# The Adaptive Method of Lines

## CDEs Group Presentation

Shian Su, Ria Szeredi and Kenneth Young

University of Melbourne

23rd of May, 2016

# Motivation

**Method of lines**

- Discretise in space (ie using finite difference) to generate a system of ODEs.
- Solve the system using some solver such as ode15s.
- Temporal accuracy handled by the ODE solver, however spatial accuracy results from the discretisation used.

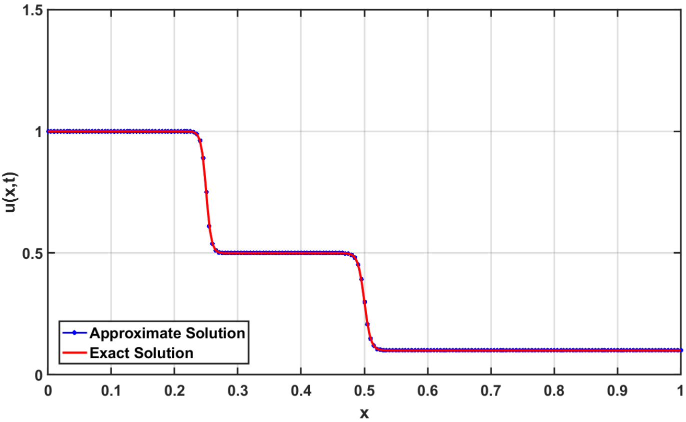
# Motivation

**Problems with Uniform Mesh**

- Method of lines discretises the space uniformly.
- Low density uniform mesh doesn't approximate steep fronts well.
- Using a high density uniform mesh to achieve high spatial accuracy wastes nodes in regions of low activity.

# Method of Lines - Burgers' Equation

- Bad approximation in regions of high spatial activity when grid is uniform. Here we use 201 nodes.



# Motivation

## Mesh adaptation

- Want to adapt the spatial mesh such that nodes are efficiently placed.
    - Static moving mesh.
    - Static grid refinement.
    - Dynamic mesh.

# Equidistribution Principle

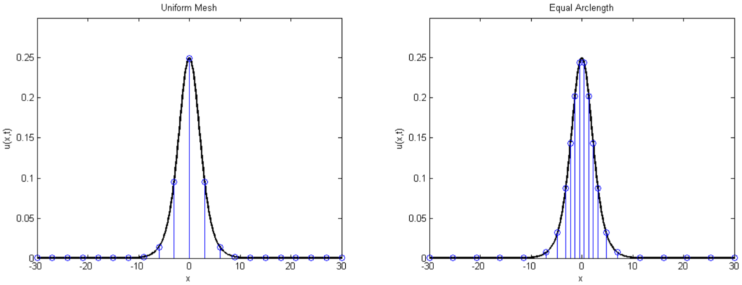
**Tracking the action**

- We want to assign nodes to areas with high activity.
- Use some monitor function $m(x)$ to measure the activity in a region.
- Choose the $x_i$ such that for all $i$,
$$\int_{x_{i-1}}^{x^i} m(x)\,\mathrm{d}x = \int_{x_i}^{x^{i+1}} m(x)\,\mathrm{d}x \text{ (equidistribution).}$$
- In practice equidistribution is considered optimal, but some suboptimal methods may be used to have roughly equal distribution in order to reduce computational effort.

# Choice of Monitor Function

- Arc length $m(x) = \sqrt{\alpha + u_x^2(x)}$.
- Local curvature $m(x) = |u_{xx}(x)|$.
- Other options involving many tuning parameters exist.
- Since the true solution is not known, derivatives must be estimated. Can use natural splines to interpolate current solution and use its derivatives as approximation.

