Multiresolution et méthodes adaptatives pour les problèmes dominés par la convection

Marie Postel
Laboratoire Jacques Louis Lions

Mini symposium "'Méthodes de multirésolution"'
Congrès SMAI 2009 28 mai 2009

Outline of the talk

Multiresolution methods
- Multiscale analysis
- Adaptive scheme for hyperbolic PDEs

New developments
- Fully adaptive methods
- Multiresolution *a la* Harten

Local Time Stepping Srategies

Outline

Multiresolution methods
- Multiscale analysis
- Adaptive scheme for hyperbolic PDEs

New developments
- Fully adaptive methods
- Multiresolution *a la* Harten

Local Time Stepping Srategies

Adaptive numerical methods for PDEs

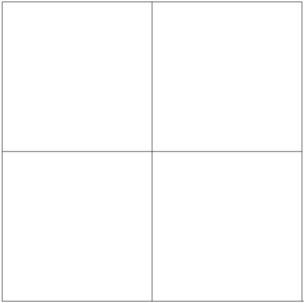
- The solution **u** is discretized on a non-uniform mesh $\mathcal{T}$ in which the resolution is locally adapted to its singularities (shocks, boundary layers, sharp gradients...).
- Goal: better trade-off between accuracy and CPU/memory requirements .
- The adaptive mesh is updated based on the a-posteriori information gained through the computation

$$(u_0, \mathcal{T}_0) \rightarrow (u_1, \mathcal{T}_1) \rightarrow \cdots \rightarrow (u_n, \mathcal{T}_n)$$

Adaptive numerical methods for PDEs

- The solution **u** is discretized on a non-uniform mesh $\mathcal{T}$ in which the resolution is locally adapted to its singularities (shocks, boundary layers, sharp gradients...).
- Goal: better trade-off between accuracy and CPU/memory requirements .
- The adaptive mesh is updated based on the a-posteriori information gained through the computation

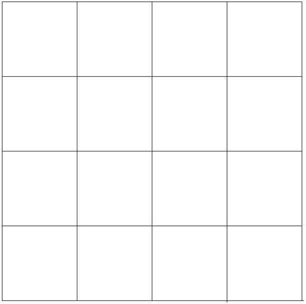
$$(u_0, \mathcal{T}_0) \to (u_1, \mathcal{T}_1) \to \cdots \to (u_n, \mathcal{T}_n)$$



Adaptive numerical methods for PDEs

- The solution **u** is discretized on a non-uniform mesh $\mathcal{T}$ in which the resolution is locally adapted to its singularities (shocks, boundary layers, sharp gradients...).
- Goal: better trade-off between accuracy and CPU/memory requirements .
- The adaptive mesh is updated based on the a-posteriori information gained through the computation

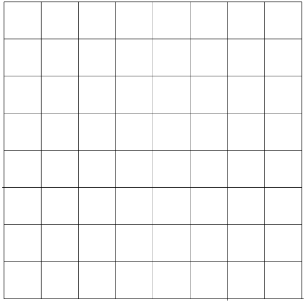
$$(u_0, \mathcal{T}_0) \to (u_1, \mathcal{T}_1) \to \cdots \to (u_n, \mathcal{T}_n)$$



Adaptive numerical methods for PDEs

- The solution **u** is discretized on a non-uniform mesh $\mathcal{T}$ in which the resolution is locally adapted to its singularities (shocks, boundary layers, sharp gradients...).
- Goal: better trade-off between accuracy and CPU/memory requirements .
- The adaptive mesh is updated based on the a-posteriori information gained through the computation

$$(u_0, \mathcal{T}_0) \to (u_1, \mathcal{T}_1) \to \cdots \to (u_n, \mathcal{T}_n)$$

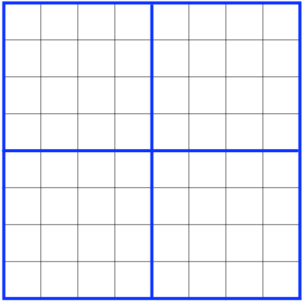


Two typical setups for adaptive methods

Steady state problems

$$F(u) = 0$$

the mesh $\mathcal{T}_n$ is refined according to local error indicators (for example based on residual $F(\mathbf{u}_n)$) and $\mathbf{u}_n \to \mathbf{u}$ as $n \to +\infty$.

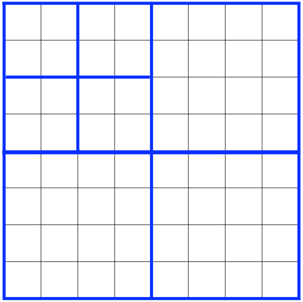


Two typical setups for adaptive methods

Steady state problems

$$F(u) = 0$$

the mesh $\mathcal{T}_n$ is refined according to local error indicators (for example based on residual $F(\mathbf{u}_n)$) and $\mathbf{u}_n \to \mathbf{u}$ as $n \to +\infty$.

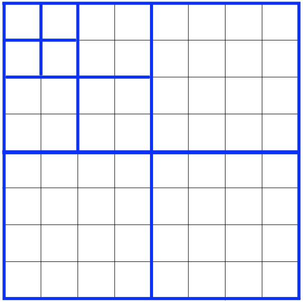


Two typical setups for adaptive methods

Steady state problems

$$F(u) = 0$$

the mesh $\mathcal{T}_n$ is refined according to local error indicators (for example based on residual $F(\mathbf{u}_n)$) and $\mathbf{u}_n \to \mathbf{u}$ as $n \to +\infty$.

