Math 6630: Numerical Solutions of Partial Differential Equations
Finite difference methods for 1D stationary problems

See LeVeque 2007, Chapter 2

Akil Narayan[1]

[1]Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

January 18, 2023

THE UNIVERSITY OF UTAH

www.sci.utah.edu

Finite difference methods

Finite difference methods are a good starting point to understand numerical methods: they are simple, easy to understand, and (typically) easy to implement.

The basic idea is to approximate derivatives in DE's with *finite* difference approximations:

$$u'(x) \approx \frac{u(x+h) - u(x)}{h} \quad \text{or} \quad \frac{u(x) - u(x-h)}{h} \quad \text{or} \quad \frac{u(x+h) - u(x-h)}{2h}$$

The above are only examples, but are conceptually useful to understand the overall picture.

Finite difference methods

Finite difference methods are a good starting point to understand numerical methods: they are simple, easy to understand, and (typically) easy to implement.

The basic idea is to approximate derivatives in DE's with *finite* difference approximations:

$$u'(x) \approx \frac{u(x+h)-u(x)}{h} \quad \text{or} \quad \frac{u(x)-u(x-h)}{h} \quad \text{or} \quad \frac{u(x+h)-u(x-h)}{2h}$$

The above are only examples, but are conceptually useful to understand the overall picture.

We will need some notation to understand how finite difference methods work: With $u(x)$ an unknown function for $x \in [0,1]$, we will *discretize* u by considering its value at $M+2$ equispaced points on $[0,1]$:

$$h := \frac{1}{M+1}, \qquad\qquad x_j := jh, \; j = 0, \ldots, M+1.$$

We will use u_j to denote our computational approximation to $u(x_j)$, i.e.,

$$u_j \approx u(x_j)$$

More notation: difference operators

It will be convenient to use some shorthand for finite difference operators.

With $u_j \approx u(x_j)$ and x_j an equidistant grid of spacing h, we define:

$$
\begin{aligned}
D_+ u(x) &:= \frac{u(x+h) - u(x)}{h}, & D_- u(x) &:= \frac{u(x) - u(x-h)}{h}, & D_0 u(x) &:= \frac{u(x+h) - u(x-h)}{2h}, \\
D_+ u_j &:= \frac{u_{j+1} - u_j}{h}, & D_- u_j &:= \frac{u_j - u_{j-1}}{h}, & D_0 u_j &:= \frac{u_{j+1} - u_{j-1}}{2h}.
\end{aligned}
$$

Note that $D_{\pm,0}$ apply in conceptually similar ways to functions $u(x)$ as to discrete values u_j.

More notation: difference operators

It will be convenient to use some shorthand for finite difference operators.

With $u_j \approx u(x_j)$ and x_j an equidistant grid of spacing h, we define:

$$D_+ u(x) := \frac{u(x+h) - u(x)}{h}, \quad D_- u(x) := \frac{u(x) - u(x-h)}{h}, \quad D_0 u(x) := \frac{u(x+h) - u(x-h)}{2h},$$

$$D_+ u_j := \frac{u_{j+1} - u_j}{h}, \quad D_- u_j := \frac{u_j - u_{j-1}}{h}, \quad D_0 u_j := \frac{u_{j+1} - u_{j-1}}{2h}.$$

Note that $D_{\pm,0}$ apply in conceptually similar ways to functions $u(x)$ as to discrete values u_j.

These difference operators are convenient for shorthand. For example:

$$D_+ u(x) \approx u'(x) + \mathcal{O}(h), \quad D_- u(x) \approx u'(x) + \mathcal{O}(h), \quad D_0 u(x) \approx u'(x) + \mathcal{O}(h^2).$$

We can chain these operators to approximate higher order derivatives:

$$D_+ D_- u(x) = D_- D_+ u(x) \approx u''(x) + \mathcal{O}(h^2)$$

1D steady-state diffusion

Our prototypical equation is an ODE describing the steady-state temperature distribution on a one-dimensional domain:

$$\begin{aligned} -u''(x) &= f(x), & x \in (0,1) \\ u(0) &= g_0, \\ u(1) &= g_1. \end{aligned}$$

where f, g_0, and g_1 are presumed given and known.

This is a model for steady-state heat diffusion:

- The ODE models homogeneous, isotropic heat diffusion in an environment.
- $u(x)$ is the temperature at location x.
- The boundary conditions correspond to pinning the temperature value.

