

An Accelerated Over-Relaxation Quarter-Sweep Point Iterative Method for Two-Dimensional Poisson Equation

(Kaedah Lelaran Titik Suku Sapuan Pengenduran Berlebihan Terpecut bagi Persamaan Poisson 2-Matra)

SHUKHRAT I. RAKHIMOV* & MOHAMED OTHMAN

ABSTRACT

Iterative methods, particularly over-relaxation methods, are efficiently and frequently used to solve large systems of linear equations, because in the solutions of partial differential equations, these methods are applied to systems which are resulted from different iterative schemes to discrete equations. In this paper we formulate an accelerated over-relaxation (AOR) method with the quarter-sweep iterative scheme applied to the Poisson equation. To benchmark the new method we conducted experiments by comparing it with the previous AOR methods based on full- and half-sweep iterative schemes. The results of the experiments and the estimation of the computational complexity of the methods proved the superiority of the new method.

Keywords: Accelerated over-relaxation; point iterative methods; Poisson equation

ABSTRAK

Kaedah lelaran, khususnya kaedah pengenduran berlebihan, kerap digunakan secara cekap bagi menyelesaikan sistem persamaan linear yang besar, kerana dalam penyelesaian persamaan pembezaan separa, kaedah lelaran ini digunakan ke atas sistem yang terhasil daripada skema lelaran berbeza untuk mendiskretkan persamaan. Dalam kertas kerja ini, kami memformulasikan kaedah pengenduran berlebihan tercepat (PBT) dengan menggunakan pendekatan sapuan suku dan mengaplikasikannya kepada sistem persamaan Poisson. Sebagai penanda aras kaedah terbaru ini, kami melakukan beberapa kajian dan membandingkan dengan kaedah PBT terdahulu yang menggunakan pendekatan lelaran sapuan penuh dan separuh. Hasil ujikaji tersebut dan anggaran pengiraan kekompleksitian bagi semua kaedah tersebut telah membuktikan keberkesanan kaedah baru ini.

Kata kunci: Kaedah lelaran titik; pengenduran berlebihan terpecut; persamaan Poisson

INTRODUCTION

The problem of solving partial differential equations arises in many mathematical models of scientific and engineering applications. One of the most popular and important research branches is the numerical solution of these equations, since the computer technologies are rapidly developing. The typical representative of partial differential equations is the Poisson’s equation, which is one of the keystones of electrostatics, mechanical engineering, and theoretical physics. It has a broad application in thermal conduction and fluid mechanics. A typical model problem for the two-dimensional case is formulated as follows:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \tag{1}$$

with Dirichlet boundary conditions: $u(x, y) = g(x, y)$ and $(x, y) \in \partial\Omega$.

For simplicity, we assume the domain Ω to be a square unit $[0 \leq x, y \leq 1]$. To solve the problem numerically, we uniformly discretise the domain in both x and y directions

with a grid size $h = 1/n$, where n is an arbitrary positive integer.

POINT ITERATIVE METHODS

Choosing various finite-difference schemes, we can iterate through the domain points (x_i, y_j) and get an approximate solution of the problem (1). There are three basic finite difference schemes, which produce the full-, half-, and quarter-sweep methods (Othman & Abdullah 2000).

The first of the three iterative schemes, the standard five-point formula, is used in the iterations within the full-sweep method to obtain:

$$v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j} = h^2 f_{i,j}, \tag{2}$$

where $v_{i,j}$ is an approximation to the exact solution $u(x_i, y_j)$ at the grid point (x_i, y_j) and $f_{i,j} = f(x_i, y_j)$.

The clockwise rotation of the $x-y$ axis by 45° results in the following finite-difference scheme formula:

$$v_{i+1,j+1} + v_{i-1,j-1} + v_{i+1,j-1} + v_{i-1,j+1} - 4v_{i,j} = 2h^2 f_{i,j}. \tag{3}$$

This finite-difference scheme is used in the half-sweep method (Abdullah 1991), when the iteration process goes through half of all the points and, after the convergence, the remaining points are computed directly using (2).