# Uniform vs Equidistributed Arclengths (n=21)

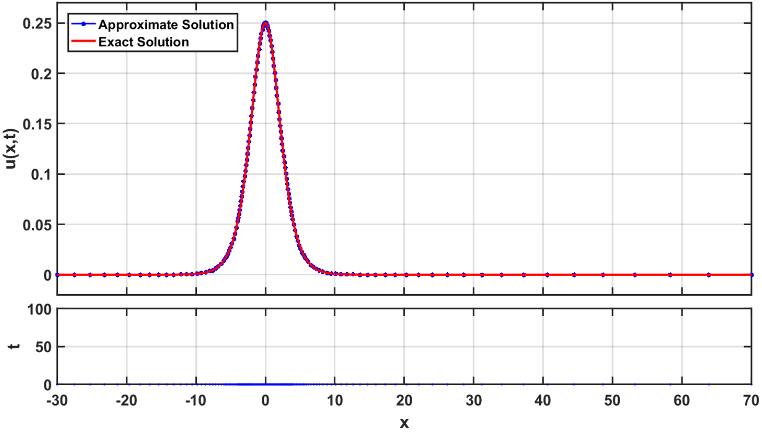


# Moving Mesh

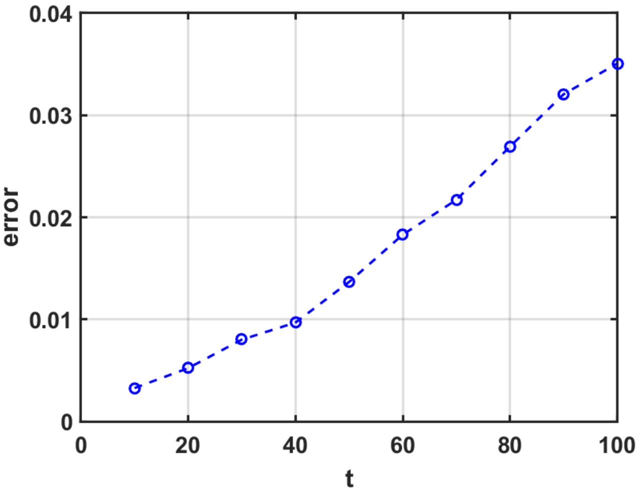
- Fix the number of nodes.
- Pause the PDE solver at a set interval to move the mesh nodes to achieve equidistribution.
- Interpolate the solution from old mesh to generate initial conditions for new mesh.

# Moving Mesh - KDV Equation Example 1

| Mesh Adaptions | # of Mesh Nodes | Computation Time |
|---|---|---|
| 10 | 151 | 8.9 sec |

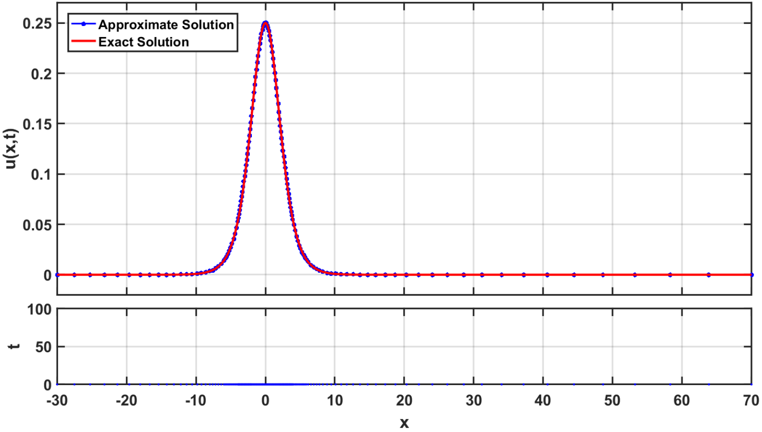


# Moving Mesh - KDV Equation Example 1 Error

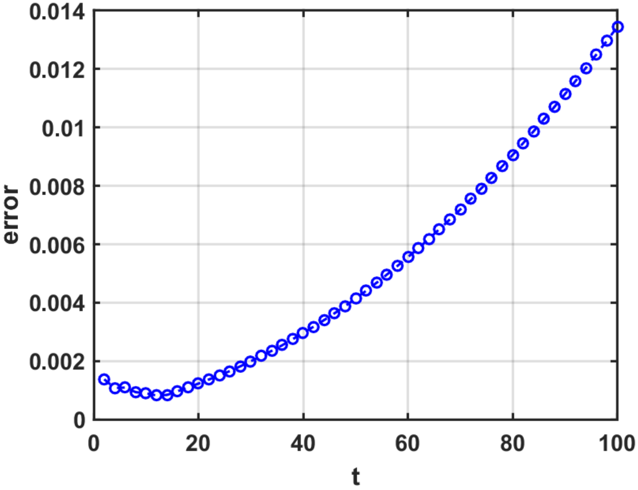


# Moving Mesh - KDV Equation Example 2

| Mesh Adaptions | # of Mesh Nodes | Computation Time |
|---|---|---|
| 50 | 151 | 19.6 sec |

