澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

1981–2021
不惑新航 揚帆追夢
SET SAIL ANEW ON THE RUBY JUBILEE

Finite Difference Methods for 1D Elliptic PDEs

MATH 3014
Monday & Thursday 14:30-15:45

Instructor: **Dr. Luo Li**
https://www.fst.um.edu.mo/personal/liluo/math3014/

Department of Mathematics
Faculty of Science and Technology

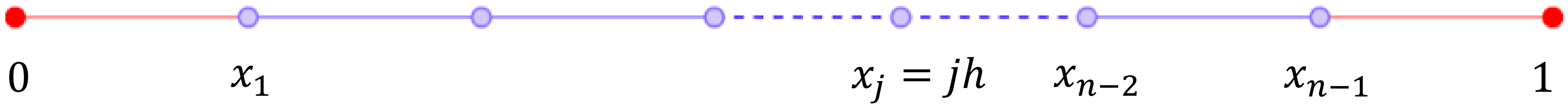
A Simple Example of a Finite Difference Method

$$u''(x) = f(x), \quad 0 < x < 1, \quad u(0) = u_a, \quad u(1) = u_b,$$

1. Generate a grid.

Discretize the domain [0, 1] by a uniform grid with spacing $h = \frac{1}{n}$.

The parameter n can be chosen according to accuracy requirement.



2. Represent the derivative by some finite difference formula

$$\phi''(x) = \lim_{\Delta x \to 0} \frac{\phi(x - \Delta x) - 2\phi(x) + \phi(x + \Delta x)}{(\Delta x)^2}$$

$$u''(x_i) \approx \frac{u(x_i - h) - 2u(x_i) + u(x_i + h)}{h^2}$$

In the finite difference method, we replace the differential equation at each grid point x_i by

$$\frac{u(x_i - h) - 2u(x_i) + u(x_i + h)}{h^2} = f(x_i) + error$$

where the error is called the *local truncation error*.

We define the finite difference (FD) solution (an approximation) for $u(x)$ at all x_i as the solution U_i

$$\frac{u_a - 2U_1 + U_2}{h^2} = f(x_1)$$

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} = f(x_i)$$

$$\frac{U_{n-2} - 2U_{n-1} + u_b}{h^2} = f(x_{n-1})$$

The set of x_{i-1}, x_i, and x_{i+1} is called the finite difference stencil.

3. Solve the system of algebraic equations. The system of algebraic equations can be written in the matrix and vector form

$$\begin{bmatrix} -\frac{2}{h^2} & \frac{1}{h^2} & & & & \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & & \\ & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\ & & & & \frac{1}{h^2} & -\frac{2}{h^2} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_{n-2} \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} f(x_1) - u_a/h^2 \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - u_b/h^2 \end{bmatrix} \qquad (2.1)$$

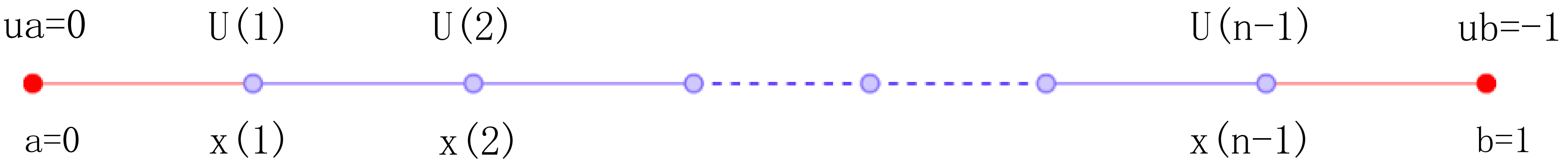
The tridiagonal system of linear equations above can be solved efficiently in $O(Cn)$ operations by the *Crout* or *Cholesky* algorithm.

4. Implement and debug the computer code. Run the program to get the output. Analyze and visualize the results (tables, plots, etc.).

5. Error analysis. Algorithmic consistency and stability implies convergence of the finite difference method.

Example

$$u''(x) = f(x), \qquad 0 < x < 1,$$
$$f(x) = -\pi^2 \cos(\pi x), \qquad u(0) = 1, \qquad u(1) = -1.$$



A Matlab Code for the Model Problem

```
function [x,U] = two_point(a,b,ua,ub,f,n)

h = (b-a)/n; h1=h*h;
A = sparse(n-1,n-1);
F = zeros(n-1,1);

for i=1:n-2,
      A(i,i) = -2/h1;
      A(i+1,i) = 1/h1;
      A(i,i+1)= 1/h1;
end
A(n-1,n-1) = -2/h1;
```

Form the matrix

```
for i=1:n-1,
      x(i) = a+i*h;
      F(i) = feval(f,x(i));
end

F(1) = F(1) - ua/h1;
F(n-1) = F(n-1) - ub/h1;

U = A\F;

return
```

Form the RHS

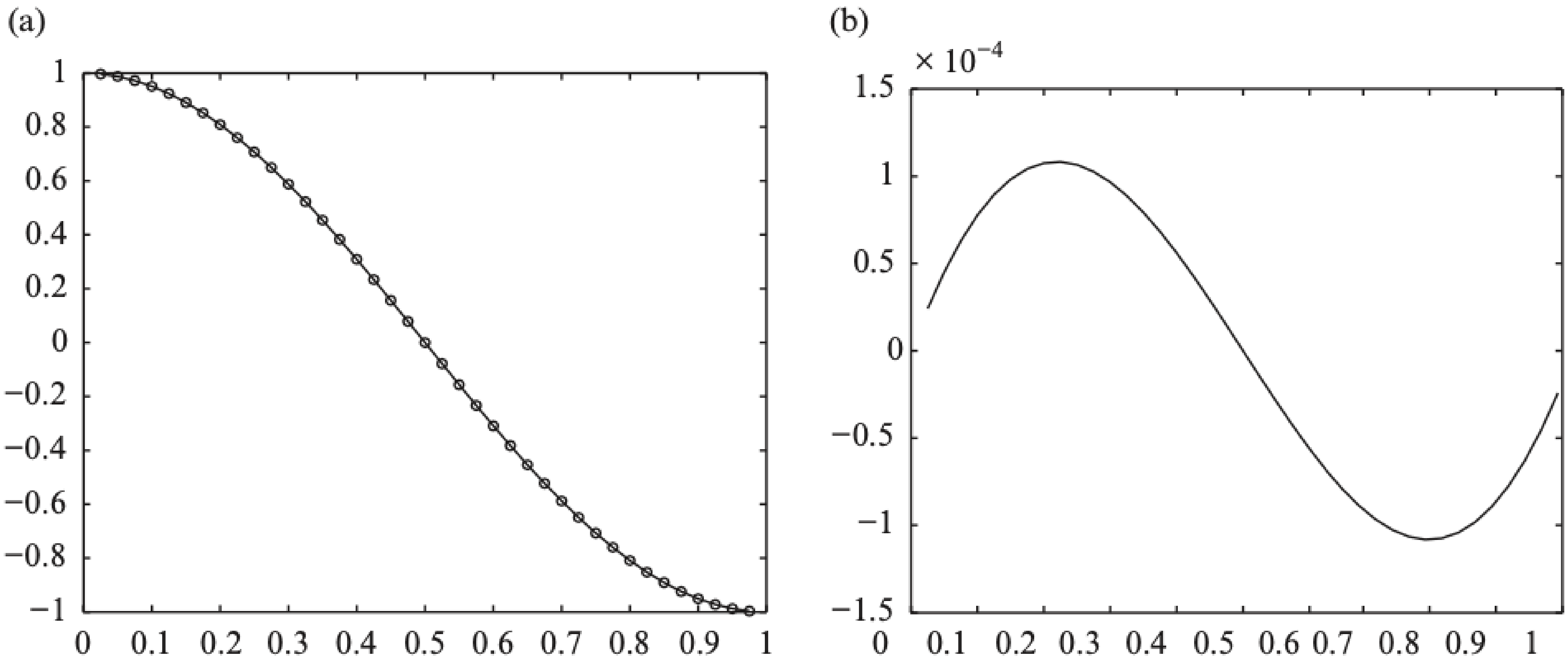


Figure 2.1. (a) A plot of the computed solution (little ‘o’s) with $n = 40$, and the exact solution (solid line). (b) The plot of the error.

