Network/Grid Computing: *Modeling, Algorithm, and Software*

Mo Mu
Joint work with Xiao Hong Zhu, Falcon Siu

Department of Mathematics
Hong Kong University of Science and Technology

•IFIP WG2.5 Workshop on "*The Changing Face of Mathematical Software*", D.C., June 3-4, 2004
•China-Sweden Workshop on *Numerical PDEs*, Beijing, June7-9, 2004

Network/Grid Computing

vs. Conventional Parallel/Distributed Computing

- Resource sharing
  - Hardware
  - Software
  - Data
  - Collaborative research/computing
- Remote vs. local computing nodes
- Service-oriented vs. client-oriented
- Communication

- More resources
- More flexibility
- More loosely coupled
  - System structures
  - Data structures
  - Models and algorithms
- More uncertainties
- Less user control

Challenges

- Software integration
  - Multi-languages
  - Multi-sources
  - Multi-nodes
- Collaborative computing
  - Multi-clients
  - Multi-models
  - Multi-methods
  - Multi-data

Scientific Computing with Multi-modeling

- Interaction of standing shock with boundary layer
- Inviscid-viscous flows
- Compressible-incompressible flows
- Turbulent-laminar flows
- Interface stability with different media
- Composite materials
- Complex systems

Modeling in Network/Grid Computing

- Different models in local regimes
- Interface coupling conditions
- Complexity across the interfaces
  - Physical
    - Discontinuity
    - Boundary layer
  - Geometrical
    - Topology
    - Moving interfaces

Applications

- Originated from a underlying problem where a global model approximation might be inappropriate, or even not be applicable– physically, mathematically, or computationally
- Reduced from a underlying global model
  - Computational efficiency
  - Approximation accuracy
  - Stiffness
  - Domain decomposition

Features of New Computational Frameworks

- Network/Grid computing (service-based)
  - Loosely coupled
  - Less central control
  - Local solvers with mature methods and existing codes
  - Less local information exposed
- Scientifically
  - Modeling complex physical systems
  - Sharp resolution of interface structures
  - More accurate and efficient in some cases

1-D, Scalar, Linear Hybrid Model
(1) Inflow on Γ

- Local models (boundary layer problem with small viscosity)

$$\begin{cases} v_t + a v_x = 0, & x < 0 \\ u_t + b u_x = \nu u_{xx}, & x > 0 \end{cases}$$

$$\nu > 0, \quad ab > 0$$

- Interface condition

$$\begin{cases} (bu - \nu u_x)\,|_\Gamma = (av)\,|_\Gamma, & \text{(flux continuity)} \\ av\,|_\Gamma = bu\,|_\Gamma, \quad \text{if } a < 0, & \text{(inflow and solution continuity)} \end{cases}$$

$$\Rightarrow u_x\,|_\Gamma = 0$$

- Boundary conditions

$$u(t,\tau) = u_\tau \text{ (boundary layer with thickness } \tau)$$

- Initial condition consistent with the boundary and interface conditions
- Fully decoupled: Viscous->inviscid

1-D, Scalar, Linear Hybrid Model (2) Outflow on Γ

- Local models: Same as before, but with a>0, b>0
- Interface condition

$$(bu - \nu u_x)|_\Gamma = (av)|_\Gamma, \qquad \text{(flux continuity)}$$

- Boundary conditions

$$v(t,-1) = v_a \ \text{(inflow)}$$

$$u(t,\tau) = 0 \ \text{(boundary layer with thickness } \tau\text{)}$$

- Fully decoupled: inviscid -> viscous

Steady State
(Outflow on Γ)

- Exact solution

$$v(x) = v_a, x \le 0$$

$$u(x) = \tfrac{a}{b} v|_{\Gamma} (1 - e^{-\frac{b(x-\tau)}{v}}), x \ge 0$$

- Boundary layer

$$\tau = O(v^{\alpha}), 0 < \alpha < 1$$

- Discontinuity at the interface

$$u|_{\Gamma} = \tfrac{a}{b} v|_{\Gamma} (1 - e^{-\frac{b}{v^{(1-\alpha)}}})$$

$$\approx \tfrac{a}{b} v|_{\Gamma}$$

Algorithm

- Inviscid solver
  - Upwind scheme
  - Explicit computation
- Viscous solver
  - Central difference plus upwind for the elliptic operator
  - Forward difference for interface condition with input from the inviscid solver
- Fully decoupled computation for different models
- **Cheap** inviscid computation with “large” spacing
- **Sharp** boundary layer structure with few grid points

