



A massively parallel program to solve the phase field formulation for crack propagation

Vahid Ziaei-Rad, Yongxing Shen

► To cite this version:

Vahid Ziaei-Rad, Yongxing Shen. A massively parallel program to solve the phase field formulation for crack propagation. 2nd ECCOMAS Young Investigators Conference (YIC 2013), Sep 2013, Bordeaux, France. hal-00855890

HAL Id: hal-00855890

<https://hal.science/hal-00855890>

Submitted on 30 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A massively parallel program to solve the phase field formulation for crack propagation

V. Ziaei-Rad*, Y. Shen

Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya (UPC BarcelonaTech), 08034 Barcelona, Spain

*vahid.ziaei@upc.edu

Abstract. *Phase field models for fracture employ a continuous field variable to model cracks. Therefore, in contrast to discrete descriptions of fracture, numerical tracking of discontinuities in the displacement field are not required. This really reduces implementation complexity. In this paper, we discuss the use of a single graphical processing unit (GPU) to accelerate the solution of a phase field formulation of a fracture problem using a regular finite element method (FEM) on unstructured meshes.*

Keywords: Phase field; GPU implementation; Explicit methods; Fracture mechanics

1 INTRODUCTION

Recently, the phase field method, an innovative approach for the numerical simulation of brittle fracture, have emerged [1, 2, 3, 4]. Phase field models for fracture employ a continuous field variable to represent cracks. The width of the transition zone between cracked and un-cracked areas is controlled by a regularization parameter which smoothes the boundary of the crack over a small region.

The major advantage of using a phase field is that the evolution of fracture surfaces follows from the solution of a coupled system of partial differential equations. In contrast to discrete descriptions of fracture, phase-field descriptions do not require numerical tracking of discontinuities in the displacement field. This greatly reduces implementation complexity, and is anticipated to be particularly advantageous when multiple branching and merging cracks are considered in three dimensions.

The numerical implementation of such models is sensitive to the choice of this parameter in conjunction with the mesh size, as the mesh has to be fine enough to resolve high gradients of the crack field appearing in the transition zones. This is one of the main computational limits and challenges of the implementation.

Recently, graphics processing units (GPUs) have had great success in accelerating many numerical computations. Therefore, massively parallel programming with GPU can be considered as a solution for the problem of high computational costs in the phase field method. The goal of this paper is to create and analyze a strategy to perform an explicit solution of the phase field formulation using the Compute Unified Device Architecture (CUDA) on NVIDIA GPUs. Here we mention [5] as one of the early applications of the GPU to finite element methods.

In Section 2, we derive the fundamental equations to be used in the phase field model. In Section 3, we present numerical solution for the functional. In Section 4, we briefly discuss about the GPU architecture, and how we have implemented the mathematical model on GPU. Finally, Section 5 outlines a representative numerical example that demonstrates the features of the proposed phase field models of fracture.

2 PROBLEM STATEMENT

We closely follow Miehe et al. [2] in establishing the governing equations. We consider a two-dimensional solid initially occupying the polygon Ω . The energy storage function ψ that describes the energy stored in the bulk of the solid per unit volume depends on the displacement field \mathbf{u} and the fracture phase field d . A fully isotropic

constitutive assumption for the degradation of energy due to fracture has the form

$$\psi(\varepsilon, d) = [(1 - d)^2 + k]\psi_0(\varepsilon), \quad (1)$$

where $\psi_0 := \psi_0(\varepsilon) = \lambda \operatorname{tr}^2[\varepsilon]/2 + \mu \operatorname{tr}[\varepsilon^2]$ is the standard free energy function of an undamaged elastic solid with elastic constants $\lambda + 2\mu > 0$ and $\mu > 0$, and $\varepsilon = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$.

In Equation (1), $d \in [0, 1]$. $d = 0$ and $d = 1$ correspond to unbroken and fully broken states, respectively. Moreover, k is a small positive parameter which circumvents the full degradation of the energy by leaving the *artificial elastic rest energy density* $k\psi_0(\varepsilon)$ at a fully broken state $d = 1$.

