide support for constructing and manipulating sparse matrices in GSL, using an API similar to the `gsl_matrix` machinery. The basic structure is called `gsl_spmatrix`. There are two supported storage formats for sparse matrices: the triplet and compressed column storage (CCS) formats. The triplet format stores triplets (i, j, x) for each non-zero element of the matrix. This notation means that the (i, j) element of the matrix A is $A_{ij} = x$. Compressed column storage stores each column of non-zero values in the sparse matrix in a continuous memory block, keeping pointers to the beginning of each column in that memory block, and storing the row indices of each non-zero element. The triplet format is ideal for adding elements to the sparse matrix structure while it is being constructed, while the compressed column storage is better suited for matrix-matrix multiplication or linear solvers.

GSL does not provide a linear sparse matrix solver for $Ax = b$, since this is a highly complex problem and many advanced software packages already exist to efficiently solve it. The routines in this section provide a simple interface to construct a sparse matrix and convert it to a compressed format, where it can easily be passed to external linear sparse matrix solvers.

The `gsl_spmatrix` structure is defined as

```
typedef struct
{
  size_t size1;
  size_t size2;
  size_t *i;
  double *data;
  size_t *p;
  size_t nzmax;
  size_t nz;
  size_t *work;
  size_t flags;
} gsl_spmatrix;
```

This defines a *size1*-by-*size2* sparse matrix. The number of non-zero elements currently in the matrix is given by *nz*. For the triplet representation, *i*, *p*, and *data* are arrays of size *nz* which contain the row indices, column indices, and element value, respectively. So if $data[k] = x$, then $x = A(i[k], p[k])$. For compressed column storage, *i* and *data* are arrays of size *nz* containing the row indices and element values, similar to the triplet case. *p* is an array of size $size2 + 1$ where $p[j]$ points to the index in *data* of the start of column *j*. Thus, if $data[k] = A(i, j)$, then $i = i[k]$ and $p[j] <= k < p[j + 1]$.

work is additional workspace needed for various operations like converting from triplet to compressed column storage. *flags* indicates the type of storage format being used (triplet or compressed column).

The routines in this extension are defined in the header file '`gsl_spmatrix.h`'.

2 Sparse matrix allocation

The functions for allocating memory for a sparse matrix follow the style of `malloc` and `free`. They also perform their own error checking. If there is insufficient memory available to allocate a matrix then the functions call the GSL error handler with an error code of `GSL_ENOMEM` in addition to returning a null pointer.

`gsl_spmatrix * gsl_spmatrix_alloc` (*const size_t* `n1`, *const size_t* `n2`) [Function]

This function allocates a sparse matrix of size *n1*-by-*n2* and initializes it to all zeros. If the size of the matrix is not known at allocation time, both *n1* and *n2* may be set to 1, and they will automatically grow as elements are added to the matrix. This function sets the matrix to the triplet representation, which is the easiest for adding and accessing matrix elements. This function tries to make a reasonable guess for the number of non-zero elements (*nzmax*) which will be added to the matrix by assuming a sparse density of 10%. The function `gsl_spmatrix_alloc_nzmax` can be used if this number is known more accurately. The workspace is of size $O(nzmax)$.

`gsl_spmatrix * gsl_spmatrix_alloc_nzmax` (*const size_t* `n1`, *const size_t* `n2`, *const size_t* `nzmax`, *const size_t* `flags`) [Function]

This function allocates a sparse matrix of size *n1*-by-*n2* and initializes it to all zeros. If the size of the matrix is not known at allocation time, both *n1* and *n2* may be set to 1, and they will automatically grow as elements are added to the matrix. The parameter *nzmax* specifies the maximum number of non-zero elements which will be added to the matrix. It does not need to be precisely known in advance, since storage space will automatically grow using `gsl_spmatrix_realloc` if *nzmax* is not large enough. Accurate knowledge of this parameter reduces the number of reallocation calls required. The parameter *flags* specifies the storage format of the sparse matrix. Possible values are

`GSL_SPMATRIX_TRIPLET`
: This flag specifies triplet storage.

`GSL_SPMATRIX_COMPCOL`
: This flag specifies compressed column storage.

The allocated `gsl_spmatrix` structure is of size $O(nzmax)$.

`int gsl_spmatrix_realloc` (*const size_t* `nzmax`, *gsl_spmatrix ** `m`) [Function]

This function reallocates the storage space for *m* to accomodate *nzmax* non-zero elements. It is typically called internally by `gsl_spmatrix_set` if the user wants to add more elements to the sparse matrix than the previously specified *nzmax*.

