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, June 7-9, 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 $\Gamma$

- Local models (boundary layer problem with small viscosity)
  \[
  \begin{align*}
  v_t + av_x &= 0, \quad x < 0 \\
  u_t + bu_x &= \nu u_{xx}, \quad x > 0 \\
  \nu &> 0, \quad ab > 0
  \end{align*}
  \]

- Interface condition
  \[
  \begin{align*}
  (bu - \nu u_x) |_{\Gamma} &= (av) |_{\Gamma}, \quad \text{(flux continuity)} \\
  av |_{\Gamma} &= bu |_{\Gamma}, \quad \text{if } a < 0, \quad \text{(inflow and solution continuity)} \\
  \Rightarrow u_x |_{\Gamma} &= 0
  \end{align*}
  \]

- Boundary conditions
  \[
  u(t, \tau) = u_{\tau} \quad \text{(boundary layer with thickness } \tau)\]

- Initial condition consistent with the boundary and interface conditions
- Fully decoupled: Viscous-$\rightarrow$inviscid
1-D, Scalar, Linear Hybrid Model

(2) Outflow on $\Gamma$

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

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

- Boundary conditions

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

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

- Fully decoupled: inviscid -> viscous
Steady State
(Outflow on $\Gamma$)

- **Exact solution**
  \[ v(x) = v_a, x \leq 0 \]
  \[ u(x) = \frac{a}{b} v \mid_\Gamma (1 - e^{-\frac{b(x - \tau)}{v}}), x \geq 0 \]

- **Boundary layer**
  \[ \tau = O(v^\alpha), 0 < \alpha < 1 \]

- **Discontinuity at the interface**
  \[ u \mid_\Gamma = \frac{a}{b} v \mid_\Gamma (1 - e^{-\frac{b}{v(1-\alpha)}}) \]
  \[ \approx \frac{a}{b} v \mid_\Gamma \]
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
  \[ u_t + \left( \frac{1}{2} u^2 \right)_x = 0 \]
  \[ v_t + \left( \frac{1}{2} v^2 \right)_x = \varepsilon v_{xx}, \quad \varepsilon > 0, \]
- Boundary condition: \( u(x_l) = 1, \ v(x_r) = -1 \)
- Interface conditions
  \[
  \begin{cases}
  \left. \frac{1}{2} u^2 \right|_\Gamma = \left. \left( \frac{1}{2} v^2 - \varepsilon v_x \right) \right|_\Gamma, & \text{(flux continuity)} \\
  \left. \frac{1}{2} u^2 \right|_\Gamma = \left. \frac{1}{2} v^2 \right|_\Gamma, & \text{(Rankine - Hugoniot condition)}
  \end{cases}
  \]
- 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 $W = \begin{bmatrix} u |_\Gamma \\ v |_\Gamma \end{bmatrix}$
- $W$ 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)
  \[ W^{n+1} = W^n - PF(W^n)|_\Gamma , n = 0,1,2,..., \]
  where $P$ is a preconditioner
Local solver 1: Inviscid Solver

- Conservation law
  \[
  \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0
  \]
  \[
  f(u) = \frac{1}{2} u^2
  \]

- Godunov Scheme
  - Riemann solver for each pair of adjacent cells
    (modified with the last cell at the interface)
    \[
    \begin{cases}
    u_t + (f(u))_x = 0, & x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}}, \quad t^n \leq t \leq t^{n+1} \\
    u(x,0) = u_0(x) = \begin{cases}
    U_l = U^n_j, & \text{if } x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}} \\
    U_r = U^n_{j+1}, & \text{if } x_{j+\frac{1}{2}} < x < x_{j+\frac{3}{2}}
    \end{cases}
    \end{cases}
    \]
  - Cell average update locally on the current time level
    \[
    U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x} \left[ f(u_{RP}(x_{j+\frac{1}{2}},t)) - f(u_{RP}(x_{j-\frac{1}{2}},t)) \right]
    \]
Local Solver 2: Viscous Solver

- Viscous flow model
  \[ v_t + \left( \frac{1}{2} v^2 \right)_x = \varepsilon v_{xx}, \varepsilon > 0, \text{ where } f(v) = \frac{1}{2} v^2 - \varepsilon v_x \]

- 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

\[
\begin{aligned}
\frac{1}{2} u_\Gamma^2 - \left( \frac{1}{2} v_\Gamma^2 - \varepsilon \frac{v_1 - v_\Gamma}{h_r} \right) &= 0, \quad \text{(flux continuity)} \\
\frac{1}{2} u_\Gamma^2 - \frac{1}{2} v_\Gamma^2 &= 0, \quad \text{(Rankine - Hugoniot condition)}
\end{aligned}
\]