Extensions

- HDD of D-N type (Quateroni, et al)
  - Linear
    - 2-D, steady state or unsteady state with continuous solutions
    - Discontinuous solutions?
  - Nonlinear?
    - Linearization+HDD

Hybrid Burgers Equations

- Simplified 1-D model for standing shock-boundary layer interaction
  Conservation laws
$$\begin{cases} u_t + (\frac{1}{2}u^2)_x = 0 \\ v_t + (\frac{1}{2}v^2)_x = \varepsilon v_{xx}, \ \ \varepsilon > 0, \end{cases}$$
- Boundary condition: $u(x_l) = 1$, $v(x_r) = -1$
- Interface conditions

$$\begin{cases} \frac{1}{2}u^2 \,|_\Gamma = (\frac{1}{2}v^2 - \varepsilon v_x)\,|_\Gamma, & \text{(flux continuity)} \\ \frac{1}{2}u^2 \,|_\Gamma = \frac{1}{2}v^2 \,|_\Gamma, & \text{(Rankine-Hugoniot condition)} \end{cases}$$

- Vs. traditional domain decomposition framework
  - Local models
  - Interface condition
- Fully decoupled in special cases such as 1-D steady state

Framework of Interface Relaxation

- Select interface variables, say $\mathbf{W} = \begin{bmatrix} \mathbf{u}|_\Gamma \\ \mathbf{v}|_\Gamma \end{bmatrix}$
- **W** is the solution implicitly defined by the interface conditions **F**(**W**) = 0
- Evaluation of **F(W**) usually involves local PDE solvers
- Relaxation with the approximation of W at the interface (preconditioned "R-F" or residual correction on interface)

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} - \mathbf{PF}(\mathbf{W}^{n})|_\Gamma, n = 0,1,2,\ldots,$$

where P is a preconditioner

Local solver 1: Inviscid Solver

- Conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \qquad f(u) = \frac{1}{2}u^2$$

- Godunov Scheme
  - Riemann solver for each pair of adjacent cells (modified with the last cell at the interface)

$$\begin{cases} u_t + (f(u))_x = 0, \quad x_{j-\frac{1}{2}} < x < x_{j+\frac{3}{2}}, \quad t^n \le t \le t^{n+1} \\ u(x,0) = u_0(x) = \begin{cases} U_l = U_j^n, & \text{if } x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}} \\ U_r = U_{j+1}^n, & \text{if } x_{j+\frac{1}{2}} < x < x_{j+\frac{3}{2}} \end{cases} \end{cases}$$

  - Cell average update locally on the current time level

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}[f(u_{RP}(x_{j+\frac{1}{2}}, t)) - f(u_{RP}(x_{j-\frac{1}{2}}, t))]$$

Local Solver 2: Viscous Solver

- Viscous flow model

$$v_t + (\frac{1}{2}v^2)_x = \varepsilon v_{xx}, \varepsilon > 0, \text{ where } f(v) = \frac{1}{2}v^2 - \varepsilon v_x$$

- Explicit scheme
- Spatial discretization:
  - convection term: upwind
  - viscous term: central finite difference
- Update local solution on the current time level

Relaxation with Newton-type Preconditioner

- Interface condition

$$\begin{cases} \frac{1}{2}{u_\Gamma}^2 - (\frac{1}{2}{v_\Gamma}^2 - \varepsilon \frac{v_1 - v_\Gamma}{h_r}) = 0, & \text{(flux continuity)} \\ \frac{1}{2}{u_\Gamma}^2 - \frac{1}{2}{v_\Gamma}^2 = 0, & \text{(Rankine - Hugoniot condition)} \end{cases}$$

- Relaxation with "Newton iteration"
  - P = $J^{-1}(W^{(n)})$

$$\begin{bmatrix} u_\Gamma \\ v_\Gamma \end{bmatrix}^{(n+1)} = \begin{bmatrix} u_\Gamma \\ v_\Gamma \end{bmatrix}^{(n)} + \begin{bmatrix} \frac{1}{2} \frac{[({v_\Gamma}^{(n)})^2 - ({u_\Gamma}^{(n)})^2] + 2{v_\Gamma}^{(n)}(v_1 - {v_\Gamma}^{(n)})}{{u_\Gamma}^{(n)}} \\ v_1 - {v_\Gamma}^{(n)} \end{bmatrix}$$

