Methods for Enhancing the Speed of Numerical Calculations for the Prediction of the Mechanical Behavior of Parts Made Using Additive Manufacturing

Mohammad Nikoukar, Nachiket Patil, Deepankar Pal, and Brent Stucker
Department of Industrial Engineering, University of Louisville, Louisville, KY 40292

Abstract

Finite element modeling (FEM) is one of the most common methods for predicting the thermo-mechanical properties of 3D structures. Since FEM was developed primarily to analyze and optimize structures that would then be mass-produced, the time for modeling was small compared to the time required to produce the components. With the advent of Additive Manufacturing (AM) it is now possible to produce and test complex parts more quickly than FEM methods can predict their mechanical performance. As such, an enhanced numerical method for quickly solving for the mechanical behavior of components is needed to fully take advantage of the speed and versatility of this new manufacturing paradigm.

In order to enhance the computational efficiency of FEM, a novel method was developed to adapt FEM for prediction of fundamental deformation responses of AM-produced parts. A general FEM strategy comprised of constructing the stiffness and external stimuli (such as laser power or pressure) as matrices and vectors respectively has been formulated. Thermo-mechanical response is calculated by obtaining the compliance matrix from the stiffness matrix and then multiplying the corresponding values of the compliance matrix with the external stimulus vector. Obtaining the compliance matrix from the stiffness matrix is accomplished, in most cases, using a well-known Cholesky algorithm which starts by transforming the stiffness matrix into a lower triangular matrix with zeros above its diagonal [1]. In this study, the Cholesky algorithm has been improved by identification of discrete sparse bands and by eliminating many zero multiplications in the lower triangular matrix to obtain the thermo-mechanical response much faster than currently available algorithms. In addition, the vector based storage strategy of the above-mentioned discrete sparse bands and multipliers have been used to save computer storage space, including free cache memory, resulting in faster computations. An example showing the time advantage of this new framework over previously used algorithms to obtain the deformation response of an additively manufactured axial beam is provided along with its theoretical background.

Keywords: Cholesky algorithm, stiffness matrix, additive manufacturing, thermo-mechanical properties prediction, finite element analysis

1- Introduction

Finite element analysis (FEA) is one of the common ways of analyzing deformation in structures under mechanical loads or thermal shocks. This method is a general method which can fit a wide range of different problems including structures with complex geometries and various loads such as tension, shear and torque. Most FEA uses commercial software packages such as ANSYS which provide users with a wide variety of options to input boundary conditions and geometry into the software and solve it. For instance, I.A. Roberts et al. used ANSYS Multiphysics FEA to simulate their model consisting of five 30 mm layers of TiAl6V4 [2]. Most of the available software packages are general and they cannot provide enough freedom for researchers to exactly input their models into the software. This limitation may also increase the calculation time for specific problems which can cause serious problems for large scale models.

Solution strategies used in commercial software include exact and heuristic algorithms. One of the well-known algorithms in this area is the Cholesky algorithm which is only useful for positive definite matrices [3]. Due to the usefulness of the Cholesky algorithm for a wide variety of problems, subroutines can be implemented within various software packages. For instance, a specific package called CHOLMOD has been developed for factorization of sparse matrices [4] and Saunders has used Cholesky algorithms to solve large scale linear programming problems [5].

2- Background

A Cholesky algorithm is used to solve linear equations in the form of AX=B, where A is real, symmetric and positive definite. X and B can be either rectangular matrices or vectors. These forms of linear equations are the same as finite element equations. In FEA, the stiffness matrix (K) can be considered as the variable A since it is symmetric and positive definite. External forces vector (F) can serve as X and displacement vector (U) can be considered as B. This changes the B=AX equation into U=KF. Figure (1) shows a system of linear equations in matrix form.

$$\begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix}$$

Figure (1) - matrix form of AX=B equation

To solve the abovementioned linear equation, A should be factored using Cholesky decomposition. The Cholesky algorithm transforms the A matrix into a lower or upper triangular matrix. Equation (1) expresses this mathematically.

$$A=U^{T}U\text{, if only upper triangular part of matrix A is given}$$
$$A=L^{T}L\text{, if only lower triangular part of matrix A is given} \qquad (1)$$

In equation (1), L represents a lower triangular matrix and U represents an upper triangular matrix [6]. There are two main methods for carrying out the Cholesky decomposition which are known as forward and backward substitution.

