Lawrence Livermore National Laboratory

ISCR Institute for Scientific Computing Research

CASC

July 16, 2001

Parallel and Robust Multigrid Techniques on Structured Grids

Ignacio Martín Llorente
www.dacya.ucm.es/nacho

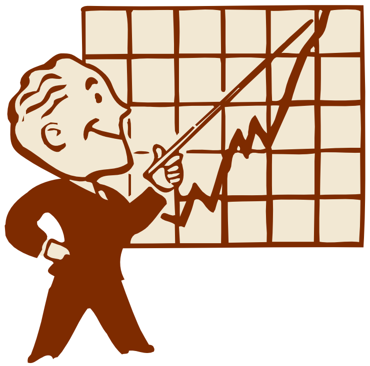
Departamento de Arquitectura de Computadores y Automática

Departamento de Arquitectura de Computadores y Automática
Facultad de Ciencias Fïsicas
Universidad Complutense de Madrid
Spain

¿Do the scientific community really need robust methods for their simulation codes?

"Civilization advances by extending the number of important operations which we can perform without thinking about them."

Alfred North Whitehead
An Introduction to Mathematics, 1911



The need for speed:

- The design cycle for aerospace vehicles must be significantly shortened
- CFD will continue to replace a larger portion of wind tunnel and flight testing

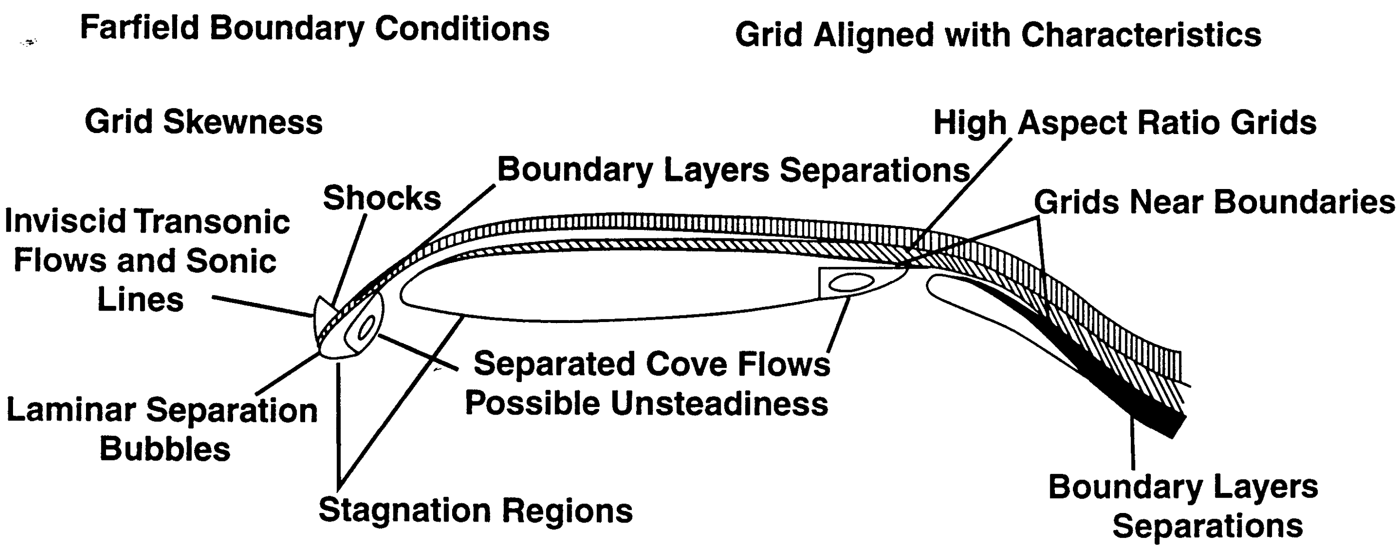
Progress in numerical simulation requires the effective combination of advances in:

- Algorithm development
- Understanding the underlying physics
- Computer hardware

¿Can we obtain an optimal (algorithmic and architectural) solver for CFD problems?

Textbook Multigrid Barriers

Bran98

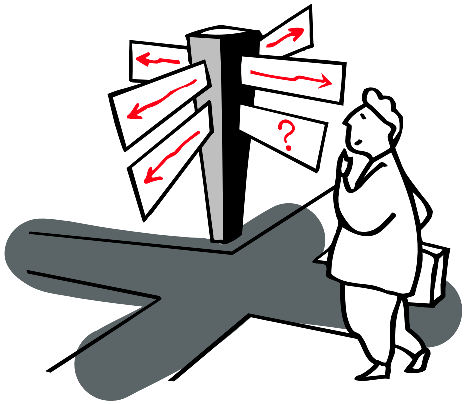


Personal opinion:

It does not exist a key method optimal for all cases

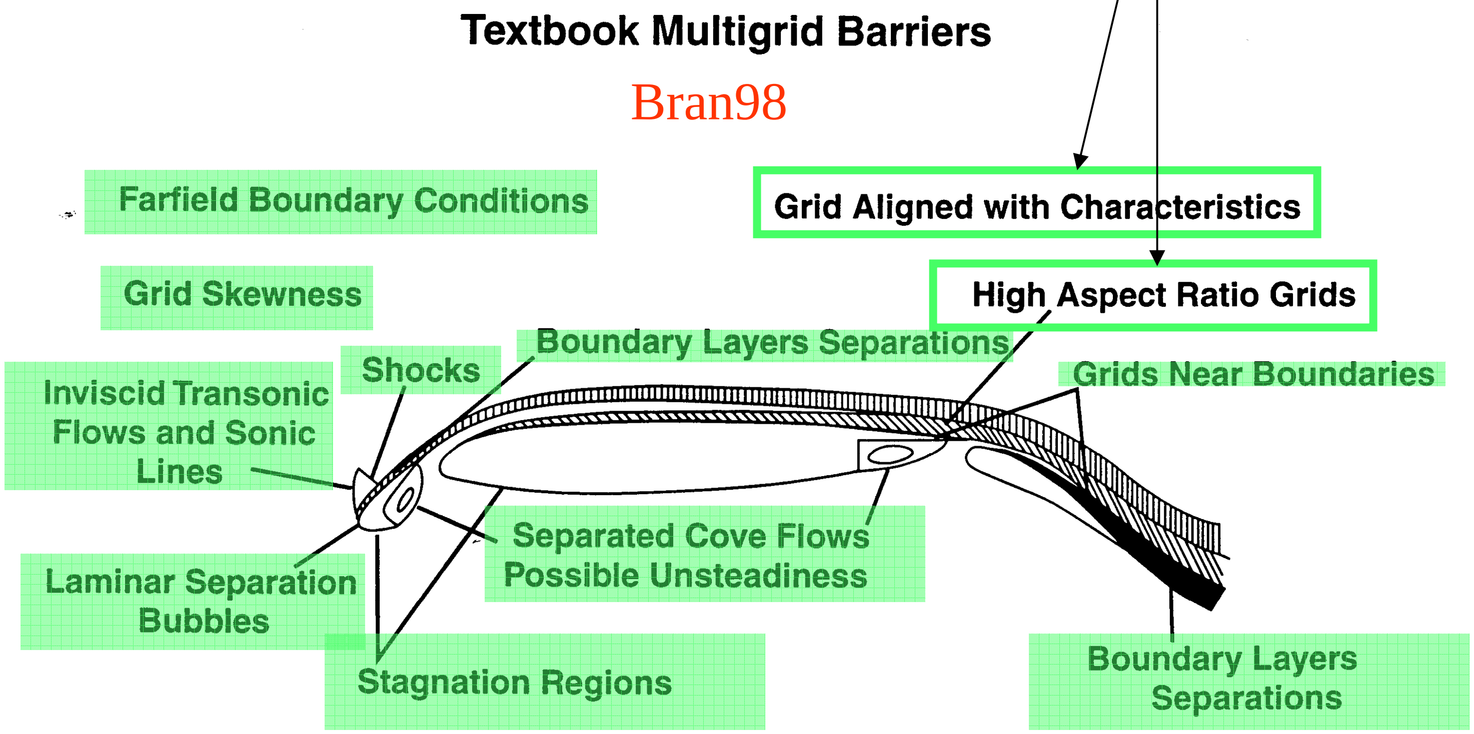
There are many possible paths towards faster CFD

- Precondiotining
- Separation of elliptic and hyperbolic parts
- ...



We present one alternative based on implicit schemes and semicoarsening

In particular, we present one building block to achieve optimal CFD by relieving



Algorithmic and Architectural issues of

Part 1: Multigrid for the diffussion problem

↓

Part 2: Extension to the convection problem

↓

Part 3: Extension to the incompressible Navier-Stokes equations

on structured grids

Properties that describe the efficiency of an iterative method:

- **Convergence rate:** Reduction in error per cycle
- **Computational work per cycle:** Operation count to execute one cycle

What is a robust method?

=> Method able to efficiently solve a wide range of problems

We should define it more precisely by setting up a set of suitable test problems

=> Multigrid technique with a convergence rate independent of:

- Grid size
- Anisotropy
- And for our case example: Reynolds number and yaw angle

Properties that are essential to use the full potential of current and future computing systems in each single iteration

- **Memory usage:** Memory waste
- **Parallel efficiency:** How the parallel setting is exploited?
- **Parallel scalability:** Is the parallel efficiency maintained for a higher number of processors?
- **Cache-memory exploitation:** Space and temporal locality in the memory accesses in order to reduce the number of cache misses

Parallel scalability is quite important in solving very large systems on massively parallel computers

=> How does the solver perform as both the number of processors and the grid size are increased?

Ways to increase the problem size with the number of processors

Accuracy critical scaling model

Memory-bounded (isomemory) (SuNi90)

Efficiency-bounded (isoefficiency) (GrGK93)

Time-bounded (isotime) (Gustafson, Gust88)

Time critical scaling model

Fixed-size (Amdahl's law)

Each scaling model has its own scalability metrics (LITi97, LITV96):

We focus on memory-bounded scaling model

$$E(N,p)=\frac{T(N,1)}{T(N,p)p}$$

WE PROPOSE TWO SCALIBILITY METRICS FOR MEMORY-BOUNDED SCALING:

Is the efficiency maintained as the problem size is increased linearly with the number of processors (*memory-bounded*)?

From (*N*,*p*) to a system *n* times larger (*nN*,*np*): $S_E(p,np)=\frac{E(nN,np)}{E(N,p)}=\frac{T(nN,1)T(N,p)}{T(N,1)T(nN,nP)}\frac{1}{n}$

Is the execution time maintained as the problem size is increased linearly with the number of processors (*memory-bounded*)?

From (*N*,*p*) to a system *n* times larger (*nN*,*np*): $S_T(p,np)=\frac{T(N,p)}{T(nN,np)}$ (*scaled efficiency*)

If the computational complexity of the algorithm is linear O(*N*) then $T(nN,1) = n\,T(N,1)$ and $S_E(p,np)= S_T(p,np)$

Iterative cycle working out of cache

Details of this research in

I. M. Llorente and F. Tirado, **Relationships between Efficiency an Execution Time of Full Multigrid Methods on Parallel Computers**, IEEE Transactions on Parallel and Distributed Systems, Vol. 8, Nº 6, 1997, pp. 562-573

I. M. Llorente, F. Tirado and L. Vázquez, **Some Aspects about the Scalability of Scientific Applications on Parallel Computers**, Parallel Computing, Vol. 22, pp. 1169-1195, 1996

¿Can we obtain an optimal solver for the diffussion operator on highly stretched grids?

Introduction

- Description of the model problem
- Components of multigrid
- Anisotropies in the discrete operator
- What is a robust solver?
- Review of robust alternatives

Comparation of two robust alternatives

- Convergence factor
- Memory requirements
- Cache-memory exploitation
- Convergence rate per work unit

Parallel implementation

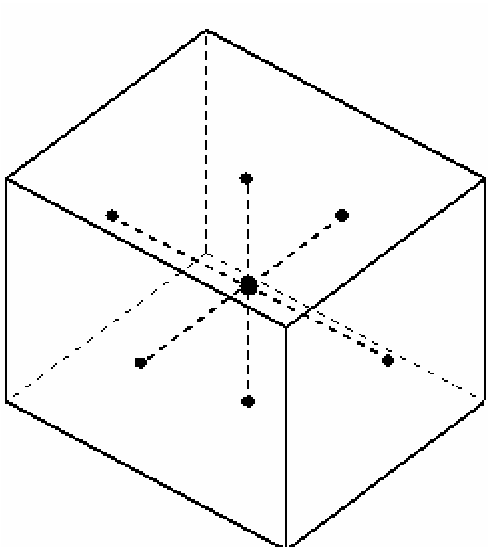
- Parallel architectures
- Parallel implementation of the alternating-plane approach
- Parallel implementation of the semicoarsening approach
- Architectural advantages of the 1-D decomposition

Conclusions of the first part

- **3-D Anisotropic Diffussion Equation on a rectangular domain with Dirichlet Boundary Conditions**

$$\alpha \frac{\partial^2 \phi}{\partial x^2} + \beta \frac{\partial^2 \phi}{\partial y^2} + \gamma \frac{\partial^2 \phi}{\partial z^2} = S_\phi(x,y,z)$$

- **Discretized by a Finite volume cell-centered 7-point operator**

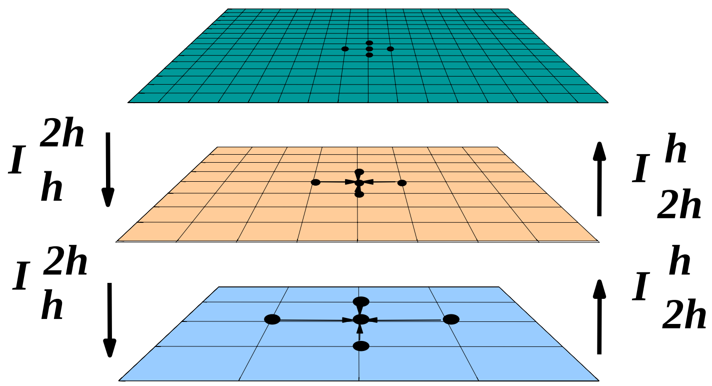


$$\frac{2\alpha}{\Delta x}(X_p \phi_{i+1jk} - (X_p + X)\phi_{ijk} + X\phi_{i-1jk})$$
$$+ \frac{2\beta}{\Delta y}(Y_p \phi_{ij+1k} - (Y_p + Y)\phi_{ijk} + Y\phi_{ij-1k})$$
$$+ \frac{2\gamma}{\Delta z}(Z_p \phi_{ijk+1} - (Z_p + Z)\phi_{ijk} + Z\phi_{ijk-1}) = S_{\phi_{ijk}}$$

Two fundamental principles:

- **relaxation : standard iterative method / smoother**
  It is used to eliminate oscillatory components of the error

- **coarse grid correction**
  On a coarser grid, low-frequency components appear more oscillatory



standard coarsening : doubling the mesh size in all coordinate directions

- **FAS (Full approximation scheme)** scheme
  It can be applied to solve **non-linear equations**

General FAS (γ_1 ,γ_2) – V cycle for the solution of the system

$$L_0 u_0 = f_0$$

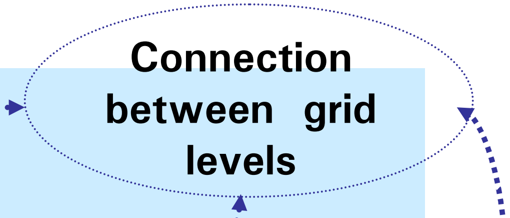
- **Pre-smoothing :** Apply γ_1 sweeps of the smoothing method on $\mathbf{L}_{0l}u_{l0} = f_0$

FOR level = 1 TO L-1

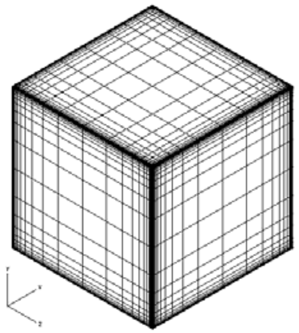
- **Computation of residual** $r_{level-1} = f_{level-1} - \mathbf{L}_{level-1} u_{level-1}$
- **Restriction** of residual $r_{level} = \mathbf{R}\, r_{level-1}$
- **Restriction** of current approximation $\mathbf{u'}_{level} = \mathbf{R}\, u_{level-1}$
- **Computation of right-hand side** $f_{level-} = r_{level} + \mathbf{L}_{level} \mathbf{u'}_{level}$
- **Pre-smoothing :** Apply γ_1 sweeps of the smoothing method on $\mathbf{L}_{level}u_{level} = f_{level}$

FOR level = L-2 TO 1

- **Correction** of current approximation $u_{level} = u_{level} + \mathbf{P}\, (u_{level+1} - \mathbf{u'}_{level+1})$
- **Post-smoothing :** Apply γ_2 sweeps of the smoothing method on $\mathbf{L}_{level}u_{level} = f_{level}$



◆ Our discrete operator

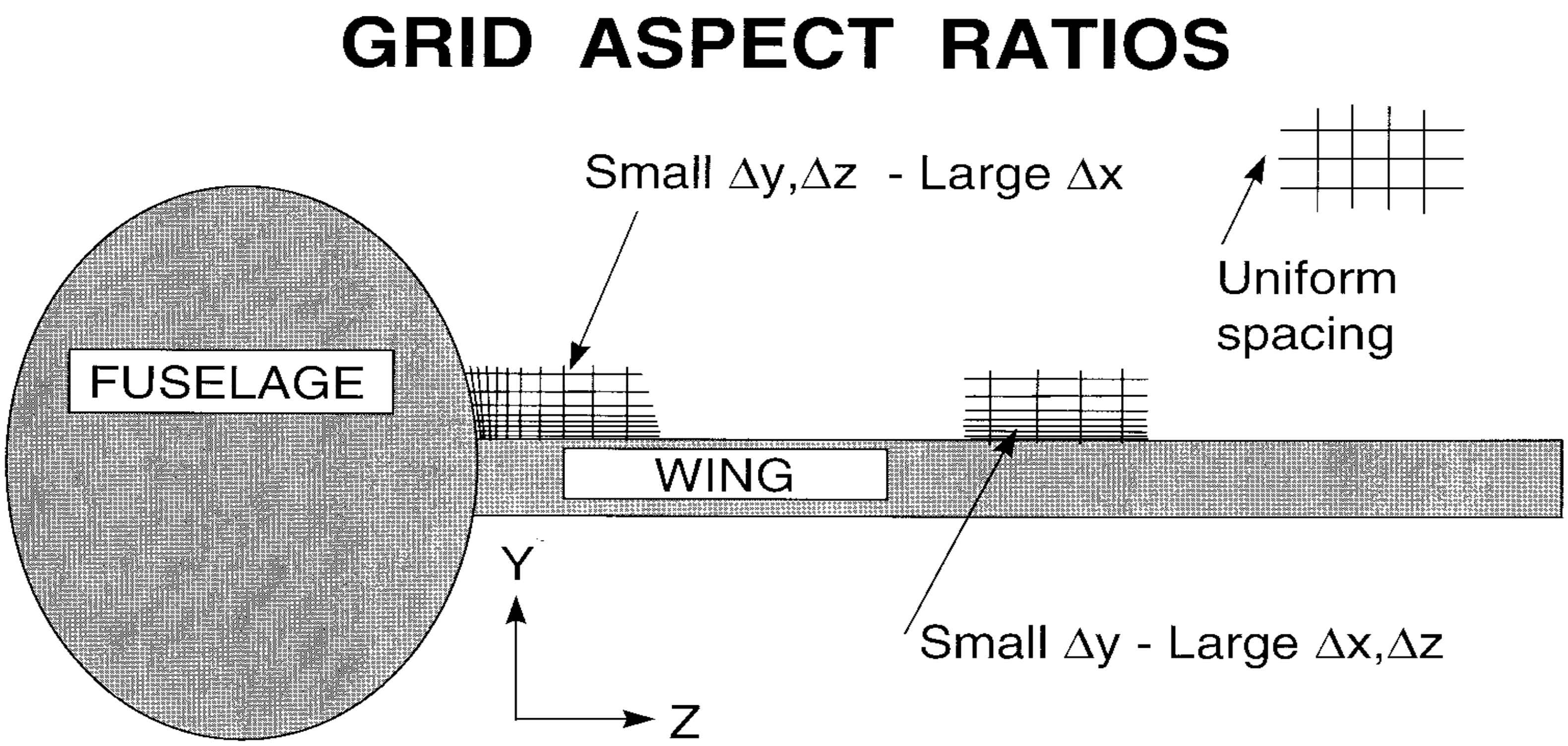


$$\frac{2\alpha}{\Delta x}\left(X_p\phi_{i+1jk} - (X_p + X)\phi_{ijk} + X\phi_{i-1jk}\right) + \frac{2\beta}{\Delta y}\left(Y_p\phi_{ij+1k} - (Y_p + Y)\phi_{ijk} + Y\phi_{ij-1k}\right) + \frac{2\gamma}{\Delta z}\left(Z_p\phi_{ijk+1} - (Z_p + Z)\phi_{ijk} + Z\phi_{ijk-1}\right) = S\phi_{ijk}$$

•Physical Anisotropy :
Different **c**oefficients in each direction

• Computational Anisotropy :
Use of a exponential stretched grid in the three dimensions.

Grid stretching in order to pack points into regions with large gradients



- The **convergence of standard multigrid** (*point smoother combined with full coarsening*) deteriorates dramatically in presence of anisotropies

- **Brandt's fundamental block relaxation rule** states that all strongly coupled unknowns (coordinates with relative larger coefficients) should be relaxed simultaneously

  ➢ Implicit (line or plane) relaxation combined with full coarsening

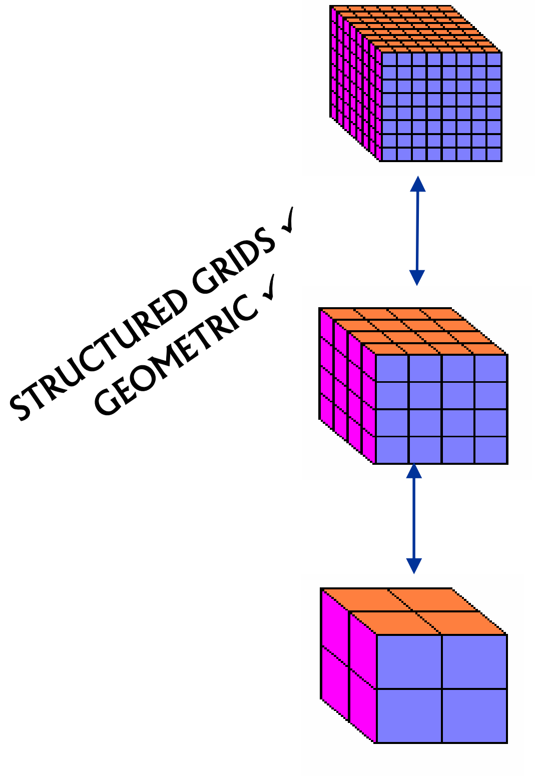
- **However if**

  - The nature of the anisotropy is not known beforehand, or
  - The aspect ratios or the equation coefficients differ with each other throughout the computational domain

  => A robust solver is needed

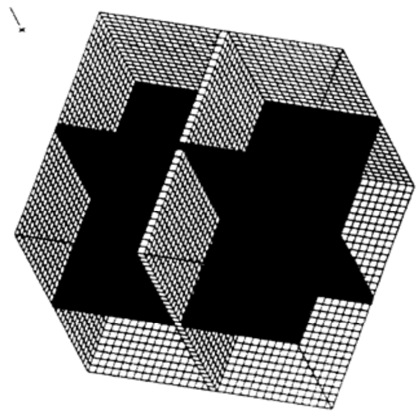
Alternating-direction plane smoothers in combination with full coarsening

(Bran84, LlMe00, ...)



X , Y and Z coarsening

Explore all possibilities

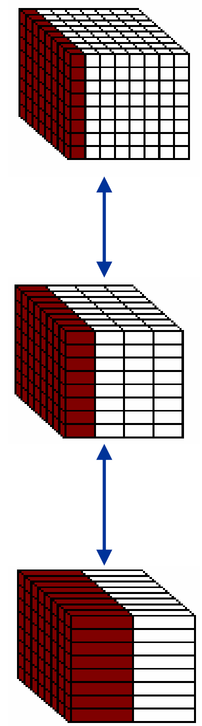


☺ Fast convergence rate that improves with stretching	☹ Resolution of the 2-D problems ☹ Parallel implementation of the 2-D solvers

Block smoothing in combination with semicoarsening

(DMRR89, Scha98, BrFJ00, ...)

