Simultaneous Space-Time Adaptive Solution of Partial Differential Equations *

Jahrul Alam

Department of Mathematics and Statistics
McMaster University, Canada

and

Nicholas Kevlahan

Department of Mathematics and Statistics
McMaster University, Canada

* **Adaptive wavelet and multiscale methods for partial differential equations June 3 - 5, 2004, Banff International Research Station**

Collaborators

- **O. V. Vasilyev (University of Colorado at Boulder)**
- **D. Goldstein (University of Colorado at Boulder)**

Outline

Outline

- **Motivation**

Outline

- **Motivation**
- **Adaptive wavelet collocation method**

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem
- Multilevel elliptic solver

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem
- Multilevel elliptic solver
- Adaptive nonlinear solver

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem
- Multilevel elliptic solver
- Adaptive nonlinear solver
- Results and discussion

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem
- Multilevel elliptic solver
- Adaptive nonlinear solver
- Results and discussion
- Conclusion and future direction

Outline

- Motivation
- Adaptive wavelet collocation method
- Application to PDEs
  - Elliptic problem
  - Nonlinear evolution problem
- Multilevel elliptic solver
- Adaptive nonlinear solver
- Results and discussion
- Conclusion and future direction

Motivation

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**

Motivation

- **Engineering problems:** $\longrightarrow$ **partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology:**

Motivation

- **Engineering problems:** $\longrightarrow$ **partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology: storm,**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology: storm, surge,**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology: storm, surge, tide...**

Motivation

- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology: storm, surge, tide...**
  3. **Bio-medical engineering**

Motivation

- Engineering problems: $\longrightarrow$ partial differential equations.
- Localized structures:
  - might occur intermittently anywhere
  - might change their location and scales
- Example:
  1. Turbulence computation
  2. Meteorology: storm, surge, tide...
  3. Bio-medical engineering, Nano-technology

Motivation

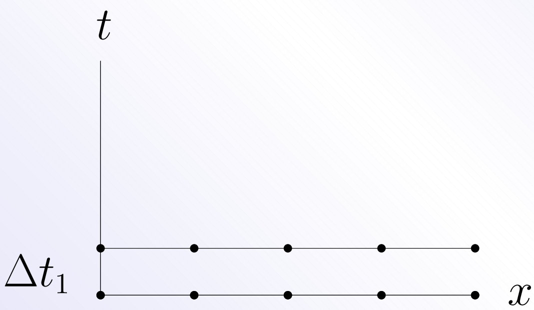
- **Engineering problems: $\longrightarrow$ partial differential equations.**
- **Localized structures:**
  - **might occur intermittently anywhere**
  - **might change their location and scales**
- **Example:**
  1. **Turbulence computation**
  2. **Meteorology: storm, surge, tide...**
  3. **Bio-medical engineering, Nano-technology**
- **Uniform grid for such a problem is not suitable**

Motivation

- **Grid should adapt in space and time**

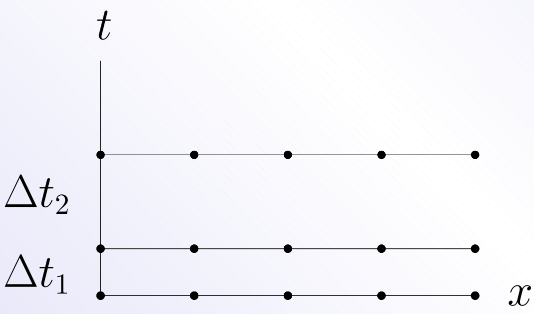
Motivation

- **Grid should adapt in space and time**



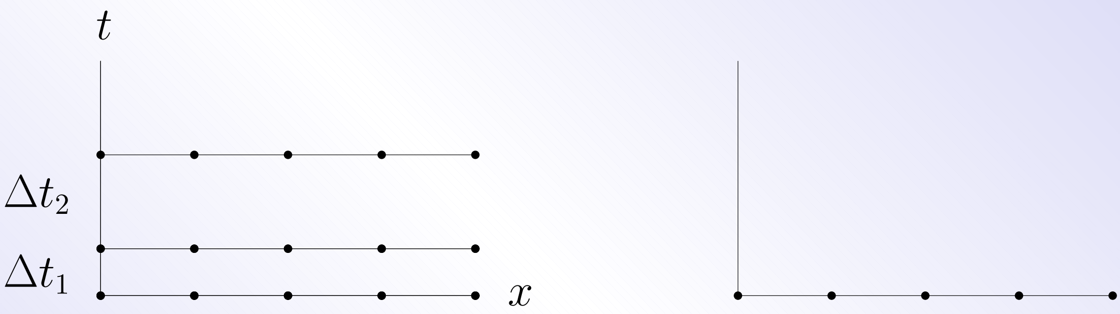
Motivation

- **Grid should adapt in space and time**



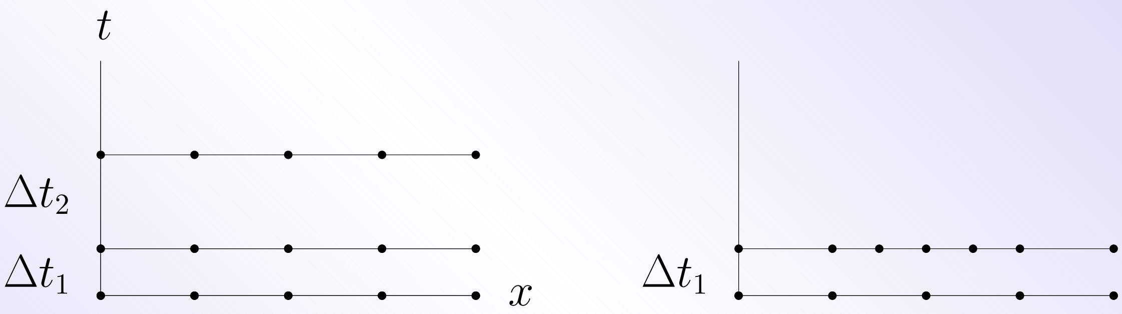
Motivation

- **Grid should adapt in space and time**



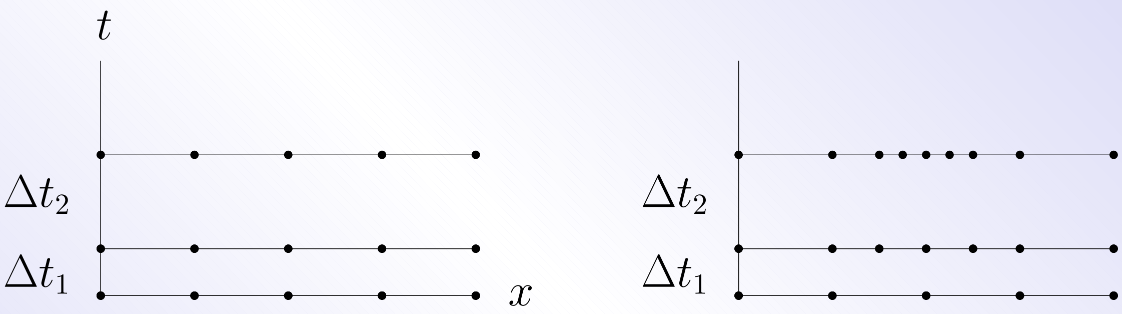
Motivation

- **Grid should adapt in space and time**



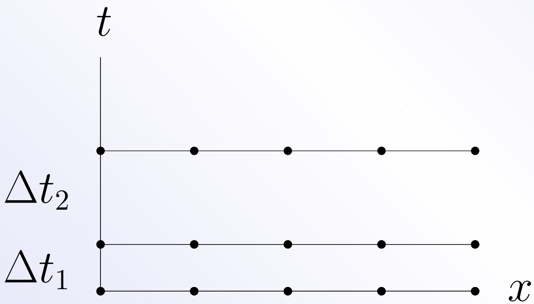
Motivation

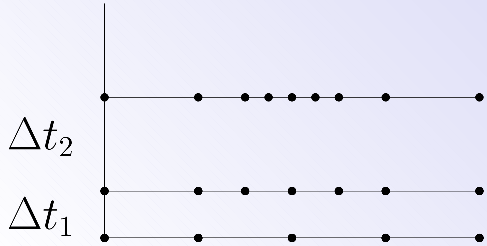
- **Grid should adapt in space and time**

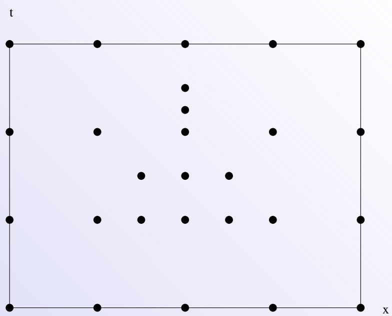


Motivation

- **Grid should adapt in space and time**







$\longleftarrow$ **Space-time adaptive grid**

Motivation: wavelet decomposition

Motivation: wavelet decomposition

- **What are wavelets?**

Motivation: wavelet decomposition

- **What are wavelets?**

  A set of basis functions that are ....

