

Da modelação até a simulação numérica

Stéphane Clain, Rui Ralha

Centro de Matemática, Universidade do Minho

setembro 2011

Motivações e exemplos

Método dos Volumes Finitos

Computação científica intensiva

Motivações e exemplos

Convecção de um poluente

Um poluente (químico ou radioactivo) de fracção massica α desloca-se num fluido.

$$\partial_t \alpha + \nabla \cdot (V\alpha) - \nabla \cdot (A\nabla \alpha) = s$$

- s é a fonte de poluição
- $V(x, t)$ é a velocidade do fluido
- A é o coeficiente de difusão .



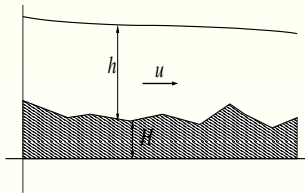
convecção difusão de um poluente

Shallow-water

Problema de água, ondas, inundação, tsunamis.

$$\partial_t h + \nabla \cdot (hu) = 0,$$

$$\partial_t (hu) + \nabla \cdot (hu \otimes u + g \frac{h^2}{2}) = -gh \nabla H$$



- $h(x, t)$ altura da água
- $u(x, t)$ velocidade da água
- g gravidade
- $H(x, t)$ topografia do fundo

Sistema de Euler

Equação dos gases compressíveis sem viscosidade

$$\begin{aligned}\partial_t \rho + \nabla \cdot (\rho u) &= 0, \\ \partial_t (\rho u) + \nabla \cdot (\rho u \otimes u + P \text{Id}) &= g, \\ \partial_t (E) + \nabla \cdot (u(E + P)) &= 0.\end{aligned}$$

- ρ densidade
- u velocidade
- P pressão
- E energia total

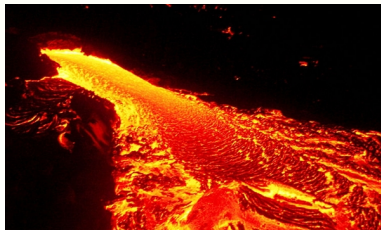


Equação dos fluidos incompressíveis com viscosidade

Água, atmosfera, lava, polímeros,...

$$\begin{aligned}\rho \partial_t(u) + \rho u \nabla u + \nabla P &= \mu \Delta u + g, \\ \nabla \cdot u &= 0.\end{aligned}$$

- ρ densidade
- u velocidade
- P pressão
- μ viscosidade



Equação da fronteira livre

Fogo de foresta, bolha

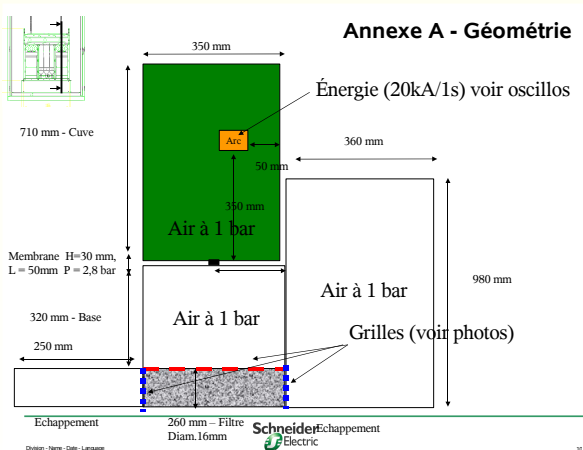
$$\partial_t \phi + u \cdot \nabla \phi + \varepsilon \Delta \phi = 0,$$

$$\Gamma_t = \{x \text{ tal que } \phi(x, t) = 0\}.$$

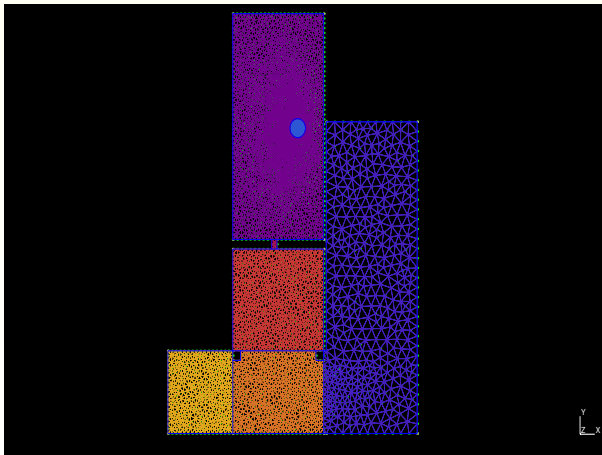
- ϕ função superfície de nível
- u velocidade da frente
- ε coeficiente de curvatura
- Γ_t posição da frente no tempo t .