STRUCTURED GRIDS ✓
GEOMETRIC ✓



X coarsening

☺	**1-D parallel implementation**
☺	**Easier to implement than the alternating-plane approach**

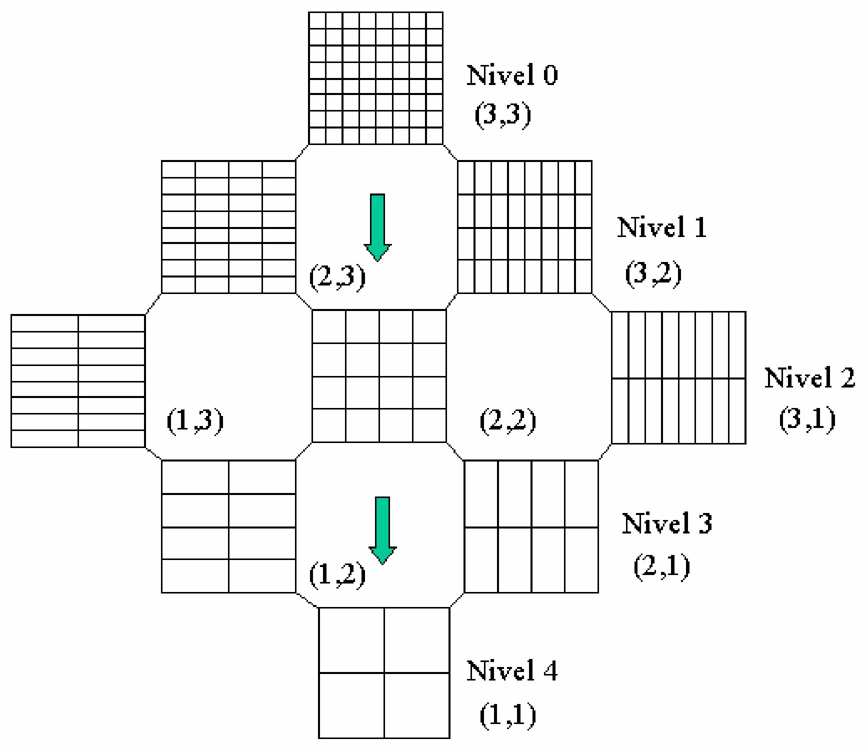
☹	**Resolution of the 2-D problems**
☹	**The 2-D problem size remains fixed in the coarsening process (memory waste and higher work per cycle)**

Recombination of the corrections of more than one semicoarsening grid

multiple semicoasening

(Muld89, OvRo93,...)

STRUCTURED GRIDS ✓
GEOMETRIC ✓



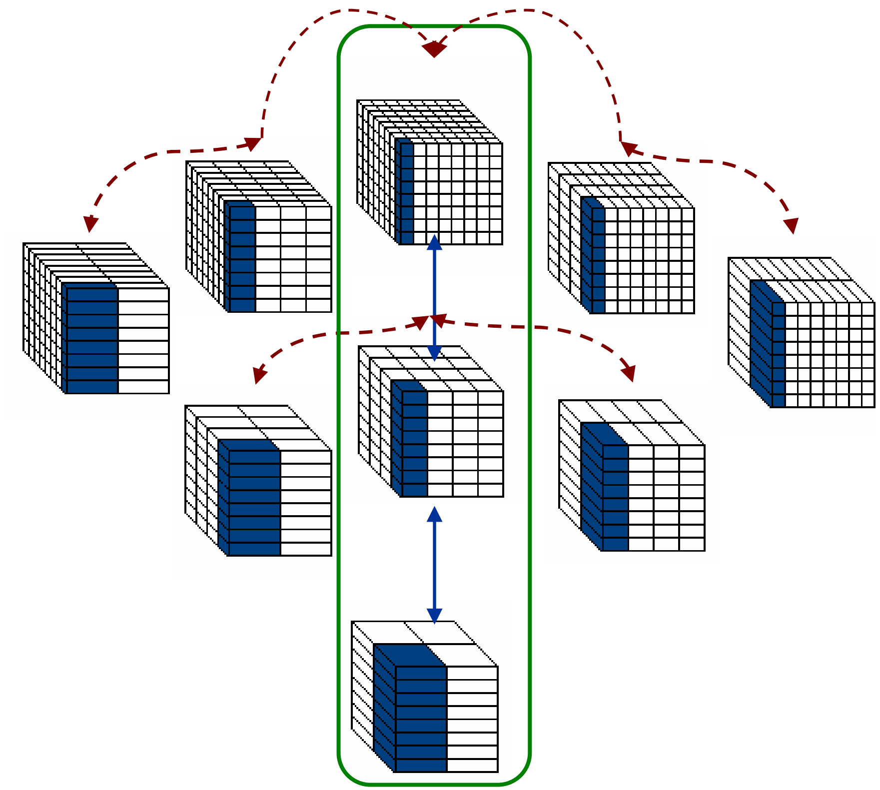
☺ **Two level parallelism**

☹ **Difficult 3-D generalization**
☹ **Recombination operator**
☹ **Memory waste**

Standard coarsening combined with a semicoarsened smoother
flexible multiple semicoasening

(Oost95, Stub97, ...)

STRUCTURED GRIDS ✓
GEOMETRIC ✓



X and Y coarsening with line relaxation

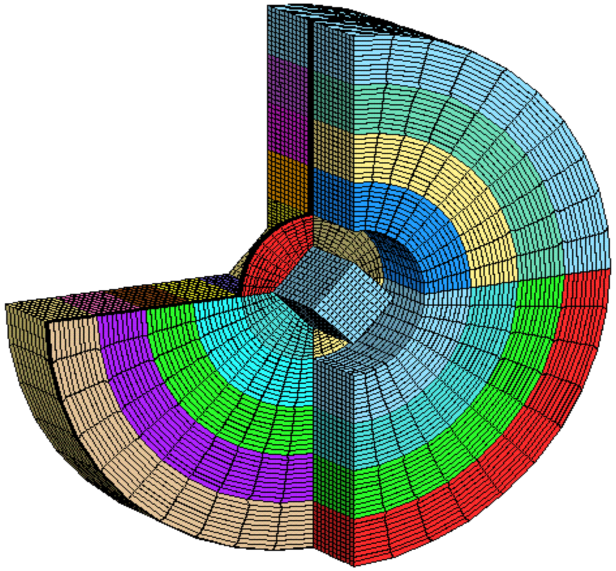
☺ Easier parallel implementation	☹ Memory management

Point-wise smoothing combined with a fully adaptive coarsening process

algebraic multigrid

(BrMR82, RuSt87, CFHJ00, ...)

UNSTRUCTURED GRIDS ✓
ALGEBRAIC ✓



An Algebraic Multigrid Tutorial, Van Emde Henson

☺ **Black box**	☹ **Parallel implementation** ☹ **Setup time** ☹ **Data management**

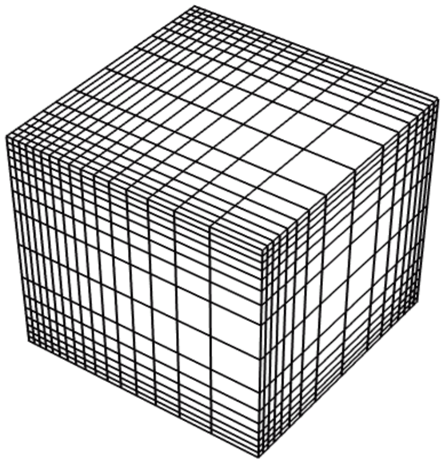
We focus on:

- **Alternating-plane smoother combined with full coarsening** (LlMe00, ...)
- **Plane smoother combined with semicoarsening** (PSEL01, ...)

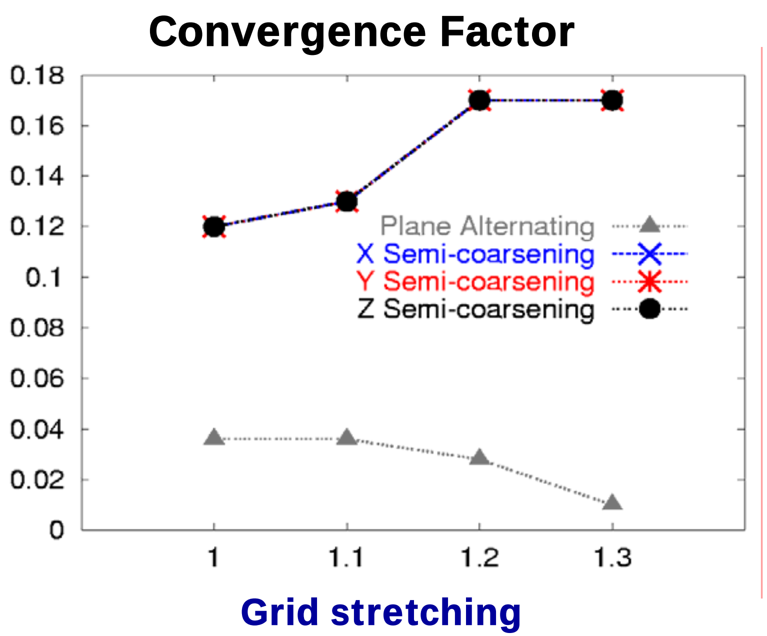
Results obtained with:

- Homogeneous problem with random initial guess
- V(1,1) cycle
- Restriction done by full weighted operator
- Trilinear (*plane alternating*) and linear (*semicoarsening*) interpolation for the prologator
- Zebra plane relaxation

Computational Anisotropy : 3-D exponential stretched grid



Simulations for the isotropic equation on a 64x64x64 grid with different stretching factors.



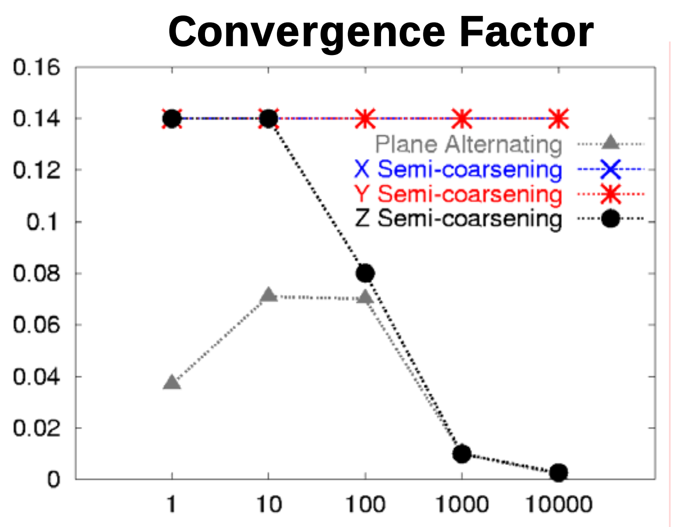
- Each **semi-coarsening** approach exhibits the same convergence factor. There is no privileged direction
- The **alternating-plane** smoother improves its convergence factor as the stretching grows

Alternating-plane

Physical Anisotropy: Coefficients are increased in two directions

$$a\frac{\partial^2 \phi}{\partial x^2} + b\frac{\partial^2 \phi}{\partial y^2} + c\frac{\partial^2 \phi}{\partial z^2} = 0$$

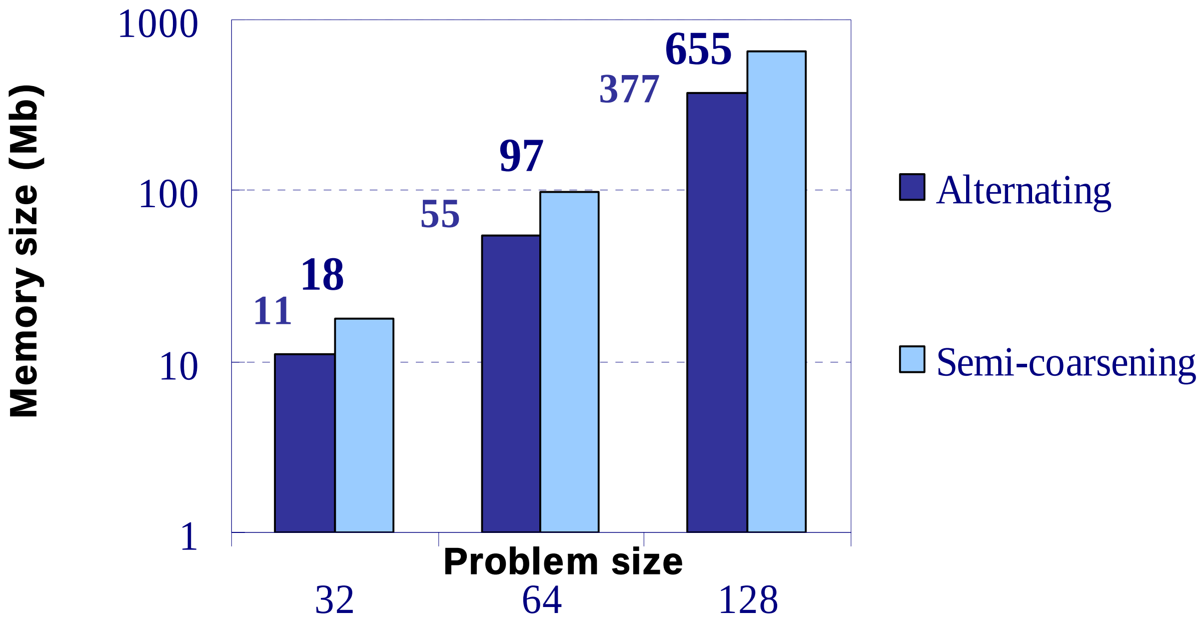
Simulations for the anisotropic equation on a uniform 64x64x64 grid



c=1 a = b = 1,10,100,1000,10000

- Best **Semi-coarsening** procedure the one that keeps coupling of connected unknowns
- **Alternating-plane** smoother becomes a direct solver for high anisotropies

Alternating-plane and semicoarsening solving the connected unknowns



- Memory requirements of the semi-coarsening approach are about twice as large as the alternating-plane approach

Alternating-plane

Factors to consider:

- Floating point operations per Multigrid cycle
- Cache memory exploitation

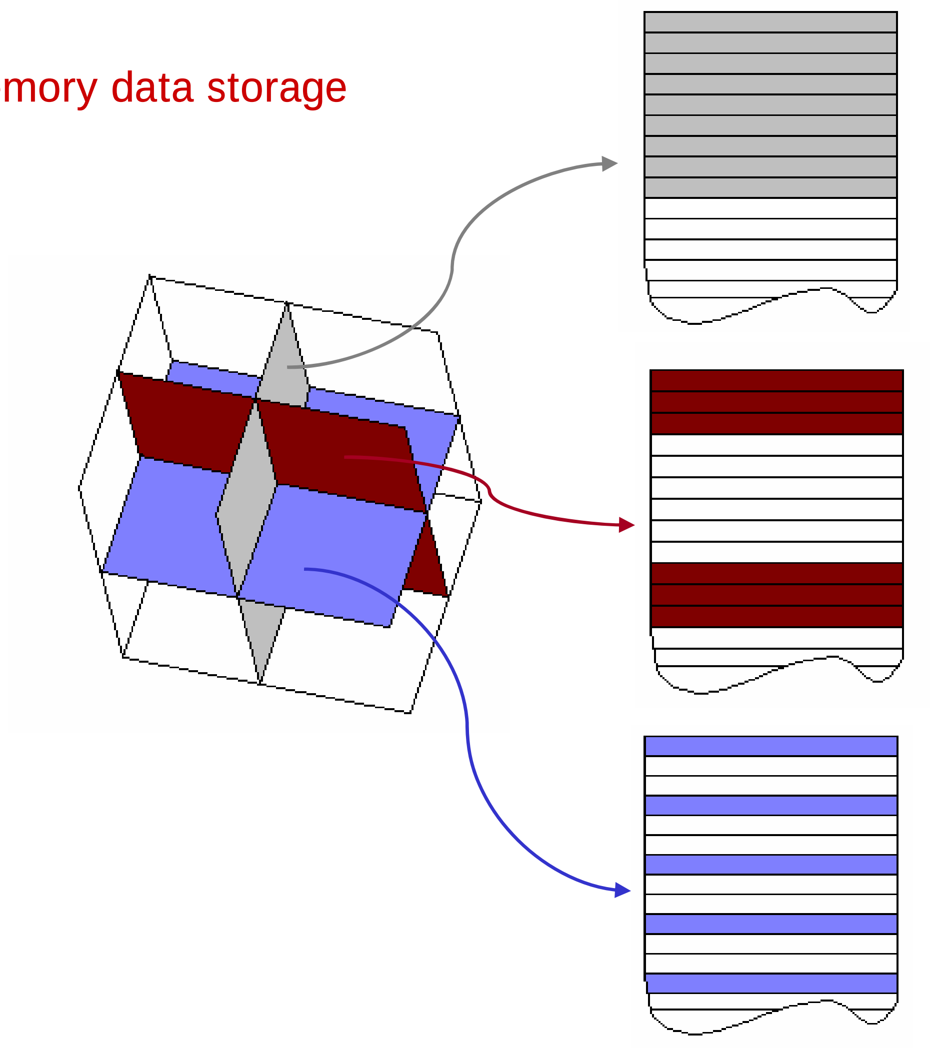
Floating point operations per Multigrid cycle:

- 26 % larger on the alternating-plane approach

Alternating-plane

Cache memory exploitation

Memory data storage



• Plane **YZ** (X= constant)
Suitable for X-semi-coarsening

• Plane **XZ** (Y= constant)
Suitable for Y-semi-coarsening

• Plane **XY** (Z= constant)
Suitable for Z-semi-coarsening

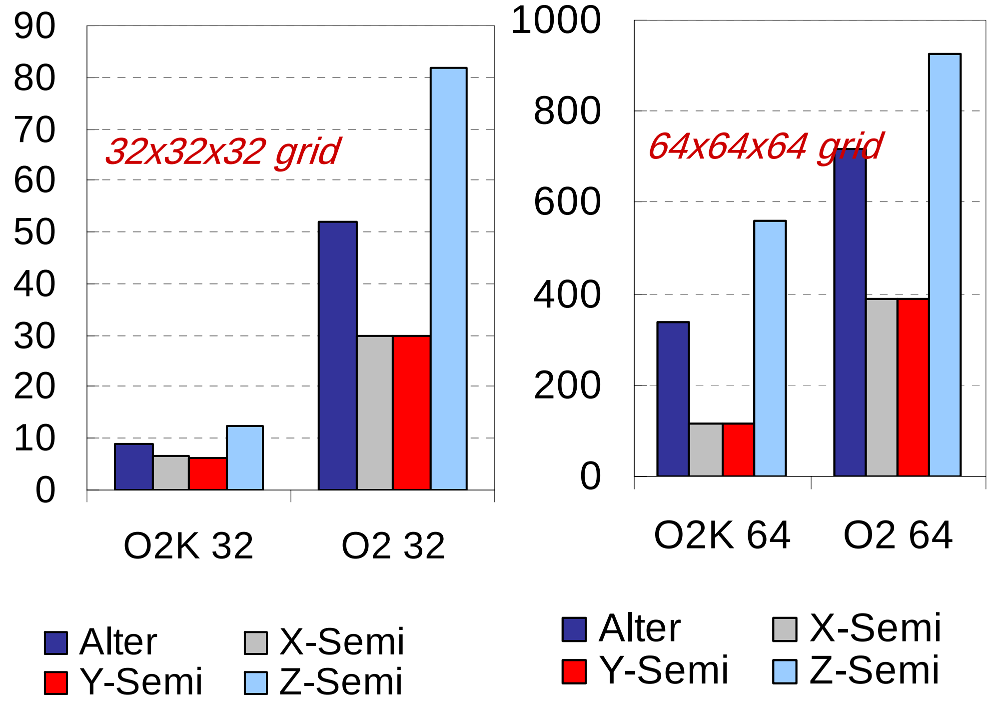
X and Y semicoarsening

Cache memory exploitation

- **Y-Semicoarsening** and **X-Semicoarsening** have the same behavior (L2 cache block size).
- **Z-Semicoarsening** produces more L2 cache misses due to the memory storage scheme.

- **SGI O2 Workstation**
  L2 : 1-MB unified -Ofast=ip32_10k
- **SGI Origin 2000 system (O2K)**
  L2 : 4-MB unified -Ofast=ip27

Millions of L2 cache Misses



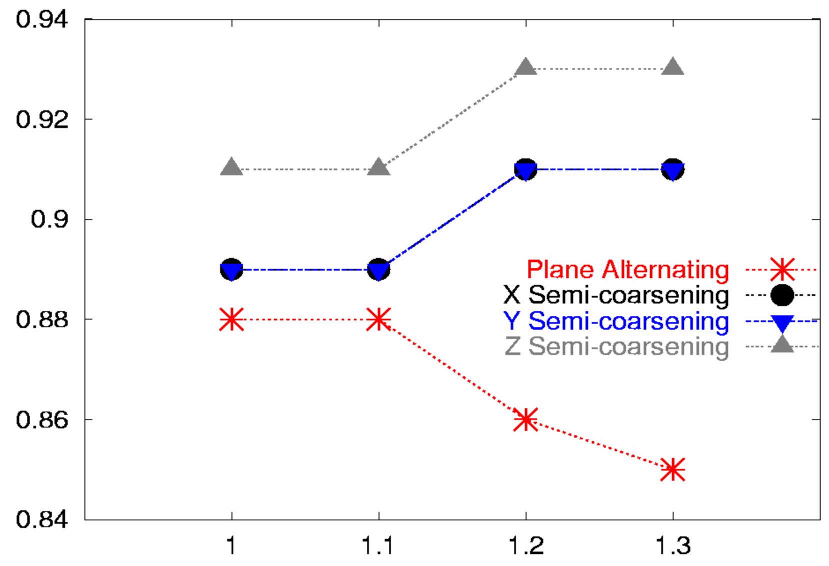
Cost per cycle : 38% larger on the **alternating-plane** approach

Definitive metric

Work Unit: Time consumed in computing the system metrics on the finest level

Simulations for the isotropic equation on a 64x64x64 grid with different stretching factors

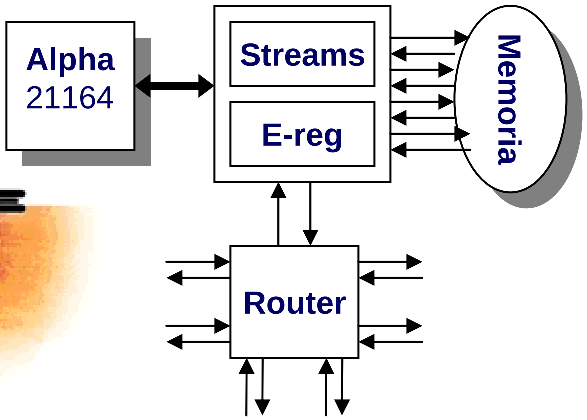
Convergence Factor per Work Unit (O2K)

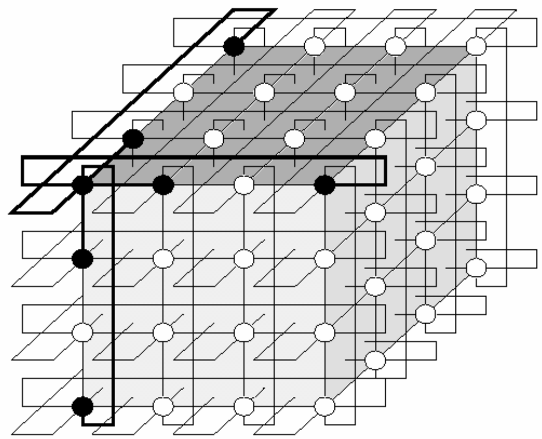


Grid stretching

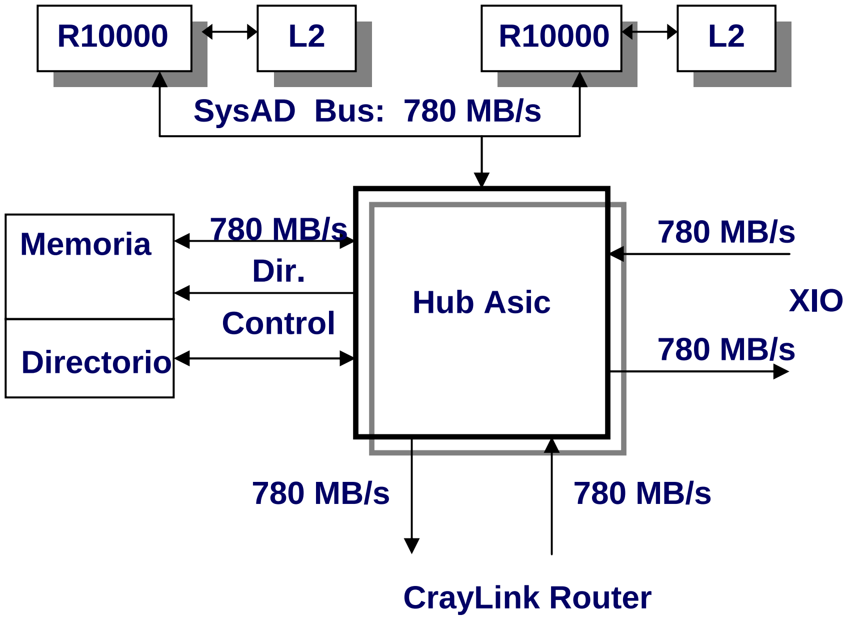
- The alternating-plane approach reduces the same amount of error in less time

