

Universidade Federal de Minas Gerais
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica
Plano de Curso de Doutorado

Solução de métodos sem malha usando o paralelismo das unidades de processamento gráfico (GPUs)

Lucas Pantuza Amorim

Orientador: Prof. Renato Cardoso Mesquita
Área de Concentração: Sistemas de Computação e Telecomunicações
Linhas de Pesquisa: Eletromagnetismo Aplicado (APE)

Belo Horizonte, Fevereiro de 2012

Resumo

Apresentaremos aqui, uma proposta de pesquisa a ser realizada no programa de doutorado do Programa de Pós-Graduação em Engenharia Elétrica. Como alternativa aos métodos computacionais que trazem soluções aproximadas, como o método dos elementos finitos (FEM) [?], existe o método sem malha (*Meshless*), que basicamente oferece resultados aproximados tão bons quanto sem ter que tratar problemas inerentes à geração da malha. Existe, no entanto, um alto custo de processamento neste método, que vem sendo tratado usando-se algoritmos paralelos, como nos trabalhos de [?], [?], [?] e [?].

Nossa proposta é estender a pesquisa realizada por [?], que implementou uma solução paralela em CPUs do método sem malha “Local Petrov-Galerkin” para a arquitetura CUDA da nVidia [?], fazendo uso dos múltiplos núcleos das modernas placas de vídeo (GPUs). A idéia é tratarmos tanto problemas bidimensionais (2D) quanto tridimensionais (3D).

1 Objetivo

O objetivo desta pesquisa é apresentar a solução de métodos sem malha em unidade de processamento gráfico (GPUs). Para isso, partiremos da solução apresentada por [?] adaptando os algoritmos que proveem o paralelismo em processadores com múltiplos núcleos para a arquitetura CUDA da nVidia [?]. Ainda, como objetivo secundário, estenderemos a solução para problemas em 3D.

2 Relevância da Contribuição

Vários fenômenos físicos tratados pelas Engenharias e pela Física podem ser descritos por equações diferenciais parciais em conjunto com restrições adicionais, as chamadas condições de contorno ou condições de fronteira [?].

Quando tratamos problemas de geometria simples podemos, em alguns casos, resolvê-lo analiticamente. Para todos os demais precisamos utilizar métodos computacionais que trazem soluções aproximadas, como o método dos elementos finitos (FEM) [?] e o método das diferenças finitas (FDM) [?]. Ambos necessitam de uma malha para representar o domínio do problema e, por isso, a precisão dos resultados numéricos destes métodos está diretamente ligada à boa formação dos elementos que a constituem. Uma alternativa é o método sem malha, que não necessita de uma malha para representar o domínio e depende, exclusivamente, de uma nuvem de um conjunto de nós independentes.

Existem diversos métodos sem malhas, como por exemplo os mais tradicionais “*Element-Free Galerkin (EFG)*” [?] e “*Meshless Local Petrov-Galerkin (MLPG)*” [?] e outros como o “*Radial Point Interpolation Method (RPIM)*” [?].

Em particular, [?] apresentou um trabalho utilizando o MLPG. Durante a implementação da solução, percebeu-se que as contribuições dos nós para o sistema matricial global são independentes. Trata-se, portanto, do cumprimento de uma das premissas base para o paralelismo na solução do problema, alternativa testada e validada utilizando processadores de 4 núcleos.

Figura 1: Comparativo entre GPUs e CPUs em termos de *Floating point operations* (FLOPs). Gráfico retirado de [?]

Em alguns casos, diferente da abordagem seguida por [?], aplicações científicas são adaptadas para execução nas modernas unidade de processamento gráfico (GPUs) ou mesmo clusters de GPUs, cuja capacidade de processamento tem evoluído a um ritmo exponencial [?] (Figura 1). Tradicionalmente, a diferença fundamental entre os processadores (CPUs) e GPUs é o fato de que as CPUs são otimizadas para cálculos sequenciais (executando aplicativos diversos), enquanto as GPUs são otimizadas para cálculos massivamente paralelos. Há um tempo atrás essa divisão era bem mais nítida, já que as placas 3D processavam apenas triângulos, texturas e efeitos simples. Entretanto, com a introdução do uso de *shaders* (que nada mais são do que pequenos aplicativos destinados a executarem tarefas específicas na composição das cenas) elas ganharam a capacidade de também executarem código sequencial, assim como os processadores [?]. Desta forma, além do processamento de gráficos para o qual originalmente foram desenvolvidas, as GPUs são capazes de assumir o processamento de aplicações mais elaboradas que eram exclusivamente realizados em CPUs.