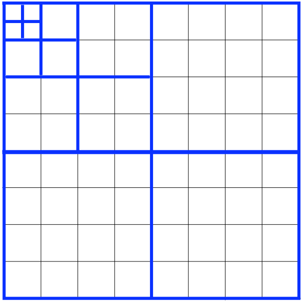


Two typical setups for adaptive methods

Steady state problems

$$F(u) = 0$$

the mesh $\mathcal{T}_n$ is refined according to local error indicators (for example based on residual $F(\mathbf{u}_n)$) and $\mathbf{u}_n \to \mathbf{u}$ as $n \to +\infty$.

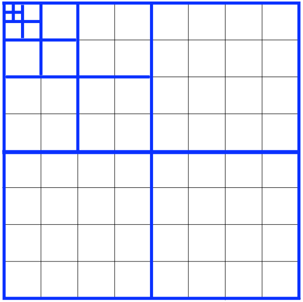


Two typical setups for adaptive methods

Steady state problems

$$F(u) = 0$$

the mesh $\mathcal{T}_n$ is refined according to local error indicators (for example based on residual $F(\mathbf{u}_n)$) and $\mathbf{u}_n \to \mathbf{u}$ as $n \to +\infty$.

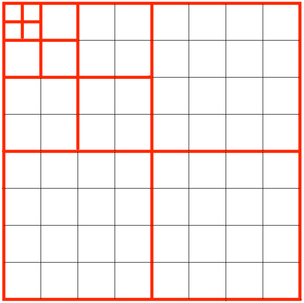


Two typical setups for adaptive methods

Evolution problems

$$\partial_t \mathbf{u} = \epsilon(\mathbf{u})$$

the numerical solution $\mathbf{u}_n$ approximates $\mathbf{u}(., n\Delta t)$ and the mesh $\mathcal{T}_n$ is dynamically updated from time step n to $n+1$.

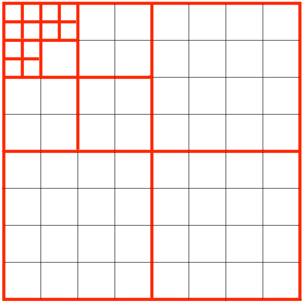


Two typical setups for adaptive methods

Evolution problems

$$\partial_t \mathbf{u} = \epsilon(\mathbf{u})$$

the numerical solution $\mathbf{u}_n$ approximates $\mathbf{u}(., n\Delta t)$ and the mesh $\mathcal{T}_n$ is dynamically updated from time step n to $n+1$.

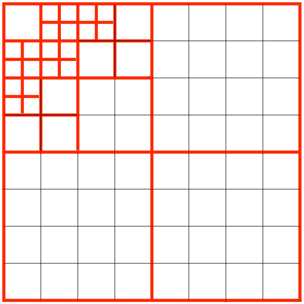


Two typical setups for adaptive methods

Evolution problems

$$\partial_t \mathbf{u} = \epsilon(\mathbf{u})$$

the numerical solution $\mathbf{u}_n$ approximates $\mathbf{u}(., n\Delta t)$ and the mesh $\mathcal{T}_n$ is dynamically updated from time step n to $n + 1$.

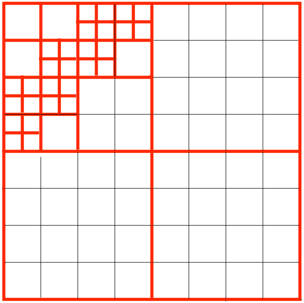


Two typical setups for adaptive methods

Evolution problems

$$\partial_t \mathbf{u} = \epsilon(\mathbf{u})$$

the numerical solution $\mathbf{u}_n$ approximates $\mathbf{u}(., n\Delta t)$ and the mesh $\mathcal{T}_n$ is dynamically updated from time step n to $n+1$.

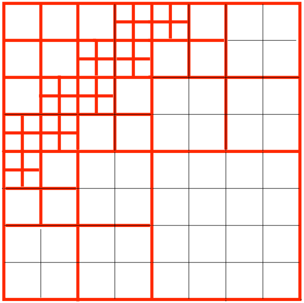


Two typical setups for adaptive methods

Evolution problems

$$\partial_t \mathbf{u} = \epsilon(\mathbf{u})$$

the numerical solution $\mathbf{u}_n$ approximates $\mathbf{u}(., n\Delta t)$ and the mesh $\mathcal{T}_n$ is dynamically updated from time step n to $n+1$.



Multiresolution method

Context: systems of hyperbolic PDEs

$$\partial_t u + \mathrm{Div}_x F(u) = 0$$

Difficulties:

- singularities
- theory : entropy weak solutions
- numerical analysis : costly schemes and limited order of convergence

Multiscale analysis

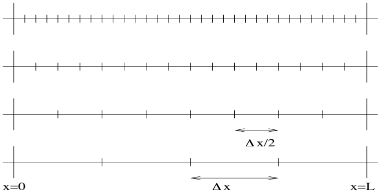
- Answer: adaptive mesh refinement
- Difficulties:
  - implementation : displacement of the singularities $\rightarrow$ mesh
  - convergence analysis
- Existing approaches:
  - AMR (Adaptative Mesh Refinement) (Berger, Oliger, ...)
  - Adaptive multiresolution flux evaluation (Harten, Abgrall, Chiavassa-Donat, ...)
  - Galerkin methods in wavelet spaces (Bacri-Mallat-Papanicolaou, Dahmen-Cohen-Masson, Maday-Perrier, Bertoluzza, ...)
- Fully adaptive multiresolution scheme
  - Harten's discrete multiresolution framework and link with wavelet theory
  - Finite volume scheme on a time adaptive grid