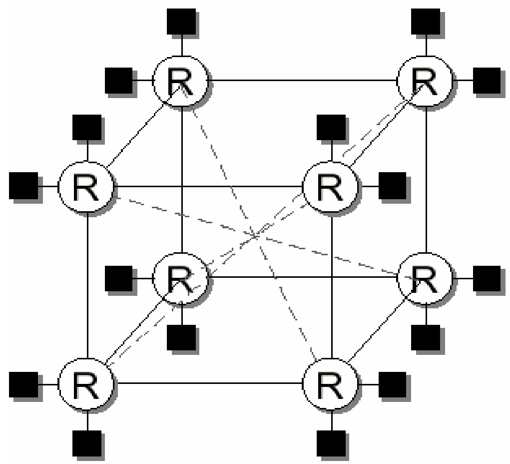
CRAY T3E: 48 Alpha 21164





SGI Origin 2000: 32 MIPS R10000





DD + MG : Domain decomposition

- **Domain decomposition** is applied on the finest grid and multigrid inside each block
    - It **deteriorates** the **convergence** of the sequential algorithm
    - It implies **fewer communications** since exchanges of data are only required on the finest grid level

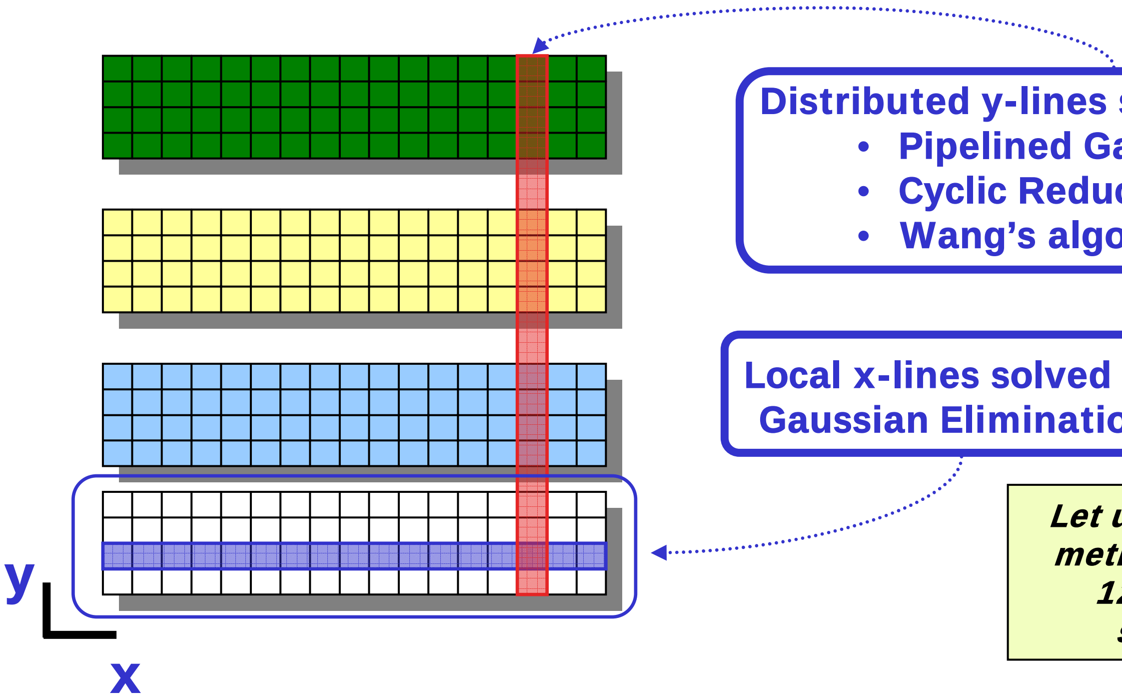
MG+ DD : Grid partitioning

- **Domain decomposition** is applied at each level
    - It **retains** the **convergence** of the sequential algorithm
    - It implies **more communication** overheads than domain decomposition approaches since exchanges of data are required on each grid level

- Regardless of the data partitioning applied, it requires the solution of **tridiagonal systems** of equations **distributed** among the processors because the combination of alternating-line smoothers and full coarsening is applied to solve the planes

- Solving the line is the **most time consuming task** of our code (around 80 %)

- An **alternating-line 2D solver** can be used for **estimating the parallel efficiency** of the whole application

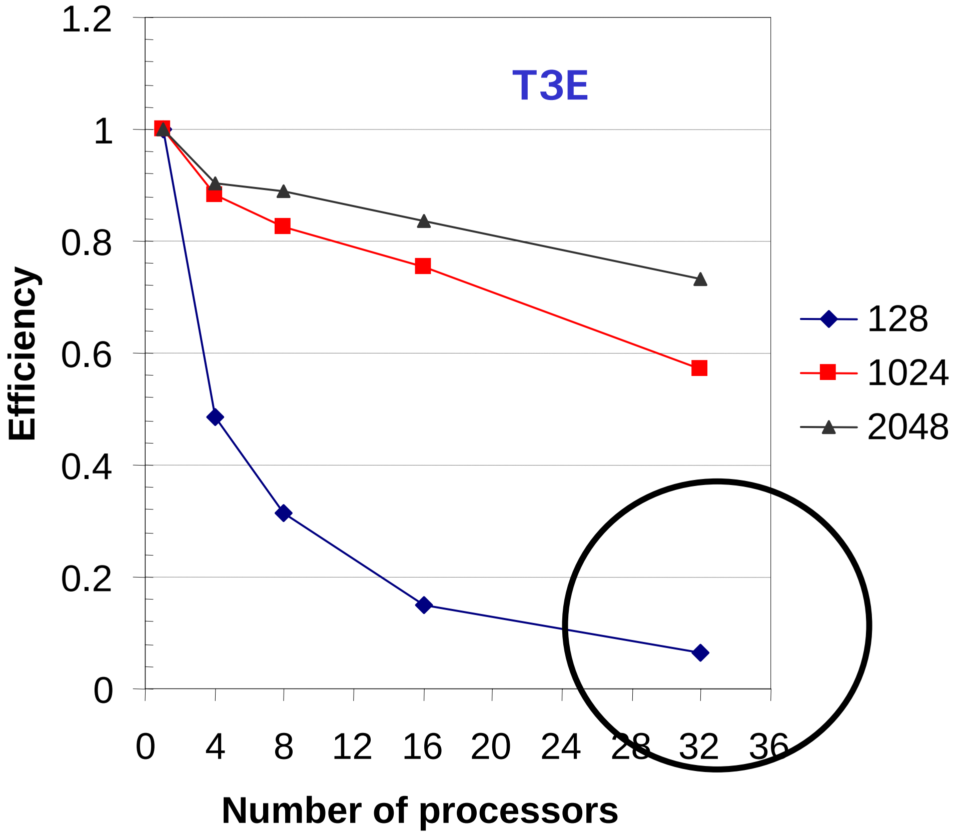


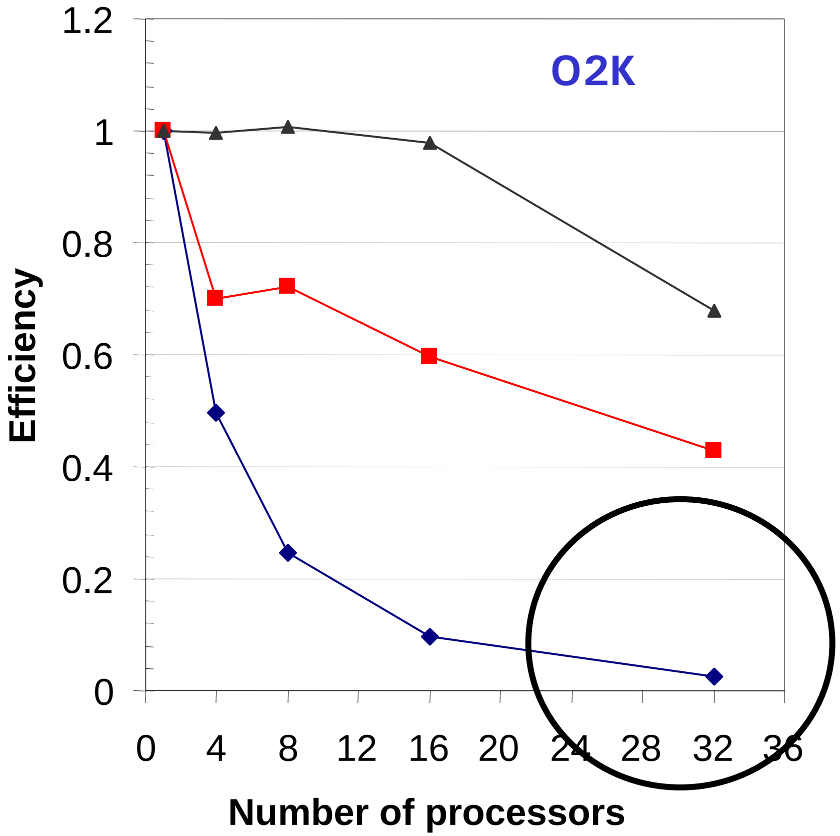
Distributed y-lines solved by:
- **Pipelined Gaussian Elimination**
- **Cyclic Reduction**
- **Wang's algorithm**

Local x-lines solved by Gaussian Elimination

Let us study the efficiency of these methods on parallel computers for 128x128 and 1024x1024 2-D systems (1-D partitioning)

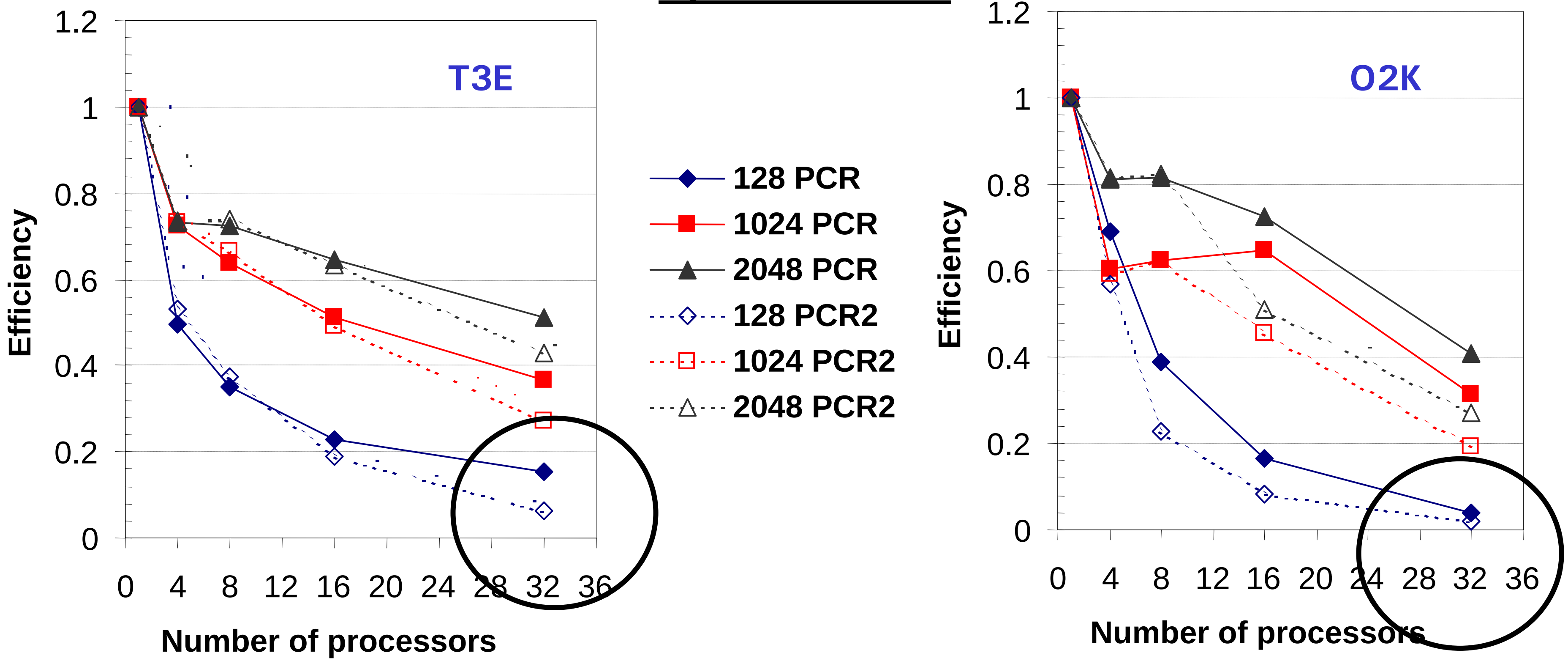
Pipelined Gaussian Elimination (PGE)





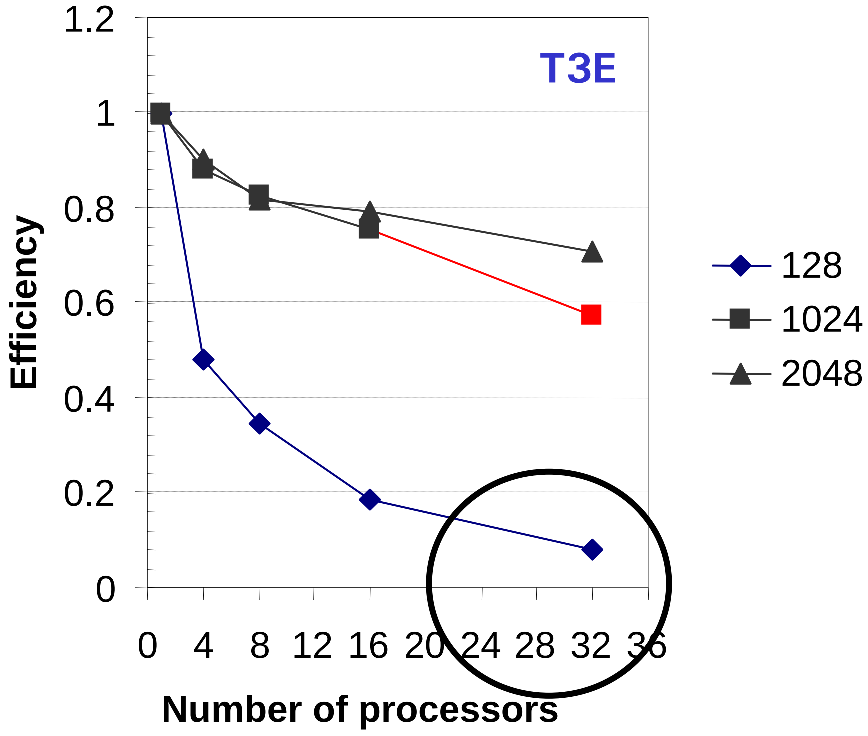
- High efficiencies for large 2D problems
- Current memory limits do not allow 3D problems to be solved where their corresponding 2D planes are big enough to obtain satisfactory efficiencies

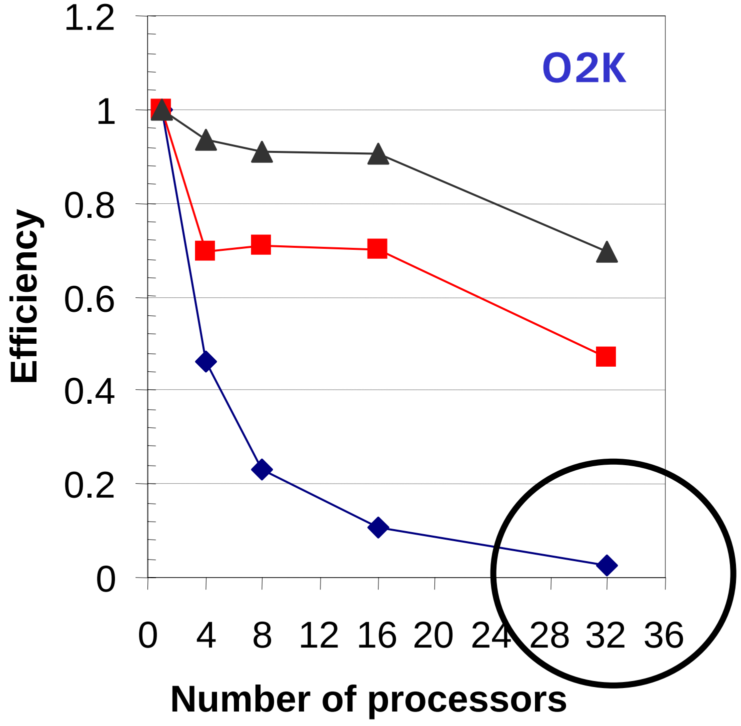
Cyclic Reduction



- High efficiencies for large 2D problems
- Current memory limits do not allow 3D problems to be solved where their corresponding 2D planes are big enough to obtain satisfactory efficiencies

Wang's algorithm



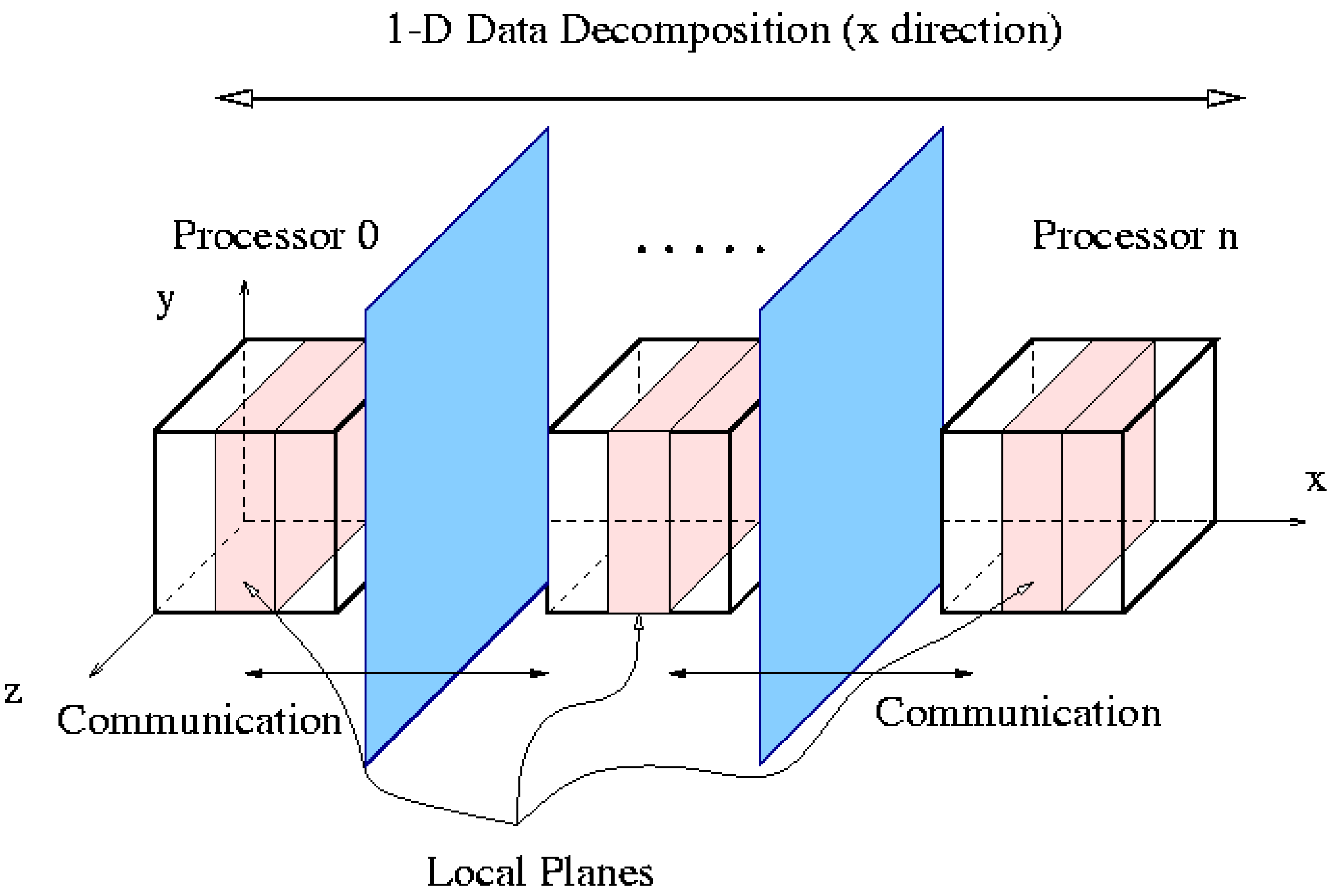


- High efficiencies for large 2D problems
- Current memory limits do not allow 3D problems to be solved where their corresponding 2D planes are big enough to obtain satisfactory efficiencies

Details of this research in

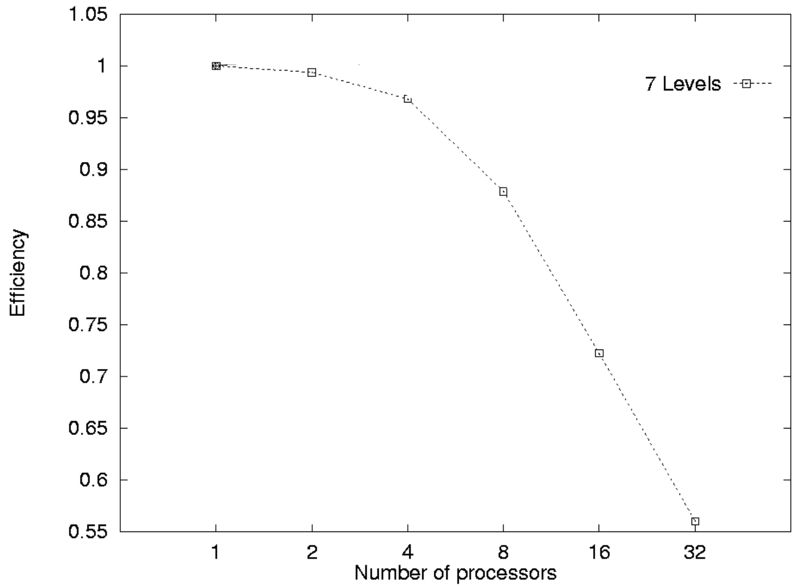
D. Espadas, M Prieto, I. M. Llorente and F.Tirado, **Solution of Alternating-line Processes on Modern Parallel Computers**, In Proceedings of the 28th. International Conference on Parallel processing, ICPP '99. Aizu-Wakamatsu (Japan), September 1999. Published by the IEEE Computer Society, pp. 208-215

- A **1-D data decomposition** on the semicoarsened direction can be chosen so that a **parallel tridiagonal** solver is **not needed**

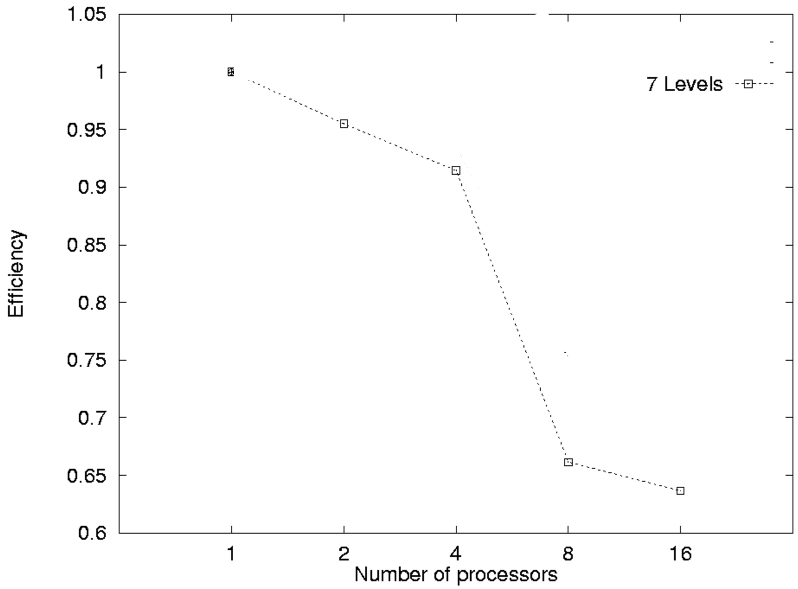


One V-cycle for the isotropic equation on a 64x64x64 grid

Cray T3E

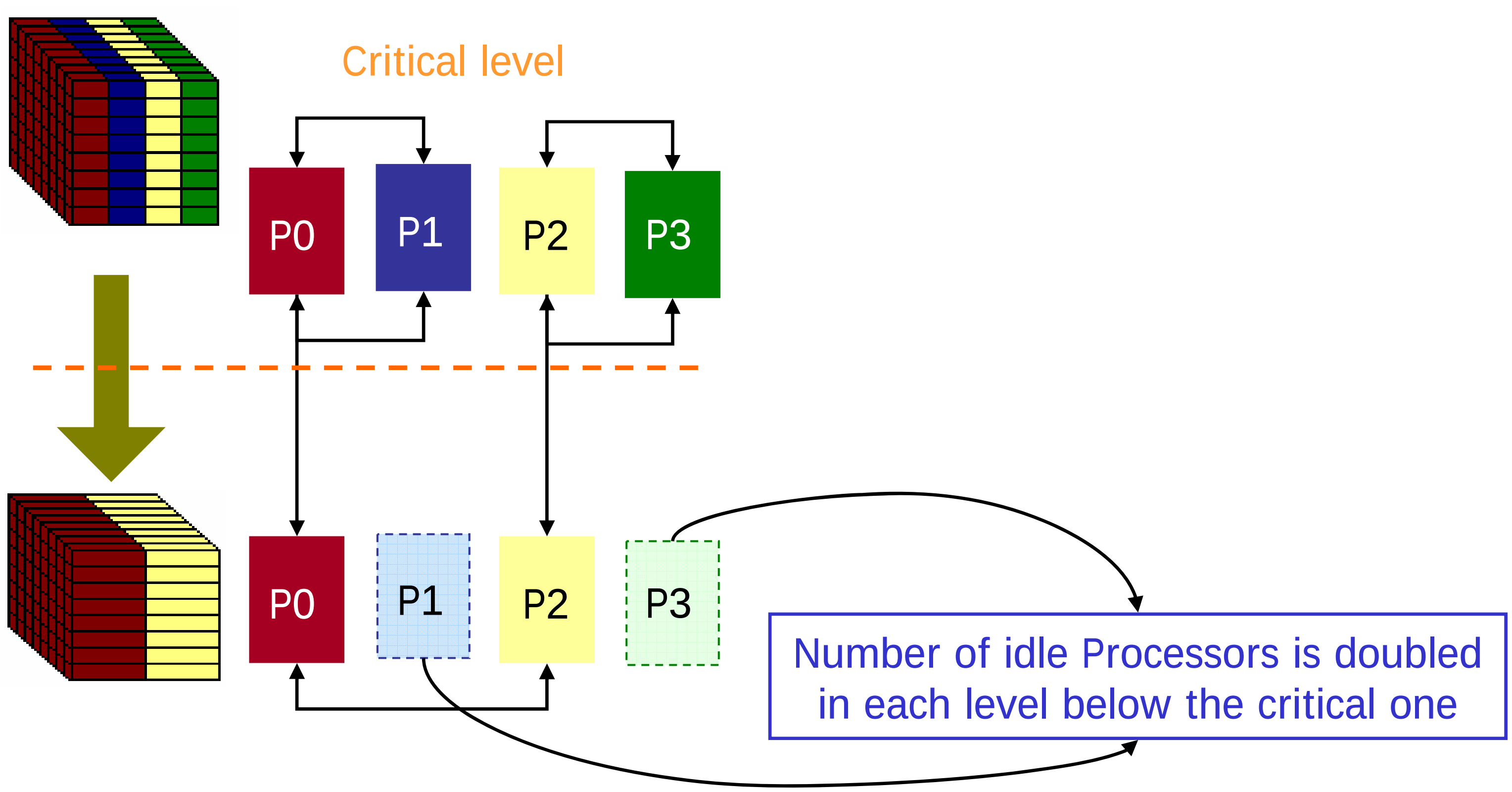


Origin 2000



- Can the efficiency be improved?

