Applied Mathematical Sciences, Vol. 1, 2007, no. 10, 453 - 462

Monte Carlo Method for Finding the Solution of Dirichlet Partial Differential Equations

Behrouz Fathi Vajargah

Department of Mathematics
Guilan University, Rasht, Iran
fathi@guilan.ac.ir

Kianoush Fathi Vajargah

Department of Statistics
Tehran Islamic Azad University
North Branch,Tehran, Iran
fathi_kia10@yahoo.com

Abstract

In this paper we present a new Monte Carlo algorithm for finding the solution of Dirichlet partial differential equations (DPDE). Here, we show the solution of a given DPDE, can be found by a new efficient and accurate Monte Carlo algorithm. The Monte solution is obtained based on the inverse of A which is calculated based on inversion of B. Also, the solution of linear system for DPDE can be obtained efficiently by the algorithm presented in this paper. The given results by new algorithm significantly more accurate than the other numerical algorithms presented in the area of the Monte Carlo methods.

Mathematics Subject Classification: 34A30

Keywords: Markov Chain, Monte Carlo, Dirichlet, Partial Differential Equations, Poisson.

1 Introduction

Dirichlet problem is one of the most famous problems in the area of partial differential equations (PDE). The iterative numerical solution can be found in [2]. In this paper, we present the new accurate Monte Carlo solution which is more accurate than the results given by any Monte Carlo algorithms presented in [1] and [5-7]. It is well known that the Dirichlet PDE (DPDE) system is applied to find a partial differentiable equation over a given domain $D \subseteq R^n$ with boundary R, for $(x, y) \in D$ satisfy in

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

and $u(x, y) = g(x, y)$ for $(x, y) \in R$, where $g = g(x, y)$ is some prescribed function. Equation (1) with $f(x, y) \neq 0$, is called the Poisson equation, also with $f(x, y) = 0$, it is known as the Laplac equation. To obtain the methods of numerical computation, we replace the PDE in a given equation. With applying Taylor formula we have:

$$u(x+h, y) = u(x, y) + h\frac{\partial u(x, y)}{\partial x} + \frac{1}{2}h^2\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{1}{6}h^3\frac{\partial^3 u(x, y)}{\partial x^3} + ... \tag{2}$$

$$u(x-h, y) = u(x, y) - h\frac{\partial u(x, y)}{\partial x} + \frac{1}{2}h^2\frac{\partial^2 u(x, y)}{\partial x^2} - h^3\frac{1}{6}\frac{\partial^3 u(x, y)}{\partial x^3} + ... \tag{3}$$

Subtracting two recent equations (2)and (3) is shown that:

$$\frac{\partial u(x, y)}{\partial x} \approx \frac{1}{2h}[u(x+h, y) - u(x-h, y)].$$

Similarly, we have

$$\frac{\partial u(x, y)}{\partial y} \approx \frac{1}{2k}[u(x, y+k) - u(x, y-k)].$$

Adding the equations (2)and (3) and ignore the terms based on $h^4, h^5, ...$ we have:

$$u(x+h, y) + u(x-h, y) \approx 2u(x, y) + h^2\frac{\partial^2 u(x, y)}{\partial x^2}.$$

Then we obtain

$$\frac{\partial^2 u(x, y)}{\partial x^2} \approx \frac{1}{h^2}[u(x+h, y) - 2u(x, y) + u(x-h, y)]$$

Similarly,

$$\frac{\partial^2 u(x, y)}{\partial x^2} \approx \frac{1}{k^2}[u(x, y+k) - 2u(x, y) + u(x, y-k)]$$

