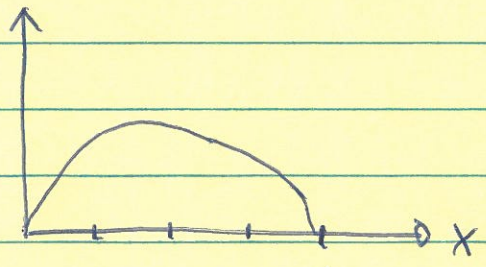
Lecture 16

Mesh adaptation — Equidistribution principle (I)

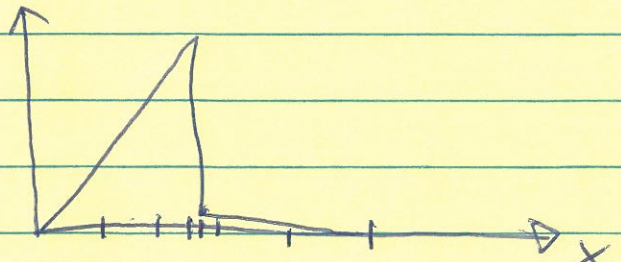
- Basic idea
- An optimization problem
- Equidistributing mesh
- ~~regularization of the monitor function~~
- ~~Interpolation error bounds~~

Basic idea of mesh adaptation

to place more points in regions where the error is large (or the solution has large variation) and less points in regions where the error is small (or the solution is smooth).



a uniform mesh is good for smooth functions



an adaptive mesh is needed for functions that have non-smooth regions

Q: Mathematical description of the basic idea?

An optimization problem

Problem. Consider a function $u = u(x)$ defined on $[0,1]$ and a partition

$$x_0 = 0 < x_1 < \cdots < x_J = 1$$

is given. Assume that we use interpolation

$$I_h u(x) = \sum_{j=0}^{J} u_j \varphi_j(x)$$

to approximate $u(x)$. For a given J, we want to known the optimal distrib. of mesh nodes $x_1, x_2, \ldots, x_{J-1}$ such that the interpolation error

$$e_h(x) = u(x) - I_h u(x)$$

has the smallest ~~b~~upper bound (in the ~~maximum~~ L^2 norm).

Solution

Recall from Lecture 13 that

$$|u(x) - I_h u(x)| \le \frac{h_j^2}{8} \max_{x_{j-1} \le y \le x_j} |u''(y)|$$

$$\forall\ x \in (x_{j-1}, x_j)$$

Then, $$\|e_h\|^2_{L^2(0,1)} = \sum_{j=1}^{J} \int_{x_{j-1}}^{x_j} |e_h|^2 \, dx$$

$$\le \sum_{j=1}^{J} \frac{h_j^5}{64} \max_{x_{j-1} \le y \le x_j} |u''(y)|^2$$

If the mesh is uniform, we have

$$\|e_h\|^2_{L^2(0,1)} \le \frac{h^4}{64} \max_{0\le y\le 1} |u''(y)|^2$$

or $$\|e_h\|_{L^2(0,1)} \le \frac{h^2}{8} \max_{0\le y\le 1} |u''(y)|$$

$$= \frac{1}{8J^2} \max_{0\le y\le 1} |u''(y)|$$

Infact, if we use a more careful estimate for the interpolation error on $[x_{j-1}, x_j]$, we can get

$$\|e_h\|_{L^2(0,1)} \le \frac{C}{J^2} \|u''\|_{L^2(0,1)}$$

Now we want to see what the optimal distribution of $x_1, \cdots, x_{J-1}$ is to have a smallest upper bound of the error,

$$E(x_1, \cdots x_{J-1}) = \sum_{j=1}^{J} \frac{h_j^5}{64} \max_{x_{j-1}\le y\le} |u''(y)|^2$$

We may want to do:

$$\min_{x_1, \cdots x_{J-1}} E(x_1, \cdots, x_{J-1})$$

But this direct optimization is hard, if not impossible to solve.

An indirect solution procedure.
Guess what the solution is.