Existe, então, uma nova abordagem para solução de problemas usando métodos sem malha: com o objetivo de diminuir o tempo de processamento até a solução do problema segundo quaisquer um dos métodos já citados, autores como [?], [?], [?] veem trabalhando com processamento paralelo em CPUs, mas pouco foi feito explorando GPUs, como o trabalho apresentado por [?] usando RPIM.

2.1 Linguagem de Programação

Com o novo potencial das GPUs, os fabricantes investiram em ferramentas de desenvolvimento capazes de liberar o potencial oculto das placas [?]. A *nVidia* apresentou o CUDA, a ATI o Brook+, e em seguida surgiu o OpenCL, um *framework* de computação desenhado para possibilitar o desenvolvimento de software para plataformas e ambientes heterogêneos [?].

O CUDA, particularmente, permite acessar as funções das placas usando funções em *C* (*C* for CUDA), o que o torna uma ferramenta relativamente simples para desenvolvedores familiarizados com a linguagem. Existem também *wrappers* que permitem desenvolver em Java (jCUDA), *C#* (CUDA.NET) ou até mesmo Python (PyCUDA) [?]. Embora os potenciais benefícios sejam grandes, as dificuldades também são, já que o desenvolvimento de aplicativos massivamente paralelos exige não apenas um bom domínio da linguagem escolhida, mas toda uma nova metodologia de desenvolvimento, especialmente em situações onde o paralelismo não é óbvio.

2.2 Problemas em três dimensões (3D)

Independente do método sem malha adotado, é notório que a maior parte dos trabalhos são desenvolvidos utilizando problemas de geometria em 2D. Isto se deve ao fato de ter implementação muito mais simples e, muitas vezes, ser suficiente para testar a hipótese do trabalho. No entanto, em problemas reais, precisamos muitas vezes trabalhar com problemas 3D. Existem trabalhos, como o desenvolvido por [?] que tratam geometrias em 3D, mas ainda há muito o que se explorar cientificamente, sobretudo com propostas de soluções de processamento paralelo em GPUs, objetivo desta pesquisa.

3 Metodologia, Disciplinas a Serem Cursadas e Cronograma

1. Fundamentação teórica para realização da pesquisa cursando disciplinas relacionadas a métodos sem malha, eletromagnetismo computacional e processamento paralelo:
 - EEE933D: Tóp. Especiais em Eng. Comp. e Teleco, Métodos sem Malhas;
 - EEE933A: Tóp. Especiais em Eng. Comp. e Teleco, Planejamento e Análise de Experimentos;
 - DCC865: Projeto e Análise de Algoritmos;
 - DCC837: Programação Paralela;
2. Pesquisa bibliográfica em Métodos sem Malha e outros métodos e suas implementações em GPUs, como por exemplo:
 - *Total Lagrangian Explicit Dynamic algorithm (TLED, FEM)* [?];
 - *Parallel realisation of the element-by-element FEM* [?];
 - *Modified radial point interpolation method (Modified RPIM, Meshless)* [?].
3. Estudo exaustivo de códigos OpenCL e Cuda;
4. Proposição de uma abordagem de implementação para Métodos sem Malha adaptada para execução em GPUs;
5. Exame de qualificação;
6. Extensão da solução para problemas em três dimensões;
7. Apresentação dos resultados para a comunidade científica que tratam de métodos sem malha e/ou processamento paralelo em GPUs, através de publicação em revista.
8. Doutorado sanduíche;
9. Redação da tese de doutorado;
10. Defesa da tese de doutorado.

Tarefas	Semestres							
	1	2	3	4	5	6	7	8
1	•	•	•					
2		•	•	•				
3	•	•	•	•				
4			•	•				
5				•				
6					•	•		
7				•	•	•	•	
8					•	•		
9					•	•	•	•
10								•

Tabela 1: Cronograma

1. Cumprimento das disciplinas
2. Projeto de Tese (I e II)
3. Estudo exaustivo de códigos OpenCL e Cuda
4. Experimentos
5. Exame de qualificação
6. Submissão de resultados a uma revista científica
7. Doutorado sanduíche
8. Redação da tese de doutorado
9. Defesa da tese de doutorado

4 Regime de dedicação e recursos

Durante todo o curso será adotado o regime de dedicação exclusiva, com carga horária cumprida nos laboratórios de otimização e projeto assistido por computador (LOPAC), onde os recursos de infraestrutura estarão disponíveis.