The efficiency deteriorates for a large number of processors due to the load imbalance below the critical level (level with one plane per processor)



Alternatives to relieve the load imbalance

Agglomeration on Coarsest Grid (MoDe99, …):

- Grids below the critical level are solve on a single processor
  - ☞ It could reduce the communication overhead
- It increases the execution time because the plane-wise smoother is very expensive

Parallel Superconvergent Multigrid (FrMc88, …):

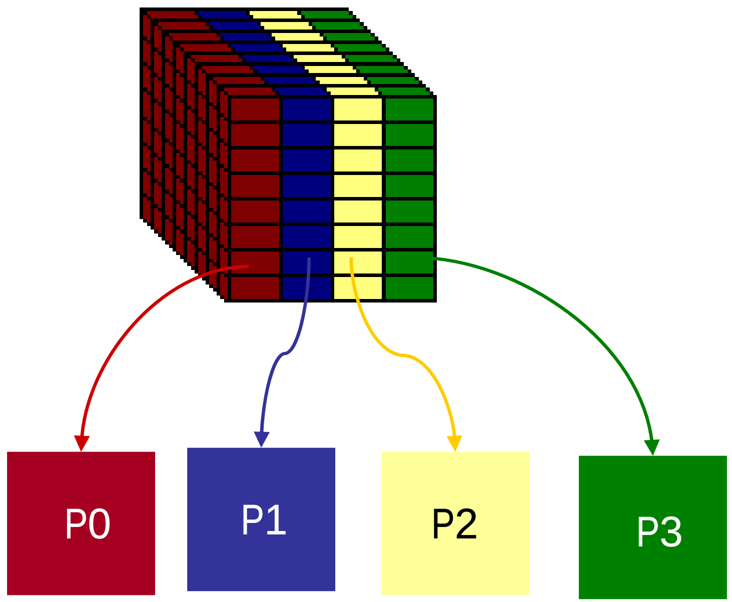
- It keeps the processor busy below the critical level using multiple coarse grids
  - ☞ It could improve the convergence rate
- It increases the execution time because of the time needed for merging the solutions

U-cycle Method (XiSc97, …):

- It solves the problem on the critical level by applying an certain number of sweeps
  - ☞ It could avoid idle processors
- It increases the execution time in the simulation because convergence rate of each cycle becomes lower

U-cycle approach

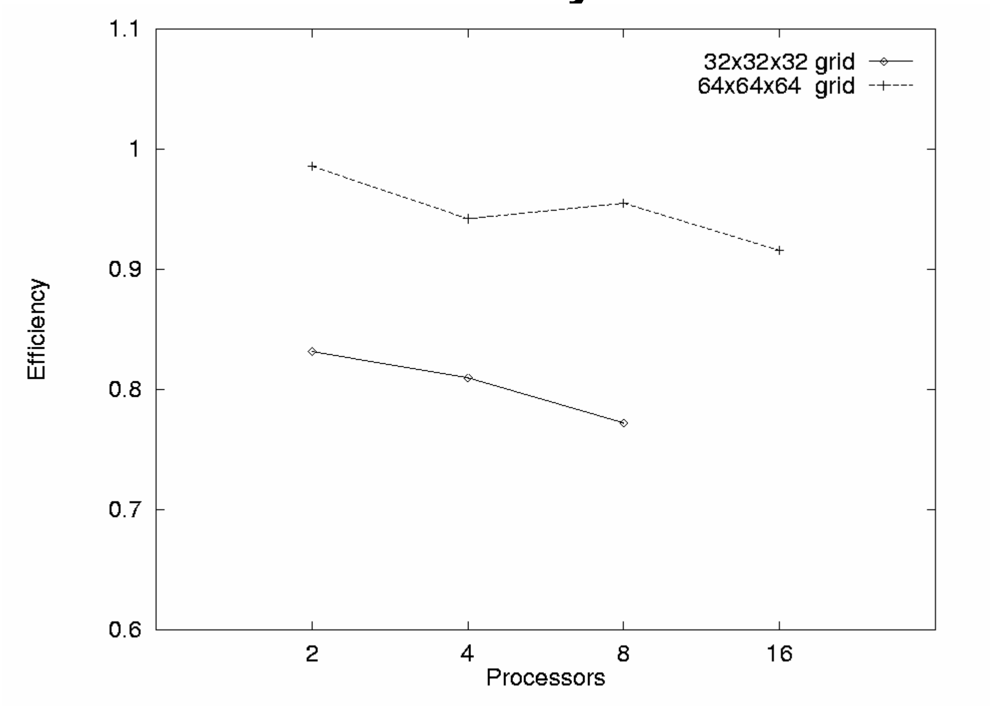
- The number of grid levels has been fixed so that each **processor has one plane** on the coarsest level **(critical level)**



- Critical Level for 4 processors
- **Coarsest** level fixed by the number of processors used
  - **May be too large to keep the convergence rate**

One U-cycle for the isotropic equation on 32x32x32 and 64x64x64 grids

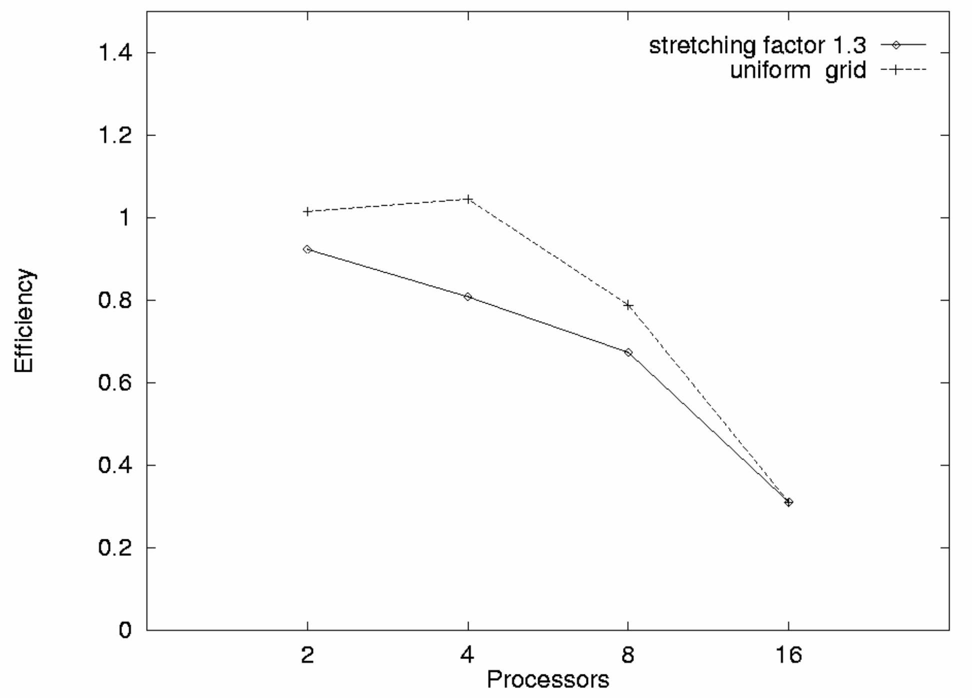
Cray T3E



- Higher efficiency than the pure V-cycle! (0.92 vs. 0.7 for 16 processors)
- **What about the convergence rate?**
  - **Realistic efficiency**

Cray T3E

Realistic efficiency using U-cycles reaching a residual norm of 10^{-12} on a 64x64x64 grid



- Lower efficiency than the pure V-cycle! (0.3 vs. 0.7 for 16 processors)

Hybrid Smoother:

1. **The number of levels can be defined and is a trade-off between algorithmic and architectural properties**

   - **The higher the number of levels the higher the convergence rate**
   - **The smaller the number of levels the higher the parallel efficiency**

2. **It uses as smoother:**

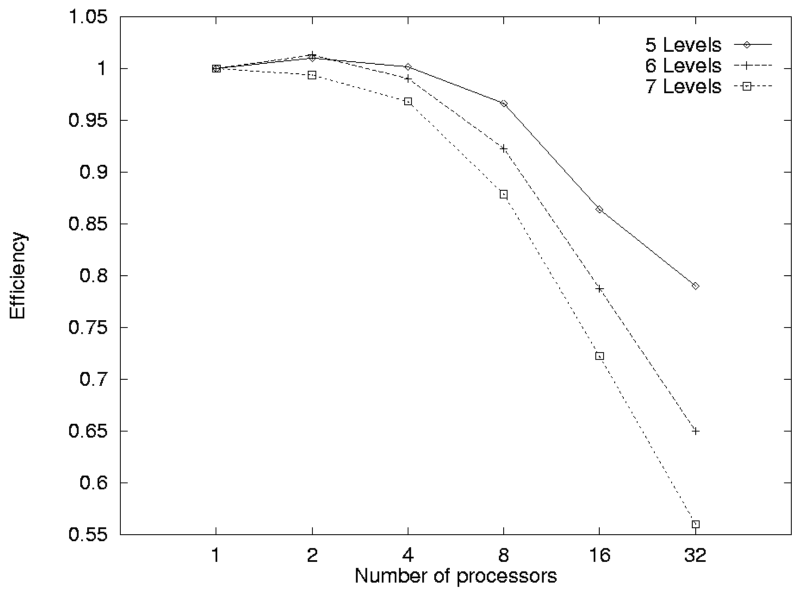
   - **Zebra smoother in and above the critical level**
   - **Damped Jacobi in and below that level**

It improves the granularity of the smoother but deteriorates its convergence rate

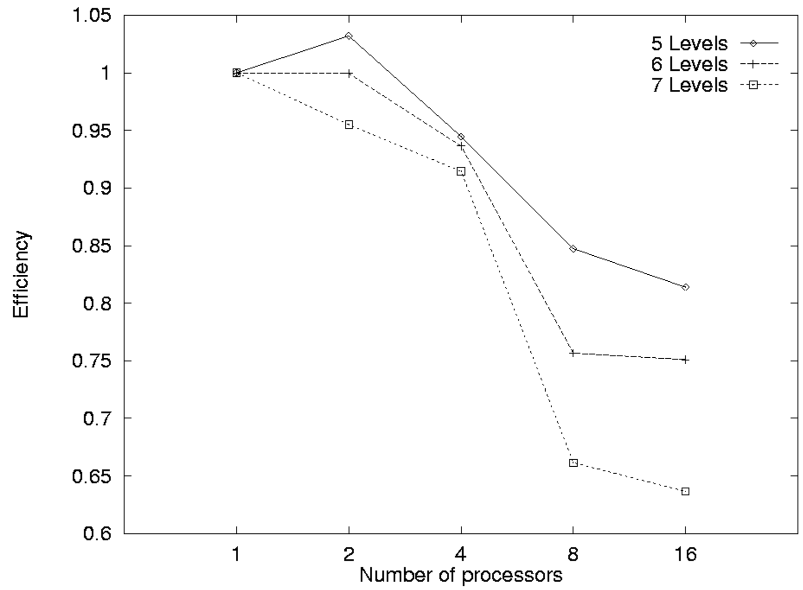
Trade-off between algorithmic and architectural properties

Realistic efficiency using Hybrid-cycles reaching a residual norm of 10^{-12} on a 64x64x64 grid

Cray T3E



Origin 2000



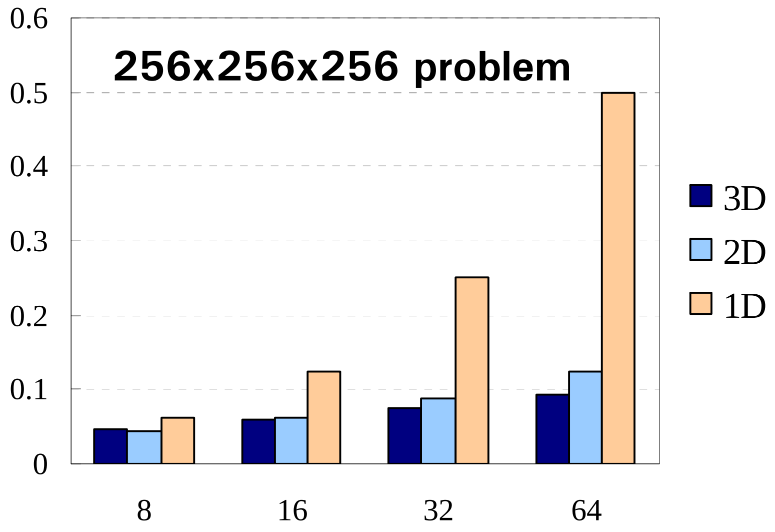
- Therefore going down to the coarsest level is not the most efficient choice
- The efficiency for the hybrid approach on 16 processors is higher (0.87 vs. 0.7 for pure V-Cycle and vs. 0.3 for the U-Cycle)

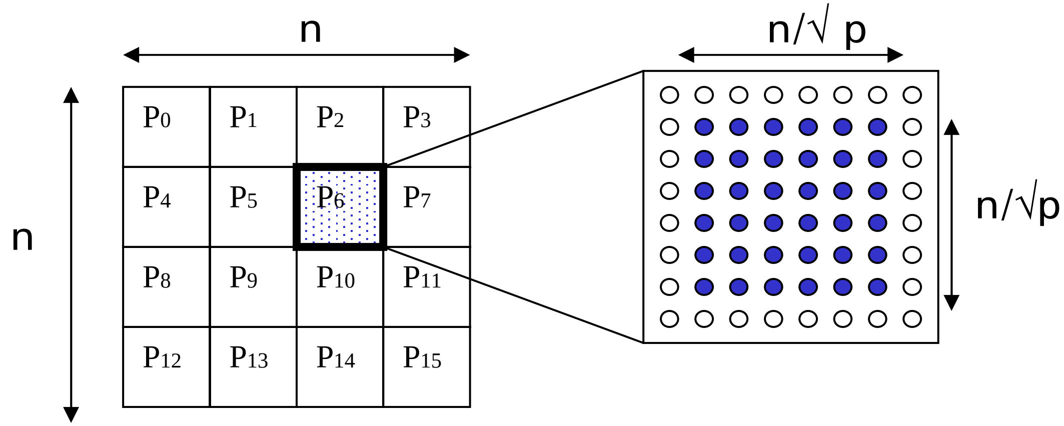
Details of this research in

D. Espadas, M Prieto, I. M. Llorente and F.Tirado, **Solution of Alternating-line Processes on Modern Parallel Computers**, In Proceedings of the 28th. International ConfICPP '99). Aizu-Wakamatsu (Japan), September 1999. Published by the IEEE Computer Society, pp. 208-215

- **Traditional wisdom says that 3-D decomposition of 3-D problems (for example a Poisson multigrid solver with point smoothing) leads to a lower inherent communication-to-computation ratio**
- **The impact becomes greater as the number of processors increases**

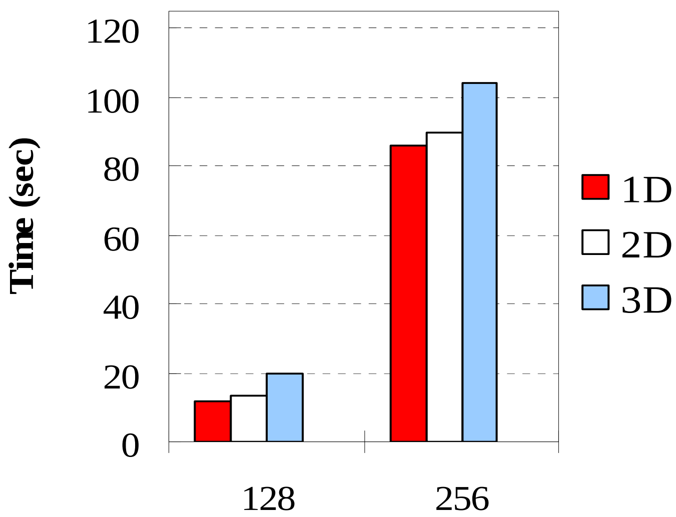
- Communications grow proportionally to the size of the boundaries
- Computations grow proportionally to the size of its entire partition
- Communication- computation ratio
  - ⇨ Perimeter-surface ratio in 2D
  - ⇨ Surface-volume ratio in 3D

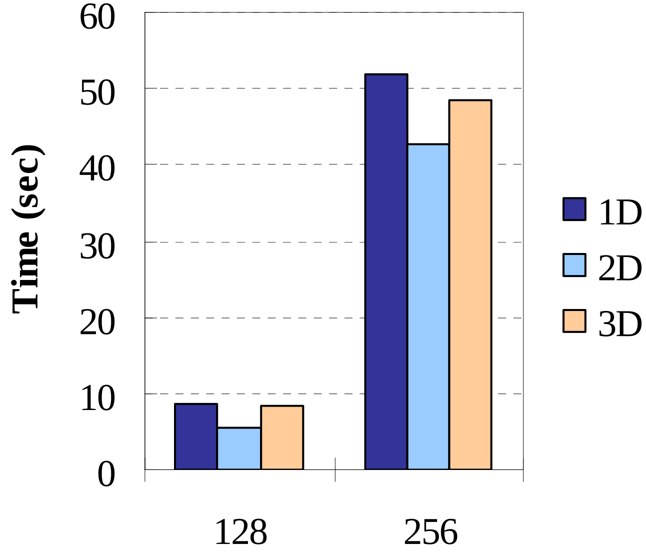




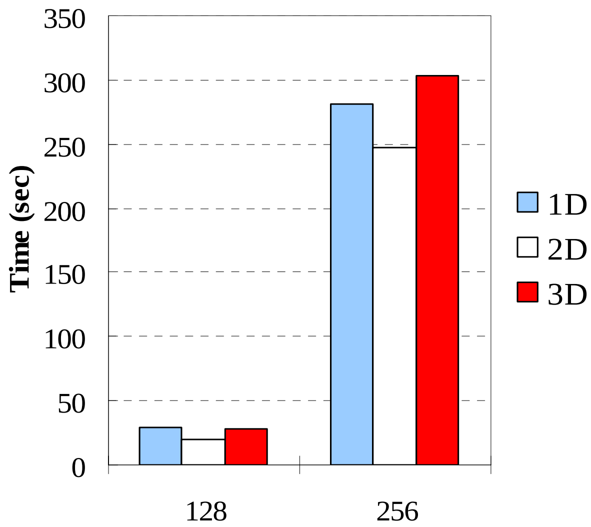
16 processors **32 processors**

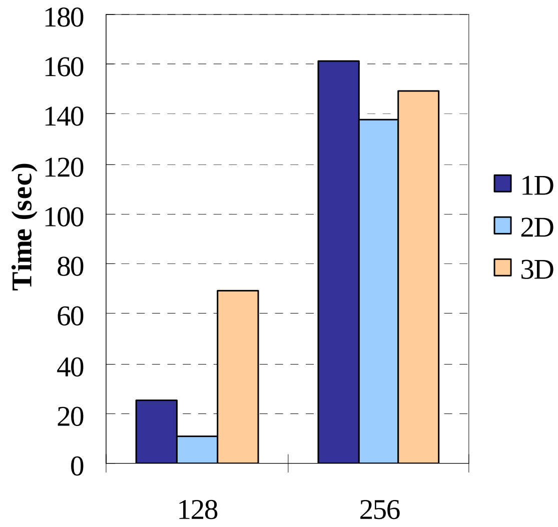
Cray T3E





Origin 2000





2-D decomposition on 32-processor SGI Origin 2000

- 22 % better than the 1-D decomposition
- 8 % better than the 3-D decomposition

- Data partitioning is a trade-off between the improvement of the message data locality and the efficient exploitation of the underlying communication system

- Using up to 32 processors, in both systems, an appropriate 2-D decomposition, where boundaries with poor spatial locality are not needed, solves that trade-off

We should also note that a lower-dimensional partitioning program

- **is easier to code**
- **allows the implementation of fast sequential algorithms in the non-partitioned directions**
- **execution time is similar to 2-D or 3-D**

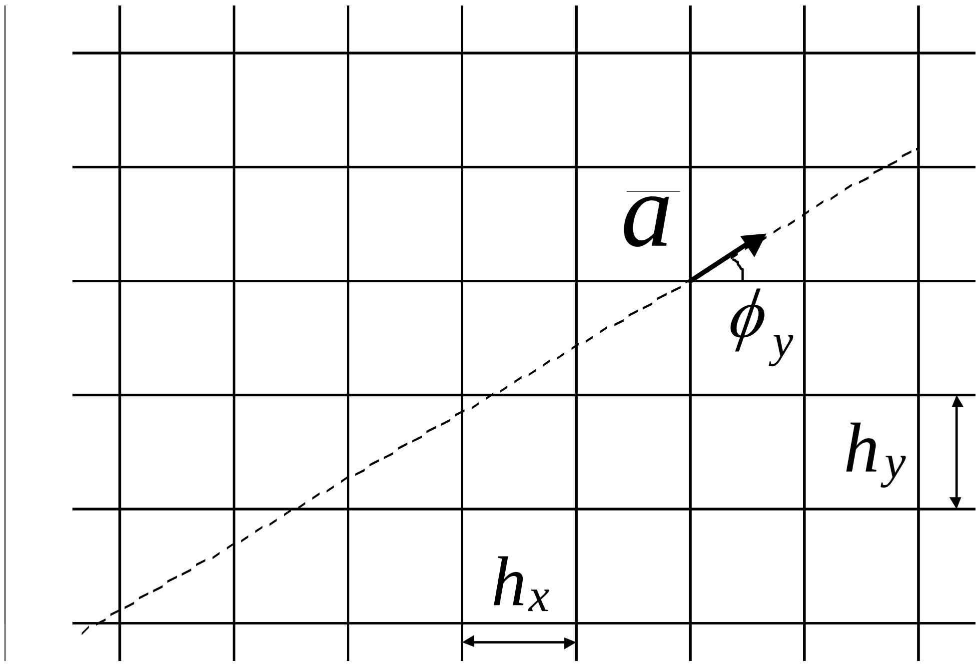
Details of this research in

M. Prieto, I. M. Llorente and F. Tirado, **Data Locality Exploitation in the Decomposition of Regular Domain Problems**, IEEE Transactions on Parallel and Distributed Systems, Vol. 11, Nº 11, 2000, pp. 1141-1150

M. Prieto, I. M. Llorente and F. Tirado, **A Revision of Regular Domain Partitioning**, SIAM News Vol 33 Number 1, January-February 2000

- The alternating-plane approach presents:
  - Higher convergence rates that improves with the anisotropy strength
  - Lower memory requirements
  - Higher execution time per cycle
- In summary, better convergence per work unit
- However, its parallel implementation is not efficient since it requires the solution of distributed 2-D systems (difficult and poor efficient implementation)
- The 1-D parallel version of the semicoarsening approach is more efficient using a tradeoff between the V- and U-cycles (hybrid smoother)

¿Can we obtain an optimal solver for the convection operator?