1D steady-state diffusion

Our prototypical equation is an ODE describing the steady-state temperature distribution on a one-dimensional domain:

$$\begin{aligned} -u''(x) &= f(x), & x \in (0,1) \\ u(0) &= g_0, \\ u(1) &= g_1. \end{aligned}$$

where f, g_0, and g_1 are presumed given and known.

We construct a finite-difference (FD) scheme for this equation as follows:

$$u''(x) \longrightarrow D_+ D_- u_j = \frac{1}{h^2}\left(u_{j-1} - 2u_j + u_{j+1}\right)$$

Thus, we have:

$$\begin{aligned} -u_{j-1} + 2u_j - u_{j+1} &= h^2 f_j, & j = 1, \ldots, M, \\ u_0 &= g_0, \\ u_{M+1} &= g_1, \end{aligned}$$

where we define $f_j = f(x_j)$.

The scheme

$$
\begin{aligned}
-u_{j-1} + 2u_j - u_{j+1} &= h^2 f_j, & j &= 1, \ldots, M, \\
u_0 &= g_0, \\
u_{M+1} &= g_1,
\end{aligned}
$$

If we define vectors,

$$
\boldsymbol{u} = (u_1, \ldots, u_M)^T, \qquad \boldsymbol{f} = (f_1, \ldots, f_M)^T,
$$

where the vector $\boldsymbol{u}$ contains our unknowns, we have the linear system,

$$
\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,
$$

and the matrix $\boldsymbol{A}$ is symmetric:

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

<u>Goal</u>: compute the vector $\boldsymbol{u}$.

Numerical considerations

To summarize: we have discretized the ODE

$$
\begin{aligned}
-u''(x) &= f(x), & x \in (0,1) \\
u(0) &= g_0, \\
u(1) &= g_1
\end{aligned}
$$

to obtain the linear system

$$
\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,
$$

Some observations worth noting:

- $u \mapsto u''$ is a *symmetric* operator, and $\boldsymbol{A}$ is a symmetric matrix.
- $\boldsymbol{A}$ is invertible (actually, its spectrum is explicitly computable)
- $\boldsymbol{A}$ is *sparse*, having only $3M - 2$ nonzero entries.
- The naive computational cost of this approach is $\mathcal{O}(M^3)$, as that is the brute-force cost to invert an $M \times M$ matrix.
- For this particular problem, there are $\mathcal{O}(M)$ algorithms to solve the linear system.

Numerical considerations

To summarize: we have discretized the ODE

$$
\begin{aligned}
-u''(x) &= f(x), & x \in (0,1) \\
u(0) &= g_0, \\
u(1) &= g_1
\end{aligned}
$$

to obtain the linear system

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,$$

Some observations worth noting:

- $u \mapsto u''$ is a *symmetric* operator, and $\boldsymbol{A}$ is a symmetric matrix.
- $\boldsymbol{A}$ is invertible (actually, its spectrum is explicitly computable)
- $\boldsymbol{A}$ is *sparse*, having only $3M - 2$ nonzero entries.
- The naive computational cost of this approach is $\mathcal{O}(M^3)$, as that is the brute-force cost to invert an $M \times M$ matrix.
- For this particular problem, there are $\mathcal{O}(M)$ algorithms to solve the linear system.

We can compute $\boldsymbol{u}$. But is it true that $u_j \approx u(x_j)$?
Does the numerical solution become more accurate as $h \downarrow 0$?

Consistency, I

Our high-level questions regard *convergence* of the scheme.

Before addressing these, consider a simpler question about "consistency".

The **Local Truncation Error** (LTE) τ for a scheme is the residual of the scheme when the *exact* solution $u(x_j)$ is inserted in place of u_j.

$$\tau_j := (D_+ D_- u(x_j) - f(x_j))$$

This is the error in the ODE statement at x_j due to our discretization.

Consistency, I

Our high-level questions regard *convergence* of the scheme.

Before addressing these, consider a simpler question about "consistency".

The **Local Truncation Error** (LTE) τ for a scheme is the residual of the scheme when the *exact* solution $u(x_j)$ is inserted in place of u_j.

$$\tau_j := (D_+ D_- u(x_j) - f(x_j))$$

This is the error in the ODE statement at x_j due to our discretization.

An exercise shows that,

$$\tau_j = c h^2 u^{(4)}(x_j) + \mathcal{O}(h^4), \qquad \left(c = 1/12\right)$$

where c is an absolute constant.

