

# particle Generalized FEM -basic idea and selected examples

Rong Tian

Institute of Computing Technology, Chinese Academy of Sciences,  
Kexueyuan Nanlu 6, Haidian, Beijing 100190, China  
rongtian@ncic.ac.cn

## ABSTRACT

A 'head-to-toe', naturally parallelizable finite element approach is proposed. The method is designed to be scalable with an unstructured mesh of any complicated geometry and any large size. In large deformation analysis the method also offers strong mesh distortion tolerance. The numerical performance of the method is verified and demonstrated through numerical examples in 3D.

## General Terms

Theory

## Keywords

particle generalized finite element method, partition of unity, parallel finite element method, mesh distortion.

## 1. INTRODUCTION

Powered by petaflops (1015 flop operations per second) supercomputers, numerical simulation steps into a completely new era; a parallel FE code at this scale is expected to use processor cores of order of hundreds of thousands. The above mentioned conventional parallel FE procedure would be hampered by its poor scalability in pre- and post-processing (Figure 1(a)). As a typical example, unstructured mesh generation at scale is no longer a routine task and already a de-facto bottleneck of scalability. Visualization and IO of data in tera- and petabytes also would not be feasible without resorting to parallel techniques[5]. In this paper, a particle generalized FEM is proposed with two purposes: (a) easy implementation at scale; (b) high tolerance to mesh distortion. Figure 1(b) is used to illustrate the proposed method. Compared with the conventional parallel FEM, the proposed method is expected to overcome the serious bottleneck of unstructured mesh generation and to be embarrassing parallel in the pre and the post processes.

## 2. THE METHOD

### 2.1 Basic idea

Let  $\bigcup M_i^{geo}$  denote an assembly of meshes  $M_i^{geo}$  that are non-overlapping, non-conforming, and completely independent meshes. The mesh assembly forms a non-overlapping and seamless decomposition of  $\Omega$ , and is called *geometric mesh* or *physical mesh*, labeled by superscript "geo". Let  $M^{math} = cell(h_x \times h_y \times h_z)$  be a Cartesian structured grid called *math grid* that covers  $\Omega$  and it is independent of  $\Omega$ .  $h_x, h_y, h_z$  are respectively the math cell size in  $x, y, z$  coordinate direction. The vertex of the geometric mesh and the math grid is called, respectively, *physical node* and *math node*, and the element of each set of mesh is termed, respectively, *geometric/physical element*, and *math cell*. For the problem of interest, given a strong form of PDEs

$$\begin{cases} \mathbf{A}(\mathbf{u}) = 0; & in \Omega \\ \mathbf{B}(\mathbf{u}) = 0; & in \Gamma \end{cases} \quad (1)$$

which are, respectively, the differential equation set to be satisfied in the volume of material,  $\Omega$ , and on the boundary of the domain,  $\Gamma$ , a general statement of the Galerkin weighted residual approach is expressed as

$$\int_{\Omega} \mathbf{N}^T \mathbf{A}(\mathbf{N}\mathbf{u}) d\Omega + \int_{\Gamma} \mathbf{N}^T \mathbf{B}(\mathbf{N}\mathbf{u}) d\Gamma = 0 \quad (2)$$

The reason that the pGFEM is called a *particle* method is that the geometric mesh, as it is mainly used for a quadrature purpose, merely appears in form of a Gauss point set during momentum equation solving (but the location and the weight of the Gauss points are still updated by reference to the geometric mesh). As such, during the solving phase of the momentum equations, although we employ two sets of meshes, the geometric mesh is actually really used as a set of points. In addition, the connectivity between a given Gauss point and a math node is not pre-defined but in a runtime/dynamic manner. The particle feature and the dynamic connectivity make the proposed method more like a particle method of the meshfree class.[?]