Discrete multiresolution framework

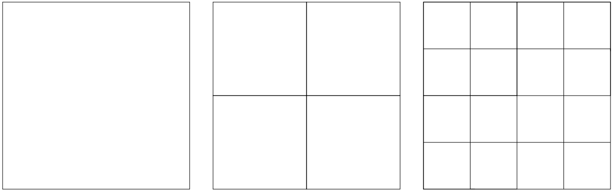
In 1D Discretization of a function $u(x)$ by its

- point values $u_{j,k} = u(k2^{-j})$
- mean values $u_{j,k} = 2^j \int_{k2^{-j}}^{(k+1)2^{-j}} u(x)\mathrm{d}x$

Dyadic hierarchy of grids
$S_j,\ j = 0, \ldots, J$

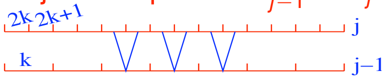


In 2D



Encoding / Decoding (mean value discretization)

Projection operator P^j_{j-1} of S_j on S_{j-1}: $U_{j-1} = P^j_{j-1} U_j$.



$$u_{j-1,k} = \frac{1}{2}(u_{j,2k} + u_{j,2k+1})$$

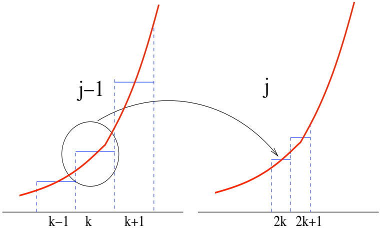
Prediction operator P^{j-1}_j from S_{j-1} to S_j : $\widehat{U}_j = P^{j-1}_j U_{j-1}$
is an approximation of U_j

- local
- consistent

$P^j_{j-1} P^{j-1}_j = I$

- exact for polynomials

of degree $r = 2n$



$$\widehat{u}_{j,2k} = u_{j-1,k} + \sum_{l=1}^{n} \gamma_l \left(u_{j-1,k+l} - u_{j-1,k-l}\right)$$

Multiscale decomposition

Prediction error:

$$u_{j,2k} - \hat{u}_{j,2k}$$

(consistence$\Rightarrow$)

$$= \hat{u}_{j,2k+1} - u_{j,2k+1}$$

Details :

$$d_{j,k} = u_{j,2k} - \hat{u}_{j,2k}$$

errors on all but one subdivisions

$$\begin{array}{c} \boxed{\text{Details } D_j + \text{Mean values } U_{j-1}} \\ \Updownarrow \\ \boxed{\text{Mean values } U_j} \end{array}$$

$$\begin{aligned} U_J &\Leftrightarrow (U_{J-1}, D_J) \\ &\Leftrightarrow (U_{J-2}, D_{J-1}, D_J) \\ &\ \ \vdots \\ &\Leftrightarrow (U_0, D_1, \ldots, D_J) \\ &\quad = MU_J = (d_\lambda)_{\lambda \in \nabla_J} \end{aligned}$$

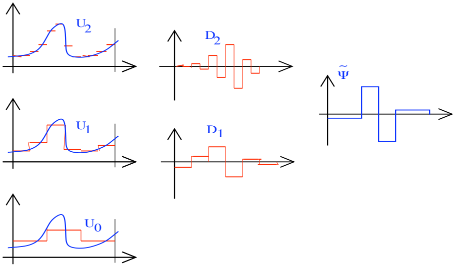
$$U_J \underset{\text{Decoding}}{\overset{\text{Encoding}}{\rightleftarrows}} MU_J$$

complexity

in $\mathcal{O}(\mathrm{Card}(S_J))$

Biorthogonal wavelet framework

$u = \sum_\lambda d_\lambda \Psi_\lambda,$
$d_\lambda = < u, \widetilde{\Psi}_\lambda >,$
$\widetilde{\Psi}_\lambda = 2^{j/2}\widetilde{\Psi}(2^j . - k),$
$\lambda = (j, k).$

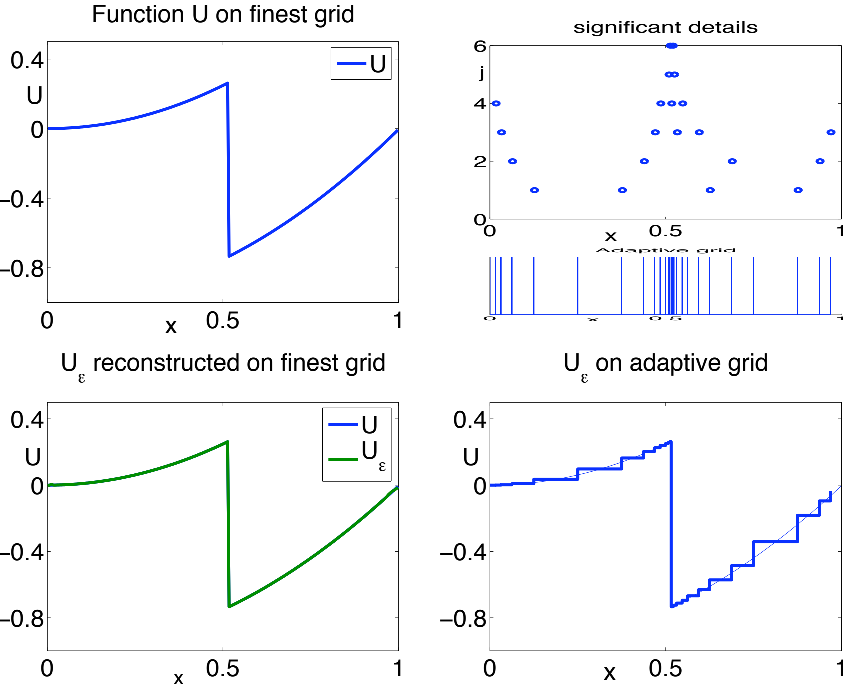


Measure of the local smoothness
Polynomial exactness of degree r

$$|d_{j,k}| \leq C2^{-js}$$
if $u \in C^s(\text{Supp } \widetilde{\Psi}_{j,k}),$
with $s \leq r$.