Questions About This Example:

- How do we know whether a finite difference method works or not?

If it works, how accurate is it? Specifically, what is the error of the computed solution?

- How do we deal with boundary conditions other than Dirichlet conditions

(involving only function values) as above, notably Neumann conditions (involving derivatives) or mixed boundary conditions?

- Do we need different finite difference methods for different problems? If so, are the procedures similar?

- How do we know that we are using the most efficient method? What are the criteria, in order to implement finite difference methods efficiently?

Fundamentals of Finite Difference Methods

The Taylor expansion is the most important tool in the analysis of FDM:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \cdots + \frac{h^k}{k!}u^{(k)}(\xi)$$

where $x < \xi < x + h$

Forward, Backward, and Central Finite Difference Formulas for $u'(x)$ at a point $\bar{x}$:

Forward FD: $$\Delta_+ u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x})}{h} \sim u'(\bar{x}), \qquad (2.4)$$

Backward FD: $$\Delta_- u(\bar{x}) = \frac{u(\bar{x}) - u(\bar{x}-h)}{h} \sim u'(\bar{x}), \qquad (2.5)$$

Central FD: $$\delta u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x}-h)}{2h} \sim u'(\bar{x}). \qquad (2.6)$$

$$u'(\bar{x}) = \lim_{h\to 0} \frac{u(\bar{x}+h) - u(\bar{x})}{h}$$

"Close to but usually not exactly"

$$\Delta_+ u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x})}{h} \sim u'(\bar{x})$$

h: step size

The slope of the secant line that connects the two points $(\bar{x}, u(\bar{x}))$ and $(\bar{x}+h, u(\bar{x}+h))$

To determine how closely $\Delta_+ u(\bar{x})$ represents $u'(\bar{x})$

$$u(\bar{x}+h) = u(\bar{x}) + u'(\bar{x})h + \frac{1}{2}u''(\xi)\,h^2,$$ where $0 < \xi < h$

The error estimate: $$E_f(h) = \frac{u(\bar{x}+h) - u(\bar{x})}{h} - u'(\bar{x}) = \frac{1}{2}u''(\xi)h = O(h),$$

p-th order accurate: $$E(h) = Ch^p, \qquad p > 0$$

Forward finite difference

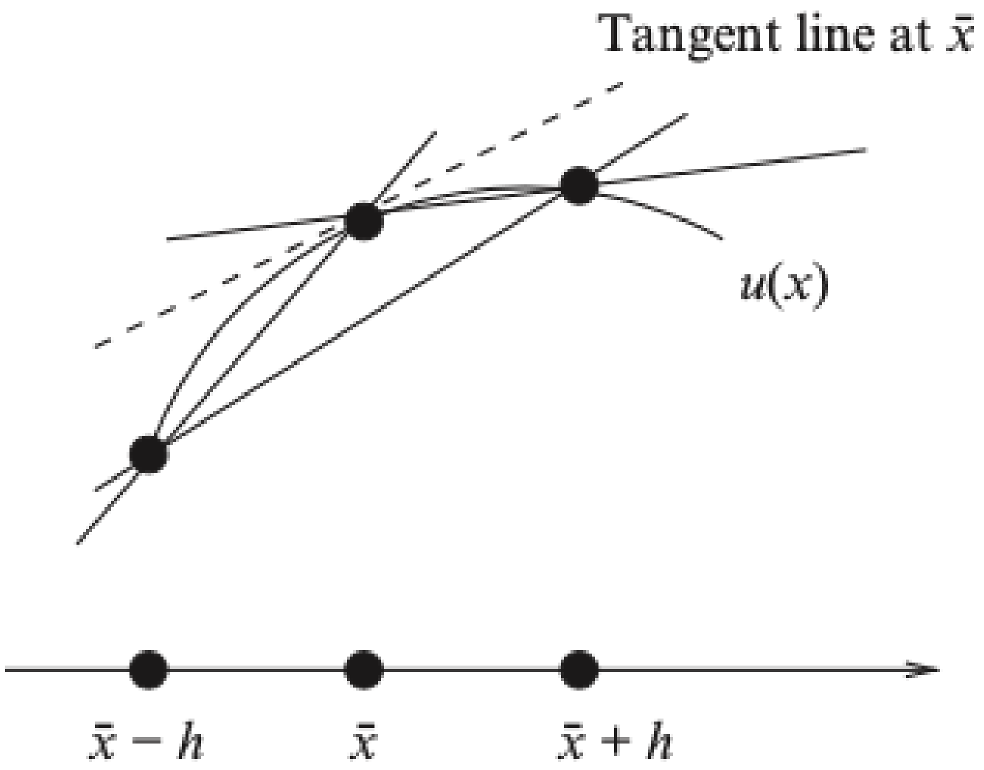
$$\Delta_+ u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x})}{h}$$

Backward finite difference

$$\Delta_- u(\bar{x}) = \frac{u(\bar{x}) - u(\bar{x}-h)}{h}$$

Central finite difference

$$\delta u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x}-h)}{2h}$$



Central Finite Difference Formula

From the Taylor expansion

$$u(x+h) = u(x) + hu'(x) + \frac{1}{2}u''(x)h^2 + \frac{1}{6}u'''(x)h^3 + \frac{1}{24}u^{(4)}(x)h^4 + \cdots ,$$

$$u(x-h) = u(x) - hu'(x) + \frac{1}{2}u''(x)h^2 - \frac{1}{6}u'''(x)h^3 + \frac{1}{24}u^{(4)}(x)h^4 + \cdots ,$$

Second-order accurate

$$E_c(h) = \frac{u(\bar{x}+h) - u(\bar{x}-h)}{2h} - u'(\bar{x}) = \frac{1}{6}u'''(\bar{x})h^2 + \cdots = O(h^2)$$

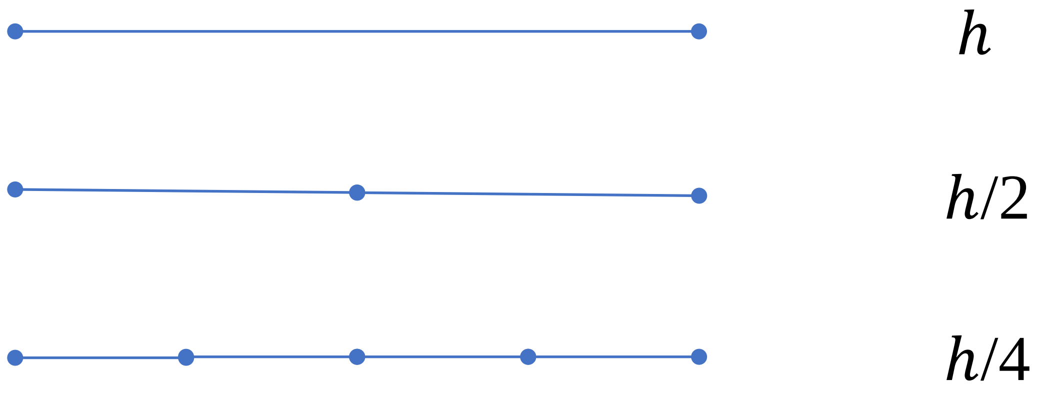
Relation with the forward, backward FD

$$\delta u(\bar{x}) = \frac{u(\bar{x}+h) - u(\bar{x}-h)}{2h} = \frac{1}{2}\left(\Delta_+ + \Delta_- \right) u(\bar{x})$$

Verification of the Method

How do we know that our code is bug-free and our analysis is correct?

