LECTURE 28: LINEAR OPERATOR SPLITTING

A first example of problem-specific algorithms is afforded by consideration of the steady-state diffusion problem

$$-\nabla^2 u = -\Delta u = f, \tag{1}$$

where $u(\boldsymbol{x}), \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d$ can be understood to denote the temperature in an infinitesimal volume positioned at $\boldsymbol{x}$ in d-dimensional space, and f is a local rate of heat generation. The above arises from setting the time derivative zero in $\partial_t u - \Delta u = f$. Though often stated in this thermal language, the Poisson equation (1) is generally valid for unresolved transport by Brownian motion. The mathematical concept of an "infinitesimal volume" is interpreted as setting some length scale ℓ much smaller than the length scale L characterizing the size of the domain Ω. As an example, for heat conduction in a column of water of length $L = 1\text{m}$ the length scale of a quasi-infinitesimal volume can be considered as, say, $\ell = 1\,\mu\text{m}$. There is no physical significance to the mathematical limit process $\ell \to 0$ due to the discrete structure of matter, and for all practical purposes setting $\ell = 1\,\mu\text{m}$ is an acceptable threshold to delimit a phenomenon of interest to behavior that can be neglected. In this case the phenomenon of interest is the average transport of thermal energy in volumes of size greater than $\mathcal{O}(\ell^d)$. The detailed Brownian motion of the water molecules that occurs on length scales $s \approx 1\,\text{nm} \ll \ell$ can be neglected and is said to be *unresolved*. The only observable effect of this motion is that temperature gradients lead to a heat flux as described by $\boldsymbol{f}(u) = -\alpha \nabla u$ (Fourier's law). The same equation (1) arises in epidemiology when $\ell = 10\,\text{m}$, an average separation between an infected and susceptible individual, $L = 10\,\text{km}$, the size of a large city, and u is reinterpreted as the percentage of infected individuals in the population. Again, the detailed Brownian steps of size $s \approx 10\,\text{cm} \ll \ell$ taken by individuals can be neglected.

1. Poisson equation discretization

Understanding the underlying unresolved Brownian motion is useful in constructing numerical solutions. For $f = 0$, (1) becomes $\nabla^2 u = 0$, which states that there is no net heat flux in an infinitesimal volume, $\text{div} \cdot (\alpha \nabla u) = 0$, colloquially: "what flows in on one side goes out on the other". A function that satisfies Laplace's equation $\nabla^2 u = 0$ is said to be *harmonic*. For $d = 1$, the ordinary differential equation $\partial_x^2 u = 0$ is obtained with solution $u(x) = a + bx$, and boundary conditions u_0 at $x = 0$ and u_1 at $x = 1$ gives $u(x) = u_0 + (u_1 - u_0)x$. A temperature difference $u_0 \neq u_1$ at the boundaries induces diffusive fluxes that lead to the mean value $u(1/2) = (u_0 + u_1)/2$ at the midpoint. An analogous statement is made for $d > 1$ starting from Green's formula on ball B with spherical boundary S of radius R

$$\int_B (u \Delta v - v \Delta u)\, \mathrm{d}\omega = \int_{S=\partial B} \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) \mathrm{d}\sigma,$$

for scalar functions u, v. For $d = 3$ and $v = 1/r$ the mean value theorem

$$u(\boldsymbol{x}) = \frac{1}{4\pi R^2} \int_{S=\partial B} u(\boldsymbol{y})\, \mathrm{d}\sigma(\boldsymbol{y}), \tag{2}$$

is obtained, which states that the value of a harmonic function is the average of the values on a surrounding sphere. For $d = 2$ the analogous statement is

$$u(\boldsymbol{a}) = \frac{1}{2\pi r} \int_0^{2\pi} u(\boldsymbol{a} + r\cos\theta\, \boldsymbol{e}_x + r \sin\theta\, \boldsymbol{e}_y)\, \mathrm{d}\theta. \tag{3}$$

Midpoint quadrature of (3) over four subintervals gives

$$u_0 = \frac{u_1 + u_2 + u_3 + u_4}{4}. \tag{4}$$

Apply the above on a grid covering some arbitrary domain Ω with boundary $\Sigma = \partial\Omega$ to obtain

$$u_{i,j} = \frac{1}{4}(u_{i,j-1} + u_{i-1,j} + u_{i+1,j} + u_{i,j+1}). \tag{5}$$