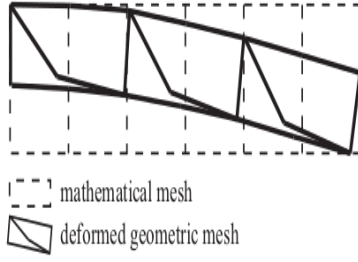
### 2.2 Step total Lagrangian formulation

Let  $\mathbf{x}^n$  be an intermediate configuration. Assume the material can be mapped one by one between the current configurations  $\mathbf{x}$  and  $\mathbf{x}^n$ , i.e., there is no cracking and splitting of a material point, we express  $\mathbf{x}$  with respect to  $\mathbf{x}^n$  as[3].

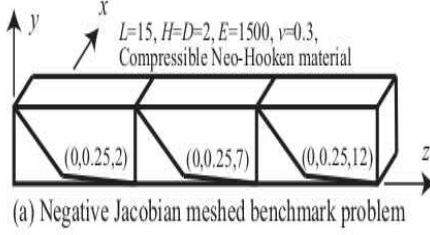
$$\mathbf{x}(\mathbf{x}^n, t) = \mathbf{x}^n + \Delta\mathbf{u}(t) \quad (3)$$

where  $\Delta\mathbf{u}(t)$  is finite deformation.

## 3. RELATIONS WITH THE EXISTING METHODS



**Figure 1: A typical pGFEM mesh in large deformation analysis**



**Figure 2: Negative Jacobian meshed benchmark problem**

The pGFEM follows a line of a partition of unity-based generalized FEM (Strouboulis et al. (2001)) and is designed to combine together with the major advantage of the Material Point Method (Sulsky et al. (1994), Zhang et al. (2006)) in avoiding mesh distortion in tracing large deformation and deformation discontinuity. The same as the MPM, the pGFEM solves the momentum equation on an independent and regular math grid. The MPM is an explicit method and hence the data projection, such as mass and momentum etc, between the material points and the background mesh has to be done every single time step.[1] The pGFEM is designed as an implicit method and the step total Lagrangian formulation is used to avoid frequent data projection between two meshes. If deformation is moderately large and continuous, then  $\mathbf{x}^n$  is never necessarily updated and the pGFEM becomes a total Lagrangian method.

#### 4. SELECTED EXAMPLES

The test is carried out in a nonlinear large deformation setup of the cantilever problem in Figure 3. The material is taken as a compressible Neo-Hookean material. The point preconditioned conjugate gradient method is used to solve the system equation. Line search is used to search the solution of Newton iterations. The load at the free end is applied evenly in 10 steps. To make the test general, the intermediate configuration  $\mathbf{x}^n$  is updated every load step so that the math grid aligns the outer boundary of the cantilever only at the first load step and from the first step on the math grid misaligns with the geometric mesh like that in Figure 4. Noted is that this is the worst scenario in practical use and it is used on purpose to test the method's robustness.[2]

MergeSort	0.05s	0.62s	7.07s
QuickSort	0.03s	0.40s	4.75s
M_QuickSort	0.04s	0.42s	5.04s
MergeSort_Stack	0.06s	0.68s	7.71s
QuickSort_Stack	0.04s	0.50s	5.72s
M_QuickSort_Stack	0.04s	0.48s	5.47s

#### 5. CONCLUDING REMARKS

A particle generalized FEM has been developed, motivated by the potential scalability issue of parallel FE simulation at extreme scale. The pGFEM is designed to be naturally parallelizable. The method has a potential in overcoming the difficulties in pre- and post processes at scale, in particular making unstructured mesh generation easy. Analytical and numerical studies showed the method is also not sensitive to mesh distortion.[4]

#### 6. ACKNOWLEDGMENTS

Without Qingxu, nothing will be much worth doing.

#### 7. REFERENCES

- [1] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.
- [2] H. S. D. Sulsky, Z. Chen. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, (118):179–196, 1994.
- [3] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
- [4] S. Salas and E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.
- [5] R. Tian. Co-design thinking toward exascale computing. *Information Technology Letter*, 10(3):50–63, November 2010.