特殊構造を有する大規模連立一次方程式の一解法とその応用

鈴木　誠道† 　仇　莉†

本論文では特殊構造を有する連立一次方程式の直接解法に対する一つのアプローチを提案する。これは筆者らの自動倉庫システムや生産ラインなどのモデル化に現れた大規模連立一次方程式の解を効率的に求める過程からヒントを得ている．大規模システムは多くの場合特殊構造をもつ．上のシステムも特殊構造を有しており、その細部構造までフルに利用した解法を考案し効率的な解析を行つた．そのアイデアは、連立方程式のいくつかの変数の値を既知として、他の変数をこれらの変数を用いて表すことである．これは元の連立方程式を実質的により小規模の連立方程式群に分割することである。この方式がより一般の特殊構造をもつ連立一次方程式にも有効に適用可能なことを例を挙げて示す.

A Unified Approach for Solution of a Large System of Linear Equations with Special Structures and its Applications

Shigemichi Suzuki† and Qiu Li†

We present here an approach for direct solution of a system of linear equations with structures. The approach is motivated by our analysis of a large scale system of linear equations obtained in modeling systems of automatic warehousing and production lines. Large scale systems often have special structures. The systems mentioned above have special structures. By exploiting the structure we have developed efficient methods of analysis. The idea is to assume the values of certain variables known and express the rest of variables in terms of assumed variables. The process is, in effect, to decompose the original equations into a number of smaller scale systems. We will show that the approach can be applied to a wider class of problems.

1. Introduction

We propose here direct solution methods for a system of linear equations with special structures. The work is motivated by our research on queuing-systems analysis of serial production lines with unreliable machines and intermediate buffers[1]. The system of linear equations derived from the analysis has a special structure[2),3)]. We devised a method of solution for such a system which is some generality to be applicable to other types of problems such as the Dirichlet problem of discrete Poisson's equation. We will show that the present approach performs better than conventional ones[4)~7)] by two examples from quite distinct origins.

2. Basic Idea

Let A be a nonsingular square matrix of the order n and consider a system of linear equations $Ax = b$. It can be easily proved that the equation can be transformed to the following form by interchanging rows (equations) and columns (variables) of A and of elements of b such that super diagonal matrices $A_{i,i+1}(i = 1, 2, \cdots, k)$ are nonsingular:

$$\begin{bmatrix} A_{11} & A_{12} & 0\cdots & 0 \\ A_{21} & A_{22} & A_{23} & 0\ \dot{\cdot}.\ 0 \\ \vdots & \vdots & \ddots & A_{(k-1),k} \\ A_{k1} & A_{k2} & \cdots & A_{kk} \end{bmatrix} x = b. \qquad (1)$$

The above system of equations has k blocks, the solution vector x and the right-hand-side vector b which consist of k subvectors $x^{(i)}$ and $b^{(i)}(i = 1, 2, \cdots, k)$. We will try to find or recognize a "*good*" transformation in the sense that it can help to reduce the overall computational complexity involved in solving the original linear equations compared with conventional methods.

Assuming that $x^{(1)}$ is known, the other solution subvectors $x^{(i)}(i = 2, 3, \cdots, k)$ can be obtained as

$$x^{(i)} = A^{-1}_{i-1,i}(b^{(i-1)} - \sum_{j=1}^{i-1} A_{i-1,j}x^{(j)}), \qquad (2)$$
$$(i = 2, 3, \cdots, k).$$

The solution subvectors thus obtained can be expressed in terms of $x^{(1)}$ as

$$x^{(i)} = c_i + F_i x^{(1)} \qquad (i = 1, 2, \cdots, k), \qquad (3)$$

where c_i and F_i are defined recursively as

† 千葉工業大学大学院,Graduate School,Chiba Institute of Technology

$$c_1 = 0, \qquad F_1 = I, \tag{4}$$

$$c_i = A_{i-1,i}^{-1}(b^{(i-1)} - \sum_{j=1}^{i-1} A_{i-1,j}c_j), \tag{5}$$

$$F_i = -A_{i-1,i}^{-1}(\sum_{j=1}^{i-1} A_{i-1,j}F_j), \tag{6}$$

$(i = 2, 3, \cdots, k)$.