Thresholding
Set of significant details

$$u \to \mathcal{T}_\Lambda u := \sum_{\lambda \in \Lambda} d_\lambda \Psi_\lambda$$
$$\Lambda = \Lambda_\varepsilon := \{\lambda, |d_\lambda| \geq \varepsilon\}.$$


Function U on finest grid
U
x
significant details
j
x
Adaptive grid
x
U_ε reconstructed on finest grid
U
U_ε
x
U_ε on adaptive grid
U
x

- Thresholding

$$\Gamma_\varepsilon = \{\lambda \in \nabla_J, |d_\lambda| \geq \varepsilon_{|\lambda|}\}, \quad \varepsilon_j = 2^{d(j-J)}\varepsilon.$$

$$\mathcal{T}_\varepsilon(d_\lambda) = \begin{cases} d_\lambda \text{ if } \lambda \in \Gamma_\varepsilon \\ 0 \text{ otherwise} \end{cases}$$

$$\mathcal{A}_\varepsilon U_J = \mathcal{M}^{-1}\mathcal{T}_\varepsilon \mathcal{M} U_J.$$

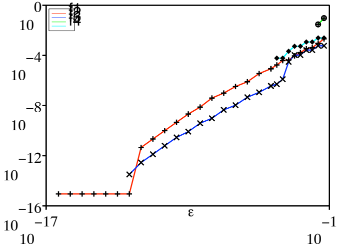
- Adaptive grid $S^\varepsilon = S(\Gamma^\varepsilon)$

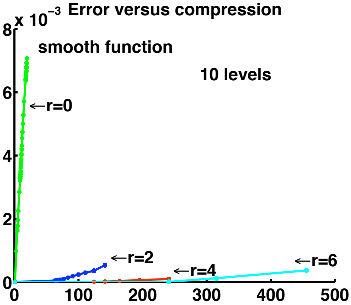
$$(d_\lambda)_{\lambda\in\Gamma_\varepsilon} \longleftrightarrow (U_\lambda)_{\lambda\in S_\varepsilon}.$$

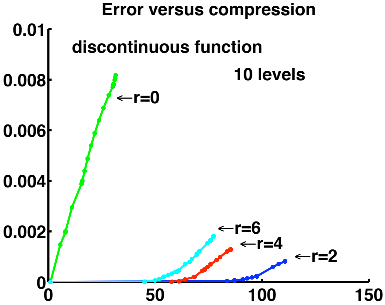
Tree structure for the adaptive tree Γ_ε $\Rightarrow$ Complexity of the coding / decoding algorithm in $\mathcal{O}(\mathrm{Card}(\Gamma_\varepsilon))$.

Control of error

$$\|\mathcal{A}_\varepsilon U_J - U_J\|_{L^1} \leq C\varepsilon$$







Adaptive scheme for hyperbolic PDEs

$$\partial_t u + \mathrm{Div}_x F(u) = 0.$$

Reference scheme on finest grid S_J:

$$\begin{aligned} U_J^{n+1} &= \mathcal{E}_J U_J^n \\ U_\lambda^{n+1} &= U_\lambda^n + B(U_\nu^n; \nu \in V_\lambda \subset S_J), \quad \lambda \in S_J \end{aligned}$$

Harten's algorithm still on finest grid S_J

$$U_{\varepsilon,\lambda}^{n+1} = U_{\varepsilon,\lambda}^n + B^\varepsilon(U_{\varepsilon,\nu}^n; \nu \in V_\lambda \subset S_J), \quad \lambda \in S_J$$

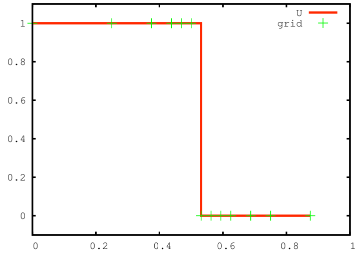
Fully adaptive algorithm on $S(\Gamma_\varepsilon^n)$

$$U_\varepsilon^{n+1} = \mathcal{E}_\varepsilon U_\varepsilon^n.$$

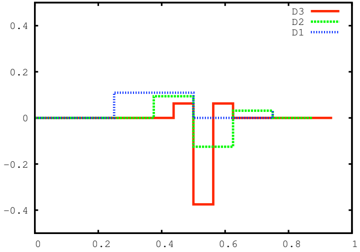
Evolution of $(U_\varepsilon^n, \Gamma_\varepsilon^n)$ into $(U_\varepsilon^{n+1}, \Gamma_\varepsilon^{n+1})$

- Encoding of the solution at time t_n
- Thresholding to obtain the tree Γ_ε^n
- Prediction
- Decoding
- Evolution

Solution at time t_n



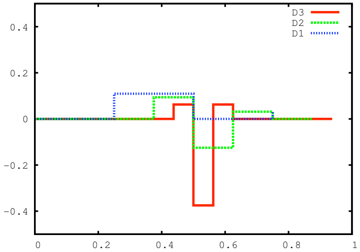
Details at time t_n



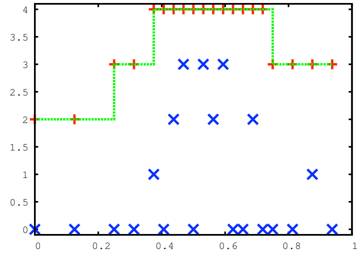
Evolution of $(U_\varepsilon^n, \Gamma_\varepsilon^n)$ into $(U_\varepsilon^{n+1}, \Gamma_\varepsilon^{n+1})$