Complications arise for general boundaries Σ, but for a square domain $\Omega = [0,1] \times [0,1]$ grid points $(x_i = ih, y_j = jh)$ align with the boundary, $h = 1/(n+1)$. Instead of two indices, one for each spatial direction, organize the grid values u through a single index $k = (j-1)n + i$, with u_k denoting the value at the interior points (x_i, y_j). The vector of interior grid values $\boldsymbol{u} = [\, u_1 \ \ldots \ u_m \,]$ has $m = n^2$ components, and the mean value theorem leads to the linear system

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{6}$$

◦

where $\boldsymbol{A}$ has a regular sparsity pattern induced by the uniform spacing of the grid

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & 0 & \dots & 0 & -1 & 0 & \dots & \dots & 0 \\ -1 & 4 & -1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ & \ddots & \ddots & \ddots & \ddots & & & \ddots & & \end{bmatrix}.$$

The vector $\boldsymbol{b}$ contains the boundary values, for example $b_1 = u_{1,0} + u_{0,1}$.

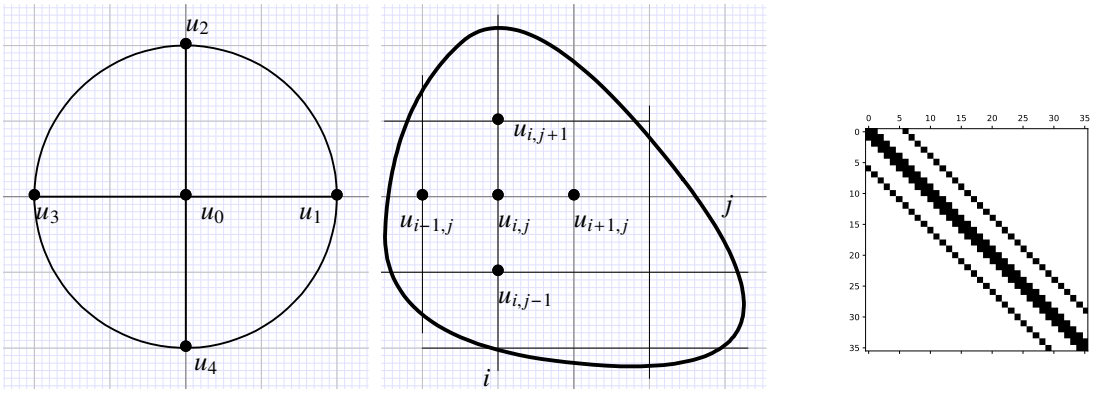

Figure 1. Left: Mean value theorem leads to $u_0 = (u_1 + u_2 + u_3 + u_4)/4$. Middle: Five-point finite difference stencil for Laplace operator. Right: Structure of $\boldsymbol{A}$ matrix resulting from discretization of Laplace operator.

The system (6) was obtained by discretization of the mean-value integral (3). The same linear system is also obtained by discretization of the differential equation

$$-\nabla^2 u = -u_{xx} - u_{yy} = 0,$$

where indices denote differentiation. The minus sign arises from compatibility with the unsteady form of the heat equation $\partial_t u - \nabla^2 u = f$. A centered finite difference approximation of the derivatives on the uniform grid leads to

$$u_{xx}(x_i, y_j) \cong \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2}, u_{yy}(x_i, y_j) \cong \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2},$$

and (6) is recovered. For the Poisson equation $f \neq 0$ the right hand side changes to

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b} + h^2 \boldsymbol{f} = \boldsymbol{c} \tag{7}$$

with $f_k = f(x_i, y_j)$. It is often the case that the same discrete system arises from both the differential and the integral formulation of a conservation law on a uniform grid.

