



# **Computação na Modelagem numérica da Atmosfera parte02**

**(Paulo Yoshio Kubota, Ph.D.)**

Cachoeira Paulista-SP  
CPTEC/INPE  
21 Agosto 2024



# Sumário



- 1. Previsão numérica de Tempo.**
  - a. Modelo Matemático (Equações Governantes)**
  - b. Tipo de Grade Numérica**
  - c. Forma de Discretização**

**Qual a relação  
Meteorologia-  
Computação...**



- 1. Otimização do tempo computacional das PNT.**
  - a. Otimização Serial.**
  - b. Otimização Paralela**
  - c. Paralelismo de Memória Compartilhada usado no MCGA-CPTEC (OpenMP)**
  - d. Paralelismo de Memória Distribuída usado no MCGA-CPTEC (MPI)**

**Que Tipo de Profissional é  
Desejado**

- 1. Impacto para Sociedade.**
  - a. Produtos de PNT com Alta Qualidade Com Baixo Custo.**
  - b. Tipo de Supercomputador e Consumo de Energia Elétrica**

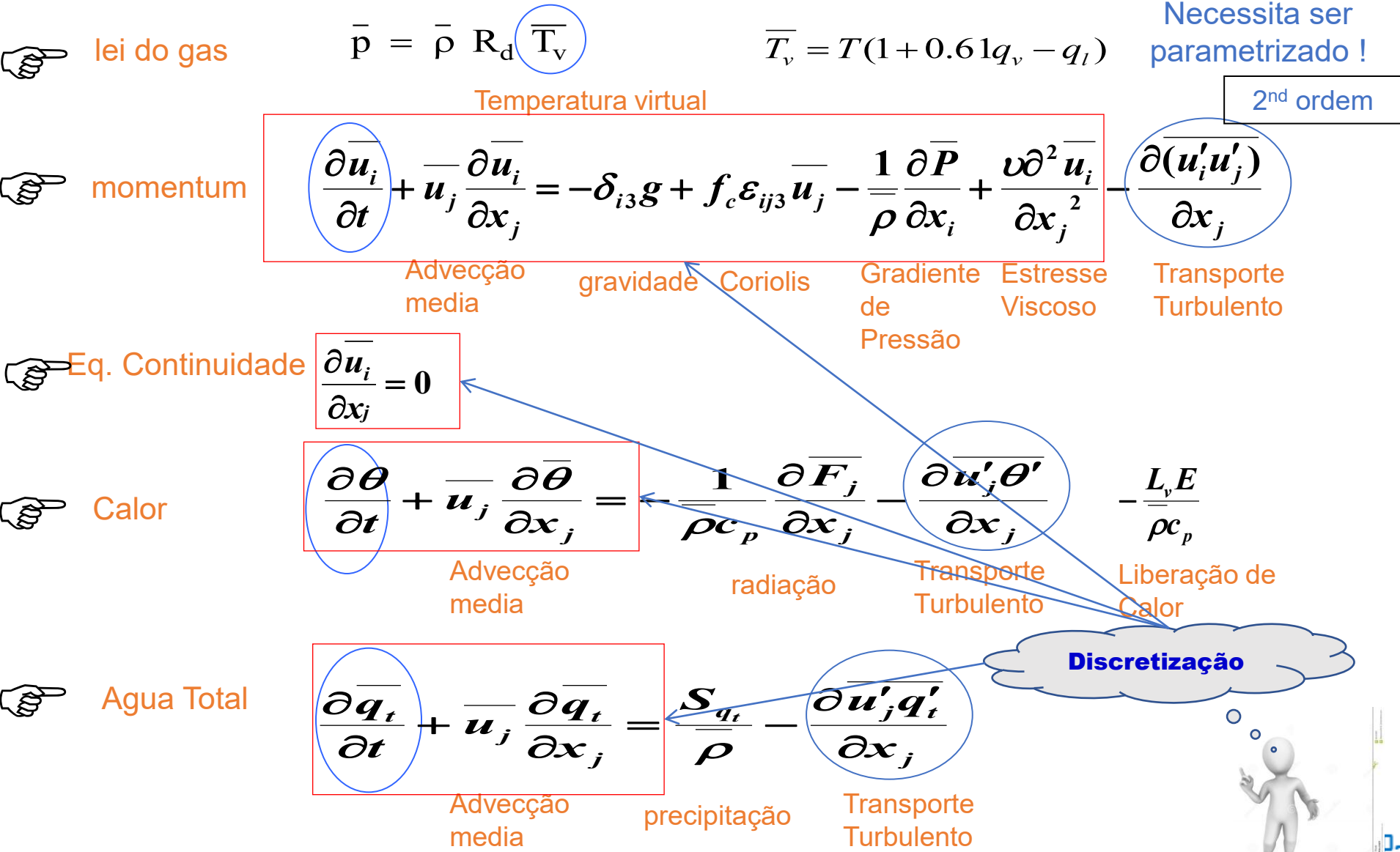






# Equações governantes do estado médio

Média de Reynolds  $A = \bar{A} + A'$  Nas equações de Navier Stokes

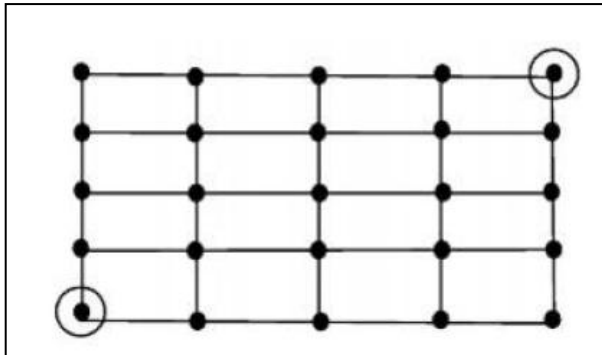




# Tipo de Malha Numérica



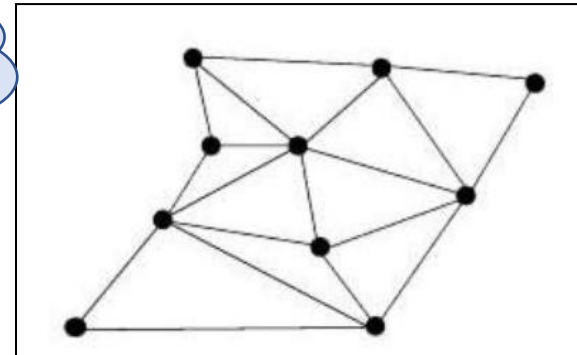
## ESTRUTURADA (consistes de retângulos 2D)



Onde  
Discretizar?



## NÃO-ESTRUTURADA (consistes de triângulos 2D)



### Vantagens:

- A indexação sozinha, mostra uma conectividade limpa,
- Fácil de manipular os índices e armazenar na computação

### Desvantagens:

- Restrições à ortogonalidade e razão de aspecto.
- Difícil envolver geometrias complexas.
- Menos eficiente para malhas localmente refinadas.

### Vantagens:

- Informações de conectividade para cada célula precisam ser armazenadas
- Difícil de armazenar e manipular dados na computação

### Desvantagens:

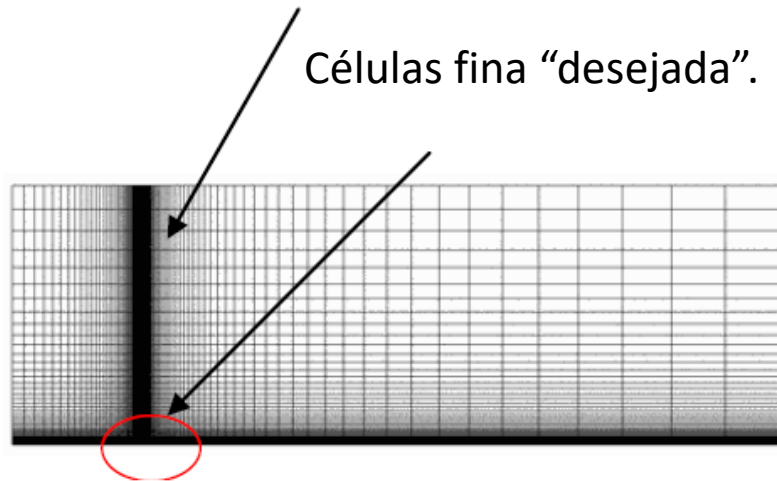
- Mais fácil envolver forma complexas.
- Muito eficiente para malhas localmente refinadas



# Tipo de Malha Numérica

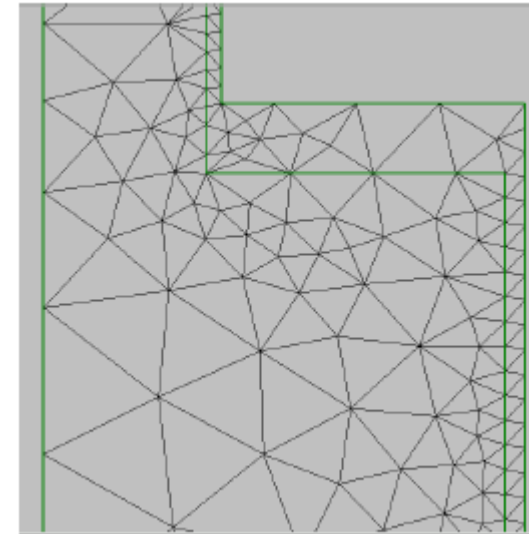


## ESTRUTURADA (consistes de retângulos 2D)



- Geração de Malha (Difícil e manual)
- Usada para Diferença Finitas , e Volume finito