- Relaxation with “Newton iteration”

\[
\begin{aligned}
P &= J^{-1}(W^{(n)}) \\
\begin{bmatrix} u_\Gamma^{(n+1)} \\ v_\Gamma^{(n+1)} \end{bmatrix} &= \begin{bmatrix} u_\Gamma^{(n)} \\ v_\Gamma^{(n)} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} (v_\Gamma^{(n)})^2 - (u_\Gamma^{(n)})^2 + 2v_\Gamma^{(n)}(v_1 - v_\Gamma^{(n)}) \\ v_\Gamma^{(n)}(v_1 - v_\Gamma^{(n)}) \end{bmatrix}
\end{aligned}
\]
Relaxation with Newton-type Preconditioner

- **Interface condition**
  \[
  \begin{cases}
  \frac{1}{2} u_\Gamma^2 - \left( \frac{1}{2} v_\Gamma^2 - \epsilon \frac{v_1 - v_\Gamma}{h_r} \right) = 0, \\
  \frac{1}{2} u_\Gamma^2 - \frac{1}{2} v_\Gamma^2 = 0,
  \end{cases}
  \]
  (flux continuity, Rankine - Hugoniot condition)

- **Relaxation with “Newton iteration”**
  \[
  \begin{bmatrix}
  u_\Gamma \\
  v_\Gamma
  \end{bmatrix}^{(n+1)} = \begin{bmatrix}
  u_\Gamma \\
  v_\Gamma
  \end{bmatrix}^{(n)} - PF\left(\begin{bmatrix}
  u_\Gamma \\
  v_\Gamma
  \end{bmatrix}^{(n)}\right)
  \]

- **P = J^{-1}(W^{(n)})**
Numerical Experiments

• Boundary condition:
  \[ U_a = 1, \ V_b = -1 \]

• 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

\[
\begin{align*}
  u_0(x) &= 1.0, \quad -1.0 \leq x \leq 0.0 \\
  v_0(x) &= -200x + 1.0, \quad 0.0 \leq x \leq 0.01 \\
  u_0(x) &= \sin\left(-\frac{\pi x}{2}\right), \quad -1.0 \leq x \leq 0.0 \\
  v_0(x) &= -50x - 0.5, \quad 0.0 \leq x \leq 0.01
\end{align*}
\]
### Inviscid flow part

<table>
<thead>
<tr>
<th>Set A</th>
<th>time step</th>
<th>24979</th>
</tr>
</thead>
<tbody>
<tr>
<td>unew(0)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(1)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(2)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(3)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(4)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(5)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(6)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(7)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(8)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(9)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(10)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(11)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(12)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(13)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(14)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(15)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(16)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(17)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(18)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(19)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(20)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(21)</td>
<td>0.99999922</td>
<td></td>
</tr>
</tbody>
</table>

### Viscous flow part

<table>
<thead>
<tr>
<th>Set A</th>
<th>time step</th>
<th>24979</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnew(0)</td>
<td>-0.99999922</td>
<td></td>
</tr>
<tr>
<td>vnew(1)</td>
<td>-0.99999922</td>
<td></td>
</tr>
<tr>
<td>vnew(2)</td>
<td>-0.99999964</td>
<td></td>
</tr>
<tr>
<td>vnew(3)</td>
<td>-0.99999987</td>
<td></td>
</tr>
<tr>
<td>vnew(4)</td>
<td>-0.99999996</td>
<td></td>
</tr>
<tr>
<td>vnew(5)</td>
<td>-0.99999999</td>
<td></td>
</tr>
<tr>
<td>vnew(6)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(7)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(8)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(9)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(10)</td>
<td>-1.00000000</td>
<td></td>
</tr>
</tbody>
</table>

### Inviscid flow part

<table>
<thead>
<tr>
<th>Set B</th>
<th>time step</th>
<th>601</th>
</tr>
</thead>
<tbody>
<tr>
<td>unew(0)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(1)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(2)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(3)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(4)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(5)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(6)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(7)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(8)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(9)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(10)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(11)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(12)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(13)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(14)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(15)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(16)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(17)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(18)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(19)</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>unew(20)</td>
<td>0.99998921</td>
<td></td>
</tr>
<tr>
<td>unew(21)</td>
<td>1.00000000</td>
<td></td>
</tr>
</tbody>
</table>

### Viscous flow part

<table>
<thead>
<tr>
<th>Set B</th>
<th>time step</th>
<th>601</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnew(0)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(1)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(2)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(3)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(4)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(5)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(6)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(7)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(8)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(9)</td>
<td>-1.00000000</td>
<td></td>
</tr>
<tr>
<td>vnew(10)</td>
<td>-1.00000000</td>
<td></td>
</tr>
</tbody>
</table>
Hybrid Burgers Equation