The values of the elements of $x^{(1)}$ are obtained by substituting the expression for subvectors $x^{(i)}(i = 2, 3, \cdots, k)$ in (3) into the k-th block of the transformed system and solving the derived system

$$\sum_{j=1}^{k} A_{k,j}F_j x^{(1)} = b^{(k)} - \sum_{j=1}^{k} A_{k,j}c_j. \tag{7}$$

The solution procedure is valid since the matrix A and the submatrices $A_{i,i+1}(i = 1, 2, \cdots, k-1)$ are nonsingular.

Whether applications of the procedure are effective or not in solving systems of linear equations is heavily dependent on the special structures of the systems. We will present two examples in the following sections to show how the proposed method can be effectively applied.

3. Applications to the Dirichlet Problem of Discrete Poisson's Equation

Consider a five-point finite difference approximation to the problem and partition the region into $(l+1) \times (m+1)$ squares. Let $u_{i,j}$ be the value of u at (i,j)-grid point, then the finite-difference approximation of the Dirichlet problem is described by the following system of linear equations:

$$\begin{bmatrix} -B & I & 0 & \cdots & 0 \\ I & -B & I & 0 & 0 \\ \cdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -B & I \\ 0 & 0 & \cdots & I & -B \end{bmatrix} u = f, \tag{8}$$

where I is a unit matrix of order l and B is an $l \times l$ matrix shown as follows:

$$b_{i,i} = 4, b_{i,j} = -1(i = 1, 2, \cdots l, |i-j| = 1), \tag{9}$$

and the solution $u^T = (u^{(1)}, u^{(2)}, \cdot, u^{(m)})$ and the right-hand-side vector $f^T = (f^{(1)}, f^{(2)}, \cdot, f^{(m)})$ are defined by

$$(u^{(j)})^T = (u_{1,j}, u_{2,j}, \cdots, u_{l,j}) \quad (j = 1, 2, \cdots, m),$$

$$(f^{(j)})^T = (f_{1,j}, f_{2,j}, \cdots, f_{l,j}) \quad (j = 1, 2, \cdots, m),$$

where each $f_{i,j}$ is evaluated from the boundary values associated with the grid point (i,j).

3.1 Computational procedure

Assume that the solution subvector $u^{(1)}$ is known. Then the rest of the solution subvectors can be obtained as follows:

$$\begin{cases} u^{(2)} = f^{(1)} + Bu^{(1)}, \\ u^{(j)} = f^{(j-1)} + Bu^{(j-1)} - u^{(j-2)} \\ (j = 3, 4, \cdots, m) \end{cases} \tag{10}$$

The equation for $u^{(1)}$ can be derived from the last-block equation of Equation(8) by substituting $u^{(j)}(j = m-1, m)$ expressed as functions of $u^{(1)}$ in it (note here there are only two subvectors in the last-block equation). Observe here that $u^{(j)}(j = 2, 3, \cdots, m)$ can be expressed in terms of $u^{(1)}$ as

$$u^{(j)} = p_j + Q_{j-1}u^{(1)}(j = 1, 2, \cdots, m), \tag{11}$$

where $(l \times 1)$ vectors p_j and $(l \times l)$ matrices Q_j are obtained recursively as

$$\begin{cases} p_1 = 0, \qquad Q_1 = I, \\ p_2 = f^{(1)}, \qquad Q_2 = B, \\ p_j = f^{(j-1)} + Bp_{j-1} - p_{j-2}, \\ Q_{j-1} = BQ_{j-2} - Q_{j-3} \\ (j = 3, 4, \cdots, m+1) \end{cases} \tag{12}$$