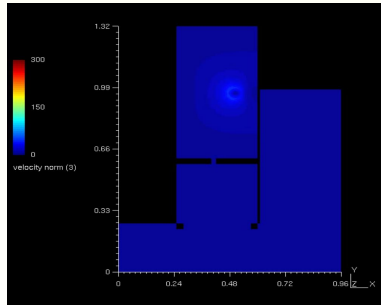
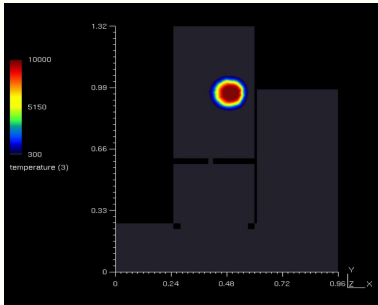
Disjuntor meio tensão .



Malha 2D do disjuntor.



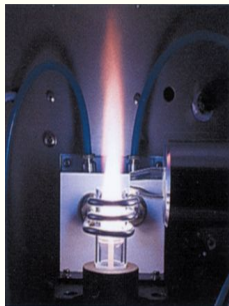
Simulação numérica do disjuntor MV (temperatura e velocidade)



A tocha ICP (Inductively Coupled Plasma)

Objectivo: Simulação numérica da tocha do LAEPT

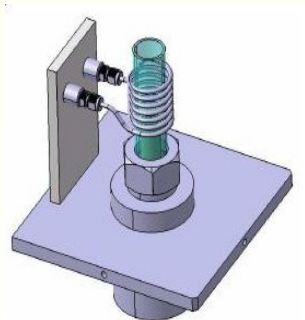
- Frequência 64 MHz.
- Potência 4kW.
- Tensão de 4.5kV sob 600 mA.
- Geometria axisimétrica.
- Injecção do gás com "swirl".
- Dimensões $5\text{ cm} \times 10\text{ cm}$



Aplicação Estudo da radiação na atmosfera de marte ou titã.

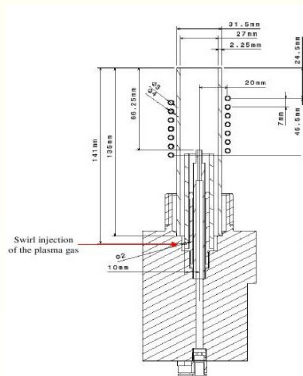
Temperatura da chama com pressão atmosférica

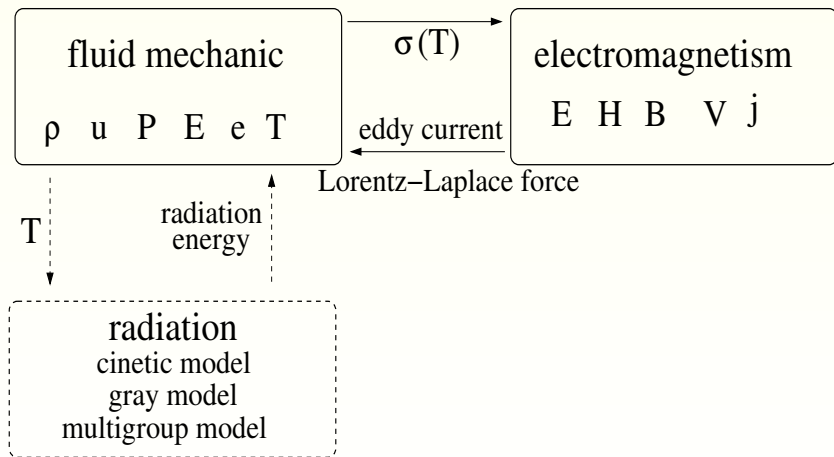
- Argon, Ar 100%: temperatura $\approx 7000 - 8000K$.
- Terra, N_2 80%, O_2 20% : temperatura $\approx 4000 - 5000K$.
- Marte, N_2 3%, CO_2 97% : temperatura $\approx 6000K$.
- Titã, N_2 98%, CH_4 2% : temperatura $\approx 3500 - 4000K$.