`void gsl_spmatrix_free` (*gsl_spmatrix ** `m`) [Function]

This function frees the memory associated with the sparse matrix *m*.

3 Accessing sparse matrix elements

`double gsl_spmatrix_get` (*const gsl_spmatrix * `m`, const size_t `i`, const size_t `j`*) [Function]

This function returns element (*i*,*j*) of the matrix *m*. The matrix may be in triplet or compressed format.

`int gsl_spmatrix_set` (*gsl_spmatrix * `m`, const size_t `i`, const size_t `j`, const double `x`*) [Function]

This function sets element (*i*,*j*) of the matrix *m* to the value *x*. The matrix must be in triplet representation.

4 Initializing sparse matrix elements

Since the sparse matrix format only stores the non-zero elements, it is automatically initialized to zero upon allocation. The function `gsl_spmatrix_set_zero` may be used to re-initialize a matrix to zero after elements have been added to it.

`int gsl_spmatrix_set_zero` (*gsl_spmatrix * m*) [Function]

This function sets (or resets) all the elements of the matrix *m* to zero.

5 Copying sparse matrices

`gsl_spmatrix * gsl_spmatrix_memcpy` (*const gsl_spmatrix * `src`*) [Function]

This function makes a copy of the sparse matrix *src* and returns a pointer to a newly allocated copy, which must be freed by the caller when no longer needed. The matrix *src* may be in either triplet or compressed format.

`gsl_spmatrix * gsl_spmatrix_transpose_memcpy` (*const gsl_spmatrix * `src`*) [Function]

This function computes the transpose of *src* and stores it in a newly allocated matrix which is returned by the function. This matrix should be freed by the caller using `gsl_spmatrix_free` when no longer needed. The matrix *src* may be in either triplet or compressed format.

6 Sparse matrix operations

`gsl_spmatrix * gsl_spmatrix_add` (*const gsl_spmatrix* * `a`, *const gsl_spmatrix* * `b`) [Function]

This function adds the two matrices $a + b$ and stores the result in a newly allocated matrix which is returned. The result should be freed with `gsl_spmatrix_free` when no longer needed. The two matrices must have the same dimensions.

`int gsl_spmatrix_scale` (*gsl_spmatrix* * `m`, *const double* `x`) [Function]

This function scales all elements of the matrix m by the constant factor x. The result $m(i, j) \leftarrow xm(i, j)$ is stored in m.

7 Sparse matrix properties

`size_t gsl_spmatrix_nnz` (*const gsl_spmatrix * `m`*) [Function]

This function returns the number of non-zero elements in *m*.

`int gsl_spmatrix_equal` (*const gsl_spmatrix * `a`, const gsl_spmatrix * `b`*) [Function]

This function returns 1 if the matrices *a* and *b* are equal (by comparison of element values) and 0 otherwise. The matrices *a* and *b* must be either both triplet format or both compressed format for comparison.

8 Finding maximum and minimum elements of sparse matrices

int gsl_spmatrix_minmax (*const gsl_spmatrix* * m, *double* * min_out, *double* * max_out) [Function]

This function returns the minimum and maximum elements of the matrix *m*, storing them in *min_out* and *max_out*.

9 Sparse matrix compressed format

GSL supports the compressed column format, in which the non-zero elements in each column are stored contiguously in memory.

`gsl_spmatrix * gsl_spmatrix_compress` (*const gsl_spmatrix * T*) [Function]

This function creates a sparse matrix in compressed column format from the input sparse matrix T which must be in triplet format. A pointer to a newly allocated matrix is returned. The calling function should free the newly allocated matrix when it is no longer needed.

10 Conversion between sparse and dense matrices

The `gsl_spmatrix` structure can be converted into the dense `gsl_matrix` format and vice versa with the following routines.

`int gsl_spmatrix_d2sp` (*gsl_spmatrix * S*, *const gsl_matrix * A*) [Function]

This function converts the dense matrix A into sparse triplet format and stores the result in S.

`int gsl_spmatrix_sp2d` (*gsl_matrix * A*, *const gsl_spmatrix * S*) [Function]

This function converts the sparse matrix S into a dense matrix and stores the result in A. S must be in triplet format.

11 Sparse BLAS operations

GSL supports a limited number of BLAS operations for sparse matrices.