To facilitate following discussions we will introduce a series of polynomials $S_j(v)$ $(j = 2, 3, \cdots)$ with a variable v by recurrence relations:

$$\begin{cases} S_0(v) = 1, \\ S_1(v) = v, \\ S_j(v) = vS_{j-1}(v) - S_{j-2}(v) \\ (j = 2, 3, \cdots). \end{cases} \tag{13}$$

Using polynomials S_j $(j = 0, 1, \cdots)$, $u^{(j)}$ can be expressed as:

$$u^{(j)} = \sum_{i=1}^{j-1} S_{j-i}(B)f^{(i)} + S_{j-1}(B)u^{(1)} \tag{14}$$

$(j = 2, 3, \cdots, m)$.

Using Equations (12) and (14) we can derive the system of linear equations for $u^{(1)}$ as:

$$-S_m(B)u^{(1)} = p_{m+1}. \tag{15}$$

We wish to preserve the sparsity of matrices involved in the computational procedure as much as possible. For this purpose we first observe that the matrix B can be transformed to a diagonal matrix D by an orthogonal transformation $D = V^TBV$, where the diagonal elements $d_1, d_2, \cdots, d_l$ of D are eigenvalues of B and the i-th column vector of V is the normalized eigenvector corresponding to the eigenvalue d_i. With this diagonalization property premultiplying both sides of Equation (15) by V^T yields

$$-S_m(D)V^Tu^{(1)} = V^Tp_{m+1}. \tag{16}$$

The solution for $u^{(1)}$ can be obtained as

$$u^{(1)} = -V(S_m(D))^{-1}V^Tp_{m+1}. \tag{17}$$

The solution process for Equation (8) will be complete after we substitute the expression (17) to the first equation in (10) to compute $u^{(2)}$ and proceed to evaluate the rest of $u^{(j)}$'s by the second equation in (10).

At this point we will note that the eigenvalues and eigenvectors can be explicitly given by

$$d_i = 4 - 2cos(\frac{i\pi}{(l+1)}) \qquad (i = 1, 2, \cdots, l), \quad (18)$$

$$v_{j,i} = r_i sin(\frac{ij\pi}{(l+1)}) \qquad (i, j = 1, 2, \cdots, l), \quad (19)$$

where $v_{j,i}$ is the j-th element of the i-th column eigenvector of B, and r_i is the normalization constant of the vector.

We are now in the position to clarify the whole computational procedure in sequence:

- Compute eigenvalues $d_1, d_2, \cdot, d_l$ and eigenvectors V of B by Equations(18) and (19).
- Compute $S_m(D)^{-1}$ by recursion.
- Compute p_{m+1} recursively by Equation(12).
- Compute $u^{(1)}$ by Equation (17).
- Compute $u^{(j)} (j = 2, 3, \cdots, m)$ by Equation (10).

4. Applications to Equilibrium-State Equations of Queuing Systems

4.1 Model

The model is concerned with a serial production line with unreliable machines and having intermediate buffers with finite capacities. On assumptions made about arrivals of work pieces at the production line, service time, time to failure, repair time at each machine, and some other operating rules of the line, the the system can be modeled as a Markov process.

Let n be the number of machines in a production line, $M_i + 1$ be the capacity of intermediate buffer B_i including capacity one of machine $(i+1)$ $(i = 1, 2, \cdots, n-1)$. Let $N_n(M_1, M_2, \cdots, M_{n-1})$ be the total number of the system states of the production line, then it will be expressed recursively as:

$$\begin{aligned} & N_n(M_1, M_2, \cdots, M_{n-1}) \\ = \; & 2(M_1 + 1)N_{n-1}(M_2, M_3, \cdots, M_{n-1}) + \\ & + N_{n-2}(M_3, M_4, \cdots, M_{n-1}) \\ & (n = 2, 3, \cdots), \end{aligned}$$

