国际著名数学图书——影印版

A Multigrid Tutorial
Second Edition

多重网格法教程
（第2版）

William L. Briggs
Van Emden Henson 著
Steve F. McCormick

TSINGHUA UNIVERSITY PRESS
siam

A Multigrid Tutorial
Second Edition

多重网格法教程
（第2版）

William L. Briggs
Van Emden Henson 著
Steve F. McCormick

TSINGHUA UNIVERSITY PRESS
北京

siam

William L. Briggs, Van Emden Henson, Steve F. McCormick
A Multigrid Tutorial , Second Edition
ISBN: 0-898714-62-3

北京市版权局著作权合同登记号　图字：01-2008-0783

图书在版编目（CIP）数据

多重网格法教程 = A Multigrid Tutorial：第2版：英文/（美）布雷格斯（Briggs, W.L.）等著. -- 影印本. --北京：清华大学出版社，2011.2
（国际著名数学图书）
ISBN 978-7-302-24495-0

责任编辑：陈朝晖
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：175×245　**印　张**：13.25
版　　次：2011 年 2 月第 1 版　　**印　　次**：2011 年 2 月第1 次印刷
印　　数：1～3000
定　　价：29.00 元

产品编号：027807-01

Preface to the Second Edition

Twelve years have passed since the publication of the first edition of *A Multigrid Tutorial.* During those years, the field of multigrid and multilevel methods has expanded at a tremendous rate, reflecting progress in the development and analysis of algorithms and in the evolution of computing environments. Because of these changes, the first edition of the book has become increasingly outdated and the need for a new edition has become quite apparent.

With the overwhelming growth in the subject, an area in which I have never done serious research, I felt remarkably unqualified to attempt a new edition. Realizing that I needed some help, I recruited two experts to assist with the project. Steve McCormick (Department of Applied Mathematics, University of Colorado at Boulder) is one of the original researchers in the field of multigrid methods and the real instigator of the first edition. There could be no better collaborator on the subject. Van Emden Henson (Center for Applied Scientific Computing, Lawrence Livermore National Laboratory) has specialized in applications of multigrid methods, with a particular emphasis on algebraic multigrid methods. Our collaboration on a previous SIAM monograph made him an obvious choice as a co-author.

With the team in place, we began deliberating on the content of the new edition. It was agreed that the first edition should remain largely intact with little more than some necessary updating. Our aim was to add a roughly equal amount of new material that reflects important core developments in the field. A topic that probably should have been in the first edition comprises Chapter 6: FAS (Full Approximation Scheme), which is used for nonlinear problems. Chapter 7 is a collection of methods for four special situations that arise frequently in solving boundary value problems: Neumann boundary conditions, anisotropic problems, variable-mesh problems, and variable-coefficient problems. One of the chief motivations for writing a second edition was the recent surge of interest in algebraic multigrid methods, which is the subject of Chapter 8. In Chapter 9, we attempt to explain the complex subject of adaptive grid methods, as it appears in the FAC (Fast Adaptive Composite) Grid Method. Finally, in Chapter 10, we depart from the predominantly finite difference approach of the book and show how finite element formulations arise. This chapter provides a natural closing because it ties a knot in the thread of variational principles that runs through much of the book.

There is no question that the new material in the second half of this edition is more advanced than that presented in the first edition. However, we have tried to create a safe passage between the two halves, to present many motivating examples, and to maintain a tutorial spirit in much of the discourse. While the first half of the book remains highly sequential, the order of topics in the second half is largely arbitrary.

The FAC examples in Chapter 9 were developed by Bobby Philip and Dan Quinlan, of the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory, using AMR++ within the Overture framework. Overture is a parallel object-oriented framework for the solution of PDEs in complex and moving geometries. More information on Overture can be found at http://www.llnl.gov/casc/Overture.

We thank Irad Yavneh for a thorough reading of the book, for his technical insight, and for his suggestion that we enlarge Chapter 4. We are also grateful to John Ruge who gave Chapter 8 a careful reading in light of his considerable knowledge of AMG. Their suggestions led to many improvements in the book.