------ Grid Refinement Analysis



- For a first-order method, the error should decrease by a factor of two $O(h)$
- For a second-order method the error should be decrease by a factor of four $O(h^2)$

Example Consider the function $u(x) = \sin x$ at $x = 1$, where the exact derivative is of course $\cos 1$.

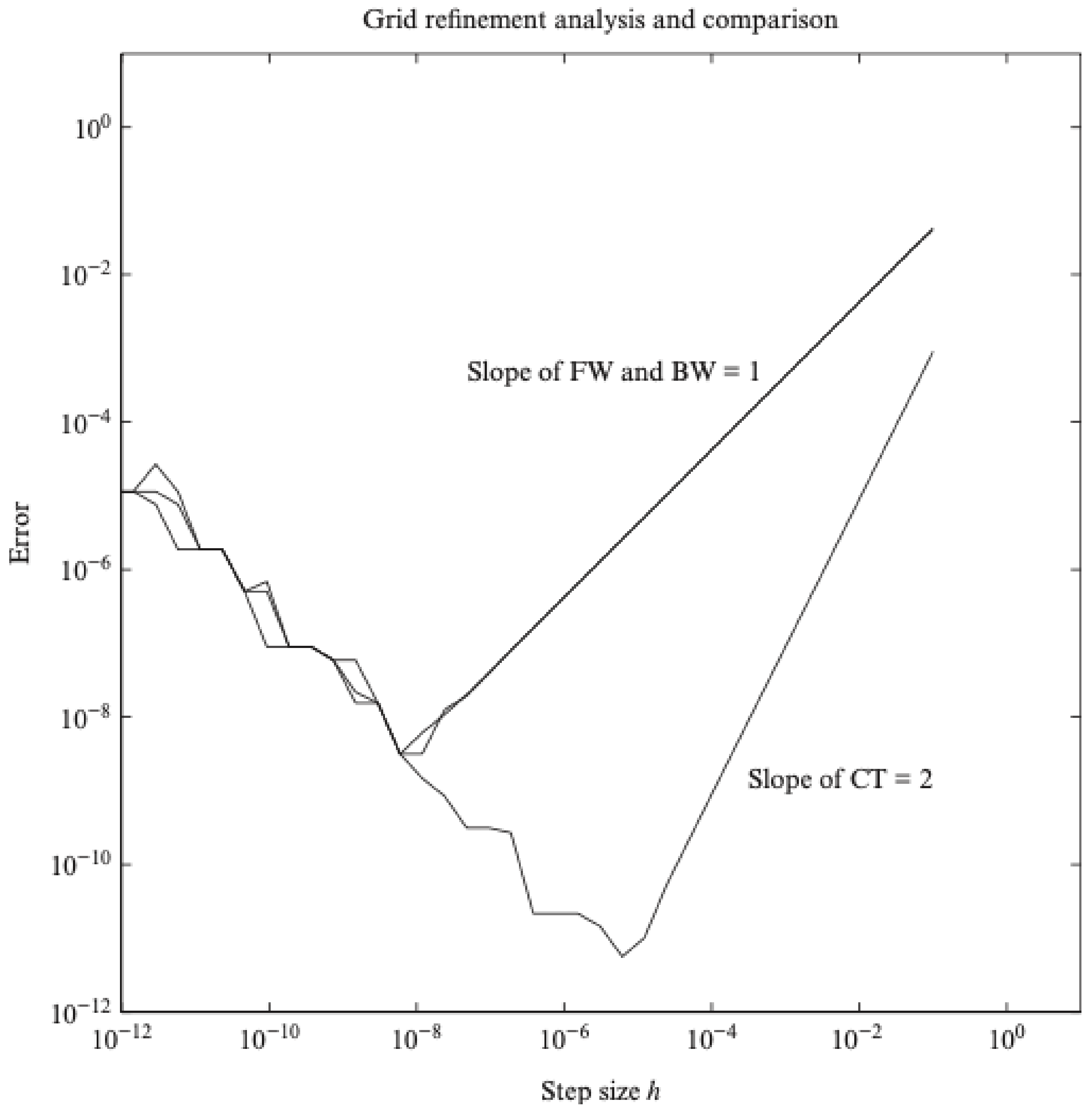
Forward (FW): $(\sin(1+h) - \sin(1))/h - \cos(1);$

Backward (BW): $(\sin(1) - \sin(1-h))/h - \cos(1);$

Central (CT): $\sin(1+h) - \sin(1-h))/(2h) - \cos(1);$

```
clear; close all
h = 0.1;
for i=1:5,
    a(i,1) = h;
    a(i,2) = (sin(1+h)-sin(1))/h - cos(1);
    a(i,3) = (sin(1) - sin(1-h))/h - cos(1);
    a(i,4) = (sin(1+h)-sin(1-h))/(2*h)- cos(1);
    h = h/2;
end
```

```
a = abs(a);
h1 = a(:,1);
e1 = a(:,2); e2 = a(:,3); e3 = a(:,4);
loglog(h1,e1,h1,e2,h1,e3)
axis('equal'); axis('square’)
axis([1e-6 1e1 1e-6 1e1])
gtext('Slope of FW and BW = 1')
gtext('Slope of CD =2’)
```



Grid refinement analysis and comparison
Error
Step size h
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}
Slope of FW and BW = 1
Slope of CT = 2

%	h	forward	backward	central
%	1.0000e-01	-4.2939e-02	4.1138e-02	-9.0005e-04
%	5.0000e-02	-2.1257e-02	2.0807e-02	-2.2510e-04
%	2.5000e-02	-1.0574e-02	1.0462e-02	-5.6280e-05
%	1.2500e-02	-5.2732e-03	5.2451e-03	-1.4070e-05
%	6.2500e-03	-2.6331e-03	2.6261e-03	-3.5176e-06

- As h gets smaller, round-off errors become evident and eventually dominant.

- The best h can be estimated by balancing the formula error and the round-off errors.