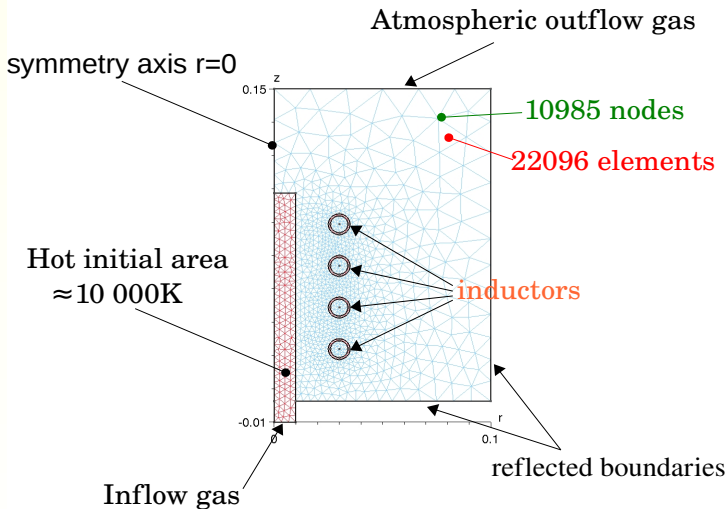
Modelação e simulação da tocha:

- Modelo evolutivo em tempo (simular a ignição).
- Considerar geometrias complexas.
- Simetria de rotação relativamente ao eixo da tocha.
- Tempo de computação razoável (algumas horas).
- Aumentar a complexidade dos modelos, passo a passo.

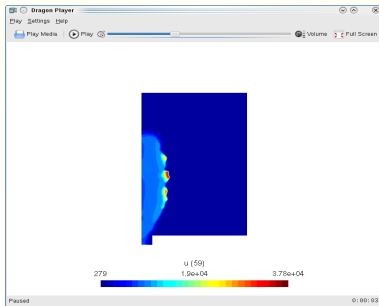




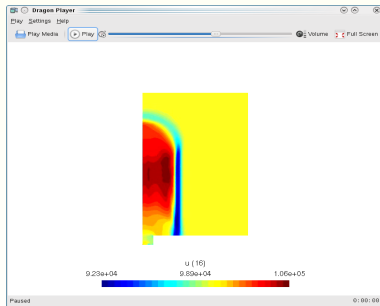
Configuração



Temperatura



Temperatura



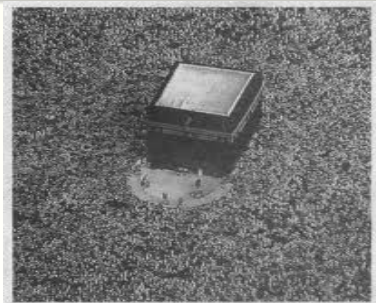
Pressão

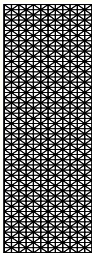
Método dos volumes finitos

Movimento de um grupo de pessoas

$$\partial_t \rho + \nabla(V\rho) = 0, \quad \rho(t = 0, \cdot) = \rho^0(\cdot)$$

- $\rho(x, t)$ densidade da população, $x \in \Omega$, $t > 0$.
- V velocidade das pessoas.

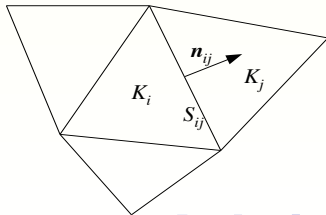




\mathcal{T}_h Malha do domínio poligonal $\Omega \subset \mathbb{R}^2$ em triângulos .