Relaxation with Newton-type Preconditioner

- Interface condition

$$\begin{cases} \frac{1}{2}{u_\Gamma}^2 - (\frac{1}{2}{v_\Gamma}^2 - \varepsilon\frac{v_1 - v_\Gamma}{h_r}) = 0, & \text{(flux continuity)} \\ \frac{1}{2}{u_\Gamma}^2 - \frac{1}{2}{v_\Gamma}^2 = 0, & \text{(Rankine-Hugoniot condition)} \end{cases}$$

- Relaxation with “Newton iteration”

$$\begin{bmatrix} u_\Gamma \\ v_\Gamma \end{bmatrix}^{(n+1)} = \begin{bmatrix} u_\Gamma \\ v_\Gamma \end{bmatrix}^{(n)} - PF(\begin{bmatrix} u_\Gamma \\ v_\Gamma \end{bmatrix}^{(n)})$$

- P = $J^{-1}(W^{(n)})$

Numerical Experiments

- Boundary condition:

  Ua = 1, Vb = -1

- Unique solution of the steady state
  - Continuous locally
  - Standing shock at the interface

    U(x) = 1, V(x) = -1

- Initial conditions: Set A and Set B

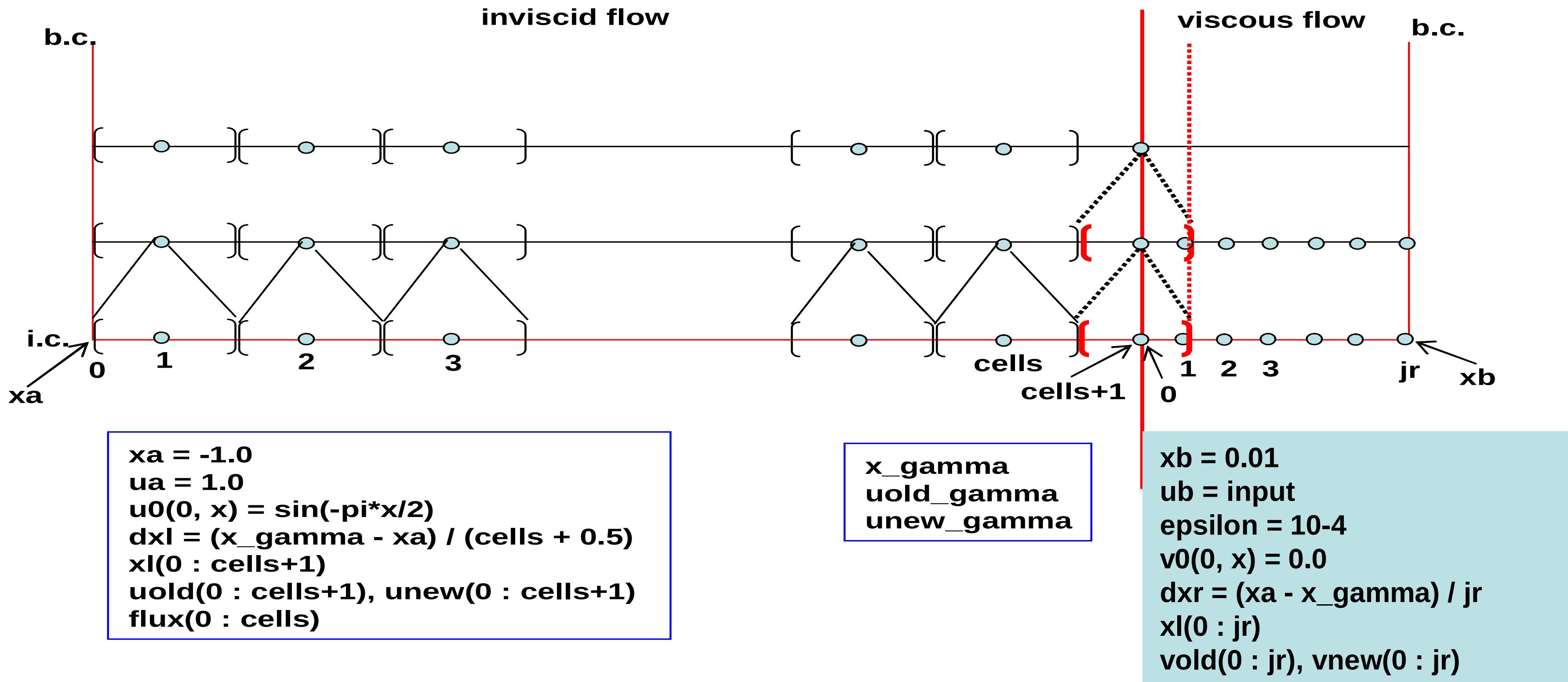
$$\begin{cases} u_0(x) = 1.0, & -1.0 \le x \le 0.0 \\ v_0(x) = -200x + 1.0, & 0.0 \le x \le 0.01 \end{cases}$$

$$\begin{cases} u_0(x) = \sin(-\frac{\pi x}{2}), & -1.0 \le x \le 0.0 \\ v_0(x) = -50x - 0.5, & 0.0 \le x \le 0.01 \end{cases}$$