Motivation: wavelet decomposition

- **What are wavelets?**

  A set of basis functions that are localized in space and scale

Motivation: wavelet decomposition

- **What are wavelets?**

  A set of basis functions that are localized in space and scale

- **Represent a function in terms of wavelet basis:**

Motivation: wavelet decomposition

- **What are wavelets?**

  A set of basis functions that are localized in space and scale

- **Represent a function in terms of wavelet basis:**

$$u(x) = \sum_{j=0}^{\infty} \sum_{k \in \mathcal{K}^j} d_k^j \psi_k^j(x)$$

Motivation: wavelet decomposition

- **What are wavelets?**

  A set of basis functions that are localized in space and scale

- **Represent a function in terms of wavelet basis:**

$$u(x) = \sum_{j=0}^{\infty} \sum_{k \in \mathcal{K}^j} d_k^j \psi_k^j(x)$$

- **Wavelets:**
  - **follow intermittency in position and scale**
  - **provide automatic grid adaptation**

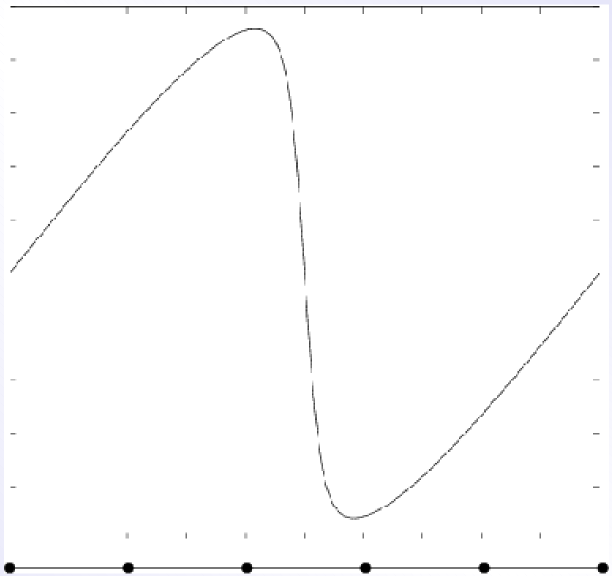
Adaptive wavelet collocation method

Adaptive wavelet collocation method

- **Sampling a function on a grid**

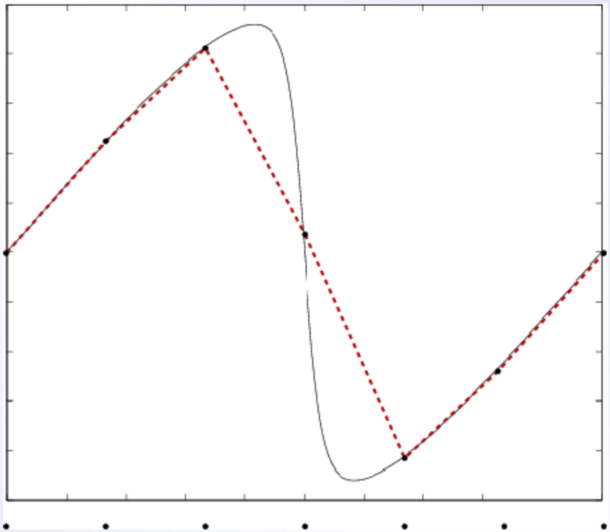
Adaptive wavelet collocation method

- **Sampling a function on a grid**



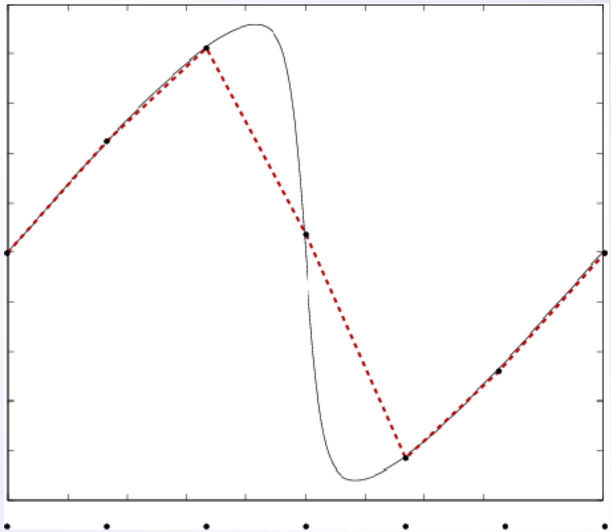
Adaptive wavelet collocation method

- **Sampling a function on a grid**



Adaptive wavelet collocation method

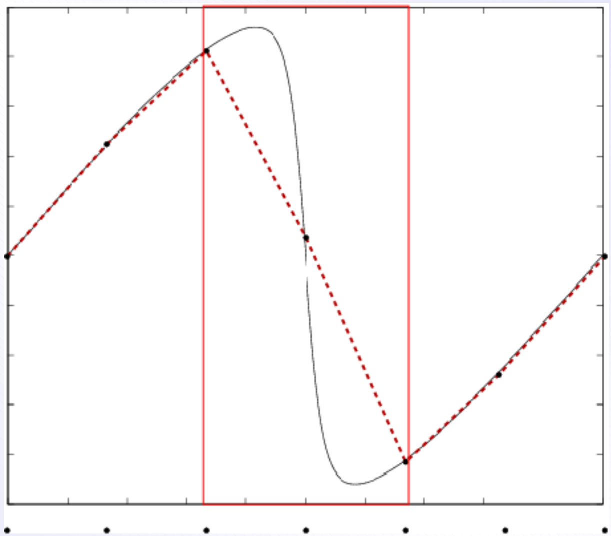
- Sampling a function on a grid



- Grid refinement is not required everywhere

Adaptive wavelet collocation method

- Sampling a function on a grid



- Grid refinement is not required everywhere

Adaptive wavelet collocation method: Cont'd...

Adaptive wavelet collocation method: Cont'd...

- **Nested dyadic grid:**

Adaptive wavelet collocation method: Cont'd...

- **Nested dyadic grid:**

$$G^j = \{x_k^j \in \mathbb{R} :$$

Adaptive wavelet collocation method: Cont'd...

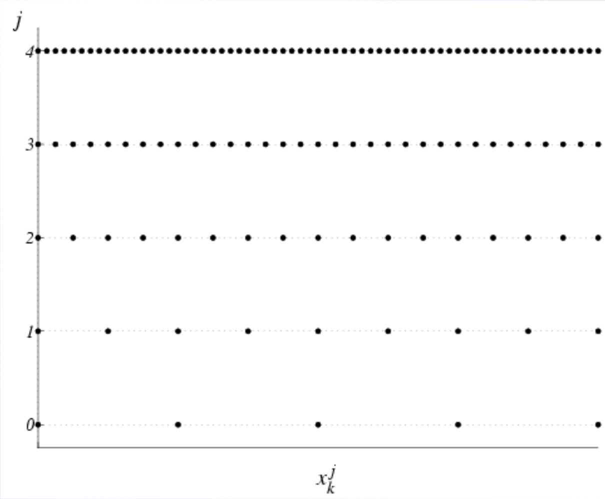
- **Nested dyadic grid:**

$$G^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}\}$$

Adaptive wavelet collocation method: Cont'd...