2. Matrix splitting iteration

The above discussion of the underlying physics of the Poisson equation productively guides construction of numerical solution procedures. Solving the linear system (7) by general factorizations such as $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ or $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ is costly in terms of memory usage since the sparisty pattern is lost. For the uniform grid and square domain considered above the matrix $\boldsymbol{A}$ need not be explicitly stored at all since $a_{i,i} = 4$, and $a_{i,j} = -1$ when $|i - j| = 1$ or $|i - j| = n$. The discrete mean value theorem (5) suggests that some approximation $\boldsymbol{u}^{(l)}$ can be improved by the iteration

$$u_{i,j}^{(l+1)} = \frac{1}{4}\left(u_{i,j-1}^{(l)} + u_{i-1,j}^{(l)} + u_{i+1,j}^{(l)} + u_{i,j+1}^{(l)}\right). \tag{8}$$

The above is known as Jacobi iteration, and can be stated in matrix form by expressing $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U}, \tag{9}$$

with $\boldsymbol{L},\boldsymbol{U}$ containing non-zero components of $\boldsymbol{A}$ below, above the diagonal, and $\boldsymbol{D}$ containing the diagonal of $\boldsymbol{A}$. In contrast to the *multiplicative* decompositions considered up to now, the QR, LU, SVD, eigen or Schur decompositions, the decomposition (9) is now *additive*. Note that in (9) $\boldsymbol{L},\boldsymbol{U}$ are strictly lower, upper diagonal matrices with zeros on the diagonal in contrast to the notation for the standard LU factorization algorithm. Recall that the utility of matrix multiplication was associated with the representation of linear mapping composition. Additive decompositions such as (9) generally are useful when separating different aspects of a physical process, and are a simple example of *operator splitting*. For the discrete Poisson system Jacobi iteration can be expressed as

$$\boldsymbol{A}\boldsymbol{u}=\boldsymbol{c}\Rightarrow(\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U})\,\boldsymbol{u}=\boldsymbol{c}\Rightarrow\boldsymbol{u}^{(l+1)}=\boldsymbol{D}^{-1}\,(\boldsymbol{c}-\boldsymbol{L}\boldsymbol{u}^{(l)}-\boldsymbol{U}\boldsymbol{u}^{(l)}). \tag{10}$$

Several variants of the idea can be pursued. The matrix splitting (9) is useful in theoretical convergence analysis, but implementations directly use (8) within loops over the (i,j) indices. Updated values can be immediately utilized leading to either of the following iterations

$$\boldsymbol{u}^{(l+1)}=\boldsymbol{D}^{-1}\,(\boldsymbol{c}-\boldsymbol{L}\boldsymbol{u}^{(l+1)}-\boldsymbol{U}\boldsymbol{u}^{(l)}),\boldsymbol{u}^{(l+1)}=\boldsymbol{D}^{-1}\,(\boldsymbol{c}-\boldsymbol{L}\boldsymbol{u}^{(l)}-\boldsymbol{U}\boldsymbol{u}^{(l+1)}), \tag{11}$$

depending on loop organization. These are known as *Gauss-Seidel iterations*. Convergence might accelerated by extrapolation,

$$\boldsymbol{u}^{(l+1)}=\boldsymbol{u}^{(l)}+\omega\,[\boldsymbol{D}^{-1}\,(\boldsymbol{c}-\boldsymbol{L}\boldsymbol{u}^{(l)}-\boldsymbol{U}\boldsymbol{u}^{(l+1)})-\boldsymbol{u}^{(l)}]=(1-\omega)\boldsymbol{u}^{(l)}+\omega\boldsymbol{D}^{-1}\,(\boldsymbol{c}-\boldsymbol{L}\boldsymbol{u}^{(l)}-\boldsymbol{U}\boldsymbol{u}^{(l+1)}), \tag{12}$$

where the new iteration is continued by factor ω in the direction of the Gauss-Seidel update. When $\omega>1$ this is known as successive over-relaxation (SOR) and goes further in the Gauss-Seidel direction. Choosing $0<\omega<1$ leads to successive under-relaxation.