Introduction

- Multigrid for convection problems
- Model problem

Our approach

- Narrow discretization
- Cross-characteristic interaction

Results

- Convergence rate
- Parallel efficiency

Conclusions of the second part

Multigrid is highly efficient to solve elliptic operators

However, it fails to solve nonelliptic operators

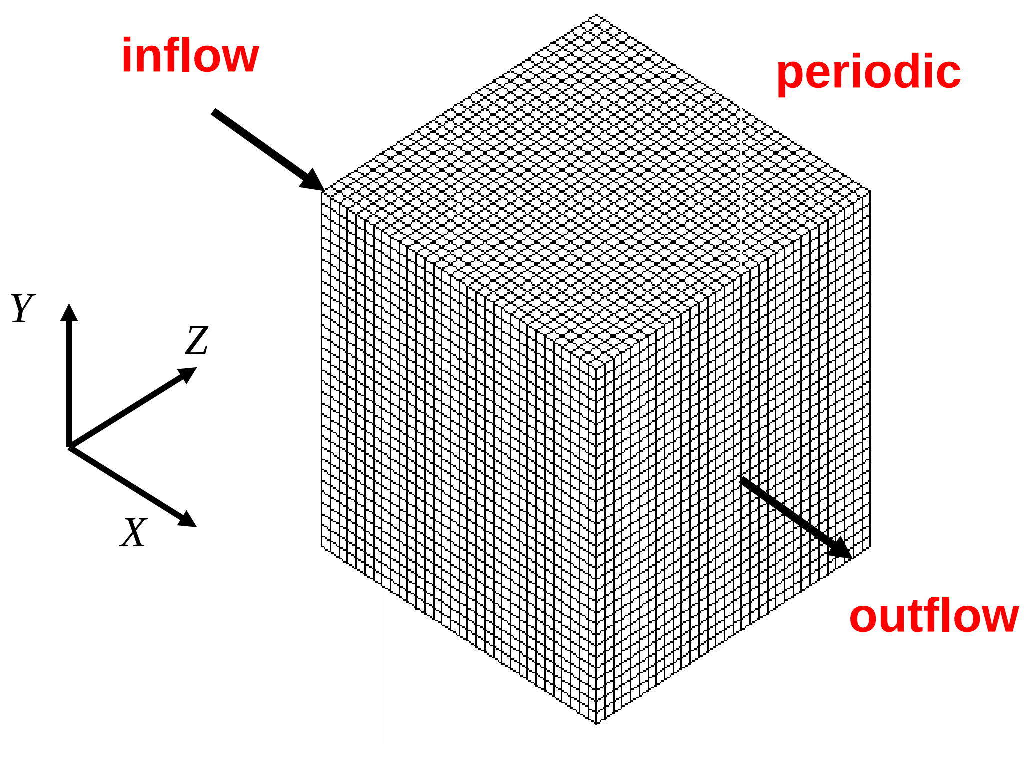
In many cases, the nonelliptic part is represented by the **convection operator**
For example, Navier-Stokes solved by multigrid based on a distributive smoother

$$a\frac{\partial U}{\partial x}+b\frac{\partial U}{\partial y}+c\frac{\partial U}{\partial z}=F(x,y,z)$$

Downstream marching is the more efficient sequential solver for this operator
- ☺ Solve linear upwind operators in one sweep
- ☺ Solve nonlinear operators in few sweeps

However:
- ☹ A defect-correction scheme must be applied if the discretization is not fully upwind
- ☹ Very low parallel efficiency due to the sequential marching

$$a\frac{\partial U}{\partial x}+b\frac{\partial U}{\partial y}+c\frac{\partial U}{\partial z}=F(x,y,z)$$



[Disk99]

NARROW DISCRETIZATION

+

SEMICOARSENING

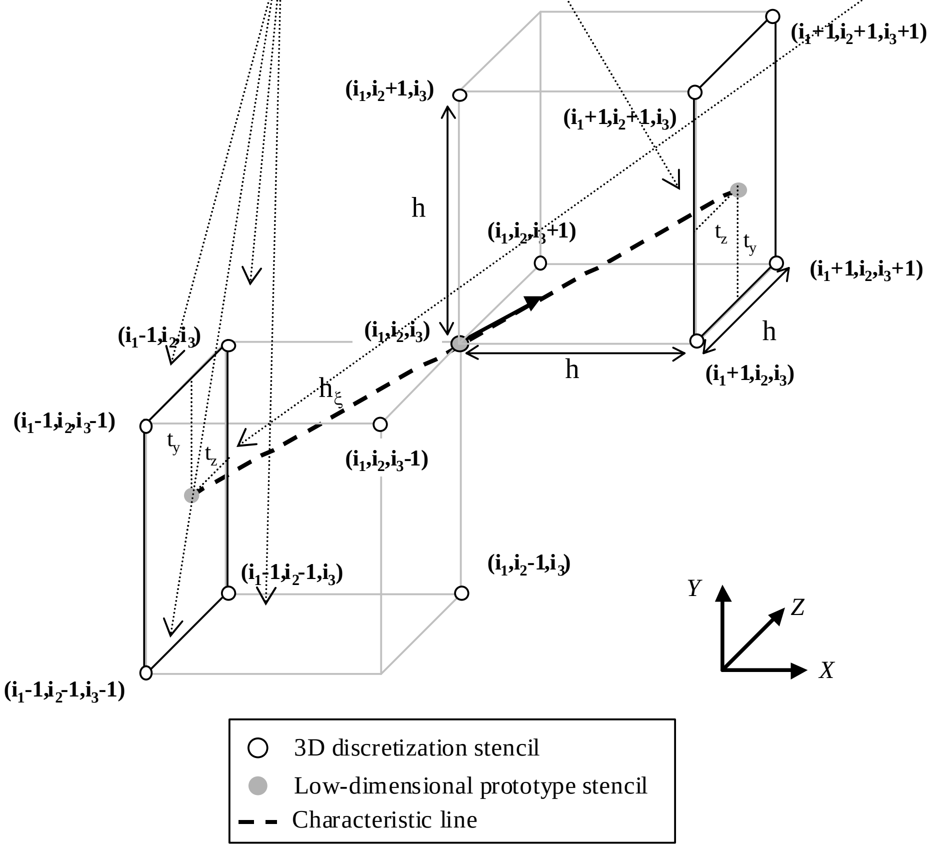
+

CORRECTION OF OPERATORS TO MAINTAIN THE SAME CROSS-CHARACTERISTIC INTERACTION IN ALL GRIDS

+

FOUR-COLOR PLANE IMPLICIT SMOOTHER

- The discretization follows the **characteristic line** of the operator
- The full-dimensional operator is obtained by replacing **values at ghost points** by weighted averages at adjacent **genuine grid points**



¿What is the cross-characteristic interaction (CCI)?

- The CCI induced by a discrete operator is estimated by the coefficients of the lowest pure cross-characteristic derivatives appearing in the first differential approximation
- In our simpler case, CCI appears only because on interpolation in the y-z plane

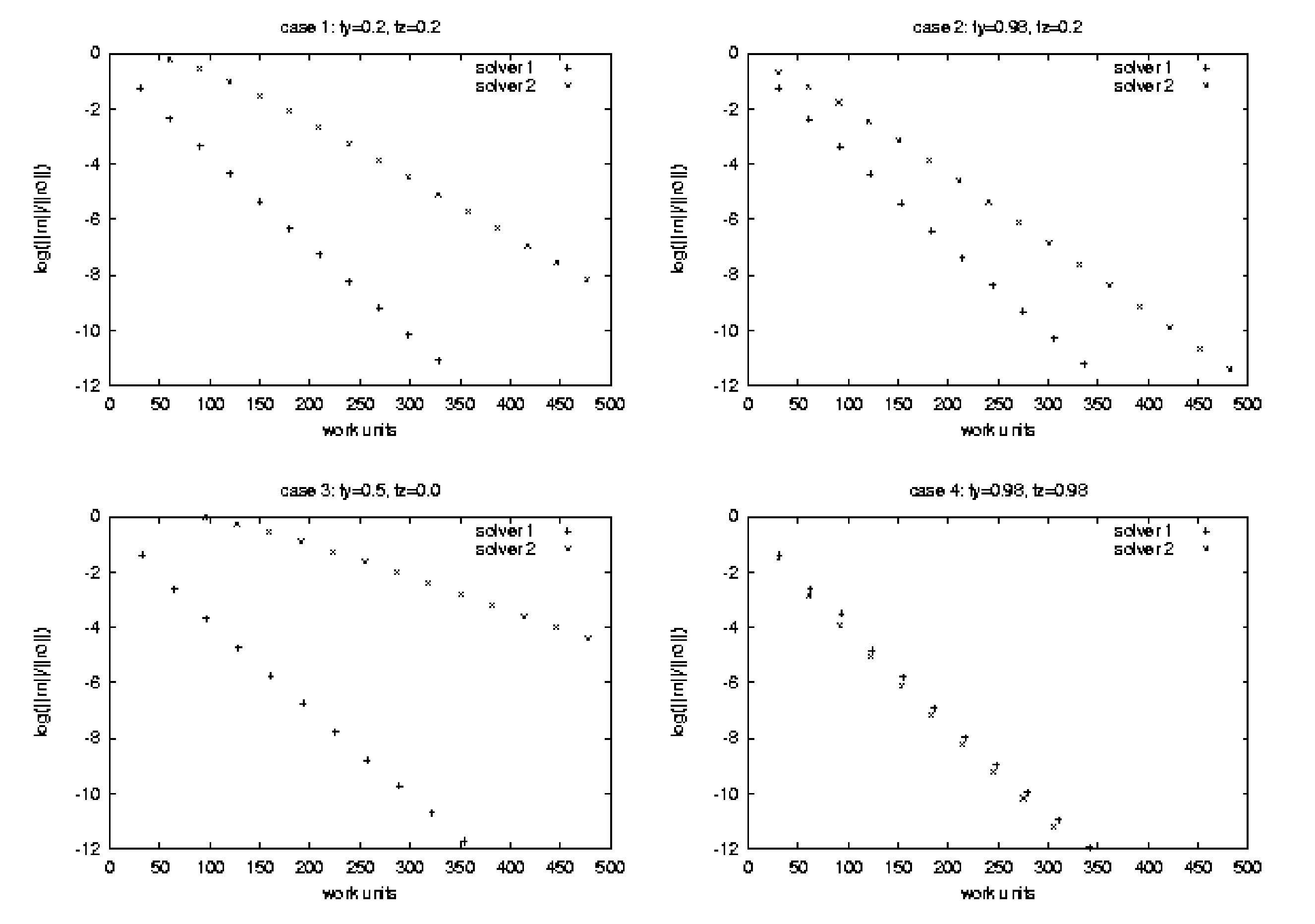
Main difficulty in constructing an efficient solver for nonelliptic operators:

- Poor coarse-grid approximation to fine-grid characteristic error components
- The coarse-grid CCI is lower than required in a narrow discretization on a semicoarsened grid

Solution:

- We supply additional terms (*explicit CCI*) in coarse-grid discretizations so the total coarse-grid CCI would be the same as on the fine grid

Residual versus work units



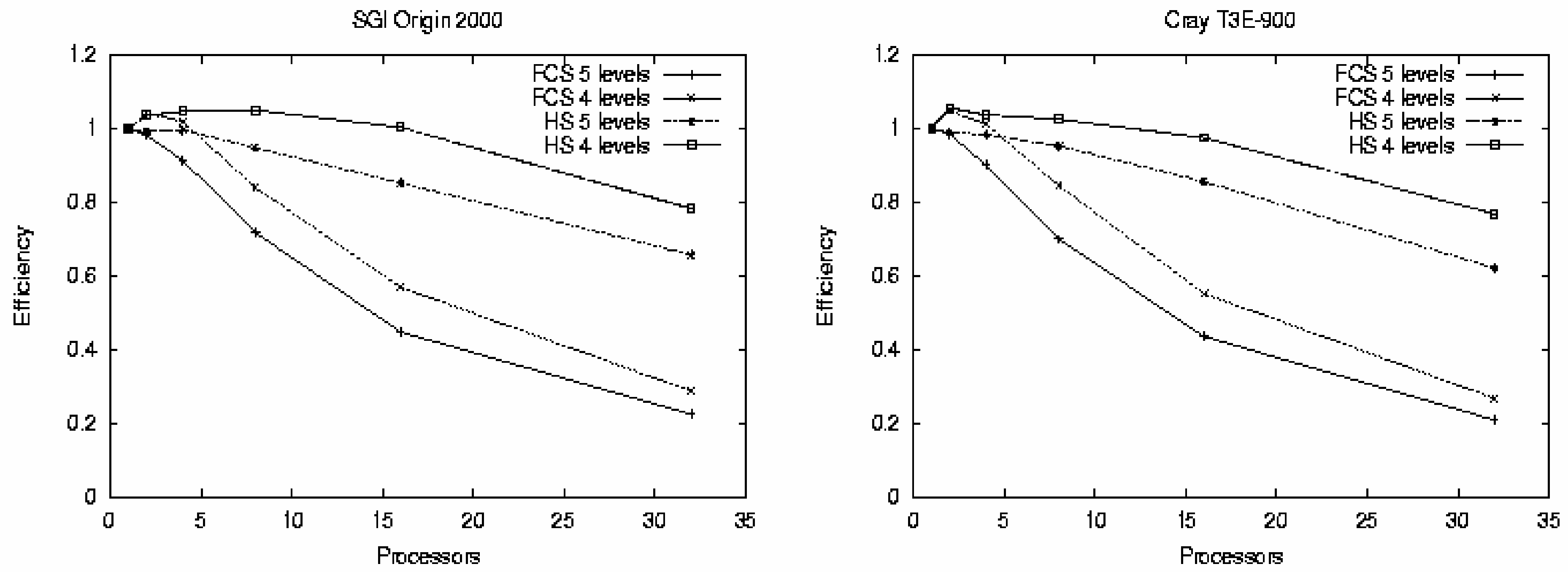
$$t_y = \frac{b}{a}$$

$$t_z = \frac{c}{a}$$

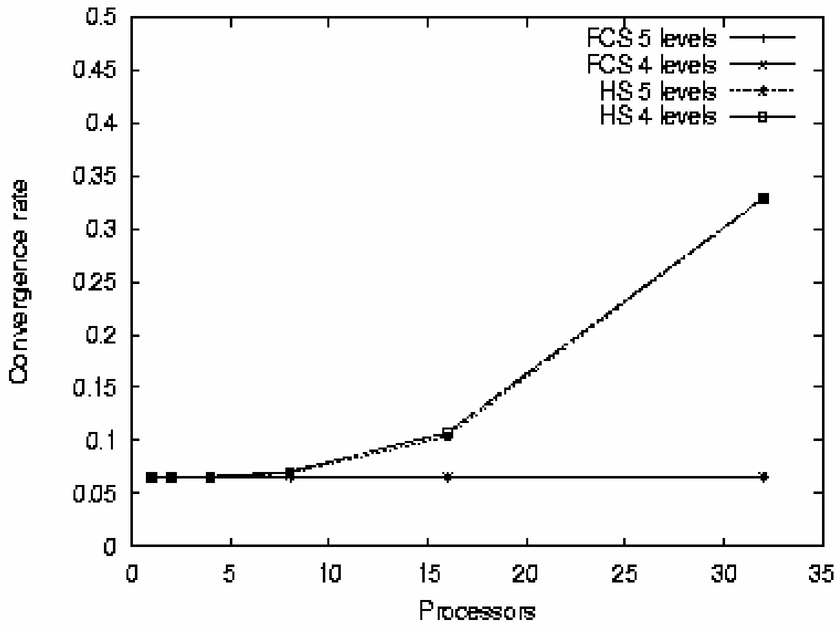
Solver 1: With explicit CCI terms
Solver 2: Without explicit CCI terms

Grid-independent convergence rate 0.09 for any angles of nonalinment

Efficiency of one Hybrid-cycle



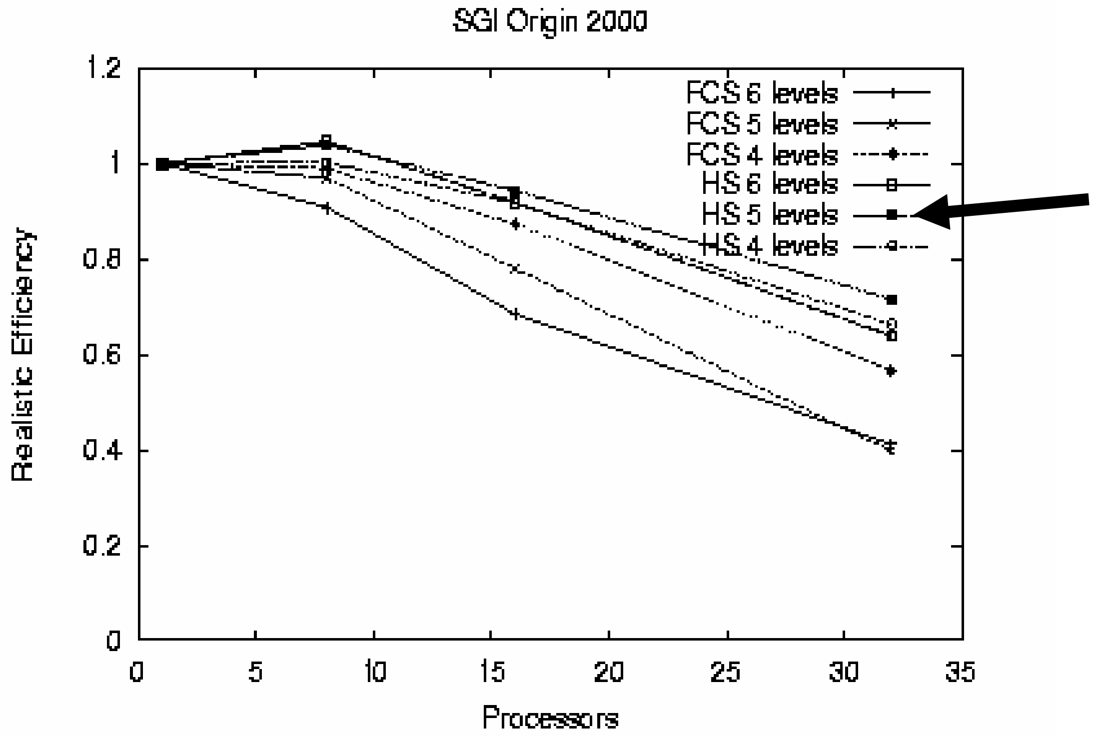
- **What about the convergence rate?**
  - **Realistic efficiency**



It deteriorates the convergence properties

The *Realistic Parallel Efficiency* considers the execution time to reach the final solution (to reach a certain residual norm)

Realistic efficiency using Hybrid-cycles reaching a residual norm of 10^{-12}



Again the best choice is a **trade-off** between the **parallel and numerical properties**

Details of this research in

I. M. Llorente, M. Prieto-Matias and B. Diskin, **An Efficient Parallel Multigrid Solver for 3-D Convection Dominated Problems**, in press, Parallel Computing

- We have proposed a multigrid algorithm to solve in a parallel setting a convection operator that is sequential in nature
- Such operator appears in many practical problems in CFD
  - For example, distributive smoothers
- We have studied different alternatives to implement the solver on a parallel computer
- The 1-D decomposition with a hybrid smoother appears to be a tradeoff between parallel and convergence properties
- Satisfactory efficiencies (higher than 0.8) are obtained up to 32 processors

¿Can previous conclusions be extended for the incompressible Navier-Stokes Equations?

Introduction

- Description of the problem
- Multigrid approach
- The coupled smoother
- The plane smoother

Boundary layer of a flat plate at yaw

- Domain
- Boundary conditions
- Non-zero yaw angle

Results

- Validation with Blasius theory
- Convergence rate

Parallel implementation

- 1-D decomposition
- 2-D decomposition
- Parallel architecture
- Analysis of the interconnection alternatives
- Analysis of the execution node alternatives
- Scalability

Relation to the MG/NAS parallel benchmark

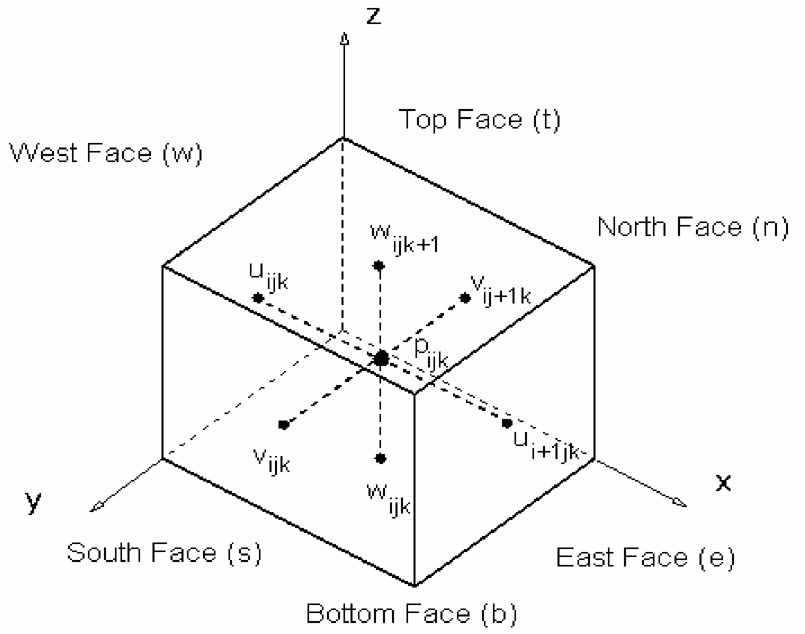
- Description of MG/NAS
- Our code as a benchmark
- Comparation to our code

Conclusions of the third part

- Dimensionless steady-state incompressible Navier-Stokes equations:

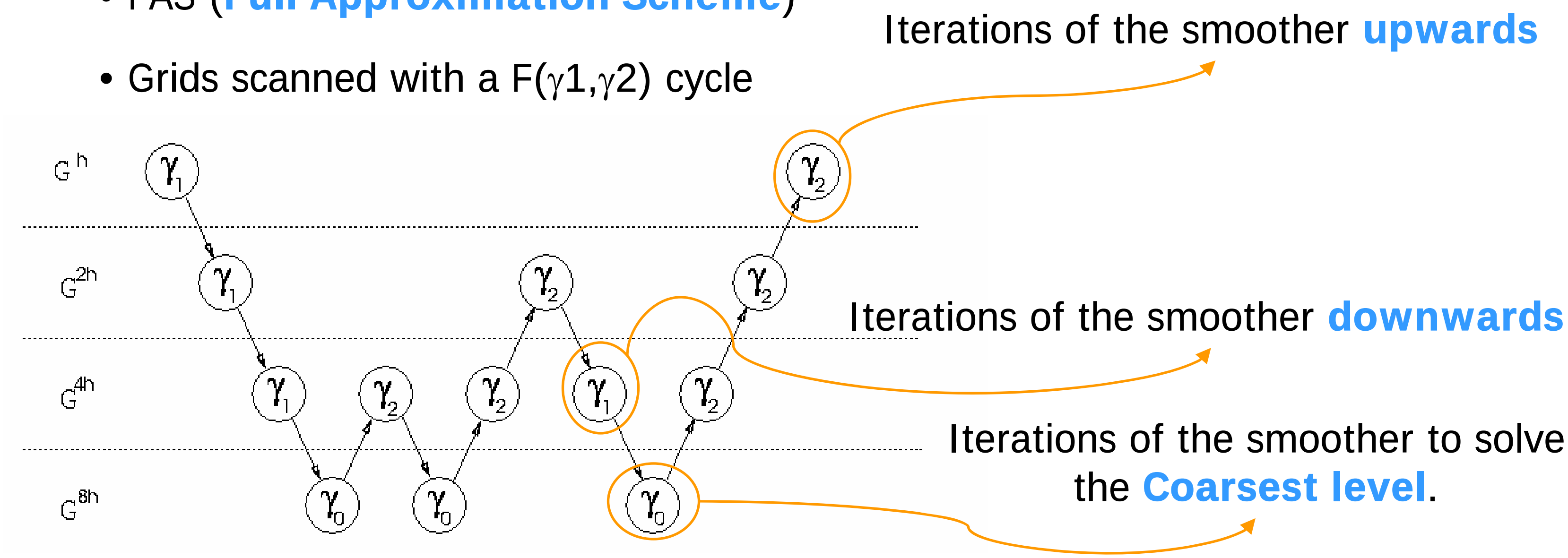
$$(u \cdot \nabla) u = -\nabla p + \frac{1}{Re} \Delta u$$
$$\nabla \cdot u = 0$$

- Discrete system obtained using a **finite volume** technique
- **Discretization** over an orthogonal structured grid
- **Staggered arrangement** of unknowns
- The second-order operator is obtained using a **QUICK** scheme [HaHG92] solved via **defect-correction** inside the multigrid cycle [OGWW98]

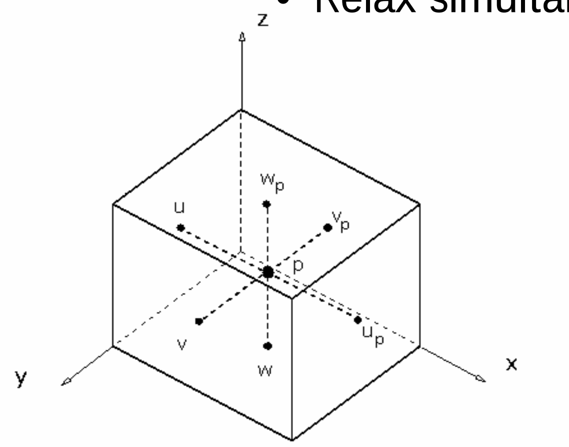


- **Multigrid strategy : Full Multi-Grid** (FMG).
  - Coarsest → Finest.
  - Reduce the algebraic errors below the discretization error in one FMG cycle.