Equidistributing mesh

Mesh satisfying

$$h_j \left[\max_{x_{j-1} \le y \le x_j} |u''(y)| \right]^{\frac{2}{5}} = \frac{\sigma_h}{J}$$

$j = 1, 2, \ldots, J$

where $\sigma_h = \sum_{j=1}^{J} h_j \left[\max_{x_{j-1} \le y \le x_j} |u''(y)| \right]^{2/5}$

$\sim \int_0^1 |u''(x)|^{\frac{2}{5}} dx$ Riemann Sum as $\max_j h_j \to 0$

For this mesh,

$$E(x_1, \ldots, x_J) = \frac{1}{64} \sum_{j=1}^{J} \left(\frac{\sigma_h}{J} \right)^5$$

$$= \frac{1}{64} \frac{\sigma_h^5}{J^4}$$

Thus $\|e_h\|_{L^2(0,1)} \le \left(\frac{1}{64} \frac{\sigma_h^5}{J^4} \right)^{1/2} = \frac{\sigma_h^{5/2}}{8 J^2}$

$$\sim \frac{1}{8J^2} \left(\int_0^1 |u''(x)|^{\frac{2}{5}} dx \right)^{5/2}$$

Summary:

On a uniform mesh

$$\|e_h\|_{L^2(0,1)} \leq \frac{C}{J^2}\left(\int_0^1 |u''|^2 dx\right)^{1/2}$$

On an equidistributing mesh

$$\|e_h\|_{L^2(0,1)} \leq \frac{C}{J^2}\left(\int_0^1 |u''|^{\frac{2}{5}} dx\right)^{5/2}$$

$$h_j \left(\max_{x_j \leq y \leq x_{j+1}} |u''(y)|\right)^{\frac{2}{5}} = \frac{\sigma_h}{J}$$

$j = 1, \ldots, J$

Known:

$$\left(\int_0^1 |u''|^{\frac{2}{5}} dx\right)^{5/2} \approx \left(\int_0^1 |u''|^2 dx\right)^{1/2}$$

when u'' is smooth

$$\left(\int_0^1 |u''|^{\frac{2}{5}} dx\right)^{5/2} \ll \left(\int_0^1 |u''|^2 dx\right)^{1/2}$$

when u'' is not smooth

Example

$$u'' = \frac{1}{x^{\frac{5}{3}}} : \quad \int_0^1 |u''|^{\frac{2}{5}} dx < +\infty$$

$$\int_0^1 |u''|^2 dx = +\infty.$$

#

Lecture 17

Mesh adaptation — Equidistribution principle (II)

- Regularization
- Linear interpolation error in H^1 semi-norm
- Application to Numerical solution of TPBVPs.

Equidistributing mesh: (recall)

L^2 norm of the interpolation error

$$\|e_h\|^2_{L^2(\Omega)} \le C \sum_{j=1}^{J} h_j^5 \max_{x_{j-1}\le y\le x_j} |u''(y)|^2$$

$$= C \sum_{j=1}^{J} \left[h_j \left(\max_{x_{j-1}\le y\le x_j} |u''(y)| \right)^{2/5} \right]^5$$

$$= C \frac{\sigma_h^5}{J^4}$$

$$\approx \frac{C}{J^4} \left(\int_0^1 |u''(x)|^{\frac{2}{5}} dx \right)^{5/2}$$

$$h_j \left(\max_{x_{j-1}\le y\le x_j} |u''(y)| \right)^{2/5} = \frac{\sigma_h}{J} \qquad j = 1, \ldots, J$$

Regularization of the monitor function

If for some j

$$\max_{x_{j-1} \le y \le x_j} |u''(y)| = 0$$

then the mesh does not exist.

Thus we need to regularize the bound:
add a small $\alpha_h > 0$:

$$\|e_h\|^2_{L^2(\Omega)} \le C \sum_{j=1}^{J} h_j^5 \left(\alpha_h + \max_{x_{j-1} \le y \le x_j} |u''(y)|^2\right)$$

$$= C \sum_{j=1}^{J} \left[h_j \left(\alpha_h + \max_{x_{j-1} \le y \le x_j} |u''(y)|^2\right)^{1/5}\right]^5$$

$$= C \frac{\sigma_h^5}{J^4}$$

$$\approx \frac{C}{J^4} \left[\int_0^1 \left(\alpha_h + |u''(x)|^2\right)^{1/5} dx\right]^5$$