- Encoding
- Thresholding to obtain the tree Γ_ε^n
- Prediction of the tree $\widetilde{\Gamma}_\varepsilon^{n+1}$ and grid $S(\widetilde{\Gamma}_\varepsilon^{n+1})$ (Harten heuristics or more refined strategy)
- Decoding
- Evolution

Details at time t_n

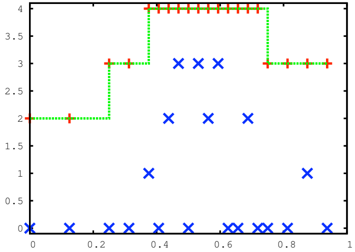


Tree $\widetilde{\Gamma}_\varepsilon^{n+1}$ and grid $S(\widetilde{\Gamma}_\varepsilon^{n+1})$



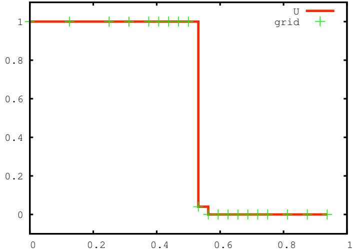
Evolution of $(U_\varepsilon^n, \Gamma_\varepsilon^n)$ into $(U_\varepsilon^{n+1}, \Gamma_\varepsilon^{n+1})$

Tree $\widetilde{\Gamma}_\varepsilon^{n+1}$ and grid $S(\widetilde{\Gamma}_\varepsilon^{n+1})$S



- Encoding
- Thresholding
- Prediction
- Decoding on the grid $S(\widetilde{\Gamma}_\varepsilon^{n+1})$
- Evolution with fluxes computed using reconstructed solution

Solution at time t_{n+1}



Error analysis

Theorem

Cohen, Kaber, Müller, P., Math. Comp., 2003

$$\|U_J^n - U_\varepsilon^n\|_{L^1} \leq Cn\varepsilon$$

Hypotheses

- *More demanding rules for the refinement of the tree*
- *Smoothness of the underlying wavelet*
- *Scalar equation, on 1D or cartesian multi-D grid*
- *Local reconstruction at the finest level*

Recent relaxation of the last hypothesis by Hovhannisyan & Müller

Outline

Recent developements and trends

Workshop on Multiresolution and Adaptive Methods for Convection-Dominated Problems
January 2223, 2009
Paris, France
Promotion of multiresolution and other adaptive techniques for complex applications where convection is the prevailing phenomenon.

- Robustness of the multiresolution method : wide variety of extensions
- Importance of data structures / parallelisation
- Problem dependent performances / comparison between Multiresolution and AMR
- Local Time Stepping strategies

http://www.ann.jussieu.fr/mamcdp09

New applications

- Incompressible Navier-Stokes equations. Two-dimensional steady incompressible flow. Müller-Stiriba, Bramkamp-Lamby-Müller
- Compressible Navier-Stokes equations. Schneider-Farge-Nguyen, Chiavassa-Donat-Boiron
- Compressible Euler equations. Schneider-Roussel, Müller-Stiriba
- Diphasic compressible flows.
  Slugging in pipelines. Coquel-Tran-MP-Nguyen-Andrianov
  Laser-Induced Cavitation. Bubbles Müller-Bachmann-Kröninger-Kurz-Helluy
- Reaction-diffusion equations. 2D thermo-diffusive flames, 3D flame balls. Schneider-Roussel-Gomes-Domingues
- Plasma simulation - Vlasov equation. Sonnendrücker, Campos-Pinto, Mehrenberger
- Shallow water flows Lamby-Müller-Stiriba, Chiavassa-Donat-Gavara

New tools

- Turbulent weakly compressible 3d mixing layer, Coherent Vortex Simulation. Schneider & Farge
  Penalisation method around obstacle Chiavassa *et al*
- Anisotropic mesh refinement
  Cohen-Dyn-Hecht-Mirebeau
- Treatment of source terms with well balanced schemes
  Chiavassa-Donat-Gavara
- Data structures / Parallelisation
  Müller *et al*, Sonnendrücker *et al*
- Local Time stepping
  Müller *et al*, Schneider *et al* , Coquel *et al*, Faille-Nataf

Data structure

In the fully adaptive scheme, one cell is viewed with respect to

- its neighbours in the adaptive grid when updating the solution with the numerical scheme
- its parents and children when updating the adaptive grid at each time step

Data structures become crucial with 3D applications and / or when going parallel

Data structure

- Tree structure (Schneider - Roussel) : Full priority to the multiresolution
- Hash tables (Müller - Voss) : compromise between multiresolution efficiency and implementation of the scheme on the adaptive grid
- Sparse data structure (Sonnendrücker - Latu)

O. Roussel et al. / Journal of Computational Physics 188 (2003) 493–523

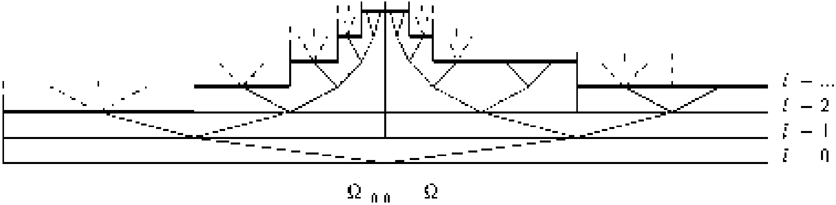


Fig. 1. Example of graded tree data structure in 1D for $s = 1$, $s' = 2$.