Deborah Poulson, Lisa Briggeman, Donna Witzleben, Mary Rose Muccie, Kelly Thomas, Lois Sellers, and Vickie Kearn of the editorial staff at SIAM deserve thanks for coaxing us to write a second edition and for supporting the project from beginning to end. Finally, I am grateful for the willingness of my co-authors to collaborate on this book. They should be credited with improvements in the book and held responsible for none of its shortcomings.

Bill Briggs
November 15, 1999
Boulder, Colorado

Preface to the First Edition

Assuming no acquaintance with the subject, this monograph presents the essential ideas that underlie multigrid methods and make them work. It has its origins in a tutorial given at the Third Copper Mountain Conference on Multigrid Methods in April, 1987. The goal of that tutorial was to give participants enough familiarity with multigrid methods so that they could understand the following talks of the conference. This monograph has been written in the same spirit and with a similar purpose, although it does allow for a more realistic, self-paced approach.

It should be clear from the outset that this book is meant to provide a basic grounding in the subject. The discussion is informal, with an emphasis on motivation before rigor. The path of the text remains in the lowlands where all of the central ideas and arguments lie. Crossroads leading to higher ground and more exotic topics are clearly marked, but those paths must be followed in the Suggested Reading and the Exercises that follow each chapter. We hope that this approach will give a good perspective of the entire multigrid landscape.

Although we will frequently refer to *the* multigrid method, it has become clear that multigrid is not a single method or even a family of methods. Rather, it is an entire approach to computational problem solving, a collection of ideas and attitudes, referred to by its chief developer Achi Brandt as *multilevel methods.*

Originally, multigrid methods were developed to solve boundary value problems posed on spatial domains. Such problems are made discrete by choosing a set of grid points in the domain of the problem. The resulting discrete problem is a system of algebraic equations associated with the chosen grid points. In this way, a physical grid arises very naturally in the formulation of these boundary value problems.

More recently, these same ideas have been applied to a broad spectrum of problems, many of which have no association with any kind of physical grid. The original multigrid approach has now been abstracted to problems in which the grids have been replaced by more general levels of organization. This wider interpretation of the original multigrid ideas has led to powerful new techniques with a remarkable range of applicability.

Chapter 1 of the monograph presents the model problems to which multigrid methods were first applied. Chapter 2 reviews the classical iterative (relaxation) methods, a firm understanding of which is essential to the development of multigrid concepts. With an appreciation of how the conventional methods work and why they fail, multigrid methods can be introduced as a natural remedy for restoring and improving the performance of the basic relaxation schemes. Chapters 3 and 4 develop the fundamental multigrid cycling schemes and discuss issues of implementation, complexity, and performance. Only in Chapter 5 do we turn to some theoretical questions. By looking at multigrid from a spectral (Fourier mode) point of view and from an algebraic (subspace) point of view, it is possible to give an explanation of why the basic multigrid cycling scheme works so effectively.

Not surprisingly, the body of multigrid literature is vast and continues to grow at an astonishing rate. The Suggested Reading list at the end of this tutorial [see the bibliography in the Second Edition] contains some of the more useful introductions, surveys, and classical papers currently available. This list is hardly exhaustive. A complete and cumulative review of the technical literature may be found in the *Multigrid Bibliography* (see Suggested Reading), which is periodically updated. It seems unnecessary to include citations in the text of the monograph. The ideas presented are elementary enough to be found in some form in many of the listed references.

Finally, it should be said that this monograph has been written by one who has only recently worked through the basic ideas of multigrid. A beginner cannot have mastered the subtleties of a subject, but often has a better appreciation of its difficulties. However, technical advice was frequently necessary. For this, I greatly appreciate the guidance and numerous suggestions of Steve McCormick, who *has* mastered the subtleties of multigrid. I am grateful to John Bolstad for making several valuable suggestions and an index for the second printing. For the fourth printing the Suggested Reading section has been enlarged to include six recently published books devoted to multigrid and multilevel methods. A genuinely new development is the creation of mg-net, a bulletin board/newsgroup service which is accessible by sending electronic mail to `mgnet@cs.yale.edu`. For the real production of this monograph, I am grateful for the typing skills of Anne Van Leeuwen and for the editorial assistance of Tricia Manning and Anne-Adele Wight at SIAM.