$$h_j \left(\alpha_h + \max_{x_{j-1} \le y \le x_j} |u''(y)|^2\right)^{1/5} = \frac{\sigma_h}{J}$$

$$\sigma_h = \sum_j h_j \left(\alpha_h + \max_{x_{j-1} \le y \le x_j} |u''(y)|^2\right)^{1/5}$$

$$\approx \int_0^1 \left(\alpha_h + |u''(x)|^2\right)^{1/5} dx$$

Choice of α_h: not too big and not too small

Suggestion:

$$\alpha_h = \left(\sum_j h_j \max_{x_{j-1} \le y \le x_j} |u''(y)|^{\frac{2}{5}} \right)^5$$

$$\sim \left(\int_0^1 |u''(x)|^{\frac{2}{5}} dx \right)^5$$

It can be shown that:

$$\sigma_h \simeq \int_0^1 \left(\alpha_h + |u''(x)|^{\frac{2}{5}} \right)^{\frac{1}{5}} dx$$

$$\le 2 \int_0^1 (u''(x))^{\frac{2}{5}} dx .$$

The equidistributing mesh can be written as:

$$h_j \rho_j = \frac{\sigma_h}{J} , \quad j = 1, \dots, J$$

$$\rho_j = \left(\alpha_h + \max_{x_{j-1} \le y \le x_j} |u''(y)|^2 \right)^{\frac{1}{5}}$$

is called the monitor function (error density)

$$\sigma_h = \sum_j h_j \rho_j$$

Equivalently:

$$\rho_j = \left(1 + \frac{1}{\alpha_h} \max_{x_{j-1} \le y \le x_j} |u''(y)|^2 \right)^{\frac{1}{5}} .$$

Continuous form: $x = x(\xi) : [0,1] \to [0,1]$

$\uparrow \Omega_c \qquad \Omega$

$x_j = x(\xi_j) \quad j = 0 \cdots J$

$\xi_j, j = 0, 1, \cdots, J$ a uniform mesh

$$h_j = x_j - x_{j-1} = \frac{x(\xi_j) - x(\xi_{j-1})}{\Delta\xi} \cdot \Delta\xi$$

$$= \frac{dx}{d\xi}(\xi_j)\,\Delta\xi \qquad \Delta\xi \propto \frac{1}{J}$$

$$h_j \rho_j \sim \frac{dx}{d\xi}\rho(x)\,\Delta\xi$$

$$\frac{dx}{d\xi}\rho(x) = \sigma$$

← Equidistribution principle

$$\Rightarrow \int_{\Omega_c} \frac{dx}{d\xi}\rho(x)\,d\xi = \sigma \int_{\Omega_c} d\xi$$

$$\sigma = \int_\Omega \rho(x)\,dx$$

$$\rho(x) = \left(1 + \frac{1}{\alpha}|u''(x)|^2\right)^{\frac{1}{5}}$$

$$\alpha = \left(\int_0^1 |u''(x)|^{\frac{2}{5}}\,dx\right)^5$$

Linear

Interpolation error in H^1 semi-norm

$$\|e_h'\|^2_{L^2(\Omega)} = \sum_j \int_{x_{j-1}}^{x_j} |e_h'|^2 dx$$

$$\leq C \sum_j h_j^3 \max_{x_{j-1}\leq y\leq x_j} |u''(y)|^2$$

$$\leq C \sum_j h_j^3 \left[\alpha_h + \max_{x_{j-1}\leq y\leq x_j} |u''(y)|^2\right]$$

$$\leq C\alpha_h \sum_j \left[h_j \left(1 + \frac{1}{\alpha_h} \max_{x_{j-1}\leq y\leq x_j} |u''(y)|^2\right)^{1/3}\right]^3$$

$$\rho_j = \left(1 + \frac{1}{\alpha_h} \max_{x_{j-1}\leq y\leq x_j} |u''(y)|^2\right)^{1/3}$$

$$h_j \rho_j = \frac{\sigma_h}{J}, \quad \sigma_h = \sum_j h_j \rho_j \approx \int_0^1 \left(1 + \frac{1}{\alpha} |u''(x)|^2\right)^{1/3} dx$$