If we substitute equations (2)and (3) into the Poisson equations (1) and considering h=k we obtain:

$$u(x+h, y) + u(x, y+h) + u(x-h, y) + u(x, y-h) - 4u(x, y) = h^2 f(x, y)$$

This is a difference equation corresponding to (3). Hence for the Laplace equation (2) the corresponding difference equation is:

$$u(x+h,y)+u(x,y+h)+u(x-h,y)+u(x,y-h)-4u(x,y)=0$$

h is called the mesh size. In the solving of Dirichlet problem in a region R we first choose h and introduce in R a grid consisting of equidistant horizontal and vertical straight lines of distance h. Their intersections are called mesh points or nodes. This easily concludes a linear system where the number of unknown variables is equal to the number of mesh points in R. Also, the coefficient matrix of linear system is sparse. In practice, this matrix is large, since for obtaining the solution of the linear systems we need high accuracy results. But it is clear that it will cause for large dimensions computations, storage problem of the entries of the given coefficient matrix. Then we have to use numerical methods. Here we explain how we can use a stochastic method, which is called Monte Carlo method, to find the solution of linear system caused in DPDE.
Let we have $R=\{(x,y)|a<x<b,c<y<d\}$, the first important things are selecting the numbers m and n and the increments of steps h and k with $h=\frac{(b-a)}{n}$ and $k=\frac{(d-c)}{m}$. Then we can consider a partition on the [a,b] to n equal parts with width h and another partition on [c,d] with m equal parts with width k so. Thus we have a grid with meshes on the pairs (x_i,y_j) where $x_i=a+ih$, $y_i=b+jk$ with $i=0,1,...,n, j=0,1,...,m$. For $i=0,1,...,n, j=0,1,...,m$ and boundary conditions: $u(x_0,y_j)=g(x_0,y_j)$, $u(x_n,y_j)=g(x_n,y_j)$, for all $j=0,1,...,m$; and $u(x_i,y_0)=g(x_i,y_0)$, $u(x_i,y_m)=g(x_i,y_m)$, for all $i=1,...,n-1$, also, it is easily can be proved that $i=1,...,n-1, j=1,...,m-1$ (the method considered in the following is concluded by solving of the difference equations, which is called the method of difference central)[2]:

$$2[(\frac{h}{k})^2+1]w_{i,j}-(w_{i+1,j}+w_{i-1,j})-(\frac{h}{k})^2(w_{i,j+1}+w_{i,j-1})=-h^2f(x_i,y_j) \quad (4)$$

and for $j=0,1,...,m:$

$$w_{0,j}=g(x_0,y_j),w_{n,j}=g(x_n,y_j)$$

$$w_{i,0}=g(x_i,y_0),w_{i,m}=g(x_i,y_m)$$

where $w_{i,j}$ is the estimated value of $u(x_i,y_j)$. With regards to equation (4) for all the points (x_i,y_j) where they are the corner of the boundary mesh, we have a linear system with $(n-1)\times(m-1)$ of difference equations and the number of unknown variables are $(n-1)\times(m-1)$ and its variables are $w_{i,j}$ to $u(x_i,y_j)$ for the interior points of grid. If we consider the rearrangement of the given linear system by the method of Varga [2], for $i = 1, 2, ..., m - 1$ and $j = 1, 2, ..., n - 1$ by setting:

$$l = i + (m - 1 - j)(n - 1)$$

$$w_i = w_{i,j} \text{ and } P_l = (x_i, y_j).$$

This arrangement of the equations (4), will provide a numbering of mesh points from the left to right and from up to down of the grid.

2 Generalizing of the Monte Carlo for Solving Linear Systems

We suppose

$$Bx = f, \tag{5}$$

where B is a $n \times n$ non-singular matrix and f is a given vector. For an identity matrix I, if we consider $A = I - B$, we can rewrite SLAE (5) as then we can convert (5) to its iterative form $x^{k+1} = Ax^k + f,$. In Monte Carlo calculations for SLAE, we use the maximum norm of matrix A given by $||A|| = max_i \sum_{j=1}^{n} |a_{ij}|$ then with

$$\rho(A) \leq ||A|| < 1 \tag{6}$$

(where $\rho(A)$ is the spectral radius of A) x^k tends to the unique solution $x = (I - A)^{-1} f$. The number of Markov chains is given by

