Scientific Research Publishing

Applied Mathematics, 2018, 9, 749-763
http://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

A Poisson Solver Based on Iterations on a Sylvester System

Michael B. Franklin, Ali Nadim

Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA, USA
Email: ali.nadim@cgu.edu

How to cite this paper: Franklin, M.B. and Nadim, A. (2018) A Poisson Solver Based on Iterations on a Sylvester System. *Applied Mathematics*, **9**, 749-763.
https://doi.org/10.4236/am.2018.96052

Received: June 1, 2018
Accepted: June 26, 2018
Published: June 29, 2018

Abstract

We present an iterative scheme for solving Poisson's equation in 2D. Using finite differences, we discretize the equation into a Sylvester system, $AU+UB=F$, involving tridiagonal matrices A and B. The iterations occur on this Sylvester system directly after introducing a deflation-type parameter that enables optimized convergence. Analytical bounds are obtained on the spectral radii of the iteration matrices. Our method is comparable to Successive Over-Relaxation (SOR) and amenable to compact programming via vector/array operations. It can also be implemented within a multigrid framework with considerable improvement in performance as shown herein.

Keywords

Poisson's Equation, Sylvester System, Multigrid

1. Introduction

Poisson's equation $\nabla^2 u = f$, an elliptic partial differential equation [1], was first published in 1813 in the *Bulletin de la Société Philomatique* by Siméon-Denis Poisson. The equation has since found wide utility in applications such as electrostatics [2], fluid dynamics [3], theoretical physics [4], and engineering [5]. Due to its expansive applicability in the natural sciences, analytic and efficient approximate solution methods have been sought for nearly two centuries. Analytic solutions to Poisson's equation are unlikely in most scientific applications because the forcing or boundary conditions on the system cannot be explicitly represented by means of elementary functions. For this reason, numerical approximations have been developed, dating back to the Jacobi method in 1845. The linear systems arising from these numerical approximations are solved either directly, using methods like Gaussian elimination, or iteratively. To this

day, there are applications solved by direct solvers and others solved by iterative solvers, depending largely on the structure and size of the matrices involved in the computation. The 1950s and 1960s saw an enormous interest in relaxation type methods, prompted by the studies on optimal relaxation and work by Young, Varga, Southwell, Frankel and others. The books by Varga [6] and Young [7] give a comprehensive guide to iterative methods used in the 1960s and 1970s, and have remained the handbooks used by academics and practitioners alike [8].

The Problem Description

The Poisson equation on a rectangular domain is given by

$$\nabla^2 u = f \quad \text{in } \Omega = \{x, y \mid 0 \le x \le a, 0 \le y \le b\} \tag{1}$$

where $u = u(x, y)$ is to be solved in the 2D domain Ω, and $f(x, y)$ is the forcing function. Typical boundary conditions for this equation are either Dirichlet, where the value of $f(x, y)$ is specified on the boundary, or Neumann, where the value of the normal derivative is specified on the boundary. These are given mathematically as,

$$u = g_D \quad \text{or} \quad \partial u / \partial n \equiv \hat{\boldsymbol{n}} \cdot \nabla u = g_N \quad \text{on } \partial\Omega, \tag{2}$$

where $\hat{\boldsymbol{n}}$ is the outward unit normal along $\partial\Omega$ and g_D and g_N are the function values specified by Dirichlet or Neumann boundary conditions. It is also possible to have mixed boundary conditions along the boundary, where some edges have Dirichlet and some have Neumann, so long as the problem is well-posed. Furthermore, edges could be subject to Robin boundary conditions of the form $c_1 u + c_2 \partial u / \partial n = g_R$. Any numerical scheme designed to solve the Poisson equation should be robust in its ability to incorporate any form of boundary condition into the solver. A detailed discussion of boundary condition implementation is given in the Appendix. Discretizing (1) using central differences with equal grid size $\Delta x = \Delta y = h$ leads to an $M \times N$ rectangular array of unknown U, such that $U_{i,j} \approx u(x_i, y_j)$ (assuming that a and b are both integer multiples of h). This discretization leads to a linear system of the form $AU + UB = F$, *the Sylvester equation*, which can be solved either directly or iteratively. The direct method utilizes the Kronecker product approach [9], given by

$$K\boldsymbol{u} = \boldsymbol{f} \quad \text{where } K = \text{kron}(I, A) + \text{kron}(B^{\mathrm{T}}, I) \tag{3}$$

where $\boldsymbol{u}$ and $\boldsymbol{f}$ are appropriately ordered $MN \times 1$ column vectors obtained from the $M \times N$ arrays U and F, and K is a sparse $MN \times MN$ matrix. The Kronecker product, $\text{kron}(P, Q)$, of any two matrices P and Q is a partitioned matrix whose ij^{th} partition contains matrix Q multiplied by component p_{ij} of P. Due to the potentially large size of the system given in (3), direct solvers are not the preferred solution approach. Specifically addressed here is an iterative approach to solving the Sylvester equation,

$$AU^{\text{int}} + U^{\text{int}}B = F^{\text{int}}, \tag{4}$$

where U^{int} is the $m \times n$ array of interior unknowns (*not* including the known boundary values when Dirichlet boundary conditions are given) with $m = M - 2$ and $n = N - 2$. Operator matrices A and B are given by the $O\left(h^2\right)$ finite difference approximation to the second derivative,

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}. \tag{5}$$

For an array of unknowns $U^{\text{int}} \in \mathbb{R}^{m\times n}$, the operator matrices are of dimension $A \in \mathbb{R}^{m\times m}$, and $B \in \mathbb{R}^{n\times n}$. The matrix structure of U should be modified such that the first index i corresponds to the x-direction, and the second index j corresponds to the y-direction. With this orientation, multiplying U^{int} by the matrix A on the left approximates the second derivative in the x-direction, and multiplying U^{int} by the matrix B on the right approximates the second derivative in the y-direction.