Taking the grid spacing $2h$ in standard five-point formula will result in the quarter-sweep approximation scheme:

$$v_{i+2,j} + v_{i-2,j} + v_{i,j+2} + v_{i,j-2} - 4v_{i,j} = 4h^2 f_{i,j}. \quad (4)$$

In this approach, the iteration process is carried out through a quarter of all the points. The other quarter of the points lying on the rotated grid are computed by (3) and the remaining half by (2), see (Othman 1998).

Among the three methods, the quarter-sweep iterative scheme has the fastest convergence rate. But for the mean computational error, the quarter-sweep method is less accurate than the full-sweep method, but better than the half-sweep method.

SUCCESSIVE AND ACCELERATED OVER-RELAXATIONS

David Young (1954) introduced the successive over-relaxation (SOR) method, which is popularly used to speed up the convergence of the Gauss-Seidel method for solving a linear system of equations. The formulation of the method is as follows:

$$u^{(k+1)} = (D - \omega L)^{-1} [(\omega U - (1 - \omega)D)u^{(k)} + \omega b], \quad (5)$$

where $A = D - L - U$, D -diagonal, L -negative lower triangular, U -negative upper triangular matrices, and ω is the relaxation factor. The choice of relaxation factor depends upon the properties of the coefficient matrix. ω can be calculated in practice by consecutively choosing values in range between 1 and 2 with some precision until the optimal value is obtained. Note that choosing $\omega = 1$ will result in the Gauss-Seidel method. However, there is a theoretical optimal value given by the formula:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2}},$$

where ρ is the spectral radius of the Jacobi iteration matrix. In our experiments, we use the experimental value of ω , which is generated during the process of iterations. The previous experiments (Abdullah 1991; Othman & Abdullah 1998; Ali & Chong 2007) have shown the experimental value of ω to be in good agreement with the theoretical value.

Hadjidimos (1978) devised the accelerated over-relaxation (AOR) method, which is based on the SOR method, but involved one more relaxation parameter to manage the convergence. The formulation of the AOR is as follows:

$$u^{(k+1)} = L_{r,\omega} + \omega (D - rL)^{-1} b, \\ L_{r,\omega} = (I - rD)^{-1} [(1 - \omega)I + (\omega - r)D^{-1}L + \omega D^{-1}U]. \quad (6)$$

The Jacobi, Gauss-Seidel, and SOR iterative methods are special cases of this method. Choosing $\omega = 1$ and $r = 0$ results in the Jacobi method, choosing $\omega = r = 1$ results in the Gauss-Seidel method, and choosing $\omega = r$ results in the SOR method. There is no general formula to determine the optimal values of ω and r , and the author states that r should be chosen not far from the value ω of the corresponding SOR method (Hadjidimos 1978).

Generally, the SOR method is implemented with all the three iterative schemes. But the AOR method has been shown to work well with the full-sweep and the half-sweep iterative schemes (Ali & Chong 2007). In this paper, we implement the AOR method with the quarter-sweep iterative scheme using the finite-difference and perform comparative experiments with the AOR full- and half-sweep methods. The AOR quarter-sweep algorithm is described in the next section.

THE AOR QUARTER-SWEEP ITERATIVE METHOD

In the quarter-sweep iterative scheme, the grid points in the domain Ω are divided into three types of points: \circ , \square , and \bullet (Figure 1).

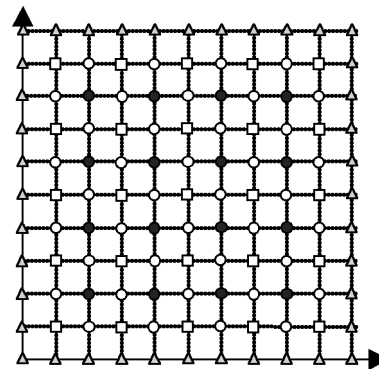


FIGURE 1. The representation of the domain in the quarter-sweep approach

Such a division requires the size of the grid n to be even. So the quarter-sweep method is suitable for an even n only.