Contents

Chapter 1

Model Problems

Multigrid methods were originally applied to simple boundary value problems that arise in many physical applications. For simplicity and for historical reasons, these problems provide a natural introduction to multigrid methods. As an example, consider the two-point boundary value problem that describes the steady-state temperature distribution in a long uniform rod. It is given by the second-order boundary value problem

$$
\begin{aligned}
-u''(x) + \sigma u(x) &= f(x), \qquad 0 < x < 1, \quad \sigma \geq 0, && (1.1)\\
u(0) = u(1) &= 0. && (1.2)
\end{aligned}
$$

While this problem can be handled analytically, our present aim is to consider numerical methods. Many such approaches are possible, the simplest of which is a finite difference method (finite element formulations will be considered in Chapter 10). The domain of the problem $\{x : 0 \leq x \leq 1\}$ is partitioned into n subintervals by introducing the grid points $x_j = jh$, where $h = 1/n$ is the constant width of the subintervals. This establishes the grid shown in Fig. 1.1, which we denote Ω^h.

At each of the $n-1$ interior grid points, the original differential equation (1.1) is replaced by a second-order finite difference approximation. In making this replacement, we also introduce v_j as an approximation to the exact solution $u(x_j)$. This approximate solution may now be represented by a vector $\mathbf{v} = (v_1, \ldots, v_{n-1})^T$, whose components satisfy the $n - 1$ linear equations

$$
\begin{aligned}
\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} + \sigma v_j &= f(x_j), \qquad 1 \leq j \leq n-1, && (1.3)\\
v_0 = v_n &= 0.
\end{aligned}
$$

Defining $\mathbf{f} = (f(x_1), \ldots, f(x_{n-1}))^T = (f_1, \ldots, f_{n-1})^T$, the vector of right-side values, we may also represent this system of linear equations in matrix form as

$$
\frac{1}{h^2}\begin{bmatrix} 2+\sigma h^2 & -1 & & & \\ -1 & 2+\sigma h^2 & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -1 \\ & & & & -1 & 2+\sigma h^2 \end{bmatrix} \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_{n-1} \end{bmatrix}
$$

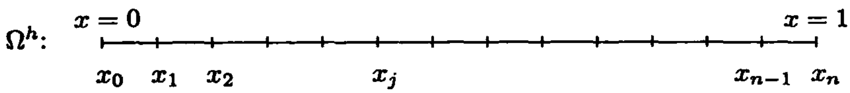

Figure 1.1: *One-dimensional grid on the interval* $0 \le x \le 1$. *The grid spacing is* $h = \frac{1}{n}$ *and the* j*th grid point is* $x_j = jh$ *for* $0 \le j \le n$.

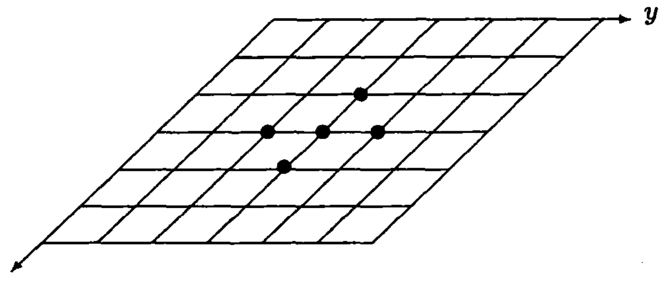

Figure 1.2: *Two-dimensional grid on the unit square. The solid dots indicate the unknowns that are related at a typical grid point by the discrete equations* (1.5).

or even more compactly as $A\mathbf{v} = \mathbf{f}$. The matrix A is $(n-1) \times (n-1)$, tridiagonal, symmetric, and positive definite.

Analogously, it is possible to formulate a two-dimensional version of this problem. Consider the second-order partial differential equation (PDE)

$$-u_{xx} - u_{yy} + \sigma u = f(x, y), \quad 0 < x < 1, \quad 0 < y < 1, \quad \sigma > 0. \tag{1.4}$$

With $\sigma = 0$, this is the Poisson equation; with $\sigma \neq 0$, it is the Helmholtz equation. We consider this equation subject to the condition that $u = 0$ on the boundary of the unit square.