2. Sylvester Iterative Scheme

Examining (4) it might seem natural to move one term to the right-hand side of the equation to achieve an iterative scheme such as:

$$\begin{aligned} AU + UB = F \quad &\Rightarrow \quad AU = F - UB \\ \Rightarrow \quad U^{k+1} &= A^{-1}F - A^{-1}U^k B, \end{aligned} \tag{6}$$

However, this scheme *diverges*, and an alternative approach is required to iterate on the Sylvester system. An appropriate method is to break up the iterative scheme into two "half-steps" as follows

1) First half-step: $U^k \to U^*$

$$\begin{aligned} AU = F - UB \quad &\Rightarrow \quad \left(A - \alpha I + \alpha I\right)U = F - UB \\ \Rightarrow \quad &\left(A - \alpha I\right)U^* = F - U^k\left(B + \alpha I\right) \end{aligned} \tag{7}$$

2) Second half-step: $U^* \to U^{k+1}$

$$\begin{aligned} UB = F - AU \quad &\Rightarrow \quad U\left(B - \beta I + \beta I\right) = F - AU \\ \Rightarrow \quad &U^{k+1}\left(B - \beta I\right) = F - \left(A + \beta I\right)U^* \end{aligned} \tag{8}$$

where U^* is some intermediate solution between steps k and $k+1$. Rearranging, this leads to

$$\begin{aligned} \left(I - \frac{1}{\alpha}A\right)U^* &= U^k\left(I + \frac{1}{\alpha}B\right) - \frac{1}{\alpha}F, \\ U^{k+1}\left(I - \frac{1}{\beta}B\right) &= \left(I + \frac{1}{\beta}A\right)U^* - \frac{1}{\beta}F. \end{aligned} \tag{9}$$

The iterative scheme (9) is similar to the Alternating Direction Implicit (ADI)

formulation [10], where Poisson's equation is reformulated to have pseudo-time dependency,

$$\frac{\mathrm{d}U}{\mathrm{d}t} = AU + UB - F, \tag{10}$$

which achieves the solution to Equation (4) when it reaches steady-state. This method is separated into two half-steps, the first time step going from time $k \to k+1/2$ treating the *x*-direction implicitly and the *y*-direction explicitly. The second time step then goes from time $k+1/2 \to k+1$, treating the *y*-direction implicitly and the *x*-direction explicitly. The two half-steps are,

$$\begin{aligned} \frac{U^{k+1/2} - U^k}{\Delta t/2} &= \frac{1}{h^2}\left[AU^{k+1/2} + U^k B\right], \\ \frac{U^{k+1} - U^{k+1/2}}{\Delta t/2} &= \frac{1}{h^2}\left[AU^{k+1/2} + U^{k+1} B\right], \end{aligned} \tag{11}$$

which leads to

$$\begin{aligned} \left(I - \frac{\Delta t}{2}A\right)U^{k+1/2} &= U^k\left(I + \frac{\Delta t}{2}B\right) - \frac{\Delta t}{2}F, \\ U^{k+1}\left(I - \frac{\Delta t}{2}B\right) &= \left(I + \frac{\Delta t}{2}A\right)U^{k+1/2} - \frac{\Delta t}{2}F. \end{aligned} \tag{12}$$

This iteration procedure looks nearly identical to our Sylvester iterations given in (9) with $\Delta t/2$ replaced by the unknown parameters $1/\alpha$ and $1/\beta$. However in our formulation, there is no pseudo-time dependency introduced. Instead, the eigenvalues of our operator matrices A and B are *deflated* to produce an iterative scheme that optimally converges, and finding the values of the parameters α and β becomes an optimization problem.

Convergence

After the Sylvester Equation (4) is modified into the iterative system (9), the iterative scheme can be written as a single step by substituting the expression for the intermediate solution U^* into the second step of the iterative process; this yields the single update equation for U^{k+1} given by

$$\begin{aligned} U^{k+1} = &\left(I + \frac{1}{\beta}A\right)\left(I - \frac{1}{\alpha}A\right)^{-1} U^k \left(I + \frac{1}{\alpha}B\right)\left(I - \frac{1}{\beta}B\right)^{-1} \\ &- \left[\frac{1}{\alpha}\left(I + \frac{1}{\beta}A\right)\left(I - \frac{1}{\alpha}A\right)^{-1} - \frac{1}{\beta}I\right] F \left(I - \frac{1}{\beta}B\right)^{-1}. \end{aligned} \tag{13}$$

Assuming that an exact solution U^{exact} exists that exactly satisfies the linear system (4), *i.e.* $AU^{\text{exact}} + U^{\text{exact}}B = F$, we define the error between the k^{th} iteration and the exact solution as

$$E^k \equiv U^k - U^{\text{exact}}. \tag{14}$$

Finding an update equation for the error is done by subtracting the error at the k^{th} step from the error at the $(k+1)^{st}$ step, noting that the expressions in-

volving the forcing F disappear, we arrive at

$$E^{k+1} = PE^kQ, \tag{15}$$

where the matrices P and Q are given by

$$P = \left(I + \frac{1}{\beta}A\right)\left(I - \frac{1}{\alpha}A\right)^{-1}, \quad Q = \left(I + \frac{1}{\alpha}B\right)\left(I - \frac{1}{\beta}B\right)^{-1}. \tag{16}$$

