Numerical Models for Atmospheric Simulations

Ramaz Botchorishvili
Faculty of Exact and Natural Sciences
Tbilisi State University
Georgia

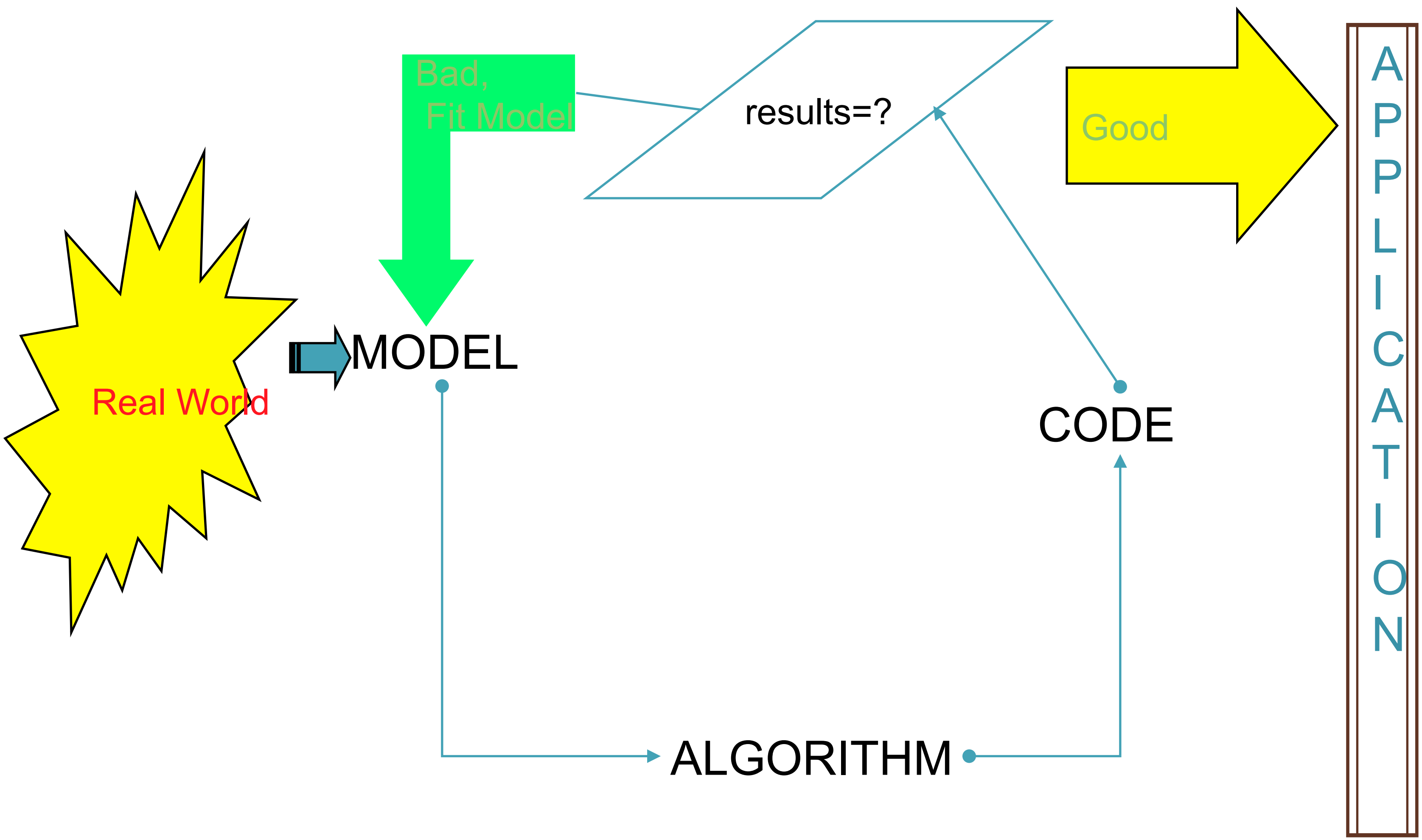
Outline

- Is numerical analysis important ?
  - Low accuracy of numerical schemes can not be compensated by means of increasing computational power
  - Qualitative properties of exact solution must be respected otherwise numerical models can fail

Outline

- Some approaches
  - need for large time step?
    - locally one directional implicit schemes
  - need for refinement and derefinement?
    - multischeme:
      - enables dynamic adaptation of computational meshes and discretization schemes

Modelling



Bad,
Fit Model
results=?
Good
Real World
MODEL
CODE
ALGORITHM
APPLICATION

Is development of new numerical methods important?

- model = set of equations
- development of new model
  - properties of solutions
  - elaboration of numerical schemes
  - studying convergence theoretically
  - coding, implementation
  - construction of test cases
  - studying convergence practically
- Expensive – is it worth of efforts?

Problem 1, governing equation

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{N} \frac{\partial A_i(u)}{\partial x_i} + b(u) \sum_{i=1}^{N} \frac{\partial z_i(x)}{\partial x_i} = 0, \qquad t \geqslant 0, \ \ x \in \mathbb{R}^N,$$

$$u(0,x) = u_0(x), \qquad u_0(x) \in L^1 \cap L^\infty,$$

Entropy solution

$$\frac{\partial S(u)}{\partial t} + \sum_{i=1}^{N} \frac{\partial \eta_i(u)}{\partial x_i} + S'(u) b(u) \sum_{i=1}^{N} \frac{\partial z_i(x)}{\partial x_i} \leqslant 0$$

Steady states, equilibrium states

$$\sum_{i=1}^{N} \frac{\partial}{\partial x_i}(D_i(u) + z_i(x)) = 0, \qquad D_i(u) = \int_0^u \frac{a_i(s)}{b(s)} \mathrm{d}s < +\infty, \quad 1 \leqslant i \leqslant N.$$