Set A time step = 24979	Set B time step = 601
Inviscid flow part	**Inviscid flow part**
unew(0) = 1.00000000	unew(0) = 1.00000000
unew(1) = 1.00000000	unew(1) = 1.00000000
unew(2) = 1.00000000	unew(2) = 1.00000000
unew(3) = 1.00000000	unew(3) = 1.00000000
unew(4) = 1.00000000	unew(4) = 1.00000000
unew(5) = 1.00000000	unew(5) = 1.00000000
unew(6) = 1.00000000	unew(6) = 1.00000000
unew(7) = 1.00000000	unew(7) = 1.00000000
unew(8) = 1.00000000	unew(8) = 1.00000000
unew(9) = 1.00000000	unew(9) = 1.00000000
unew(10) = 1.00000000	unew(10) = 1.00000000
unew(11) = 1.00000000	unew(11) = 1.00000000
unew(12) = 1.00000000	unew(12) = 1.00000000
unew(13) = 1.00000000	unew(13) = 1.00000000
unew(14) = 1.00000000	unew(14) = 1.00000000
unew(15) = 1.00000000	unew(15) = .99999999
unew(16) = 1.00000000	unew(16) = .99999997
unew(17) = 1.00000000	unew(17) = .99999988
unew(18) = 1.00000000	unew(18) = .99999948
unew(19) = 1.00000000	unew(19) = .99999765
unew(20) = 1.00000000	unew(20) = .99998921
unew(21) = .99999922	unew(21) = 1.00000000
Viscous flow part	**Viscous flow part**
vnew(0) = -.99999922	vnew(0) = -1.00000000
vnew(1) = -.99999922	vnew(1) = -1.00000000
vnew(2) = -.99999964	vnew(2) = -1.00000000
vnew(3) = -.99999987	vnew(3) = -1.00000000
vnew(4) = -.99999996	vnew(4) = -1.00000000
vnew(5) = -.99999999	vnew(5) = -1.00000000
vnew(6) = -1.00000000	vnew(6) = -1.00000000
vnew(7) = -1.00000000	vnew(7) = -1.00000000
vnew(8) = -1.00000000	vnew(8) = -1.00000000
vnew(9) = -1.00000000	vnew(9) = -1.00000000
vnew(10) = -1.00000000	vnew(10) = -1.00000000

Hybrid Burgers Equation

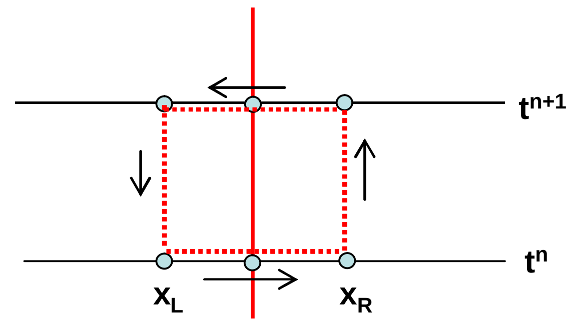
- Modified HMM (Engquist, E)
- Interface Relaxation
  - Local solvers: (Inviscid, Viscous)
  - Interface Relaxer: Finite volume



Interface Treatment

- Finite volume

$$U_\Gamma^{n+1} = U_\Gamma^n - \frac{\Delta t}{(x_R - x_L)}\left(f_R^{n+\frac{1}{2}} - f_L^{n+\frac{1}{2}}\right)$$