Denoting the m eigenvalues of $A \in \mathbb{R}^{m\times m}$ by λ_k^A, and the n eigenvalues of $B \in \mathbb{R}^{n\times n}$ by λ_k^B, the corresponding *deflated* eigenvalues of the iteration matrices P and Q are

$$\begin{aligned} \lambda_k^P &= \frac{1+\left(\lambda_k^A/\beta\right)}{1-\left(\lambda_k^A/\alpha\right)} = \frac{\alpha}{\beta}\left(\frac{\beta+\lambda_k^A}{\alpha-\lambda_k^A}\right), \quad k = 1,2,\cdots,m, \\ \lambda_k^Q &= \frac{1+\left(\lambda_k^B/\alpha\right)}{1-\left(\lambda_k^B/\beta\right)} = \frac{\beta}{\alpha}\left(\frac{\alpha+\lambda_k^B}{\beta-\lambda_k^B}\right), \quad k = 1,2,\cdots,n. \end{aligned} \tag{17}$$

A sufficient condition for convergence of the iterative process is achieved if the spectral radii of both iteration matrices P and Q are less than one,

$$\rho(P) \equiv \max\left|\lambda_k^P\right| < 1 \quad \text{and} \quad \rho(Q) \equiv \max\left|\lambda_k^Q\right| < 1. \tag{18}$$

The error at each consecutive iteration is decreased by the product of $\rho(P)$ and $\rho(Q)$,

$$\begin{aligned} &E^{k+1} = PE^kQ, \\ &\left\|E^{k+1}\right\| = \rho(P)\rho(Q)\left\|E^k\right\|, \\ &\left\|E^{k+1}\right\| = \left(\rho(P)\rho(Q)\right)^k\left\|E^0\right\|, \end{aligned} \tag{19}$$

where E^0 is the initial error. Often in practical applications, the exact solution is not known, so the error E^k cannot be computed directly. In this case, the preferred measure in iterative schemes is given by the residual, which measures the difference of the left and right hand sides of the linear system being solved. This will be further discussed in the Results section.

3. Finding Optimal Parameters α and β

Finding α and β is an optimization problem for achieving the fastest convergence rate of the Sylvester iterative scheme (9). Given the operator matrices A and B and their respective eigenvalues λ_k^A and λ_k^B, it seems feasible to find optimal values of α and β to minimize the spectral radii of the iteration matrices P and Q given in Equations (17). From (15) the error E^{k+1} is found by multiplying by P on the left, and Q on the right, thus the convergence is governed by the spectral radii of both P and Q.

Figure 1 shows the eigenvalues of P and Q for arbitrary m and n, given by Equation (17), plotted vs. the eigenvalues of A and B for some parameters α and β. It can be seen that as λ^A or λ^B get large in magnitude, the values of λ^P or λ^Q approach $-\alpha/\beta$ and $-\beta/\alpha$, respectively. This implies that if $\alpha \neq \beta$,

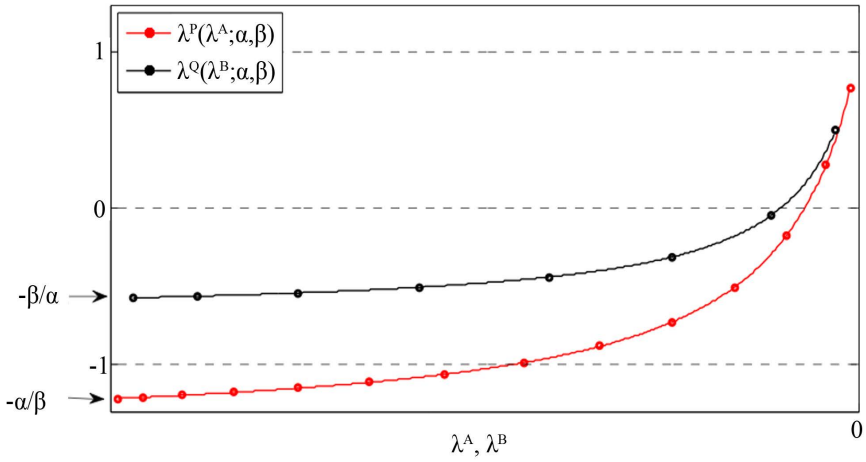

Figure 1. Eigenvalues $\lambda^P\left(\lambda^A;\alpha,\beta\right)$ and $\lambda^Q\left(\lambda^B;\alpha,\beta\right)$ vs. λ^Q and λ^B for given constant α and β. In this figure, $\alpha>\beta$, so $\rho(P)=\alpha/\beta>1$, and the scheme will diverge.

the high frequency eigenvalues of P and Q, in magnitude, will be greater than one, thus convergence condition (18) will not be satisfied. This provides the restriction for convergence that,

$$\alpha=\beta. \tag{20}$$

This optimal value of $\alpha=\beta$ will henceforth be called α^*. It is important to note that the operator $A\in\mathbb{R}^{m\times m}$ or $B\in\mathbb{R}^{n\times n}$ with the larger dimension

$$\ell\equiv\max(m,n), \tag{21}$$

has a larger range of eigenvalues. Figure 1 shows $m>n$, (*i.e.* $\ell=m$) so it can be seen that $\min\left(\lambda^A\right)<\min\left(\lambda^B\right)$ and $\max\left(\lambda^A\right)>\max\left(\lambda^B\right)$. This property of the eigenvalues is important when calculating an expression for c, which will soon prove to be a highly useful parameter for an adaptive approach to smoothing. Letting $\alpha=\beta=\alpha^*$ in (17) gives the following expression for the eigenvalues of P and Q:

$$\begin{aligned}\lambda_k^P&=\frac{\alpha^*+\lambda_k^A}{\alpha^*-\lambda_k^A},\quad k=1,2,\cdots,m,\\ \lambda_k^Q&=\frac{\alpha^*+\lambda_k^B}{\alpha^*-\lambda_k^B},\quad k=1,2,\cdots,n.\end{aligned} \tag{22}$$

Finding the optimal parameter α^* is done by considering the error reduction of Sylvester iterations on an arbitrary initial condition U_0. Assume that U_0 can be decomposed into its constituent error (Fourier) modes, ranging from low frequency (smooth) to high frequency (oscillatory) modes. Given that U_0 contains error modes of all frequencies, the most conservative method would be to choose α^* such that the spectral radii $\rho(P)$ and $\rho(Q)$ are minimized over the full range of frequencies. This ensures that all modes of error are efficiently relaxed, and convergence is governed by the product of spectral radii.

