

# Aplicação de Bibliotecas para Cômputo Numérico de Equações Diferenciais em Arquitetura CUDA

**Gledson Oliveira<sup>1</sup>; Angelo Amâncio Duarte<sup>2</sup>**

1. Bolsista PROBIC/UEFS, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana, e-mail: [gledsdson.1@gmail.com](mailto:gledsdson.1@gmail.com)

2. Ângelo Amâncio Duarte, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: [angeloduarte@uefs.br](mailto:angeloduarte@uefs.br)

**PALAVRAS-CHAVE:** Bibliotecas de Cômputo Numérico; computação Paralela; CUDA

## INTRODUÇÃO

Abordar numericamente problemas que utilizam equações diferenciais ordinárias (ODE - *Ordinary Differential Equation*) tornou-se inevitável, visto que atualmente muitos deles atingiram tal complexidade que o uso de métodos analíticos tornou-se inviável (Sandes, 2012). Um tratamento computacional a partir de bibliotecas de cômputo numérico são adequadas a esta situação, pois elas possuem um conjunto de algoritmos prontos para encontrar uma solução de diversos casos, otimizando assim o trabalho de produção.

Atualmente, está em desenvolvimento no Laboratório de Computação de Alto Desempenho (LaCAD) da Universidade Estadual de Feira de Santana (UEFS), um trabalho de pesquisa com bibliotecas numéricas de alto desempenho que solucionam numericamente ODEs, usando como caso de estudo um sistema de equações diferenciais de segunda ordem que calculam a distância relativa entre um detrito espacial e um satélite. Este modelo é utilizado na astronomia para calcular possibilidade de colisões entre satélites e detritos (Jesus, 2012). Tal problema foi tratado utilizando uma biblioteca em serial de cômputo numérico e paralelizado utilizando arquitetura multi-core de memória compartilhada com OpenMP, obtendo assim melhor desempenho de computo.

Um outro caminho para tratar esta situação é utilizando paralelismo em GPUs. Por causa do altíssimo poder computacional dessa arquitetura que é naturalmente paralela, é possível manipular enormes cargas de trabalho e acelerar o cômputo de métodos numérico com uso de uma *Application Programming Interface* (API). Devido a estas características, diversos pesquisadores de todo o mundo estão desenvolvendo algoritmos para esta plataforma, usando-a para cálculos matemáticos de propósito geral. (Buriol, 2009)

O CUDA (Compute Unified Device Architecture) é um modelo de programação desenvolvidos para utilizar de forma eficiente o poder das GPUs. Esta tecnologia foi desenvolvida pela NVIDIA e permite a utilização de unidade de processamento gráficos (GPUs) da NVIDIA para a computação paralela. No site da NVIDIA ([http://www.NVIDIA.com.br/object/CUDA\\_home\\_br.html](http://www.NVIDIA.com.br/object/CUDA_home_br.html)) encontram-se diversas pesquisas que utilizam CUDA para uma aplicação em GPU como simulação de efeitos físicos, exploração de gás e petróleo, processamento de imagens na área médica, pesquisa científica e muitos outros. (Buriol, 2009)

Nesse sentido, este plano de trabalho objetiva demonstrar o estudo de bibliotecas para cômputo numérico de ODEs em uma aplicação paralela em GPUs utilizando a arquitetura CUDA. Foram usados como base de teste programas já conhecidos e desenvolvidos pela equipe do laboratório para a resolução do problema da astronomia, paralelizando-os no ambiente de GPU. Nesse sentido, foi realizado uma comparação

entre o desempenho das aplicações em GPU com outra plataformas de cômputo paralelo já testada, a de memória compartilhada utilizando OpenMP.

## MATERIAL E MÉTODOS OU METODOLOGIA

A pesquisa fez uso dos computadores instalados no Laboratório de Computação de Alto Desempenho (LaCAD), localizado na sala I12 no LABOTEC3, prédio de engenharia de computação.

A metodologia utilizada no desenvolvimento desta pesquisa envolveu adaptar os códigos que utilizam bibliotecas numéricas para solucionar Equações Diferenciais dominadas pelo LaCAD para fazer uso de cômputo paralelo utilizando GPUs utilizando uma placa TESTA da NVIDIA que se encontra em posse do laboratório. Foi feita uma comparação entre os resultados obtidos em GPU com a abordagem para cômputo paralelo utilizando arquitetura multi-core de memória compartilhada.

. Foi feito também, escrita do método e resultados obtidos, divulgação dos tutoriais no site do LaCAD.

