Implementation of Longman's Method for Numerical Integration on Graphics Hardware

Euclides Mesquita[1], Josué Labaki[2] and Luiz Otávio Saraiva Ferreira[3]

1. Professor, Dept. of Computational Mechanics- FEM, State University at Campinas, São Paulo, Brazil, e-mail: euclides@fem.unicamp.br

2. Ph.D. Student, Mechanical Engineering, State University at Campinas, São Paulo, Brazil, e-mail: labaki@fem.unicamp.br

3. Professor, Dept. of Computational Mechanics- FEM, State University at Campinas, São Paulo, Brazil, e-mail: lotavio@fem.unicamp.br

There is a growing trend towards solving problems of computational mechanics by parallelization strategies. The more traditional approach is to implement the parallelization procedures on CPUs based on the MPI or Open MP paradigms. Recent efforts have been made to implement computational tasks, which are amenable to parallelization on graphics hardware (GPU) [1]. This work addresses the implementation of the numerical integration scheme proposed by Longman [2] to perform improper integration of oscillating and decaying functions on a GPU. The need for numerical realization of integrals extending to infinity of decaying and oscillatory functions appear in many classical problems of mathematical physics, such as inverse Fourier and Hankel integral transforms. The Longman strategy is well suited to parallelization schemes. It consists mainly in two steps. Initially a given number of 'volumes' must be integrated. In this step each volume integration may be performed independently. In the second step a classical data reduction method, for which there is a standard parallelization procedure, is applied to the integrated volumes leading to the result of the integration. In the present work the implementation described can handle simultaneously two improper integrations, for instance, real and imaginary parts of the integral transforms. The computational efficiency of distinct levels of parallelization is addressed. The volumes integration can be parallelized. Within each volume standard Gaussian quadrature is applied, which, in turn, can also be parallelized. The code was developed on a NVIDIA CUDA programming environment and executed on a GeForce GTX 280 graphics card in a regular Intel Core2Duo CPU. The accuracy and efficiency of the implemented strategies are investigated using an improper integration with known close form solution.

[1] Owens, J. D. et all., "A Survey of General-Purpose Computation on Graphics Hardware", Computer Graphics, **26,** 1 (2007), pp. 80-113.

[2] Longman, I. M., "Note on the Computing of Infinite Integrals of Oscillatory Functions". Proc. Camb. Phil. Soc. **52** (1956), pp. 764–768.