Data structure

- Tree structure (Schneider - Roussel) : Full priority to the multiresolution
- Hash tables (Müller - Voss) : compromise between multiresolution efficiency and implementation of the scheme on the adaptive grid
- Sparse data structure (Sonnendrücker - Latu)

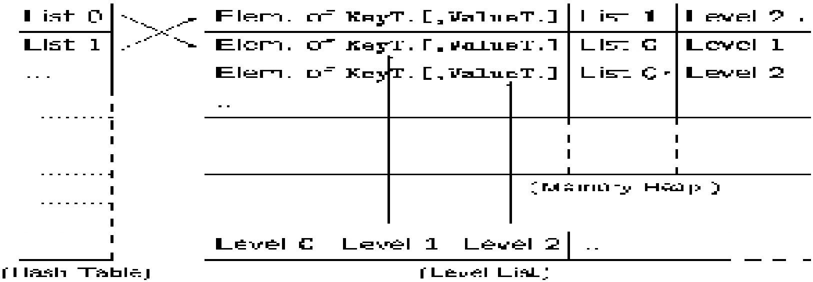


FIGURE 6. Linked hash map.

FIG.: Courtesy ¨Mueller *et al*

Parallelisation of adaptive methods

Multiple criteria

- Load balancing partitionning of the adaptive mesh
- Minimization of the sub domains interfaces
- Efficiency of the partitionning to be used in a time adaptive algorithm

Space Filling Curves	Graph partitionning
(Peano-Hilbert) YODA (Metzmeyer, Hoenen) RAMSES (CEA) QUADFLOW (Mogosan, Müller)	(Karypis, Kumar) GGP, GGGP, BKL, etc. METIS, PETSc Code_Aster (CEA-EDF-Inria) Scotch (Pellegrini)

Multilevel cost effective techniques

Cost-reduction implementation : NO memory gain! but NO data-structures needed

- Bihari-Harten JCP (1996)SISC (1997) (1D-2D/tensor-product CA-MR) Bihari AIAA (2003) (CA-MR unstructured) FV
- Abgrall-Harten SINUM (1996) (CA-MR unstructured) FV
- Dahmen, Gottschlich-Müller,Müller Num. Math (2000) (curvilinear meshes, Cell-Average Framework) FV
- Cohen, Dyn, Kaber, MP JCP (2000) (2D-unstructured) FV
- Chiavassa-Donat SISC-(2001) (2D/tensor-product PV-MR) FD

Recent Applications

FD schemes

- Well-Balanced (TVDB) schemes for Shallow water flows (Donat-Gavarra). 2D: CPU Gain from 5.5 to 7 (for grids from 128^2 to 512^2
- Penalization method for compressible Navier Stokes flows around obstacle (hig Mach number interactions) [Chiavassa,Donat,Boiron], 2D: CPU Gain from 3 to 5.5 (for grids from 512^2 to 1536^2)
- Propagation of waves in porous medium [Chiavassa, Lombart, Piraux]

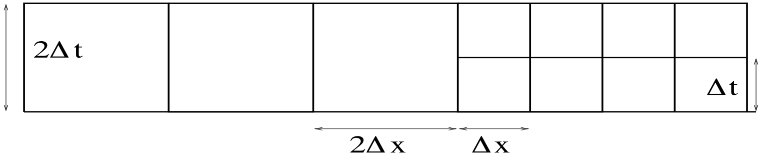
Outline

From multiresolution to locally adaptive time stepping

Standard MR scheme

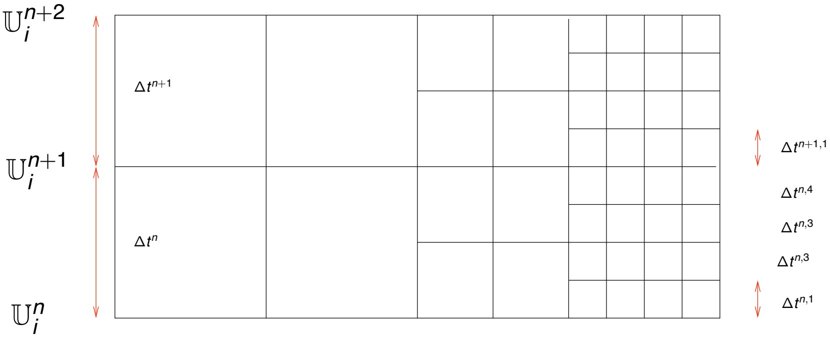
- At a given time, same time step used for all cells
- Time step ruled by CFL condition $\Delta t^n < \min_j \dfrac{\Delta x_j}{\mu(\mathbb{U}_j^n)}$.

Principle of the LTS strategy: use a time-step adapted to the local size of the cell constant $\lambda = \Delta t / \Delta x$
(Berger, Colella '89, Mueller, Stiriba '06)



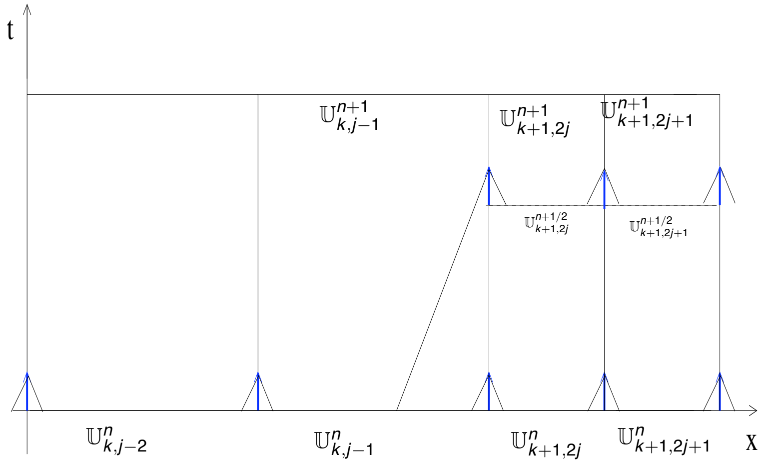
Goal : further reduction of # calls to flux functions and expensive state laws $\Rightarrow$ CPU $\downarrow$