Problem I, numerical schemes

Standard cell centered

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+\sum_k\sum_l\frac{|\Gamma_{jk}^l|}{|C_j|}A(u_j^n,u_k^n,\vec{n}_{jk}^l)+\frac{b(u_j^n)}{|C_j|}\sum_k\sum_l|\Gamma_{jk}^l|\langle z(\vec{x}_k),\vec{n}_{jk}^l\rangle=0,$$

Equilibrium type 1

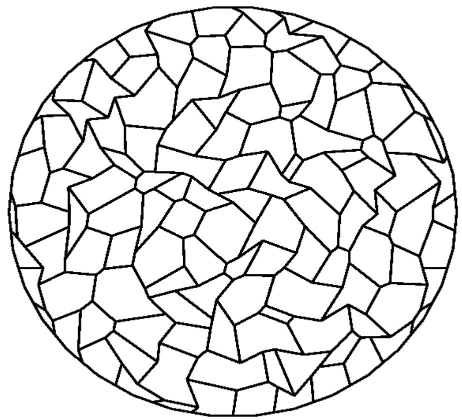
$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+\frac{1}{|C_j|}\sum_k\sum_l|\Gamma_{jk}^l|A(u_j^n,u_{k,l-}^n,\vec{n}_{jk}^l)=0,$$

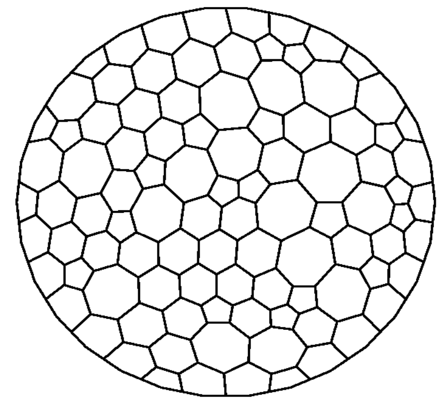
Equilibrium type 2

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+\frac{1}{|C_j|}\sum_k\sum_l|\Gamma_{jk}^l|A(u_j^n,u_k^n,\vec{n}_{jk}^l)+$$

$$+\sum_k\sum_l\frac{|\Gamma_{jk}^l|}{|C_j|}\frac{A(u_j^n,u_k^n,\vec{n}_{jk}^l)-A(u_j^n,u_j^n,\vec{n}_{jk}^l)}{\langle D(u_k^n),\vec{n}_{jk}^l\rangle-\langle D(u_j^n),\vec{n}_{jk}^l\rangle}\cdot\left(\langle z(\vec{x}_k),\vec{n}_{jk}^l\rangle-\langle z(\vec{x}_j),\vec{n}_{jk}^l\rangle\right)=0.$$

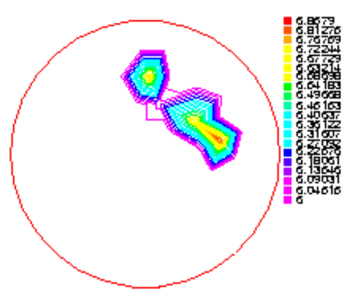
Problem 1, meshes and results

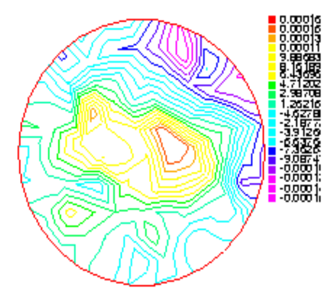




Equilibrium scheme:
irregular FV mesh,
high accuracy

Standard scheme:
smoothed FV mesh,
low accuracy





Problem 1, Laptop vs PC Cluster

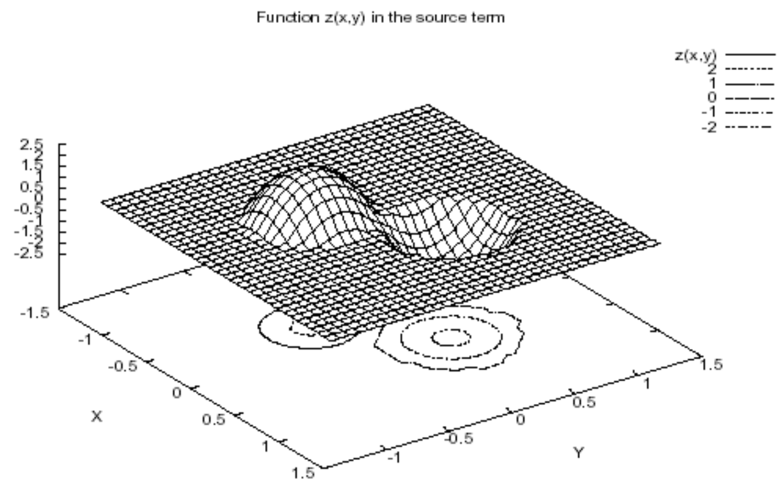
- Solve 2D problem on a Laptop using equilibrium scheme
- Solve the same one dimensional problem on a PC Cluster using standard scheme
- Compare:
- Computing time
- Accuracy