- K_i o elemento de volume.
- $S_{ij} = K_i \cap K_j$ o lado comum.

- $\nu(i)$ conjunto dos índices dos elementos K_j adjacentes a K_i .
- \mathbf{n}_{ij} o vector normal exterior a K_i sobre S_{ij} .



⇒ *Integração sobre o volume K_i*

$$\int_{K_i} \partial_t \rho dx + \int_{K_i} \nabla \cdot (V \rho) dx = 0.$$

⇒ *Fórmula de Green-Gauss*

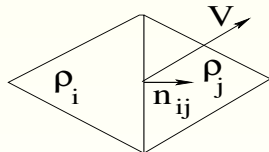
$$\frac{d}{dt} \int_{K_i} \rho(x, t) dx + \int_{\partial K_i} \rho(x, t) V \cdot \mathbf{n} ds = 0.$$

⇒ *Integração entre t^n e $t^{n+1} = t^n + \Delta t$*

$$\begin{aligned} \int_{K_i} \rho(x, t^{n+1}) dx &= \int_{K_i} \rho(x, t^n) dx \\ &- \sum_{j \in \nu(i)} \int_{t^n}^{t^{n+1}} \int_{S_{ij}} \rho(x, t) V \cdot \mathbf{n}_{ij} ds. \end{aligned}$$

Aproximação da função :

$$\rho_i^n \approx \frac{1}{|K_i|} \int_{K_i} \rho(x, t^n) dx,$$

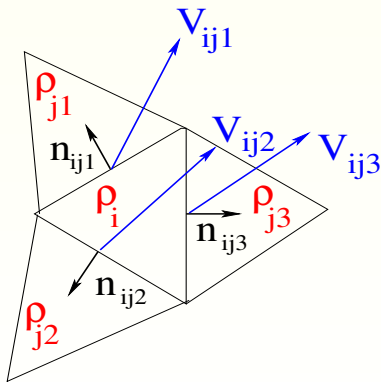


fluxo "upwind"

Aproximação do fluxo :

$$\frac{1}{\Delta t |S_{ij}|} \int_{t^n}^{t^{n+1}} \int_{S_{ij}} \rho(x, t) V \cdot \mathbf{n}_{ij} = [V \cdot \mathbf{n}_{ij}]^+ \rho_i^n + [V \cdot \mathbf{n}_{ij}]^- \rho_j^n$$

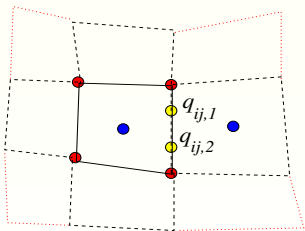
$$\rho_i^{n+1} = \rho_i^n - \sum_{j \in \nu(i)} \frac{\Delta t |S_{ij}|}{|K_i|} ([V_{ij} \cdot n_{ij}]^+ \rho_i^n + [V_{ij} \cdot n_{ij}]^- \rho_j^n).$$



"Design" do esquema volumes finitos

Desafio actual: Esquema V. F. de muito alta ordem

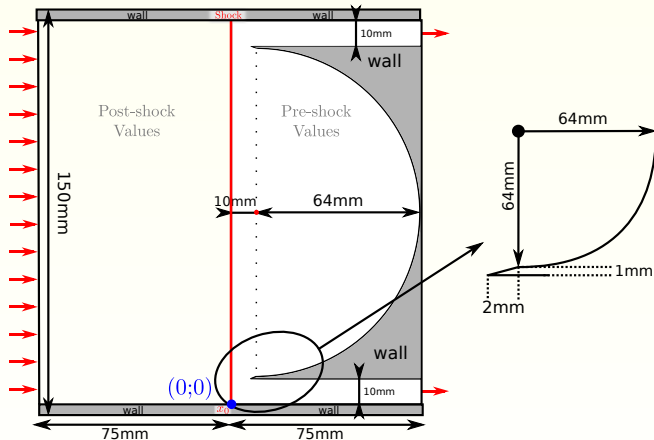
$$\rho_i^{n+1} = \rho_i^n - \sum_{j \in \nu(i)} \frac{\Delta t |S_{ij}|}{|K_i|} \sum_{r=1}^R \xi_r ([V_{ij,r} \cdot n_{ij}]^+ \rho_{ij,r}^n + [V_{ij,r} \cdot n_{ij}]^- \rho_{ji,r}^n).$$