Evolution of the solution at intermediate time



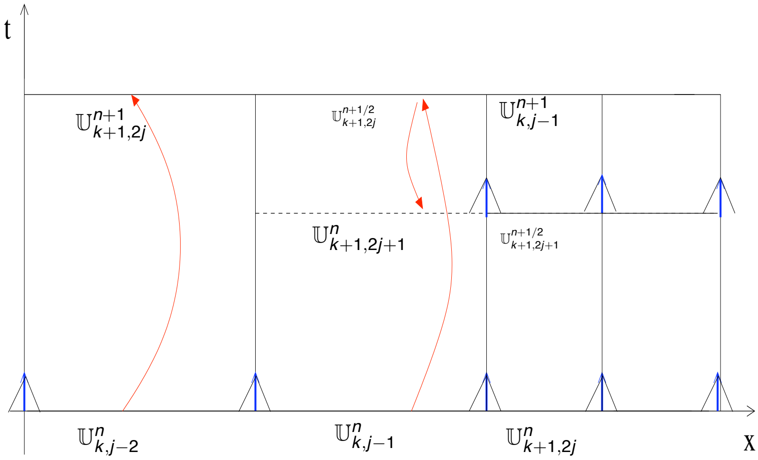
Synchronization at intermediate times

Osher-Sanders strategy



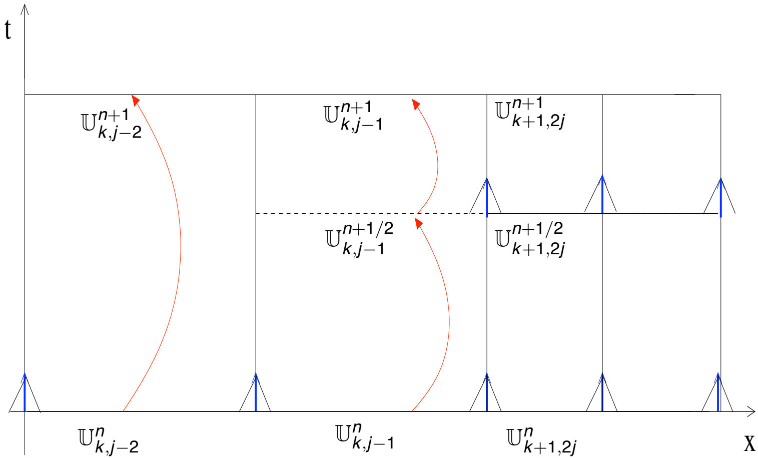
Synchronization at intermediate times

Schneider-Roussel-Gomes-Domingues strategy



Synchronization at intermediate times

Müller & Stiriba strategy



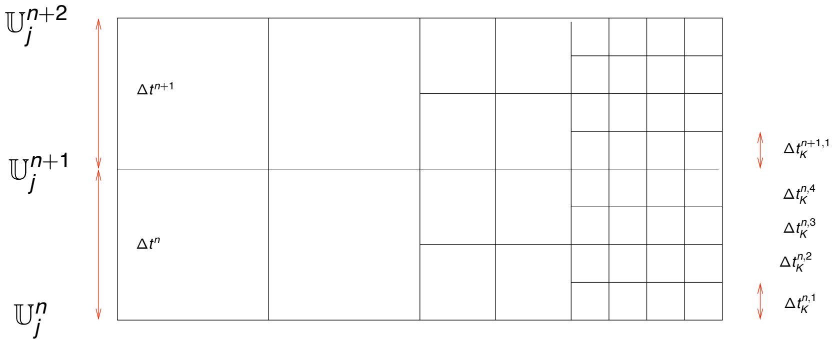
Local Time Stepping (LTS)
Computation of the time-step

Definitions

Macro time step $\Delta t^n \leftrightarrow$ largest cells

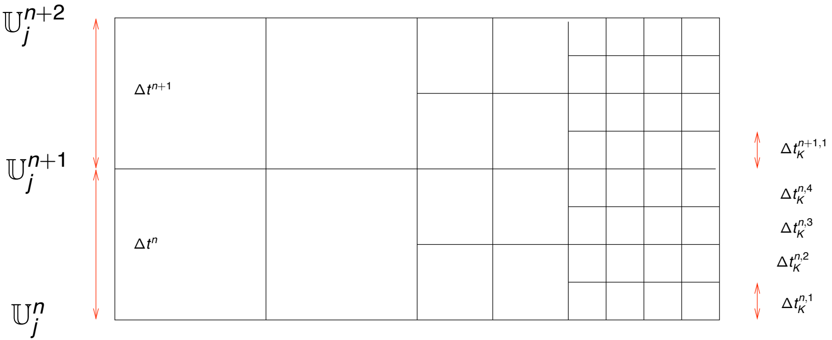
Micro time step $\Delta t_K^{n,i} \leftrightarrow$ smallest cells

$$\Delta t^n = \sum_{i=1}^{2^K} \Delta t_K^{n,i}$$



Local Time Stepping (LTS)
Computation of the time-step

- The micro time-steps decrease during each macro time-step
- They are updated along with the solution while ensuring the stability condition