As before, this problem may be cast in a discrete form by defining the grid points $(x_i, y_i) = (ih_x, jh_y)$, where $h_x = \frac{1}{m}$ and $h_y = \frac{1}{n}$. This two-dimensional grid is also denoted Ω^h and is shown in Fig. 1.2. Replacing the derivatives of (1.4) by second-order finite differences leads to the system of linear equations

$$\begin{gathered}\frac{-v_{i-1,j} + 2v_{ij} - v_{i+1,j}}{h_x^2} + \frac{-v_{i,j-1} + 2v_{ij} - v_{i,j+1}}{h_y^2} + \sigma v_{ij} = f_{ij}, \\ v_{i0} = v_{in} = v_{0j} = v_{mj} = 0, \qquad 1 \le i \le m-1, \ 1 \le j \le n-1.\end{gathered} \tag{1.5}$$

As before, v_{ij} is an approximation to the exact solution $u(x_i, y_j)$ and $f_{ij} = f(x_i, y_j)$.

There are now $(m-1)(n-1)$ interior grid points and the same number of unknowns in the problem. We can choose from many different orderings of the unknowns. For the moment, consider the *lexicographic* ordering by lines of constant i. The unknowns of the ith row of the grid may be collected in the vector $\mathbf{v}_i =$

$(v_{i1}, \ldots, v_{i,n-1})^T$ for $1 \le i \le m-1$. Similarly, let $\mathbf{f}_i = (f_{i1}, \ldots, f_{i,n-1})^T$. The system of equations (1.5) may then be given in block matrix form as

$$\begin{bmatrix} B & -aI & & \\ -aI & B & -aI & \\ \cdot & \cdot & \cdot & \\ & \cdot & \cdot & -aI \\ & & -aI & B \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{v}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{f}_{m-1} \end{bmatrix}.$$

This system is symmetric, block tridiagonal, and sparse. It has block dimension $(m-1) \times (m-1)$. Each diagonal block, B, is an $(n-1) \times (n-1)$ tridiagonal matrix that looks much like the matrix for the one-dimensional problem. Each off-diagonal block is a multiple, $a = \frac{1}{h_x^2}$, of the $(n-1) \times (n-1)$ identity matrix I.

Matrix Properties. The matrices produced by the discretization of self-adjoint boundary value problems have some special properties that are desirable for many numerical methods. Let A with elements a_{ij} be such a matrix. It is generally symmetric ($A = A^T$) and sparse (a large percentage of the elements are zero). These matrices are also often *weakly diagonally dominant*, which means that, in magnitude, the diagonal element is at least as large as the sum of the off-diagonal elements in the same row:

$$\sum_{j \neq i}^{n} |a_{ij}| \le |a_{ii}| \quad \text{for} \quad 1 \le i \le n.$$

These matrices are also *positive definite*, which means that, for all vectors $\mathbf{u} \neq \mathbf{0}$, we have $\mathbf{u}^T A \mathbf{u} > 0$. This property is difficult to interpret, but there are several alternate characterizations. For example, a symmetric positive definite matrix has real and positive eigenvalues. It can also be shown that if A is symmetric and diagonally dominant with positive diagonal elements, then A is positive definite. One other matrix property arises in the course of our work: a symmetric positive definite matrix with positive entries on the diagonal and nonpositive off-diagonal entries is called an *M-matrix*.

We occasionally appeal to stencils associated with discrete equations. For the one-dimensional model problem, the stencil representation of the matrix is

$$A = \frac{1}{h^2}(-1 \quad 2 + \sigma h^2 \quad -1).$$

The two-dimensional stencil for $h_x = h_y = h$ is

$$A = \frac{1}{h^2}\begin{pmatrix} & -1 & \\ -1 & 4 + \sigma h^2 & -1 \\ & -1 & \end{pmatrix}.$$

Stencils are useful for representing operators that interact locally on a grid. However, they must be used with care near boundaries.

The two model linear systems (1.3) and (1.5) provide the testing ground for many of the methods discussed in the following chapters. Before we proceed, however, it is useful to give a brief summary of existing methods for solving such systems.