$$\alpha_h = \left(\sum_j h_j \max_{x_{j-1}\leq y\leq x_j} |u''(y)|^{2/3}\right)^3$$

$$\sim \left(\int_0^1 |u''|^{\frac{2}{3}} dx\right)^3$$

$$\|e_h'\|^2_{L^2(\Omega)} \leq C\alpha_h \sum_j \left(\frac{\sigma_h}{J}\right)^3$$

$$= C\alpha_h \frac{\sigma_h^3}{J^2}$$

$$\|e_h'\|_{L^2(\Omega)} \leq \frac{C}{J} \left(\int_0^1 |u''|^{\frac{2}{3}} dx\right)^{3/2} = O\left(\frac{1}{J}\right).$$

Continuous form:

$$\begin{cases} \frac{dx}{d\xi}\rho(x) = \sigma \\ \sigma = \int_0^1 \rho(x)\,dx \\ \rho(x) = \left(1 + \frac{1}{\alpha}|u''(x)|^2\right)^{1/3} \\ \alpha = \left(\int_0^1 |u''|^{\frac{2}{3}}\,dx\right)^3 \end{cases}$$

▶ Application to Adaptive mesh solution

$$\begin{cases} -\varepsilon \frac{d^2u}{dx^2} + u\frac{du}{dx} = f(x) \\ u(0) = 0,\ u(1) = 0 \end{cases}$$

$x = x(\xi)$:

Change of variables: $u = u(x(\xi))$

$$\frac{du}{d\xi} = \frac{du}{dx}\cdot\frac{dx}{d\xi} \quad \text{(chain rule)}$$

$$\Rightarrow \frac{du}{dx} = \frac{du}{d\xi}\,\frac{1}{\frac{dx}{d\xi}}$$

$$\frac{d^2u}{dx^2} = \frac{d}{dx}\left(\frac{1}{x_\xi}\frac{du}{d\xi}\right)$$

$$= \frac{1}{x_\xi}\frac{d}{d\xi}\left(\frac{1}{x_\xi}\frac{du}{d\xi}\right)$$

$$-\frac{\varepsilon}{x_\xi}\frac{d}{d\xi}\left(\frac{1}{x_\xi}\frac{du}{d\xi}\right)+\frac{u}{x_\xi}\frac{du}{d\xi}=f(x(\xi))$$

$u=u(\xi)$ and $x=x(\xi)$ are unknown functions.

Define $x=x(\xi)$:

$$\begin{cases}\frac{dx}{d\xi}\rho(x(\xi))=\sigma \\ \rho(x)=\left(1+\frac{1}{\alpha}|u''(x)|^2\right)^{1/5}\end{cases}$$

$$\Rightarrow \quad \begin{cases}\frac{d}{d\xi}\left(\rho\frac{dx}{d\xi}\right)=0 \\ x(0)=0,\ x(1)=1\end{cases}$$

Coupled system:

$$\begin{cases}-\frac{\varepsilon}{x_\xi}\frac{d}{d\xi}\left(\frac{1}{x_\xi}\frac{du}{d\xi}\right)+\frac{u}{x_\xi}\frac{du}{d\xi}=f(x(\xi)) \\ \frac{d}{d\xi}\left(\rho\frac{dx}{d\xi}\right)=0 \\ u(0)=0,\ u(1)=0 \\ x(0)=0,\ x(1)=1\end{cases}$$

#

Lecture 18

Adaptive moving mesh PDE methods for time dependent problems

- moving meshes
- transformed PDEs
- MMPDE approach
- Numerical results

Moving meshes

For time dependent problems, it is natural to use a time dependent mesh:

$$x_0(t) \equiv 0 < x_1(t) < \cdots < x_J(t) \equiv 1$$

It is convenient to view this mesh as the image of a fixed, uniform mesh under a coordinate transformation. Denote the coordinate transformation by

$$x = x(\xi, t) : [0,1] \to [0,1].$$

$$\xi_j = \frac{j}{J}, \quad j = 0, 1, \ldots, J$$

$x_j(t) = x(\xi_j, t)$.