2-1- Forward substitution

In forward substitution the lower triangular matrix of A is used with non-zero diagonals. The matrix representation for forward substitution is shown in figure (2).

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Figure (2) - Cholesky decomposition using forward substitution

In this method, the first value can be calculated by multiplying the first row of the lower triangular matrix into the first row of the X matrix and setting it equal to the first element of the B matrix. a_{11} and b_1 are known in the first equation ($a_{11}.x_1=b_1$) and the value of x_1 can be easily calculated. The second equation ($a_{21}.x_1+ a_{22}.x_2=b_2$) only involves x_1 and x_2 so the value of x_2 can also be calculated since we already know the value of x_1 from the first equation. By continuing this way to the nth equation ($a_{n1}.x_1+ a_{n2}.x_2+ . . . + a_{nn}.x_n=b_n$) all the values x_1 through x_{n-1} are known and allow us to calculate x_n . Equation (2) shows the resulting formulas from this method [7-9].

$$\begin{aligned} x_1 &:= b_1/a_{11} \\ x_2 &:= (b_2 - a_{21}x_1)/a_{22} \\ x_3 &:= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\ &\vdots \\ x_n &:= (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn} \end{aligned} \tag{2}$$

2-2- Backward substitution

In backward substitution the upper triangular matrix is used and the calculations start at the last row of the matrix and continue toward the first row of the matrix, which has the maximum number of elements. Figure (3) shows the system of equations for solving the B=AX equation using the upper triangular matrix.

$$\begin{bmatrix} a_{11} & \cdots & a_{1,n-1} & a_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

Figure (3) - Cholesky decomposition using backward substitution

As shown in figure 3 the calculations should be started from multiplication of the last row of the upper triangular matrix into the last row of the X matrix and set equal to the last row of the B matrix($a_{nn}.x_n=b_n$). Similar to the forward substitution method, the first equation only has one unknown variable which can be calculated from the equation. The second equation ($a_{n-1n-1}.x_{n-1}$+ $a_{n-1n}.x_n=b_{n-1}$) also has one unknown variable, x_{n-1}, and can be solved for this variable. This is continued towards the first line of the upper triangular matrix to calculate all the unknown variables [7-9]. Backward substitution is summarized mathematically in equation (3).

$$\begin{aligned} x_n &:= b_n/a_{nn} \\ x_{n-1} &:= (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1} \\ x_{n-2} &:= (b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n)/a_{n-2,n-2} \\ &\vdots \\ x_1 &:= (b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n)/a_{11} \end{aligned} \tag{3}$$

3- FEA of AM manufactured parts using the Cholesky algorithm

Finite element analysis is a useful way of analyzing additively manufactured parts to study their thermo-mechanical properties. Therefore the Cholesky algorithm can be useful for conducting FEA of additive manufacturing. However, AM-produced parts FEA and general FEA have differences which should be taken into account. In AM there are some numbers in the stiffness matrix that are very small and do not have a meaningful effect on the final solution. Although these values are very small, considering them in calculations will increase the required memory. In the presented Cholesky method an adjustable threshold could be employed for different classes of problems with a threshold of 10^{-4} employed on row-by-row multiplication of the lower triangular matrices used during the forward substitution to filter any small numbers during multiplication. This helps the program run faster while keeping the maximum error within a desired range.

3-1-Stiffness matrix of AM manufactured part

The only non-zero values in these matrices are in a certain range of bands depending on the dominant axial and shear stiffness counterparts of the structure. The layer-by-layer information of the structure is contained in the diagonal band segments. Figure (4) shows an example of a stiffness matrix of 7 layers comprising an additively manufactured part with each layer comprising one element.