$N \geq (\frac{0.6745}{\varepsilon} . \frac{||f||}{(1-||A||)})^2$, and the length of Markov chain by $T = k < \frac{log(\frac{\delta}{||f||})}{log||A||}$[1,8]. Generally, the required random selected elements of the matrix A are defined by its non-zero elements. These elements can be selected via the following Markov chain,

$$s_0 \longrightarrow s_1 \longrightarrow ... \longrightarrow s_k, \tag{7}$$

where s_i, $i = 1, 2, ..., k$ belongs to the state space $S = \{1, 2, ..., n\}$. Also, consider that for α, β,$p_0 = P(s_0 = \alpha)$ and $P(s_{j+1} = \beta | s_j = \alpha) = p_{\alpha\beta}$ which are the initial distribution and one step transition probability of Markov chain (7), respectively. Probabilities $p_{\alpha\beta}$ define a transition probability matrix $P = [p_{ij}]$ [3]. We say that the distribution $(p_1, ..., p_n)^t$ is acceptable for vector g, and the distribution $p_{\alpha\beta}$ is acceptable for A if:
$p_\alpha > 0$ where $g_\alpha \neq 0$, $p_\alpha \geq 0$ when $g_\alpha = 0$
and
$p_{\alpha\beta} > 0$ when $a_{\alpha\beta} \neq 0$, $p_{\alpha\beta} \geq 0$ and $a_{\alpha\beta} \geq 0$ where $a_{\alpha\beta} = 0$.
In the case of MC without absorption state we require $\sum_{\beta=1}^{n} p_{\alpha\beta} = 1$, for all $\alpha = 1, 2, ..., n$. We define:

$$W_0 = 1, W_j = W_{j-1} \frac{a_{s_{j-1}s_j}}{p_{s_{j-1}s_j}}. \tag{8}$$

Let $T_k[g] = \frac{g_{s_0}}{p_{s_0}} \sum_{j=0}^{k} W_j f_{s_j}$ then $E[T_k(g)] =< g, x^{k+1} >$ and with above condition (6) we have $\lim_{k\to\infty} E[T_k(g)] =< g, x >$[7]. Considering N paths $s_0^m \to s_0^m \to ... \to s_k^m$ $m = 1, 2, ..., N$on the coefficient matrix (even the matrix should be inverted) we have the Monte Carlo estimated solution by: $\widehat{\Theta} = \frac{1}{N} \sum_{s=1}^{N} T_k^{(s)}(g) \approx< g, x^{k+1} >$ and if we consider $g = (\underbrace{0, 0, ..., 0, 1}_{i}, 0, ..., 0)^t$, then we have $\widehat{\Theta} = \frac{1}{N} \sum_{s=1}^{N} T_k^{(s)}(g) \approx x_i$, now, we consider a linear system that we transfer (5) to the form $x^{(k+1)} = Ax^{(k)} + f$, the condition (6) is not valid. Then the Monte Carlo solution cannot be converged to the unique exact solution of the linear system (5). The way to find the solution of linear system (5) is that we transfer (5) to an equivalent linear system such as

$$x = Tx + h \tag{9}$$

where $\rho(T) \leq ||T|| < 1$ and the solution of (9) is the same as solution of $x = Ax + f$. Here, we explain that how we can reach to the system (9).

3 Monte Carlo Method for Matrix Inversion

Let us assume that all the conditions for MC methods for the solution of SLAE, which we used in section 2 are valid here too. In previous section, we discussed the solution of SLAE by MC methods, but we aim to obtain the inverse of an arbitrary non-singular matrix by MC. Thus, we require a review of the general MC idea to find the inverse matrix by Monte Carlo method.