▶ transformed PDEs

Consider the Burgurs equation

$$\begin{cases} u_t + uu_x = \varepsilon u_{xx} & x \in (0,1) \\ u(0,t)=0,\ u(1,t)=0 \end{cases}$$

We want to transform this equation from (x,t) to (ξ,t) under $x = x(\xi,t)$.

$$v(\xi,t) = u(x(\xi,t),t)$$

$$\frac{\partial v}{\partial \xi} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} \iff v_\xi = u_x x_\xi$$

$$\dot{v} = \frac{\partial v}{\partial t} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial u}{\partial t} \iff \dot{v} = u_x \dot{x} + u_t$$

$$\begin{cases} u_t = \dot{v} - u_x \dot{x} = \dot{v} - \dfrac{\dot{x}}{x_\xi} v_\xi \\ u_x = \dfrac{v_\xi}{x_\xi} \end{cases}$$

$$\begin{cases} \dot{v} + (v - \dot{x})\dfrac{v_\xi}{x_\xi} = \dfrac{\varepsilon}{x_\xi}\dfrac{\partial}{\partial \xi}\left(\dfrac{v_\xi}{x_\xi}\right) & \xi \in (0,1) \\ v(0,t)=0,\ v(1,t)=0 \end{cases}$$

▶ <u>mesh equation</u>

How to determine $x = x(\xi,t)$?

Equidistribution principle.

$$\frac{\partial x}{\partial \xi}\rho(x) = \sigma$$

$$\rho(x) = (1 + |u'(x)|^2)^{1/5}$$

$$\sigma = \int_0^1 \rho(x)\,dx$$

$$\Rightarrow \begin{cases} \frac{\partial}{\partial \xi}\left(\rho(x)\frac{\partial x}{\partial \xi}\right) = 0 \\ x(0,t) = 0 \\ x(1,t) = 1 \end{cases}$$

Coupled system

$$\begin{cases} \dot{v} + (v - \dot{x})\frac{v_\xi}{x_\xi} = \frac{\varepsilon}{x_\xi}\frac{\partial}{\partial \xi}\left(\frac{1}{x_\xi} v_\xi\right) \\ \frac{\partial}{\partial \xi}\left(\rho(x)\frac{\partial x}{\partial \xi}\right) = 0 \\ v(0,t) = 0 \quad v(1,t) = 0 \\ x(0,t) = 0 \quad x(1,t) = 1 \end{cases}$$

Central FiD discretization on ξ_j, $j = 0, 1, \dots, J$

$$\begin{cases} \dot{v}_j + (v_j - \dot{x}_j)\frac{v_{j+1} - v_{j-1}}{x_{j+1} - x_{j-1}} = \frac{2\varepsilon}{x_{j+1} - x_{j-1}}\left[\frac{v_{j+1} - v_j}{x_{j+1} - x_j} - \frac{v_j - v_{j-1}}{x_j - x_{j-1}}\right] \\ \qquad j = 1, 2, \dots, J-1 \quad (**) \\ \rho_{j+1/2}(x_{j+1} - x_j) - \rho_{j-1/2}(x_j - x_{j-1}) = 0 \quad (*) \\ \qquad j = 1, \dots, J-1 \\ v_0(t) = 0, \quad v_J(t) = 0 \\ x_0(t) = 0, \quad x_J(t) = 1 \end{cases}$$

DAEs not easy to solve.

MMPDE approach

$$\frac{\partial}{\partial \xi}\left(\rho \frac{\partial x}{\partial \xi}\right) = 0$$

$$\Rightarrow \quad \dot{x} = \frac{1}{\tau}\frac{\partial}{\partial \xi}\left(\rho(x)\frac{\partial x}{\partial \xi}\right)$$

$\tau > 0$: parameter used to control how ~~fast~~ quick the mesh movement ~~co~~responds to the changes in ρ.