- **Nested dyadic grid:**

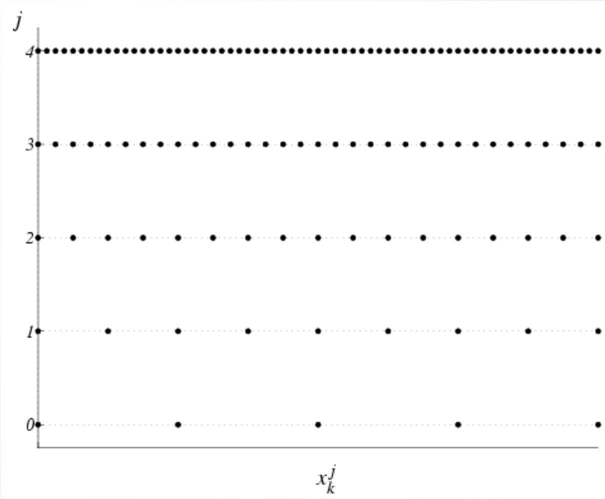
$$G^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}\}$$



Adaptive wavelet collocation method: Cont'd...

- **Nested dyadic grid:**

$$G^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}\}$$

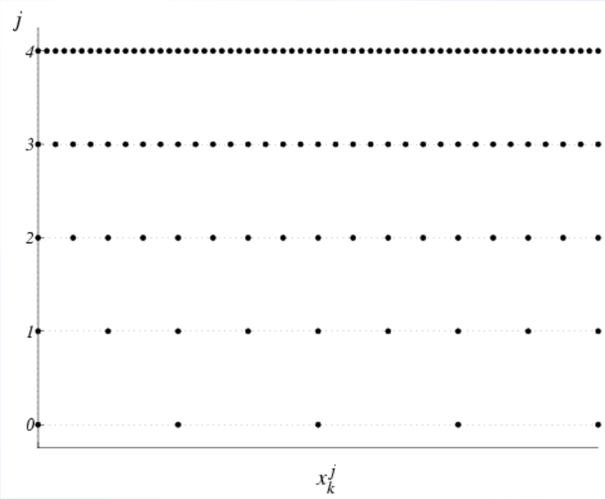


- **Easy to see the nestedness property:**

Adaptive wavelet collocation method: Cont'd...

- **Nested dyadic grid:**

$$G^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}\}$$



- **Easy to see the nestedness property:**

$$G^j \subset G^{j+1} \quad \text{i.e.} \quad x_{2k}^{j+1} = x_k^j$$

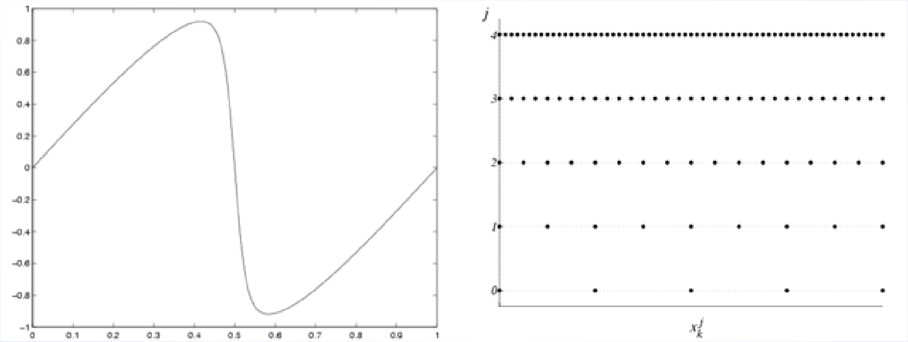
Adaptive wavelet collocation method: Cont'd...

Adaptive wavelet collocation method: Cont'd...

- **Adaptive wavelet grid:**

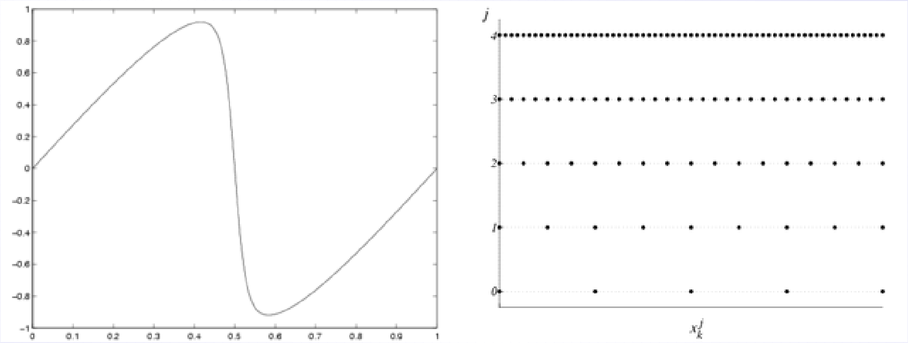
Adaptive wavelet collocation method: Cont'd...

- **Adaptive wavelet grid:**



Adaptive wavelet collocation method: Cont'd...

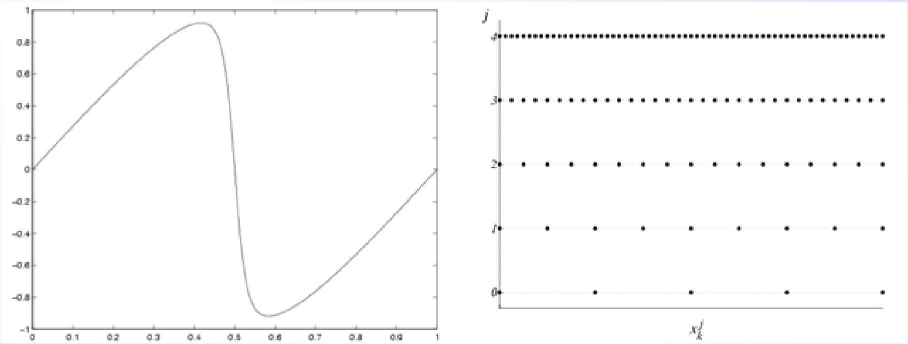
- **Adaptive wavelet grid:**



$$u(x) = \sum_{j=0}^{\infty} \sum_{k \in \mathcal{K}^j} d_k^j \psi_k^j(x)$$

Adaptive wavelet collocation method: Cont'd...

- **Adaptive wavelet grid:**

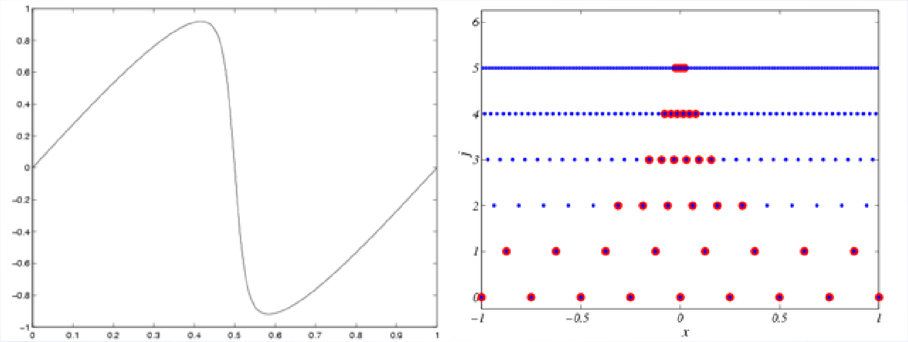


$$u(x) = \sum_{j=0}^{\infty} \sum_{k \in \mathcal{K}^j} d_k^j \psi_k^j(x)$$

$$G^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}\}$$

Adaptive wavelet collocation method: Cont'd...