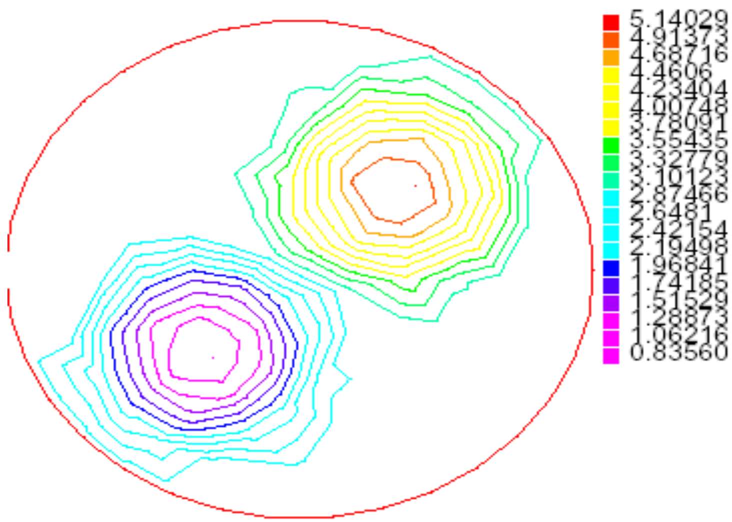
- Laptop data:
- INTEL P3 800Mhz
- 128MB RAM

- PC Cluster data:
- INTEL P4 Foster XEON 2Ghz
- 2GB RDRAM

Problem 1, Laptop vs PC Cluster

Source and ET solution





Problem 1, Laptop vs PC Cluster, Results

Laptop

- 98 nodal points
- Computing time 25sec.,
- Error 8.9E-6

PC Cluster

- 1000 nodal points per CPU
- Computing time

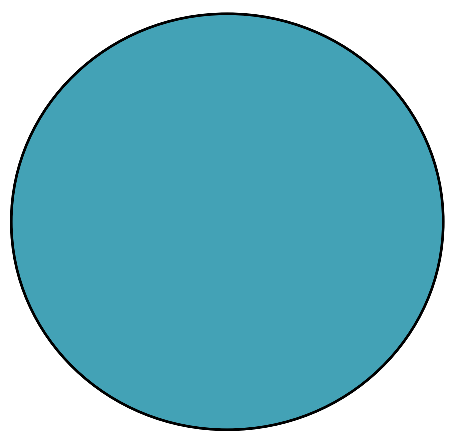
Processors	2	4	8	16
$C-norm$	0.0156	$7.8 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$	$1.9 \cdot 10^{-3}$
L^1-norm	0.0139	$3.8 \cdot 10^{-3}$	$9.8 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$
Computing time(sec.)	32	71	147	512

Problem 1: Conclusion

- Low accuracy of numerical schemes can not be compensated by means of increasing computational power

Problem 2

- Computational domain = unit circle
- Initial value = 0
- Boundary value = 1
- Gouverning equation = inviscid Burgers



Problem 2, "equivalent" form of governing equations

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{2} \frac{\partial u^2/2}{\partial x_i} = 0$$

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{2} u \frac{\partial u}{\partial x_i} = 0$$

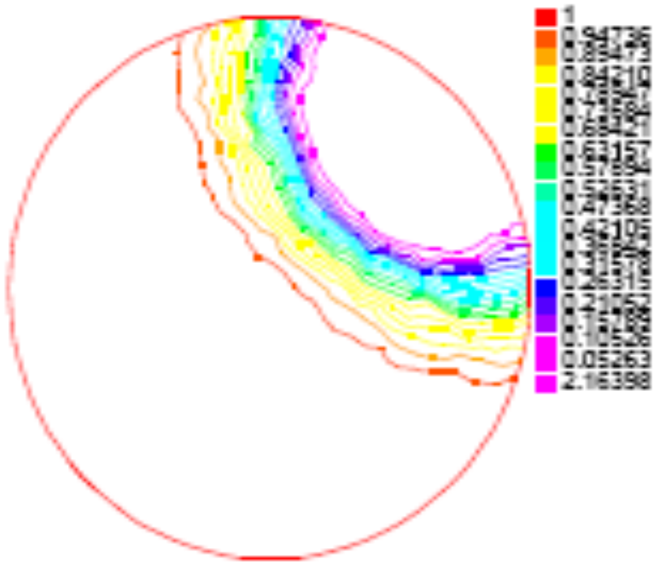
$$v = u^2$$

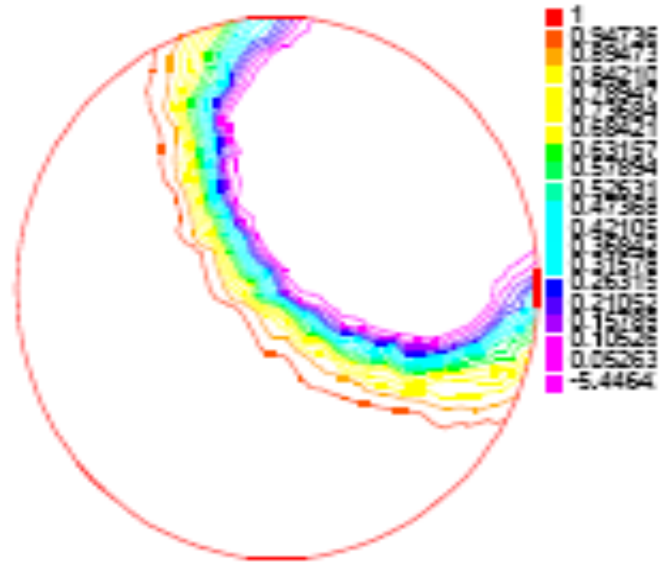
$$\frac{\partial v}{\partial t} + \sum_{i=1}^{2} \frac{\partial 2/3 v^{2/3}}{\partial x_i} = 0$$