`int gsl_spblas_dgemv` (*const double* `alpha`, *const gsl_spmatrix ** `A`, *const gsl_vector ** `x`, *const double* `beta`, *gsl_vector ** `y`) [Function]

This function computes the matrix-vector product and sum $y \leftarrow \alpha A x + \beta y$, where A is sparse and the vectors x and y are dense. The matrix A may be in triplet or compressed format.

`gsl_spmatrix * gsl_spblas_dgemm` (*const double* `alpha`, *const gsl_spmatrix ** `A`, *const gsl_spmatrix ** `B`) [Function]

This function computes the sparse matrix-matrix product $C = \alpha AB$. A pointer to the newly allocated matrix C is returned and should be freed using `gsl_spmatrix_free` when no longer needed. The matrices A and B must be in compressed format.

12 Examples

This example program demonstrates the sparse matrix routines on the solution of a simple 1D Laplace equation on $[0, 1]$:

$$\frac{d^2u(x)}{dx^2} = f(x) = -\pi^2 \sin(\pi x)$$

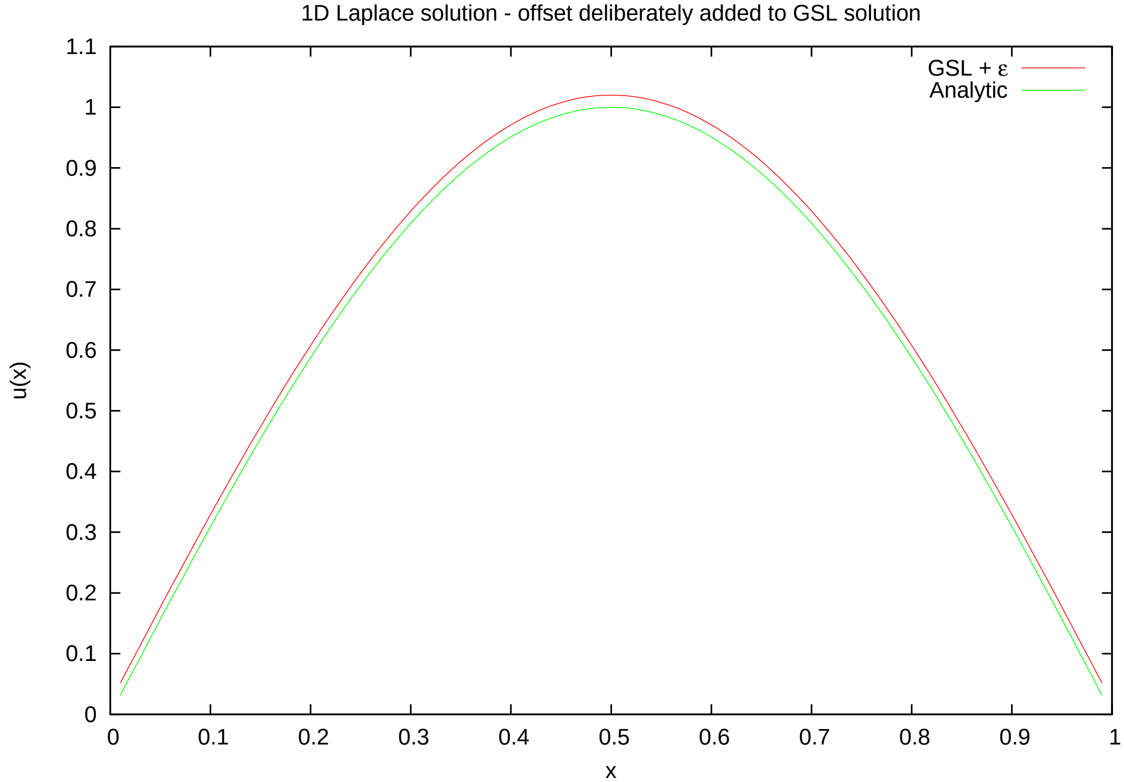
with boundary conditions $u(0) = u(1) = 0$. The analytic solution of this simple problem is $u(x) = \sin \pi x$. We will solve this problem by finite differencing the left hand side to give

$$\frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1}) = f_i$$

Defining a grid of N points, $h = 1/(N-1)$. In the finite difference equation above, $u_0 = u_{N-1} = 0$ are known from the boundary conditions, so we will only put the equations for $i = 1...N-2$ into the matrix system. The resulting matrix equation is