τ large: slow movement

τ small quick movement

typically: $\tau = 10^{-1} \sim 10^{-3}$.

$$\begin{cases} \dot{v}_j + (v_j - \dot{x}_j)\dfrac{v_{j+1}-v_{j-1}}{x_{j+1}-x_{j-1}} = \dfrac{2\varepsilon}{x_{j+1}-x_{j-1}}\left[\dfrac{v_{j+1}-v_j}{x_{j+1}-x_j} - \dfrac{v_j - v_{j-1}}{x_j - x_{j-1}}\right] \\ \qquad j = 1, \cdots, J-1 \\ \dot{x}_j = \dfrac{1}{\tau}\dfrac{1}{\Delta\xi^2}\left[\rho_{j+\frac{1}{2}}(x_{j+1}-x_j) - \rho_{j-\frac{1}{2}}(x_j - x_{j-1})\right] \\ \qquad j = 1, \cdots, J-1 \\ v_0 = 0, \; v_J = 0 \\ x_0 = 0, \; x_J = 1 \end{cases}$$

$$\rho_{j+\frac{1}{2}} = \frac{1}{2}(\rho_j + \rho_{j+1})$$

$$\rho_j = \left(1 + \frac{1}{\alpha^2}\left|\frac{v_{j+1}-v_j}{x_{j+1}-x_j}\left(\frac{v_{j+1}-v_j}{x_{j+1}-x_j} - \frac{v_j - v_{j-1}}{x_j - x_{j-1}}\right)\right|^2\right)^{1/5}$$

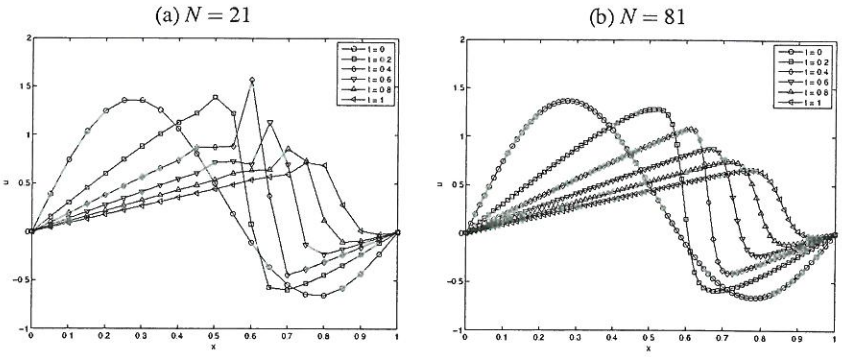


Fig. 1.1 Computed solutions obtained with a uniform mesh for Burgers' equation with $\varepsilon = 10^{-2}$ are shown at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0.

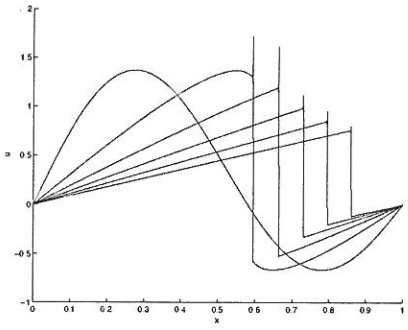


Fig. 1.2 Computed solutions at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0 obtained with a uniform mesh of 2001 points for Burgers' equation with $\varepsilon = 10^{-4}$.

solution computed with a uniform mesh of 2001 points for the case $\varepsilon = 10^{-4}$. Use of a very fine uniform mesh is expensive in terms of computer time and memory, and much more so for two- and three-dimensional problems, making mesh adaptation necessary.

1.2.2 Finite difference method on an adaptive moving mesh

The adaptive solution of the model problem requires that the mesh points be concentrated around the steep front and dynamically adjusted to follow the front as it propagates in time. Such a dynamically adjusting mesh is referred to as an *adaptive moving mesh*.

Adaptive mesh movement is often best understood by interpreting the problem in terms of a suitable coordinate transformation. Specifically, we assume for the

(a) Computed solution. (b) Mesh trajectories.

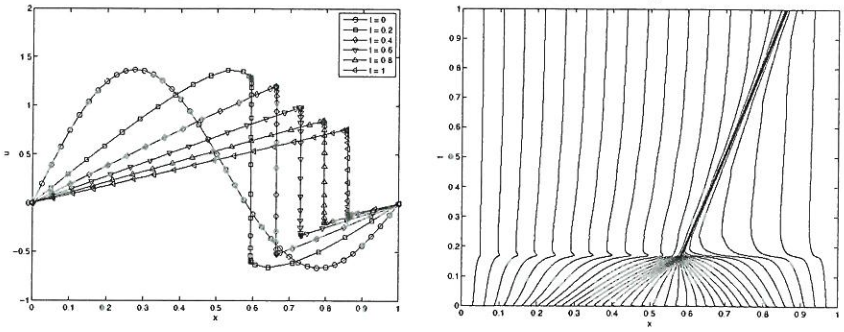


Fig. 1.3 (a) The computed solution for Burgers' equation with $\varepsilon = 10^{-4}$, obtained with an adaptive moving mesh of 41 points, is shown at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0. (b) The corresponding mesh trajectories.