Problem 2, theoretical estimate of propagation speed

- Original = 0.5
- Nonconservative = 1
- Change of variables = 1/3

Problem 2, numerical results





Original variable

Transformed variable

Problem 2, analysis

- Scheme Based on Transformation of Variables Produces Wrong Propagation Speed
- Mesh Refinement Does not improve Accuracy of the Scheme Based on Variable Transformation
- Scheme Based on Variable Transformation Converges to Wrong Solution
- Mesh Refinement Improves Accuracy of the Scheme in Original Variables

Analysis

- Non Conservative Scheme Produces Wrong Propagation Speed
- Mesh Refinement Does not improve Accuracy of the Nonconservative Scheme
- Nonconservative Scheme Converges to Wrong Solution
- Mesh Refinement Improves Accuracy of the Conservative Scheme

Problem 2, conclusion

- Smoothness of solution was not taken into account and entropy condition is not valid that lead to wrong solution

↓

- Qualitative properties of exact solution must be respected otherwise numerical models can fail

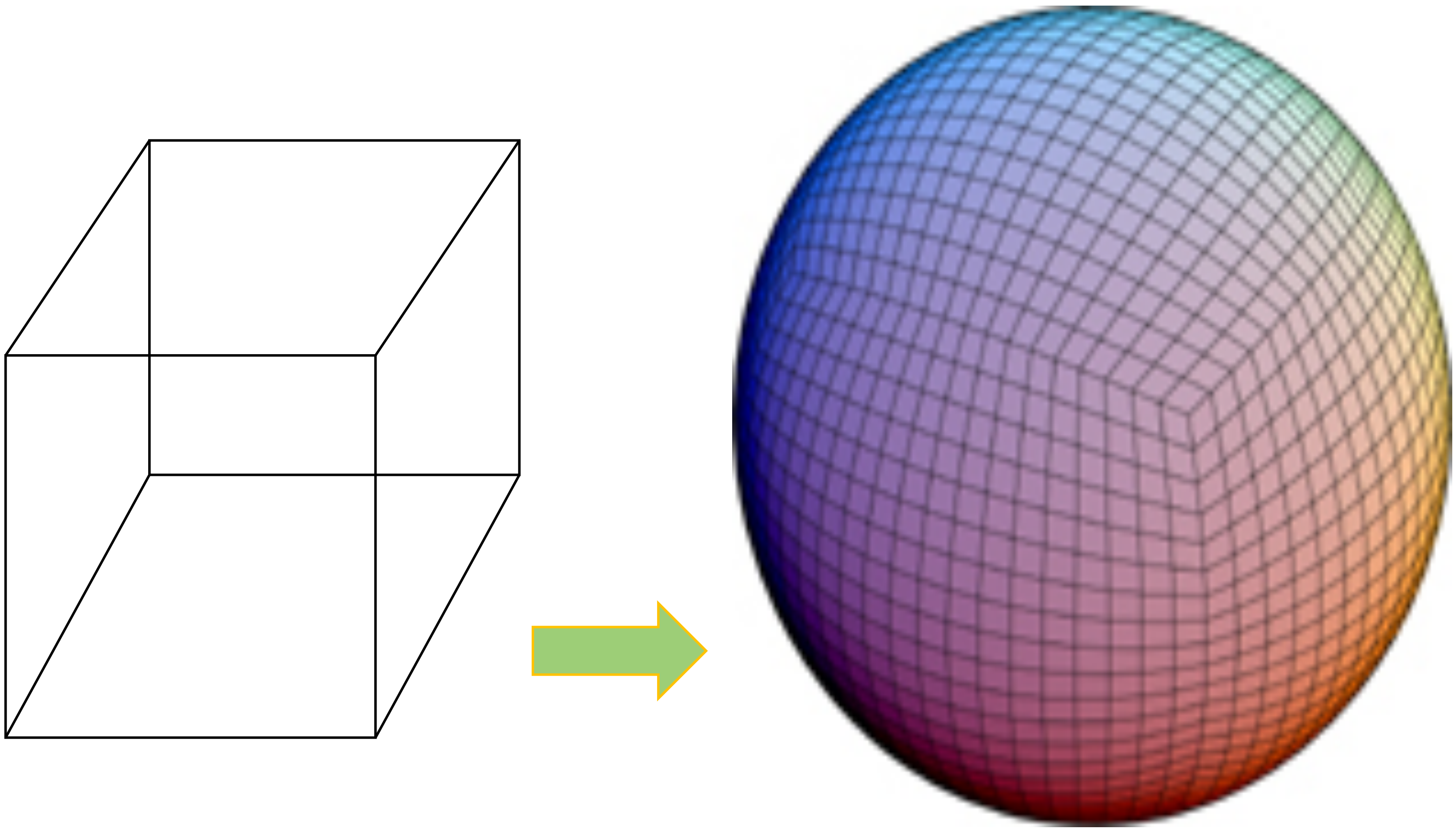
Good Numerical Model

- Produces accurate enough solution at low computational cost
  - computational time
  - storage/memory
  - number of processors