- **Adaptive wavelet grid:**



$$u(x) = \sum_{j=0}^{\infty} \sum_{k \in \mathcal{K}^j} d_k^j \psi_k^j(x)$$

$$G_a^j = \{x_k^j \in \mathbb{R} : x_k^j = 2^{-j}k, k \in \mathcal{Z}, j \in \mathcal{Z}, \ |d_k^j| \geq \epsilon\}$$

Application to PDEs

Application to PDEs

- **Consider a PDE:**

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Boundary conditions

$$\mathcal{B}u = u_0 \quad x \in \partial\Omega$$

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Boundary conditions

$$\mathcal{B}u = u_0 \quad x \in \partial\Omega$$

- **Example:**

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Boundary conditions

$$\mathcal{B}u = u_0 \quad x \in \partial\Omega$$

- **Example:**

1. Elliptic:

$$\mathcal{L} := \partial_{xx} + \partial_{yy}$$

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Boundary conditions

$$\mathcal{B}u = u_0 \quad x \in \partial\Omega$$

- **Example:**

1. Elliptic:

$$\mathcal{L} := \partial_{xx} + \partial_{yy}$$

2. Parabolic:

$$\mathcal{L} := \partial_t - \partial_{xx}$$

Application to PDEs

- **Consider a PDE:**

$$\mathcal{L}u = f, \quad x \in \Omega,$$

where $\mathcal{L}$ is any partial differential operator.

Boundary conditions

$$\mathcal{B}u = u_0 \quad x \in \partial\Omega$$

- **Example:**

1. Elliptic:

$$\mathcal{L} := \partial_{xx} + \partial_{yy}$$

2. Parabolic:

$$\mathcal{L} := \partial_t - \partial_{xx}$$

- **Classical solution procedure**

  Sequence of algebraic problem (via ODE solver)

- **Our goal**

  A single algebraic problem

Application to PDEs: Cont'd...

Application to PDEs: Cont'd...

- **Wavelet transform:**

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j}^{\mathcal{O}(\mathcal{N})}$$

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j}^{\mathcal{O}(\mathcal{N})}$$

- **Calculating derivatives:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j \quad \Longrightarrow \quad \frac{\partial u}{\partial x}(x_k^j)}^{\mathcal{O}(\mathcal{N})}$$

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j}^{\mathcal{O}(\mathcal{N})}$$

- **Calculating derivatives:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j \quad \Longrightarrow \quad \frac{\partial u}{\partial x}(x_k^j)}^{\mathcal{O}(\mathcal{N})}$$

- **Reduce to an algebraic system:**

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j}^{\mathcal{O}(\mathcal{N})}$$

- **Calculating derivatives:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j \quad \Longrightarrow \quad \frac{\partial u}{\partial x}(x_k^j)}^{\mathcal{O}(\mathcal{N})}$$

- **Reduce to an algebraic system:**

$$Lu = f$$

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j}^{\mathcal{O}(\mathcal{N})}$$

- **Calculating derivatives:**

$$\overbrace{u(x_k^j) \quad \Longrightarrow \quad d_k^j \quad \Longrightarrow \quad \frac{\partial u}{\partial x}(x_k^j)}^{\mathcal{O}(\mathcal{N})}$$

- **Reduce to an algebraic system:**

$$Lu = f$$

- **Solve the system:**

Application to PDEs: Cont'd...

- **Wavelet transform:**

$$\overbrace{u(x_k^j) \Longrightarrow d_k^j}^{\mathcal{O}(\mathcal{N})}$$

- **Calculating derivatives:**

$$\overbrace{u(x_k^j) \Longrightarrow d_k^j \Longrightarrow \frac{\partial u}{\partial x}(x_k^j)}^{\mathcal{O}(\mathcal{N})}$$

- **Reduce to an algebraic system:**

$$Lu = f$$

- **Solve the system: Multilevel adaptive wavelet solver**

Application to PDEs: Cont'd...

Application to PDEs: Cont'd...

- **Wavelet grid:**

Application to PDEs: Cont'd...

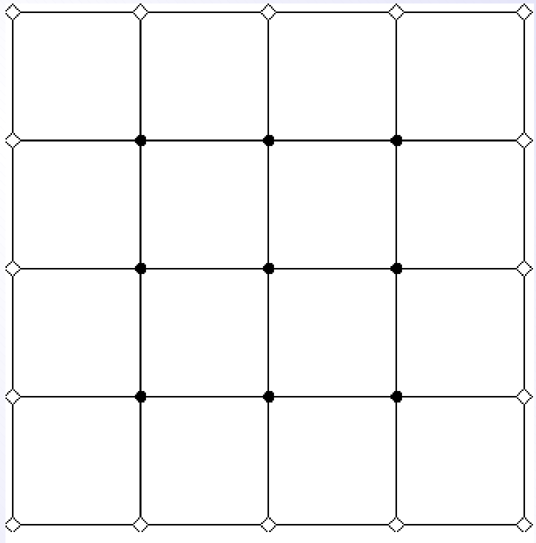
- **Wavelet grid: internal points**

Application to PDEs: Cont'd...

- **Wavelet grid: internal points, boundary points**

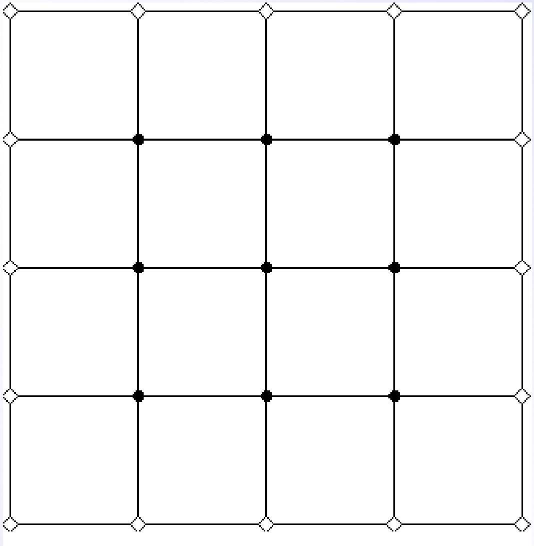
Application to PDEs: Cont'd...

- **Wavelet grid: internal points, boundary points**



Application to PDEs: Cont'd...

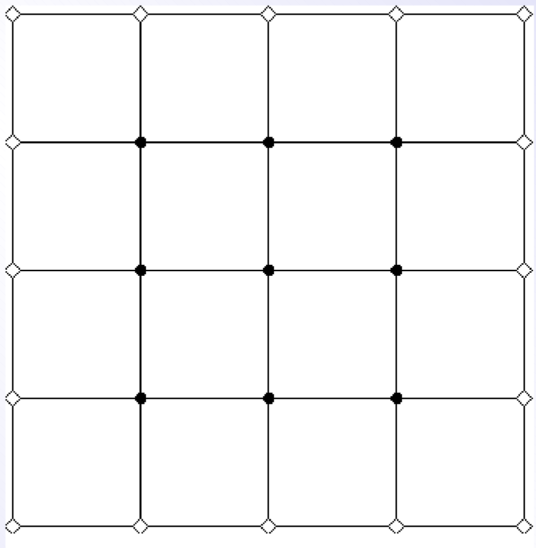
- **Wavelet grid: internal points, boundary points**



- *Internal points*

Application to PDEs: Cont'd...

- **Wavelet grid: internal points, boundary points**

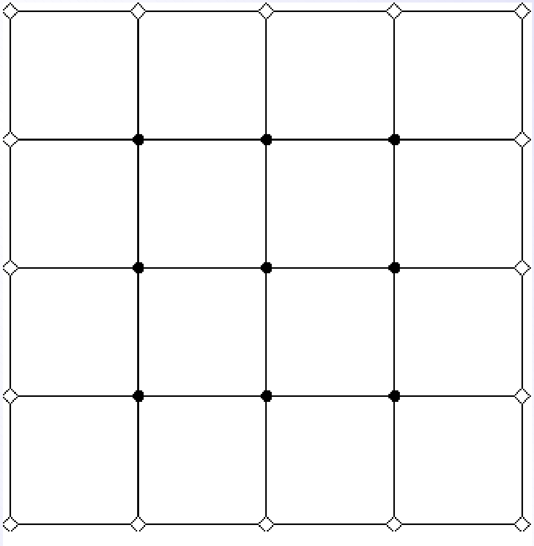


- *Internal points*

◇ *Boundary points*

Application to PDEs: Cont'd...