Deriving FD Formulas Using the Method of Undetermined Coefficients

Goal: To approximate a first derivative to second-order accuracy

$$u'(\bar{x}) \sim \gamma_1 u(\bar{x}) + \gamma_2 u(\bar{x}-h) + \gamma_3 u(\bar{x}-2h)$$

"one-sided" finite difference

Tool: Taylor expansion

$$\begin{aligned}
&\gamma_1 u(\bar{x}) + \gamma_2 u(\bar{x}-h) + \gamma_3 u(\bar{x}-2h) \\
&\quad = \gamma_1 u(\bar{x}) + \gamma_2 \left(\left(u(\bar{x}) - hu'(\bar{x}) + \frac{h^2}{2} u''(\bar{x}) - \frac{h^3}{6} u'''(\bar{x}) \right) \right. \\
&\qquad + \gamma_3 \left(\left(u(\bar{x}) - 2hu'(\bar{x}) + \frac{4h^2}{2} u''(\bar{x}) - \frac{8h^3}{6} u'''(\bar{x}) \right) \right) + O(\max|\gamma_k| h^4)
\end{aligned}$$

$$u(\bar{x}-2h) \quad u(\bar{x}-h) \quad u(\bar{x})$$

$$\begin{cases} \gamma_1 + \gamma_2 + \gamma_3 = 0 \\ -h\gamma_2 - 2h\gamma_3 = 1 \\ h^2\gamma_2 + 4h^2\gamma_3 = 0 \end{cases} \Rightarrow \gamma_1 = \frac{3}{2h}, \quad \gamma_2 = -\frac{2}{h}, \quad \gamma_3 = \frac{1}{2h}$$

$$u'(\bar{x}) = \frac{3}{2h}u(\bar{x}) - \frac{2}{h}u(\bar{x}-h) + \frac{1}{2h}u(\bar{x}-2h) + O(h^2).$$

Another one-sided finite difference formula?

$$u(\bar{x}) \quad u(\bar{x}+h) \quad u(\bar{x}+2h)$$

2.3.1 FD Formulas for Second-order Derivatives

We can apply finite difference operators twice to get finite difference formulas to approximate the second-order derivative $u''(\bar{x})$, *e.g.*, the central finite difference formula

$$\begin{aligned}\Delta_+\Delta_- u(\bar{x}) &= \Delta_+ \frac{u(\bar{x}) - u(\bar{x}-h)}{h} \\ &= \frac{1}{h}\left(\frac{u(\bar{x}+h) - u(\bar{x})}{h} - \frac{u(\bar{x}) - u(\bar{x}-h)}{h}\right) \\ &= \frac{u(\bar{x}-h) - 2u(\bar{x}) + u(\bar{x}+h)}{h^2} \\ &= \Delta_-\Delta_+ u(\bar{x}) = \delta^2 u(\bar{x}) \end{aligned} \tag{2.18}$$

approximates $u''(\bar{x})$ to $O(h^2)$.

Finite difference operators can be used to derive approximations for partial derivatives

$$\delta_x \delta_y u(\bar{x}, \bar{y})$$
$$= \frac{u(\bar{x}+h, \bar{y}+h) + u(\bar{x}-h, \bar{y}-h) - u(\bar{x}+h, \bar{y}-h) - u(\bar{x}-h, \bar{y}+h)}{4h^2}$$
$$\approx \frac{\partial^2 u}{\partial x \partial y}(\bar{x}, \bar{y}), \qquad (2.20)$$

Here we use the x subscript on δ_x to denote the central finite difference operator in the x direction, and so on.

Consistency, Stability, Convergence

Global Error $\mathbf{E} = \mathbf{U} - \mathbf{u}$

$$\begin{cases} \mathbf{U} = [U_1, U_2, \ldots, U_n]^T & : \text{ the approximate solution} \\ \mathbf{u} = [u(x_1), u(x_2), \ldots, u(x_n)] & : \text{ the exact solution} \end{cases}$$

A smallest upper bound for the error vector:

- The maximum or infinity norm $\|\mathbf{E}\|_\infty = \max_i\{|e_i|\}$
- The 1-norm $\|\mathbf{E}\|_1 = \sum_i h_i |e_i|$ analogous to $\int |e(x)|\, dx$
- The 2-norm $\|\mathbf{E}\|_2 = (\sum_i h_i |e_i|^2)^{1/2}$ analogous to $(\int |e(x)|^2\, dx)^{1/2}$

Definition 2.1. A finite difference method is called convergent if $\lim_{h \to 0} \|\mathbf{E}\| = 0$

If $\|E\| \le Ch^p,\ p > 0$, the finite difference method has p-th order accurate.

Local truncation errors refer to the differences between the original differential equation and its finite difference approximations at grid points.

The original differential equation: → $u''(x)=f(x), \quad 0<x<1, \quad u(0)=u_a, \quad u(1)=u_b,$

Finite difference approximation: → $\frac{U_{i-1}-2U_i+U_{i+1}}{h^2}=f(x_i)$

Local truncation error of the finite difference scheme at x_i is

$$T_i=\frac{u(x_i-h)-2u(x_i)+u(x_i+h)}{h^2}-f(x_i), \qquad i=1,2,\ldots,n-1.$$

Two steps to obtain local truncation error:

1. Move the right-hand side to the left-hand side
2. Substituting the true solution $u(x_i)$ for U_i.

Definition 2.2. A finite difference scheme is called *consistent* if $\lim_{h\to 0} T(x) = 0$.

How to check whether or not a finite difference scheme is consistent?

------ Perform Taylor expansion for all the terms in the local truncation error at a master grid point x_i.

$$T(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} - u''(x) = \frac{h^2}{12}u^{(4)}(x) + \cdots = O(h^2)$$

To obtain $|T(x)| \le Ch^2$, we let $C = \max_{0\le x\le 1} |\frac{1}{12}u^{(4)}(x)|$

➡ The difference scheme is consistent and the discretization is second-order accurate.

However, consistency can not guarantee the convergence of a scheme, and we need to satisfy another condition, namely, its *stability*.

Definition of Stability

$$A\mathbf{u} = \mathbf{F} + \mathbf{T}, \qquad A\mathbf{U} = \mathbf{F} \qquad \Longrightarrow \qquad A(\mathbf{u} - \mathbf{U}) = \mathbf{T} = -A\mathbf{E}, \qquad (2.24)$$

Error of the solution

Local truncation error

$\mathbf{E} = \mathbf{U} - \mathbf{u}$, $\mathbf{F}$ takes the boundary condition into account.

If A is nonsingular, then $\|\mathbf{E}\| = \|A^{-1}\mathbf{T}\| \le \|A^{-1}\|\|\mathbf{T}\|$.

Definition 2.3. A finite difference method for the BVPs is stable if A is invertible and

$$\|A^{-1}\| \le C, \quad \text{for all} \quad 0 < h < h_0, \qquad (2.25)$$

where C and h_0 are two constants that are independent of h.

Theorem 2.4. A consistent and stable finite difference method is convergent.

"convergent = consistent + stable"

$$\|\mathbf{E}\| = \|A^{-1}\mathbf{T}\| \le \|A^{-1}\|\,\|\mathbf{T}\|$$

Recall that we already have $|T(x)| \le \bar{C}h^2$, which means $\lim_{h\to 0} \|\mathbf{T}\| = 0$

We want $\lim_{h\to 0} \|\mathbf{E}\| = 0$, then we need

$$\|A^{-1}\| \le C, \quad \text{for all} \quad 0 < h < h_0$$

⇕

Definition of a "stable" scheme

Local truncation error:	T_i = LHS-RHS with U_i substituted by $u(x_i)$ at x_i
Order of the discretization of the scheme:	$T_i = O(h^p)$ for all x_i, or $T(x) = O(h^p)$
Order of the finite difference method:	$\|\|\mathbf{E}\|\| < Ch^p$
Consistence: when $h \to 0$, error of the scheme → 0	$\lim_{h\to 0} T_i = 0$ or $\lim_{h\to 0} T(x) = 0$ or $\lim_{h\to 0} \|\|\mathbf{T}\|\| = 0$
Stability:	$\|\|\mathbf{A}^{-1}\|\| < C$
Convergence: when $h \to 0$, error of the solution → 0	$\lim_{h\to 0} \|\|\mathbf{E}\|\| = 0$

Usually it is easy to prove consistency but more difficult to prove stability.