3.1 Naive Monte Carlo Method For MI

Consider a SLAE (5). Under condition (6) and with $x^0 = \overrightarrow{0}$ and for $k = 0, 1, 2, ..., T^0 = I$,

$$x^{(k+1)} = \sum_{m=0}^{k} A^m f$$

converges for any non-singular matrix A and

$$\lim_{k\to\infty} x^k = \lim_{k\to\infty} \sum_{m=0}^{k} A^m f = (I - A)^{-1} f = x.$$

(where $B^{-1} = [b_{ij}^{-1}]_{i,j=1,2,...,n} = I + A + ... + A^m + ...$, and the i^{th} coordinate of x is $x_i = \sum_{k=1}^{n} b_{ik}^{-1} f$). By setting $f_k = e_k = (\underbrace{0, ..., 0, 1}_{k}, 0, ..., 0)^t$ we have $x_i = a_{ij}^{-1}$. And the corresponding unbiased estimator is:

$$T_l(b_{ik}^{-1}) = \sum_{m|i_m=r} W_m.$$

Therefore, the Monte Carlo inverse for the entries of matrix B, for $k = 1, 2, ..., n$ is given by:

$$b_{ik}^{-1} \approx \frac{1}{N} \sum_{s=1}^{N} [\sum_{m|i_m=r} W_m^{(s)}].$$

where $(m|i_m = r)$ means that summation only obtain for all $i_m = r$ will be obtained. The number of Markov Chains in the case of matrix inversion can be obtained by and $N \geq (\frac{0.6745}{\varepsilon} . \frac{1}{(1-||A||)})^2$ and $T = k \leq \frac{log\delta}{log||A||}$ [1].
The solution of a linear system and the elements of inverse matrix of a non singular matrix given by discussed MC method is merely under sufficient condition of converging (6). The main important aim of this paper is considering the case that the condition (6) is not valid.

3.2 MC Method for finding the MI of a general nonsingular Matrix

Suppose that B is an arbitrary non-singular matrix and we are aiming to find an accurate inversion by Monte Carlo methods. It is clear that we can efficiently obtain the inverse of a diagonally dominant matrix [1,5,6]. The main aim of this research is to obtain the inverse of matrix B based on a split with the main part being a diagonally dominant matrix. The question is, how this transformation can be done for all types of matrices? We note that if the following concepts do not depend on diagonally dominant matrices, then the matrix B below can be any arbitrary matrix. In [1,9] it has been obtained the inverse and the solution of the systems based on a diagonally dominant coefficient matrix. Thus if we follow these concepts for general cases, via the following split we reach to obtain the inversion of a diagonally dominant matrix. It means that if we wish to obtain the inverse of a general matrix by Monte Carlo method, we first obtain the inverse of a diagonally dominant matrix as explained in the following algorithm. Now, we consider a given non-singular matrix B, and if we consider A to be a non-singular matrix then we can perform the following split on B:

if S is a diagonal matrix such that $S = \sum_{i=1}^{n} S_i$ where S_i is a matrix with rank one and all of its elements are zero with only one non-zero element in position (i, i) . If we suppose $B_0 = B$, $B_n = B + S$ therefore, the inversion of B_n is:

$$B_i^{-1} = B_{i+1}^{-1} + \frac{B_{i+1}^{-1} S_{i+1} B_{i+1}^{-1}}{1 - trace(S_{i+1} B_{i+1}^{-1})}, i = n-1, n-2, ..., 1, 0 \qquad (10)$$

In this case we have applied the fact that we can write the matrix . Now, we come back to our main aim which is obtaining the inversion of an arbitrary non-singular matrix B. First, we consider the split $B = A - S$ such that A is a diagonally dominant matrix based on matrix B. A simple rule of making A as a diagonally dominant matrix created from B is that we add $||B||$ (or a number bigger than norm of B) to all the diagonal elements of B, the off diagonal elements remaining unaffected. In this case, the matrix S is a diagonal matrix with all diagonal elements equal to the norm of B (or another fixed number bigger than norm of A, depending on matrix A) and all its off diagonal elements are zero. First, we can obtain the inverse of A which is a diagonally dominant matrix and secondly we use the above algorithm given in (13). It is clear that if we consider A as a diagonally dominant matrix with diagonal elements added to a number greater than $||A||$, in the most of the cases, the rate of convergence of the algorithm is fast and the computation becomes more accurate, also. This means we can control the norm of matrix. This is the flexibility of this method.