- Information from inviscid solver

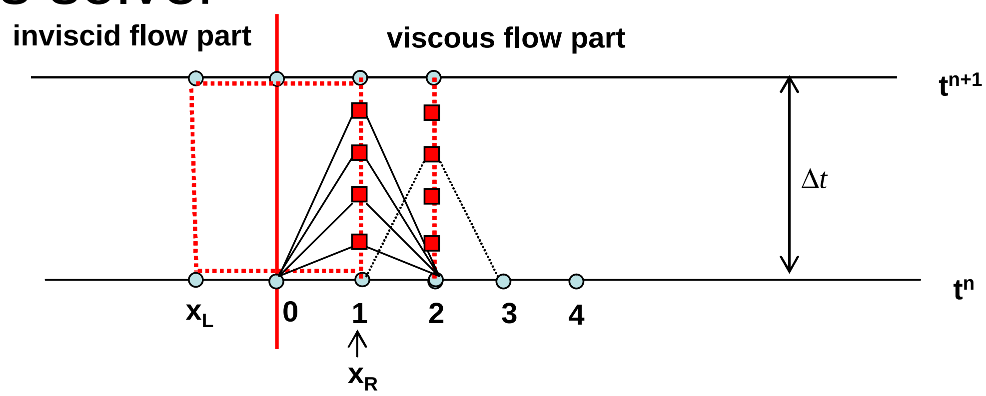
$$f_L^{n+\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u_{RP}(x_L,t))dt$$

$$= \frac{1}{\Delta t} f(u_{RP}(x_L,t^{n+1}))\Delta t \quad \text{(since } u_{RP}(x_L,t) \text{ is constant at } t^n \le t \le t^{n+1}\text{)}$$

$$= f(u_{RP}(x_L,t^{n+1}))$$

- Information from viscous solver

$$f_R^{n+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \tilde{f}(v(x_R,t))dt$$



Set A dt = 0.0004, time step = 73

Inviscid flow part

```
unew(   0 ) =        1.000000
unew(   1 ) =        1.000000
unew(   2 ) =        1.000000
unew(   3 ) =        1.000000
unew(   4 ) =        1.000000
unew(   5 ) =        1.000000
unew(   6 ) =        1.000000
unew(   7 ) =        1.000000
unew(   8 ) =        1.000000
unew(   9 ) =        1.000000
unew(  10 ) =        1.000000
unew(  11 ) =        1.000000
unew(  12 ) =        1.000000
unew(  13 ) =        1.000000
unew(  14 ) =        1.000000
unew(  15 ) =        1.000000
unew(  16 ) =        1.000000
unew(  17 ) =        1.000000
unew(  18 ) =        1.000000
unew(  19 ) =        1.000000
unew(  20 ) =        1.000000
unew(  21 ) =        1.000001
```

Viscous flow part

```
vnew(   0 ) =        1.000001
vnew(   1 ) =         .999990
vnew(   2 ) =         .999866
vnew(   3 ) =         .998501
vnew(   4 ) =         .983498
vnew(   5 ) =         .819807
vnew(   6 ) =        -.819817
vnew(   7 ) =        -.983507
vnew(   8 ) =        -.998511
vnew(   9 ) =        -.999876
vnew(  10 ) =       -1.000000
```

Set 2	dt = 0.0004

```
time step   =        406
```

Inviscid flow part

```
unew(   0 ) =      1.00000000
unew(   1 ) =      1.00000000
unew(   2 ) =      1.00000000
unew(   3 ) =      1.00000000
unew(   4 ) =      1.00000000
unew(   5 ) =      1.00000000
unew(   6 ) =      1.00000000
unew(   7 ) =       .99999999
unew(   8 ) =       .99999996
unew(   9 ) =       .99999976
unew(  10 ) =       .99999879
unew(  11 ) =       .99999433
unew(  12 ) =      -.36233667
unew(  13 ) =      -.99999925
unew(  14 ) =      -.99999989
unew(  15 ) =      -.99999999
unew(  16 ) =     -1.00000000
unew(  17 ) =     -1.00000000
unew(  18 ) =     -1.00000000
unew(  19 ) =     -1.00000000
unew(  20 ) =     -1.00000000
unew(  21 ) =     -1.00000000
```

Viscous flow part

```
vnew(   0 ) =     -1.00000000
vnew(   1 ) =     -1.00000000
vnew(   2 ) =     -1.00000000
vnew(   3 ) =     -1.00000000
vnew(   4 ) =     -1.00000000
vnew(   5 ) =     -1.00000000
vnew(   6 ) =     -1.00000000
vnew(   7 ) =     -1.00000000
vnew(   8 ) =     -1.00000000
vnew(   9 ) =     -1.00000000
vnew(  10 ) =     -1.00000000
```

Computational Components of Interface Relaxation

- A collection of fully decoupled local solvers
  - Provided by whatever best services available on grid nodes
  - With their own
    - Geometric domains
    - Physical models
    - Computational methods