3. Convergence analysis

Turning now from algorithm construction to analysis of its behavior, simplify notation by letting $\boldsymbol{u}_k$ denote the current iterate. The previous notation $\boldsymbol{u}^{(l)}$ was convenient since individual components were referenced as in $u_{i,j}^{(l)}$, but convergence analysis is determined by the properties of the operator splitting and not of the current iterate. Introduce the error $\boldsymbol{\delta}_k$ at iteration k as the difference between the exact solution $\boldsymbol{u}$ and the current iterate $\boldsymbol{u}_k$, $\boldsymbol{\delta}_k=\boldsymbol{u}-\boldsymbol{u}_k$. Also introduce the *residual* $\boldsymbol{r}_k=\boldsymbol{c}-\boldsymbol{A}\boldsymbol{u}_k=\boldsymbol{A}\,\boldsymbol{\delta}_k$, and the correction to the current iterate $\boldsymbol{e}_k=\boldsymbol{u}_{k+1}-\boldsymbol{u}_k$
The above methods can be formulated as a residual correction algorithm through the steps:

1. residual computation, $\boldsymbol{r}_k=\boldsymbol{c}-\boldsymbol{A}\boldsymbol{u}_k$
2. correction computation, $\boldsymbol{e}_k=\boldsymbol{B}\boldsymbol{r}_k$
3. approximation update, $\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\boldsymbol{e}_k$

When $\boldsymbol{B}=\boldsymbol{A}^{-1}$ the exact solution is recovered in one step

$$\boldsymbol{e}_k=\boldsymbol{A}^{-1}(\boldsymbol{c}-\boldsymbol{A}\boldsymbol{u}_k)=\boldsymbol{u}-\boldsymbol{u}_k\Rightarrow\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\boldsymbol{e}_k=\boldsymbol{u}.$$

Iterative methods use some approximation of the (unknown) inverse, $\boldsymbol{B}\cong\boldsymbol{A}^{-1}$. Jacobi iteration uses $\boldsymbol{B}=\boldsymbol{D}^{-1}$ since

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\boldsymbol{D}^{-1}[\boldsymbol{c}-(\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U})\,\boldsymbol{u}_k]=\boldsymbol{D}^{-1}[\boldsymbol{c}-(\boldsymbol{L}+\boldsymbol{U})\,\boldsymbol{u}_k],$$

recovering (10). Table 1 shows several common choices for $\boldsymbol{B}$. Two key aspects govern the choice of the inverse approximant:

1. Computational efficiency stated as a requirement that each iteration cost either $\mathcal{O}(m)$ or $\mathcal{O}(m\log m)$ operations;
2. Capturing the essential aspects of $\boldsymbol{A}$.

Jacobi	$\boldsymbol{D}^{-1}$	Forward Gauss-Seidel	$(\boldsymbol{D}+\boldsymbol{L})^{-1}$
Weighted Jacobi	$\omega\boldsymbol{D}^{-1}$	Backward Gauss-Seidel	$(\boldsymbol{D}+\boldsymbol{U})^{-1}$
SOR	$\omega(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}$	Symmetric Gauss-Seidel	$(\boldsymbol{D}+\boldsymbol{U})^{-1}\boldsymbol{D}(\boldsymbol{D}+\boldsymbol{L})^{-1}$
Symmetric SOR	$\omega(2-\omega)(\boldsymbol{D}+\omega\boldsymbol{U})^{-1}\boldsymbol{D}\,(\boldsymbol{D}+\omega\boldsymbol{L})$	Richardson	$\omega\boldsymbol{I}$

Table 1. Common iterative methods

The iteration converges to the solution if $\|\boldsymbol{\delta}_k\| \to 0$ for increasing k. The error at iteration $k+1$ is expressed as

$$\boldsymbol{\delta}_{k+1} = \boldsymbol{u} - \boldsymbol{u}_{k+1} = \boldsymbol{u} - (\boldsymbol{u}_k + \boldsymbol{B}\boldsymbol{r}_k) = \boldsymbol{\delta}_k - \boldsymbol{B}\boldsymbol{A}(\boldsymbol{u} - \boldsymbol{u}_k) = (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})\,\boldsymbol{\delta}_k = (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})^{k+1}\,\boldsymbol{\delta}_0. \tag{13}$$