Referring to the lower curve in Figure 2, the conservative method of

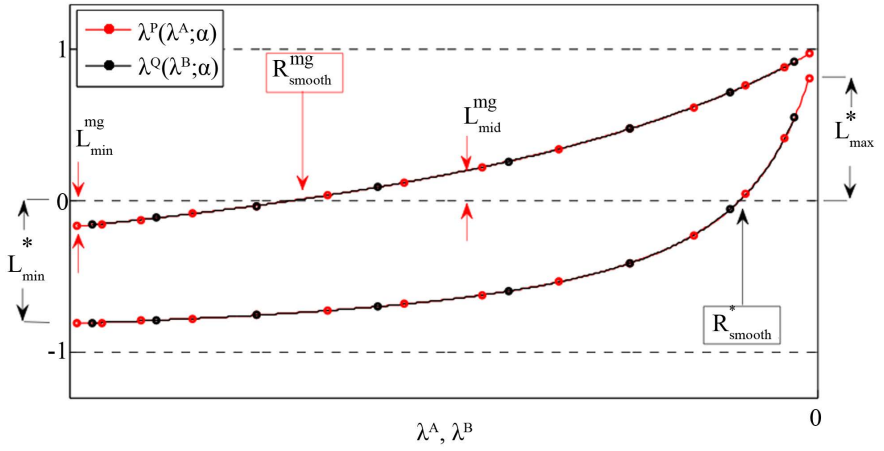

Figure 2. Eigenvalues $\lambda^P\left(\lambda^A;\alpha\right)$ and $\lambda^Q\left(\lambda^B;\alpha\right)$ illustrating the quantities involved in computing the optimal parameters α^* and α^*_{mg}, and their respective optimal smoothing regions R^*_{smooth} and R^{mg}_{smooth}. Note that the upper curve illustrates the method of determining α^* in the multigrid formulation, while the lower curve is for determining α^* in conservative Sylvester iterations.

determining α^* would be to set $L^*_{\min} = L^*_{\max}$ and according to (22),

$$-\left(\frac{\alpha^* + \lambda^A_{\min}}{\alpha^* - \lambda^A_{\min}}\right) = \left(\frac{\alpha^* + \lambda^A_{\max}}{\alpha^* - \lambda^A_{\max}}\right). \tag{23}$$

Noting that eigenvalues for all dimensions collapse onto the curves shown in **Figure 2**, this conservative approach "locks in" the value of the *larger* operator's spectral radius, thus providing an upper bound for convergence. **Figure 2** shows $m > n$, so $\rho(P) > \rho(Q)$, so convergence will be limited by $\rho(P)$. Equation (23) can then be solved for α^* giving

$$\alpha^* = \sqrt{\left|\min\left(\lambda^A_{\min}, \lambda^B_{\min}\right)\right| \times \left|\max\left(\lambda^A_{\max}, \lambda^B_{\max}\right)\right|}, \tag{24}$$

where absolute values are introduced as a reminder that λ^A, λ^B are negative. This value of the parameter α^* most uniformly smooths all frequencies for any arbitrary U_0 containing all frequency modes of error. It can be seen that the spectral radii of P and Q shown in **Figure 2** occur at either endpoint, and the minimum amplitude occurs near the intersection of the curve with the axis. Varying the parameter α^* in (22) controls the intersection point, and thus creates an effective "optimal smoothing region", denoted R^*_{smooth}. Modes of error associated with this optimal smoothing region will be damped fastest, which makes Sylvester iterations highly adaptive in nature. This adaptive nature of Sylvester iterations lends itself nicely to a multigrid formulation.

The Sylvester multigrid formulation is based on the philosophy that most iterative schemes, including Sylvester iterations, relax high frequency modes fastest, leaving low frequency components relatively unchanged [11]. On all grids traversed by a multigrid V-cycle, the high frequency modes are eliminated fastest by finding the optimal parameter value α^*_{mg} such that $L^{\text{mg}}_{\min} = L^{\text{mg}}_{\text{mid}}$, as

shown in the upper curve of Figure 2. The height $L_{\text{mid}}^{\text{mg}}$ is essentially the distance above the axis associated with the approximate "middle" eigenvalue in the range of λ^A or λ^B. This equality gives the following optimal parameter value for the Sylvester multigrid method

$$\alpha_{\text{mg}}^{*} = \sqrt{\left|\min\left(\lambda_{\min}^{A}, \lambda_{\min}^{B}\right)\right| \times \left|\min\left(\lambda_{\text{mid}}^{A}, \lambda_{\text{mid}}^{B}\right)\right|}, \tag{25}$$

where, if $m \neq n$, the minimum (*i.e.*, most negative) middle eigenvalue $\lambda_{\text{mid}} \equiv \left(\lambda_{\min} + \lambda_{\max}\right)/2$ is chosen to shrink the optimal smoothing region $R_{\text{smooth}}^{\text{mg}}$ such that high frequencies are smoothed most effectively. This choice of optimal parameter can be observed in Figure 2 to drastically decrease the magnitude of λ^P and λ^Q associated with high frequencies which significantly enhance relaxation in accordance with the multigrid philosophy.