To prove the convergence of the central finite difference scheme

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} = f(x_i) \qquad \text{for} \qquad u''(x) = f(x)$$

We examine the condition $\|A^{-1}\| \le C, \quad \text{for all} \quad 0 < h < h_0$

where

$$A = \begin{bmatrix} -\frac{2}{h^2} & \frac{1}{h^2} & & & & \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & & \\ & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\ & & & & \frac{1}{h^2} & -\frac{2}{h^2} \end{bmatrix}$$

Lemma 2.5. *Consider a symmetric tridiagonal matrix $A \in R^{n\times n}$ whose main diagonals and off-diagonals are two constants, d and α, respectively. Then the eigenvalues of A are*

$$\lambda_j = d + 2\alpha \cos\left(\frac{\pi j}{n+1}\right), \quad j = 1, 2, \ldots, n, \tag{2.26}$$

and the corresponding eigenvectors are

$$x_k^j = \sin\left(\frac{\pi k j}{n+1}\right), \quad k = 1, 2, \ldots, n. \tag{2.27}$$

Theorem 2.6. The central finite difference method for $u''(x) = f(x)$ and a Dirichlet boundary condition is convergent.

Proof From the finite difference method, we know that the finite difference coefficient matrix $A \in R^{(n-1)\times(n-1)}$ and it is tridiagonal with $d = -2/h^2$ and $\alpha = 1/h^2$, so the eigenvalues of A are

$$\lambda_j = -\frac{2}{h^2} + \frac{2}{h^2}\cos\left(\frac{\pi j}{n}\right) = \frac{2}{h^2}\left(\cos(\pi j h) - 1\right).$$

Noting that the eigenvalues of A^{-1} are $1/\lambda_j$ and A^{-1} is also symmetric, we have[2]

$$\begin{aligned}\|A^{-1}\|_2 &= \frac{1}{\min|\lambda_j|} \\ &= \frac{h^2}{2(1-\cos(\pi h))} = \frac{h^2}{4\sin^2\frac{\pi h}{2}} \approx \frac{1}{\pi^2}.\end{aligned}$$

C

We can proof this theorem by the following three steps:

1. If λ_j is the j-th eigenvalue of A, show that the eigenvalue of A^{-1} is $\frac{1}{\lambda_j}$.

   $Ax = \lambda_j x \;\Rightarrow\; A^{-1}Ax = \lambda_j\, A^{-1}x \Rightarrow \frac{x}{\lambda_j} = A^{-1}x \Rightarrow$ the eigenvalue of A^{-1} is $\frac{1}{\lambda_j}$.

2. If A is symmetric, show that A^{-1} is also symmetric.
   Since $(A^{-1})^T = (A^T)^{-1}$and $A^T = A$, so $(A^{-1})^T = A^{-1}$.

3. Show that $||A^{-1}||_2 = \frac{1}{\min|\lambda_j|}$.
   Since $||A||_2 = \sqrt{\lambda_{max}(A^T A)}$ and $A^T = A$, we have $||A||_2 = \max|\lambda_j|$,
   so $||A^{-1}||_2 = \max\frac{1}{|\lambda_j|} = \frac{1}{\min|\lambda_j|}$.

1D Sturm–Liouville problem

$$(p(x)u'(x))' - q(x)u(x) = f(x), \quad a < x < b, \tag{2.32}$$

$$u(a) = u_a, \quad u(b) = u_b, \quad \text{or other BC.} \tag{2.33}$$

Theorem 2.8. *If* $p(x) \in C^1(a,b)$, $q(x) \in C^0(a,b)$, $f(x) \in C^0(a,b)$, $q(x) \geq 0$ *and there is a positive constant such that* $p(x) \geq p_0 > 0$, *then there is unique solution* $u(x) \in C^2(a,b)$.

Steps to develop finite difference method

Step 1: Generate a grid.

$$x_i = a + ih, \quad h = \frac{b-a}{n}, \quad i = 0, 1, \ldots, n$$

Step 2: Substitute derivatives with finite difference formulas at each grid point.

Define $x_{i+\frac{1}{2}} = x_i + h/2$, so $x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = h$.

① Using the central finite difference formula at a typical grid point x_i with half grid size

$$\frac{p_{i+\frac{1}{2}} u'(x_{i+\frac{1}{2}}) - p_{i-\frac{1}{2}} u'(x_{i-\frac{1}{2}})}{h} - q_i u(x_i) = f(x_i) + \boxed{E_i^1} \rightarrow Ch^2$$

② Applying the central finite difference scheme for the first-order derivative at $x_{i+1/2}$ and $x_{i-1/2}$

$$\frac{p_{i+\frac{1}{2}} \frac{u(x_{i+1}) - u(x_i)}{h} - p_{i-\frac{1}{2}} \frac{u(x_i) - u(x_{i-1})}{h}}{h} - q_i u(x_i) = f(x_i) + E_i^1 + \boxed{E_i^2} \rightarrow Ch^2$$

The consequent finite difference solution $U_i \approx u(x_i)$ is then defined as the solution of the linear system of equations

$$\frac{p_{i+\frac{1}{2}}\, U_{i+1} - \left(p_{i+\frac{1}{2}} + p_{i-\frac{1}{2}}\right) U_i + p_{i-\frac{1}{2}}\, U_{i-1}}{h^2} - q_i U_i = f_i$$

for $i = 1, 2, \ldots, n-1$.

$$A\mathbf{U} = \mathbf{F}$$

$$\mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_{n-2} \\ U_{n-1} \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} f(x_1) - \frac{p_{1/2}u_a}{h^2} \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{p_{n-1/2}u_b}{h^2} \end{bmatrix}$$

- Symmetric
- Negative definite
- Weakly diagonally dominant

$$A = \begin{bmatrix} -\frac{p_{1/2}+p_{3/2}}{h^2} - q_1 & \frac{p_{3/2}}{h^2} & & & \\ \frac{p_{3/2}}{h^2} & -\frac{p_{3/2}+p_{5/2}}{h^2} - q_2 & \frac{p_{5/2}}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & & \frac{p_{n-3/2}}{h^2} & -\frac{p_{n-3/2}+p_{n-1/2}}{h^2} - q_{n-1} \end{bmatrix}$$

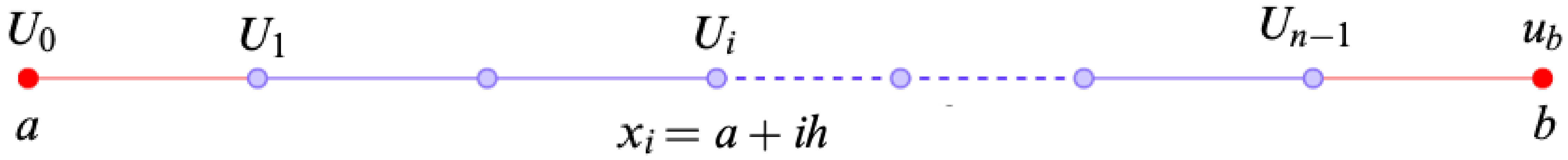
2.7 The Ghost Point Method for Boundary Conditions Involving Derivatives