The repeated matrix multiplication indicates that the eigenstructure of the iteration matrix $\boldsymbol{M} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}$ determines iteration convergence. Indeed the above is simply power iteration for $\boldsymbol{M}$ and can be expected to converge as

$$\boldsymbol{\delta}_k \to \mu_1^k c_1 \boldsymbol{q}_1,$$

with $(\mu_1, \boldsymbol{q}_1)$ the eigenpair that corresponds to the largest eigenvalue in absolute value, $\boldsymbol{M}\boldsymbol{q}_1 = \mu_1 \boldsymbol{q}_1$, known as the spectral radius of $\boldsymbol{M}$, denoted by $\rho(\boldsymbol{M})$. Clearly, the above iterations will exhibit linear order of convergence when $\rho(\boldsymbol{M}) < 1$. The rate of convergence s_k at iteration k is estimated by the Rayleigh quotient

$$s_k = \frac{\boldsymbol{\delta}_k^T \boldsymbol{M}\,\boldsymbol{\delta}_k}{\boldsymbol{\delta}_k^T\,\boldsymbol{\delta}_k} = \frac{\boldsymbol{\delta}_k^T\,\boldsymbol{\delta}_{k+1}}{\boldsymbol{\delta}_k^T\,\boldsymbol{\delta}_k},$$

and is monitored in implementations of iterative methods, and determined by the eigenvalues $\lambda = 1 - \mu$ of $\boldsymbol{B}\boldsymbol{A}$.

The eigenstructure of $\boldsymbol{B}\boldsymbol{A}$ is difficult for arbitrary matrices $\boldsymbol{A}$, but can be carried out when $\boldsymbol{A}$ has special structure induced by known physical phenomena. The relation between analytical and numerical formulations plays an essential role in convergence analysis. The diffusion equation (5) leads to a symmetric matrix $\boldsymbol{A}, \boldsymbol{A} = \boldsymbol{A}^T$ due to two aspects:

1. the chosen discretization is symmetric using centered finite differences;
2. the operator itself exhibits symmetry.

Insight into the above two aspects is most readily gained from the one-dimensional case $-u_{xx} = f$ with homogeneous Dirichlet boundary conditions $u(0) = u(1) = 0$. The linear system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$ obtained from the centered finite difference discretization

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i, i = 1, \ldots, m, h = 1/(m+1), u_0 = u_{m+1} = 0,$$

has a symmetric tridiagonal system matrix $\boldsymbol{A} = \mathrm{diag}([\,-1\;\;2\;\;-1\,])$. The $\boldsymbol{A}^T = \boldsymbol{A}$ symmetry can be expressed through scalar products in a way that generalizes to differential operators. Recall that a real-valued scalar product $(\boldsymbol{u}, \boldsymbol{v})$ must satisfy symmetry $(\boldsymbol{v}, \boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{v})$. For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$ the standard inner product $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v}$ has this property. Consider the action of the operator $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ on the two terms. If $(\boldsymbol{A}\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{A}\boldsymbol{v})$ the operator $\boldsymbol{A}$ is said to be symmetric. For the inner product

$$(\boldsymbol{A}\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{A}\boldsymbol{u})^T \boldsymbol{v} = \boldsymbol{u}^T \boldsymbol{A}^T \boldsymbol{v}, (\boldsymbol{u}, \boldsymbol{A}\boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{A}\boldsymbol{v},$$

and the two expressions are equal if $\boldsymbol{A} = \boldsymbol{A}^T$. The same approach extends to the $d = 1$ diffusion operator $L = -\partial_{xx}$ using the scalar product

$$(f, g) = \int_0^1 f(x)\, g(x)\, \mathrm{d}x.$$

Applying integration by parts

$$(\partial_{xx} f, g) = \int_0^1 f''(x)\, g(x)\, \mathrm{d}x = [f' g]_{x=0}^{x=1} - \int_0^1 f'(x)\, g'(x)\, \mathrm{d}x = [f' g]_{x=0}^{x=1} - [f g']_{x=0}^{x=1} + \int_0^1 f(x)\, g''(x)\, \mathrm{d}x = b + (f, \partial_{xx} g).$$

For homogeneous boundary conditions $f(0) = f(1) = g(0) = g(1) = 0$, the symmetry condition is satisfied. Note that symmetry involves both the operator anId the boundary conditions of the problem. For $d = 2$ the scalar product

$$(u, v) = \int_\Omega u(x, y)\, v(x, y)\, \mathrm{d}x\mathrm{d}y,$$

is defined on the unit square $\Omega = [0, 1] \times [0, 1]$, and for homogeneous Dirichlet boundary conditions two applications of Green's formula leads to $(\nabla^2 u, v) = (u, \nabla^2 v)$, and the Laplace operator is symmetric.