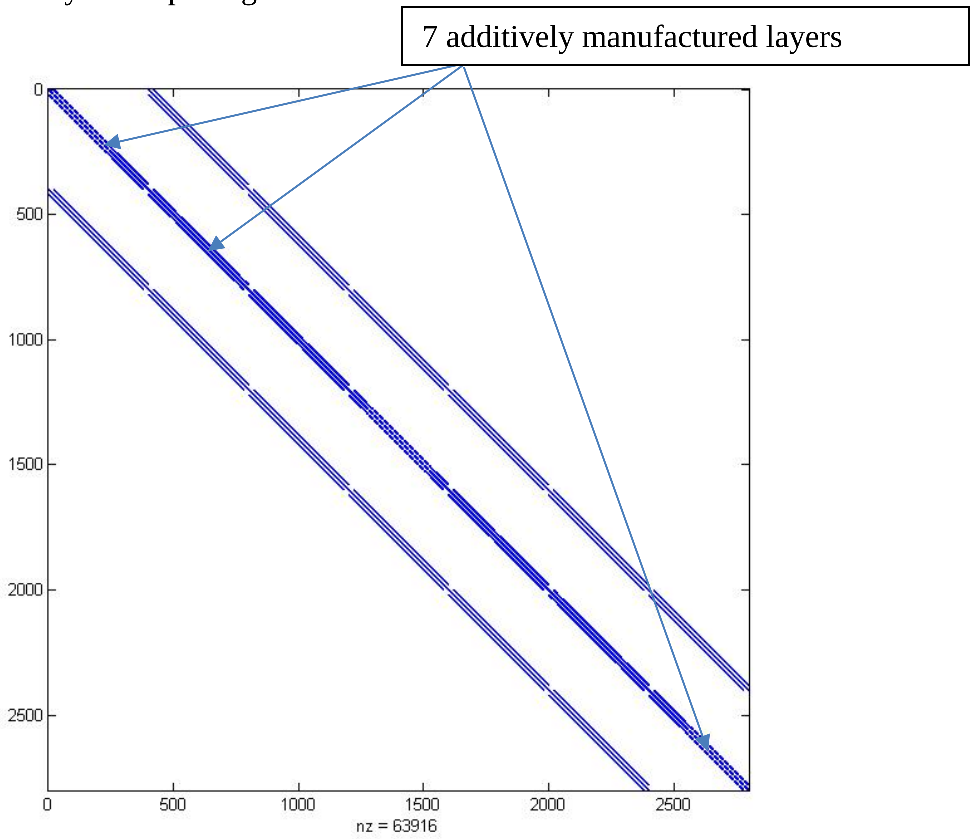

Figure (4) - sparse stiffness matrix of a 3 dimensional additively manufactured part

This matrix belongs to an additively manufactured part which is meshed using a hex mesh with 8 nodes per element. The alignment of each band represents the connection between nodes within each layer and its previous and next layers. In other words, these modified matrices have a specific form which is shown in figure (5).

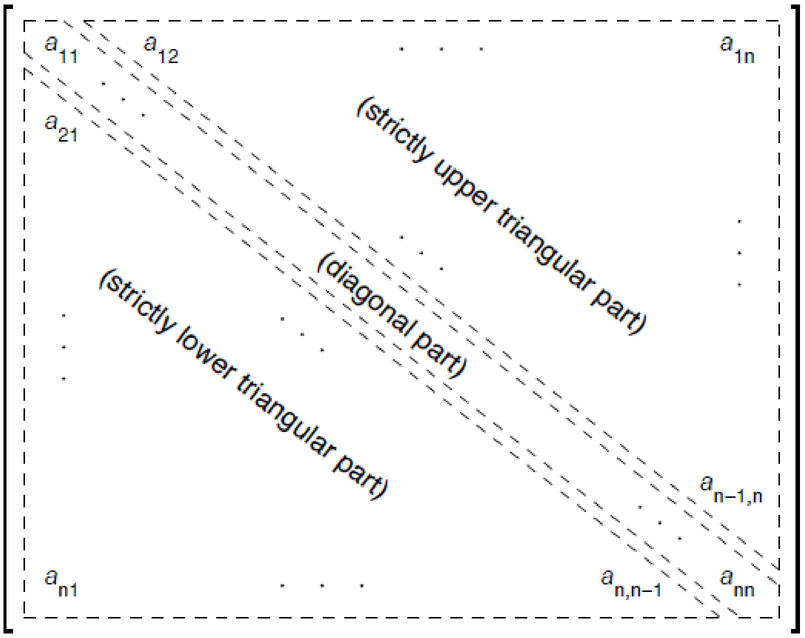

Figure (5) - general form of stiffness matrix of an additively manufactured part

As shown in figure (5), the diagonal part of the matrix shows the connection of the nodes with themselves and with surrounding nodes within the same layer. The upper and lower triangular parts represent the connection of the node with previous and next layers.