(See Appendix A for the definition of Sobolev space $H^1(0,1)$.) For a given time $T > 0$, the Galerkin formulation of the model problem is the following: Find $u(\cdot,t) \in V$ for $0 < t \leq T$ such that

$$\int_0^1 u_t \phi \, dx = \int_0^1 \left(-\varepsilon u_x + \frac{1}{2} u^2 \right) \phi_x dx, \quad \forall \phi \in V, \ 0 < t \leq T \tag{1.27}$$

and the initial condition (1.3) holds. The equation (1.27) is obtained by multiplying (1.1) by ϕ, integrating both sides of the resulting equation over the interval $(0,1)$, and integrating by parts on the right-hand side.

We consider the linear finite element approximation on the uniform mesh $\mathcal{T}_h$ in (1.4). Specifically, define the basis functions (hat functions)

$$\phi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}}, & \text{for } x \in [x_{j-1}, x_j] \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j}, & \text{for } x \in [x_j, x_{j+1}] \\ 0, & \text{otherwise} \end{cases} \qquad j = 1, \ldots, N \tag{1.28}$$

and let V^h be the $(N-2)$-dimensional subspace of V spanned by the basis functions $\phi_2, \ldots, \phi_{N-1}$, i.e.,

$$V^h = \text{ span}\{\phi_2, \ldots, \phi_{N-1}\}.$$

A linear finite element approximation $u^h(\cdot,t) \in V^h$ for $0 < t \leq T$ to the exact solution u of the model problem is then required to satisfy

$$\int_0^1 u_t^h \phi \, dx = \int_0^1 \left(-\varepsilon u_x^h + \frac{1}{2} \left(u^h \right)^2 \right) \phi_x dx, \quad \forall \, \phi \in V^h, \ 0 < t \leq T \tag{1.29}$$

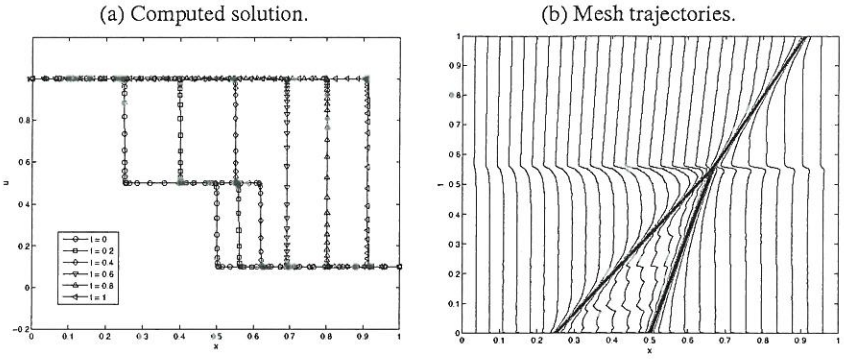


Fig. 1.7 (a) The finite difference solution for Burgers' equation with $\varepsilon = 10^{-4}$, obtained with an adaptive moving mesh of 61 points, is shown at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0. (b) The corresponding mesh trajectories.

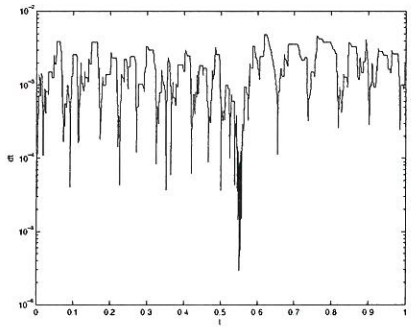


Fig. 1.8 The time step size used in the adaptive moving mesh solution of Burgers' equation with $\varepsilon = 10^{-4}$ and 61 points is plotted as function of time. The relative and absolute tolerances for the time step control are taken as $rtol = 10^{-6}$ and $atol = 10^{-4}$, respectively, for the Matlab ODE solver "ode15i" (using a backward differentiation formula of order 5).