During the past 50 years, a tremendous amount of work was devoted to the numerical solution of sparse systems of linear equations. Much of this attention was given to structured systems such as (1.3) and (1.5) that arise from boundary value problems. Existing methods of solution fall into two large categories: *direct methods* and *iterative* (or *relaxation*) methods. This tutorial is devoted to the latter category.

Direct methods, of which Gaussian elimination is the prototype, determine a solution exactly (up to machine precision) in a finite number of arithmetic steps. For systems such as (1.5) that arise from a two-dimensional elliptic equation, very efficient direct methods have been developed. They are usually based on the fast Fourier transform or the method of cyclic reduction. When applied to problems on an $n \times n$ grid, these methods require $O(n^2 \log n)$ arithmetic operations. Because they approach the minimum operation count of $O(n^2)$ operations, these methods are nearly optimal. However, they are also rather specialized and restricted primarily to systems that arise from separable self-adjoint boundary value problems.

Relaxation methods, as represented by the Jacobi and Gauss–Seidel iterations, begin with an initial guess at a solution. Their goal is to improve the current approximation through a succession of simple updating steps or iterations. The sequence of approximations that is generated (ideally) converges to the exact solution of the linear system. Classical relaxation methods are easy to implement and may be successfully applied to more general linear systems than most direct methods [23, 24, 26].

As we see in the next chapter, relaxation schemes suffer from some disabling limitations. Multigrid methods evolved from attempts to overcome these limitations. These attempts have been largely successful: used in a multigrid setting, relaxation methods are competitive with the fast direct methods when applied to the model problems, and they have more generality and a wider range of application.

In Chapters 1–5 of this tutorial, we focus on the two model problems. In Chapters 6–10, we extend the basic multigrid methods to treat more general boundary conditions, operators, and geometries. The basic methods can be applied to many elliptic and other types of problems without significant modification. Still more problems can be treated with more sophisticated multigrid methods.

Finally, the original multigrid ideas have been extended to what are more appropriately called *multilevel methods*. Purely algebraic problems (for example, network and structural problems) have led to the development of *algebraic multigrid* or *AMG*, which is the subject of Chapter 8. Beyond the boundaries of this book, multilevel methods have been applied to time-dependent problems and problems in image processing, control theory, combinatorial optimization (the traveling salesman problem), statistical mechanics (the Ising model), and quantum electrodynamics. The list of problems amenable to multilevel methods is long and growing. But first we must begin with the basics.

Exercises

1. **Derivative (Neumann) boundary conditions.** Consider model problem (1.1) subject to the *Neumann boundary conditions* $u'(0) = u'(1) = 0$. Find the system of linear equations that results when second-order finite differences are used to discretize this problem at the grid points $x_0, \ldots, x_n$. At the end

points, x_0 and x_n, one of many ways to incorporate the boundary conditions is to let $v_1 = v_0$ and $v_{n-1} = v_n$. (We return to this problem in Chapter 7.) How many equations and how many unknowns are there in this problem? Give the matrix that corresponds to this boundary value problem.

2. **Ordering unknowns.** Suppose the unknowns of system (1.5) are ordered by lines of constant j (or y). Give the block structure of the resulting matrix and specify the dimensions of the blocks.

3. **Periodic boundary conditions.** Consider model problem (1.1) subject to the *periodic boundary conditions* $u(0) = u(1)$ and $u'(0) = u'(1)$. Find the system of linear equations that results when second-order finite differences are used to discretize this problem at the grid points $x_0, \ldots, x_{n-1}$. How many equations and unknowns are there in this problem?

4. **Convection terms in two dimensions.** A convection term can be added to the two-dimensional model problem in the form

$$-\epsilon(u_{xx} + u_{yy}) + au_x = f(x).$$

Using the grid described in the text and second-order central finite difference approximations, find the system of linear equations associated with this problem. What conditions must be met by a and ϵ for the associated matrix to be diagonally dominant?