## NÃO-ESTRUTURADA (consistes de triângulos 2D)



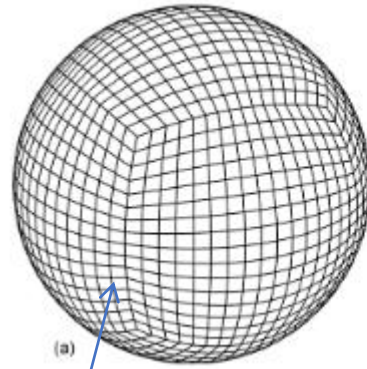
- Geração de Malja (Facil e geralmente automatica)
- Usada para Elemento Finitos e Volume finito



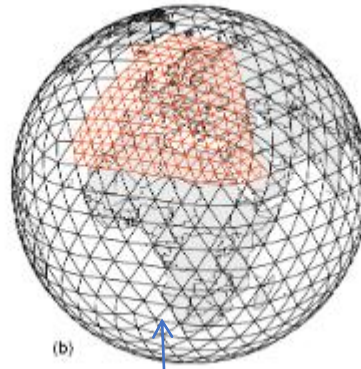




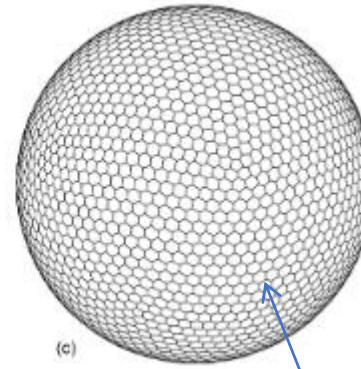
# Tipo de Malha Numérica



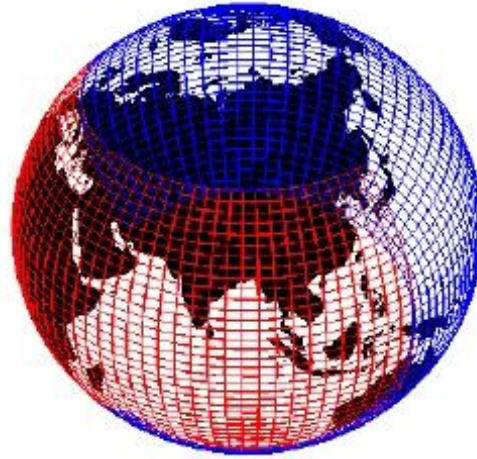
a) Esfera cubada



b) Icosaédrica triangular



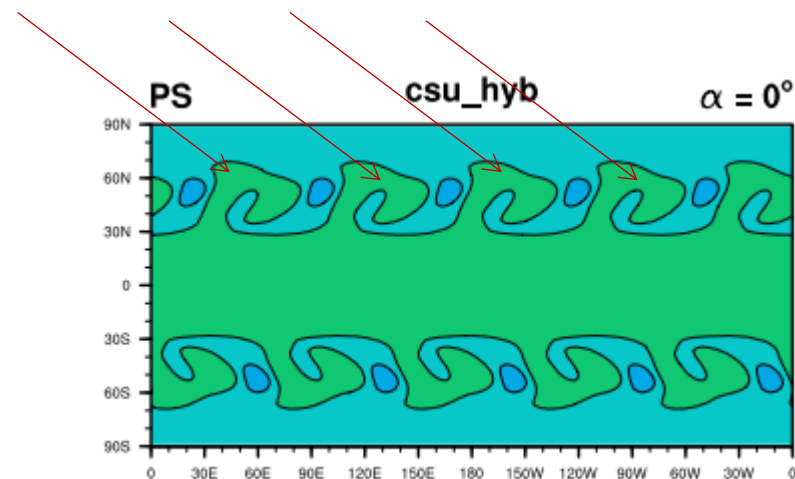
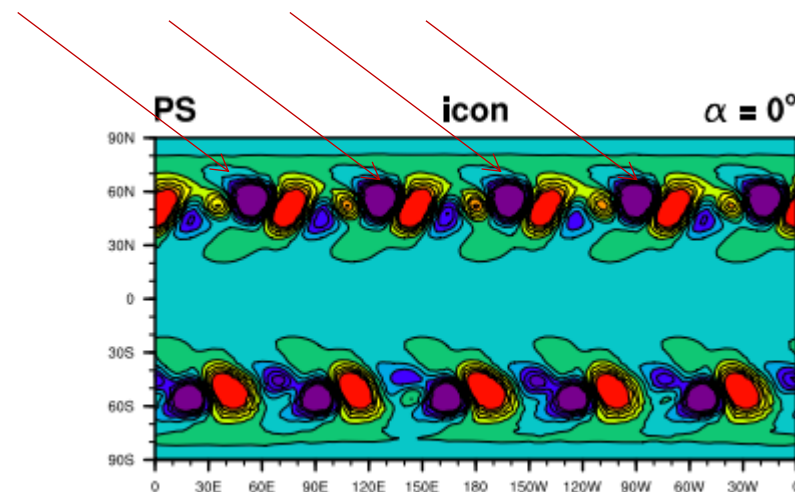
c) Icosaédrica hexagonal





## Métodos Numéricos: Problemas

Ambos os modelos **OLAM**, **ICON** e **CSU** são construídos em uma grade icosaédrica (resultados da oficina de 2008). Isso leva a um **realce** do modo de onda  **$k = 5$** .



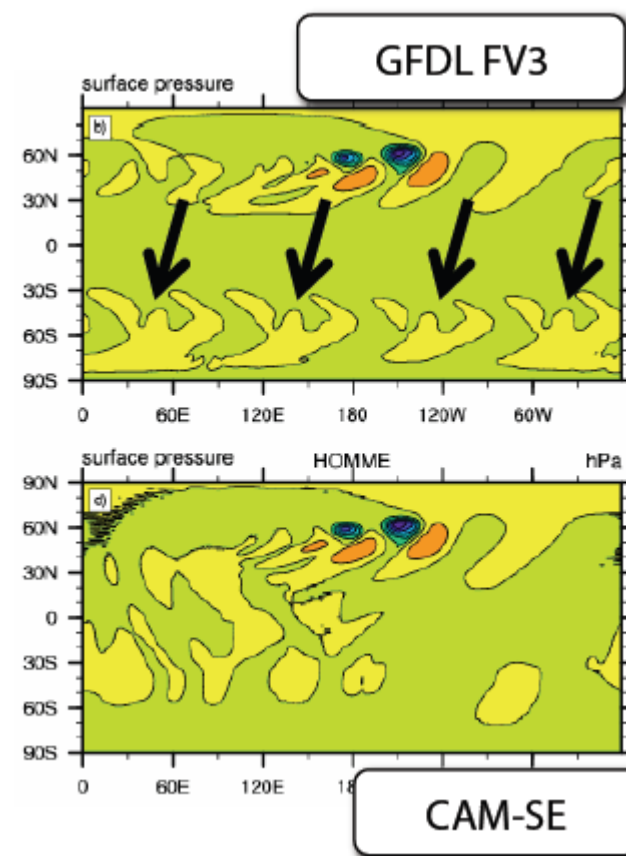
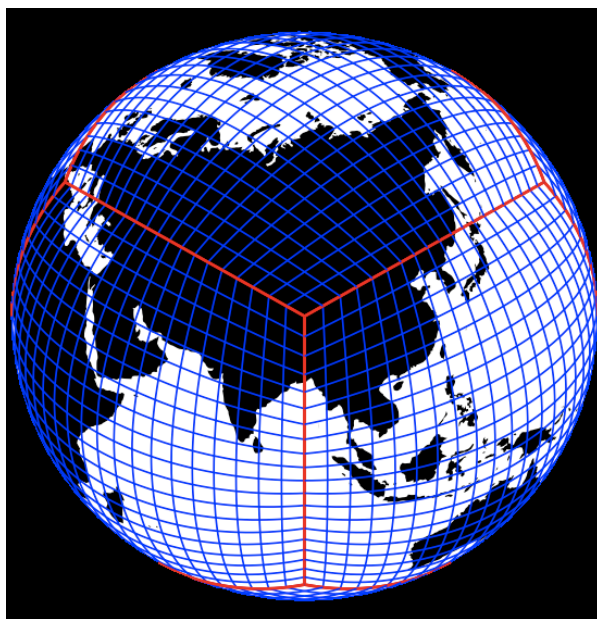




## Métodos Numéricos: Problemas

Tanto o modelo **GFDL FV3** (FVcubed) como o modelo **CAM-SE** (elemento espectral) são construídos na **esfera em cubos**. Isso leva a um **realce** do modo de onda  **$k = 4$** .