Computational Components of Interface Relaxation (cont'd)

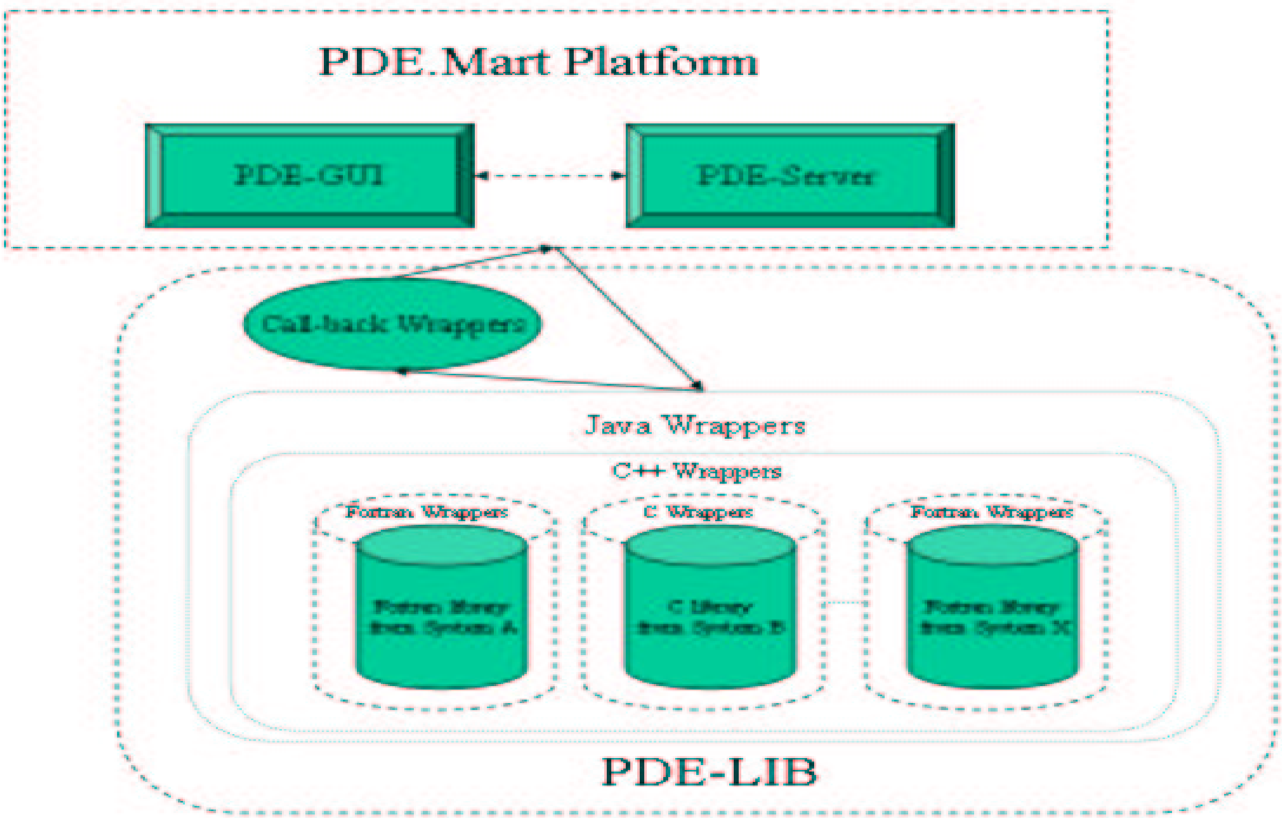
- Interface relaxer
  - Provided by the client or host server
  - Limited development and computational overhead
  - Global coupling and central control
    - Invoke local solvers with the available interface information
    - Collect the computed boundary information from local solvers
    - Update interface information with preconditioned residual correction or alike to relax the interface constraints
    - Iterate until the interface conditions are all satisfied

Changing Face of Mathematical Software

To cope with the changes in

- Computing platforms
- Scientific modeling
- Computational methodologies

System Structure



PDE.Mart

- Network-based PSE à Grid computing
- Client-server protocol
  - Building relaxer
  - Distributing and deploying local solvers
- Object-oriented Java platform
  - PDE solving objects: Domain, PDE, Mesh, Discrete, Indexing, Solution
  - Plug-and-play
    - Local solver composition
    - Computational steering
- Software integration with native codes from multi-sources and in multi-languages
  - PDE-API
  - API-based wrapper framework