With these constitutive assumptions at hand, the evolution of energy is governed by the constitutive expression

$$\sigma := \partial_\varepsilon \psi = [(1 - d)^2 + k]\sigma_0(\varepsilon)$$

in terms of the stress tensor of an undamaged solid $\sigma_0 := \lambda \operatorname{tr}[\varepsilon] \mathbf{1} + 2\mu \varepsilon$.

The momentum balance equation for the fully isotropic case is written as:

$$\operatorname{div} \sigma + \rho \mathbf{b} = \rho \ddot{\mathbf{u}}$$

Finally, the time-dependent crack evolution can be represented by

$$\dot{d} = \frac{1}{\eta} \left\langle g_c l \Delta d + 2(1 - d)\psi_0(\nabla_s \mathbf{u}) - \frac{g_c}{l} d \right\rangle_+ \quad (2)$$

where $\langle x \rangle_+ := (x + |x|)/2$, the parameter g_c is a constitutive threshold value related to the critical Griffith-type fracture energy, l is the length-scale parameter, and η is viscosity which determines a viscous over-force response that governs the evolution of the phase field. Note that $\dot{d} \geq 0$.

With regard to the displacement field, the exterior surface of the body Ω is partitioned as $\partial\Omega = \Gamma_u \cup \Gamma_t$ into a part Γ_u , where the displacement is prescribed by the Dirichlet condition $\mathbf{u}(x, t) = \mathbf{u}_D(x, t)$, and a part Γ_t , in which the traction is imposed as Neumann-type boundary conditions, $\sigma \cdot \mathbf{n}(\mathbf{x}, t) = \mathbf{t}_N(\mathbf{x}, t)$. Note that $\Gamma_u \cap \Gamma_t = \emptyset$.

3 NUMERICAL SOLUTION

We discretize the displacement and phase fields with an unstructured triangular mesh and standard P_1 elements. For time integration, we adopt the central difference method for the displacement field. Including damping, the semidiscrete equation of motion is

$$\mathbf{M}\mathbf{a} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{u} = \mathbf{F} \quad (3)$$

where \mathbf{M} is the mass matrix, \mathbf{C} is the viscous damping matrix, \mathbf{K} is the stiffness matrix, and \mathbf{F} is the vector of applied forces. Vectors \mathbf{u} , \mathbf{v} , and \mathbf{a} are the discrete displacement, velocity, and acceleration vectors, respectively.

To start the process, having $\mathbf{u}_0 := \mathbf{u}(\mathbf{x}, 0)$ and $\mathbf{v}_0 := \mathbf{v}(\mathbf{x}, 0)$ as initial conditions, $\mathbf{a}_0 := \mathbf{a}(\mathbf{x}, 0)$ is calculated using Equation (3). Then, a recursion relation below determines \mathbf{a}_{n+1} .

$$\left(\mathbf{M} + \frac{1}{2} \Delta t \mathbf{C} \right) \mathbf{a}_{n+1} = \mathbf{F}_{n+1} - \mathbf{C} \tilde{\mathbf{v}}_{n+1} - \mathbf{K} \tilde{\mathbf{u}}_{n+1}$$

where the predictors can be found as below

$$\tilde{u}_{n+1} = u_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n \quad \tilde{v}_{n+1} = v_n + \frac{\Delta t}{2} a_n$$

After that, Equation (4) can be used to calculate \mathbf{u}_{n+1} and \mathbf{v}_{n+1} , respectively.

$$u_{n+1} = \tilde{u}_{n+1} \quad v_{n+1} = \tilde{v}_{n+1} + \frac{\Delta t}{2} a_{n+1} \quad (4)$$

For the phase field, we adopt a second-order Runge-Kutta method. The matrix form of (2) is written as

$$\overline{\mathbf{M}} \dot{\mathbf{d}} + \overline{\mathbf{K}} \mathbf{d} = \mathbf{F} \quad (5)$$