## RESULTADOS E DISCUSSÃO

Foi encontrada somente 1 biblioteca que se adequa ao que estava sendo buscado, a ODEINT (AHNERT e MULANSKY, 2011). ODEINT é uma biblioteca desenvolvida especificamente para cálculo numérico de Equações diferenciais ordinárias. Ela possui em seus pacotes, diversos métodos numéricos como o de Euler e o de Runge-Kutta. Ela possui também embutido em sua implementação funções que facilitam a adaptação do código a arquiteturas de memória compartilhada e GPUs o que demonstrou ser uma grande vantagem para quem deseja adaptar um código nestas arquiteturas.

Para estudar o comportamento da biblioteca foi utilizado o seguinte sistema de equações diferenciais de segunda ordem como caso de estudo:

$$\begin{aligned}\ddot{x} - 2\omega\dot{y} + 3\omega^2x &= f_x, \\ \ddot{y} + 2\omega\dot{x} &= f_y, \\ \ddot{z} + 3\omega^2z &= f_z.\end{aligned}$$

**Figura 1:** Equações utilizadas como caso estudo

Este sistema matemático é um modelo de problema da astronomia (JESUS, 2012) para realizar cálculo de manobras evasivas entre um satélite e um detrito espacial por meio do cálculo da distância relativa entre um satélite qualquer e um detrito a uma determinada distância do centro da terra. Estas equações foram escolhidas para estudo por possuírem valores padrões da saída conhecidos oriundos de uma pesquisa que mostra a solução analítica do mesmo, possibilitando validar com mais precisão os resultados encontrados por meio do uso das bibliotecas. Outro motivo que levou a escolha destas equações foi aproveitar um problema de pesquisa em andamento no LaCAD com parceria com o professor Antônio Delson de Jesus do Departamento de Física da UEFS, para o estudo de manobras de evasivas de veículos espaciais.

Nesse sentido, foi possível realizar testes de comparação de tempo execução da ODEINT usando o método numérico de Runge Kutta (MATHEWS), por se tratar de um método numérico já trabalhado em pesquisas anteriormente desenvolvidas no laboratório e com isso já se possuía resultados de medição de tempo de execução em

ambientes de cômputo serial e multicore fazendo uso de OpenMP (CHANDRA, 2001). Nesse sentido, foi possível realizar um comparativo entre os resultados anteriormente obtidos, com a medição de tempo de execução em CUDA.

Arquitetura	1° Execução	2° Execução	3° Execução
Serial	0.13	0.12	0.14
Multicore	0.47	0.37	0.41
GPU	11.38	11.40	11.30

Tabela 1: Comparativo de tempo de execução da ODEINT

Nesta tabela, é possível observar o tempo de cômputo em serial, multicore e em GPU na arquitetura CUDA da bibliotecas ODEINT para o algoritmos numéricos de Runge-Kutta de 4ª ordem. No experimento foram utilizados um intervalo tempo de 0 a 2000 segundos com uma precisão de  $10^{-4}$  no passo do tempo de integração. Assim, foram feitos um total de 100000 iterações para alcançar o resultado.

Com base neste experimento, percebeu-se que o melhor desempenho da ODEINT é quando a mesma é executada em serial, ou seja, sem fazer uso de paralelismo nas arquiteturas descritas. Este resultado demonstra que a mesma não possui boa viabilidade quando se pretende trabalhar com computação de alto desempenho, mesmo apresentando boa exatidão e precisão, já que a mesma apresentou uma precisão média de  $10^{13}$ , porque com a propagação do erro oriundo dos métodos numéricos testados, só foi encontrado uma divergência entre as soluções analítica e numérica após a 13ª casa decimal.

Percebeu-se também que em uma abordagem em ambientes de cômputo em paralelo a ODEINT demonstrou maior desempenho de tempo quando executada em um arquitetura multicore, uma vez que o tempo de execução (Tabela 1) da mesma foi menor que em GPU utilizando CUDA. Deste modo, uma vez que a diferença de tempo entre as arquiteturas paralelas estudadas é significativa, torna-se perfeitamente cabível um estudo mais aprofundado a fim de verificar os motivos que levaram a isto.