3-2- Cholesky algorithms for FEA of AM made parts

A Cholesky algorithm can be used for solving finite element problems for parts made using additive manufacturing. This algorithm treats AM problems exactly the same as other problems which have populated stiffness matrices. This means the Cholesky algorithm calculates all the values in the matrix while it is known beforehand that many of these values are zeros and are not required to be calculated. Considering the portion of the non-zero values to the zero values in the stiffness matrix – which is around 1-5 percent of the total number of values – clearly demonstrates that doing a normal Cholesky for an AM problem increases the number of calculations. In other words, although the Cholesky method is very efficient for solving simultaneous equations; even by doing a Cholesky for an AM problem many of the calculations are unnecessary. The main motivation for the current study is to modify the current Cholesky algorithm to eliminate all unnecessary calculations to provide faster solutions for FEA of AM made parts.

4- Proposed methodology

The proposed solution is a stepwise procedure which focuses on creating a lower triangular matrix that can be used for the forward substitution method. To illustrate the procedure and for comparison purposes a 3-D example of a thermal problem is used in this paper.

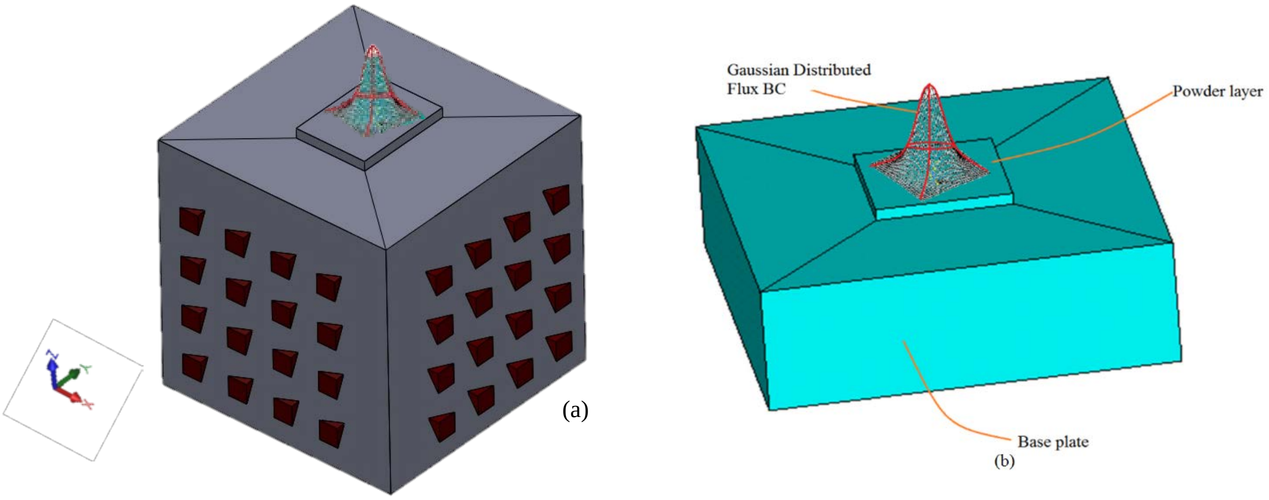

Figure (6) – (a) Geometry of the thermal problem with periodic boundary conditions on opposite sides, a Gaussian flux on the top surface and a constant temperature of 353 K at the base surface. (b) Boundary conditions for metal laser sintering

As shown in figure (6), a 3-D thermal problem is created in which the temperature of the base plate is kept at 353 K and both side planes perpendicular to the X-axis are assumed to have a constraint of having equal temperatures. The same condition holds for the both side planes which are perpendicular to the Y-axis. The top plane has a Gaussian flux with a beam diameter of 100 microns.

4-1- presented Cholesky procedure

The procedure of the presented algorithm is briefly explained in this section. This program is written in MATLAB and the results of this program are compared with the Cholesky subroutine of MATLAB.

First step- Every band of the stiffness matrix (K) will be stored as a vector from beginning to the end. This reduces the required storage memory of the system tremendously. A secondary MATLAB program is used to extract the bands from the stiffness matrix and export them as a text file which is readable by FORTRAN.