To find analytical expressions for α^* and α_{mg}^{*}, it is necessary to have values for λ^A and λ^B. For Dirichlet boundary conditions, analytical expressions for λ^A and λ^B are derived below, but for Neumann boundary conditions numerical approaches are necessary to find λ^A and λ^B. The operator matrices *A* and *B* are each of tridiagonal form,

$$\begin{bmatrix} d_0 & d_1 & & & \\ d_1 & d_0 & d_1 & & \\ & \ddots & \ddots & \ddots & \\ & & d_1 & d_0 & d_1 \\ & & & d_1 & d_0 \end{bmatrix} \in \mathbb{R}^{p\times p}. \tag{26}$$

Tridiagonal matrices with *constant* diagonals, such as *A* and *B* for Dirichlet boundary conditions, have analytical expressions for their eigenvalues given by

$$\lambda_k = d_0 + 2d_1 \cos\left(\frac{k\pi}{p+1}\right), \quad k = 1, 2, \cdots, p \tag{27}$$

where *p* is the arbitrary dimension of the matrix [12]. Neumann boundary conditions alter the upper and lower diagonals of *A* or *B*, thus there is no analytical form of eigenvalues for Neumann boundary conditions. Using (5) and (27) gives the following analytic form of the eigenvalues of the tridiagonal matrices *A* and *B*,

$$\lambda_k = \frac{2}{h^2}\left(-1 + \cos\left(\frac{k\pi}{p+1}\right)\right), \quad k = 1, 2, \cdots, p, \tag{28}$$

which achieves minimum and maximum values given by

$$\lambda_{\min} = \frac{2}{h^2}\left(-1 + \cos\left(\frac{p\pi}{p+1}\right)\right) \text{ and } \lambda_{\max} = \frac{2}{h^2}\left(-1 + \cos\left(\frac{\pi}{p+1}\right)\right), \tag{29}$$

respectively. Using (24), (25), and (29) the analytic expressions for optimal parameters for both conservative and multigrid approaches are given by

$$\begin{aligned} \alpha^* &= \frac{2}{h^2}\sqrt{\left(-1 + \cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-1 + \cos\left(\frac{\pi}{\ell+1}\right)\right)}, \\ \alpha_{\text{mg}}^{*} &= \frac{\sqrt{2}}{h^2}\sqrt{\left(-1 + \cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-2 + \cos\left(\frac{\pi}{\ell+1}\right) + \cos\left(\frac{\ell\pi}{\ell+1}\right)\right)}, \end{aligned} \tag{30}$$

where again $\ell \equiv \max(m,n)$. Having expressions for α^* and α^*_{mg} allows λ^P and λ^Q to be found analytically using (22) which subsequently allows the spectral radii of the iteration matrices P and Q to be calculated. Knowing the spectral radii of the iteration matrices P and Q is highly advantageous, as it allows for an analysis of the Sylvester iterative scheme.

4. Analysis

The analysis of standard Sylvester iterations can be performed and describes the error reduction with each consecutive iteration using (19). Having the optimal parameters given by (30) and eigenvalues of P and Q in (22), the spectral radii can be calculated to be

$$\begin{aligned}\rho(P) &= \frac{-1+\cos\left(\frac{m\pi}{m+1}\right)+\sqrt{\left(-1+\cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-1+\cos\left(\frac{\pi}{\ell+1}\right)\right)}}{-1+\cos\left(\frac{m\pi}{m+1}\right)-\sqrt{\left(-1+\cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-1+\cos\left(\frac{\pi}{\ell+1}\right)\right)}},\\ \rho(Q) &= \frac{-1+\cos\left(\frac{n\pi}{n+1}\right)+\sqrt{\left(-1+\cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-1+\cos\left(\frac{\pi}{\ell+1}\right)\right)}}{-1+\cos\left(\frac{n\pi}{n+1}\right)-\sqrt{\left(-1+\cos\left(\frac{\ell\pi}{\ell+1}\right)\right)\left(-1+\cos\left(\frac{\pi}{\ell+1}\right)\right)}}.\end{aligned} \tag{31}$$

Rewriting the last expression of (19), we see that

$$\frac{\left\|E^k\right\|}{\left\|E^0\right\|} \sim \left(\rho(P)\rho(Q)\right)^k. \tag{32}$$

If we want to reduce our error to $\left\|E^k\right\| \sim \epsilon\left\|E^0\right\|$ and we wish to know how many iterations it will take to achieve this error reduction, using (32) we set $\left(\rho(P)\rho(Q)\right)^k \sim \epsilon$, and solving for k, we find it will take

$$k \sim \frac{\log(\epsilon)}{\log\left(\rho(P)\rho(Q)\right)} \tag{33}$$

iterations to reduce the error by ϵ. Here log can be with respect to any base, as long as the same one is used in both the numerator and denominator; e.g., the natural log can be used. Recall that the *exact* solution U^{exact} of (4) is only an *approximate* solution of the differential Equation (1) we are actually solving. Due to this, we can only expect accuracy of the truncation error of the approximation. With an $O(h^2)$ method, $U^{\text{exact}}_{i,j}$ differs from $U(x_i, y_j)$ on the order of h^2 so we cannot achieve better accuracy than this no matter how well we solve the linear system. Thus, it is practical to take ϵ to be something proportional to the expected global error, e.g. $\epsilon = Ch^2$ for some fixed C [12].