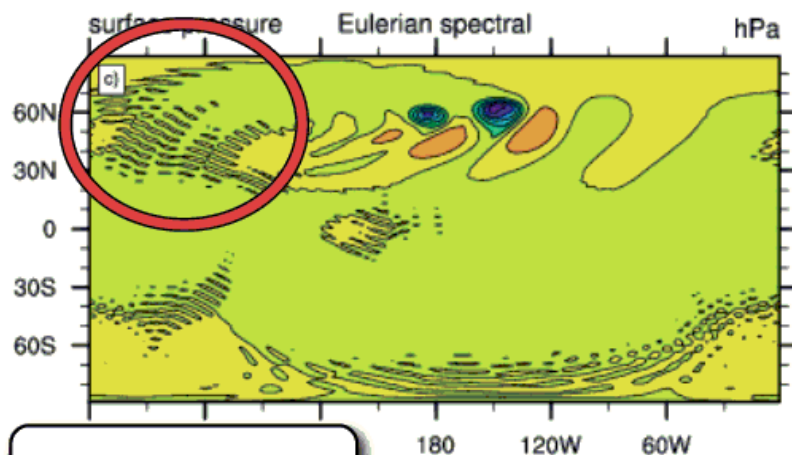
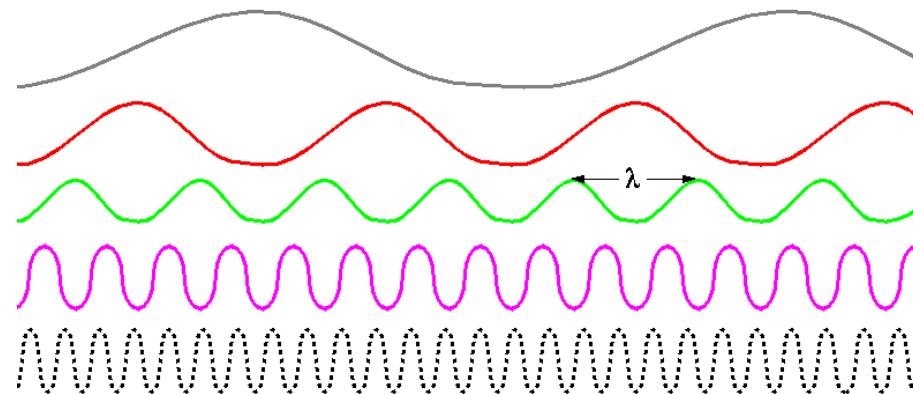
O uso de alta ordem numérica no CAM-SE é mais eficaz para reprimir este modo.



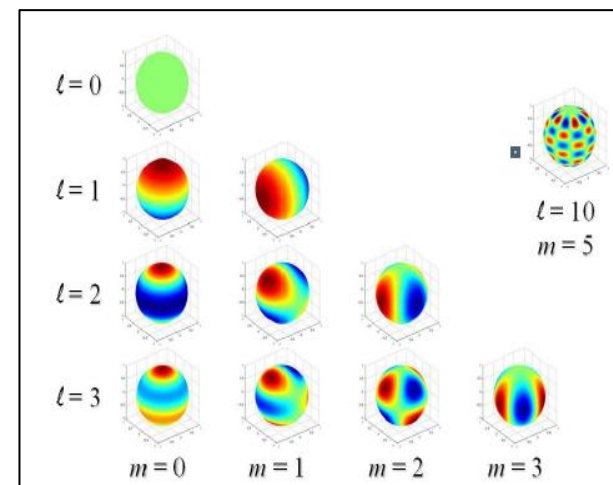


## Métodos Numéricos: Problemas

CAM-EUL (Euleriana) e BAM usam métodos espectrais, que são conhecidos por serem sensíveis ao ruído espectral. Este ruído é caracterizado por oscilações rápidas devido ao realce do modo de alta frequência.



CAM-EUL





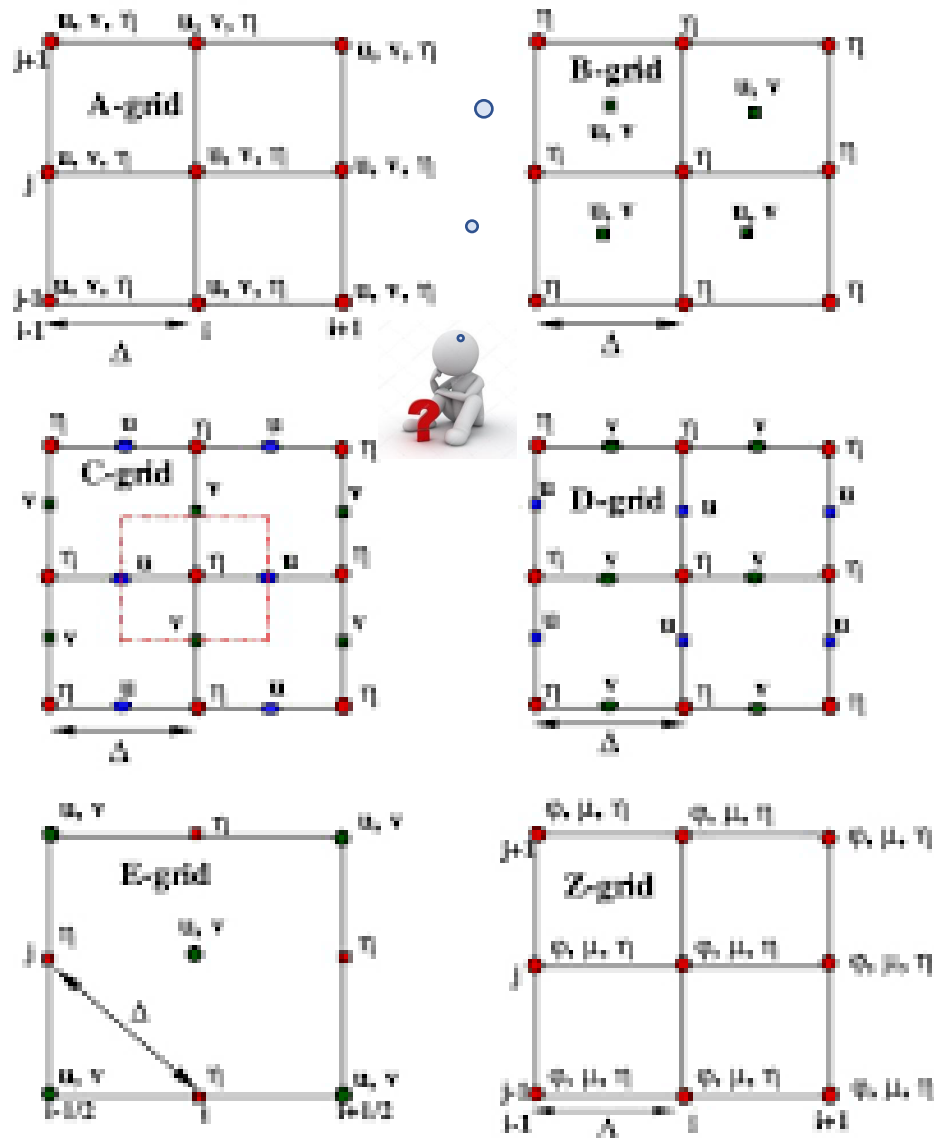
# MPAS-Malha de Voronoi???