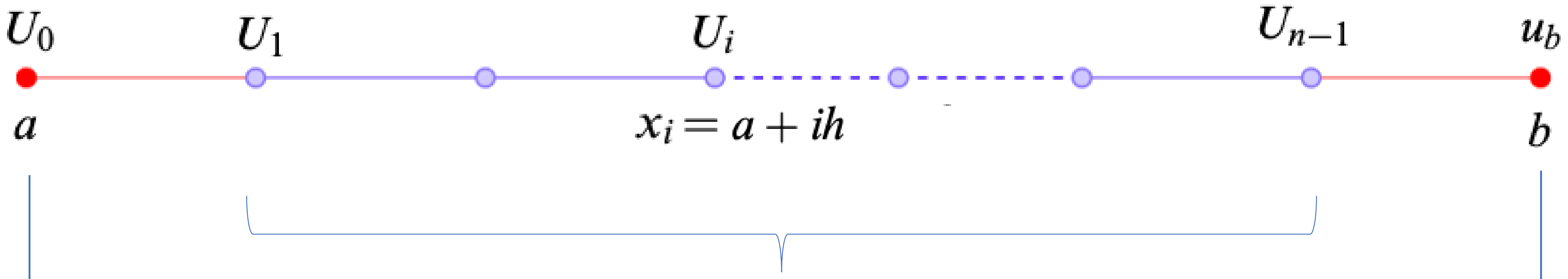
Neumann and mixed (Robin) boundary conditions

$$u''(x) = f(x), \quad a < x < b,$$

$$u'(a) = \alpha, \qquad u(b) = u_b\,,$$

where the solution at $x = a$ is unknown.





Interior grid points

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} = f(x_i)$$

An additional equation at $x = a$

$$\frac{U_1 - U_0}{h} = \alpha \quad \text{or} \quad \frac{-U_0 + U_1}{h^2} = \frac{\alpha}{h}$$

Known value,
no need of equation

Slightly different here! Not $-\frac{2}{h^2}$

Unknown

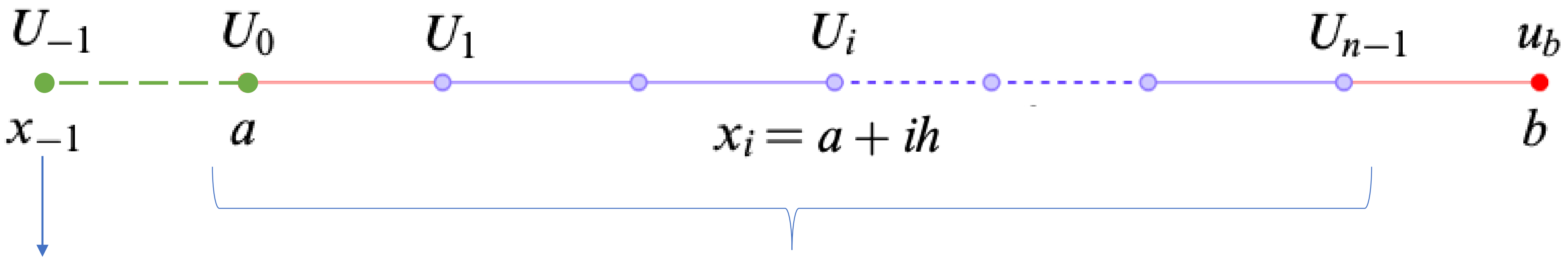
$$\begin{bmatrix} -\frac{1}{h^2} & \frac{1}{h^2} & & & & \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & & \\ & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\ & & & & \frac{1}{h^2} & -\frac{2}{h^2} \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{n-2} \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{\alpha}{h} \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{u_b}{h^2} \end{bmatrix}. \qquad (2.40)$$

Note: only First-order due to the boundary condition at $x = a$

$$E_f(h) = \frac{u(\bar{x}+h) - u(\bar{x})}{h} - u'(\bar{x}) = \frac{1}{2}u''(\xi)h = O(h)$$

Contribution from Dirichlet boundary condition

The Ghost Point Method



A ghost grid point

$$x_{-1} = x_0 - h = a - h$$

Interior grid points

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} = f(x_i)$$

$$\frac{U_1 - U_{-1}}{2h} = \alpha,$$

Insert into

$$\frac{U_{-1} - 2U_0 + U_1}{h^2} = f_0,$$

$$\frac{-U_0 + U_1}{h^2} = \frac{f_0}{2} + \frac{\alpha}{h}$$

First-order Method

$$
\begin{bmatrix}
-\frac{1}{h^2} & \frac{1}{h^2} & & & & \\
\frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & & \\
& \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\
& & \ddots & \ddots & \ddots & \\
& & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
& & & & \frac{1}{h^2} & -\frac{2}{h^2}
\end{bmatrix}
\begin{bmatrix}
U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{n-2} \\ U_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
\frac{\alpha}{h} \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{u_b}{h^2}
\end{bmatrix}
$$

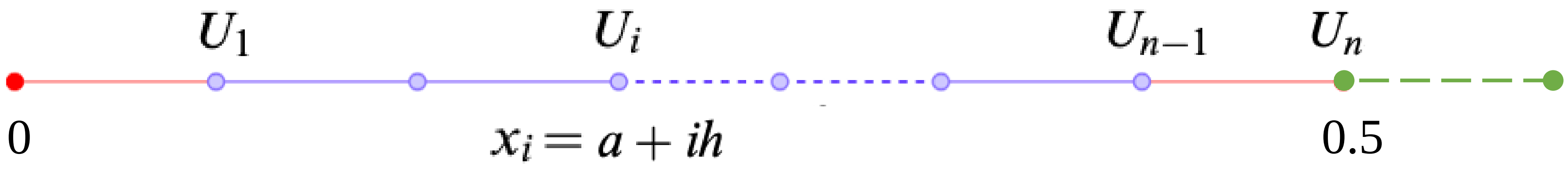
Send-order Method (Ghost Point Method)

$$
\begin{bmatrix}
-\frac{1}{h^2} & \frac{1}{h^2} & & & & \\
\frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & & \\
 & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\
 & & & & \frac{1}{h^2} & -\frac{2}{h^2}
\end{bmatrix}
\begin{bmatrix}
U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{n-2} \\ U_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
\frac{f_0}{2} + \frac{\alpha}{h} \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{u_b}{h^2}
\end{bmatrix}
$$

Comparison of the two finite difference methods

$$\begin{cases} u''(x) = f(x) \\ f(x) = -\pi^2 \cos \pi x, \\ u(0) = 1,\ \boxed{u'(0.5) = -\pi}, \end{cases}$$

The exact solution is $u(x) = \cos \pi x$.