Consistency, I

Our high-level questions regard *convergence* of the scheme.

Before addressing these, consider a simpler question about "consistency".

The **Local Truncation Error** (LTE) τ for a scheme is the residual of the scheme when the *exact* solution $u(x_j)$ is inserted in place of u_j.

$$\tau_j := \left(D_+ D_- u(x_j) - f(x_j)\right)$$

This is the error in the ODE statement at x_j due to our discretization.

An exercise shows that,

$$\tau_j = c h^2 u^{(4)}(x_j) + \mathcal{O}(h^4),$$

where c is an absolute constant. Our estimates will use the norm of the LTE:

$$\boldsymbol{\tau} = (\tau_1, \ldots, \tau_M)^T, \|\tau\|_2^2 = h \sum_{j=1}^{M} |\tau_j|^2.$$

Note that M (the size of $\boldsymbol{\tau}$) scales like $1/h$, and that the 2-norm is scaled by h. This scaling factor is sensible:

$$\int_0^1 \tau^2(x) \mathrm{d}x \approx h \sum_{j=1}^{M} \tau^2(x_j)$$

Consistency, II

Clearly a "small" LTE is desirable – a particular notion of "small" is called consistency.

Definition

We say that a numerical scheme is *consistent* if

$$\lim_{h \downarrow 0} \|\boldsymbol{\tau}\|_2 = 0.$$

In our particular case, we have,

$$\|\boldsymbol{\tau}\|_2 = \mathcal{O}(h^2) \xrightarrow{h \downarrow 0} 0,$$

hence our discretization is consistent.

Because we know that the LTE is $\mathcal{O}(h^2)$, we might also say that the scheme is consistent to second order.

Note that consistency does *not* immediately translate into accuracy of the computed numerical solution, though it does suggest what we should expect.

Consistency, II

Clearly a "small" LTE is desirable – a particular notion of "small" is called consistency.

Definition

We say that a numerical scheme is *consistent* if

$$\lim_{h \downarrow 0} \|\boldsymbol{\tau}\|_2 = 0.$$

In our particular case, we have,

$$\|\boldsymbol{\tau}\|_2 = \mathcal{O}(h^2) \xrightarrow{h \downarrow 0} 0,$$

hence our discretization is consistent.

Because we know that the LTE is $\mathcal{O}(h^2)$, we might also say that the scheme is consistent to second order.

Note that consistency does *not* immediately translate into accuracy of the computed numerical solution, though it does suggest what we should expect.

Consistency, II

Clearly a "small" LTE is desirable – a particular notion of "small" is called consistency.

Definition

We say that a numerical scheme is *consistent* if

$$\lim_{h \downarrow 0} \|\boldsymbol{\tau}\|_2 = 0.$$

In our particular case, we have,

$$\|\boldsymbol{\tau}\|_2 = \mathcal{O}(h^2) \xrightarrow{h \downarrow 0} 0,$$

hence our discretization is consistent.

Because we know that the LTE is $\mathcal{O}(h^2)$, we might also say that the scheme is consistent to second order.

Note that consistency does *not* immediately translate into accuracy of the computed numerical solution, though it does suggest what we should expect.

Stability, I

In order to translate consistency into scheme accuracy, we will need the scheme to "behave well" for small h. This is stability.

Recall our scheme is

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,$$

and that everything on the right hand side is an input parameter (f, g_0, g_1).

Thus, abstractly we can view our scheme as the input-to-output map,

$$\boldsymbol{f}, g_0, g_1 \quad \xrightarrow{\boldsymbol{A}^{-1}} \quad \boldsymbol{u},$$

and hence we need $\boldsymbol{A}^{-1}$ to behave well.

Definition

We say that our scheme is *stable* if

$$\left\|\boldsymbol{A}^{-1}\right\|_2 \leqslant C \quad \text{for all } h \text{ sufficiently small},$$

where C is independent of h.

Note that the size of $\boldsymbol{A}$ depends on h, and in particular goes to infinity as h goes to 0.

Stability, I

In order to translate consistency into scheme accuracy, we will need the scheme to "behave well" for small h. This is stability.

Recall our scheme is

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,$$

and that everything on the right hand side is an input parameter (f, g_0, g_1).

Thus, abstractly we can view our scheme as the input-to-output map,

$$\boldsymbol{f}, g_0, g_1 \quad \xrightarrow{\boldsymbol{A}^{-1}} \quad \boldsymbol{u},$$

and hence we need $\boldsymbol{A}^{-1}$ to behave well.