To calculate the order of work required asymptotically as $h \to 0$, (*i.e.* $m \to \infty$) using (33) and our choice for ϵ, we see that

$$k \sim \frac{\log(C)+2\log(h)}{\log\left(\rho(P)\rho(Q)\right)}. \tag{34}$$

The expressions for $\rho(P)$ and $\rho(Q)$ in (31) contain several cosine terms which can be Taylor expanded about different values. Cosines with arguments like πx can be expanded about $x=1$ or $x=0$ depending on the form of x, namely

$$\begin{aligned}\cos(\pi x) &\sim -1+\frac{\pi^2}{2}(x-1)^2+\mathcal{O}\left((x-1)^3\right) \quad \text{for } x\approx 1,\\ \cos(\pi x) &\sim 1-\frac{\pi^2}{2}x^2+\mathcal{O}\left(x^3\right) \quad \text{for } x\approx 0,\end{aligned} \tag{35}$$

where, from (31), the form of x is something like $m/(m+1)$ or $1/(m+1)$, which clearly approach one or zero, respectively, in the limit that $m\to\infty$. Using these expansions, along with the fact that $1/(1-x)\sim 1+x+O\left(x^2\right)$ for $x\ll 1$ to simplify the spectral radii, we arrive at the following

$$\rho(P)\sim\rho(Q)\sim 1-\frac{\pi}{\ell+1}+\frac{1}{4}\left(\frac{\pi}{\ell+1}\right)^2, \tag{36}$$

when $m,n\gg 1$. Since $h=1/(m+1)$, (34) combined with (36) gives the following order of work needed for convergence to within $\epsilon\sim Ch^2$:

$$k\sim\frac{-2\log(m+1)}{2\log\left(1-\frac{\pi}{\ell+1}\right)}\sim\frac{\ell}{\pi}\log(m), \tag{37}$$

where only linear terms are used from (36), and the latter simplified expression can be deduced by using the property that $\log(1+x)\sim x+O\left(x^2\right)$ for $x\ll 1$. Note that when $m=n$, the order of work for Sylvester iterations is $k\sim(m/\pi)\log(m)$, which is comparable to the work necessary for the Successive Over-Relaxation (SOR) algorithm to solve Poisson's equation [12]. This will be our basis for comparison in the Results section for standard Sylvester iterations.

5. Results

Problems solved by Sylvester iterations can, in general, be written shorthand as $\mathcal{L}U=F$, where $\mathcal{L}$ is a linear operator. In the case of Poisson's equation, $\mathcal{L}$ is the Laplacian operator. As an error measure, the discrete $\|\cdot\|_2$ norm of the residual, $r\equiv F-\mathcal{L}U$, can be measured at each iteration. This number provides the stopping criterion for our iterative schemes, namely the iterations are run until

$$\left\|r^{(k)}\right\|=\left\|F-\mathcal{L}U^{(k)}\right\|<\text{tol}\times\left\|r^{(0)}\right\|, \tag{38}$$

where $r^{(0)}$ is the initial residual, and tol is the tolerance. The tolerance is set to machine precision $\text{tol}\sim 10^{-16}$ to illustrate the *asymptotic* convergence rate,

$$q^{(k)}=\frac{\left\|r^{(k)}\right\|}{\left\|r^{(k-1)}\right\|}, \tag{39}$$

however, in practice, the discretization error $O\left(h^2\right)$ is the best accuracy that can be expected. These numerical results were run using MATLAB on a 1.5 GHz

Mac PowerPC G4. The model problem that is solved is given by

$$\begin{aligned} &\nabla^2 u = -2\left[y^2\left(1-6x^2\right)\left(1-y^2\right)+x^2\left(1-6y^2\right)\left(1-x^2\right)\right] \quad \text{in } \Omega, \\ &u = 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{40}$$

where $\Omega = \{x, y \mid 0 \le x \le 1, 0 \le y \le 1\}$, and whose exact solution

$$u^{\text{exact}}(x,y) = \left(x^2 - x^4\right)\left(y^4 - y^2\right), \tag{41}$$

is known so errors can be computed [11]. This model problem is used to show performance of both standard and multigrid Sylvester iterations. In all cases, the initial guess $U^{(0)}$ of the iterative scheme is a normalized random array and can be assumed to contain all modes of error.

5.1. Standard Sylvester Iterations

For comparison, standard Sylvester iterations were tested against Successive Over-Relaxation (SOR) with Chebyshev acceleration (see e.g., [13]). In SOR with Chebyshev acceleration, one uses odd-even ordering of the grid and changes the relaxation parameter ω at each half-step, which converges to the optimal relaxation parameter. The results are shown in Table 1. It is important to note that SOR iterations involve no matrix inversions, whereas Sylvester iterations do, thus the CPU time measure might not be an appropriate gauge for this particular comparison. From these results, it is clear that standard Sylvester iterations are comparable to SOR, and in most cases, converge to within tolerance in fewer iterations than SOR. An unfortunate artifact of standard iterative schemes is that as system size increases, so does the spectral radii governing the convergence. This can be observed in Table 1, as asymptotic convergence rates for each method $q_{\text{sylvester}}$ and q_{sor} steadily increase, thus requiring higher numbers of iterations to solve within tolerance. The number of Sylvester iterations required to converge is consistent with the predicted number of iterations given by Equation (33) letting $\epsilon = \text{tol} = 10^{-16}$.

5.2. Multigrid Sylvester Iterations

In multigrid Sylvester iterations, the performance of the $V(\nu_1, \nu_2)$-cycle using Sylvester iterations is compared to that using the traditional Gauss-Seidel (GS)

Table 1. Sylvester iterations vs. successive over-relaxation (SOR).

System Size	Sylvester	SOR			Sylvester	SOR
$M \times N$	Iterations	Iterations	$q_{\text{sylvester}}$	q_{sor}	CPU Time	CPU Time
16×16	84	93	0.638	0.674	0.059	0.072
32×32	173	185	0.806	0.819	0.474	0.612
64×64	352	368	0.899	0.905	4.591	5.082
128×128	709	735	0.949	0.951	43.13	48.28
256×256	1473	1469	0.975	0.975	744.5	786.7

Table 2. Multigrid sylvester iterations vs. multigrid gauss-seidel (GS) iterations.