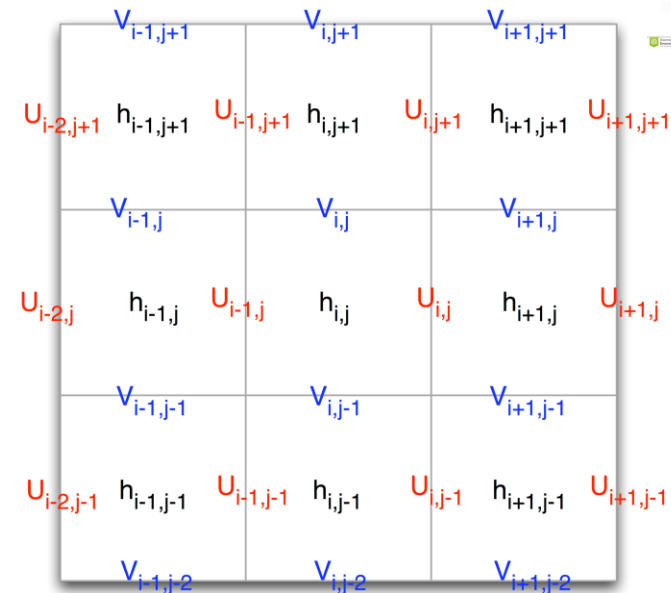


# Tipo de Grade Numérica

Porque Diferentes  
grades

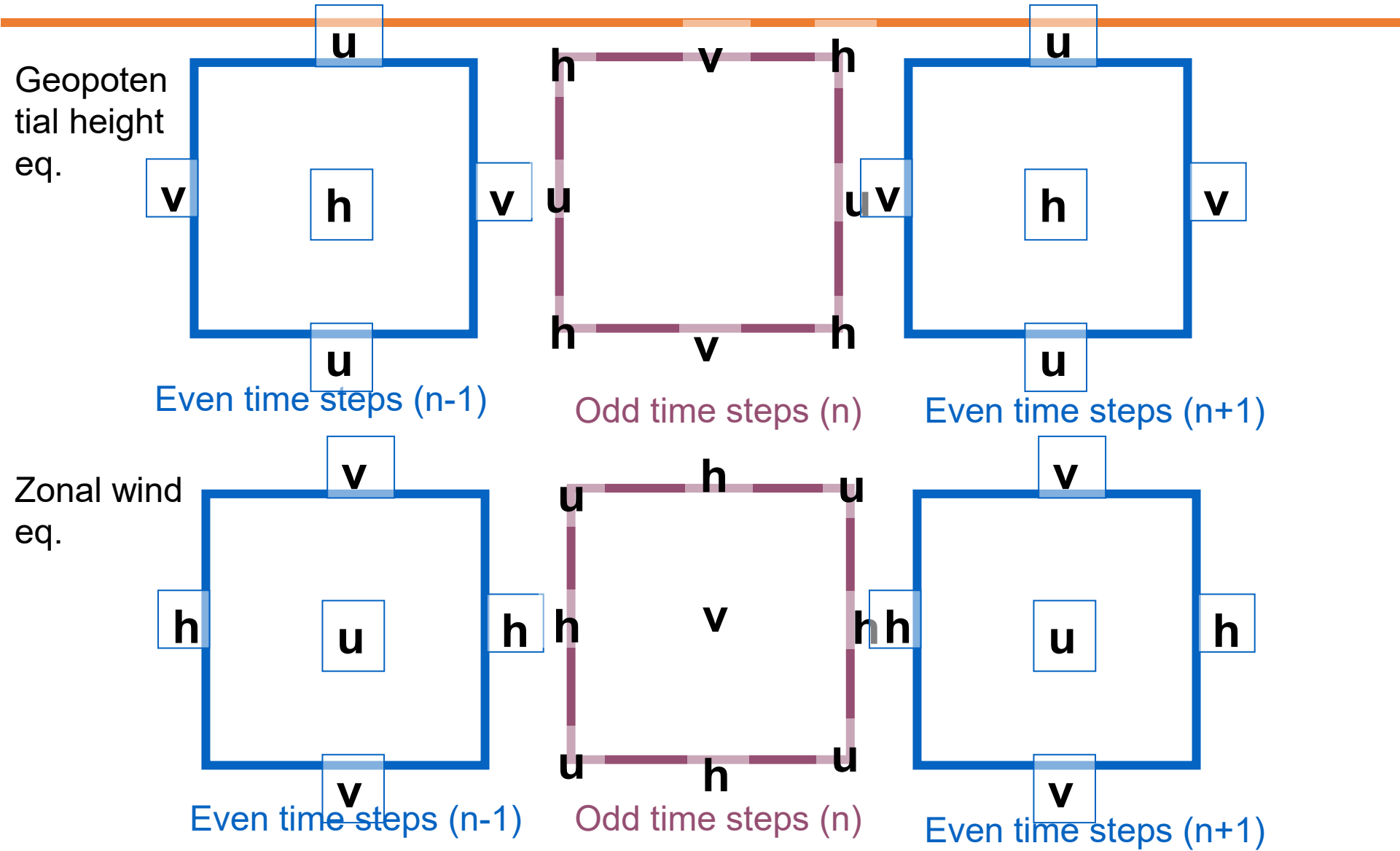


Melhorar a  
estabilidade e a  
conservação.  
Massa, Energia..





# Tipo de Malha Numérica (Grade Eliassen )







# Forma de Discretização

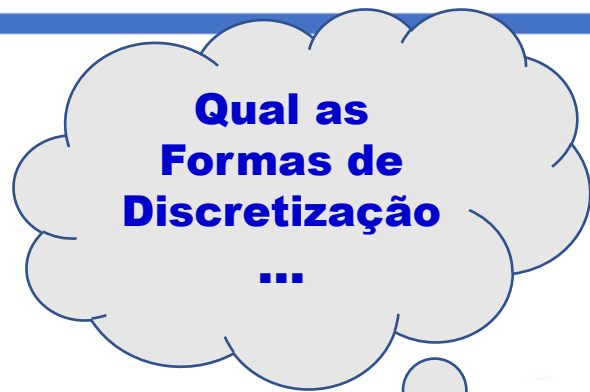


1. Método de Diferenças Finitas: **BRAMS**, WRF (forma DE FLUXO)

2. Método de Volume Finito: (ETA\*\*, MONAN)

3. Método de Elementos Finitos (CAM-SE)

4. Método de Espectral: (BAM)

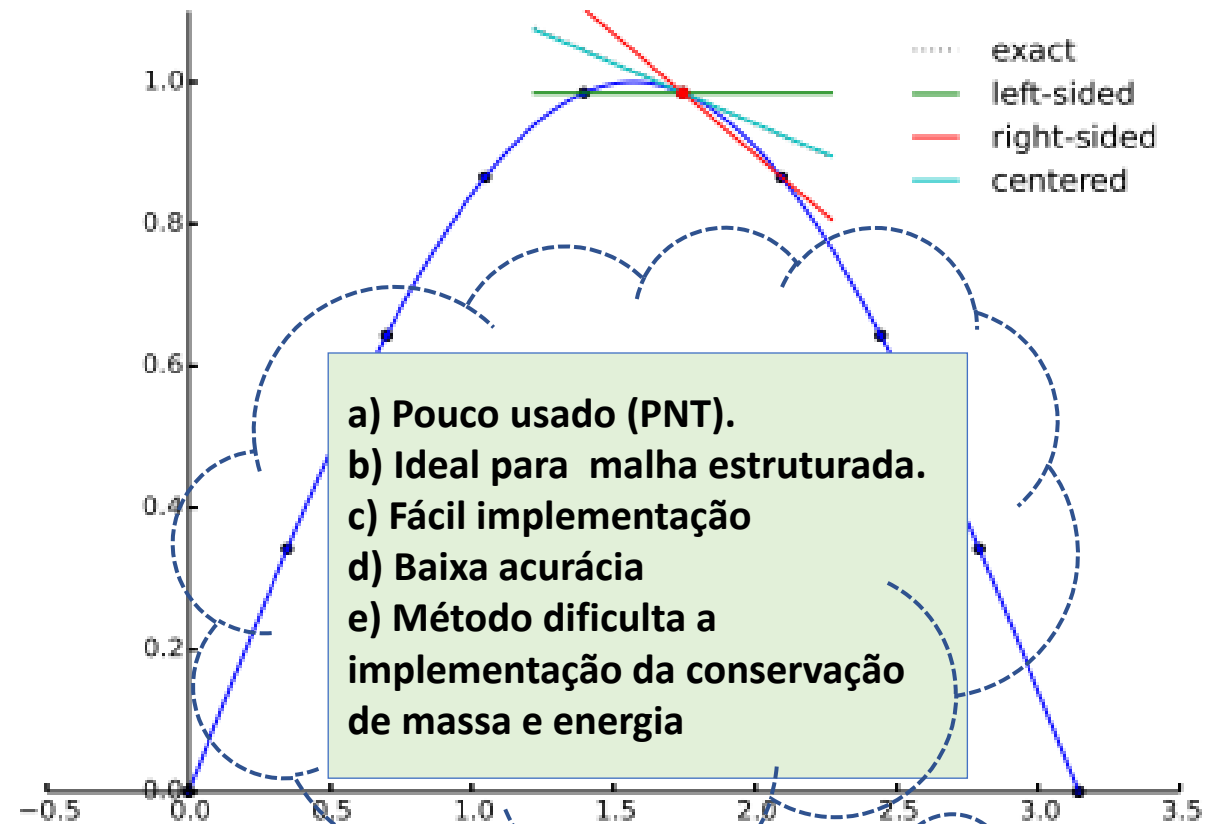




# Forma de Discretização



## 1. Método de Diferenças Finitas:



- a) Pouco usado (PNT).
- b) Ideal para malha estruturada.
- c) Fácil implementação
- d) Baixa acurácia
- e) Método dificulta a implementação da conservação de massa e energia

	$V_{i-1,j+1}$		$V_{i,j+1}$		$V_{i+1,j+1}$	
$U_{i-2,j+1}$	$h_{i-1,j+1}$	$U_{i-1,j+1}$	$h_{i,j+1}$	$U_{i,j+1}$	$h_{i+1,j+1}$	$U_{i+1,j+1}$
	$V_{i-1,j}$		$V_{i,j}$		$V_{i+1,j}$	
$U_{i-2,j}$	$h_{i-1,j}$	$U_{i-1,j}$	$h_{i,j}$	$U_{i,j}$	$h_{i+1,j}$	$U_{i+1,j}$
	$V_{i-1,j-1}$		$V_{i,j-1}$		$V_{i+1,j-1}$	
$U_{i-2,j-1}$	$h_{i-1,j-1}$	$U_{i-1,j-1}$	$h_{i,j-1}$	$U_{i,j-1}$	$h_{i+1,j-1}$	$U_{i+1,j-1}$
	$V_{i-1,j-2}$		$V_{i,j-2}$		$V_{i+1,j-2}$	

$$\left. \frac{da}{dx} \right|_i \approx \frac{a_i - a_{i-1}}{\Delta x}$$

$$\left. \frac{da}{dx} \right|_i \approx \frac{a_{i+1} - a_i}{\Delta x}$$

$$\left. \frac{da}{dx} \right|_i \approx \frac{a_{i+1} - a_{i-1}}{2\Delta x}$$





# Forma de Discretização

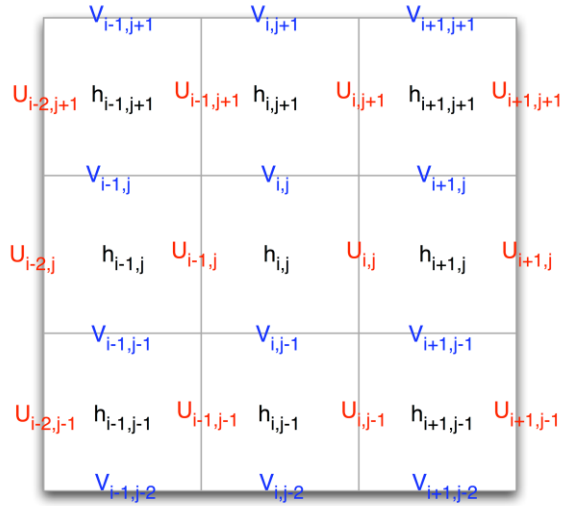
A lei de conservação

## 1. Método de Volume Finito:

$$\frac{\partial q}{\partial t} + \frac{1}{J} \nabla \cdot \vec{F} = S$$

Integra sobre um elemento  $Z$  com contorno  $\partial Z$  a aplicando o teorema da divergência. Teremos:

- a) Muito usado (PNT).
- b) Ideal para malha estruturada e não estruturada .
- c) Implementação mais elaborada
- d) Alta acurácia (depende do graus)
- e) Método facilita a conservação de massa e energia



$$\frac{\partial \vec{q}}{\partial t}$$

Evolução no tempo do estado do elemento médio

+

$$\oint_{\partial Z} \vec{F} \cdot \vec{n} ds$$

Fluxo Através do elemento de contorno

=

$$\int_Z S dV$$

Temo Fonte do elemento médio



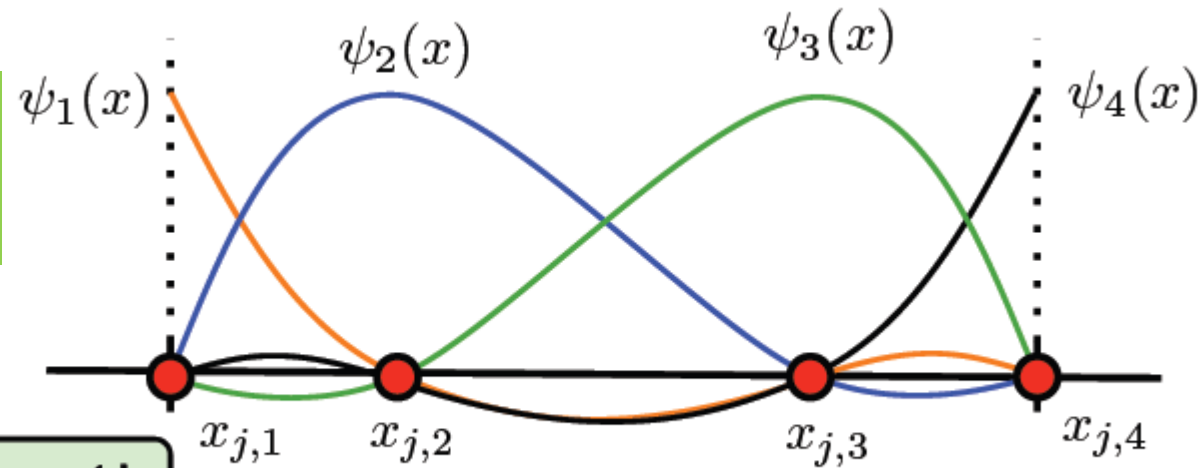
# Forma de Discretização



## 1. Método de Elementos Finitos:

$$q(x, t) = \sum_{n=1}^N a_n(t) \psi_n(x)$$

Funções  
Básicas do  
Elemento J



Element j

- a) Pouco usado (PNT).
- b) Ideal para malha não estruturada .
- c) Implementação bem elaborada
- d) Alta acurácia
- e) Método facilita a conservação de massa e energia

$V_{i-1,j+1}$	$V_{i,j+1}$	$V_{i+1,j+1}$
$U_{i-2,j+1}$ $h_{i-1,j+1}$	$U_{i-1,j+1}$ $h_{i,j+1}$	$U_{i,j+1}$ $h_{i+1,j+1}$ $U_{i+1,j+1}$
$V_{i-1,j}$	$V_{i,j}$	$V_{i+1,j}$
$U_{i-2,j}$ $h_{i-1,j}$	$U_{i-1,j}$ $h_{i,j}$	$U_{i,j}$ $h_{i+1,j}$ $U_{i+1,j}$
$V_{i-1,j-1}$	$V_{i,j-1}$	$V_{i+1,j-1}$
$U_{i-2,j-1}$ $h_{i-1,j-1}$	$U_{i-1,j-1}$ $h_{i,j-1}$	$U_{i,j-1}$ $h_{i+1,j-1}$ $U_{i+1,j-1}$
$V_{i-1,j-2}$	$V_{i,j-2}$	$V_{i+1,j-2}$

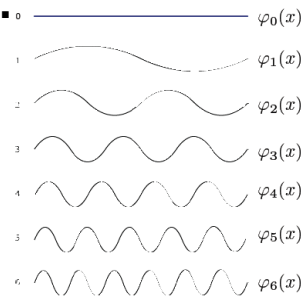




# Forma de Discretização



## 1. Método de Espectral:



$$\phi(x,t) = \sum_{k=1}^N a_k(t) \phi_k(x)$$

$$\psi_k(x) = e^{-ikx}$$

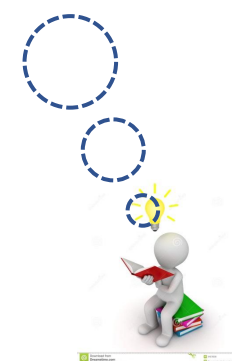
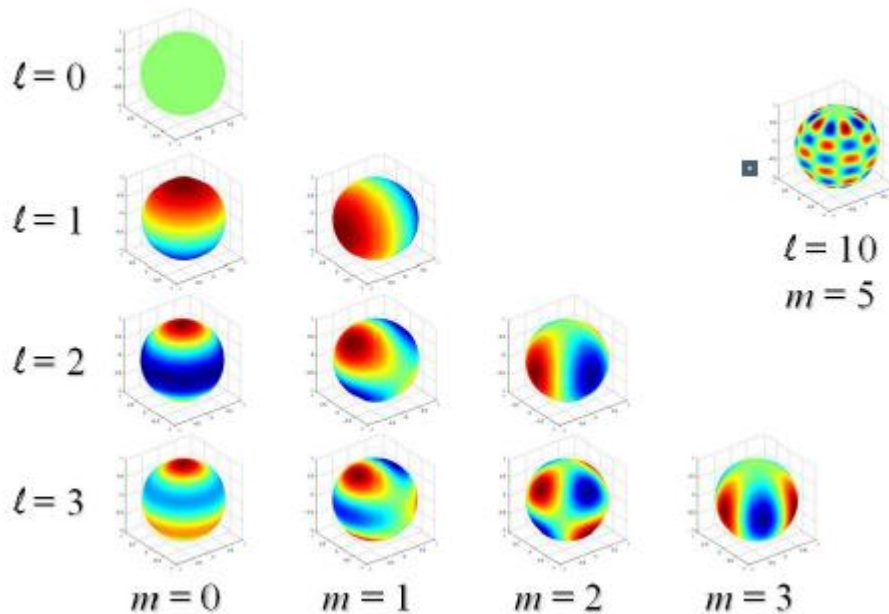
Harmônico Linear: Ortogonalidade

$$\int_S \psi_k \psi_n dS = \begin{cases} 2\pi, & \text{se } k = n \\ 0, & \text{se } k \neq n \end{cases}$$

Harmônicos Esféricos: Ortogonalidade

$$\int_S \phi_{l,m} \phi_{k,n} dS = \begin{cases} I_{l,m}, & k = l \text{ e } m = n \\ 0, & k \neq l \text{ ou } m \neq n \end{cases}$$

a) Muito usado antigamente (PNT).  
c) Implementação bem elaborada  
d) Alta acurácia  
e) Método facilita a conservação de massa.







# 1. Otimização do tempo computacional das PNT.

a. Otimização Serial.

b. Otimização Paralela

c. Paralelismo de Memória Compartilhada  
usado no **MCGA-CPTEC** (OpenMP)

d. Paralelismo de Memória Distribuída usado  
no **MCGA-CPTEC** (MPI)

e. Processamento me **GPU** (OPENACC)





## Compilers:

(Intel, PGI, CRAY, GNU, etc.)