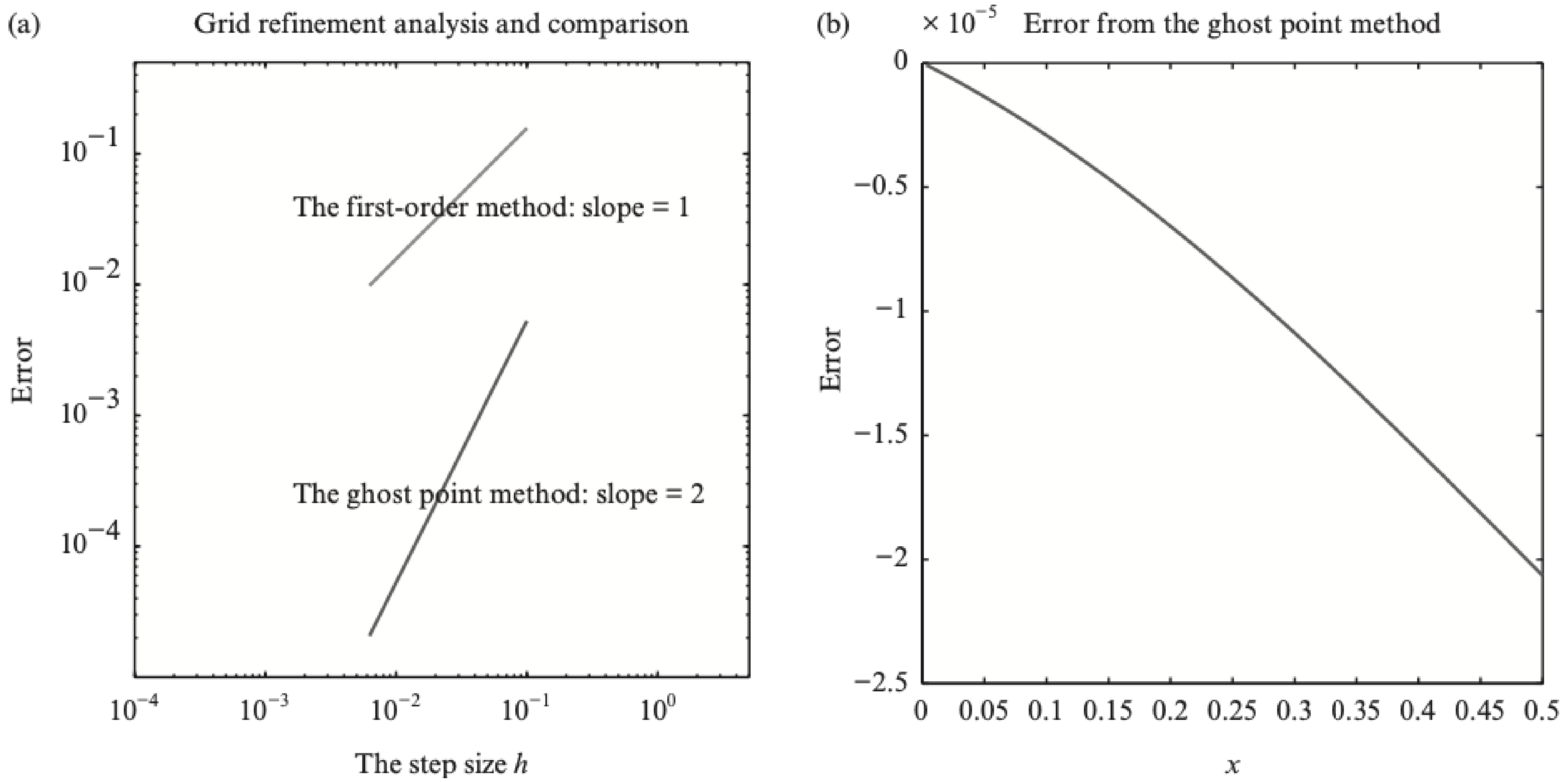


Figure 2.4. (a) A grid refinement analysis of the ghost point method and the first-order method. The slopes of the curves are the order of convergence. (b) The error plot of the computed solution from the ghost point method.

Dirichlet on both ends

```
function [x,U] = two_point(a,b,ua,ub,f,n)

h = (b-a)/n; h1=h*h;
A = sparse(n-1,n-1);
F = zeros(n-1,1);

for i=1:n-2,
    A(i,i) = -2/h1;
    A(i+1,i) = 1/h1;
    A(i,i+1)= 1/h1;
end
A(n-1,n-1) = -2/h1;
```

Form the matrix

```
for i=1:n-1,
    x(i) = a+i*h;
    F(i) = feval(f,x(i));
end

F(1) = F(1) - ua/h1;
F(n-1) = F(n-1) - ub/h1;

U = A\F;

return
```

Form the RHS

Dirichlet and Neumann on different sides

```
function [x,U] = ghost_at_b(a,b,ua,uxb,f,n)

h = (b-a)/n; h1=h*h;
A = sparse(n,n);
F = zeros(n,1);

for i=1:n-1,
      A(i,i) = -2/h1;
      A(i+1,i) = 1/h1;
      A(i,i+1)= 1/h1;
end
A(n,n) = -2/h1;
A(n,n-1) = 2/h1;
```

Form the matrix

```
for i=1:n,
      x(i) = a+i*h;
      F(i) = feval(f,x(i));
end

F(1) = F(1) - ua/h1;
F(n) = F(n) - 2*uxb/h;

U = A\F;

return
```

Form the RHS

2.8 An example of a Nonlinear BVP

$$\frac{d^2u}{dx^2} - u^2 = f(x), \qquad 0 < x < \pi, \qquad (2.46)$$
$$u(0) = 0, \qquad u(\pi) = 0.$$

Using the central finite difference scheme

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} - U_i^2 = f(x_i), \quad i = 1, 2, \ldots, n-1. \qquad (2.47)$$

Discretizing a nonlinear differential equation generally produces a nonlinear algebraic system.

Simple substitution method

Given an initial guess $U^{(0)}(x)$, we proceed

Unknown

Known value: treated as a coefficient

$$\frac{U_{i-1}^{k+1} - 2U_i^{k+1} + U_{i+1}^{k+1}}{h^2} - U_i^k U_i^{k+1} = f(x_i), \qquad k = 0, 1, \ldots, \qquad (2.48)$$

k=k+1

Involving a two-point BVP at each iteration.

Newton’s method

A nonlinear system of equations $\mathbf{F}(\mathbf{U}) = 0$ is obtained if we discretize （2.46）

$$\begin{cases} F_1(U_1, U_2, \ldots, U_m) = 0, \\ F_2(U_1, U_2, \ldots, U_m) = 0, \\ \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ F_m(U_1, U_2, \ldots, U_m) = 0, \end{cases} \tag{2.49}$$

where

$$F_i(U_1, U_2, \ldots, U_m) = \frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} - U_i^2 - f(x_i), \quad i = 1, 2, \ldots, n-1.$$

Given an initial guess $\mathbf{U}^{(0)}$, the Newton iteration is

$$\begin{cases} J(\mathbf{U}^{(k)})\Delta\mathbf{U}^{(k)} = -\mathbf{F}(\mathbf{U}^{(k)}), \\ \mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \Delta\mathbf{U}^{(k)}, \end{cases} \quad k = 0, 1, \ldots$$

where $J(\mathbf{U})$ is the Jacobian matrix defined as

$$\begin{bmatrix} \frac{\partial F_1}{\partial U_1} & \frac{\partial F_1}{\partial U_2} & \cdots & \frac{\partial F_1}{\partial U_m} \\ \frac{\partial F_2}{\partial U_1} & \frac{\partial F_2}{\partial U_2} & \cdots & \frac{\partial F_2}{\partial U_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_m}{\partial U_1} & \frac{\partial F_m}{\partial U_2} & \cdots & \frac{\partial F_m}{\partial U_m} \end{bmatrix} = \frac{1}{h^2} \begin{bmatrix} -2-2h^2U_1 & 1 & & \\ 1 & -2-2h^2U_2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & -2-2h^2U_{n-1} \end{bmatrix}.$$

Initial guess: $U_i^0 = x_i(\pi - x_i)$ Mesh size: $n = 40$ Tolerance: $tol = 10^{-8}$