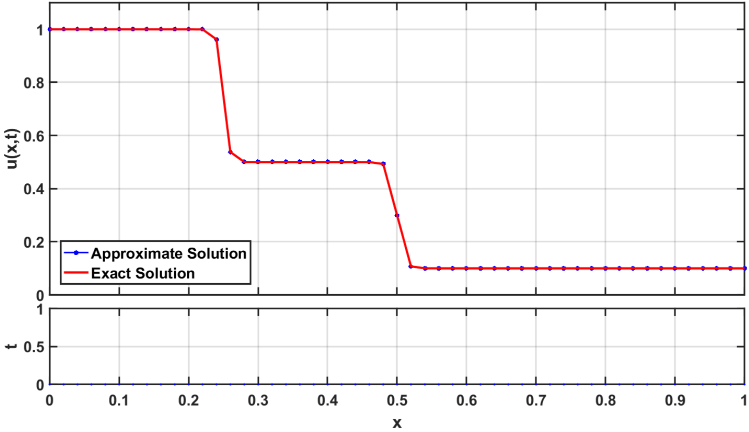


# Moving Mesh - KDV Equation Example 2 Error

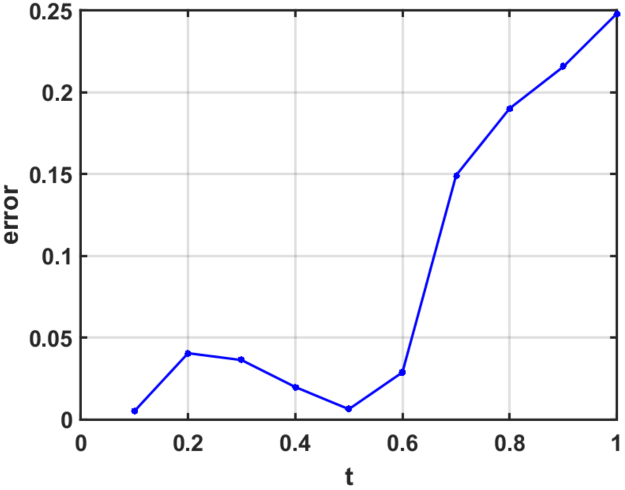


# Moving Mesh - Burgers' Equation Example 1

| Mesh Adaptions | # of Mesh Nodes | Computation Time |
|---|---|---|
| 10 | 375 | 5.8 sec |