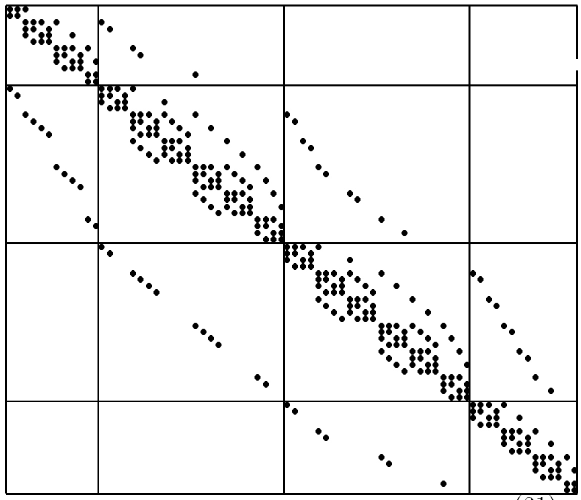
where $N_1(\bullet) = 2, N_0(\bullet) = 0$.

The number of system states blows up as n increases. The system of equilibrium equations for the Markov process will become as

$$xQ = 0, \qquad xe = 1, \qquad (20)$$

where Q is a matrix denoting the state-transition rates, x is a row vector of the steady state probabilities and e is a columnn vector with each of its elements being 1.

We will now take an example. The example is for a production line with 3 machines and 2 intermediate buffers having capacities 2 for each of them. There are 74 states in the system and the pattern of nonzero elements of the matrix Q will be as shown in (21).



(21)

As was pointed out in Section 1 a transformation of Equation (20) for Q as shown in (21) is not unique. A strategy here is to seek transformations which reduce the over-all computational complexity involved for solving the original system of equations. The computation includes construction of the transformation and solution of the transformed system. While the former may be combinatorial in nature and this may cause difficulties, the latter can be carried out without such difficulties and is easier than the former. At present we will contented with finding "good" transformations not for general type of equations but for specific type of equations. Then the construction of "good" transformations will be much easier, even trivial.

Now returning to transformation of Q in (21), we can easily find good transformations. One of such transformations which may be the simplest is to take the first 12 variables corresponding to the first 12 rows of Q as $x^{(1)}$ and interchange the columns of Q. The result of the transformation with interchanging of columns (no interchange of rows necessary in this case)yields matrix $\tilde{Q}$ shown in (22).

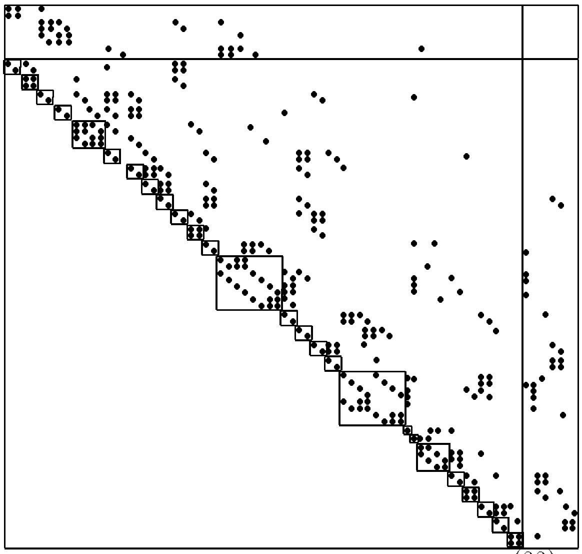
The solution of the transformed system can be obtained by the procedure described in Section 1 with much less computation time than solving the original system.

