ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Massive parallelization of the phase field formulation for crack propagation with time adaptivity

Vahid Ziaei-Rad[a], Yongxing Shen[b,*]

[a] *Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya (UPC BarcelonaTech), 08034 Barcelona, Spain*

[b] *University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, 200240, China*

Abstract

The phase field method has proven to be an important tool in computational fracture mechanics in that it does not require complicated crack tracking and is able to predict crack nucleation and branching. However, the computational cost of such a method is high due to a small regularization length parameter, which in turn restricts the maximum element size that can be used in a finite element mesh. In this work we developed a massively parallel algorithm on the graphical processing unit (GPU) to alleviate this difficulty in the case of dynamic brittle fracture. In particular, we adopted the standard finite element method on an unstructured mesh combined with second order explicit integrators. To ensure stability, we designed a time adaptivity strategy to account for the decreasing critical time step during the evolution of the fields. We demonstrated the performance of the GPU-implemented phase field models by means of representative numerical examples, with which we studied the effect of the artificial viscosity, an artificial parameter to be input, and compared the crack path branching predictions from three popular phase field models. Finally, we verified the method with convergence studies and performed a scalability study to demonstrate the desired linear scaling of the program in terms of the wall time per physical time as a function of the number of degrees of freedom.

Keywords: Phase field; GPU parallelization; Explicit methods; Fracture mechanics; Time adaptivity

1. Introduction

Understanding and predicting fracture and failure in materials is important in engineering designs. Computational modeling is a way to study the fracture phenomena, especially in cases in which experiments are impractical or too expensive. As a consequence, a wide variety of fracture models have been developed. In the case of brittle fracture, many of such models were developed based on Griffith's theory in which crack nucleation and propagation are determined by a critical value of the energy release rate. Many numerical approaches have been constructed with the standard finite element methods (FEMs) [1] or the extended finite element method (XFEM) [2–5] in conjunction with

* Corresponding author.
E-mail addresses: vahid.ziaei@upc.edu (V. Ziaei-Rad), yongxing.shen@sjtu.edu.cn (Y. Shen).

http://dx.doi.org/10.1016/j.cma.2016.04.013
0045-7825/

Griffith-type fracture models, such as the approach presented in [6]. These approaches explicitly represent cracks as discontinuities. They require: (1) either always adjusting the mesh in traditional FEMs [1] or introducing enrichments like the XFEM; and (2) extra input to predict crack nucleation and branching.

Based on energy minimization, the variational theory of fracture was proposed by Francfort and Marigo [7] to formulate brittle fracture, which takes into account both bulk elastic energy and the surface energy due to creation of cracks. Capable of predicting crack initiation, this class of approaches has attracted attention in the mathematicians' community [8–10].

The regularized version of the variational theory of fracture was introduced by Bourdin et al. [11], which was later adopted by the engineers' community as a formulation more suitable for numerical simulations. Later the *phase field* formulation of fracture has become a synonym and a more popular name for this class of methods, see, e.g., [12–14].

In essence, phase field models for fracture employ a continuous field variable, called the *phase field*, to represent cracks. Ambrosio and Tortorelli [15,16], Chambolle [17], Giacomini [18], Bourdin et al. [11], Hakim and Karma [19], and da Silva et al. [20] studied the relation between the phase field formulation and its sharp crack limit.

The main 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 explicit descriptions of cracks, phase field descriptions do not require explicitly tracking the discontinuities in the displacement field. This significantly reduces implementation complexity, and is anticipated to be particularly advantageous when multiple branching and merging cracks are considered in three dimensions [12,21]. While the phase field method has mainly been employed to study quasi-static brittle fracture [19,21–23], it has also been extended successfully to dynamic problems [20,24–28] and ductile fracture [29–31]. Moreover, more sophisticated models for the phase field approximation have been developed which lead to a higher regularity of the phase field solution than H^1, e.g., H^2 for [32] and [33]. See also [34,35] for fracture in plates and shells simulated with the phase field model.

Despite the growing literature in the application of phase field modeling of fracture, the computational cost of such models is sensitive to the choice of the regularization parameter in conjunction with the mesh size, as the mesh has to be fine enough to resolve high gradients of the phase field appearing in the transition zones between cracked and un-cracked materials, whose width is on the order of the regularization parameter [12]. This parameter can be interpreted as either a mathematical construction or an intrinsic material parameter. In this paper we will adopt the first meaning in subsequent sections. Strictly speaking, even in the quasi-static setting, the limit of gradient flows for similar Ambrosio–Tortorelli functionals as this regularization parameter tends to 0 is still a difficult unresolved problem [36,37]. The general understanding is that it has to be small enough compared to the characteristic length of the computational domain to make the simulation meaningful. Numerically resolving this regularization length scale is one of the main computational challenges of implementing such models.

Recently, graphics processing units (GPUs) which are capable of massive parallelization have had success in accelerating many numerical computations. Therefore, massively parallel programming with the GPU is a promising solution for the problem of high computational costs of the phase field methods. Here we mention Cecka et al. [38] as one of the early applications of the GPU to finite element methods.

In this paper, we present and analyze an explicit algorithm to solve for a rate-dependent formulation of the phase field modeling of brittle fracture on NVIDIA GPUs using the Compute Unified Device Architecture (CUDA). The formulation is based on dynamic force balance coupled with a gradient-type phase field evolution law due to [12]. We adopt the standard finite element method on unstructured meshes and second-order explicit time integrators for the implementation. We then develop a massively parallel program to implement the algorithm on a GPU.

Essentially the GPU favors the same or similar operations on a massive collection of data. In this context, the updating of nodal finite element data (displacement, velocity, and phase field) naturally fits this requirement. In this work we will provide a detailed algorithm to demonstrate how such a simulation can be done with GPU.

As is common to explicit methods, a critical time step is essential to ensure stability. The coupled problem at hand can be divided into an elastodynamic half problem and a phase field half problem, and the critical time step is the smaller of those of the half problems. As time elapses, the critical time step for the elastodynamic half problem increases but that for the phase field half problem decreases. Beginning from some instant, the critical time step for the entire problem decreases with time. Hence, an efficient implementation requires an efficient lower bound computation for the critical time step, which we have accomplished by taking advantage of the GPU architecture. This time adaptivity scheme will be elaborated in Section 5. With regard to the choice between explicit and implicit schemes, we refer the reader to Keyes et al. [39] for a thorough discussion. In our case, as we will demonstrate, we

Please cite this article in press as: V. Ziaei-Rad, Y. Shen, Massive parallelization of the phase field formulation for crack propagation with time adaptivity, Comput. Methods Appl. Mech. Engrg. (2016), http://dx.doi.org/10.1016/j.cma.2016.04.013