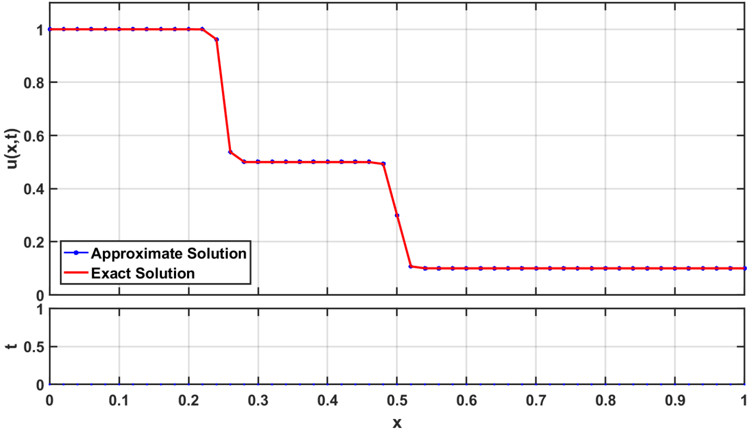


# Moving Mesh - Burgers' Equation Example 1 Error

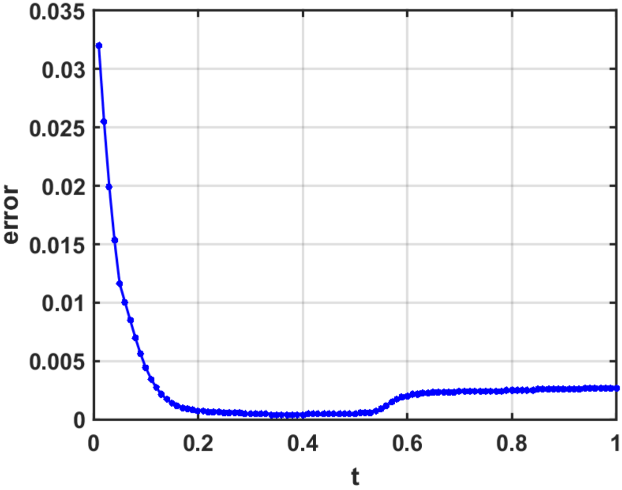


# Moving Mesh - Burgers' Equation Example 2

| Mesh Adaptions | # of Mesh Nodes | Computation Time |
|---|---|---|
| 100 | 251 | 36.1 sec |



# Moving Mesh - Burgers' Equation Example 2 Error



# Mesh Refinement

- Just as in moving mesh, the mesh is updated at set intervals.
- Procedure for each mesh adaption:
  - Reset to uniform base grid.
  - Evaluate monitor function between each pair of nodes.
  - If threshold exceeded, add nodes with uniform spacing
  - If below threshold, remove some nodes.
  - Interpolate and repeat.

# Mesh Refinement

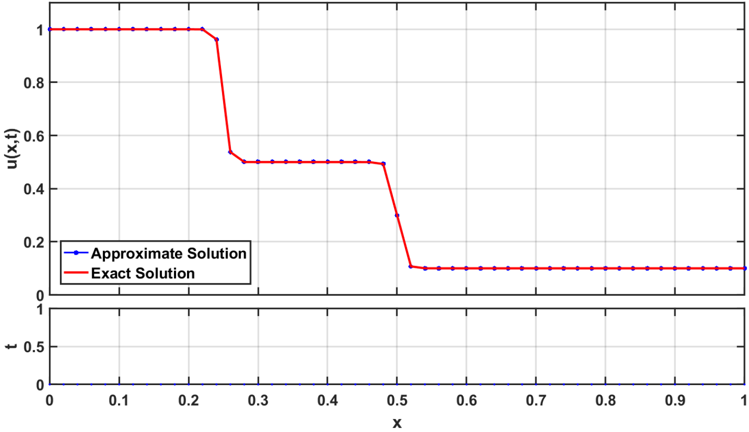
- Just as in moving mesh, the mesh is updated at set intervals.
- Procedure for each mesh adaption:
  - Reset to uniform base grid.
  - Evaluate monitor function between each pair of nodes.
  - If threshold exceeded, add nodes with uniform spacing
  - If below threshold, remove some nodes.
  - Interpolate and repeat.

# Mesh Refinement

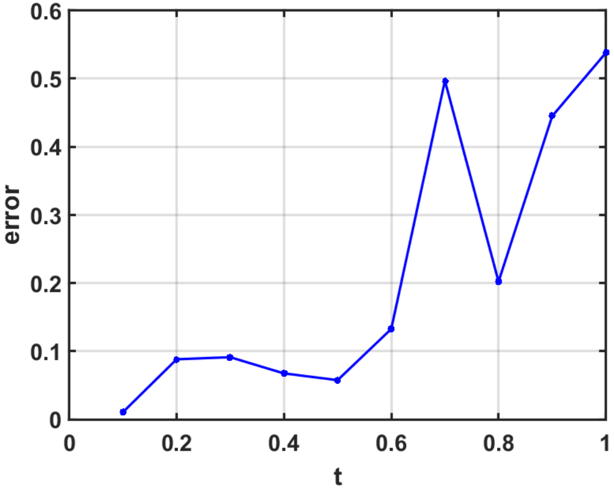
- Just as in moving mesh, the mesh is updated at set intervals.
- Procedure for each mesh adaption:
  - Reset to uniform base grid.
  - Evaluate monitor function between each pair of nodes.
  - If threshold exceeded, add nodes with uniform spacing
  - If below threshold, remove some nodes.
  - Interpolate and repeat.