The points denoted as \blacktriangle are boundary points; they are known due to the Dirichlet boundary conditions, so they are not iterated but used in the calculations. The iteration process is carried out only on the points of type \bullet . After achieving the convergence at these points, the values at the points of type \square and \circ are computed directly using (3) and (2), respectively. The algorithm is as follows:

1. Divide the grid points into the three types as in Figure 1. Compute the values of h^2 , $2h^2$, $4h^2$ and beforehand and assign them to variables H , I , and J .
2. For all the points of type \bullet , implement the accelerated over-relaxation:

$$v_{i,j} = r \left(\frac{v_{i-2,j}^{(k+1)} - v_{i-2,j}^{(k)} + v_{i,j-2}^{(k+1)} - v_{i,j-2}^{(k)}}{4} \right) + \omega \cdot \bar{v}_{i,j}^{(k+1)} + (1 - \omega) v_{i,j}^{(k)},$$

where

$$\tilde{v}_{i,j}^{(k+1)} = \frac{1}{4} \left(v_{i-2,j}^{(k+1)} + v_{i,j-2}^{(k+1)} + v_{i+2,j}^{(k)} + v_{i,j+2}^{(k)} - J \cdot f_{i,j} \right).$$

3. Check for the convergence. If the convergence is not achieved, go to the step 2. Otherwise, evaluate solutions for the other points by the following formulas:

$$v_{i,j} = \frac{1}{4} \left(v_{i-1,j-1} + v_{i+1,j-1} + v_{i+1,j+1} + v_{i-1,j+1} - I \cdot f_{i,j} \right)$$

for points of type \square and

$$v_{i,j} = \frac{1}{4} \left(v_{i-1,j} + v_{i,j-1} + v_{i+1,j} + v_{i,j+1} - H \cdot f_{i,j} \right)$$

for points of type \circ .

4. Stop.

THE COMPUTATIONAL COMPLEXITY

To estimate the time and the general performance of the algorithm, we can assess its computational complexity by examining the number of additions and multiplications needed to perform. Let $m = N - 1$, then m^2 is the number of internal mesh points. The values $H = h^2, I = 2h^2, J = 4h^2, r/2, \omega/4$ are computed beforehand and used in the iterations as constants; so we do not consider their evaluation when examining the number of operations.

For the full-sweep method, the standard formula should be iterated through all of the m^2 points. Taking the accelerated relaxation into consideration, each point requires 10 additions and 4 multiplications to be performed. Hence, for the full-sweep method, each iteration requires $10m^2$ additions and $4m^2$ multiplications.

Similarly, for the half-sweep method, each point requires 10 additions and 4 multiplications. But the number of iterative points is $m^2/2$ for an even m and $(m^2 - 1)/2 + 1$ for an odd m . Since we are comparing it with the quarter-sweep method, which requires the grid size n to be even, we have to consider the case when m is odd. Hence, each iteration requires approximately $5(m^2 + 1)$ additions and $2(m^2 + 1)$ multiplications. In addition, after the iteration process converges, the number of the remaining points (which are not iterated) is $(m^2 - 1)/2$, and the calculation of the values at these points requires $5(m^2 - 1)$ additions and $2(m^2 - 1)$ multiplications.

Finally, for the quarter-sweep method, the number of iterative points is $(m - 1)^2/4$ since m is odd. The resulting number of operations required is $5(m - 1)^2/2$ additions and $(m - 1)^2$ multiplications. The number of the points remaining is $(m + 1)^2/4$ points of type \square on the rotated grid, with $5(m + 1)^2/2$ additions and $(m + 1)^2$ multiplications overall, and $(m^2 - 1)/2$ points of type \circ on the standard grid, with $5(m^2 - 1)$ additions and $2(m^2 - 1)$ multiplications.