System Size	Sylvester	GS			Sylvester	GS
$M \times N$	V(2,1)-cycles	V(2,1)-cycles	$q_{\text{sylvester}}^{\text{mg}}$	$q_{\text{gs}}^{\text{mg}}$	CPU Time	CPU Time
16×16	12	14	0.055	0.067	1.53	1.66
32×32	13	15	0.062	0.078	2.07	2.16
64×64	13	15	0.066	0.082	3.29	2.94
128×128	13	15	0.068	0.083	6.34	4.87
256×256	13	15	0.068	0.083	24.1	15.4
512×512	13	15	0.069	0.083	151.8	102.3

iterations. The parameter ν_1 represents the number of smoothing iterations done on each level of the downward branch of the V-cycle, while ν_2 represents the number done on the upward branch. In practice, common choices are $\nu = \nu_1 + \nu_2 \leq 3$, so our performance is based on the $V(2,1)$-cycle [14]. In each case, the V-cycle descends to the coarsest grid having gridwidth $h_0 = 1/2$, and in the Sylvester implementation, the value of α_{mg}^{*} is calculated to smooth high frequencies most effectively on each grid traversed by the cycle. The results are shown in Table 2. It can be seen that the asymptotic convergence rates $q_{\text{sylvester}}^{\text{mg}}$ and $q_{\text{gs}}^{\text{mg}}$ reach steady values independent of the gridwidth *h*. This is characteristic of multigrid methods, and enables the optimality of the multigrid method. It is clear when comparing the CPU times of the Sylvester multigrid formulation in Table 2 with standard Sylvester iterations in Table 1 that the multigrid framework is *substantially* faster (e.g., 30 times faster than standard iterations for a grid of size 256×256). It can also be seen that the asymptotic convergence rates are such that $q_{\text{sylvester}}^{\text{mg}} < q_{\text{gs}}^{\text{mg}}$, thus convergence is met in fewer $V(2,1)$ cycles using Sylvester smoothing versus Gauss-Seidel smoothing.

6. Conclusion

Sylvester iterations provide an alternative iterative scheme to solve Poisson's equation that is comparable to SOR in the number of iterations necessary to converge, namely converging to discretization accuracy within $k \sim (m/\pi)\log(m)$ iterations. The true benefit of the Sylvester iterations, however, comes from its adaptive ability to smooth any range of error frequencies, thus being a perfect candidate for smoothing in a multigrid framework. Multigrid $V(2,1)$-cycles using Sylvester smoothing have an asymptotic convergence rate of $q_{\text{sylvester}}^{\text{mg}} = 0.069$ (versus $q_{\text{gs}}^{\text{mg}} = 0.083$ for Gauss-Seidel smoothing) and indicate significant improvement in efficiency over standard Sylvester iterations.

References

[1] Douglass, C., Hasse, G. and Langer, U. (2003) A Tutorial on Elliptic PDE Solvers and Their Parallelization. Society for Industrial and Applied Math, Philadelphia. https://doi.org/10.1137/1.9780898718171

[2] Feig, M., Onufriev, A., Lee, M.S., Im, W., Case, D.A. and Brooks III, C.L. (2003) Performance Comparison of Generalized Born and Poisson Methods in the Calculation of Electrostatic Solvation Energies for Protein Structures. *Journal of Computational Chemistry*, **25**, 265-284. https://doi.org/10.1002/jcc.10378

[3] Ravoux, J.F., Nadim, A. and Haj-Hariri, H. (2003) An Embedding Method for Bluff Body Flows: Interactions of Two Side-by-Side Cylinder Wakes. *Theoretical and Computational Fluid Dynamics*, **16**, 433-466. https://doi.org/10.1007/s00162-003-0090-4

[4] Trellakis, A., Galick, A.T., Pacelli, A. and Ravaioli, U. (1997) Iteration Scheme for the Solution of the Two-Dimensional Schrodinger-Poisson Equations in Quantum Structures. *Journal of Applied Physics*, **81**, 7880-7884. https://doi.org/10.1063/1.365396

[5] Saraniti, M., Rein, A., Zandler, G., Vogl, P. and Lugli, P. (1996) An Efficient Multigrid Poisson Solver for Device Simulations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **15**, 141-150. https://doi.org/10.1109/43.486661

[6] Varga, R.S. (1962) Matrix Iterative Analysis. Prentice-Hall, New Jersey.

[7] Young, D.M. (1971) Iterative Solution of Large Linear Systems. Academic Press, New York.

[8] Brezinski, C. and Wuytack, L. (2001) Numerical Analysis: Historical Developments in the 20th Century. Elsevier Science B.V., Netherlands.

[9] Van Loan, C.F. (2000) The Ubiquitous Kronecker Product. *Journal of Computational and Applied Mathematics*, **123**, 85-100. https://doi.org/10.1016/S0377-0427(00)00393-9

[10] Peaceman, D.W. and Rachford, H.H. (1955) The Numerical Solution of Parabolic and Elliptic Differential Equations. *Journal of the Society for Industrial and Applied Mathematics*, **3**, 28-41. https://doi.org/10.1137/0103003

[11] Briggs, W.L., Henson, V.E. and McCormick, S.F. (2000) A Multigrid Tutorial. 2nd Edition, Society for Industrial and Applied Math, Philadelphia. https://doi.org/10.1137/1.9780898719505

[12] LeVeque, R.J. (2007) Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. Society for Industrial and Applied Math, Philadelphia. https://doi.org/10.1137/1.9780898717839

[13] Press, W., Vetterling, W., Teukolsky, S. and Flannery, B. (1992) Numerical Recipes in Fortran. 2nd Edition, Cambridge University Press, New York.

[14] Trottenberg, U., Oosterlee, C. and Schuller, A. (2001) Multigrid. Elsevier Academic Press, London.