- **Wavelet grid: internal points, boundary points**



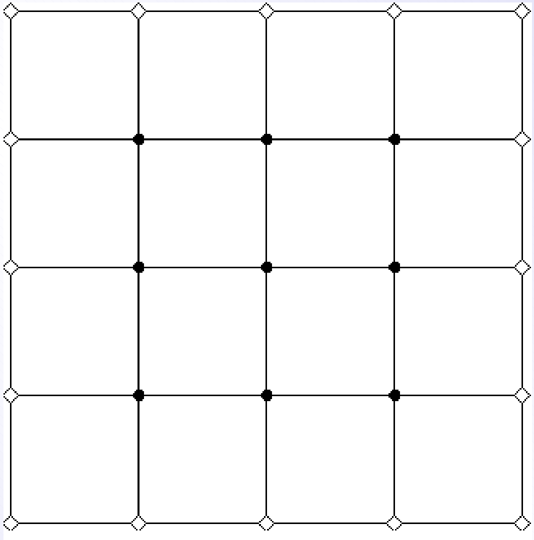
- *Internal points*

◇ *Boundary points*

Solve DE on internal points

Application to PDEs: Cont'd...

- **Wavelet grid: internal points, boundary points**



- *Internal points*

◇ *Boundary points*

Solve DE on internal points
Implement BC on Boundary points

Application to PDEs: Elliptic problems

Application to PDEs: Elliptic problems

- **Poisson equation**

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= f, \quad x \in \Omega, \\ u &= u_0, \quad x \in \partial\Omega \end{aligned}$$

Application to PDEs: Elliptic problems

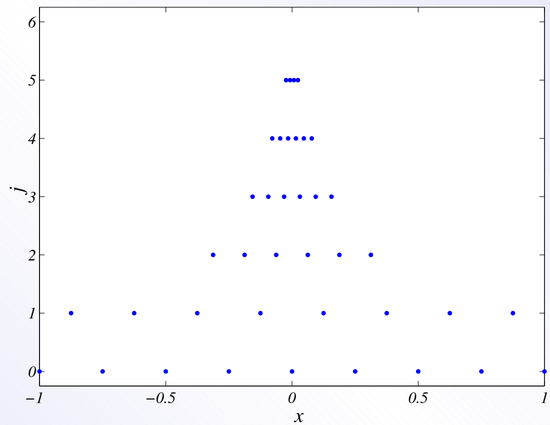
- **Poisson equation**

$$
\begin{aligned}
\frac{\partial^2 u}{\partial x^2} &= f, \quad x \in \Omega, \\
u &= u_0, \quad x \in \partial\Omega \\
&\Downarrow \\
Lu &= f
\end{aligned}
$$

Application to PDEs: Elliptic problems

- **Poisson equation**

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= f, \quad x \in \Omega, \\ u &= u_0, \quad x \in \partial\Omega \\ &\Downarrow \\ Lu &= f \end{aligned}$$

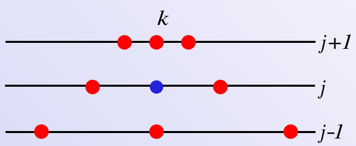


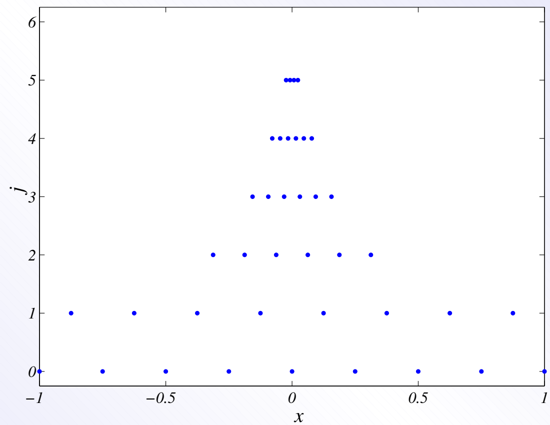
Application to PDEs: Elliptic problems

- **Poisson equation**

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= f, \quad x \in \Omega, \\ u &= u_0, \quad x \in \partial\Omega \\ &\Downarrow \\ Lu &= f \end{aligned}$$

Adjacent zone:





Application to PDEs: Elliptic problems

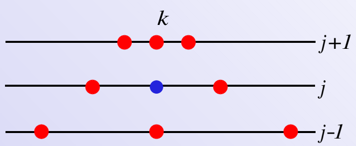
- **Poisson equation**

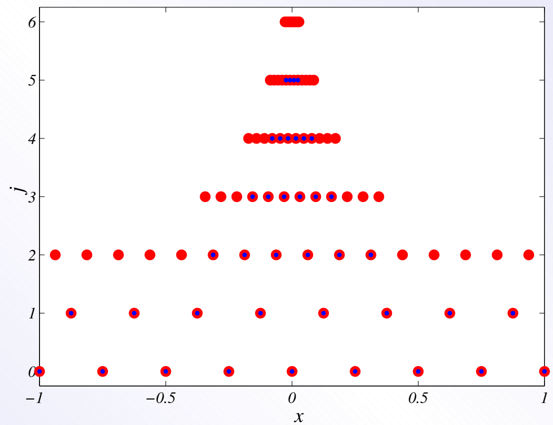
$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= f, \quad x \in \Omega, \\ u &= u_0, \quad x \in \partial\Omega \end{aligned}$$

$$\Downarrow$$

$$Lu = f$$

Adjacent zone:





Nonlinear evolution problem

Nonlinear evolution problem

- **Example:**

Nonlinear evolution problem

- **Example:**
  Navier-Stokes

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

YES

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

YES

- **What is the boundary condition at fixed time?**

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

YES

- **What is the boundary condition at fixed time?**

  We propose:

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

YES

- **What is the boundary condition at fixed time?**

  We propose:

$$\mathcal{L} u - f = 0 \quad \text{for} \quad t = t_{\max}$$

Nonlinear evolution problem

- **Example:**

  Navier-Stokes

$$\nabla \cdot \mathbf{u}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}$$

  Kuramoto-Sivashinsky

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u \partial_x u = 0$$

- **Can we reduce to an algebraic problem?**

YES

- **What is the boundary condition at fixed time?**

  We propose:

$$\mathcal{L} u - f = 0 \quad \text{for} \quad t = t_{\max}$$

evolution type boundary condition.

Multilevel elliptic solver

V-cycle:

$\mathbf{r}^J = \mathbf{f}^J - \mathbf{L}\mathbf{u}^J$
for all levels $j = J \ : \ -1 \ : \ j_{\min} + 1$
  do ν_1 steps of **approximate** solver for $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$
    $\mathbf{r}^{j-1} = I_w^{j-1} \left(\mathbf{r}^j - \mathbf{L}\mathbf{v}^j\right)$
  enddo
end
Solve for $j = j_{\min}$ level: $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$
for all levels $j = j_{\min} + 1 \ : \ +1 \ : \ J$
  $\mathbf{v}^j = \mathbf{v}^j + \omega_0 I_w^j \mathbf{v}^{j-1}$
  do ν_2 steps of **approximate** solver for $\mathbf{L}\mathbf{v}^j = \mathbf{r}^j$ **enddo**
end
$\mathbf{u}^J = \mathbf{u}^J + \omega_1 \mathbf{v}^J$
do ν_3 steps of **exact** solver for $\mathbf{L}\mathbf{u}^J = \mathbf{f}^J$ **enddo**

Adaptive nonlinear solver

V-cycle:

$\mathbf{r}^J = \mathbf{f}^J - \mathbf{L}\mathbf{u}^J$

for all levels $j = J \;:\; -1 \;:\; j_{\min} + 1$

do ν_1 steps of **approximate** solver for $\mathbf{J}(u)\mathbf{v}^j = \mathbf{r}^j$