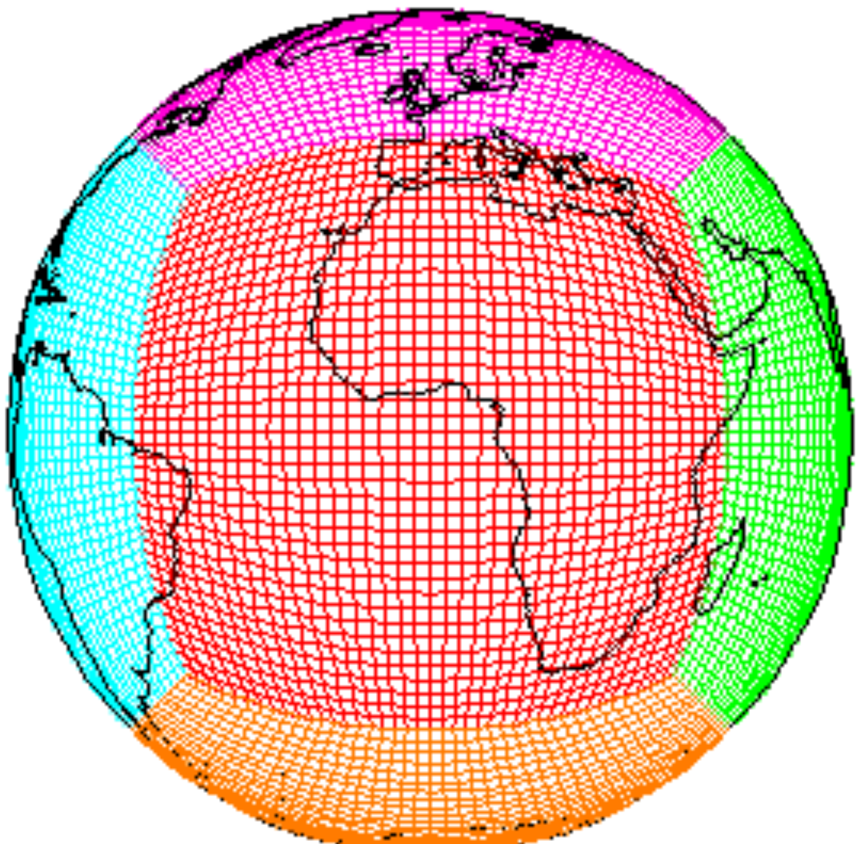
Method of Decomposition

- Universal approach
- Results in efficient methods
- Main idea:
  - reduce complex problem to several sub problems of less complexity
- Example:
  - exp(A)exp(B)=exp(A+B)
- Difficult to implement in complex computational domains

Cubed sphere



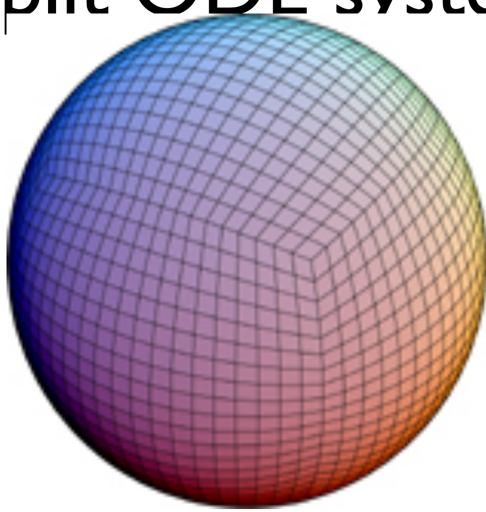
Cubed Sphere

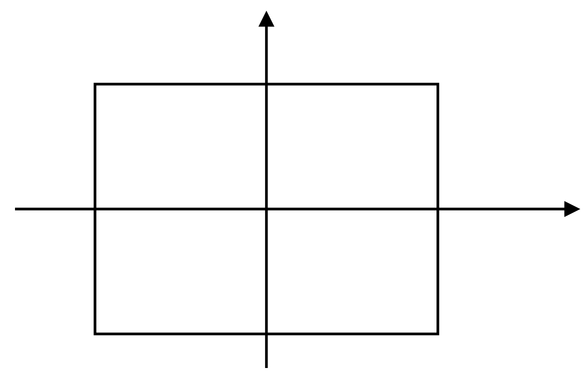


Locally one directional schemes, implicit, large time steps

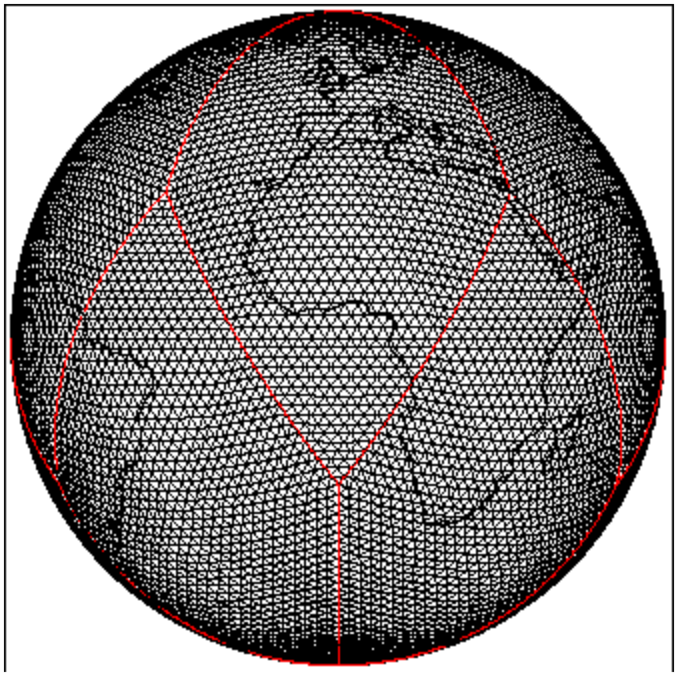
Approach:

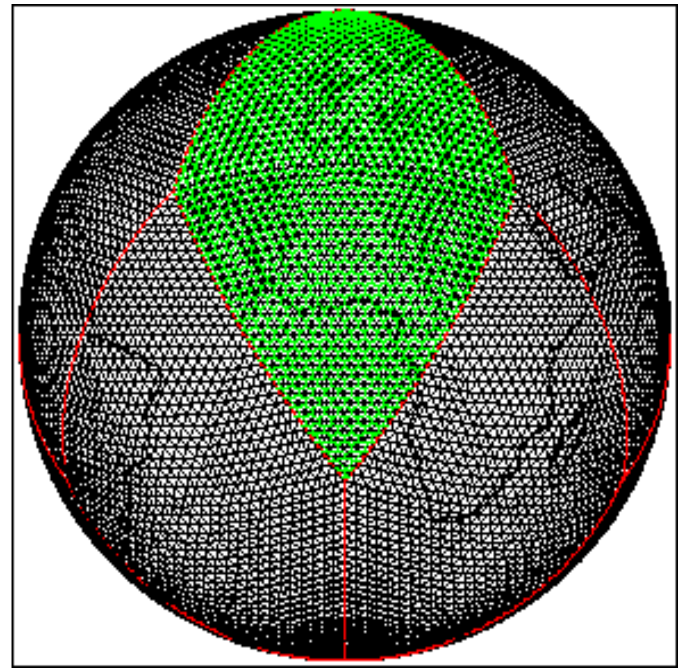
- Derive semidiscrete monotone finite volume scheme in usual way
- Split ODE system interfacewise





Locally one directional schemes for icosahedral hexagonal mesh ?





Numerical schemes

- Explicit
  - ease to implement
  - low computational cost for one time step
  - small time steps
- implicit
  - difficult to implement
  - higher computational cost per time step
  - large time steps
- What approach is more appropriate and when?

Numerical schemes

- First order accuaracy
  - ease to implement
  - low computational cost for one time step
- Higher order of accuracy
  - difficult to implement
  - higher computational cost per time step
- Which order of accuracy is more appropriate and when?

Meshes

- Fine
  - better represents geometry
  - better represents approximate solution
  - contains more nodal points, expensive
- Rough
  - cheaper, sometimes sufficient to produce accurate enough solution
- What kind of mesh is more appropriate and when?

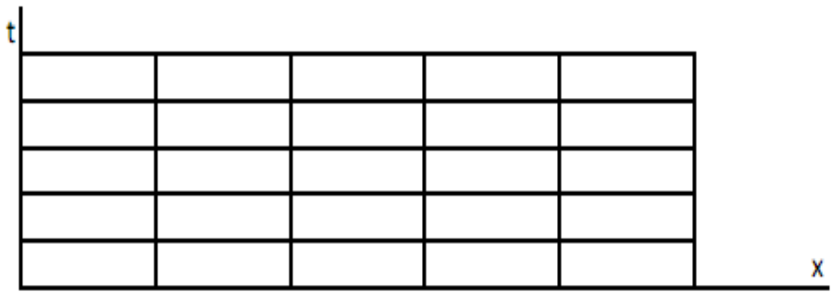
Formulation of coupling problem

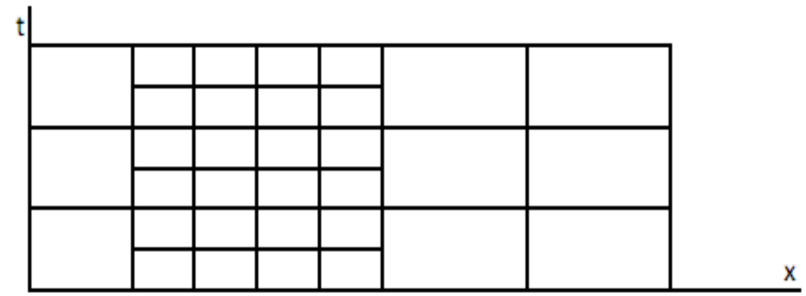
- Accuracy of computation is affected by
  - accuracy of numerical scheme
  - mesh
  - smoothness of exact solution
- Approach
  - use different numerical schemes in one algorithm
  - use different space and time steps in one algorithm
  - use smothness of solution to decide which one (scheme or mesh) to apply

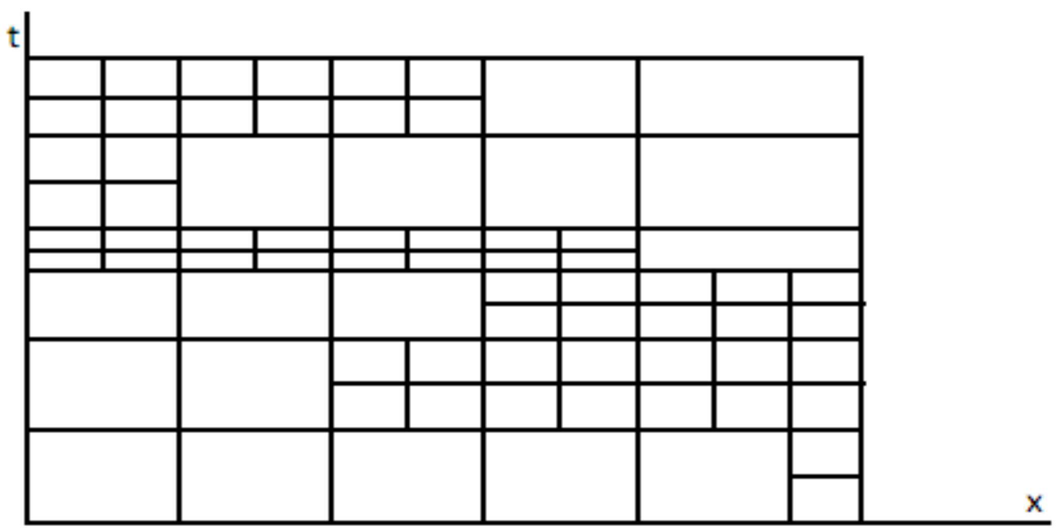
New approach: multischeme

- one mesh with two different schemes
  - incorporate two numerical flux functions in one scheme for computations in one nodal point
- two meshes with one scheme
  - use summing up of numerical flux functions corresponding to smaller time step
- coupling the above two approaches