5. **Three-dimensional problem.** Consider the three-dimensional Poisson equation

$$-u_{xx} - u_{yy} - u_{zz} = f(x, y, z).$$

Write out the discrete equation obtained by using second-order central finite difference approximations at the grid point (x_i, y_j, z_k). Assuming that the unknowns are ordered first by lines of constant x, then lines of constant y, describe the block structure of the resulting matrix.

Chapter 2

Basic Iterative Methods

We now consider how model problems (1.3) and (1.5) might be treated using conventional iterative or relaxation methods. We first establish the notation for this and all remaining chapters. Let

$$A\mathbf{u} = \mathbf{f}$$

denote a system of linear equations such as (1.3) or (1.5). We always use $\mathbf{u}$ to denote the exact solution of this system and $\mathbf{v}$ to denote an approximation to the exact solution, perhaps generated by some iterative method. Bold symbols, such as $\mathbf{u}$ and $\mathbf{v}$, represent vectors, while the jth components of these vectors are denoted by u_j and v_j. In later chapters, we need to associate $\mathbf{u}$ and $\mathbf{v}$ with a particular grid, say Ω^h. In this case, the notation $\mathbf{u}^h$ and $\mathbf{v}^h$ is used.

Suppose that the system $A\mathbf{u} = \mathbf{f}$ has a unique solution and that $\mathbf{v}$ is a computed approximation to $\mathbf{u}$. There are two important measures of $\mathbf{v}$ as an approximation to $\mathbf{u}$. One is the *error* (or *algebraic error*) and is given simply by

$$\mathbf{e} = \mathbf{u} - \mathbf{v}.$$

The error is also a vector and its magnitude may be measured by any of the standard vector norms. The most commonly used norms for this purpose are the maximum (or infinity) norm and the Euclidean or 2-norm, defined, respectively, by

$$\|\mathbf{e}\|_\infty = \max_{1 \le j \le n} |e_j| \quad \text{and} \quad \|\mathbf{e}\|_2 = \left\{ \sum_{j=1}^{n} e_j^2 \right\}^{1/2}.$$

Unfortunately, the error is just as inaccessible as the exact solution itself. However, a computable measure of how well $\mathbf{v}$ approximates $\mathbf{u}$ is the *residual*, given by

$$\mathbf{r} = \mathbf{f} - A\mathbf{v}.$$

The residual is simply the amount by which the approximation $\mathbf{v}$ fails to satisfy the original problem $A\mathbf{u} = \mathbf{f}$. It is also a vector and its size may be measured by the same norm used for the error. By the uniqueness of the solution, $\mathbf{r} = \mathbf{0}$ if and only if $\mathbf{e} = \mathbf{0}$. However, it may *not* be true that when $\mathbf{r}$ is small in norm, $\mathbf{e}$ is also small in norm.

Residuals and Errors. A residual may be defined for any numerical approximation and, in many cases, a small residual does *not* necessarily imply a small error. This is certainly true for systems of linear equations, as shown by the following two problems:

$$\begin{pmatrix} 1 & -1 \\ 21 & -20 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -1 \\ -19 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & -1 \\ 3 & -1 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Both systems have the exact solution $\mathbf{u} = (1,2)^T$. Suppose we have computed the approximation $\mathbf{v} = (1.95, 3)^T$. The error in this approximation is $\mathbf{e} = (-0.95, -1)^T$, for which $\|\mathbf{e}\|_2 = 1.379$. The norm of the residual in $\mathbf{v}$ for the first system is $\|\mathbf{r_1}\|_2 = 0.071$, while the residual norm for the second system is $\|\mathbf{r_2}\|_2 = 1.851$. Clearly, the relatively small residual for the first system does not reflect the rather large error. See Exercise 18 for an important relationship between error and residual norms.

Remembering that $A\mathbf{u} = \mathbf{f}$ and using the definitions of $\mathbf{r}$ and $\mathbf{e}$, we can derive an extremely important relationship between the error and the residual (Exercise 2):

$$A\mathbf{e} = \mathbf{r}.$$

We call this relationship the *residual equation.* It says that the error satisfies the same set of equations as the unknown $\mathbf{u}$ when $\mathbf{f}$ is replaced by the residual $\mathbf{r}$. The residual equation plays a vital role in multigrid methods and it is used repeatedly throughout this tutorial.

