The Application Strategy of Direct Solution Methodology to Matrix Equations in Hydraulic Solver Package

Young W. Na [a*], C.E. Park, Sang Y. Lee [a]
[a] *KOPEC, Safety Analyses Dept, 150 Deokjin-dong, Yuseong-gu, Daejeon 305-353, Korea*
[*] *Corresponding author: ywna@kopec.co.kr*

1. Introduction

As a part of the Ministry of Knowledge Economy (MKE) project, "Development of safety analysis codes for nuclear power plants", KOPEC has been developing the hydraulic solver pilot code package applicable to the safety analyses of nuclear power plants (NPP's).

The matrices of the hydraulic solver are usually sparse matrices.

It is well known that a direct solution method works well for a matrix system with 500 unknowns or less.

The facts that the system of matrix is a sparse one and that the position indices of non-zero elements are the same in subsequent solutions present more merit of applying direct solution methodology.

In this project, one of the typical direct matrix solver packages MA48 has been selected as matrix solver [1,2] for the application to hydraulic solver.

The selection was based on the results of test applications of its former version MA28 [3] and on the reasonably reliable performance experience of MA18 [4] applied to RELAP computer code.

2. Methods and Results

In this section the direct solution methodology, error bounds and the application strategies are described.

2.1 The Upper Bounds of Decomposition Errors

In solving a set of matrix equation **Ax**=**b** by applying the direct solution method, as the decomposition process continues, the error may buildup. This error buildup can be described as:

$$\mathbf{A}+\mathbf{E} = \mathbf{LU} \qquad (1)$$

where

A = the original matrix
x = the unknown vector
b = the right-hand-side (RHS)/ source vector
E = the error matrix of decomposition
L = the lower triangular matrix
U = the upper triangular matrix.

Rewriting this for the elements,

$$a_{ij}^{(k)} = a_{ij} - \sum_{m=1}^{k} l_{im}\, u_{mj} \quad k<i<n, \quad k<j<n. \qquad (2)$$

Using Holder's generalization of Schwarz's inequality, the upper bound on the largest element encountered during the decomposition, Reid's inequality relationship [5] can be obtained as follows:

$$|e_{ij}| \leq 3.01\ \rho\ \varepsilon\ m_{ij} \qquad (3)$$

where

ρ = the largest element modulus of the matrices encountered in the elimination
ε = the machine precision
m_{ij} = the number of operations on position (i,j) during the decomposition
e_{ij} = the element of E in position (i,j).

2.2 The Relative Error Bounds of Decomposition

Dividing both sides of the inequality relationship (3) by $|a_{ii}|$ to obtain the relative error bound,

$$|e_{ij}|/|a_{ii}| \leq 3.01\ [\rho/|a_{ii}|]\ \varepsilon\ m_{ij}. \qquad (4)$$

2.3 Residual Refinement

J.K. Reid pointed out that the satisfactory bound does not guarantee an accurate solution. This problem arises due to the conditioning of the problem. A good accuracy can be obtained by employing the iterative refinement [6]:

$$\mathbf{LU}\ \delta^{(k)} = b - A\, x^{(k)} \qquad (5)$$
$$\mathbf{x}^{(k+1)} = x^{(k)} + \delta^{(k)} \qquad k = 0, 1, \ldots$$

commencing with $\mathbf{x}^{(0)} = 0$.

Test results of the routines developed by Marlow and Reid have shown that the iteration could almost always be terminated at k = 2 or 3, usually 2.

As a conclusion, Reid recommended that this iteration may also be used to avoid a fresh decomposition in case where the modulus is moderately large. And that, if however, the modulus is so large that the process would not converge, then there is no alternative but to compute a fresh pivot sequence taking account of the numerical values of the matrix coefficients.