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & 0 & \dots \\ 0 & 1 & -2 & 1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & \dots & 1 & -2 & 1 \\ 0 & \dots & \dots & \dots & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-3} \\ f_{N-2} \end{pmatrix}$$

An example program which constructs this system using the `gsl_spmatrix` framework is given below. The system is solved using the dense GSL LU solver. The program output is shown in the following plot.




```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include <gsl/gsl_math.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_linalg.h>

#include <gslsp/gsl_spmatrix.h>

/* exact solution */
double u_exact(const double x) { return sin(M_PI * x); }

int
main()
{
  const size_t N = 100;                        /* number of grid points */
  const size_t n = N - 2;                      /* subtract 2 to exclude boundaries */
  const double h = 1.0 / (N - 1.0);            /* grid spacing */
  gsl_spmatrix *T = gsl_spmatrix_alloc(n ,n); /* triplet format */
  gsl_spmatrix *C;                             /* compressed format */
  gsl_vector *f = gsl_vector_alloc(n);         /* right hand side vector */
  gsl_vector *u = gsl_vector_alloc(n);         /* solution vector */
  size_t i;
```

```
  /* construct the sparse matrix for the finite difference equation */

  /* loop over interior grid points */
  for (i = 0; i < n; ++i)
    {
      /* u_{i+1} term, ignore at the boundary */
      if (i + 1 < n)
        gsl_spmatrix_set(T, i, i + 1, 1.0);

      gsl_spmatrix_set(T, i, i, -2.0);

      /* u_{i-1} term, ignore at the boundary */
      if (i > 0)
        gsl_spmatrix_set(T, i, i - 1, 1.0);
    }

  /* scale by h^2 */
  gsl_spmatrix_scale(T, 1.0 / (h * h));

  /* construct right hand side vector */
  for (i = 0; i < n; ++i)
    {
      double xi = (i + 1) * h;
      double fi = -M_PI * M_PI * sin(M_PI * xi);
      gsl_vector_set(f, i, fi);
    }

  /* convert to compressed column format */
  C = gsl_spmatrix_compress(T);

  /*
   * At this point, C->i, C->p, C->data contain the
   * row indices, column pointers, and matrix elements
   * of the compressed column storage format. These 3
   * arrays can be passed to external linear solvers.
   *
   * For illustration purposes, we will convert the
   * sparse matrix to a dense gsl_matrix and complete
   * the solution using a dense LU solver.
   */

  {
    gsl_matrix *A = gsl_matrix_alloc(n, n);
    gsl_permutation *p = gsl_permutation_alloc(n);
    int s;
```

```
      /* convert sparse to dense */
      gsl_spmatrix_sp2d(A, T);

      /* solve linear system A u = f */
      gsl_linalg_LU_decomp(A, p, &s);
      gsl_linalg_LU_solve(A, p, f, u);

      /* output solution */
      for (i = 0; i < n; ++i)
        {
          double xi = (i + 1) * h;
          double u_analytic = u_exact(xi);
          double u_gsl = gsl_vector_get(u, i);

          printf("%f %.12e %.12e\n", xi, u_gsl, u_analytic);
        }

      gsl_matrix_free(A);
      gsl_permutation_free(p);
    }

  gsl_spmatrix_free(T);
  gsl_spmatrix_free(C);
  gsl_vector_free(f);
  gsl_vector_free(u);

  return 0;
} /* main() */
```

13 References and Further Reading

The algorithms used by these functions are described in the following sources:

T. A. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006.

CSparse software library, https://www.cise.ufl.edu/research/sparse/CSparse/

GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful 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 non-commercially. Secondarily, 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, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

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. (Thus, 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. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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 Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document 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, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. 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, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include 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, PostScript or PDF 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.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

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 (or copies in media that commonly have printed covers) 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 computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. 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 fewer than five), unless they release you from this requirement.

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", Preserve 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. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, 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 to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

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, and that you preserve all their Warranty Disclaimers.

   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, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which 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 half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket 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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

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.

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) year your name.
Permission is granted to copy, distribute and/or modify
this document under the terms of the GNU Free
Documentation License, Version 1.2 or any later version
published by the Free Software Foundation; with no
Invariant Sections, no Front-Cover Texts, and no
Back-Cover Texts.  A copy of the license is included in
the section entitled ``GNU Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their
titles, with the Front-Cover Texts being list, and
with the Back-Cover Texts being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Function Index

Variable Index

Type Index

Concept Index