- **Multigrid Cycle:** Solve each level of the FMG algorithm
  - FAS (**Full Approximation Scheme**)
  - Grids scanned with a F(γ1,γ2) cycle



- We have chosen a coupled smoother [Vank86] instead of the distributive alternative [BrYa92]
  - ✓ **All variables** involved in each control-volume are **updated simultaneously**
    - Relax simultaneously the momentum and continuity equations

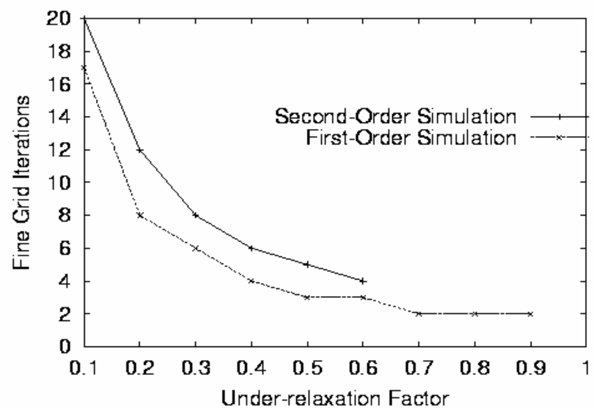


$$\begin{pmatrix} L_{ijk}^{u_{west}} & 0 & 0 & 0 & 0 & 0 & L_{p_i}^{west} \\ 0 & L_{ijk}^{u_{east}} & 0 & 0 & 0 & 0 & L_{p_{i-1}}^{east} \\ 0 & 0 & L_{ijk}^{v_{south}} & 0 & 0 & 0 & L_{p_j}^{south} \\ 0 & 0 & 0 & L_{ijk}^{v_{north}} & 0 & 0 & L_{p_{j-1}}^{north} \\ 0 & 0 & 0 & 0 & L_{ijk}^{w_{bottom}} & 0 & L_{p_k}^{bottom} \\ 0 & 0 & 0 & 0 & 0 & L_{ijk}^{w_{top}} & L_{p_{k-1}}^{top} \\ -\Delta y\Delta z & \Delta y\Delta z & -\Delta x\Delta z & \Delta x\Delta z & -\Delta x\Delta y & \Delta x\Delta y & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta u_{ijk} \\ \Delta u_{i+1jk} \\ \Delta v_{ijk} \\ \Delta v_{ij+1k} \\ \Delta w_{ijk} \\ \Delta w_{ijk+1} \\ \Delta p_{ijk} \end{pmatrix} = \begin{pmatrix} r_{ijk}^{u} \\ r_{i+1jk}^{u} \\ r_{ijk}^{v} \\ r_{ij+1k}^{v} \\ r_{ijk}^{w} \\ r_{ijk+1}^{w} \\ r_{ijk}^{m} \end{pmatrix}$$

- The solution is updated via **under-relaxation**

$$u^{n+1} = u^n + w_u \cdot \Delta u$$

$$p^{n+1} = p^n + w_p \cdot \Delta p$$



- We have compared different robust smoothers: alternating-plane vs. semicoarsening [MoLS01] and the combination of semicoarsening and plane implicit presents much better properties

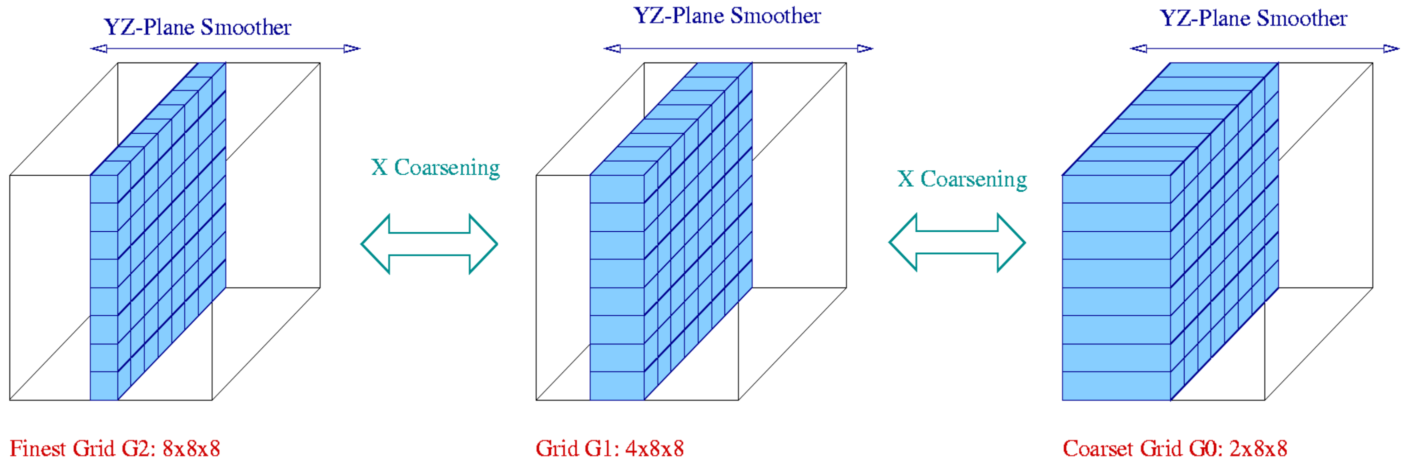
- All the velocity components and pressures contained within the plane are updated

- **Example YZ-Plane:**

$$X_k^T = (\vec{u}, \vec{v}, \vec{w}, \vec{w}^{+}, \vec{p})$$

$$\vec{u} = u_{ijk} \quad \vec{v} = v_{ijk} \quad \vec{w} = w_{ijk} \quad \vec{w}^{+} = w_{ijk+1} \quad \vec{p} = p_{ijk}$$

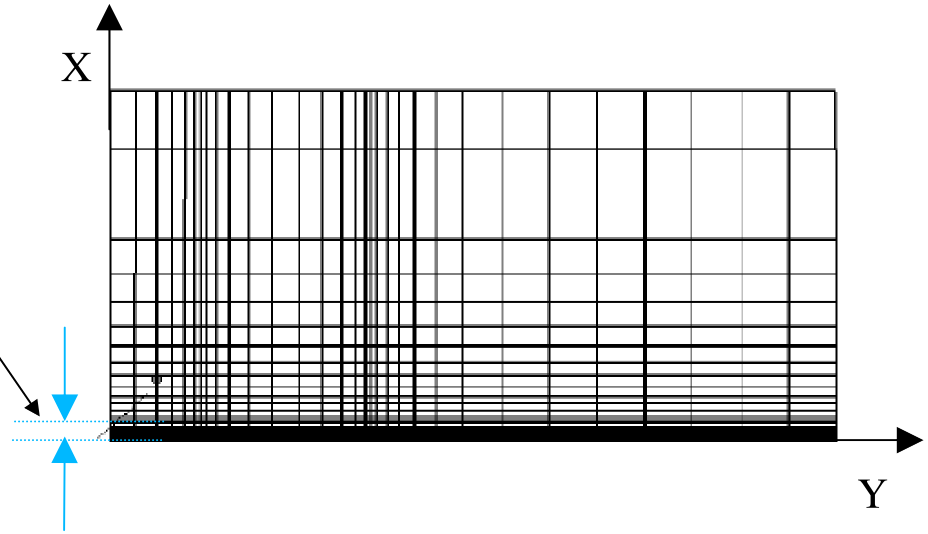
$$\forall j,k \in [0,n] \quad i = const$$



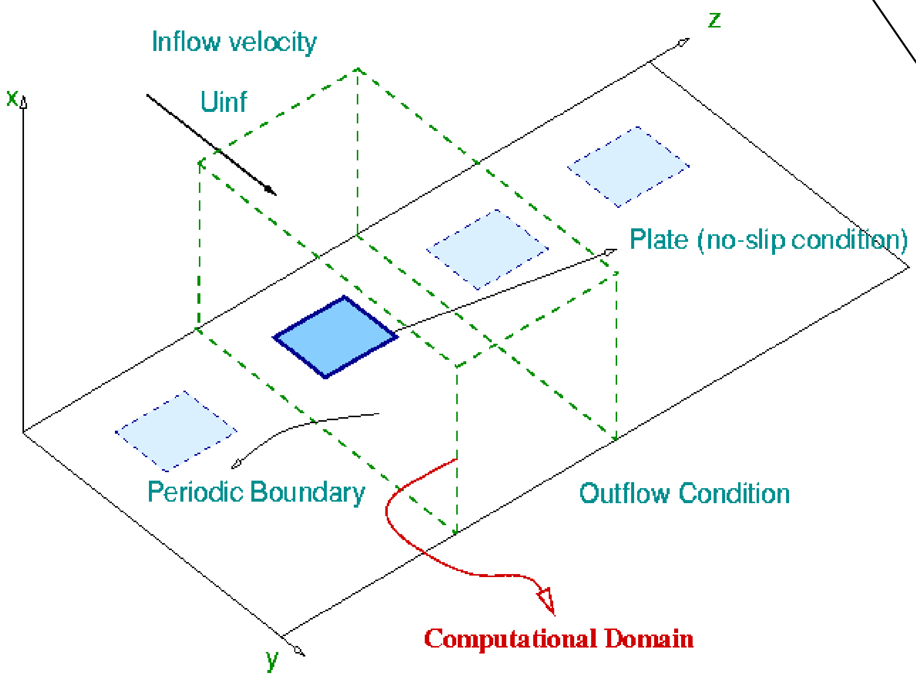
- An 2-D direct **exact solver** is not needed for the planes
- The 2-D system is approximately solved with one cycle of a **robust 2-D multigrid algorithm**
    - The 2-D algorithm is the combination of line smoothing with semicoarsening

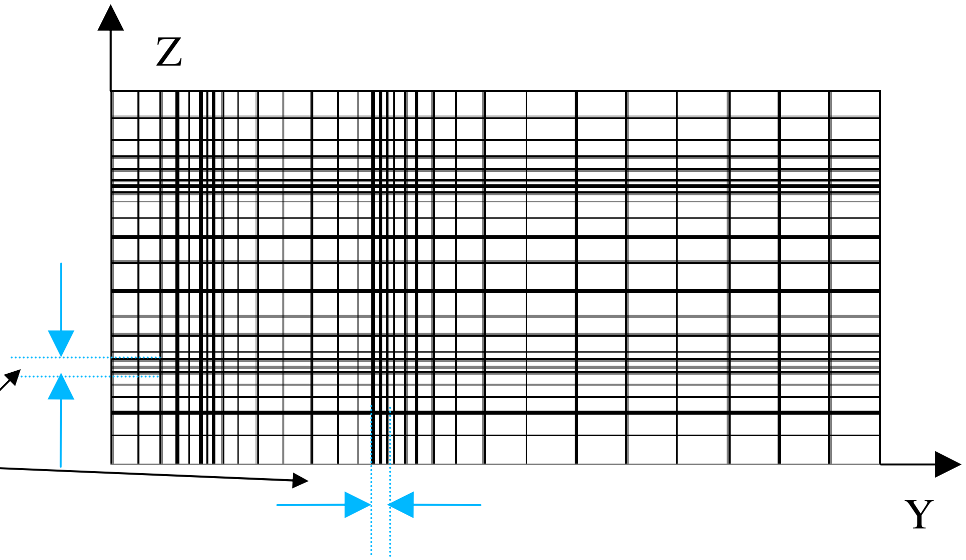
- Domain

Geometrically stretched in x direction to capture the boundary layer



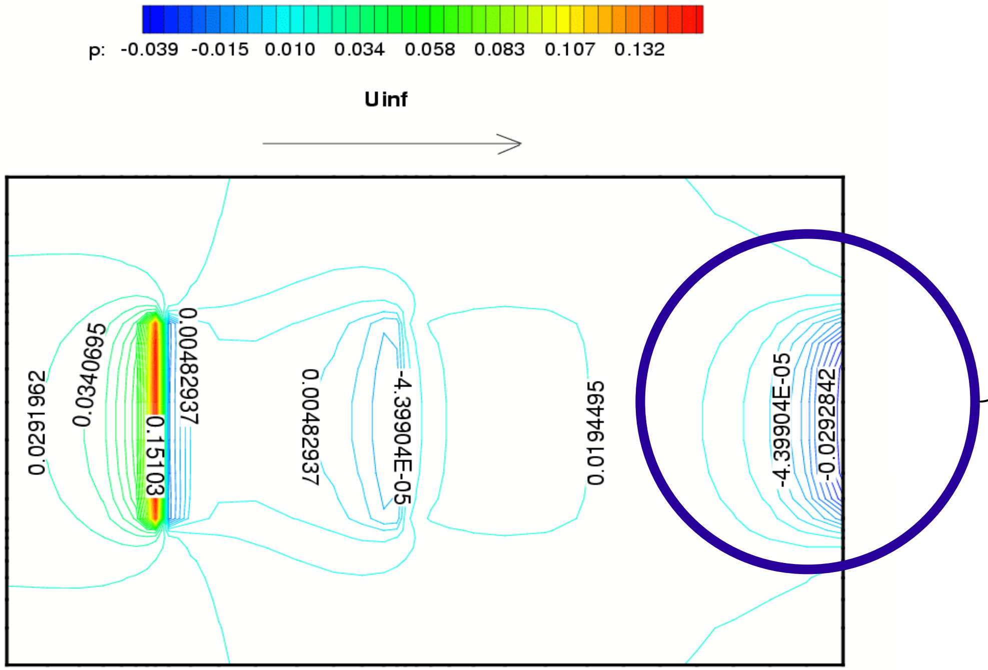
cascade of square plates

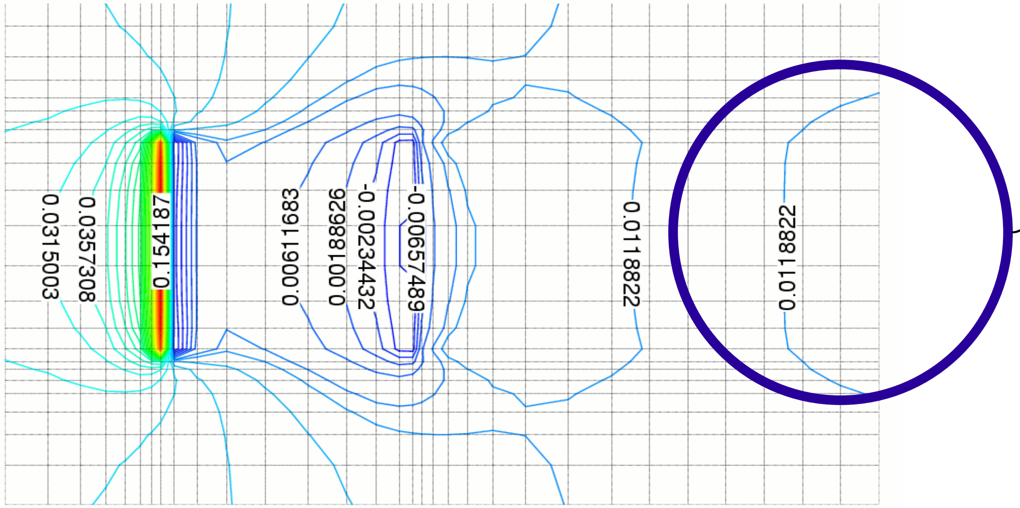




Geometrically stretched near the plate edges due to high residuals and errors

Boundary Layer of a Flat Plate at Yaw
• Boundary Conditions



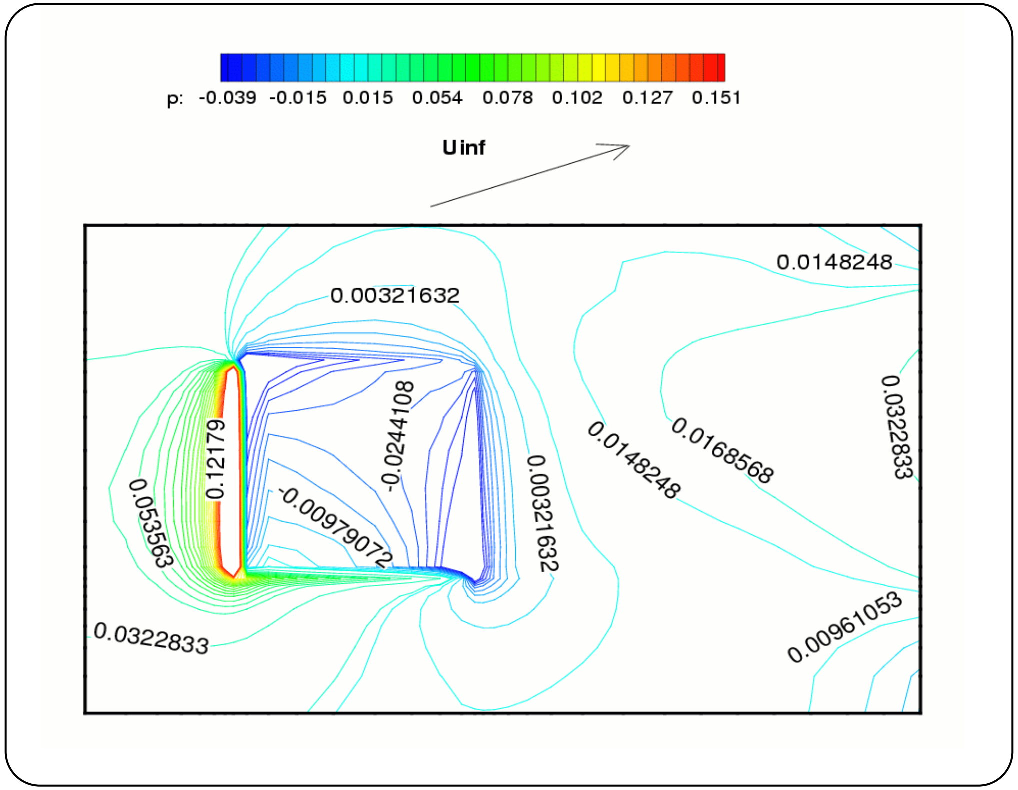


- Region of a favorable pressure gradient that accelerates the velocity at the outlet in order to match the wrong outflow condition

- The outflow boundary condition is derived using Goldstein's calculations of the velocity distribution in the wake of a finite 2-D flat plate [Gold33]
- The low pressure zone in the wake of the plate disappears

Boundary Layer of a Flat Plate at Yaw

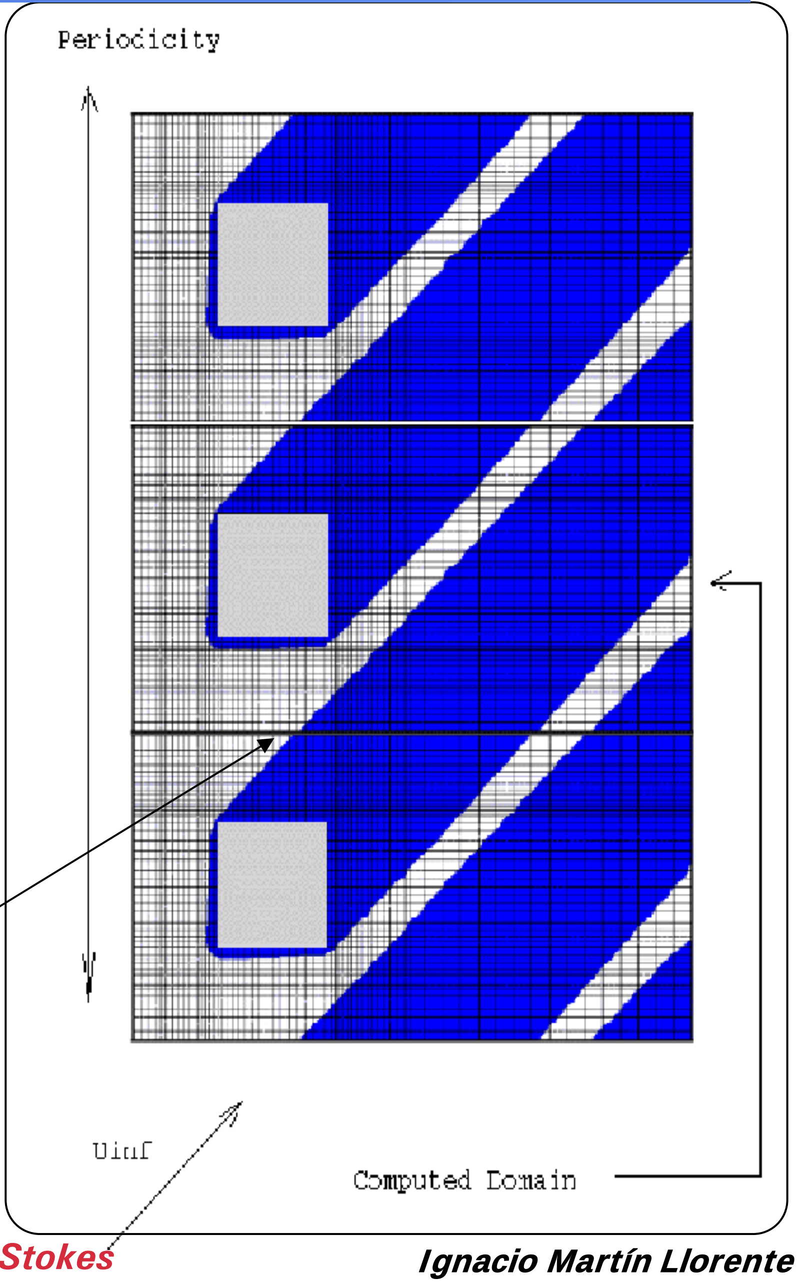
• Non-zero Yaw Angle



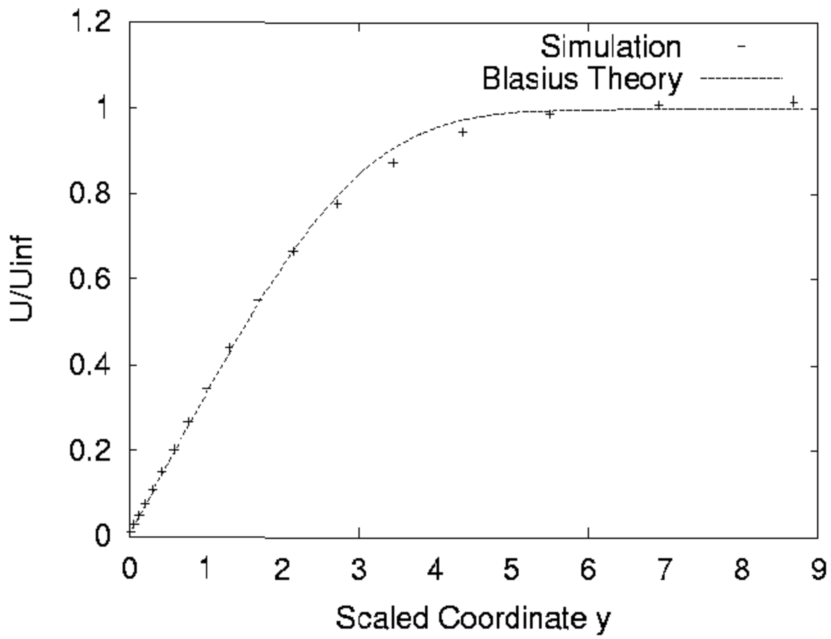
The two problems associated with this simulation are boundary layers and entering flows with non-aligned characteristics