Second step- The first value of the lower triangular matrix (L) is calculated directly from the corresponding value of the K matrix using the equation $l_{11} = \sqrt{k_{11}}$

Third step- Each l_{ij} value in the L matrix can be derived by multiplying the ith row into the jth row of the L matrix and setting it equal to the K_{ij} from the K matrix. As a result, an equation can be derived for calculating each value in the L matrix based on the previously calculated values in the L matrix and the corresponding value in the K matrix. The mathematical formulation is shown in equation (4).

$$l_{ij} = k_{ij} - (L_{i\cdot} \cdot L_{j\cdot}) \qquad (4)$$

Multiplication of the ith row into the jth row is done by using the dot product operation which is the element by element multiplication. Therefore equation (4) can be extracted into equation (5). This equation holds for all values except the diagonals in which the equation changes into equation (6).

$$l_{ij} = \left[k_{ij} - \left(l_{i1}.l_{j1} + l_{i2}.l_{j2} + \cdots + l_{ii-1}.l_{jj-1}\right)\right]/l_{ii}.l_{jj} \qquad (5)$$

$$l_{ii} = \sqrt{[k_{ii} - (l_{i1}^2 + l_{i2}^2 + \cdots + l_{i-1\,i-1}^2)]} \qquad (6)$$

Fourth step- since all the bands follow a specific pattern, by recognizing this pattern all zero multiplications can be eliminated and this help to increase the computational speed by reducing the number of flops.

Fifth step- All L_{ij} values will be stored in the form of vectors and the number of vectors are equal to the number of bands in the L matrix.

5- Results and discussions

The presented algorithm was used to solve a FEM thermal problem and as discussed in the previous sections. An iterative procedure was used to calculate the lower triangular factor of the stiffness matrix using the presented algorithm, calculating and recording the maximum error and number of flops it needs to do the Cholesky factorization. In figure (7), the blue line shows the normalized number of flops required for each threshold used to calculate the lower triangular factor of the stiffness matrix. The red line in figure (7) shows the normalized value of the maximum error for each threshold. All normalized vales are calculated by dividing the values with corresponding maximum value of the variable.

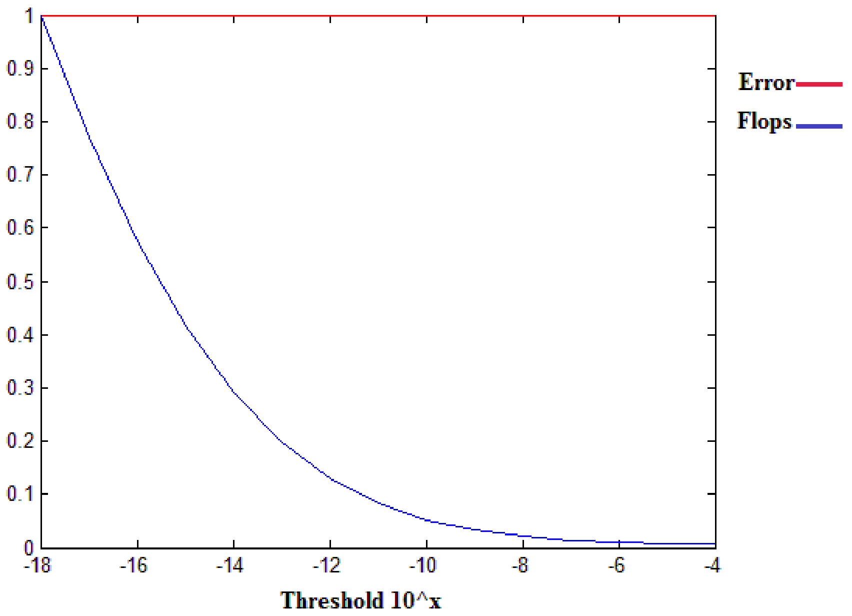

Figure (7) –Number of flops and error with respect to maximum number of flops and error respectively as a function of threshold

As shown in figure (7), by increasing the threshold, the number of values that are filtered will increase which results in decreasing the number of flops required. Since all the eliminated values are not significant values in the stiffness matrix, the normalized error remains close to 1 and the error is negligible when compared to performing a traditional Cholesky factorization. Decreasing the number of flops required increases the computation speed and decreases the storage memory required for doing the calculations. Figure (8) shows the number of flops versus the threshold log and it is clear that the number of flops decrease when increasing the value of threshold.