volumes finitos de ordem
muito elevado

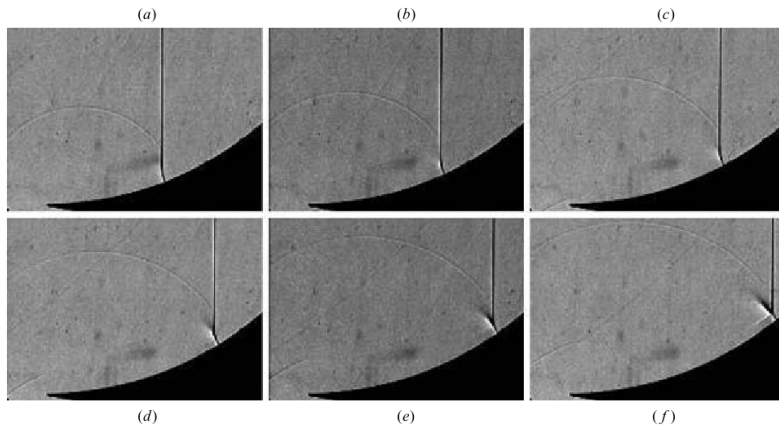
- Precisão muito mais elevada.
- Difusão numerica reduzida
- "computational cost" menos elevados para um precisão dada.
- **Algoritmos mais complexos**

A realistic case: shock on cylinder cavity



B. W. Skews and H. Kleine, *Flow features resulting from shock wave impact on a cylindrical cavity*, J. Fluid Mech., 580 (2007) 481–493.

Experimental results



Numerical simulations

Computação científica de alto desempenho 'high performance computing'

As técnicas actuais:

- Vectorização com SIMD: XMM (MultiMedia Extention 1997), SSE (Streaming SIMD Extention 1999-2006), AVX (advanced Vector Extention 2011) (7.6 Gflops *without*, 30 Gflop/s *with* per core).
- Programação Multi-core: OpenMP, Pthread. Usar 4 ate 24 cores com memoria partilhada (peak $24 \times 30 = 720$ Gflops).
- Programação GPU : CUDA, OpenCL. Utilizar as capacidades de calculo do GPU. Placas dedicadas Tesla, Fermi (500 Gflop/s).
- Programação paralela : MPI para comunicar com milhares de computadores. Exemplo: computador Jaguar Cray XT5-HE with 224 256 Opteron 6-core 2.6 GH 2 300 000 Gflop/s).

As técnicas actuais:

- Vectorização com SIMD: XMM (MultiMedia Extention 1997), SSE (Streaming SIMD Extention 1999-2006), AVX (advanced Vector Extention 2011) (7.6 Gflops *without*, 30 Gflop/s *with* per core).
- Programação Multi-core: OpenMP, Pthread. Usar 4 ate 24 cores com memoria partilhada (peak $24 \times 30 = 720$ Gflops).
- Programação GPU : CUDA, OpenCL. Utilizar as capacidades de calculo do GPU. Placas dedicadas Tesla, Fermi (500 Gflop/s).
- Programação paralela : MPI para comunicar com milhares de computadores. Exemplo: computador Jaguar Cray XT5-HE with 224 256 Opteron 6-core 2.6 GH 2 300 000 Gflop/s).

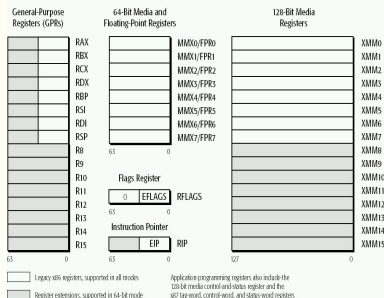
As técnicas actuais:

- Vectorização com SIMD: XMM (MultiMedia Extention 1997), SSE (Streaming SIMD Extention 1999-2006), AVX (advanced Vector Extention 2011) (7.6 Gflops *without*, 30 Gflop/s *with* per core).
- Programação Multi-core: OpenMP, Pthread. Usar 4 ate 24 cores com memoria partilhada (peak $24 \times 30 = 720$ Gflops).
- Programação GPU : CUDA, OpenCL. Utilizar as capacidades de calculo do GPU. Placas dedicadas Tesla, Fermi (500 Gflop/s).
- Programação paralela : MPI para comunicar com milhares de computadores. Exemplo: computador Jaguar Cray XT5-HE with 224 256 Opteron 6-core 2.6 GH 2 300 000 Gflop/s).