# Mesh Refinement - Burgers' Equation Example 1

| Mesh Adaptions | # of Mesh Nodes | Computation Time |
|---|---|---|
| 10 | $\approx 268.5$ | 6.0 sec |

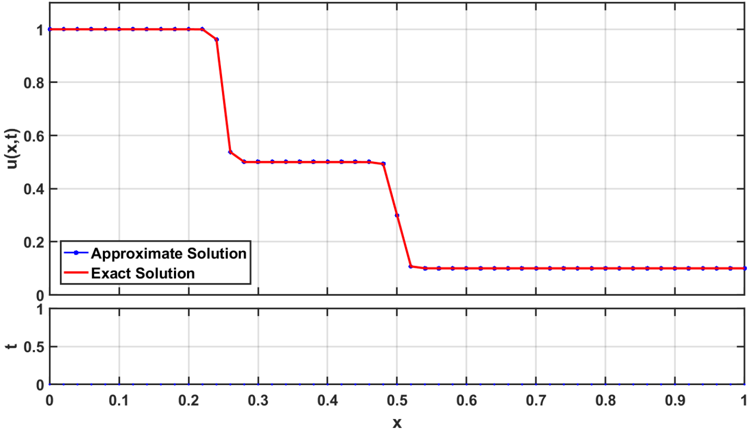


# Mesh Refinement - Burgers' Equation Example 1 Error

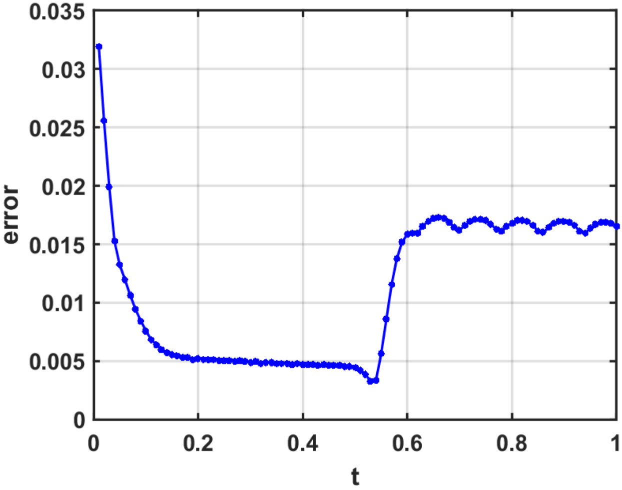


# Mesh Refinement - Burgers' Equation Example 2

| Mesh Adaption | # of Mesh Nodes | Computation Time |
|---|---|---|
| 100 | $\approx 152.5$ | 41.2 sec |



# Mesh Refinement - Burgers' Equation Example 2 Error

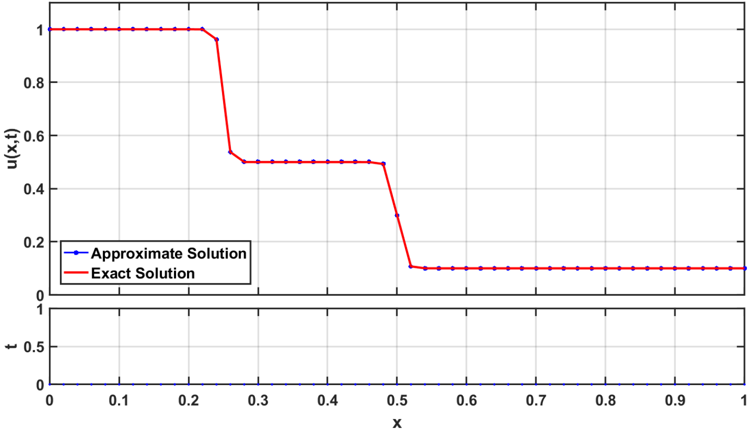


# Dynamic Method

- Refining the grid at set time steps can cause the mesh to lag behind the moving solution.
- Refining too often is computationally expensive and negates any benefits from adaptive mesh.
- In dynamic refinement methods, positions of nodes are functions of time.
- May be possible to use a priori information to generate functions for the wave to travel along. For example the characteristic curves along which the solution is constant. For a simple moving wave this tracks positions on the wave-front.
- Requires specific a priori knowledge of the problem, not always possible to derive characteristic curves.

# Dynamic Method - Burgers' Equation Example

| Mesh Adaptions | # of Mesh Points | Computation Time |
|---|---|---|
| 10 | 51 | 15.4 sec |



# Summary

- Problems arise when uniform grid used in Method of Lines.
- Solution: adapt the grid to account for regions of high activity.
- Static adaptation: refine the grid at discrete time steps.
- Dynamic adaptation: positions of nodes are functions of time.

This is the last slide.

Any questions?