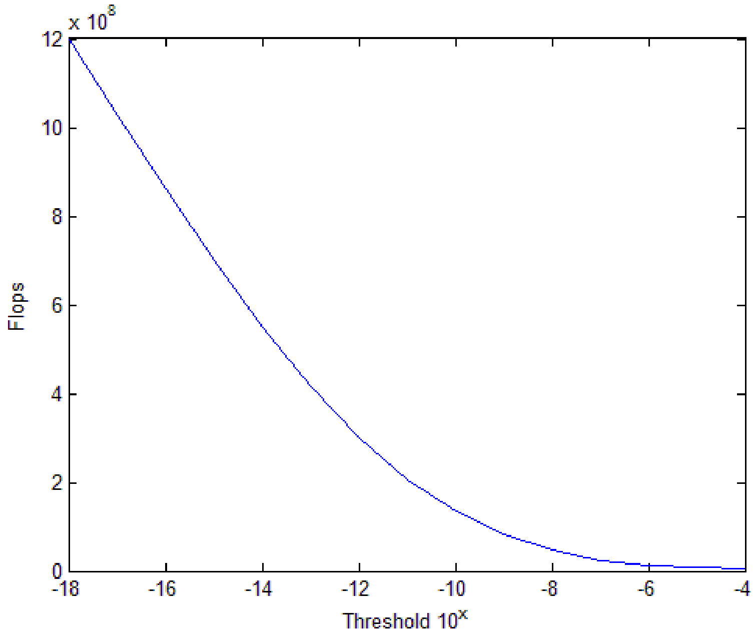

Figure (8) – Number of flops verses threshold plot of the presented algorithm

Figure (9) shows the actual error values versus the threshold and it can be understood that the error values are changing from zero to the order of 10^{-5} as the threshold changes from 10^{-18} to 10^{-4}.

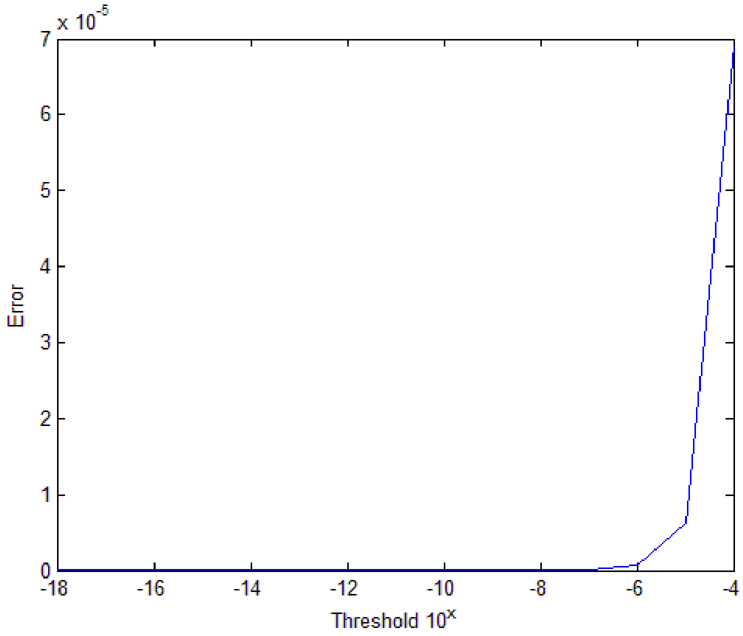

Figure (9) – Error verses threshold plot of the presented algorithm

Figure (10) shows the L matrix that has been calculated using the Cholesky subroutine of MATLAB in which the total number of non-zero values is 1,214,378.

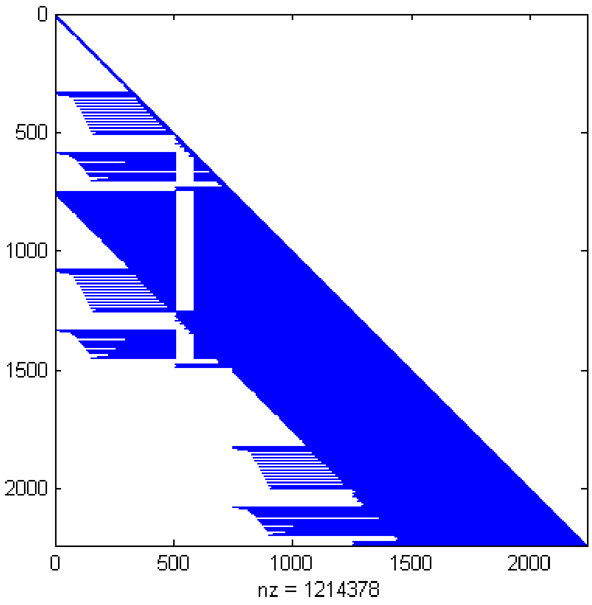

Figure (10) – L matrix that is calculated by the Cholesky subroutine of the MATLAB