For $d = 1$, $x_j = jh$, $h = 1/(m+1)$, the eigenvalues ν_k of $\boldsymbol{A} = \mathrm{diag}([\,1\;-2\;\;1\,])$ are inferred from those of the $-\partial_{xx}$ operator with homogeneous boundary conditions at $x = 0$, $x = 1$

$$-\partial_{xx} \sin(\kappa \pi x) = (\kappa \pi)^2 \sin(\kappa \pi x), \kappa \in \mathbb{N}.$$

Positing that an eigenvector $\boldsymbol{q}$ of $\boldsymbol{A}$ is the discretization of the continuum eigenfunction leads to

$$q_j = \sin(\kappa\,\pi\,x_j) = \sin(\kappa\,\pi\,jh) = \sin\left(\frac{j\,\kappa\,\pi}{m+1}\right),$$

hypothesis that is verified by the calculation of component j of $\boldsymbol{A}\boldsymbol{q}$

$$(\boldsymbol{A}\boldsymbol{q})_j = -\sin\left[\frac{(j-1)\,\kappa\,\pi}{m+1}\right] + 2\sin\left[\frac{j\,\kappa\,\pi}{m+1}\right] - \sin\left[\frac{(j+1)\,\kappa\,\pi}{m+1}\right] = 2\left[1-\cos\left(\frac{\kappa\,\pi}{m+1}\right)\right]\sin\left(\frac{j\,\kappa\,\pi}{m+1}\right),$$

thereby obtaining the eigenvalue

$$v_\kappa = 2\left[1-\cos\left(\frac{\kappa\,\pi}{m+1}\right)\right] = 4\sin^2\left[\frac{\kappa\,\pi h}{2}\right],$$

which recovers the analytical eigenvalue in the $h\to 0$ limit

$$\lim_{h\to 0}\frac{v_\kappa}{h^2} = (\kappa\,\pi)^2.$$

For Jacobi $\boldsymbol{B} = \boldsymbol{D}^{-1}$, so the eigenvalues of $\boldsymbol{B}\boldsymbol{A}$ are

$$\lambda_\kappa = 2\sin^2\left[\frac{\kappa\,\pi\,h}{2}\right], \kappa = 1, 2, \ldots, m.$$

The eigenvalues of $\boldsymbol{M} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}$ are therefore

$$\mu_\kappa = 1 - 2\sin^2\left[\frac{\kappa\,\pi\,h}{2}\right] = \cos(\kappa\,\pi\,h)$$

Replacing $h = 1/(m+1)$ the largest eigenvalue is obtained for $\kappa = 1$

$$\mu_{\max} = \mu_1 = \cos\left(\frac{\pi}{m+1}\right).$$

For large m, $\mu_1 \lessapprox 1$, and slow convergence is expected as verified in the numerical experiment from Fig. 2.

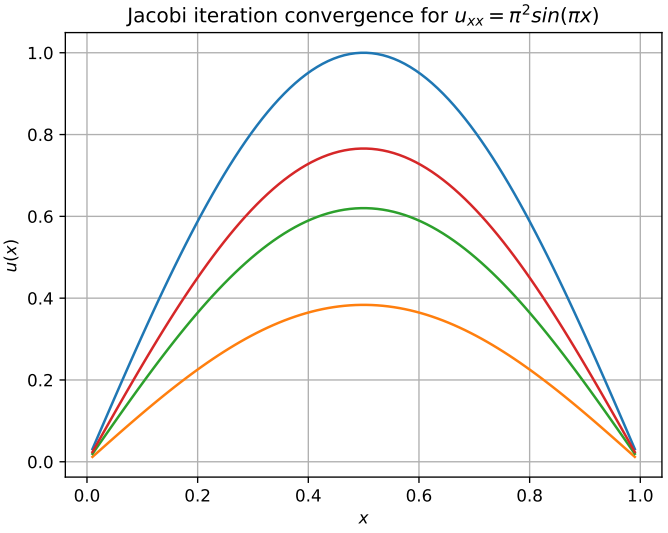

Figure 2. Convergence of Jacobi iteration. Blue: exact solution. Orange, green, red: iterates after 1000, 2000, 3000 iterations.