## Libraries:

(LAPCK, BLAS, etc)

**COMO Otimizar o  
código serial**

## Para Escrever um Código Otimizado:

- a) eficiência do algoritmo,
- b) como o executável explora a arquitetura do processador

## Instrução e Otimização do paralelismo de dados:

- a) **pipelines**: (decompõe a instrução de ponto flutuante em estágios.)
- b) **vector registers**: (um pipeline pode operar em um conjunto de dados de ponto flutuante usando registros amplos)



## Cache optimisation

- a) **Temporal**: (múltiplo uso do mesmo item de dados em um curto período de tempo enquanto os dados residem na memória cache.)
- b) **Spatial**: (refere-se ao uso de dados localizados em um bloco de endereços consecutivos)

## Gerenciamento de Memória

Gerenciar a alocação dinâmica de memória nas funções e subrotinas



## Diferentes tipos de abordagens:

\* **MPI** (paralelismo de memória distribuída)  
(BAM,ETA,BRAM,WRF)

\* **OpenMP / OpenACC\*** (paralelismo incremental / fácil,  
memória compartilhada)(BAM, BRAMS\*)

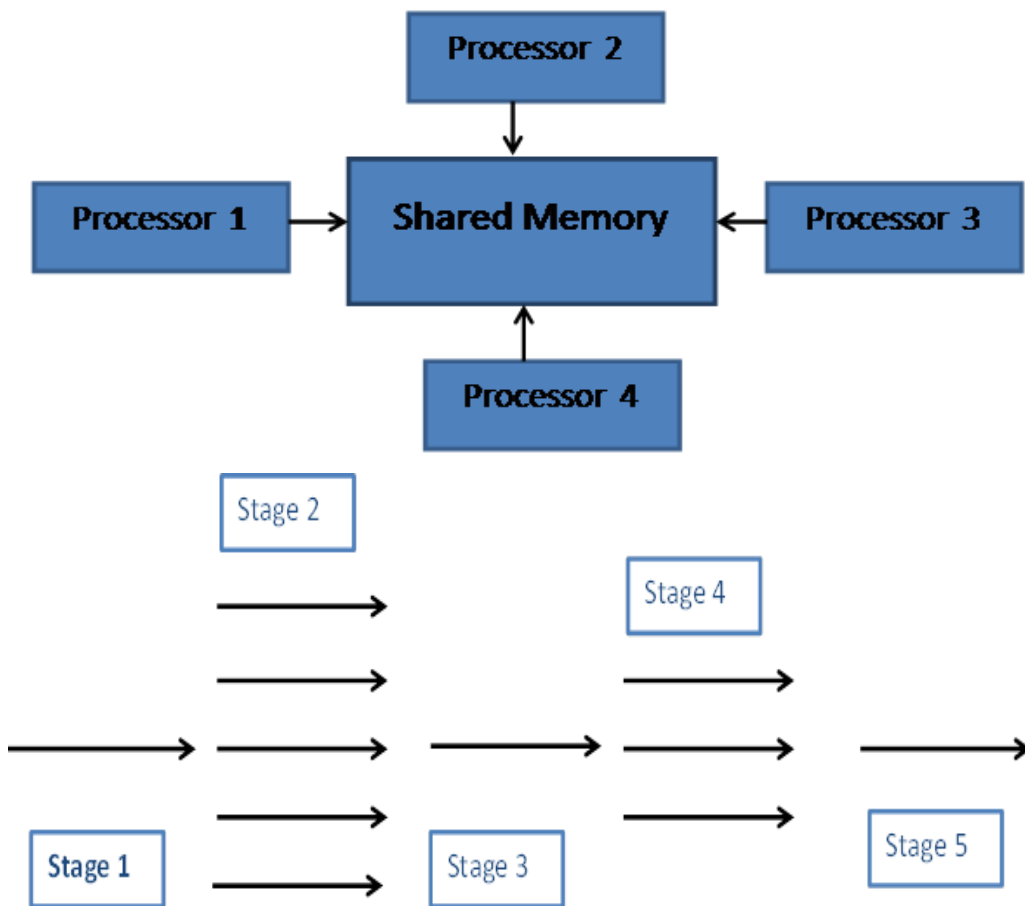
**CUDA** (E uma tecnologia proprietária da Nvidia. Desenvolvida  
especificamente para as GPUs Nvidia)

**OpenCL** (paralelismo massivo heterogêneo)

\* **Usado nos modelos do CPTEC**



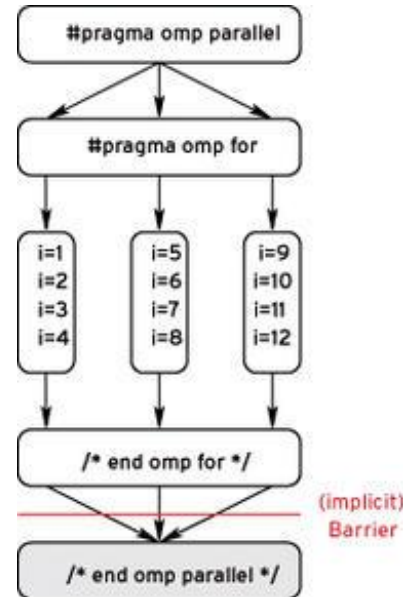
## OpenMP (paralelismo incremental / fácil, memória compartilhada)



Fonte: <http://pawangh.blogspot.com/2014/05/mpi-vs-openmp.html>

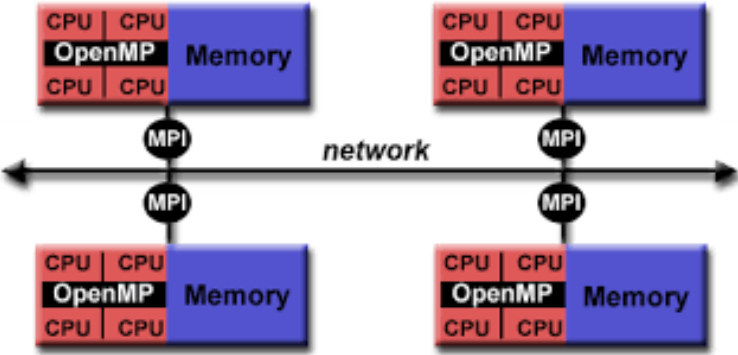
### Program hybrid

```
call OMP_SET_NUM_THREADS(4)
!$OMP PARALLEL DO PRIVATE(i)
!$OMP&                                SHARED(n)
do i=1,n
  ... computation
enddo
!$OMP END PARALLEL DO
end
```



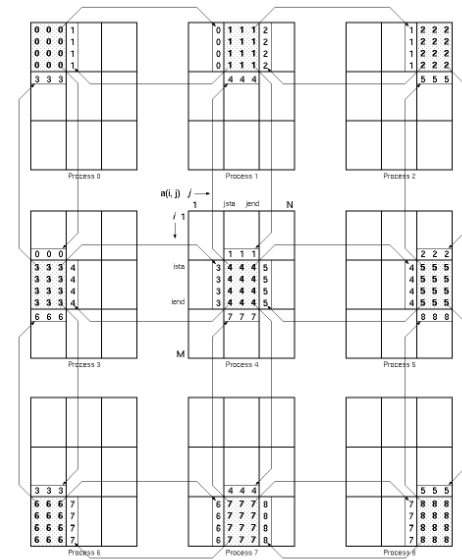


# MPI (paralelismo de memória distribuída)



## A Pseudo Hybrid Code

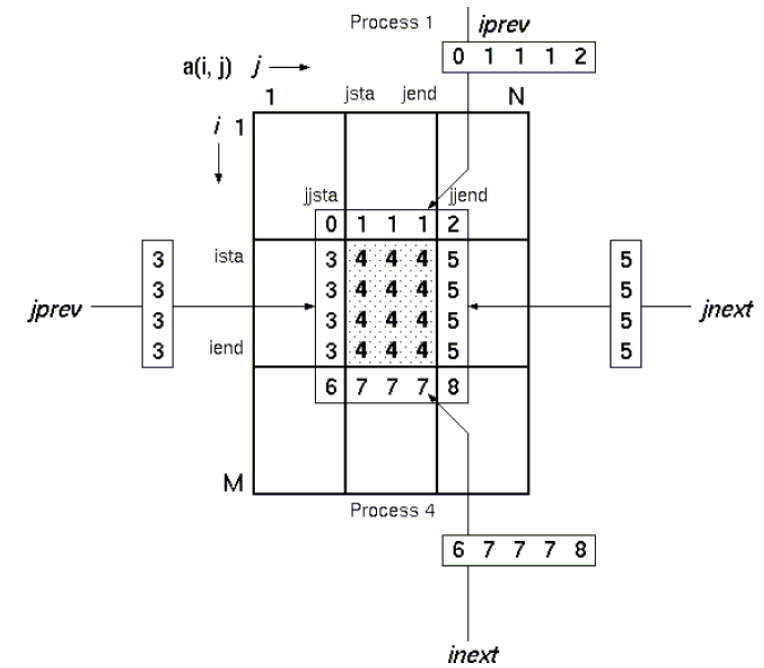
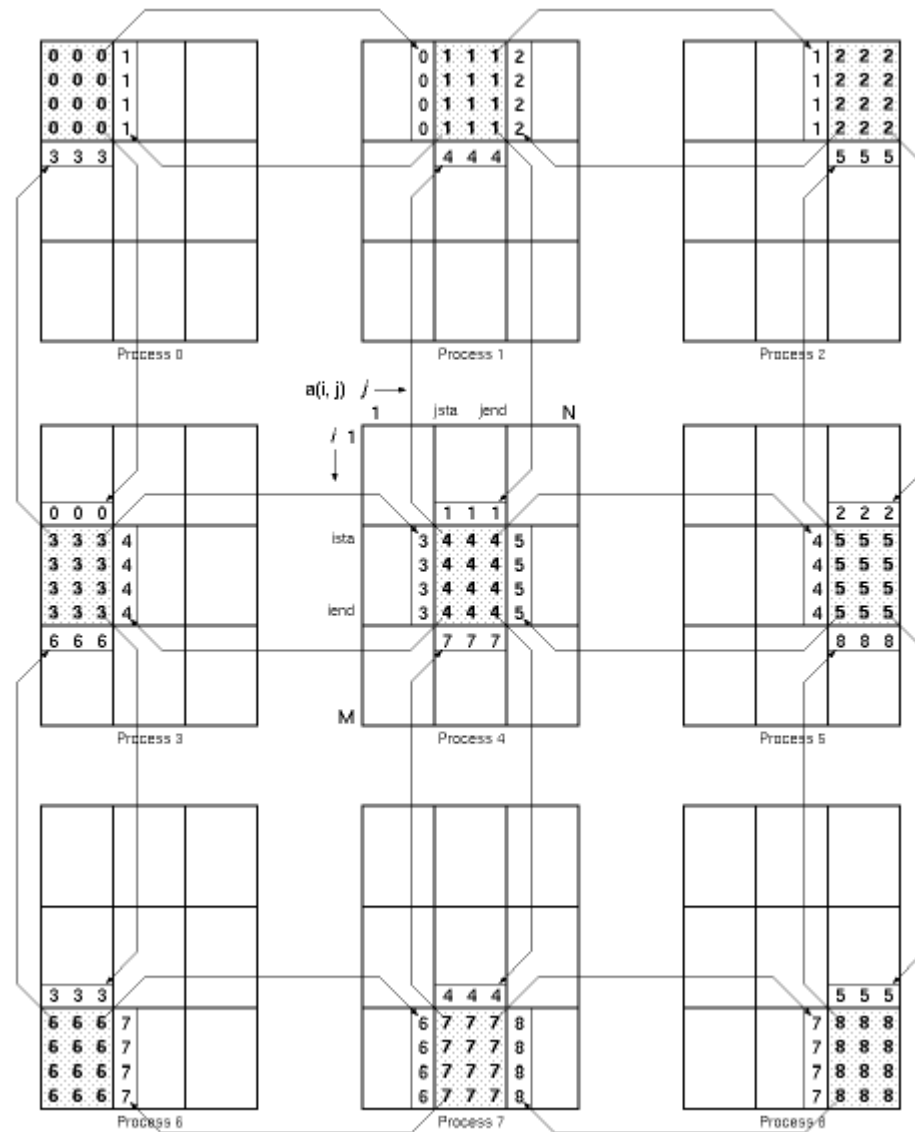
```
Program hybrid
call MPI_INIT (ierr)
call MPI_COMM_RANK (...)
call MPI_COMM_SIZE (...)
... some computation and MPI communication
call OMP_SET_NUM_THREADS(4)
!$OMP PARALLEL DO PRIVATE(i)
!$OMP&                SHARED(n)
  do i=1,n
    ... computation
  enddo
!$OMP END PARALLEL DO
... some computation and MPI communication
call MPI_FINALIZE (ierr)
end
```