$\mathbf{r}^{j-1} = I_w^{j-1}\left(\mathbf{r}^j - \mathbf{J}(u)\mathbf{v}^j\right)$

enddo

end

Solve for $j = j_{\min}$ level: $\mathbf{J}(u)\mathbf{v}^j = \mathbf{r}^j$

for all levels $j = j_{\min} + 1 \;:\; +1 \;:\; J$

$\mathbf{v}^j = \mathbf{v}^j + \omega_0 I_w^j \mathbf{v}^{j-1}$

do ν_2 steps of **approximate** solver for $\mathbf{J}(u)\mathbf{v}^j = \mathbf{r}^j$ **enddo**

end

$\mathbf{u}^J = \mathbf{u}^J + \omega_1 \mathbf{v}^J$

enddo

Result and discussion

Result and discussion

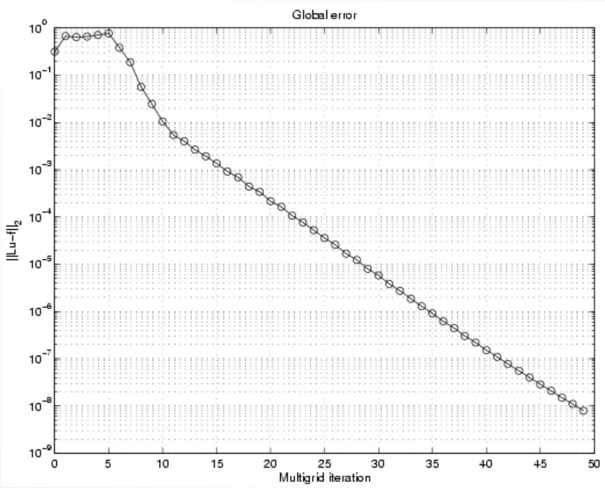
- **Adaptive nonlinear solver**

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u\partial_x u = 0$$

Result and discussion

- **Adaptive nonlinear solver**

$$\partial_t u + \partial_{xxxx} u + \partial_{xx} u + u\partial_x u = 0$$



L_2 norm of residual as a function of multigrid iteration

Result and discussion: Cont'd...

- **Burgers equation**

Result and discussion: Cont'd...

- **Burgers equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}],\ \Omega = [0, 1]$$

$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$

$$\nu = 10^{-2}$$

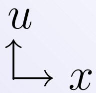
Result and discussion: Cont'd...

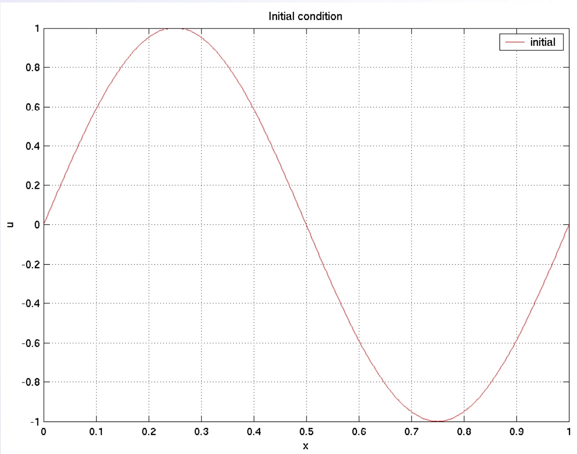
- **Burgers equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}],\ \Omega = [0, 1]$$

$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$

$$\nu = 10^{-2}$$





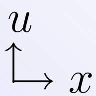
Initial condition

Result and discussion: Cont'd...

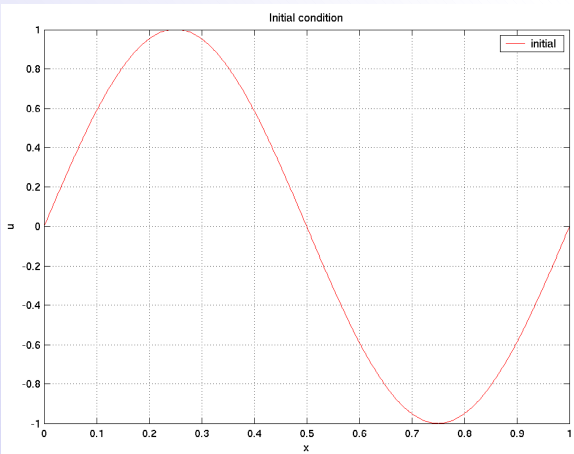
- **Burgers equation**

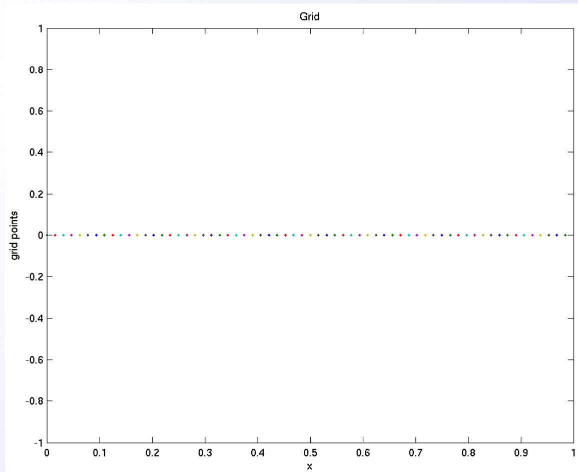
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}], \; \Omega = [0, 1]$$

$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$



$$\nu = 10^{-2}$$





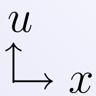
Initial condition | Grid

Result and discussion: Cont'd...

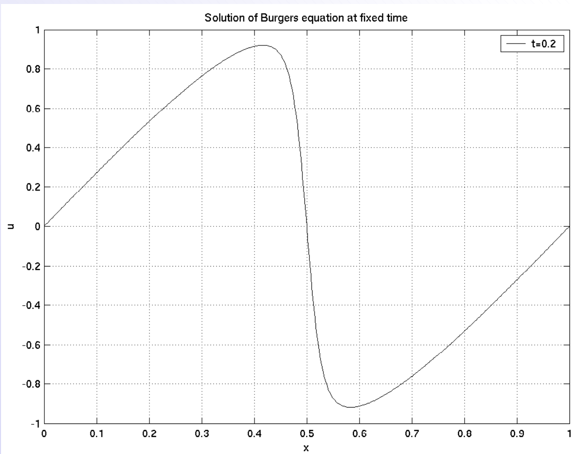
- **Burgers equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}], \; \Omega = [0,1]$$

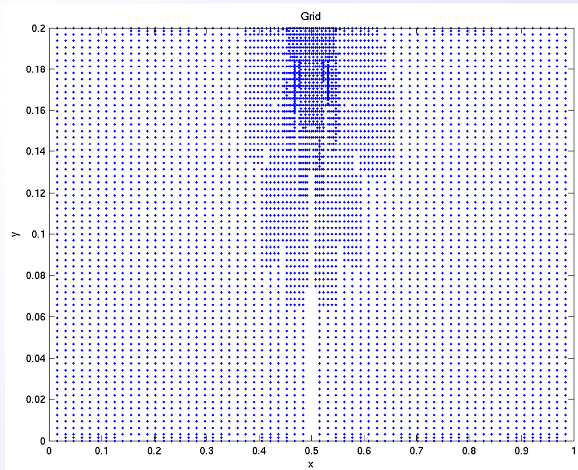
$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$



$$\nu = 10^{-2}$$



Computed solution



Adapted grid

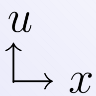
Result and discussion: Cont'd...

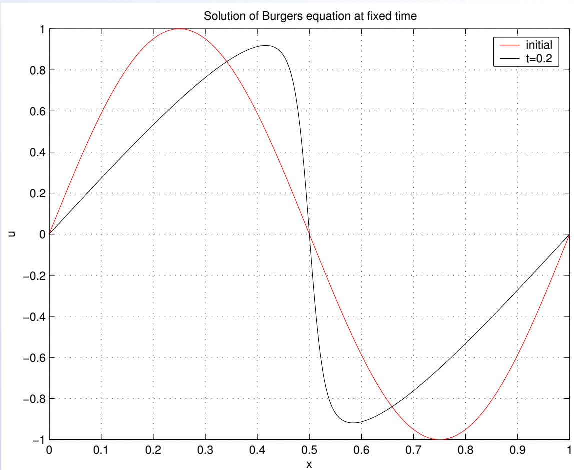
- **Burgers equation**

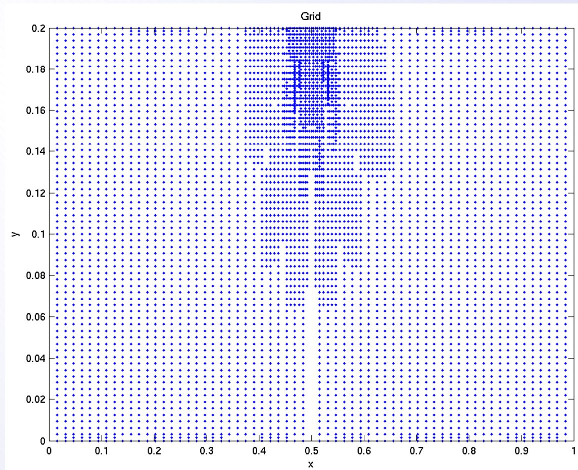
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}],\ \Omega = [0,1]$$

$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$

$$\nu = 10^{-2}$$







Computed solution **Adapted grid**

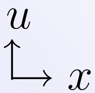
Result and discussion: Cont'd...

