Parallel Efficiency Analysis of Large Increment Method Based on OpenMP

J Hongxue1*, L Danbing2, L Xila3
1China Construction Eighth Engineering Division Co. Ltd 8/F, No.1568, Century Avenue, Pudong New Area, Shanghai, 200122, China
2School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, China
3Department of Civil Engineering, Shanghai Jiaotong University, Shanghai 200240, China
jiahongxue211@csecc.com

Abstract. Large Increment Method (LIM) is a novel force-based method based on the generalized inverse matrix theory for solving material nonlinear problems in small deformation and small displacement conditions. For the spatial parallel computation, LIM can compute the constitutive equations of each element in parallel. By analyzing the parallel-calculating characteristics of LIM, the OpenMP is used for the parallel computation in spatial domain. Through the parallelism analysis of the spatial and time domain for LIM, it shows that LIM has an excellent parallel efficiency in space parallel computation.

1. Introduction
The large increment method (LIM) is a novel force-based finite element method based on generalized inverse matrix theory for solving solid material nonlinear problems in small deformation and small displacement conditions [1]. Compared with the traditional displacement-based finite element method, the advantages of LIM are high stress calculation precision and less computation, especially it is suitable to spatial and time parallel computation. For the spatial parallel computation, LIM can parallelly compute the constitutive equations of each element.

OpenMP is a standard for shared memory parallel programming and is applicable to the shared memory of multi-core computer. In this paper, OpenMP is used for the parallel computation in spatial domain.

2. Theoretical basis and parallelism analysis of large increment method

2.1. Theoretical basis of large increment method
The governing equations of LIM include equilibrium equation, compatibility equation and constitutive relation, which can be written as:

\[
\begin{align*}
\mathbf{C} \mathbf{F} &= \mathbf{P} \quad \text{Equilibrium equation} \\
\mathbf{C}^T \mathbf{D} &= \mathbf{\delta} \quad \text{Compatibility equation} \\
\mathbf{\delta}^c &= \mathbf{\Phi}^c \left( \mathbf{F}^c, \mathbf{\epsilon}^c \right) \quad \text{Constitutive equation}
\end{align*}
\]  

(1)
where \( C \) is the system equilibrium matrix, \( F \) is the internal force vector, \( P \) is the node load vector, \( D \) is the node load vector, \( \delta \) is the deformation vector, \( \Phi^e \) is the element flexibility matrix (linear or nonlinear).

In the solution procedure, the initial generalized internal forces \( F_0 \) are obtained by solving the equilibrium equation of the structural system level (Global stage), and then the corresponding generalized deformation is obtained through the constitutive relation at the element level (Local stage).

In the global stage, the compatibility equation is used to check whether the generalized deformation meets the compatibility. If not, the conjugate gradient method is used to update the generalized internal force to obtain a new solution, and then it is substituted into the local level to obtain the generalized deformation. Repeat the above steps until the generalized deformation meets the compatibility, and the generalized internal force is the optimal solution. The iterative process is presented in the following diagram, as shown in Figure 1 [1].

![Figure 1](image1.png)

Figure 1. The illustration of optimization iterative process of LIM.

The calculation process of the LIM for elastoplastic problems is as follows:

Take some sample points in the loading history \( P(t_0) \), \( P(t_1) \), \( P(t_2) \), ..., \( P(t_r) \), where \( t_0 < t_1 < t_2 < t_3 < ... < t_r = t \), as shown in Figure 2.

![Figure 2](image2.png)

Figure 2. Sample points on the loading history.

Step1: (Global stage) Get the initial solution of the equilibrium equation \( F_0(t_j) \):

\[
F_0(t_j) = C^T \left( CC^T \right)^{-1} P(t_j), \quad (j = 0, 1, 2, ..., r) \tag{2}
\]

Step2: Iterate from \( F_n(t_j) \) to \( F_{n+1}(t_j) \):

(1) (Local Stage) Get the deformation vector \( \delta_n \) according to \( F_n \)

For element \( i \), the stress history can be expressed by series \( \left\{ F_n(t_0), F_n(t_1), ..., F_n(t_r) \right\} \), so the deformation of each time sample point is:

\[
\hat{\delta}_n(t_j) = \Phi \left( F_n(t_j), e_m^i(t_j) \right), \quad (j = 0, 1, 2, ..., r) \tag{3}
\]
where $\varepsilon'_{pn}(t_j)$ is the internal variable of the element, recorded the plastic deformation history of the $i$ element, and computed according to the previous deformation history of the element.

(2) (Global stage) Check whether the deformations are sufficiently compatible or not

$$
\varepsilon(\hat{\delta}_n(t_j)) = \left[ \delta\hat{\phi}_n(t_j) \right]_{j = 0, 1, 2, \ldots, r} \quad (4)
$$

For any $j$, $j \in \{1, 2, \ldots, r\}$, if $\varepsilon(\hat{\delta}_n(t_j)) < e_0$ (where $e_0$ is the given error tolerance), then the deformations are sufficiently compatible, no more iterations are required and go to Step 3. Otherwise, the deformation vector $\hat{\delta}_n(t_j)$ is not compatible enough, $F_n(t_j)$ should be modified to improve the compatibility of the deformation vector.