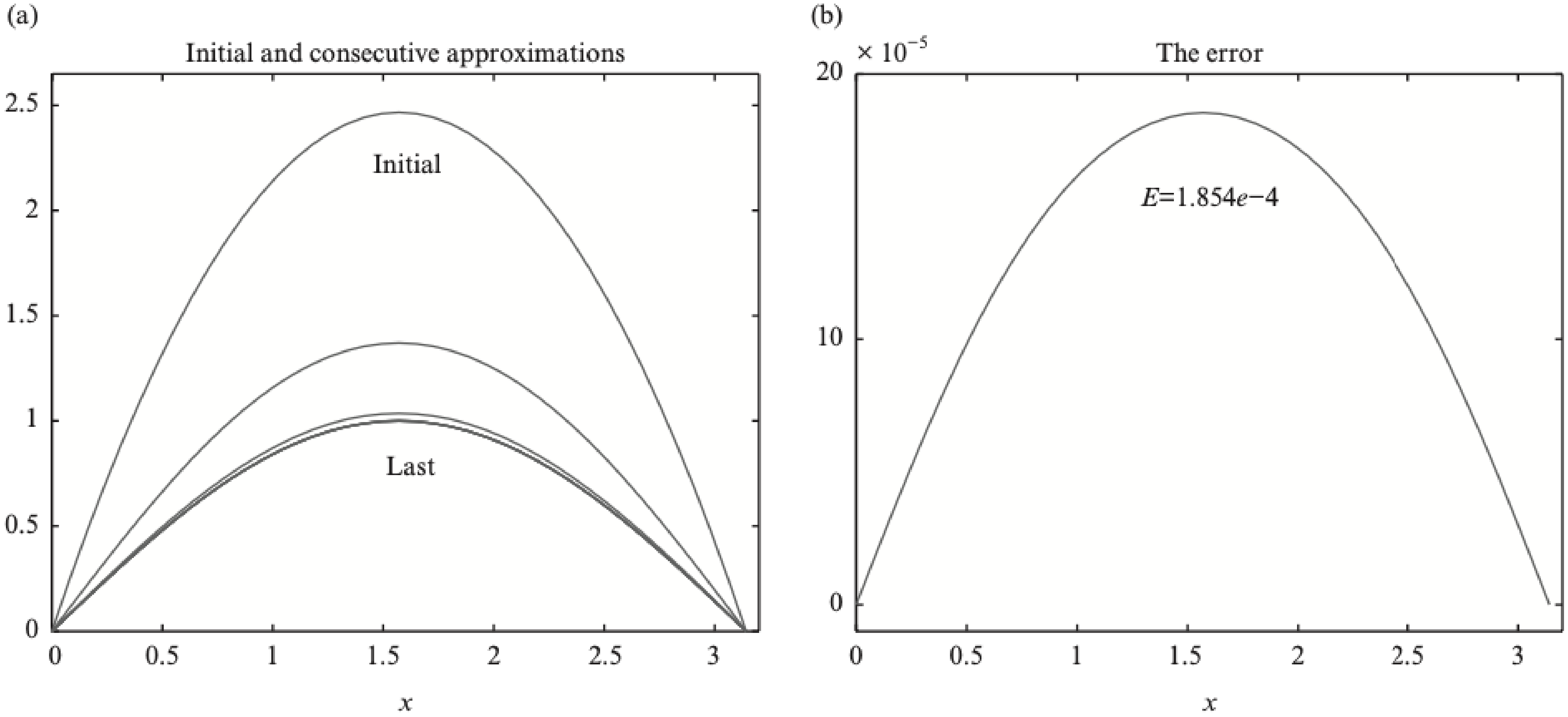


Figure 2.5. (a) Plot of the initial and consecutive approximations to the nonlinear system of equations and (b) the error plot.

Remarks on Newton method:

- It is not always easy to find $J(U)$ and it can be computationally expensive.
- Newton's method is quadratically convergent locally.
- Well-known software packages are available (MINPACK, PETSc (for parallel computing))

2.9 The Grid Refinement Analysis Technique

To validate and confirm the analysis (consistency, stability, order of convergence, etc) numerically:

- Analyse the output to see whether they agree with the ODE or PDE
- Compare the numerical solutions with experiential data
- Perform a grid refinement analysis

When there is an exact solution:

Assume a method is p-th order accurate, such that $\|E_h\| \sim Ch^p$,
if we divide h by half to get $\|E_{h/2}\|$, then

$$\text{ratio} = \frac{\|E_h\|}{\|E_{h/2}\|} \approx \frac{Ch^p}{C(h/2)^p} = 2^p, \tag{2.52}$$

$$p \approx \frac{\log\left(\|E_h\|/\|E_{h/2}\|\right)}{\log 2} = \frac{\log(\text{ratio})}{\log 2}. \tag{2.53}$$

- For a first-order method ($p = 1$), the ratio → 2
- For a second-order method ($p = 2$), the ratio → 4
- If $1 < p < 2$, the method is called superlinear convergent

When there is no exact solution

To compare a numerical solution with one obtained from a finer mesh.

Suppose the numerical solution converges and satisfies

$$u_h = u_e + Ch^p + \cdots \tag{2.54}$$

where u_h is the numerical solution and u_e is the true solution, and let u_{h_*} be the solution obtained from the finest mesh

$$u_{h_*} = u_e + C{h_*}^p + \cdots . \tag{2.55}$$

Thus we have

$$u_h - u_{h_*} \approx C\left(h^p - {h_*}^p\right), \tag{2.56}$$

$$u_{h/2} - u_{h_*} \approx C\left((h/2)^p - {h_*}^p\right). \tag{2.57}$$

From the estimates above, we obtain the ratio

$$\frac{u_h - u_{h_*}}{u_{h/2} - u_{h_*}} \approx \frac{h^p - {h_*}^p}{(h/2)^p - {h_*}^p} = \frac{2^p\,(1 - (h_*/h)^p)}{1 - (2h_*/h)^p}\,, \tag{2.58}$$

from which we can estimate the order of accuracy p. For example, on doubling the number of grid points successively we have

$$\frac{h_*}{h} = 2^{-k}, \qquad k = 2, 3, \ldots, \tag{2.59}$$

then the ratio in (2.58) is

$$\frac{\tilde{u}(h) - \tilde{u}(h^*)}{\tilde{u}(\frac{h}{2}) - \tilde{u}(h^*)} = \frac{2^p\left(1 - 2^{-kp}\right)}{1 - 2^{p(1-k)}}. \tag{2.60}$$

In particular, for a first-order method ($p = 1$) this becomes

$$\frac{\tilde{u}(h) - \tilde{u}(h^*)}{\tilde{u}(\frac{h}{2}) - \tilde{u}(h^*)} = \frac{2\left(1 - 2^{-k}\right)}{1 - 2^{1-k}} = \frac{2^k - 1}{2^{k-1} - 1}.$$

Not 2, but goes to 2

If we take $k = 2, 3, \ldots$, then the ratios above are

$$3, \qquad \frac{7}{3} \simeq 2.333, \qquad \frac{15}{7} \simeq 2.1429, \qquad \frac{31}{15} \simeq 2.067, \qquad \cdots.$$

Similarly, for a second-order method ($p = 2$), (2.60) becomes

$$\frac{\tilde{u}(h) - \tilde{u}(h^*)}{\tilde{u}(\frac{h}{2}) - \tilde{u}(h^*)} = \frac{4\left(1 - 4^{-k}\right)}{1 - 4^{1-k}} = \frac{4^k - 1}{4^{k-1} - 1},$$

Not 4, but goes to 4

and the ratios are

$$5, \qquad \frac{63}{15} = 4.2, \qquad \frac{255}{63} \simeq 4.0476, \qquad \frac{1023}{255} \simeq 4.0118, \qquad \cdots$$

when $k = 2, 3, \ldots$