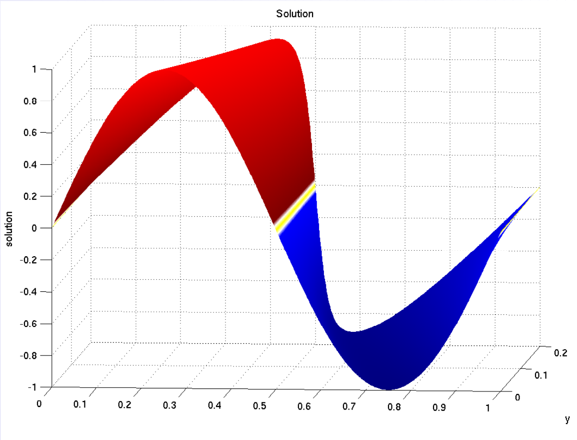
- **Burgers equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \subset \mathbb{R} \times [0, t_{\max}],\ \Omega = [0,1]$$

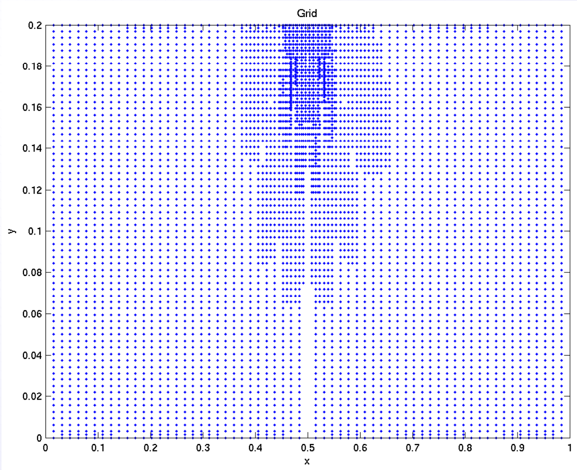
$$u(0,t) = u(1,t), \quad u(x,0) = \sin(2\pi x)$$

$$\nu = 10^{-2}$$





Solution surface
(Space-time domain)



Adapted grid

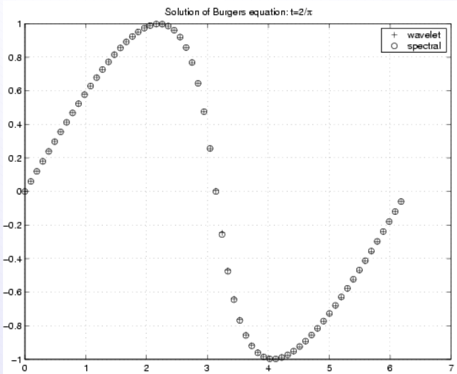
Result and discussion: Cont'd...

- **Burgers equation**

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in (-\pi, \pi)$$

$$u(-\pi, t) = u(\pi, t)$$

$$u(x, 0) = \sin(x)$$



Compare wavelet solution with a spectral code.

Result and discussion: Cont'd...

- **Moving shock**

Result and discussion: Cont'd...

- **Moving shock**

$$\frac{\partial u}{\partial t} + (u+v)\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \times [0, t_{\max}], \, \Omega = (0,2)$$

$$u(0,t) = 1, \;\; u(2,t) = -1, \quad u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right)$$

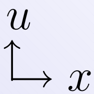
$$\nu = 10^{-2}, \quad x_0 = 0.5$$

Result and discussion: Cont'd...

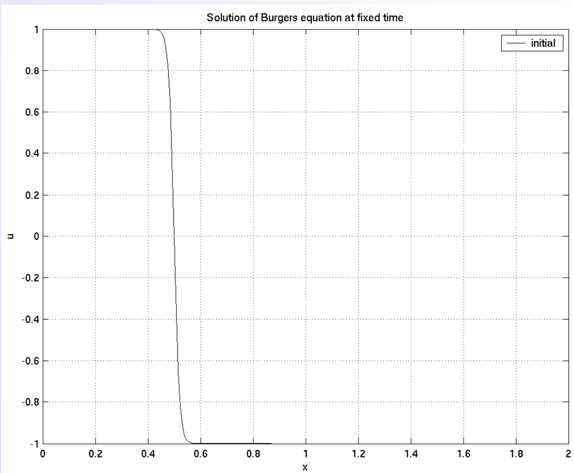
- **Moving shock**

$$\frac{\partial u}{\partial t} + (u+v)\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \times [0, t_{\max}],\ \Omega = (0,2)$$

$$u(0,t) = 1,\ \ u(2,t) = -1, \quad u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right)$$



$$\nu = 10^{-2}, \quad x_0 = 0.5$$



Initial condition

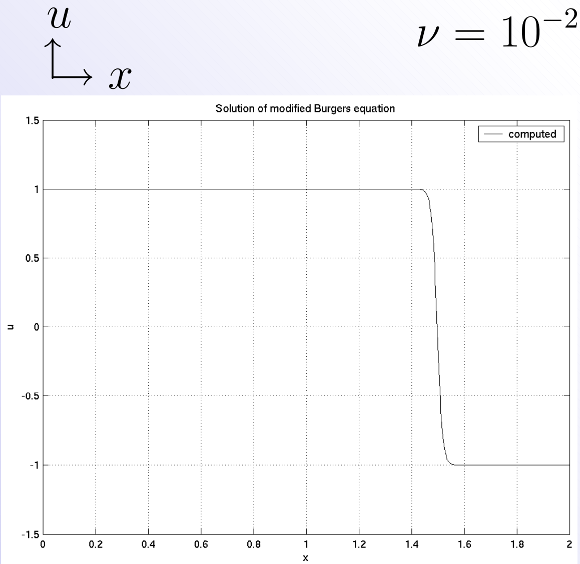
Result and discussion: Cont'd...

- **Moving shock**

$$\frac{\partial u}{\partial t} + (u+v)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \times [0, t_{\max}],\ \Omega = (0,2)$$

$$u(0,t) = 1,\ u(2,t) = -1, \quad u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right)$$

$$\nu = 10^{-2}, \quad x_0 = 0.5$$



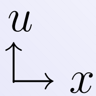
Solution at $t = 1.0$

Result and discussion: Cont'd...