Having the discrete phase field \mathbf{d} and its rate of change at the current time step $\dot{\mathbf{d}}_n$, we calculate \mathbf{d}_{n+1}^* using

$$\mathbf{d}_{n+1}^* = \mathbf{d}_n + \Delta t \dot{\mathbf{d}}_n$$

Then we use $\mathbf{d} = \mathbf{d}_{n+1}^*$ in (5) to calculate $\dot{\mathbf{d}}_{n+1}^*$ and compute \mathbf{d}_{n+1} as

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \frac{1}{2} \Delta t \left(\dot{\mathbf{d}}_{n+1}^* - \dot{\mathbf{d}}_n \right).$$

4 GPU IMPLEMENTATION

To take advantage of the GPU and considering its specific architecture [6], we have chosen to employ explicit integration and assign threads to the nodes. Hence, each thread is responsible of computing quantities of each node. First of all, the mesh is divided into several parts, each containing some number of nodes. As we have mentioned, each thread is responsible for one node, and each group of nodes is put in one thread block. In order to reduce the total number of transactions with global memory, the most frequently used data have been gathered in the shared memory. Typically, what each node needs is its coordinates, and its current field values. Each node would also need to know its neighboring nodes either within the same block, or the neighboring blocks by the means of the connectivity matrix.

After gathering all required data, each node computes the correspondent part of the calculations with regard to the numerical methods. Note that for each time step, the kernel is executed once. Therefore, each node writes its output in the global memory at the end of each time step to be read in the next execution of the kernel function corresponding to the next time step.