# MPI (paralelismo de memória distribuída)





# MPI (paralelismo de memória distribuída)



```
CALL MPI_ISEND (a(ista,jend),ilen,MPI_REAL8,jnext,1,MPI_COMM_WORLD,isend1,ierr)
CALL MPI_ISEND (a(ista,jsta),ilen,MPI_REAL8,jprev,1,MPI_COMM_WORLD,isend2,ierr)
CALL MPI_ISEND (works1(jsta),jlen,MPI_REAL8,inext,1,MPI_COMM_WORLD,jsend1,ierr)
CALL MPI_ISEND (works2(jsta),jlen,MPI_REAL8,iprev,1,MPI_COMM_WORLD,jsend2,ierr)
```

```
CALL MPI_IRECV (a(ista,jsta-1),ilen,MPI_REAL8,jprev,1,MPI_COMM_WORLD,irecv1,ierr)
CALL MPI_IRECV (a(ista,jend+1),ilen,MPI_REAL8,jnext,1,MPI_COMM_WORLD,irecv2,ierr)
CALL MPI_IRECV (workr1(jsta) ,jlen,MPI_REAL8,iprev,1,MPI_COMM_WORLD,jrecv1,ierr)
CALL MPI_IRECV (workr2(jsta) ,jlen,MPI_REAL8,inext,1,MPI_COMM_WORLD,jrecv2,ierr)
```

```
CALL MPI_WAIT (isend1, istatus, ierr)
CALL MPI_WAIT (isend2, istatus, ierr)
CALL MPI_WAIT (jsend1, istatus, ierr)
CALL MPI_WAIT (jsend2, istatus, ierr)
```

```
CALL MPI_WAIT (irecv1, istatus, ierr)
CALL MPI_WAIT (irecv2, istatus, ierr)
CALL MPI_WAIT (jrecv1, istatus, ierr)
CALL MPI_WAIT (jrecv2, istatus, ierr)
```

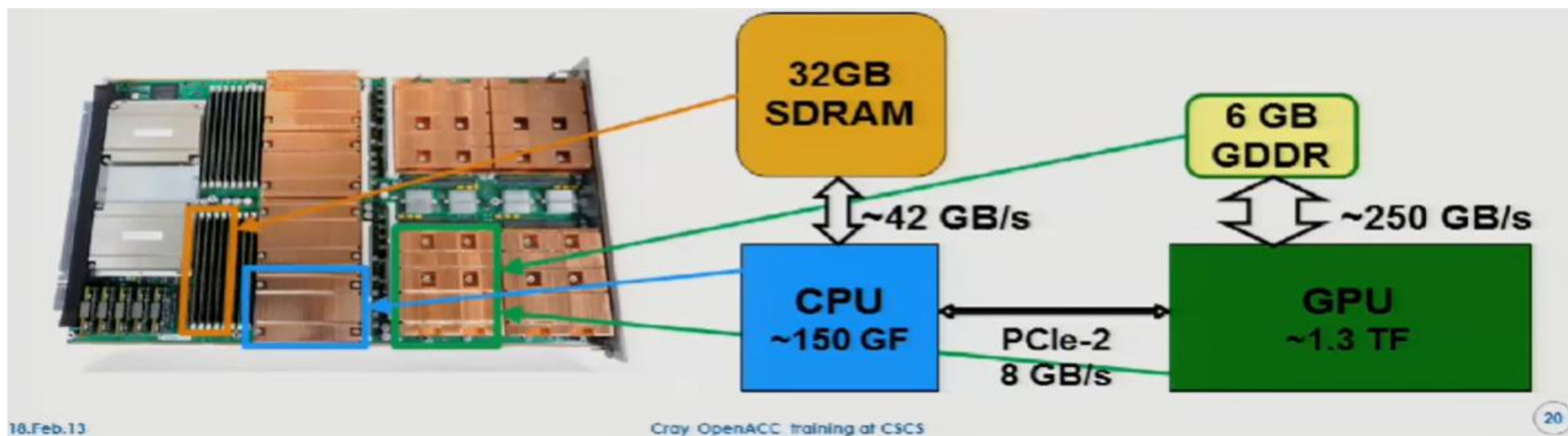


# GPU (paralelismo de memória compartilhada)



Qual a velocidade das GPUs atuais?

1. Cuidado com o exagero: "Ganhei 1000x de aceleração em uma GPU"
2. O que você deve esperar?
3. Cray XK7.
  - a) Flop / s GPU ~ 9x mais rápido do que CPU inteira com (16 núcleos)
  - b) A memória da Largura de banda: GPU -6x mais rápida que a CPU.
1. Essas proporções serão semelhantes em outros sistemas
- 2 - Além disso, é mais difícil alcançar o desempenho máximo na GPU.
  - a) Seu código precisa para se ajustar à arquitetura.
  - b) Você também precisa levar em consideração as transferências de dados entre a CPU e a GPU.





# GPU (paralelismo de memória compartilhada)



## Parallelizing a single loop

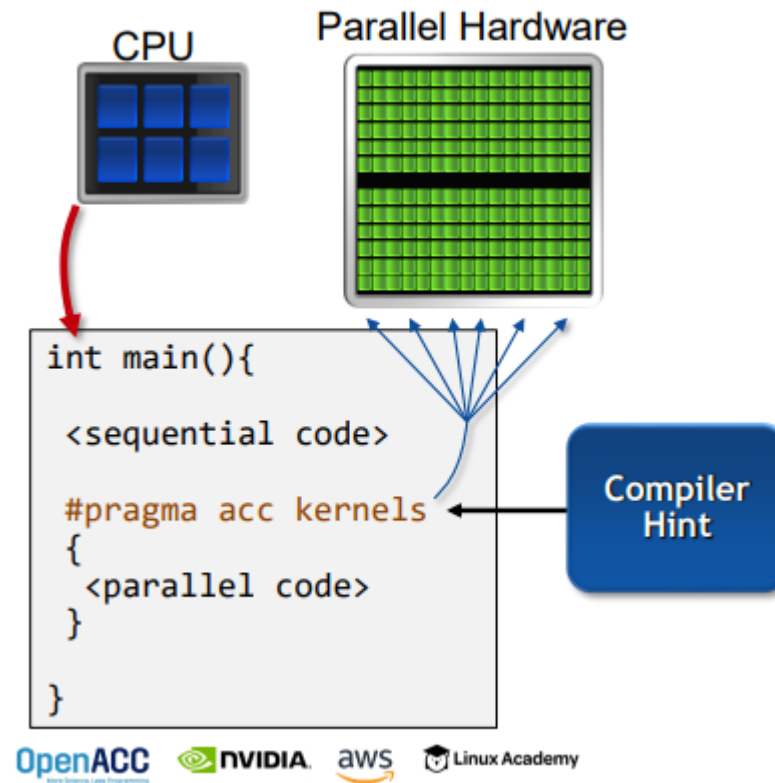
C/C++

```
#pragma acc parallel loop
for(int i = 0; i < N; i++)
    a[i] = 0;
```

Fortran

```
!$acc parallel loop
do i = 1, N
    a(i) = 0
end do
```

OpenACC



O programador dá o sinal ao compilador.  
O compilador paraleliza o código.