Algorithm

```
(1) Select $N$, Number of Markov chains, $T = k$, the length of Markov chains, $\epsilon$ and $\delta$.
(2) Read the Matrix $B_n$ from the file.
    (2.1) Split $B_n = M-K$, where $M$ is a diagonal matrix with $m_{ii} = b_{ii}, i = 1, 2, ..., n$.
    (2.2) Compute $C = M^{-1}K$.
    (2.3) Compute $||C||$ and the Number of Markov chains $N = (\frac{0.6745}{\varepsilon}.\frac{1}{(1-||C||)})^2$.
(3) For i=1 to n;
    (3.1) For j=1 to N;
        (3.1.1) Set $t_k = 0$(stopping rule), $W_0 = 1, SUM[i] = 0$.
        (3.1.2) Generate a random number and set as $i_0^{(s)} \to i_1^{(s)} \to ... \to i_k^{(s)}$,
                $s = 1, 2, ..., N$..
        (3.1.3) If $C[i_{m-1}^{(s)]} i_m^{(s)}]! = 0$.
                Loop:
            (3.1.3.1) Compute $W_j^{(s)} = W_{j-1}^{(s)} \frac{C_{i_{m-1}^{(s)} i_m^{(s)}}}{p_{i_{m-1}^{(s)} i_m^{(s)}}}$.
            (3.1.3.2) Set $i_{m-1}^{(s)} = i_m^{(s)}$ and $SUM[i] = SUM[i] + W_j^{(s)}$.
            (3.1.3.3) If $|W_j| < \delta, t_k = t_k + 1$.
            (3.1.3.4) If $t_k \geq n$
                      End of Loop.
        (3.1.4) End If.
```

(3.1.5) **Else** go to step 3.1.2.

(3.2) **End of loop j.**

(3.3) **Compute** the average of results.

(4) **End of loop i.**

(5) **Obtain** the matrix $H = (I - C)^{-1}$, then the expecting approximate matrix $Q_n = H \times M^{-1}$.

(6) **Compute** $Q_i = Q_{i+1} + \frac{Q_{i+1}K_{i+1}Q_{i+1}}{1-trace(K_{i+1}Q_{i+1})}, i = n, n-1, ..., 1, 0$, where $K_n = B_n - diag(A)$.

(7) **Set** $B_n^{-1} = Q_n$.

(8) **End of Algorithm.**

Example : Consider a Poisson equation given by $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{xy}$ in the region R with boundaries:

$$u(0, y) = 1, u(2, y) = e^{2y}, 0 \le y \le 1$$

$$u(x, 0) = 1, u(x, 1) = e^{x}, 0 \le y \le 2$$

if we consider $h = \frac{2}{3}, k = \frac{1}{3}$ we have a grid with 3 horizontal and 3 vertical straight lines the corresponding linear algebraic equations are:

$$10w_1 - w_2 - 4w_3 = 1 + 4e^{\frac{2}{3}} - \frac{32}{81}e^{\frac{4}{9}}$$

$$-w_1 + 10w_2 - 4w_4 = 5e^{\frac{4}{3}} - \frac{80}{81}e^{\frac{8}{9}}$$

$$-4w_1 + 10w_3 - w_4 = 5 - \frac{20}{81}e^{\frac{2}{9}}$$

$$4w_2 - w_3 + 10w_4 = 4 + e^{\frac{2}{3}} - \frac{68}{81}e^{\frac{4}{9}}$$

with exact solution $w = e^{xy}$, then we have:

Node at (x, y)	$(\frac{2}{3}, \frac{2}{3})$	$(\frac{4}{3}, \frac{2}{3})$	$(\frac{2}{3}, \frac{1}{3})$	$(\frac{4}{3}, \frac{2}{3})$
Exact Solution w_i	$e^{\frac{4}{9}} = 1.55962$	$e^{\frac{8}{9}} = 2.43242$	$e^{\frac{2}{9}} = 1.24884$	$e^{\frac{4}{9}} = 1.55962$
MC Solution $\widehat{w_i}$	1.55961	2.43243	1.24885	1.55963
$\vert\frac{w_i - \widehat{w_i}}{\widehat{w_i}}\vert$	1.1×10^{-5}	$1.2\ \times10^{-5}$	1.2×10^{-5}	1.1×10^{-5}

Table1 : Comparing the MC solution and exact solution.

Curtiss[4] has shown that the iterative methods such as Jacobi, SOR also Monte Carlo method can be used as an alternative of classical method. But, as he proved numerically and in theory also, the Monte Carlo method can be employed for large linear systems. In fact, the efficiency of Monte Carlo method increasingly will improve. Then the Monte Carlo for large linear systems especially for linear systems with sparse coefficient matrix using parallel computations will be advised [6,7,8].

4 Conclusion

The solution of DPDE's can be solved by classical, iterative and Monte Carlo methods. Solution of Monte Carlo method efficiently can be obtained by the algorithm described in this paper. The most suitable way for sparse matrices computations is Parallel Monte Carlo method. The details of the parallel Monte Carlo Computations can be found in [8]. In solving the DPDE we use an especial split as it has been explained here and the MC results shown high accuracy in computations. If we increase the number of employed Markov chains in MC computations the accuracy of MC computations will increase, also. Then we can make a balance between accuracy and number of Markov chains.

References

1. V.N. Alexandrov, Efficient parallel Monte Carlo methods for matrix computation, Mathematics and computers in simulation Elsevier 47 (1998) 113-122, Netherland.

2. R.L. Burden, J.D. Faires and A.C. Reynolds, *Numerical Analysis*, (1981) Massachusetts, USA.

3. E. Cinlar , *Introduction to Stochastic Processes*, Prentice-hall, Englewood Cliffs, New Jersey, (1975).

4. J.H. Curtiss, A theoretical comparison of the efficiencies of two classical methods and a Monte Carlo method for computing one component of the solution of a set of Linear Algebraic Equations, in: Proceedings of the Symposium on Monte Carlo methods, Wiley, New York, (1956), pp. 191-233.

5. I. Dimov and A.N. Karaivanova, Iterative Monte Carlo algorithms for linear algebra problems, First Workshop on Numerical Analysis and Applications, Rousse, Bulgaria, June 24-27, (1996), in : Numerical Analysis

and Its Applications, Springer Lecture Notes in Computer Science, Ser. (1996), pp. 150-160.

6. I.T. Dimov, T.T. Dimov and T.V. Grov, A new iterative Monte Carlo approach for inverse matrix problem, Journal of Computational and Applied Mathematics 92 (1998) 15-35.

7. I.T. Dimov and V.N. Alexandrov, A new highly convergent Monte Carlo method for matrix computations, Mathematics and Computers in Simulation 47 (1998) 165-181, Bulgaria Academy of science.

8. B. Fathi Vajargah and K. Fathi Vajargah, Parallel Monte Carlo methods for solving linear systems with compressed data, International Journal of Applied mathematics (2005), Sofia, Bulgaria.

9. G.M. Megson, V. Alexandrov and I. Dimov, Systolic matrix inversion using Monte Carlo method, J. Parallel algorithms Appl. 3 (1994) 311-330.

Received: October 30, 2006