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

\[ x_a = -1.0 \]
\[ u_a = 1.0 \]
\[ u0(0, x) = \sin(-\pi x/2) \]
\[ dxl = (x_{\text{gamma}} - x_a) / (\text{cells} + 0.5) \]
\[ xl(0 : \text{cells}+1) \]
\[ u0ld(0 : \text{cells}+1), u\text{new}(0 : \text{cells}+1) \]
\[ \text{flux}(0 : \text{cells}) \]

\[ x_{\text{gamma}} \]
\[ u0ld_{\text{gamma}} \]
\[ u\text{new}_{\text{gamma}} \]

\[ x_b = 0.01 \]
\[ u_b = \text{input} \]
\[ \epsilon = 10^{-4} \]
\[ v0(0, x) = 0.0 \]
\[ dxr = (x_a - x_{\text{gamma}}) / jr \]
\[ xl(0 : jr) \]
\[ v0ld(0 : jr), v\text{new}(0 : jr) \]
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
  \[ f_{L}^{n+\frac{1}{2}} \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(u_{RP}(x_L,t))dt \]
  \[ = \frac{1}{\Delta t} f(u_{RP}(x_L,t^{n+1}))\Delta t \quad \text{(since } u_{RP}(x_L,t) \text{ is constant at } t^n \leq t \leq t^{n+1}) \]
  \[ = f(u_{RP}(x_L,t^{n+1})) \]

- 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  \quad dt = 0.0004, \quad time\ step = \quad 73

**Inviscid flow part**

\begin{align*}
\text{unew}(0) &= 1.000000 \\
\text{unew}(1) &= 1.000000 \\
\text{unew}(2) &= 1.000000 \\
\text{unew}(3) &= 1.000000 \\
\text{unew}(4) &= 1.000000 \\
\text{unew}(5) &= 1.000000 \\
\text{unew}(6) &= 1.000000 \\
\text{unew}(7) &= 1.000000 \\
\text{unew}(8) &= 1.000000 \\
\text{unew}(9) &= 1.000000 \\
\text{unew}(10) &= 1.000000 \\
\text{unew}(11) &= 1.000000 \\
\text{unew}(12) &= 1.000000 \\
\text{unew}(13) &= 1.000000 \\
\text{unew}(14) &= 1.000000 \\
\text{unew}(15) &= 1.000000 \\
\text{unew}(16) &= 1.000000 \\
\text{unew}(17) &= 1.000000 \\
\text{unew}(18) &= 1.000000 \\
\text{unew}(19) &= 1.000000 \\
\text{unew}(20) &= 1.000000 \\
\text{unew}(21) &= 1.000001
\end{align*}

**Viscous flow part**

\begin{align*}
\text{vnew}(0) &= 1.000001 \\
\text{vnew}(1) &= 0.999990 \\
\text{vnew}(2) &= 0.999866 \\
\text{vnew}(3) &= 0.998501 \\
\text{vnew}(4) &= 0.983498 \\
\text{vnew}(5) &= 0.819807 \\
\text{vnew}(6) &= -0.819817 \\
\text{vnew}(7) &= -0.983507 \\
\text{vnew}(8) &= -0.998511 \\
\text{vnew}(9) &= -0.999876 \\
\text{vnew}(10) &= -1.000000
\end{align*}
<table>
<thead>
<tr>
<th>Viscous flow part</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{\text{new}}(0) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(1) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(2) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(3) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(4) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(5) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(6) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(7) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(8) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(9) = -1.00000000 )</td>
</tr>
<tr>
<td>( v_{\text{new}}(10) = -1.00000000 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inviscid flow part</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_{\text{new}}(0) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(1) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(2) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(3) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(4) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(5) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(6) = 1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(7) = 0.99999996 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(8) = 0.99999979 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(9) = 0.99999433 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(10) = -0.36233677 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(11) = -0.99999250 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(12) = -0.99999989 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(13) = -0.99999999 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(14) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(15) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(16) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(17) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(18) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(19) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(20) = -1.00000000 )</td>
</tr>
<tr>
<td>( u_{\text{new}}(21) = -1.00000000 )</td>
</tr>
</tbody>
</table>

Set 2

\[ \text{time step} = 406 \]

\[ \text{dt} = 0.0004 \]
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

[Diagram showing the system structure of PDE.Mart Platform, including PDE-GUI, PDE-Server, Call-back Wrappers, Java Wrappers, C++ Wrappers, Fortran Wrappers, C library from System A, C library from System B, and Fortran library from System X.]
PDE.Mart

- Network-based PSE
- 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