Periodicity in the velocity field

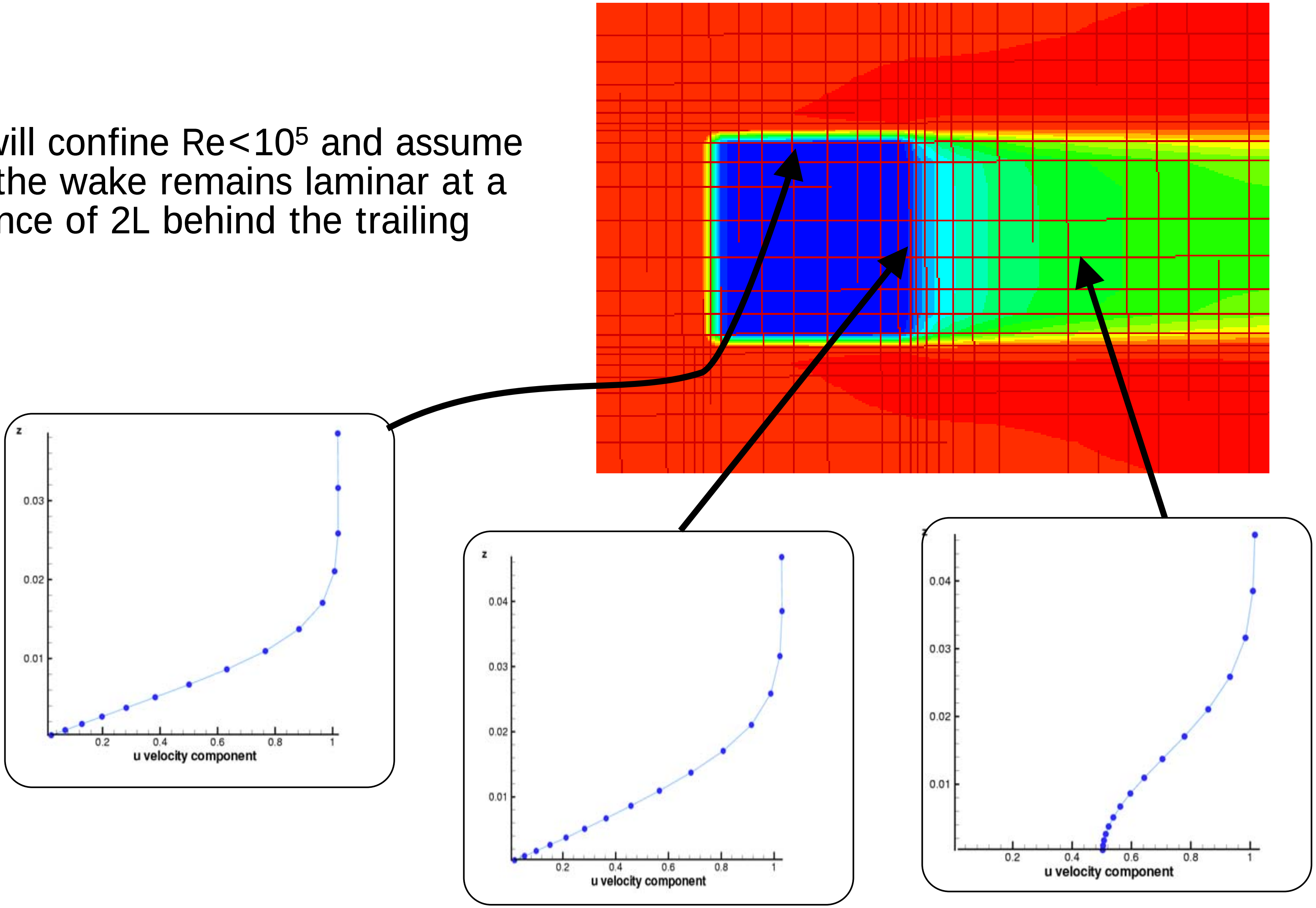
Boundary conditions are obtained by rotating Goldstein's calculations

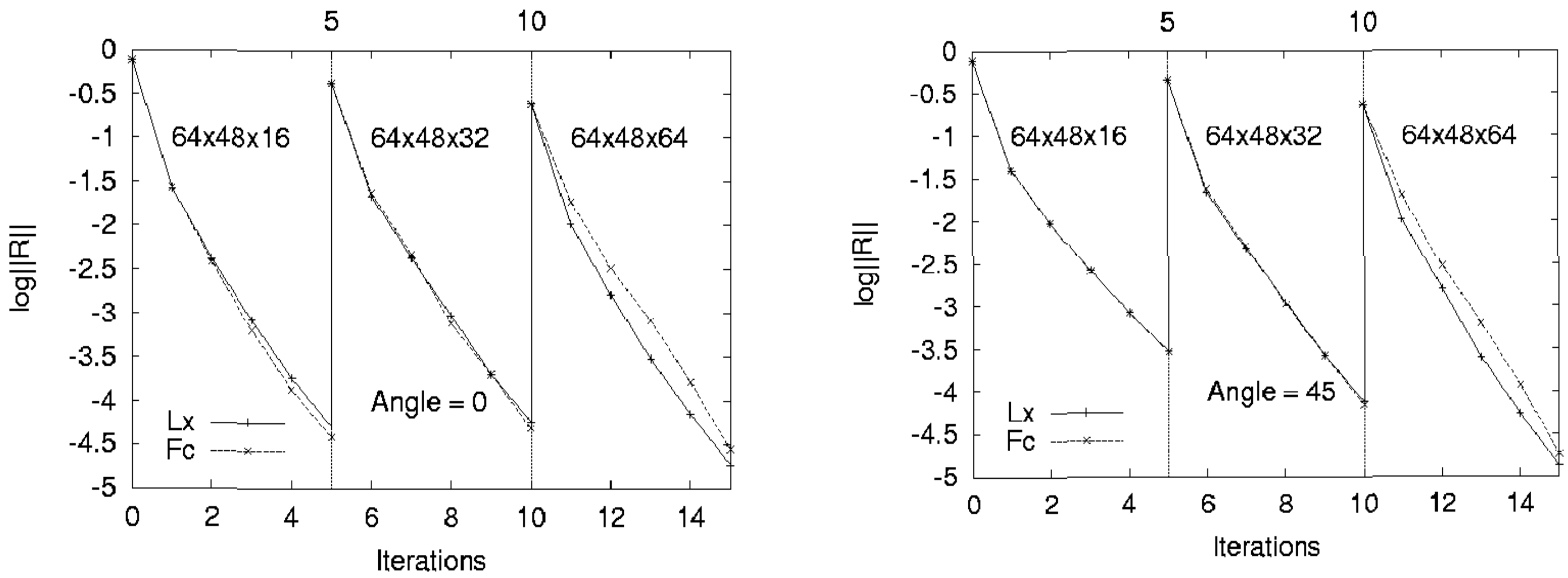


• Validation with Blasius Theory

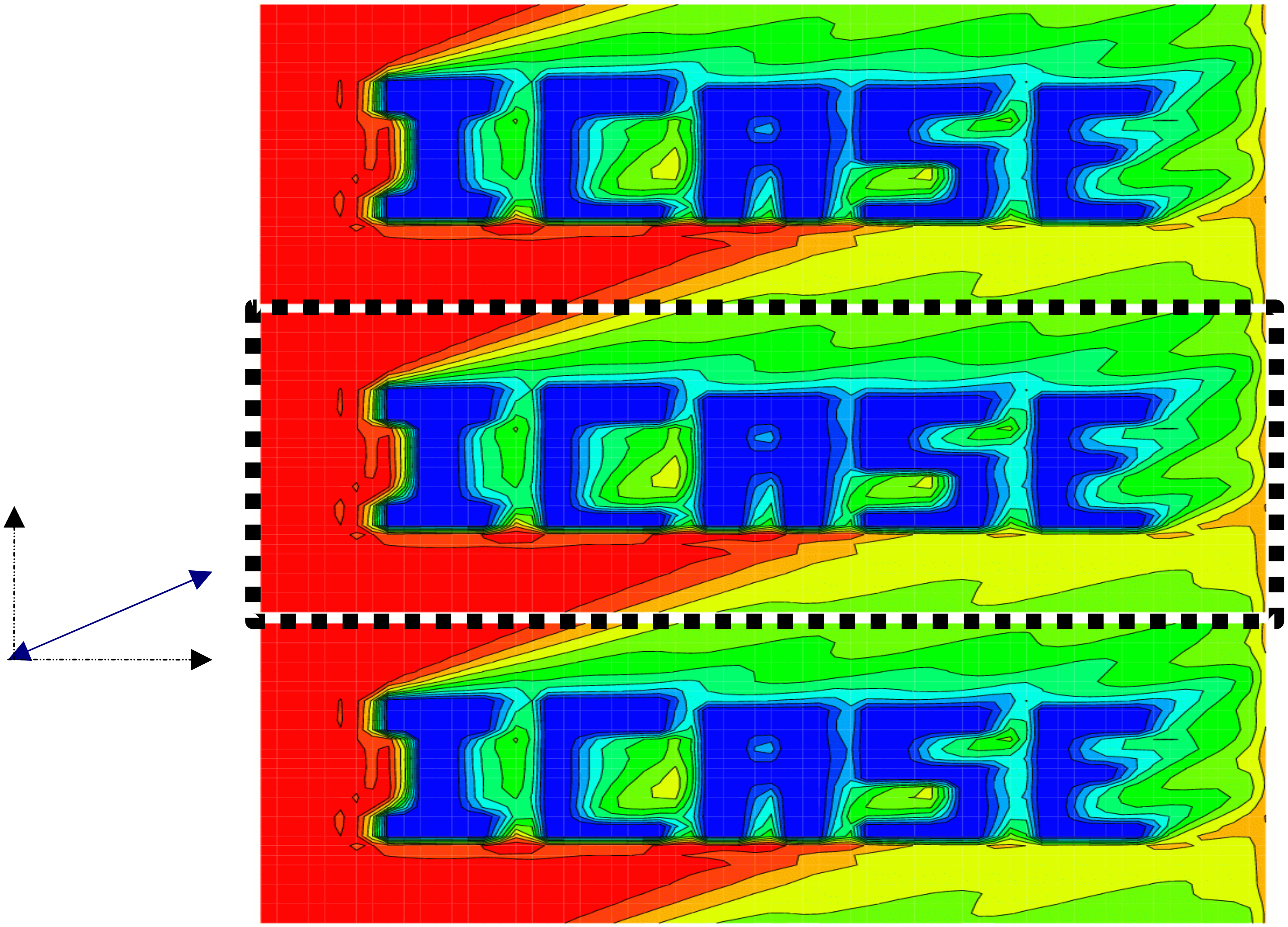


• We will confine $Re<10^5$ and assume that the wake remains laminar at a distance of 2L behind the trailing edge



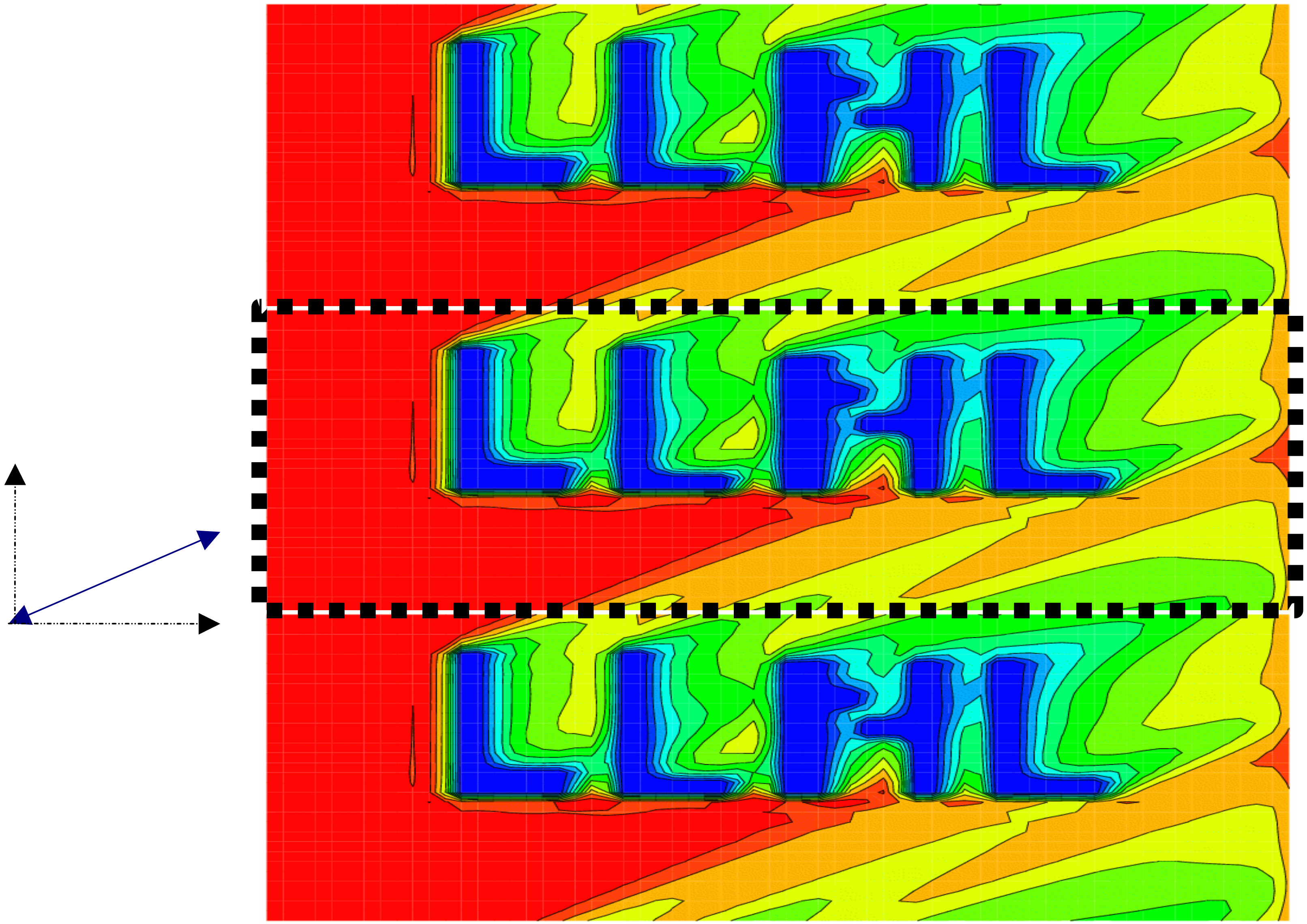


- **The convergence attained within the first five cycles is below 0.1 for both smoothers and yaw angles for Reynolds numbers bellow 10000**


ICASE
ICASE
ICASE

Results

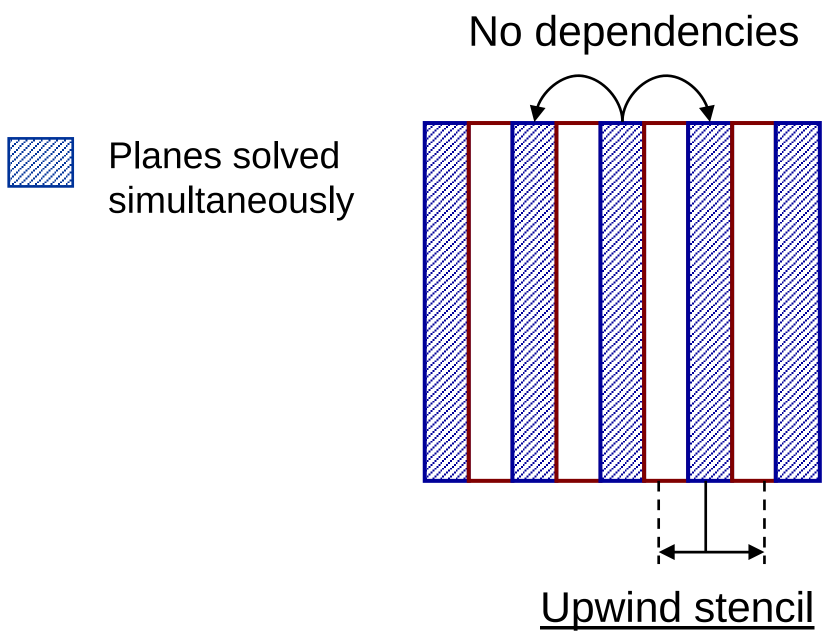
• Convergence Rate



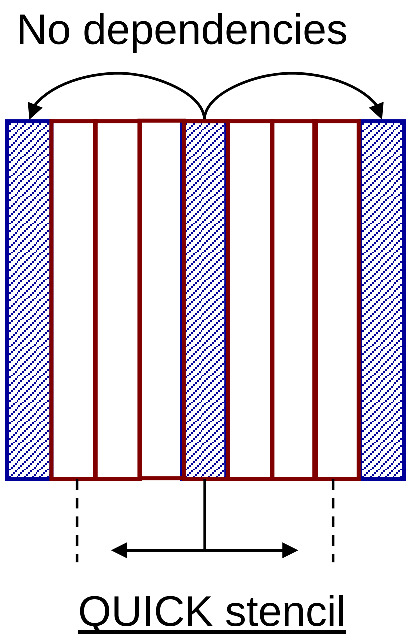
1-D decomposition on the semicoarsened direction

•Traditional **Zebra order** of planes is not possible (correction of the convective terms)

• Planes scanned in a **4-c** fashion

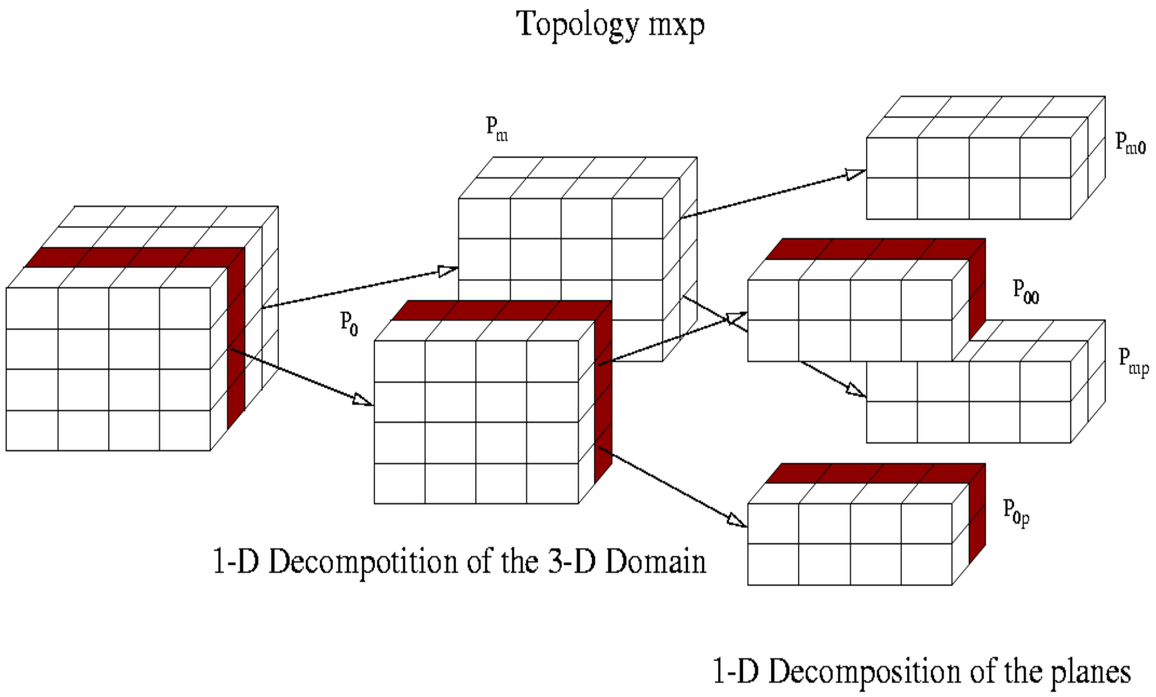


Zebra Ordering of Planes



Tri-Plane Ordering of Planes

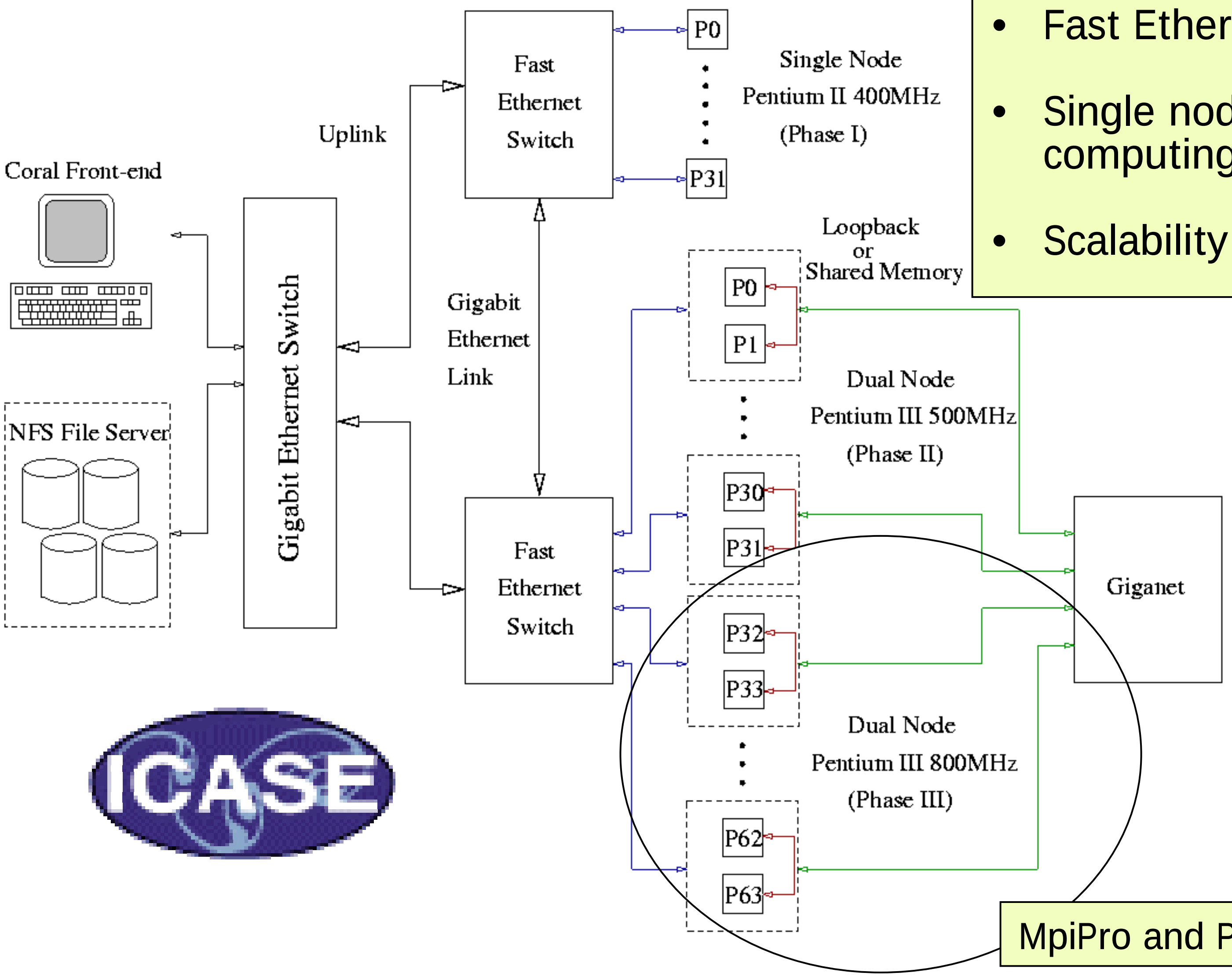
Parallel Implementation
• 2-D Decomposition



- U-cycle
- The planes are solved with Semi-coarsening + Line Smoothing (the line solvers are not distributed)

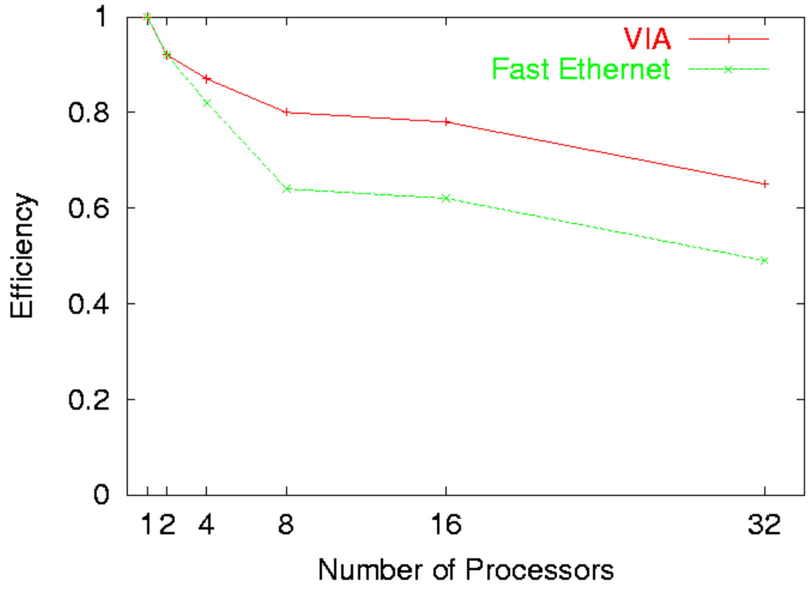
Objectives of the study:

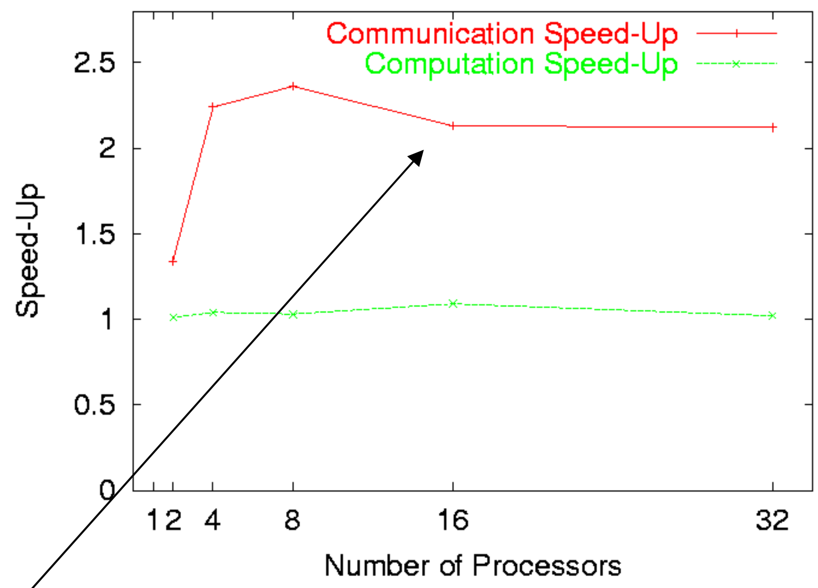
- Fast Ethernet vs. Giganet-VIA
- Single node vs. dual node computing
- Scalability



MpiPro and Porland Group compiler

Simulations using Dual-Nodes and a 32x128x128 grid

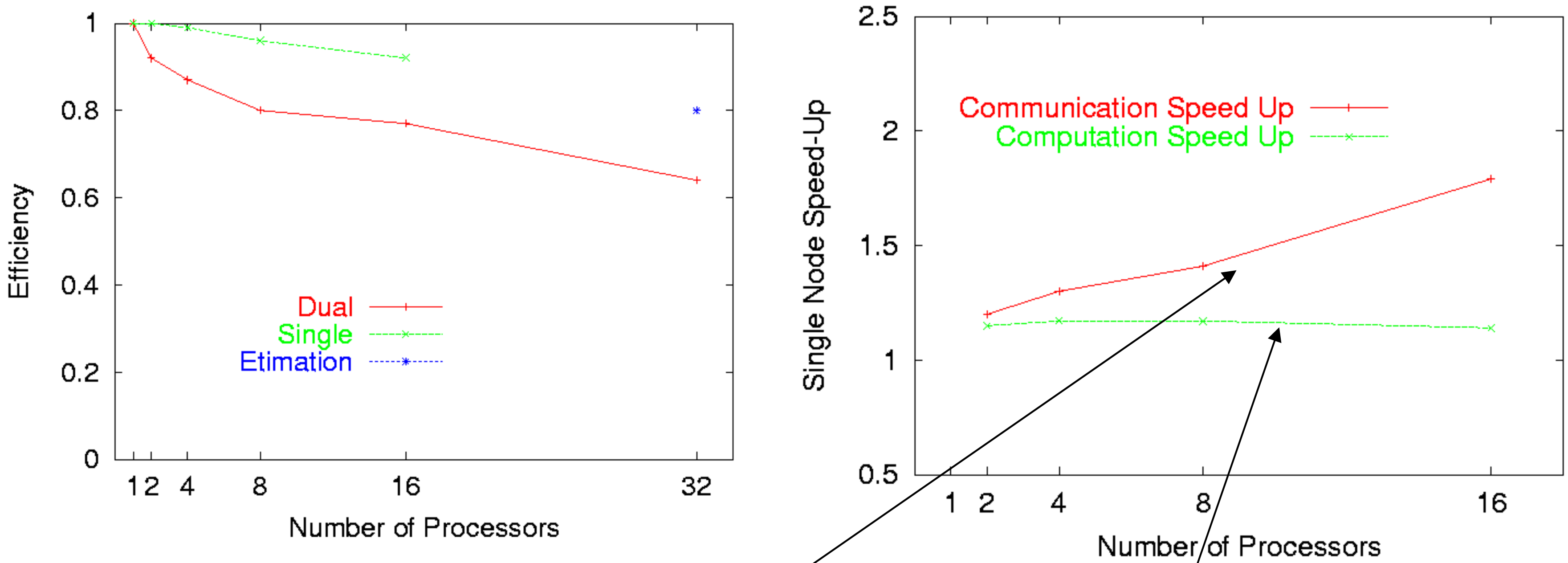




Drop in Efficiency due to the higher interconnection latency and lower bandwidth

$$\text{Speedup} = \frac{T_{TCP}}{T_{VIA}}$$

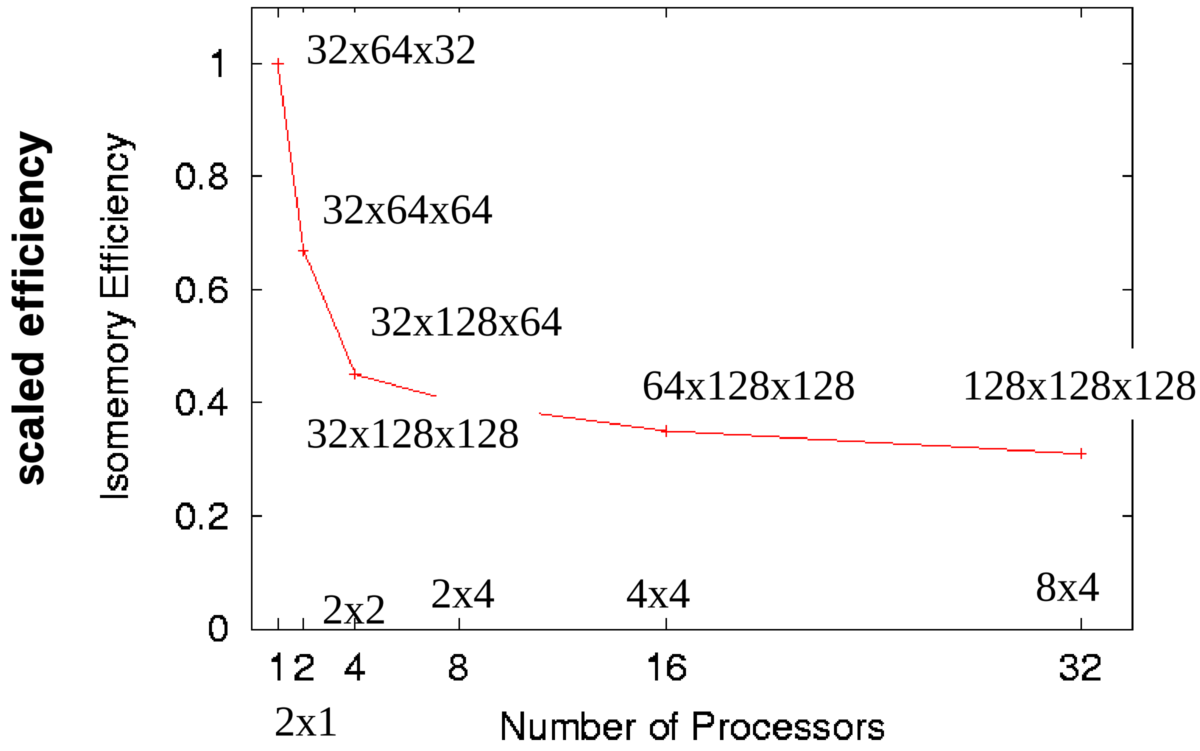
Simulations using VIA and a 32x128x128 grid



Drop in Efficiency due to both **a single network card** and memory sharing

$$\text{Speedup} = \frac{T_{TCP}}{T_{VIA}}$$

The scaled efficiency is bounded away from zero, so the pair algorithm-architecture is scalable



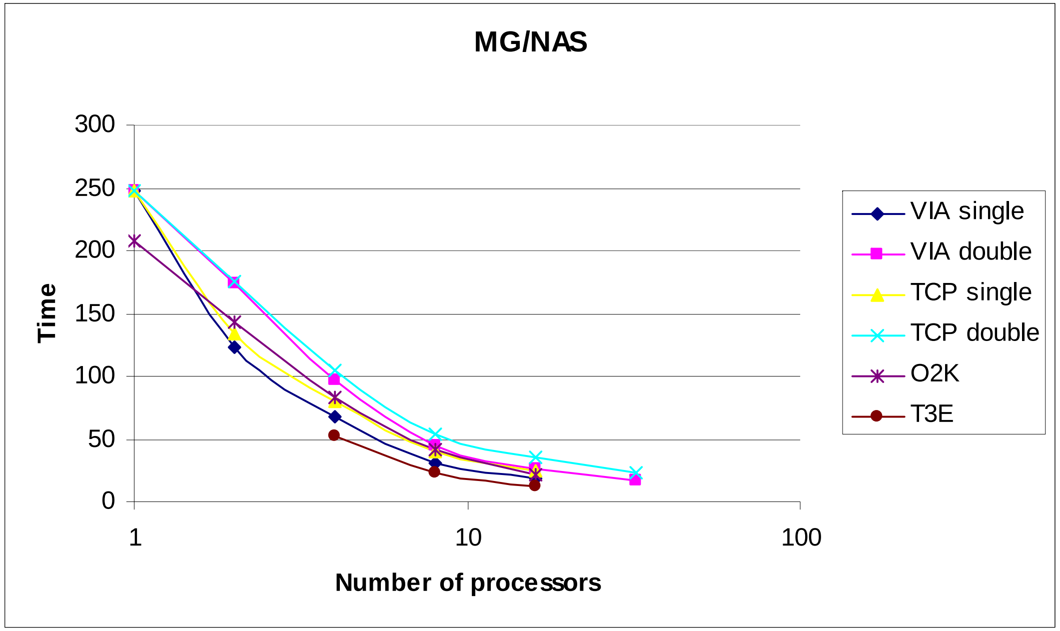
$$S_T(p,np) = \frac{T(N,p)}{T(nN,np)} = \frac{E(nN,np)}{E(N,p)}$$

Goal of the NAS Parallel Benchmark (`www.nas.nasa.gov`):

- Estimate the performance of a real CFD application on a parallel system by solving the following MPI-based source kernels: EP, MG, CG, FT, IS, LU, SP and BT

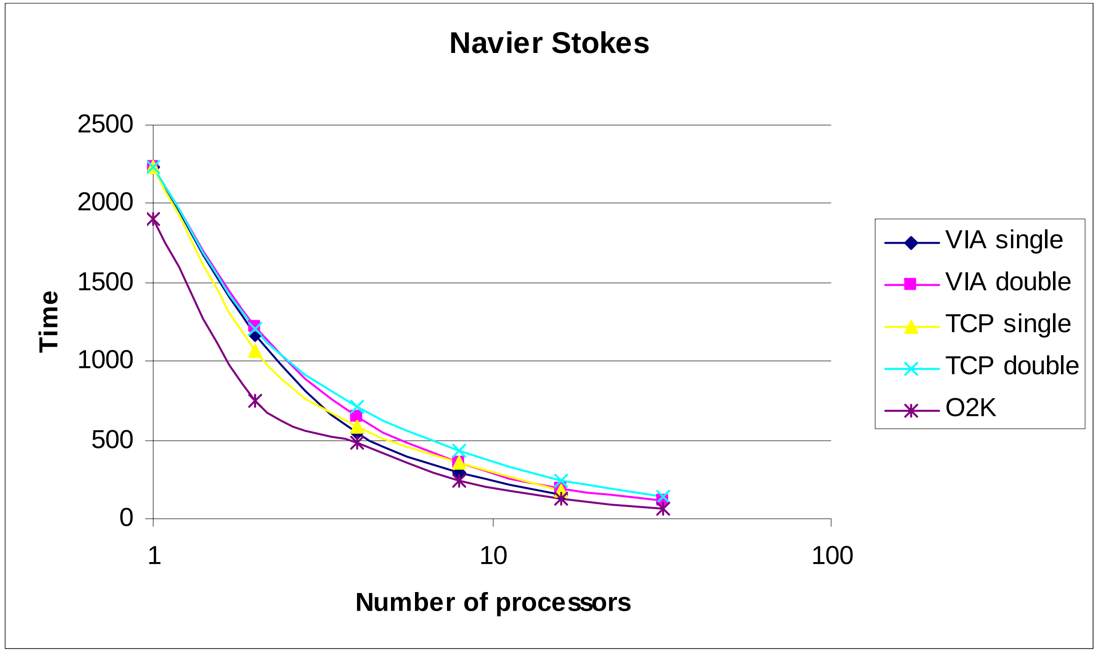
The NAS-MG multigrid benchmark solves Poisson's equation in 3-D with periodic boundary conditions using multigrid V-cycles on a uniform 256x256x256 grid (class-B, 20 cycles)

Grid partitioning is applied in the parallel implementation



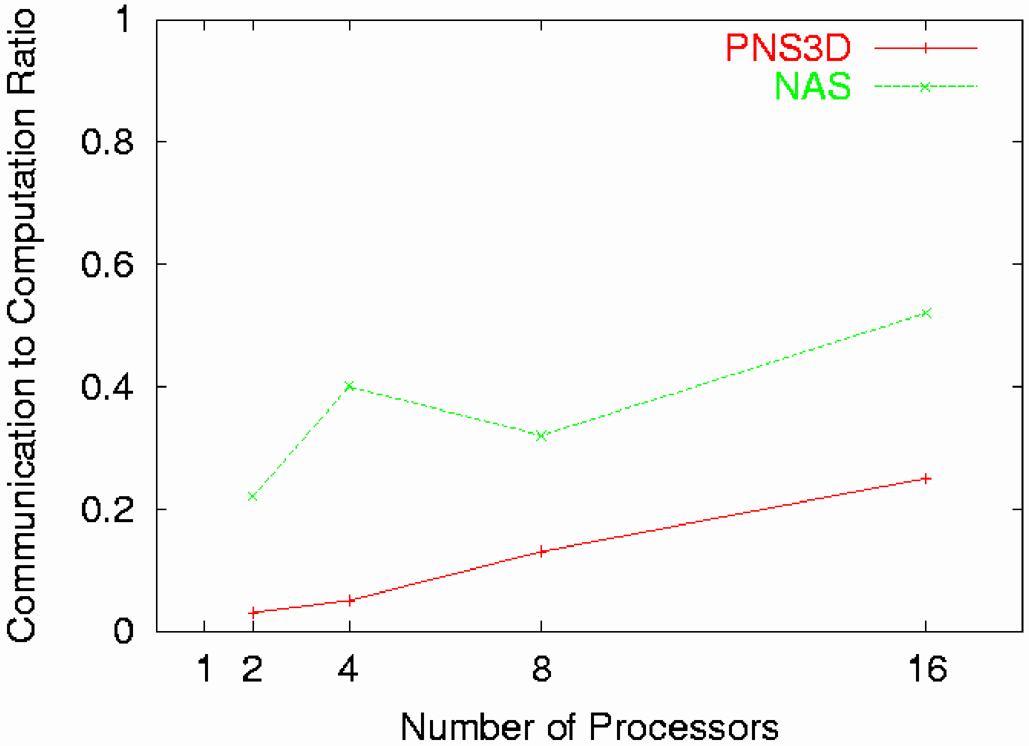
	VIA single	VIA double	TCP single	TCP double	O2K	T3E
1	248,23	248,23	248,23	248,23	207,49	
2	123,3	174,38	133,55	174,67	143,12	
4	68,4	97,23	79,5	105,36	82,75	52,1
8	31,52	44,71	40,13	53,15	41,85	22,4
16	18,33	26,82	24,26	34,73	22,23	12,9
32		17,28		23,11		

The poor data locality in the message passing affects more strongly to the MIPS 10000 processor performance (10 MB/seg with poor data locality, see PrLT00)



	VIA single	VIA double	TCP single	TCP double	O2K
1	2236,87	2236,87	2236,87	2236,87	1898,5
2	1163,19	1217,47	1060,31	1209,43	753,5
4	547,66	644,394	588,974	707,026	484
8	294,025	351,678	350,033	425,145	242
16	152,238	187,349	183,912	235,054	129,5
32		110,956		144,248	59

The plane implicit smoother exhibits a higher locality



Our code shows a higher computation to communication ratio

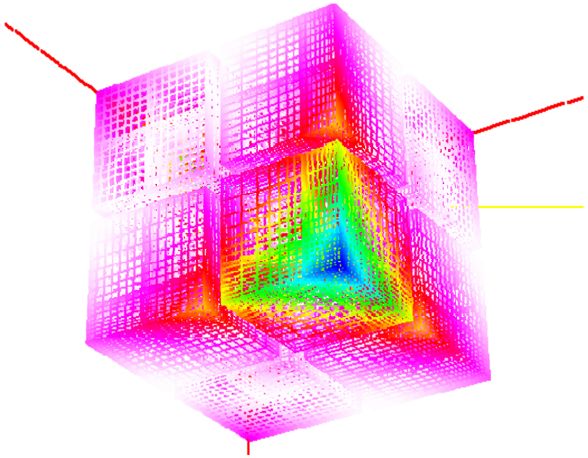
- Textbook Multigrid Convergence is attained for the mode problem (flat plate at yaw):
  - Independent of Grid Size
  - High Stretching Factors
  - Reynolds number up to 10^5
- Parallel version of the smoother exhibits similar convergence properties to the lexicographic order with good scalability
- Coral shows better performance than O2K and of course much better performance-cost ratio
- Real CFD codes shows a higher computation to communication ratio and higher memory access locality than the NAS/MG kernel due to the need of implicit smoothers. So our code characterizes better the CFD work load

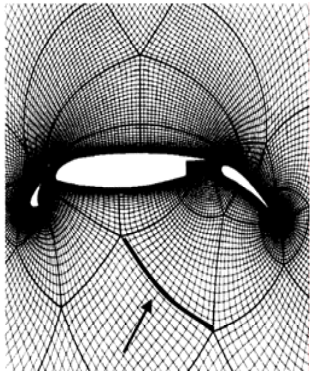
Details of this research in

R. S. Montero, I. M. Llorente and M. D. Salas, **Robust Multigrid Algorithms for the Navier-Stokes Equations**, in press, Computational Physics, Academic Press

R. S. Montero, I. M. Llorente and M. D. Salas, **Semicoarsening and Implicit Smoothers for the Simulation of a Flat Plate at Yaw**, ICASE Report No. 2001-13

- Flow over rectangular bodies sitting on a flat plate to study the flow field over a MEMS (MEMS= micro-electronic-mechanical systems) device or over buildings
- Turbulence model
- Multiblock grids with **two levels of parallelism** MPI-OpenMP (LIDM00)





- Distributive smoothers for Navier-Stokes
- Non-cartesian grids
- Incompressible Navier-Stokes

Acknowledgements and People Involved in this Research

Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Francisco Tirado Fernández
Manuel Prieto Matías
Rubén Santiago Montero

ICASE
Institute for Computer Application
In Science and Engineering

Manuel D. Salas
Boris Diskin

NASA Langley

N. Duane Melson
James L. Thomas

This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-2199. The research was also supported in part by the Spanish research grant CICYT TIC 99/0474 and the US-Spain Joint Commission for Scientific and Technological Cooperation

I am sorry if I have forgotten to include any of your references

Bran84 A. Brandt, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamic*, GMD-Sudien 85, 1984

Bran98 Achi Brandt, *Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD*, ICASE Interim Report 32, 1998

BrFJ00 P. N. Brown, R. D. Falgout, and J. E. Jones, *Semicoarsening Multigrid on Distributed Memory Machines*, SIAM Journal of Scientific Computing, Vo. 21, Nº 5, 2001

BrMR82 A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations*. Report, Inst. for Computational Studies, Fort Collins, Colo., 1982

BrYa92 Achi Brandt, and I. Yavneh, *On Multigrid Solution of High-Reynolds Incompressible Entering Flows*, Journal of Computational Physics, Vol. 101, pp. 151-164, 1992

CFHJ00 Cleary, A.J., R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, G.N. Miranda, and J.W. Ruge, *Robustness and Scalability of Algebraic Multigrid*, SIAM Journal on Scientific Computing, Vol. 21, Nº 5, 2000

Disk99 B. Diskin, *Solving Upwind-Biased Discretization II: Multigrid Solver Using Semicoarsening*, ICASE Report 99-14, March 1999

DMRR89 J. Dendy, *Multigrid methods for the three-dimensional petroleum reservoir simulation*, S. McCormick, J. Ruge, T. Russell, and S. Schaffer, Proc. tenth SPE Symposium on Reservoir Simulation, 1989

FrMc91 P. O. Frederickson, O. A. McBryan. *Normalized convergence rates for the PSMG method*. SIAM J. Sci. Stat. Comput., Vol 12, 1991, pp. 221-229

GrGK93 A. Grama, A. Gupta, and V. Kumar, *Isoefficiency Function: A Scalability Metric for Parallel Algorithms and Architectures*, IEEE Parallel and Distributed Technology, Vol. 1, Nº 3, 1993

Gold33 S. Goldstein, *On the Two-dimensional Steady Flow of Viscous Fluid behind a Solid Body*, Proc. Royal Society of London, Series A, pp. 545-573

Gust88 J. L. Gustafson, *Reevaluating Amdahl's Law*, Communications of the ACM, Vol. 31, Nº 5, 1988

HaHG92 T. Hayase, J. Humphrey, and R Greif, *A Consistently Formulated QUICK Scheme for Fast and Stable Convergence Using Finite-Volume Iteration Calculation Procedures*, Journal of Computational Physics, Vol. 98, pp. 108-118, 1992

LlMe00 I. M. Llorente, and N. D. Melson, *Behavior of Plane Relaxation Methods as Multigrid Smoothers*, Special ETNA issue on the Ninth Copper Mountain Conference on Multigrid Methods, Vol 10, 2000

LlDM00 I. M. Llorente, B. Diskin, N. D. Melson. *Plane smoothers for multiblock grids: Computational aspects*. SIAM Journal on Scientific Computing, Edited by SIAM, Vol. 22 Number 1 pp. 218-242, 2000

LlTi97 I. M. Llorente, and F. Tirado, *Relationships between Efficiency an Execution Time of Full Multigrid Methods on Parallel Computers*, IEEE Transactions on Parallel and Distributed Systems, Vol. 8, Nº 6, 1997

LlTV96 I. M. Llorente, and F. Tirado, and L. Vázquez, *Some Aspects about the Scalability of Scientific Applications on Parallel Computers*, Parallel Computing, Vol. 22, 1996

MoDe99 D. Moulton, J. Dendy. *MPI-based black box multigrid for workstation clusters*. Proc. of the ninth Copper Mountain Conference on Multigrid Methods, 1999

MoLS01 R. S. Montero, I. M. Llorente and M. D. Salas, *Robust Multigrid Algorithms for the Navier-Stokes Equations*, in press, Computational Physics, Academic Press

Muld89 W. A. Mulder. *A new multigrid approach to convection problems*. J. Comput. Phys, 83, 1989

OGWW98 C. W. Oosterlee, F. Gaspar, T. Washio, and R. Wienands, *Multigrid Line Smoothers for Higher Order Upwind Discretizations of Convection-Dominated Problems*, Journal of Computational Physics, Vol. 1, pp. 274-307, 1998

Oost95 C. W. Oosterlee. *The convergence of parallel multiblock multigrid methods*. Appl. Num. Mathematics 19, 1995

OvRo93 A. Overman, J. van Rosendale. *Mapping Robust Parallel Multigrid Algorithms to Scalable Memory Architectures*. ICASE Report 93-24 1993

Vank86 S. P. Vanka. *Block-Implicit Multigrid Solution of Navier-Stokes Equation in Primitive Variables*, Journal of Computational Physics, Vol. 65, pp. 94-121, 1986

PrLT00 M. Prieto, I. M. Llorente and F. Tirado, *Data Locality Exploitation in the Decomposition of Regular Domain Problems*, IEEE Transactions on Parallel and Distributed Systems, Vol. 11, Nº 11, 2000, pp. 1141-1150

PSEL01 M. Prieto, R. Santiago, D. Espadas, I. M. Llorente and F.Tirado, *Parallel Multigrid for Anisotropic Elliptic Equations*, Journal of Parallel and Distributed Computing, Vol. 61, No. 1, 2001

RuSt87 J. W. Ruge and K. Stuuben, *Algebraic Multigrid*, in Multigrid Methods, Frontiers in Applied Mathematics, SIAM, 1987

Scha98 S. Schaffer, *A Semicoarsening Multigrid Method for Elliptic Partial Differential Equations with Highly Discontonuos and Anisotropic Coefficient*, SIAM Journal of Scientific Computing, Vol. 20, Nº 1, 1998

SuNi90 X. Sun, and L. M. Ni, *Another View of Parallel Speedup*, Supercomputing '90 Proceedings, 1990

XiSc97 Dexuan Xie, Ridgway Scott. *The Parallel U-Cycle multigrid method*. Proc. eighth Copper Mountain Conference on Multigrid Methods. Copper Mountain, Colorado, Abril 1997.