Appendix

1) Boundary condition implementation

Solving the Poisson equation using Sylvester iterations lends itself nicely to boundary condition implementation. Dirichlet boundary conditions of the form

$$u(0,y)=u_1(y),\quad u(a,y)=u_2(y),\quad u(x,0)=u_3(x),\quad u(x,b)=u_4(x),\qquad (1)$$

where u_1, u_2, u_3 and u_4 are functions describing the edges of U, can be implemented as follows. The unknown values in the Sylvester system given in Equation (4) are the $m\times n$ array of interior values, where $m=M-2, n=N-2$. It is possible to incorporate Dirichlet boundary conditions directly into this interior system by examining the partitioned matrix product, for example AU, given by

$$\begin{bmatrix} -2 & \longleftarrow A_T^T \longrightarrow & 0 \\ \uparrow & & \uparrow \\ A_L & (A^{\text{int}})_{(m\times m)} & A_R \\ \downarrow & & \downarrow \\ 0 & \longleftarrow A_B^T \longrightarrow & -2 \end{bmatrix} \begin{bmatrix} U_{0,0} & \longleftarrow \vec{u_1}^T \longrightarrow & U_{0,N} \\ \uparrow & & \uparrow \\ \vec{u_3} & (U^{\text{int}})_{(m\times n)} & \vec{u_4} \\ \downarrow & & \downarrow \\ U_{M,0} & \longleftarrow \vec{u_2}^T \longrightarrow & U_{M,N} \end{bmatrix} \qquad (2)$$

with UB taking an analogous partitioned form. Multiplying through by h^2 associated with the operator matrices A and B, the partitioned Sylvester system for internal unknowns gives

$$A_L\cdot \mathbf{u}_1^{\mathrm{T}} + \left(A^{\text{int}}\right)\left(U^{\text{int}}\right) + A_R\cdot \mathbf{u}_2^{\mathrm{T}} + \mathbf{u}_3\cdot B_T^{\mathrm{T}} + \left(U^{\text{int}}\right)\left(B^{\text{int}}\right) + \mathbf{u}_4\cdot B_B^{\mathrm{T}} = \left(h^2\right)\left(F^{\text{int}}\right), \qquad (3)$$

where all matrix-vector products are $m\times n$ outer products. Note that the product AU incorporates Dirichlet boundary conditions in the x-direction, and UB incorporates Dirichlet boundary conditions in the y-direction. Combining the partitioned systems incorporating both A and B matrix multiplications and boundary conditions yields

$$\left(A^{\text{int}}\right)\left(U^{\text{int}}\right) = \left(h^2\right)\left(F^{\text{int}}\right) - \left(A_L\cdot \mathbf{u}_1^{\mathrm{T}} + A_R\cdot \mathbf{u}_2^{\mathrm{T}}\right) - \left(\mathbf{u}_3\cdot B_T^{\mathrm{T}} + \mathbf{u}_4\cdot B_B^{\mathrm{T}}\right), \qquad (4)$$

which is an $m\times n$ linear system for U^{int}.

For Neumann boundary conditions, the edge at which the condition is imposed becomes part of the internal unknowns in the Sylvester system. As an example, consider a Neumann boundary condition given by

$$\frac{\partial u}{\partial x} = g(y) \text{ on } x=0. \qquad (5)$$

Staying within the finite difference formulation of derivatives and letting $g(y_j)\equiv g_j$, this condition can be discretized and approximated with the $O(h^2)$ central difference approximation, which yields

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} \approx \frac{U_{i+1,j}-U_{i-1,j}}{2h} = g_j \quad \text{for } i=0,\ 0\le j\le N. \qquad (6)$$

For a Neumann condition along the edge $x=0$, the row vector $\mathbf{u}_1^{\mathrm{T}}$ described in (2) becomes a part of the internal array of unknowns U^{int}. In order to

implement this finite difference on the edge, we need to introduce a ghost layer with index $i=-1$, and pair Equation (6) to the second derivative operator AU for $i=0$. This gives

$$\begin{aligned} &\frac{U_{1,j}-U_{-1,j}}{2h}=g_j \quad \Rightarrow \quad U_{-1,j}=U_{1,j}-2hg_j, \\ &\left(\frac{\partial^2 u}{\partial x^2}\right)_{0,j} \approx \frac{\left(U_{1,j}-2hg_j\right)-2U_{0,j}+U_{1,j}}{h^2}=\frac{2U_{1,j}-2U_{0,j}}{h^2}-2\frac{g_j}{h}, \end{aligned} \tag{7}$$

which leads to the following partitioned form of AU,

$$\begin{bmatrix} -2 & \mathbf{2} & 0 & \cdots & 0 & 0 \\ \uparrow & & & & & \uparrow \\ A_L & & (A^{\text{int}})_{(m\times m)} & & & A_R \\ \downarrow & & & & & \downarrow \\ 0 & \longleftarrow & A_B^T & \longrightarrow & & -2 \end{bmatrix} \begin{bmatrix} U_{0,0} & & & & U_{0,N} \\ \uparrow & & & & \uparrow \\ U(x,0) & & (U^{\text{int}})_{((m+1)\times n)} & & U(x,b) \\ \downarrow & & & & \downarrow \\ U_{M,0} & \longleftarrow & U(a,y)^T & \longrightarrow & U_{M,N} \end{bmatrix}, \tag{8}$$

where the additional term $-2g_j/h$ is taken to the right hand side of the Sylvester system such that $F_{0,j}^{\text{int}}=F_{0,j}^{\text{int}}+2g_j/h$ for $0<j<N$. Comparing (8) to (2), the size of U^{int} changes from $m\times n$ to $(m+1)\times n$, and $A_{0,1}$ is changed from 1 to 2 (shown boldface in (8)), and the right hand side is slightly modified along that edge. Similarly, any edge with a Neumann condition can be handled in this fashion. It is clear that both Dirichlet and Neumann boundary conditions are very simple to implement in the Sylvester iteration method, and only slightly modify the structure of the arrays involved.