The upper and lower bandwidths of B are equal to 2 plus the number of sweeps used in smoothing the mesh density function. Strategies for choosing the mesh density function and the intensity parameter are considered in detail in Chapter 2.

The difference in time required to solve algebraic systems for the uniform and adaptive mesh methods turns out to be less significant in two and three dimensions, where the algebraic systems for the uniform mesh method are generally much more expensive to solve.

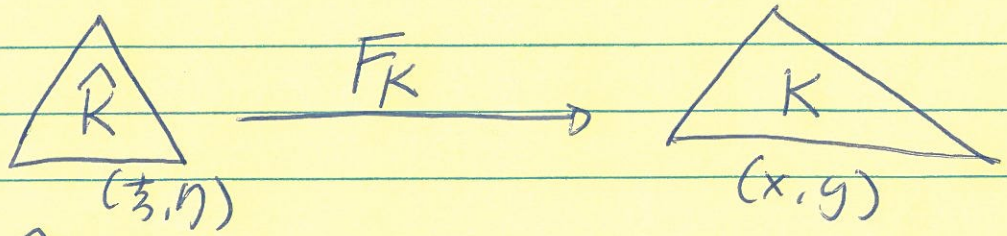
Lecture 19

Mesh adaptation in 2D

- reference element
- mesh adaptation in 2D

The reference element

A triangular mesh $\mathcal{T}_h = \{K : \text{triangles}\}$
$= \{K = F_K(\hat{K})\}$.



$|\hat{K}| = 1$, equilateral
or ~~unit~~ right triangle

$F_K : \hat{K} \to K$ affine (linear, invertible)

F_K' : Jacobian matrix

F_K': determines the shape, size, and orientation of K

Mesh adaptation in 2D:

- Mesh adaptation lies in the stability of controling the size, shape, and orientation of elements throughout the physical domain.
- Control element size, shape, and orientation through a matrix-valued function

$$M = M(x,y) \qquad (x,y) \in \Omega$$

$M(x,y)$: Monitor function/metric tensor
symmetric and positive definite

- M-uniform mesh approach:
An adaptive mesh is generated as a uniform mesh in the metric specified by $M = M(x,y)$.

It is shown that (H. and Russell, 2011)

an M-uniform mesh satisfies approximately

the alignment condition $\forall K \in \mathcal{T}_h$

$$\frac{1}{2} tr\left((F_K')^T M_K F_K' \right) = \det\left((F_K')^T M_K F_K' \right)^{1/2} \quad (*)$$

and the equidistribution condition

$$|K| \det(M_K)^{1/2} = \frac{\sigma_h}{N} \quad \forall K \in \mathcal{T}_h \quad (**)$$

where: $M_K = \frac{1}{|K|} \int_K M(x,y)\, dx\, dy$

N: total # of elements of $\mathcal{T}_h$

$$\sigma_h = \sum_K |K| \det(M_K)^{1/2}$$

- $(**)$ is the multidimensional generalization of the equidistribution principle. It determines the size of elements through $M = M(x,y)$
- $(*)$: Alignment condition

Let λ_1, λ_2 be eigenvalues of $(F_K')^T M_K F_K'$:

$$\frac{1}{2} tr(\cdot\cdot) = \frac{\lambda_1 + \lambda_2}{2}$$

$$\det(\cdot\cdot)^{1/2} = \sqrt{\lambda_1 \lambda_2}$$

alignment condition $\Rightarrow \lambda_1 = \lambda_2 = \theta_K$

$\Rightarrow$ $(F_K')^T M_K F_K' = \theta_K I$ θ_K any value

or $(F_K')^{-T}(F_K')^{-1} = \frac{1}{\theta_K} M_K$

↳ metric of the inverse coordinate transformation F_K^{-1}

$\Rightarrow$ determines the shape and orientation of K.

- The two conditions have been used successfully in designing algorithms for mesh adaptation and generation, in understanding existing algorithms, and error analysis.
- Other approaches of mesh adaptation we mainly study

  r-version or moving mesh approach

  h-version — mesh refinement

  p-version — use polynomials of different degrees.

#