**a. Produtos de PNT com Alta Qualidade Com Baixo Custo.**

**b. Tipo de Supercomputador e Consumo de Energia Elétrica**





# Produtos de PNT com Alta Qualidade Com Baixo Custo.



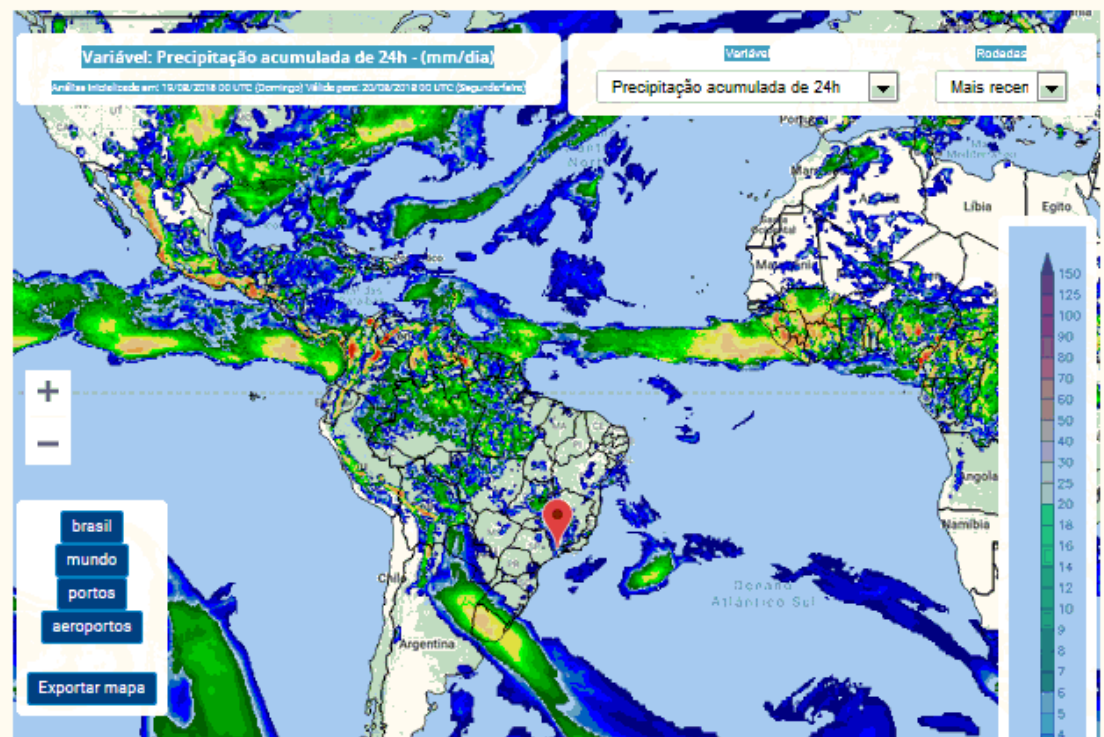
## Centro de Previsão de Tempo e Estudos Climáticos

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Tempo Clima Previsão Numérica Satélite Ondas Dados Observacionais Instrumentação Meteorológica Qualidade do Ar Pós Graduação

### MAPAS BRASIL

BAM 20x20 km Eta 05x05 km WRF 05x05 km Ensemble Ensemble Controle Meteogramas



### Porcentagem do Tempo de Execução relacionado ao TUPÃ

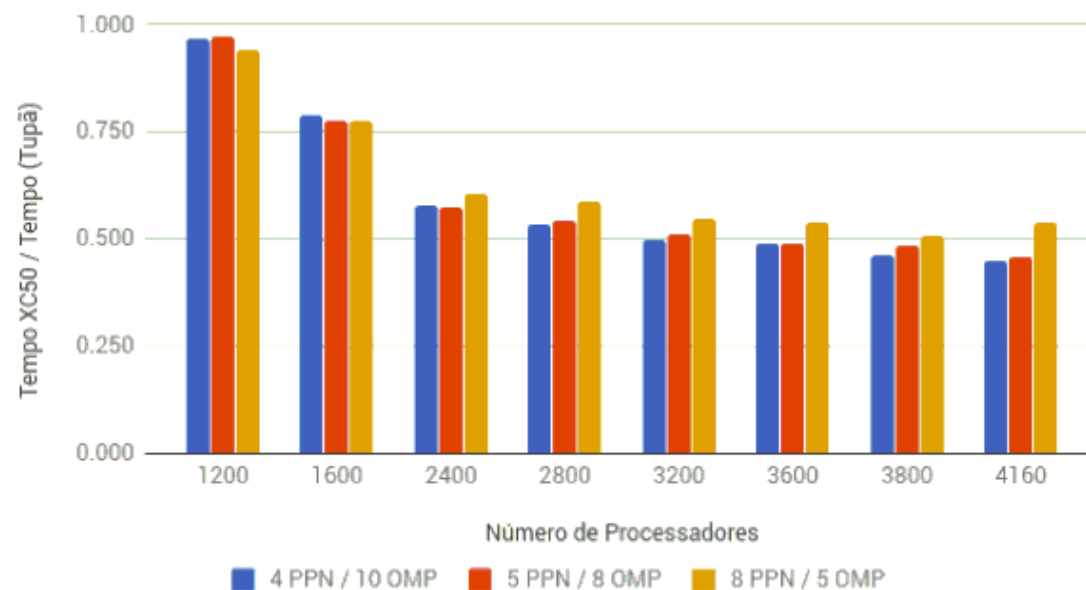


Fig.2: Relação entre o tempo de execução operacional no Tupã XE6 e no upgrade XC50 (Tempo XE6/Tempo XC50), com as diferentes configurações de MPI por nó (PPN) e threads OMP (OMP).

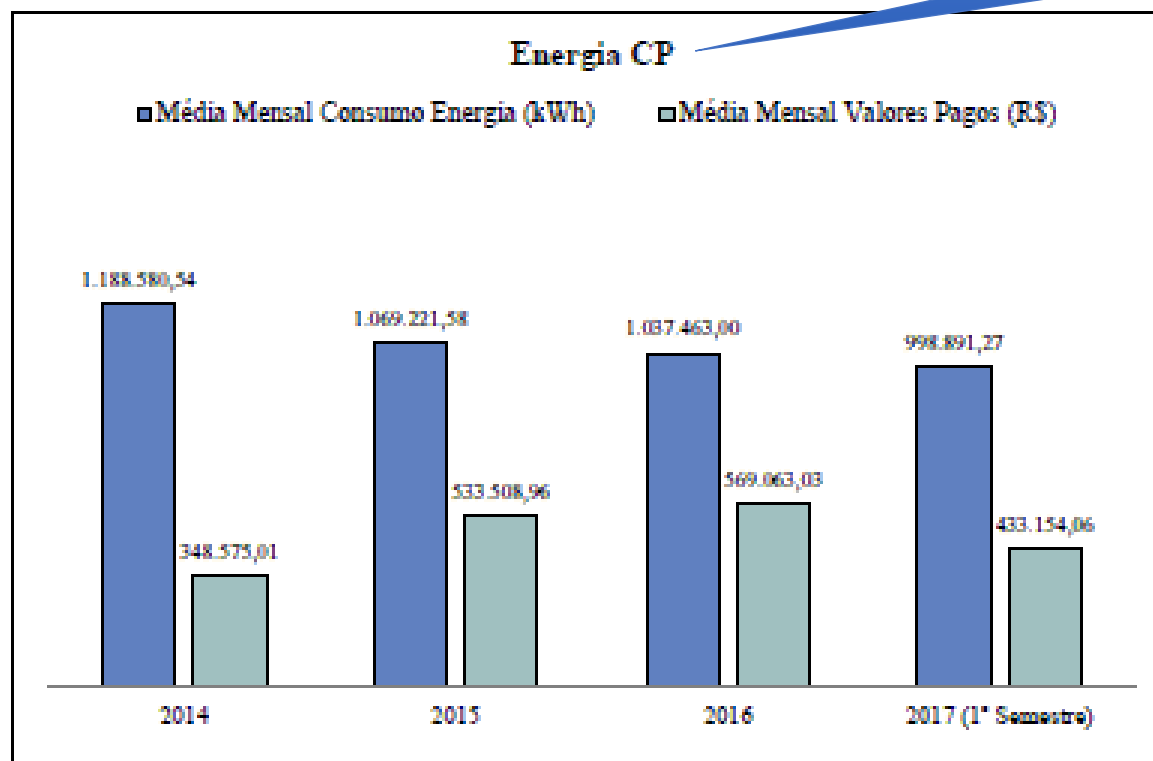


# Tipo de Supercomputador e Consumo de Energia Elétrica



No Gráfico 8 observa-se a comparação entre a redução no consumo e os pagamentos realizados para o período de 2014 a junho de 2017 pela unidade CP.

~70% gasto pelo tupa



**Modelo de PNT e Supercomputadores mais eficientes promovem um consumo menor de energia,**

Gráfico 8 – Média Mensal do Consumo de Energia em kWh de janeiro de 2014 a junho de 2017 no INPE-CP