Taking the number of iterations k into account and simplifying the values obtained, we can summarise

the computational complexity of the three AOR methods in Table 1.

TABLE 1. The computational complexity for the three methods; k is the number of iterations

Method	Additions	Multiplications
Full-sweep	$10m^2 \cdot k$	$4m^2 \cdot k$
Half-sweep	$5(m^2 + 1)(k + 1) - 10$	$2(m^2 + 1)(k + 1) - 4$
Quarter-sweep	$\frac{5}{2}(m - 1)^2(k + 3) + 20m - 10$	$(m - 1)^2(k + 3) + 8m - 4$

From the table above, we can easily see that the number of the operations decreases approximately by half for the half-sweep scheme and by 75% for the quarter-sweep scheme. The question, which remains unclear, is the number of iterations. The number of iterations varies for different iterative schemes, and we hypothesize that it decreases, while changing the full-sweep scheme to the half-sweep and the quarter-sweep schemes, provided we choose the optimal strategy for each scheme. The next section shows our hypothesis to be true in practice.

NUMERICAL EXPERIMENTS AND RESULTS

To compare the AOR quarter-sweep method with the AOR full- and half-sweep methods, experiments were carried out with the following Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{xy}, \quad (x, y) \in \Omega = [0, 1] \times [0, 1]$$

with the Dirichlet boundary conditions: $u(x, y) = e^{xy}$, for $(x, y) \in \partial\Omega$. The three methods: full-, half-, and quarter-sweep AOR methods were used to solve the equation. The methods were carried out on the following mesh sizes: 26, 50, 74, 100. The convergence test was the maximum absolute error with the error tolerance $\varepsilon = 10^{-6}$.

All the three methods were implemented with different ordering strategies. ‘‘Chess-board’’ and ‘‘wave’’ strategies were implemented with the full- and quarter-sweep iterative schemes, whereas ‘‘horizontal-zebra-line’’ strategy was implemented with the half-sweep iterative scheme. The choice of the strategies is optimal for each iterative scheme, see (Othman & Abdullah 1998).

To find the parameters ω and r for each grid-size and method, we use the following algorithm:

1. Perform experiments taking the values of $r = \omega$ from the segment (1, 2) with precision 0.1 (i.e. 1.1, 1.2, ..., 1.9), as it is done in the SOR method.
2. Define the value of ω for which the number of iterations k is minimal in the previous step (it is the optimal value of ω with the precision 0.1).
3. Within the interval ± 0.1 from the value found in the step 2, define the optimal ω_{opt} with precision

0.01 by choosing consecutive values for which k is minimal; r is taken the same as ω .

4. Perform experiments using the value of ω_{opt} and choosing consecutive values of r with precision 0.01 within the interval ± 0.1 from the ω_{opt} .
5. Define the value r_{opt} for which k is minimal.

Table 2 shows the experimental results for the methods. FCB denotes the full-sweep method with the “chess-board” strategy. HHZL denotes the half-sweep method with the “horizontal-zebra-line” strategy. QCB refers to the quarter-sweep method with the “chess-board” strategy. The experiments were run on a Sun-Fire-v240 machine with one processor running.

Using the computational complexity formulas in the previous section, we can calculate the number of computations performed to solve the problem for these grid sizes. The values $m = 25, 49, 73, 99$ and k are taken from Table . Table summarises the numbers of addition and multiplication operations needed to solve the problem for the three methods, using the formulas in Table 1.

CONCLUSION

From the results presented in the previous section, it can be seen that the quarter-sweep AOR method shows superior convergence; the number of iterations is reduced comparing with the full- and half-sweep methods nearly by 50% and 25%, respectively. The computational effort of the quarter-sweep method is nearly 25% and 50% of the computational effort of the full- and half-sweep methods, respectively. Eventually, the AOR quarter-sweep method is approximately 8 times faster than the full-sweep method and 2–3 times faster than the half-sweep method.