As técnicas actuais:

- Vectorização com SIMD: XMM (MultiMedia Extention 1997), SSE (Streaming SIMD Extention 1999-2006), AVX (advanced Vector Extention 2011) (7.6 Gflops *without*, 30 Gflop/s *with* per core).
- Programação Multi-core: OpenMP, Pthread. Usar 4 ate 24 cores com memoria partilhada (peak $24 \times 30 = 720$ Gflops).
- Programação GPU : CUDA, OpenCL. Utilizar as capacidades de calculo do GPU. Placas dedicadas Tesla, Fermi (500 Gflop/s).
- Programação paralela : MPI para comunicar com milhares de computadores. Exemplo: computador Jaguar Cray XT5-HE with 224 256 Opteron 6-core 2.6 GH 2 300 000 Gflop/s).

- SSE4 version: mais registros para limitar o acesso a memória
- AVX version: registros de 256 bits YMM. 4 operações de dupla precisão simultaneamente.
- AVX version: operações não destrutivas (com 3 registros)
- AVX version: transferência de blocos de memória com data alinhada.



registros internos

- Distribuir cálculos independentes sobre diferentes cores.
- Memória partilhadas entre os diferentes cores.
- Os compiladores realizam esta operação com `pragma`. Fácil de programar

Ciclo monocore

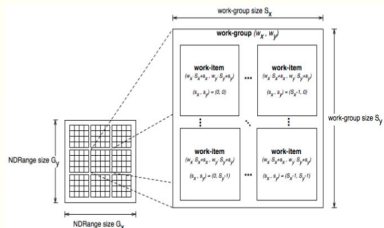
```
for(int i=0; i<999999;i++)  
    a[i] = b[i] + 9;
```

Ciclo multicore com OpenMP

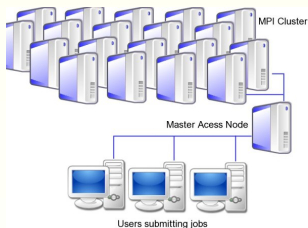
```
#include <omp.h>  
#pragma omp parallel for  
for(int i=0; i<999999;i++)  
    a[i] = b[i] + 9;
```

- O principal problema é o acesso à memória (conflitos de acesso, fluxo limitado no 'bus').

- Distribuir um conjunto de operações dentro um thread constituído de vários nós de cálculo em paralelos
- Muitas unidades de cálculo para um preço muito baixo.
- É necessário uma livreria para 'falar' com o GPU. (CUDA, OpenCL)
- Transferência da memória CPU \Leftrightarrow GPU
- Estrutura dos dados específica (blocos de memória em 2^n)



OpenCL threads



MPI cluster: PC farm

- Memória partilhada. Cada computador é um nó de calculo independente.
- Sistema de comunicação e transferência de dados (MPI).
- Programação e esquemas numéricos específicas.
- Não há limitação do número de máquinas.

- *Modelação* – > *Esquema numérica* – > *Implementação* – > *Simulação*
- Projetos volumes finitos de ordens muitas elevadas
- Projetos shallow-water Tsunami (geofísica)
- Projetos FCT: simulação de um escoamento de polímeros

Objectivos : Realizar uma implementação eficiente de alto desempenho usando as várias técnicas.

Tese de Mestrado: Propostas para realizar uma tese de mestrado no domínio da simulação numérica e o cálculo científico intensivo com técnica de vectorização , GPU ou paralelização .

Bolsa Post-Mestrado Desenvolvimento de código de computador eficiente para escoamento de polímeros. PhD thesis possível.