Figure (11) shows the L matrix that is calculated by this algorithm at a threshold of 10^{-4} and has 81,585 values in the matrix which is 6.7% of the number of values of the Cholesky subroutine of MATLAB.

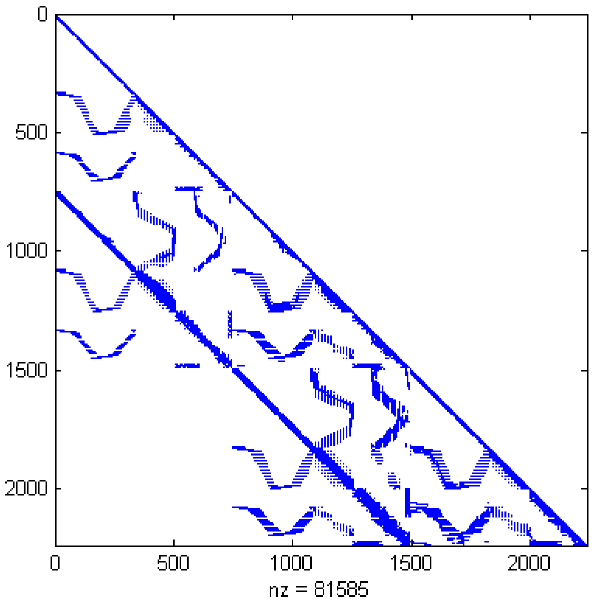

Figure (11) – L matrix that is calculated by the Cholesky subroutine of the MATLAB

6- Conclusion

Comparisons clearly show the significant improvements in the time and memory required to solve a finite element problem relevant for AM simulations. This can provide researchers a method for solving relatively large problems on standard computers. This algorithm can also be used in industrial companies for real time analysis of parts to predict their properties before the start of the part manufacturing process. Since this algorithm does not use as much memory, it can be used as an add-on to the machines' operating software packages for real time prediction of the part properties to help the operator predict part characteristics before making parts.

References

1. C Brezinski, André Louis Cholesky [1875-1918] (French), *Numerical analysis, Louvain-la-Neuve, 1995, Bull. Belg. Math. Soc. Simon Stevin* (1996), suppl., 45-50.
2. I.A. Roberts, C.J. Wang, R. Esterlein, M. Stanford, D.J. Mynors, A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing, International Journal of Machine Tools and Manufacture, Volume 49, Issues 12–13, October 2009, Pages 916-923
3. Y. Chen, T. A. Davis, W.W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate, ACM Transactions on Mathematical Software (TOMS), Volume 35 Issue 3, October 2008
4. M. A. Saunders, Large-scale linear programming using the Cholesky factorization, Stanford University Stanford, CA, USA 1972
5. Ng, E. and Peyton, B., Block Sparse Cholesky Algorithms on Advanced Uniprocessor Computers, SIAM Journal on Scientific Computing 1993 14:5, 1034-1056
6. J. Waśniewski, B. Stig Andersen, F. Gustavson, Recursive formulation of Cholesky algorithm in Fortran 90,Applied Parallel Computing Large Scale Scientific and Industrial Problems Lecture Notes in Computer Science Volume 1541, 1998, pp 574-578
7. Sheung Hun Cheng and Nicholas J. Higham (1998) *a modified Cholesky algorithm based on a symmetric indefinite factorization.* SIAM Journal on Matrix Analysis and Applications, 19 (4). pp. 1097-1110. ISSN 1095-7162
8. Tirupathi R Chandrupatla, Ashok D Belegundu, ntroduction to finite elements in engineering, Prentice-Hall,1991
9. R. B. Schnabel and E. Eskow, A new modi_ed Cholesky factorization, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1136-1158.