space-time meshes







hyperbolic conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial \mathrm{A}(u)}{\partial x} = 0$$

$$u(0,x) = u_0(x), u_0 \in L^\infty(\Re) \cap BV(\Re)$$

$$\mathrm{A}(u) \in C^1(\Re)$$

$$u(t,x) : \Re^+ \times \Re \longrightarrow \Re$$

Entropy condition

- Smooth solution does not exist in general
- Weak solution is not unique
- Entropy condition ensures uniqueness

$$\iint_{\Re_t^+ \Re} \left(\eta(u) g_t + q(u) g_x \right) dx dt \geq 0$$

$$g \in C_0^1 \left(\Re_t^+ \times \Re \right), g \geq 0$$

$$q'(u) = \eta'(u) \mathrm{A}'(u)$$

monotone schemes

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{A\left(u_{j+1}^n, u_j^n\right) - A\left(u_j^n, u_{j-1}^n\right)}{h} = 0$$

$$u_j^0 = \frac{1}{h} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_0(x)\,dx$$

- Numerical flux function ***A(u,v)***
  - consistency ***A(u,u)*** =A(u)
  - monotonicity
  - smoothness

cfl condition and convergence

$$\frac{\tau}{h}\left(\max_{|u|,|v|\le|u_0|_{L^\infty}} |A_u(u,v)| + \max_{|u|,|v|\le|u_0|_{L^\infty}} |A_v(u,v)|\right) \le 1$$

- Apriori estimates:
  - uniform bound on approximate solutions
  - uniform bound on total variation
  - discrete cell entropy inequality
- Compactness of approximate solutions
- Convergence
  - limit of the discrete entropy inequality
  - uniqueness of entropy solution

multischeme

$$\frac{u_{21}^{Ln+l+1} - u_{21}^{Ln+l}}{\tau_2} + \frac{A_2\left(u_{22}^{Ln+l}, u_{21}^{Ln+l}\right) - A_2\left(u_{21}^{Ln+l}, u_{1m}^{n}\right)}{h_2} = 0,$$

$$l = \overline{0, L-1}$$

$$\frac{u_{1m}^{n+1} - u_{1m}^{n}}{\tau_1} + \frac{\frac{1}{L}\sum_{l=0}^{L-1} A_2\left(u_{21}^{Ln+l}, u_{1m}^{n}\right) - A_1\left(u_{1m}^{n}, u_{1m-1}^{n}\right)}{h_1} = 0$$

convergence theorem

$$\frac{\tau_k}{h_k}\left(\max_{|u|,|v|\le|u_0|_{L^\infty}}\{|A_{iu}(u,v)|, i=\overline{1,s}\} + \max_{|u|,|v|\le|u_0|_{L^\infty}}\{|A_{iv}(u,v)|, i=1,s\}\right)\le 1, k=\overline{1,s}$$

Theorem. *Approximate solutions constructed by multischeme algorithm with fixed maximum L levels of refinement converge almost everywhere to the unique entropy solution of (1),(2) as h→0 under supposition that CFL condition (24) is valid.*

Refinement and derefinement

Refinement and derefinement

h_init - initial step of nonuniform mesh;
h_uni - step of uniform mesh, is equal to minimal step of nonuniform mesh
k - when $(u_j - u_{j+1}) > Kmax*h$, h decrease 2-times and saves the value during k+1 steps (hmin=h_init/10)
k - when $(u_j - u_{j+1}) < Kmin*h$, h increase 2-times and saves the value during k+1 steps (hmax=3*h_init)

Example

interval [-1,1]   h_init=0.1   Kmin=0.3   Kmax=0.5   k=3   h_uni = 0.0125

Comparison with uniform mesh

N_uni=2/0.0125=160; N_nonuni=49; difference=111;

N_uni/N_nonuni ≈3.265 (number of nodes in uniform mesh is appr. 3-times) , number of nodes in uniform mesh compared to nonuniform mesh increases appr. 226.5 %
((N_uni- N_nonuni)/ N_nonuni ≈2.265).

Refinement and derefinement

Example

interval `[-1,1]` h_init=0.1 Kmin=1 Kmax=1 `k=3` h_uni = 0.0125

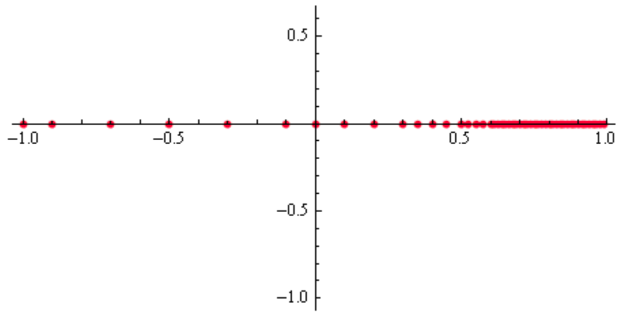
Comparison with uniform mesh

N_uni =2/0.0125=160; N_nonuni=34; difference =126;

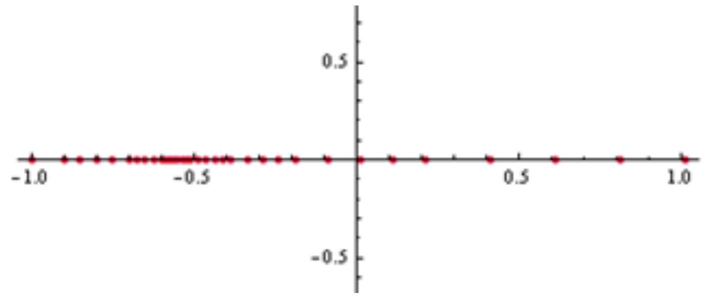
N_uni/N_nonuni ≈ 4.71 (number of nodes in uniform mesh is appr. in 5-times), number of nodes in uniform mesh compared to nonuniform mesh increases appr. on 371 %- ((N_uni- N_nonuni)/ N_nonuni ≈3.71).