Local Time Stepping (LTS)
Computation of the time-step

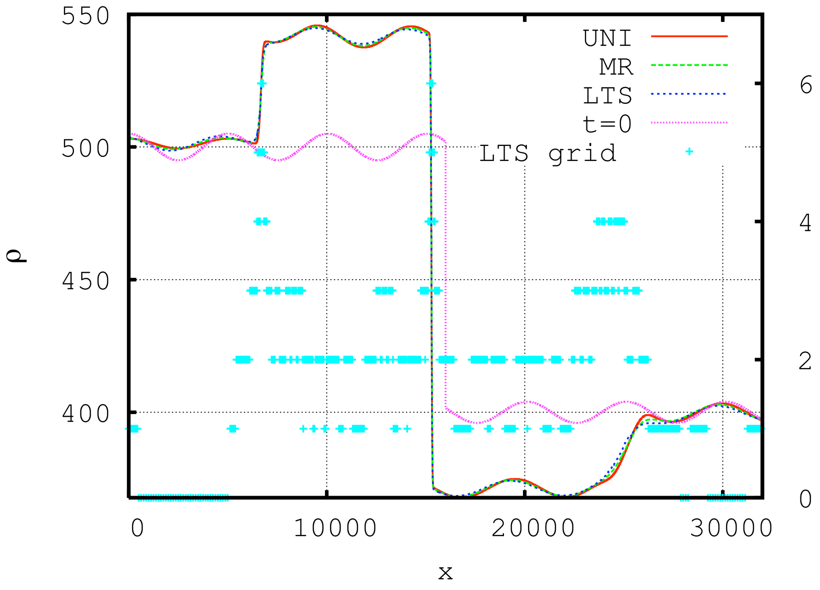
- The micro time-steps decrease during each macro time-step
- They are updated along with the solution while ensuring the stability condition
  - The first micro time-step $\Delta t^{n,1} < \min_j \frac{\Delta x_j}{\mu(\mathbb{U}_j^n)}$.
  - The others

$$\Delta t^{n,p} < \min\left(\Delta t_K^{n,p-1}, \min_j \frac{\Delta x_j}{\mu(\mathbb{U}_j^{n,p-1})}\right)$$

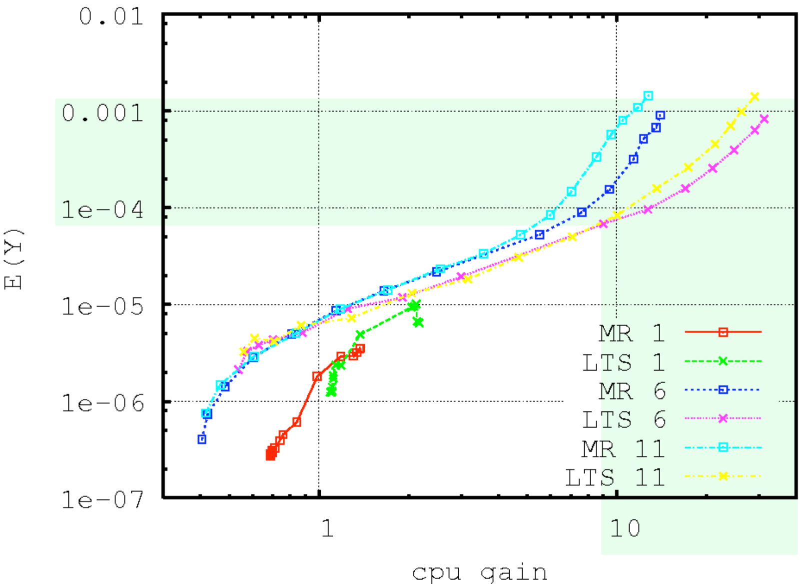
    where $\mathbb{U}_j^{n,p-1}$ is the updated solution at time $t_n + \sum_{i=1}^{p-1} \Delta t_K^{n,i}$

- The macro time-step $\Delta t^n = \sum_{i=1}^{2^K} \Delta t^{n,i}$

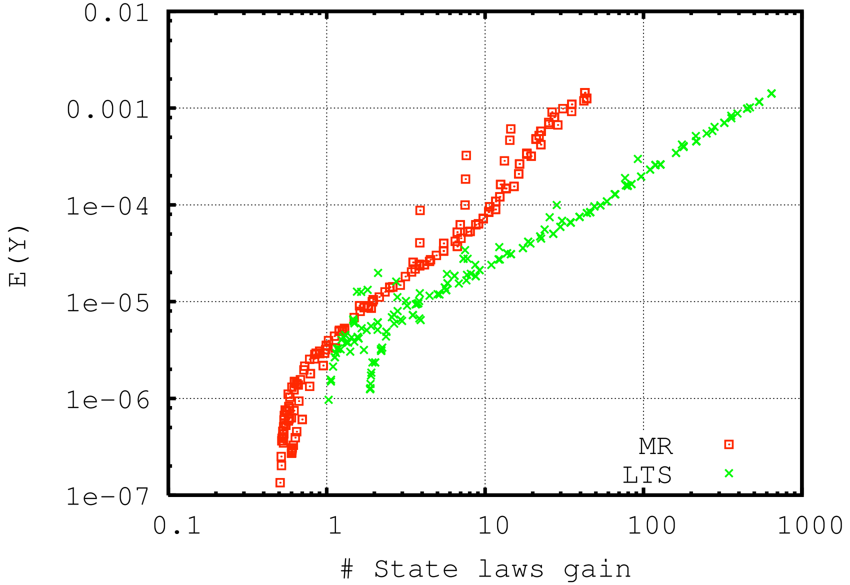
Test case with smoothly varying initial condition



Error versus computing time gain - simplistic state law



Error versus number of calls to state laws gain



Conclusions and perspectives

- Very active and uprising field
- Need for strong collaboration between computer scientists and mathematicians
- Comparison between AMR and MR
- Visit the website

  http://www.ann.jussieu.fr/mamcdp09