5. Issues of Computational Complexity

5.1 The Dirichlet problem

To simplify the evaluation of computational complexity for solution of a system of linear equations, we will take a model problem with $n \times n$ interior points. There exists n^2 equations in (8) for the model problem. There are five items to evaluate as pointed out in Section 2. The asymptotic number of operations required for each item will be:

(1) $n^2/2$, (2) n^2, (3) $3n^2$,



(22)

(4) Computation of $u^{(1)}$ by Equation (17):
$V^T p_{m+1}$ can be evaluated with $n \log n$ operations employing FFT, premultipying this by $(S_n(D))^{-1}$ requires n operations since $(S_n(D))^{-1}$ has been obtained in (2) and finally premultipying this by $-V$ to obtain $u^{(1)}$ is carried out with $n \log n$ operations employing FFT. The total number of operations needed here is $2n \log_2 n$.
(5) Computation of $u^{(j)} (j = 2, 3, \cdots, m)$ by Equation (10):
This requires $3n^2$ operations since each row of B has at most three elements.

The total number of operations for the discrete Poisson equation on a square with $n \times n$ interior grid points is now evaluated as $7.5n^2$ asymptotically ignoring $2n \log_2 n$ operations required in (4) above.

5.2 Basic scheme

To simplify evaluation of computational complexity we will take an example problem of Equation (1) with each of k subvectors $x^{(i)}$ having l variable. The total number of variables n in the system is then $n = kl$. Now let C_1,C_2, C_3, and C_4 be the numbers of operations required for computation of each of equations (3),(5),(6), and (7). Let ρ be the density of nonzero elements in the matrices $A_{i,j} (1 \le i \le j - 1 \le n - 1)$. Then the total number of operations $C = C_1 + C_2 + C_3 + C_4$ will be approximately

$$C = (\frac{4k}{3} + \frac{\rho k^2}{2}) l^3.$$

If we solve the system of equations under consideration by LU decompositons without making use of the structure and the sparsity of the matrix, then the number of operations required C_{LU} will be approximately

$$C_{LU} = \frac{(kl)^3}{3} + (kl)^2.$$

Therefore

$$\frac{C}{C_{LU}} = \frac{8 + k\rho}{2k^2}.$$

This implies that for a problem with $k = 20$ and $\rho = 0.1$, $\frac{C}{C_{LU}} = 0.0175$ and a big reduction of computational complexity is expected in this case.

6. Conclusions

A general scheme of solving a system of linear equations with special structures is proposed and applied to Dirichlet problms of the discrete Poisson's equation in a rectangle and a system of equilibrium equations of a queuing system. In the former example the number of operations required for the Dirichlet problem with $n \times n$ interior points in a square is proved to be $7.5n^2$ asymptotically compared with the estimates of $11.5n^2$ of the marching methods which is the fastest ever proposed[4]. The applications to the latter example are quite effective compared with conventional methos.

Applications to other types of problems are now under way.

参 考 文 献

1) Qiu, L. and Suzuki,S.: *Solution Method for Production Line with Machine Failures and Finite Intermediate Buffers*, Submitted to Transactions of Japanese Society of Mechanical Engineers.
2) Suzuki, S. and Qiu, L.: *A method for a serial production line with machine failures*, Abstract INFORMS'99, October 1999, Philadelphia.
3) Suzuki, S. and Qiu, L.: *Exact and approximate methods for a production line with machine failures and intermediate buffers*, Abstract EURO XVII, July 2000, Budapest.
4) Bank, R.E. and Rose, D.J.:*An $O(n^2)$ method for solving constant coefficient boundary value problems in two dimensions*, SIAM J. Numer. Anal., **12**(1975), pp529-540.
5) Hockney, R.W.: *A fast direct solution of Poisson's equation using Fourier Analysis* , J. Assoc. Comput. Mach.,**8**(1965), pp.95-113.
6) Buzbee, B.L., Golub, G.H. and Nielson, C.W.:*On direct methods for solving Poisson's equations*, SIAM J. Numer. Anal.,**8**(1970), pp.627-656.
7) Hockney, R.W.: *The potential calculation and some applications*, Methods of Computational Physics, Vol.9, B.Adler, S.Fernback and M.Rotenberg, eds., Academic Press, New York and London(1969), pp.135-211.