5 NUMERICAL EXAMPLE

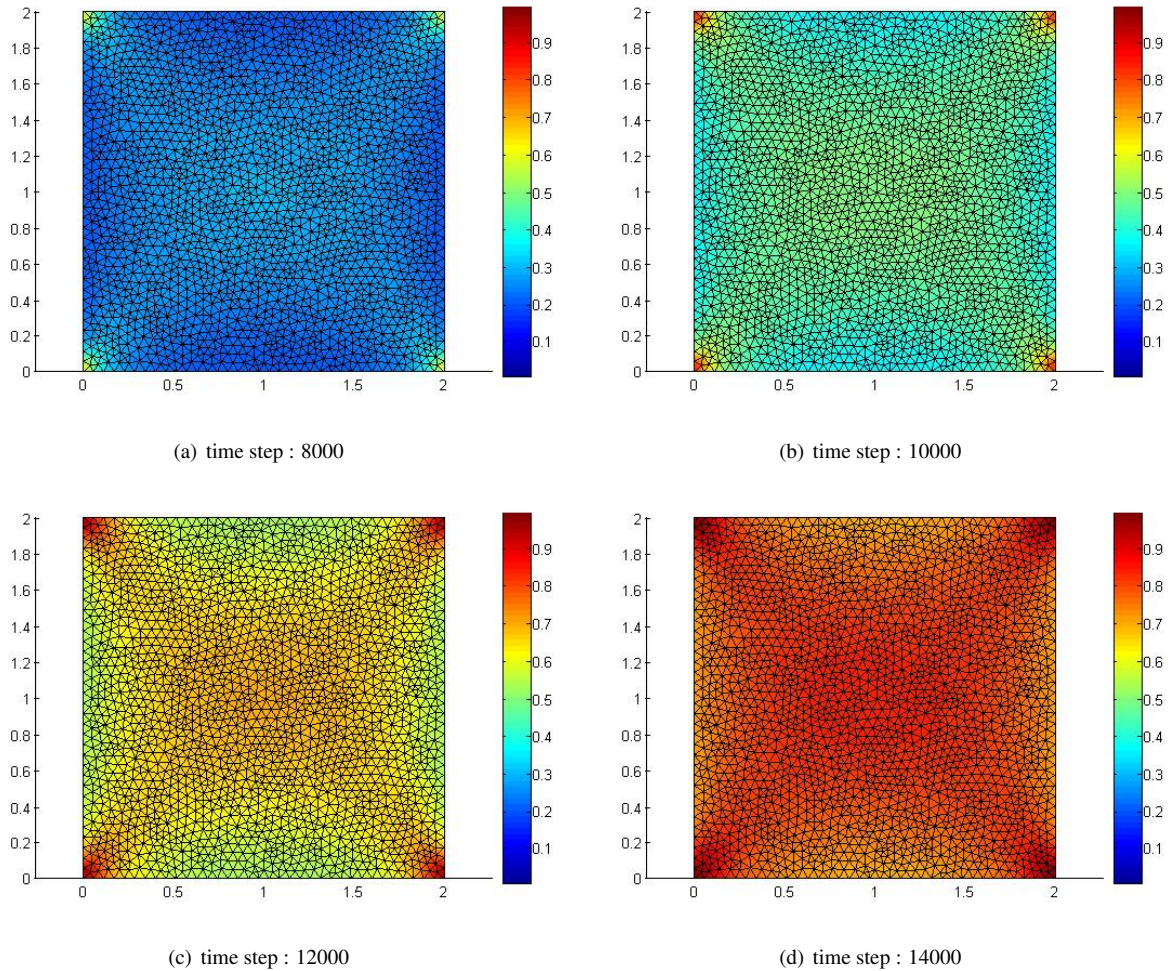


Figure 1: Phase field values at different time steps. In the last snapshot, two cracks 45 degrees to the edges are observed.

In this section, we investigate a square plate with an edge length of 0.02 mm, initially without any crack inside, under a direct tension test by displacement loading. The computation is performed with constant displacement increments $\Delta u \approx 3 \times 10^{-7}$ mm. Up to the final deformation, 2×10^4 time steps have been followed. With a discretization

of about 4000 triangular elements, an effective element size of approximately 4×10^{-4} mm is obtained. The elastic constants are chosen as $\lambda = 120$ kN/mm² and $\mu = 80$ kN/mm². We consider the critical energy release rate as $g_c = 2.7 \times 10^{-3}$ kN/mm², the length scale as $l = 7.5 \times 10^{-3}$ mm, and the viscosity as $\eta = 2.5 \times 10^{-5}$ kNs/mm². The resulting phase field values at four different stages of the deformation are illustrated in Figure 1. These four snapshots have been obtained from 8000, 10000, 12000, and 14000 time steps, respectively. In the last snapshot, two cracks 45 degrees to the edges are observed.

6 CONCLUSIONS

In this paper, we introduced the GPU implementation of an explicit method for phase-field models of crack propagation. In the nearest future, we will also consider a more realistic anisotropic assumption as in [2], where the effect of fracture occurs only in tension.

ACKNOWLEDGEMENT

This work is supported by the Marie Curie Career Integration Grant, Grant reference FP7-PEOPLE-2011-CIG-CompHydraulFrac.

REFERENCES

- [1] B. Bourdin, G.A. Francfort, and J.J. Marigo. The variational approach to fracture. *Journal of Elasticity*, 91:5–148, 2008.
- [2] C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Engineering*, 83:1273–1311, 2010.
- [3] M. Hofacker and C. Miehe. A phase field model of dynamic fracture: Robust field updates for the analysis of complex crack patterns. *International Journal for Numerical Methods in Engineering*, 93:276–301, 2012.
- [4] M. J. Borden, C. V. Verhoosel, M. A. Scott, T. J.R. Hughes, and C. M. Landis. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 217–220(0):77–95, 2012.
- [5] C. Cecka, A. J. Lew, and E. Darve. Assembly of finite element methods on graphics processors. *International Journal for Numerical Methods in Engineering*, 85(5):640–669, 2011.
- [6] D. B. Kirk and W. W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. Elsevier, Amsterdam, 2010.