Navier-Stokes: visualização em tempo real

Raphael Mesquita [1], **Pedro Von Zuben**[1], **Juliana Valerio**[1], **Silvana Rossetto**[1]

[1]Instituto de Computação - CCMN/UFRJ

{raphaelfcm,pedrohfvz}@ic.ufrj.br

Resumo. *Simulações de fluidos em tempo real requerem cálculos rápidos e precisos do modelo físico descrito pelas equações de Navier-Stokes. Neste trabalho implementamos as equações de Navier-Stokes utilizando técnicas de programação paralela em GPUs, avaliamos o desempenho alcançado e apresentamos uma forma de visualizar estas simulações.*

1. Introdução

Simulação de fluidos, apesar de ser um assunto bastante explorado, ainda traz desafios teóricos, numéricos e computacionais. Um deles está relacionado à necessidade de resolver numericamente as equações que descrevem o escoamento e visualizar a solução no mesmo instante. Tal desafio, se superado, pode, por exemplo, auxiliar pesquisadores na indústria em tomada de decisões e ser utilizado como ferramenta pedagógica para o ensino de fluidos, na área da Física.

Portanto, este trabalho visa desenvolver um programa de computador que use a programação paralela em GPU, com a tecnologia CUDA, para simular e visualizar escoamentos de fluidos newtonianos e incompressíveis através das equações de Navier-Stokes. Estendemos o trabalho final de curso de Panno [2022], utilizando mais eficientemente os recursos da GPU compatível com CUDA, permitindo a visualização em tempo real.

2. Equações de Navier-Stokes e Modelagem Numérica Bidimensional

A dinâmica de um fluido newtoniano e incompressível é descrita pelas equações de momento e conservação de massa conhecidas como equações Navier-Stokes (Fox et al. [2014]):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial (u^2)}{\partial x} - \frac{\partial (uv)}{\partial y} + g_x \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial (v^2)}{\partial y} - \frac{\partial (uv)}{\partial x} + g_y \tag{3}$$

nas quais u e v são os componentes horizontal e vertical da velocidade, p é a pressão, g_x e g_y são as componentes horizontal e vertical da força da gravidade, Re é o número adimensional de Reynolds. Completamos a formulação matemática do problema com as condições iniciais e de **Contorno de Não-Deslizamento**, como em Meneguci [2011].

Para a discretização, utilizaremos o esquema de diferenças finitas como o proposto por Griebel et al. [1998]. O domínio é discretizado, no caso bidimensional, em uma malha do tipo deslocada, na qual as componentes da velocidade para cada dimensão e a pressão são calculadas em pontos diferentes. Essa escolha evita oscilações possíveis na pressão. O esquema é apresentado com maiores detalhes em Griebel et al. [1998].

3. Equação de Poisson para pressão e implementação *Red-Black-SOR*

A equação de Poisson é uma consequência de rearranjos das equações de Navier-Stokes que resulta em um sistema linear a ser resolvido para obter a pressão em cada célula da malha. Para resolvê-lo, faz-se uso do método iterativo *successive-over-relaxation* (SOR) com um esquema de coloração *Red-Black* para as células internas da malha, denominado *Red-Black-SOR* (Konstantinidis and Cotronis [2021]). Esse método foi escolhido por ser adequado para implementações paralelas.

Um dos aspectos da tecnologia CUDA é que o acesso à memória global da GPU é feito de forma coalescente. Então, quanto menor a esparsidade entre os dados requisitados, mais eficiente será o acesso à memória global. Neste trabalho, aplicamos esse conceito na implementação do *Red-Black-SOR* dividindo a malha em duas matrizes distintas, como em Konstantinidis and Cotronis [2021]. O ganho de desempenho resultado dessa mudança, em relação à versão original, é apresentado na seção 4.

4. Resultados e Conclusão

Nesta seção, são apresentados os resultados obtidos com experimentos do problema da cavidade, proposto por Panno [2022]. Comparamos o tempo de execução das versões sequencial e paralela, originais de Panno [2022], e versão paralela modificada neste trabalho, variando os paramêtros do método Red-Black-SOR: τ (controle do passo no tempo) e ε (precisão do resultado numérico para velocidade e pressão).