2.4 Machine-dependant Calculation Error and Strategy of Applying the Direct Solution Method

From the inequality (4), to keep the maximum value of relative error within a given value, or the right-hand-side limit, RHSL $\leq \varepsilon_c$, the following condition must be met:

$$g \leq g_c \quad (6)$$

where

$$g = [\rho/|a_{ii}|]\, m_{ij} \quad (7)$$

$$g_c = c\,\varepsilon_c \quad (8)$$

$$c = 1/(3.01\ \varepsilon). \quad (9)$$

The growth of calculation error may depend on the machines with different allocations to the exponent part and quotient part of the memory.

For a 64 bit machine allocating 52 bits for quotient, for example, the above equation becomes

$$c = 1.49\text{x}10^{15} \text{ for 64 bit memory.} \quad (10)$$

Likewise, for a 16 bit machine allocating 10 bits for quotient and for a 32 bit machine allocating 24 bits for quotient, for example, the coefficient of RHS in equation (9) can be replaced by $1.49\text{x}10^2$ and by $5.57\text{x}10^6$, respectively. i.e.

$$c = 1.49\text{x}10^{2} \text{ for 16 bit memory;} \quad (11)$$

$$c = 5.57\text{x}10^{6} \text{ for 32 bit memory.} \quad (12)$$

It is easy to figure out that even two digit accuracy for decomposition is very tight to achieve in a 16 bit operation, while ten-digit accuracy can be easily achieved without much burden in a 64 bit operation.

Considering the decomposition accuracy, $\varepsilon_c = 10^{-2}$, 10^{-3}, 10^{-6} and 10^{-10} for example, the user can estimate what kind of margin for growth he or she can play with in this process. As the machine accuracy increases, the accuracy of decomposition is well guaranteed in this process. Again, this does not mean that the solution accuracy is also guaranteed, since good decomposition accuracy does not necessarily mean that the new matrix system with old pivot sequence is still well-conditioned. This means that ε_1 can be larger than ε_2 (g_1 can be larger than g_2, accordingly) for a high accuracy machine. This means that the user has to regularly check the residual to see if the solution is still accurate, even when there seems to be no problem.

Once the machine accuracy is determined, the user can set-up the strategy of checking the growth of decomposition error and switching the solution option when necessary. Since the growth of error comes from the bit truncation, it is recommended to check the solution accuracy every time when the g-factor increases certain factor of times greater than the first case. The user may decide the size of this factor as 2^m to match the character of bit operation.

The solution accuracy can be evaluated by regularly checking if the next value is small enough:

$$r_m = \max_i |\, b_i - \sum_j a_{ij} x_j \,| \quad (13)$$

2.5 Results

The method has been tested with typical 10x10 matrix equation on 64-bit machine. The result shows calculation accuracy of 14 digits as expected from the above relationships.

3. Conclusions

The test results have shown that the direct solution methodology applied to the hydraulic solver package being developed works well with typical test cases.

Acknowledgment

This study was performed under the project, "Development of safety analysis codes for nuclear power plants" sponsored by the Ministry of Knowledge Economy.

REFERENCES

[1] Aspentech, February 27, 2008, HSL 2007 MA48 Version 2.1.0 Package Specification, AERE, Harwell, Oxfordshire.
[2] I.S. Duff & J.K. Reid, October 1993, MA48, a Fortran Code for Direct Solution of Sparse Unsymmetric Linear Systems of Equations, RAL-93-072, Central Computing Department, Rutherford Appleton Laboratory, Oxon OX11 0QX.
[3] I.S. Duff, MA28 - A set of Fortran Subroutines for Sparse Unsymmetric Linear Equations, AERE - R 8730, Computer Science and Systems Division, AERE Harwell, Oxfordshire November 1980.
[4] A.R. Curtis & J.K. Reid, MA18 - Fortran Subroutines for the Solution of Sparse Sets of Linear Equations, AERE - R 6844, Theoretical Physics Division, AERE Harwell, Berkshire 1971.
[5] J.K. Reid, A note on the stability of Gaussian elimination, Journal of the Institute of Mathematics and its Application, Vol. 8, No. 3, pages 374-375, December 1971.
[6] L.W. Johnson & R.D. Riess, Numerical Analysis, 2nd Ed., Virginia Polytechnic Institute & State University, Addison-Wesley Publishing Co., 1982.