As for the error of computations, the full-sweep method is still the best one among the three methods. And the error of the quarter-sweep method is better than the error of the half-sweep method. Improving the error of computations in the quarter-sweep AOR method is the subject for the future research. Another problem, which is to be solved, is the parallelisation of the algorithm with different strategies.

TABLE 2. The experimental results of the three methods.

N	Method	ω	r	k	Time (ms)	Error
26	FCB	1.78	1.77	60	8.01	6.26×10^{-06}
	HHZL	1.70	1.71	47	3.40	2.45×10^{-04}
	QCB	1.61	1.62	32	1.28	2.20×10^{-05}
50	FCB	1.88	1.89	123	66.29	3.10×10^{-06}
	HHZL	1.83	1.84	84	21.49	6.64×10^{-05}
	QCB	1.77	1.78	62	8.29	4.56×10^{-06}
74	FCB	1.92	1.91	162	185.22	4.59×10^{-06}
	HHZL	1.89	1.88	119	69.52	3.32×10^{-05}
	QCB	1.84	1.85	91	26.76	5.28×10^{-06}
100	FCB	1.92	1.94	217	561.06	8.99×10^{-06}
	HHZL	1.90	1.92	170	178.39	1.93×10^{-05}
	QCB	1.88	1.89	123	70.89	3.06×10^{-06}

TABLE 3. The number of computations for different grid sizes

N	Method	k	Multiplications	Additions	Time (ms)
26	FCB	60	375 000	150 000	8.01
	HHZL	47	150 230	60 092	3.40
	QCB	32	50 890	20 356	1.28
50	FCB	123	2 953 230	1 181 292	66.29
	HHZL	84	1 020 840	408 336	21.49
	QCB	62	375 370	150 148	8.29
74	FCB	162	8 632 980	3 453 192	185.22
	HHZL	119	3 197 990	1 279 196	69.52
	QCB	91	1 219 690	487 876	26.76
100	FCB	217	21 268 170	8 507 268	561.06
	HHZL	170	3 027 230	3 352 280	178.39
	QCB	123	3 027 230	1 210 892	70.89

ACKNOWLEDGEMENT

The research was supported by Fundamental Research Grant Scheme (FRGS), grant no. 02-01-07321FR under the Ministry of High Education of Malaysia.

REFERENCES

- Abdullah, R. 1991. The four point explicit decoupled group (EDG) method: A fast poisson solver. *Int. J. Comp. Math.* 38: 61-70.
- Ali, N.H.M. & Chong, L.S. 2007. Group accelerated over-relaxation methods on rotated grid. *Appl. Math. Comput.* 191: 533-542.
- Hadjidimos, A. 1978. Accelerated over-relaxation method. *Math. Comput.* 32: 149-157.
- Othman, M. & A. R. Abdullah. 1998. A new technique point iterative method for solving poisson equation on MIMD computer system. *Sains Malaysia* 27: 107-117.
- Othman, M. & A. R. Abdullah. 2000. An Efficient parallel quarter-sweep point iterative algorithm for solving poisson equation on SMP parallel computer. *Pertanika J. Sci & Technol.* 8(1): 161-174.
- Young, D. M. 1954. Iterative methods for solving partial difference equations of elliptic type. *Trans. Amer. Math. Soc.* 76: 92-111.
- Shukhrat I. Rakhimov*
Institute for Mathematical Research
Universiti Putra Malaysia
43400 UPM Serdang, Selangor D.E.
Malaysia
- Mohamed Othman
Dept of Communication Tech and Network
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 UPM Serdang, Selangor D.E.
Malaysia
- *Corresponding author; email: sh.rakhimov@gmail.com
- Received: 11 September 2008
Accepted: 12 March 2009