(3) (Global stage) Calculate the search direction $S_n(t_j)$:

$$
S_n(t_j) = -\beta K_n(t_j) \beta \hat{\delta}_n(t_j) 
$$

where $K_n(t_j)$ is the current elastoplastic stiffness matrix, which can be obtained by calculating the inverse of flexibility matrix $\Phi_n(t_j)$ [2].

(4) (Global stage) Calculate the search step length $h_n(t_j)$:

$$
h_n(t_j) = -\frac{\hat{\delta}_n^T(t_j) S_n(t_j)}{S_n^T(t_j) \beta \hat{\delta}_n(t_j) S_n(t_j)} 
$$

(5) (Global stage) $F_{n,1}(t_j)$ can be calculated as

$$
F_{n,1}(t_j) = F_n(t_j) + h_n(t_j) S_n(t_j) 
$$

Step3: (Global stage) Obtain the final results:

$$
\hat{\delta}_n(t_j) = a \hat{\phi}_n(t_j) = a \Phi \left( F_n(t_j), e_{pn}(t_j) \right) 
\quad D_n(t_j) = \left( C C^T \right)^{-1} C \hat{\phi}_n(t_j) 
$$

2.2. Parallel analysis of large increment method

For an element, both the generalized deformation and the generalized internal force of the element are internal variables of the element, which can be calculated in the element level. Therefore, both the generation of the element flexibility matrix and the solution of the element constitutive equation can be calculated at the element level, which is called the local stage. Since the generation of the element flexibility matrix and the calculation of the element constitutive equation are completed at the local stage, there is no data transfer between the calculations of any element, so the parallel programming mode of shared memory based on OpenMP standard is adopted. The calculation of each element at each time sample point in the loading process can be carried out in parallel, the calculation at this level is called space parallel, as shown in Figure 3 [3-4].

Figure 3. The Spatial domain parallel computation in LIM.
3. Parallel efficiency analysis of large increment method

A practical and objective evaluation of the spatial parallelism of LIM is made by analyzing the bending problem of a plate under uniform load.

For a square plate with uniform load, the boundary condition is fixed support, the side length is 0.4m, the uniform load is 1000N/m², and the plate thickness is 0.08m. The material is assumed to be homogeneous, isotropic, elastic modulus is $2.1 \times 10^{11}$ N/m², and Poisson's ratio is 0.3. Five groups of coarse and fine meshes (3600, 4900, 6400, 8100 and 10000 elements) are used to solve the problem by using the quadrilateral four node plate element 4NQP11 respectively [5]. The problem scale N is defined as the total number of meshing elements. The changes of computing time, speedup and efficiency with the number of processors are shown in Figure 4 and 5 respectively.
From the Figure 4, it can be seen that the calculation time decreases significantly with the increase of the number of threads, and the overall trend of the parallel speedup ratio increases, which is enough to show that OpenMP can effectively improve the calculation efficiency when applied to the spatial parallel computing of LIM.

From the Figure 5, the parallel efficiency is also increasing with the increasing scale of the problem. The additional overhead will increase and the parallel efficiency will decline with the increase of the number of threads, but the overall trend of parallel efficiency is greater than that of small scale problems. Therefore, when considering space parallel computing, the size of the problem should be increased, so as to ensure high parallel speedup and parallel efficiency.
4. Conclusions
The space parallelism of the LIM is analyzed. On this basis, the basic structure of parallel program for LIM is given. Finally, through the parallel efficiency analysis of the LIM, it can be seen that LIM is well implemented in space by applying OpenMP. Compared with serial computing, the speedup of parallel computing in space is close to the ideal linear speedup with the increasing scale of the problem, which has good parallel efficiency; the speedup and parallel efficiency of parallel computing in time are also good with the increasing scale of the problem. In a word, the LIM has good parallelism in space, which lays a good foundation for computing large scale practical engineering problems in the future.

References
[1] Chunjun Z. A large increment method for material nonlinearity problems[D]. Beijing: Tsinghua University, 1996. (in Chinese)
[2] Hongxue J. Large increment method for plate and parallel computation[D]. Shanghai: Shanghai Jiao Tong University, 2015. (in Chinese)
[3] Kaijian W. Network parallel computation based on the large increment method[D]. Beijing: Tsinghua University, 2004. (in Chinese).
[4] Danbing L. The Extended application of parallel computing-based large increment method in computational solid mechanics[D]. Shanghai: Shanghai Jiao Tong University, 2012. (in Chinese)
[5] Hongxue J, Danbing L, Xila L. The force-based quadrilateral plate elements for plate analysis using large increment method[C]//Proceeding the 2nd International Conference on Civil Engineering and Building Materials, Hong Kong, 2012: 221-225.