Jacobi iteration and its variants have limited practical utility by comparison to other iterative procedures due to slow convergence, but the concept of operator spliting has wide applicability. A more consequential example of operator splitting is to consider the advection diffusion equation

$$q_t + \boldsymbol{v}\cdot\nabla q = \alpha\,\Delta q,$$

which can be interpreted as stating that the time evolution of q is due to the effect of a diffusion operator $\mathscr{A} = \alpha\,\Delta$ and an advection operator $\mathscr{B} = -\boldsymbol{v}\cdot\nabla$

$$q_t = (\mathscr{A} + \mathscr{B})q.$$

Suppose both operators are discretized leading to matrices $\boldsymbol{A},\boldsymbol{B}$ and the discrete system

$$\boldsymbol{q}_t = (\boldsymbol{A}+\boldsymbol{B})\boldsymbol{q},$$

with initial condition $\boldsymbol{q}(\boldsymbol{x}, t=0) = \boldsymbol{q}_0$. Advancing the solution by a time step Δt can be written as

$$\boldsymbol{q}(t+\Delta t) = e^{\Delta t(\boldsymbol{A}+\boldsymbol{B})}\boldsymbol{q}(t),$$

and can be separated into two stages

$$\boldsymbol{q}^{(l+1)} = e^{\Delta t\boldsymbol{A}}\, e^{\Delta t\boldsymbol{B}}\boldsymbol{q}^{(l)}.$$

The quantity $\tilde{\boldsymbol{q}} = e^{\Delta t\boldsymbol{B}}\boldsymbol{q}^{(l)}$ captures advection effects and $\boldsymbol{q}^{(l+1)} = e^{\Delta t\boldsymbol{A}}\tilde{\boldsymbol{q}}$ is the diffusion correction. Since advection and diffusion are markedly different physical effects describing resolved versus unresolved transport, it can be expected that matrices $\boldsymbol{A},\boldsymbol{B}$ have different properties that require specific solution procedures.

By contrast, the Jacobi iteration splitting $\boldsymbol{A} = \boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U}$ does not separate physical effects and simply is suggested by the sparsity of $\boldsymbol{A}$ and computational efficiency per iteration. For example, the forward Gauss-Seidel iteration with $\boldsymbol{B} = (\boldsymbol{D}+\boldsymbol{L})^{-1}$ leads to

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + (\boldsymbol{D}+\boldsymbol{L})^{-1}[\boldsymbol{c} - (\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U})\,\boldsymbol{u}_k] = (\boldsymbol{D}+\boldsymbol{L})^{-1}(\boldsymbol{c}-\boldsymbol{U}\,\boldsymbol{u}_k) \Rightarrow (\boldsymbol{D}+\boldsymbol{L})\,\boldsymbol{u}_{k+1} = \boldsymbol{c}-\boldsymbol{U}\,\boldsymbol{u}_k. \tag{14}$$

The matrix $\boldsymbol{D}+\boldsymbol{L}$ is (non-strictly) lower-triangular and (14) is easily solved by forward substitution. The implementation is very simple to express while preserving two-dimensional indexing.

Algorithm (Componentwise forward Gauss-Seidel)

for $i = 1:m_x$
 for $j = 1:m_y$
  $u(i,j) = [c(i,j) + u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1)]/4$
 end
end

In the above implementation a single memory space is used for $\boldsymbol{u}$ with new values taking place of the old, leading to just four floating point additions and one multiplication per grid point. Since the algorithm essentially takes the current average of neighboring values, it is also known as a *relaxation* method, smoothing out spatial variations in $\boldsymbol{u}$.

Though such simple implementation is desirable, the non-physical splitting $\boldsymbol{A} = \boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U}$ and associated slow convergence usually outweighs ease of coding effort and suggests looking for alternative approaches. The only scenarios where such simply implemented iterations find practical use is in parallel execution and as a preliminary modification of the system prior to use of some other algorithm.