We can now anticipate, in an imprecise way, how the residual equation can be used to great advantage. Suppose that an approximation $\mathbf{v}$ has been computed by some method. It is easy to compute the residual $\mathbf{r} = \mathbf{f} - A\mathbf{v}$. To improve the approximation $\mathbf{v}$, we might solve the residual equation for $\mathbf{e}$ and then compute a new approximation using the definition of the error

$$\mathbf{u} = \mathbf{v} + \mathbf{e}.$$

In practice, this method must be applied more carefully than we have indicated. Nevertheless, this idea of residual correction is very important in all that follows.

We now turn to relaxation methods for our first model problem (1.3) with $\sigma = 0$. Multiplying that equation by h^2 for convenience, the discrete problem becomes

$$\begin{aligned} -u_{j-1} + 2u_j - u_{j+1} &= h^2 f_j, \qquad 1 \le j \le n-1, \\ u_0 = u_n &= 0. \end{aligned} \tag{2.1}$$

One of the simplest schemes is the *Jacobi* (or simultaneous displacement) method. It is produced by solving the jth equation of (2.1) for the jth unknown and using the current approximation for the $(j-1)$st and $(j+1)$st unknowns. Applied to the vector of current approximations, this produces an iteration scheme that may be written in component form as

$$v_j^{(1)} = \frac{1}{2}\left(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j\right), \qquad 1 \le j \le n-1.$$

To keep the notation as simple as possible, the current approximation (or the initial guess on the first iteration) is denoted $\mathbf{v}^{(0)}$, while the new, updated approximation is denoted $\mathbf{v}^{(1)}$. In practice, once all of the $\mathbf{v}^{(1)}$ components have been computed, the procedure is repeated, with $\mathbf{v}^{(1)}$ playing the role of $\mathbf{v}^{(0)}$. These iteration sweeps are continued until (ideally) convergence to the solution is obtained.

It is important to express these relaxation schemes in matrix form, as well as component form. We split the matrix A in the form

$$A = D - L - U,$$

where D is the diagonal of A, and $-L$ and $-U$ are the strictly lower and upper triangular parts of A, respectively. Including the h^2 term in the vector $\mathbf{f}$, then $A\mathbf{u} = \mathbf{f}$ becomes

$$(D - L - U)\mathbf{u} = \mathbf{f}.$$

Isolating the diagonal terms of A, we have

$$D\mathbf{u} = (L + U)\mathbf{u} + \mathbf{f}$$

or

$$\mathbf{u} = D^{-1}(L + U)\mathbf{u} + D^{-1}\mathbf{f}.$$

Multiplying by D^{-1} corresponds exactly to solving the jth equation for u_j, for $1 \le j \le n-1$. If we define the Jacobi iteration matrix by

$$R_J = D^{-1}(L + U),$$

then the Jacobi method appears in matrix form as

$$\mathbf{v}^{(1)} = R_J\mathbf{v}^{(0)} + D^{-1}\mathbf{f}.$$

There is a simple but important modification that can be made to the Jacobi iteration. As before, we compute the new Jacobi iterates using

$$v_j^* = \frac{1}{2}\left(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j\right), \qquad 1 \le j \le n-1.$$

However, v_j^* is now only an intermediate value. The new iterate is given by the weighted average

$$v_j^{(1)} = (1-\omega)v_j^{(0)} + \omega v_j^* = v_j^{(0)} + \omega(v_j^* - v_j^{(0)}), \qquad 1 \le j \le n-1,$$

where $\omega \in \mathbf{R}$ is a weighting factor that may be chosen. This generates an entire family of iterations called the *weighted* or *damped Jacobi* method. Notice that $\omega = 1$ yields the original Jacobi iteration.

In matrix form, the weighted Jacobi method is given by (Exercise 3)

$$\mathbf{v}^{(1)} = [(1-\omega)I + \omega R_J]\mathbf{v}^{(0)} + \omega D^{-1}\mathbf{f}.$$

If we define the weighted Jacobi iteration matrix by

$$R_\omega = (1-\omega)I + \omega R_J,$$