Para realizar um estudo mais aprofundado do cômputo dos mesmos programas, a fim de identificar a causa da grande diferença de tempo, foi utilizado o NVPROF. O NVPROF é uma ferramenta desenvolvida pela NVIDIA para facilitar o entendimento e a otimização de aplicações em GPU por meio do monitoramento do tempo de execução da mesma. Desta forma, obteve-se os resultados da figura 1.

```

==3399== API calls:
Time(%) Time Calls Avg Min Max Name
98.86% 13.2455s 781920 16.939us 8.8370us 152.42ms cudaMemcpyAsync
1.14% 152.37ms 1 152.37ms 152.37ms 152.37ms cudaMalloc
0.00% 356.25us 83 4.2920us 291ns 153.73us cuDeviceGetAttribute
0.00% 75.906us 1 75.906us 75.906us 75.906us cuDeviceGetName
0.00% 58.682us 1 58.682us 58.682us 58.682us cudaLaunch
0.00% 51.138us 1 51.138us 51.138us 51.138us cuDeviceTotalMem
0.00% 28.041us 4 7.0100us 4.6420us 13.800us cudaFuncGetAttributes
0.00% 12.982us 1 12.982us 12.982us 12.982us cudaSetupArgument
0.00% 11.227us 11 1.0200us 636ns 1.1960us cuDeviceGetAttribute
0.00% 7.1330us 2 3.5660us 1.2280us 5.9050us cuDeviceGetCount
0.00% 6.9020us 3 2.3000us 637ns 5.2730us cudaGetDevice
0.00% 4.7560us 1 4.7560us 4.7560us 4.7560us cudaEventCreateWithFlags
0.00% 4.7420us 1 4.7420us 4.7420us 4.7420us cudaConfigureCall
0.00% 3.3640us 1 3.3640us 3.3640us 3.3640us cudaEventDestroy
0.00% 2.7670us 1 2.7670us 2.7670us 2.7670us cudaEventRecord
0.00% 1.5590us 2 779ns 346ns 1.2130us cuDeviceGet

```

Figura 1: Resultado da execução do NVProf

Com base nas informações obtidas na Figura 1, pode-se perceber que a maior parte de tempo gasto para executar a aplicação não foi com o cômputo da mesma, e sim com a transferência de dados da CPU para o device. Desta forma, uma boa estratégia que permitiria ganho de desempenho seria trabalhar com formas mais eficientes de transferência de dados entre a CPU e o device, como estudos de métodos de transferências de dados assíncronos. Isso pode ser investigado com mais profundidade em trabalhos futuros.

## **CONSIDERAÇÕES FINAIS**

O objetivo deste projeto foi pesquisar, selecionar e estudar bibliotecas públicas para cômputo numérico de equações diferenciais em arquitetura CUDA, que fizessem uso de computação de alto desempenho e que pudessem ser aplicadas a uma gama variada de problemas. Nesse sentido, foi pesquisado e estudado com base na literatura, quais bibliotecas se encaixavam a essa descrição e que pudesse ser utilizadas no LaCAD. Sendo assim, foi selecionada a ODEINT, por ser a única biblioteca pública disponível atualmente que se encaixava com características buscadas. Logo, a ODEINT foi definida como biblioteca padrão para para uso no laboratório em problemas que demandam solução de equações diferenciais ordinárias em CUDA.

## **REFERÊNCIAS**

AHNERT, K.; MULANSKY, M.; “Odeint-solving ordinary differential equations in c++,” arXiv preprint arXiv:1110.3397, 2011.

SCHEPEKE, C. Bibliotecas Numéricas Paralelas para o Cálculo Numérico. Porto Alegre: Universidade Federal do Rio Grande do Sul, Instituto de informática, 2008.

JESUS, A. D. C. RIBEIRO, R. S., ROSSI, A. NETO, E. V. Evasive Maneuvers in Space Debris Environment and Technological Parameters, Feira de Santana, p. 1-4, August 2012.

AHNERT, K. MULANSKY, M, Chapter 1. Boost.Numeric.Odeint. Disponível em:<<http://headmyshoulder.github.io/odeint-v2/doc/index.html>>. Data de acesso 25, Jul. 2016. CHANDRA, R.; DAGUM, L.; KOHR, D.; MAYDAN, D.; MCDONALD, F.;MENON, R. Parallel Programming in OpenMP. San Francisco: Morgan Kaufmann Publishers, 2001. 221 p.

Buriol, Tiago Martinuzzi. Argenta, Marco André. ACELERANDO O DESENVOLVIMENTO E O PROCESSAMENTO DE ANÁLISES NUMÉRICAS COMPUTACIONAIS UTILIZANDO PYTHON E CUDA, Curitiba, p. 1-4, Jan 2009.

Mathews, J. H.; Fink, K. K. Numerical Methods Using Matlab, 4th Edithion, Upper Saddle River, New Jersey, USA: Prentice-Hall Inc. 662.