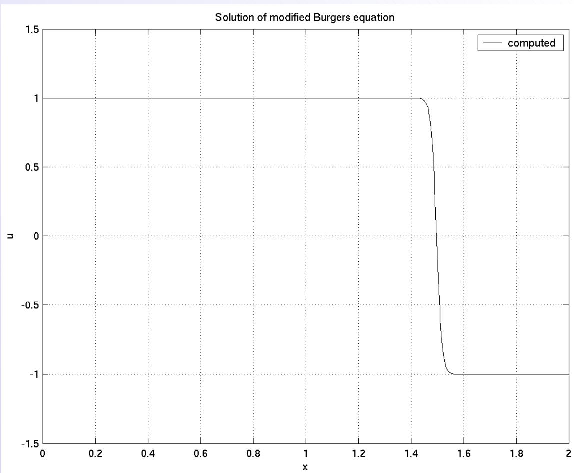
- **Moving shock**

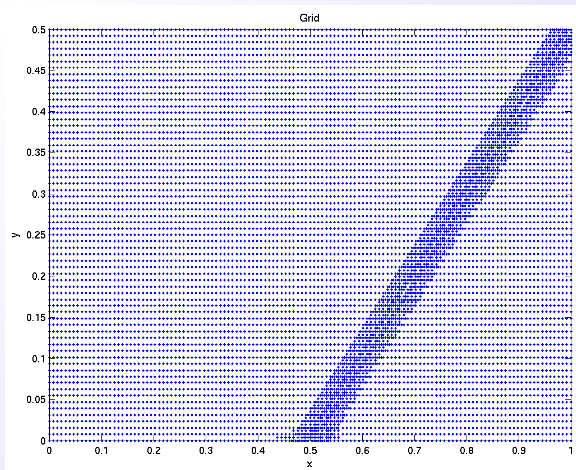
$$\frac{\partial u}{\partial t} + (u+v)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \times [0, t_{\max}], \; \Omega = (0,2)$$

$$u(0,t) = 1, \; u(2,t) = -1, \quad u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right)$$



$$\nu = 10^{-2}, \quad x_0 = 0.5$$





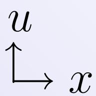
Solution at $t = 1.0$ **Adapted grid**

Result and discussion: Cont'd...

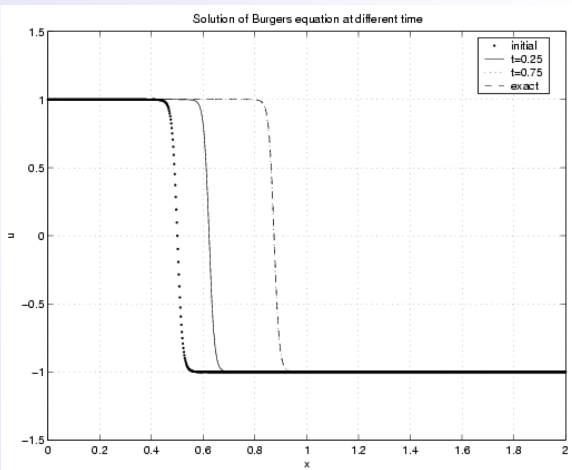
- **Moving shock**

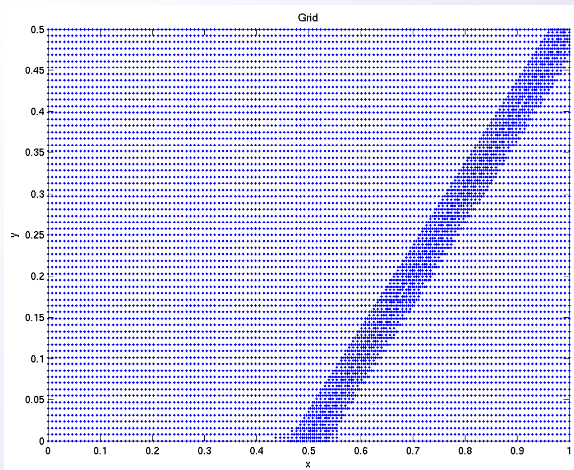
$$\frac{\partial u}{\partial t} + (u+v)\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in \Omega \times [0, t_{\max}],\ \Omega = (0,2)$$

$$u(0,t) = 1,\ u(2,t) = -1, \quad u(x,0) = -\tanh\left(\frac{x - x_0}{2\nu}\right)$$



$$\nu = 10^{-2}, \quad x_0 = 0.5$$



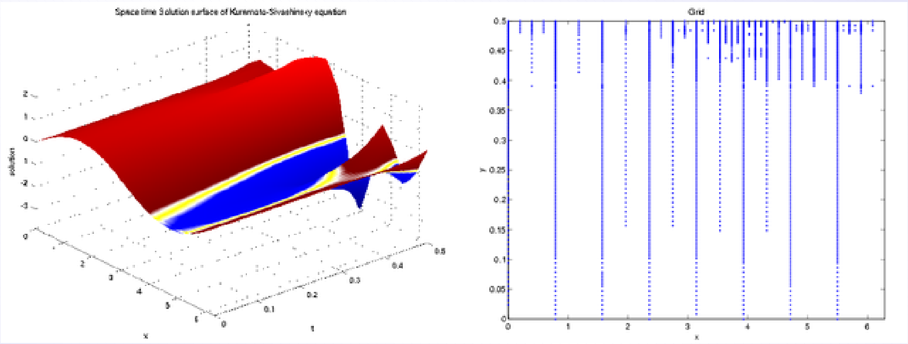


Solution **Adapted grid**

Result and discussion: Cont'd...

- **Kuramoto-Sivashinsky equation**

$$\frac{\partial u}{\partial t} + \nu_4 \partial_x^4 u + \partial_x^2 u + u \partial_x u = 0, \quad x \in \Omega \times [0, t_{\max}], \, \Omega = [0, 2\pi]$$

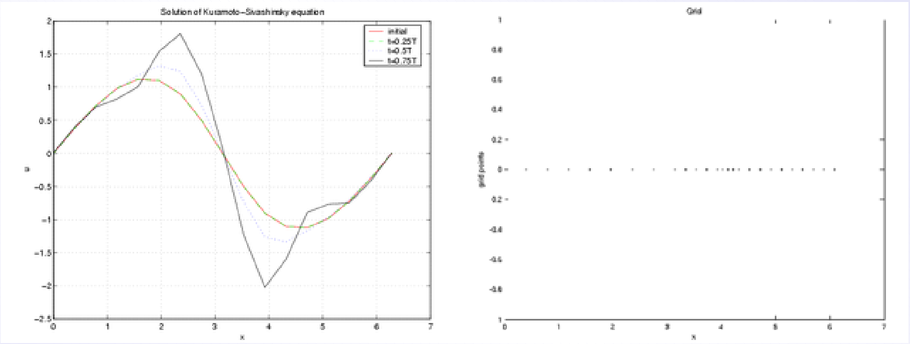


Space-time solution surface and corresponding grid

Result and discussion: Cont'd...

- **Kuramoto-Sivashinsky equation**

$$\frac{\partial u}{\partial t} + \nu_4 \partial_x^4 u + \partial_x^2 u + u \partial_x u = 0, \quad x \in \Omega \times [0, t_{\max}], \, \Omega = [0, 2\pi]$$



Fixed time solution and corresponding grid

Conclusion and Future direction

Conclusion and Future direction

- **Conclusion**

Conclusion and Future direction

- Conclusion
  - An adaptive numerical method is developed

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**

Conclusion and Future direction

- Conclusion
  - An adaptive numerical method is developed
  - An evolution type boundary condition is proposed
  - A multilevel adaptive solver has been developed

Conclusion and Future direction

- Conclusion
  - An adaptive numerical method is developed
  - An evolution type boundary condition is proposed
  - A multilevel adaptive solver has been developed
- Future plan

Conclusion and Future direction

- Conclusion
  - An adaptive numerical method is developed
  - An evolution type boundary condition is proposed
  - A multilevel adaptive solver has been developed
- Future plan
  - Drawback?

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**
  - **A multilevel adaptive solver has been developed**
- **Future plan**
  - **Drawback?**
    - Simulation to carry out for large t

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**
  - **A multilevel adaptive solver has been developed**
- **Future plan**
  - **Drawback?**
    - Simulation to carry out for large t
    - Better time stepping

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**
  - **A multilevel adaptive solver has been developed**
- **Future plan**
  - **Drawback?**
    - Simulation to carry out for large t
    - Better time stepping
  - **Solution**

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**
  - **A multilevel adaptive solver has been developed**
- **Future plan**
  - **Drawback?**
    - Simulation to carry out for large t
    - Better time stepping
  - **Solution**
    - flip and solve method

Conclusion and Future direction

- **Conclusion**
  - **An adaptive numerical method is developed**
  - **An evolution type boundary condition is proposed**
  - **A multilevel adaptive solver has been developed**
- **Future plan**
  - **Drawback?**
    * Simulation to carry out for large t
    * Better time stepping
  - **Solution**
    * flip and solve method
    * Lagrangian or variational idea

Thank You