Os resultados foram gerados numa máquina com CPU AMD Ryzen 5 1600AF (SixCore) de 3.2GHz e DRAM DDR4 com 16GB de mémoria de 1333MHz. O sistema operacional utilizado foi o Windows 10, 64-bits, e o compilador gcc 6.3.0 com flag -O3. A placa de vídeo utilizada foi a NVIDIA GeForce GTX 1050ti de 1290MHz com 4GB de DRAM GDDR5 de 7008MHz. A versão do compilador nvcc utilizado foi a 11.6.

A malha utilizada foi de 512×512. Para o número de Reynolds e os demais parâmetros que definem o comportamento da resolução numérica foram mantidos valores

	Paralelo original	Paralelo mod. (min)	Sequencial (min)	Speedup (mod.)
$\varepsilon = 10^{-5}; \tau = 0.7$	2.700	2.387	24.193	10.13
$\varepsilon = 10^{-7}; \tau = 0.7$	36.282	25.542	277.389	10.86

Tabela 1. Tempo de Execução e Speedup

fixos com base nos trabalhos de Panno [2022] e Griebel et al. [1998]. Além disso, para as versões paralelas, instanciamos 512 blocos de 512 threads. A Tabela 1 apresenta a média dos tempos de 3 execuções (em minutos) e o *speedup* para cada um dos parâmetros τ e ε, para o código original de Panno [2022] e o código modificado neste trabalho. Obtivemos *speedup* de 10.86 para a execução mais custosa, superando os resultados de Panno [2022].

Além da redução do tempo de execução, foram acrescentadas funcionalidades para a visualização do fluido, onde partículas se deslocam com base nas coordenadas do campo de velocidade em que estão a cada passo de tempo. As Figuras 1 e 2 ilustram a visualização quando a condição de estabilidade definida em Panno [2022] é alcançada.

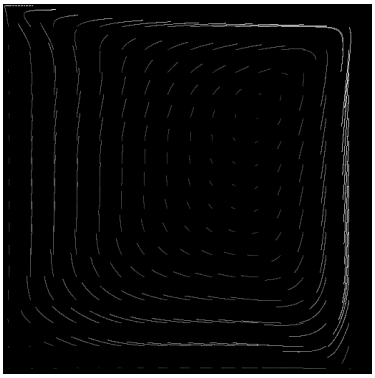

Figura 1. Visualização quadro 200

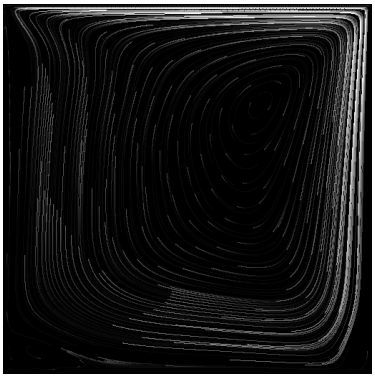

Figura 2. Visualização quadro 2400

No entanto, os cálculos ainda não são rápidos o bastante para os quadros serem renderizados a cada passo de tempo em uma taxa adequada, como 24 quadros por segundo. Portanto, continuaremos este trabalho, estudando técnicas para solucionar este problema, mantendo a corretude da solução numérica e a qualidade da visualização.

Referências

Fox, R. W., McDonald, A. T., and Pritchard, P. J. (2014). *Introdução à Mecânica dos Fluidos*. LTC Editora, Rio de Janeiro, 8 edition.

Griebel, M., T., D., and T., N. (1998). *Numerical Simulation Fluid Dynamics*. SIAM, Philadelphia, PA, 1 edition.

Konstantinidis, E. and Cotronis, Y. (2021). Accelerating the red/black sor method using gpus with cuda, PPAM 2011.

Meneguci, W. d. S. (2011). Implementação de modelos de mecânica dos fluidos computacional em sistemas de *MANY-CORE* usando c+cuda. Master's thesis, UFES.

Panno, M. (2022). Implementação sequencial e paralela das equações de navier-stokes usando c+cuda. Trabalho de conclusão de curso, UFRJ.