$$f(x) = \begin{cases} 1, \quad x < -0.5 \\ 2\cos\left(20x - \frac{1}{x^4}\right) + x, \quad x > 0.5 \\ \sin(x), \quad -0.5 \le x \le 0.5 \end{cases}$$

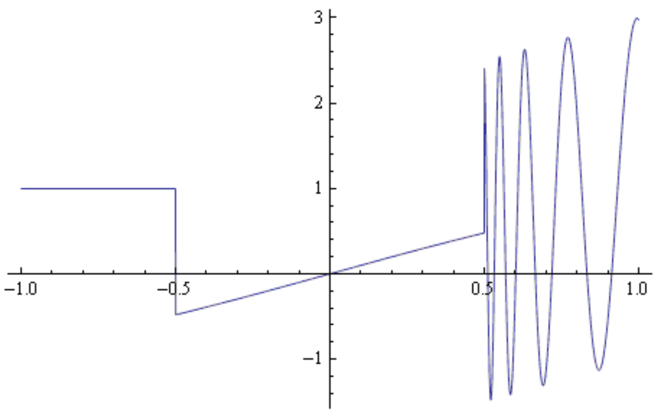
$$g(x) = \begin{cases} 1, \quad x > 0.5 \\ 2\cos\left(20x - \frac{1}{x^4}\right) + x, \quad x < -0.5 \\ \sin(x), \quad -0.5 \le x \le 0.5 \end{cases}$$



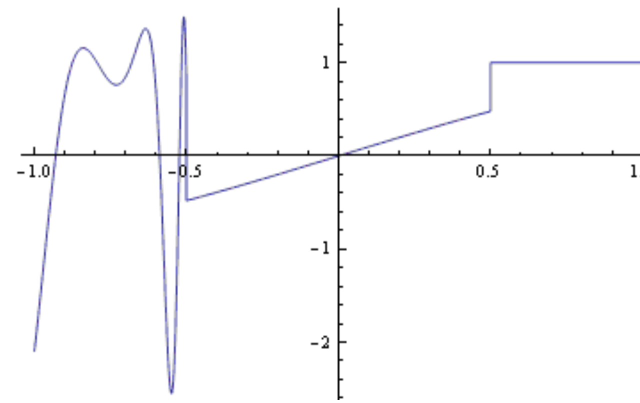
nonuniform mesh



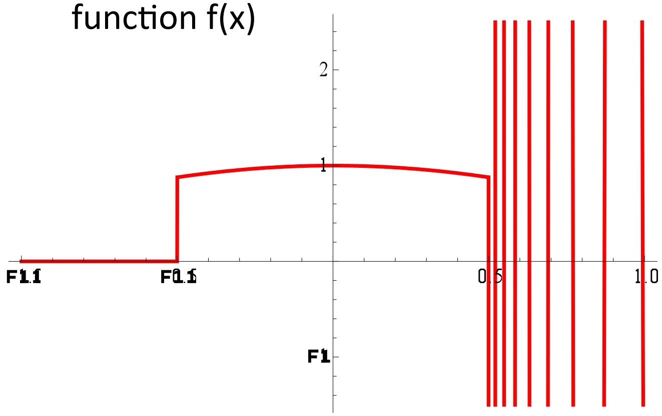
nonuniform mesh



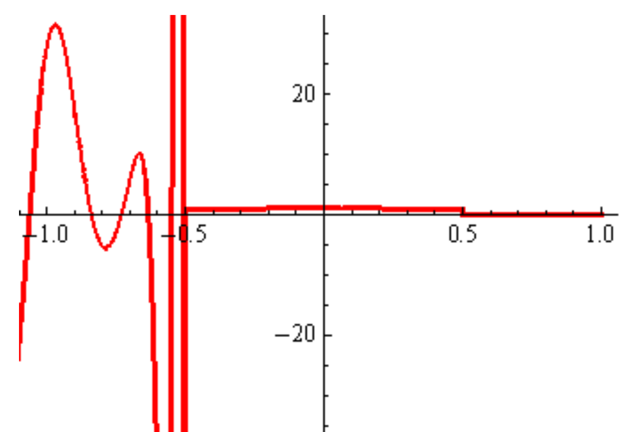
function f(x)



function g(x)



function f′(x)
(amplitude is about 130)



function g′(x)
(amplitude is about 150)

From simple equations to real models

- Systems of conservation laws
  - numerical flux formulation

- Several space dimensions
  - Cartesian meshes, triangular meshes, arbitrary finite volume
  - summing up numerical fluxes corresponding to smaller time steps along interfaces between different meshes

Possible applications

- Problems formulated in the form of mathematical models : partial differential equations, conservation laws
- Solution of the problem on some subdomains changes very fast while on other subdomains – slowly
- Application of the uniform mesh is not suitable, large nbr of unknowns, computational cost

Possible applications:

need for numerical scheme

- high order for rough meshes and first order for fine meshes
- explicit scheme for large space steps and implicit scheme for small space steps
- nonconservative formulation – multischeme for source terms
- from multischeme to multimodel: coupling of different different models via flux function?

Related problems

- What if initial datum has no bounded variation?
- What kind of compactness arguments can be used in several space dimensions?
- Data structure?
- Parallelization – load balancing

Thank you very much for your attention