Definition

We say that our scheme is *stable* if

$$\left\|\boldsymbol{A}^{-1}\right\|_2 \leqslant C \quad \text{for all } h \text{ sufficiently small},$$

where C is independent of h.

Note that the size of A depends on h, and in particular goes to infinity as h goes to 0.

Stability, I

In order to translate consistency into scheme accuracy, we will need the scheme to "behave well" for small h. This is stability.

Recall our scheme is

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M,$$

and that everything on the right hand side is an input parameter (f, g_0, g_1).

Thus, abstractly we can view our scheme as the input-to-output map,

$$\boldsymbol{f}, g_0, g_1 \quad \xrightarrow{\boldsymbol{A}^{-1}} \quad \boldsymbol{u},$$

and hence we need $\boldsymbol{A}^{-1}$ to behave well.

Definition

We say that our scheme is *stable* if

$$\left\|\boldsymbol{A}^{-1}\right\|_2 \leqslant C \quad \text{for all } h \text{ sufficiently small},$$

where C is independent of h.

Note that the size of $\boldsymbol{A}$ depends on h, and in particular goes to infinity as h goes to 0.

Stability, II

We can verify stability for our scheme. Recall that,

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

One can explicitly compute the spectrum of this matrix:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \qquad\qquad (\lambda_j, \boldsymbol{v}_j) = \left(\frac{4}{h^2}\sin^2\left(\pi h j/2\right), \sqrt{2}\sin(\boldsymbol{x} j \pi)\right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_M)^T$.

Of particular interest is that the eigenvectors of $\boldsymbol{A}$ "look" similar to those of the second derivative operator $u \mapsto u''$.

Stability, II

We can verify stability for our scheme. Recall that,

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

One can explicitly compute the spectrum of this matrix:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \qquad\qquad (\lambda_j, \boldsymbol{v}_j) = \left(\frac{4}{h^2}\sin^2\left(\pi h j/2\right), \sqrt{2}\sin(\boldsymbol{x} j \pi)\right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_M)^T$.

Of particular interest is that the eigenvectors of $\boldsymbol{A}$ "look" similar to those of the second derivative operator $u \mapsto u''$.

A linear algebraic interlude

We now review the following facts from linear algebra:

- If $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is any matrix, then it admits a singular value decomposition: there exist two unitary matrices $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$, such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The diagonal elements of $\boldsymbol{\Sigma}$ are $\{\sigma_j\}_j$, ordered such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$, are called the singular values of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is an $M \times M$ matrix, then $\|\boldsymbol{A}\|_2 = \sigma_1$, where σ_1 is the largest singular value of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is symmetric, then there exists an orthogonal matrix $\boldsymbol{U}$ and a diagonal matrix $\boldsymbol{D}$, both real-valued, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is both symmetric and invertible, then the diagonal elements of $\boldsymbol{D}$ are non-zero, and

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is symmetric, then

$$|\lambda_j| = \sigma_j,$$

  where $\{\sigma_j\}_j$ are the non-decreasing singular values of $\boldsymbol{A}$, and $\{\lambda_j\}_j$ are the eigenvalues of $\boldsymbol{A}$, ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_M|$.

A linear algebraic interlude

We now review the following facts from linear algebra:

- If $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is any matrix, then it admits a singular value decomposition: there exist two unitary matrices $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$, such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The diagonal elements of $\boldsymbol{\Sigma}$ are $\{\sigma_j\}_j$, ordered such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$, are called the singular values of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is an $M \times M$ matrix, then $\|\boldsymbol{A}\|_2 = \sigma_1$, where σ_1 is the largest singular value of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is symmetric, then there exists an orthogonal matrix $\boldsymbol{U}$ and a diagonal matrix $\boldsymbol{D}$, both real-valued, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is both symmetric and invertible, then the diagonal elements of $\boldsymbol{D}$ are non-zero, and

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is symmetric, then

$$|\lambda_j| = \sigma_j,$$

  where $\{\sigma_j\}_j$ are the non-decreasing singular values of $\boldsymbol{A}$, and $\{\lambda_j\}_j$ are the eigenvalues of $\boldsymbol{A}$, ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_M|$.

A linear algebraic interlude

We now review the following facts from linear algebra:

- If $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is any matrix, then it admits a singular value decomposition: there exist two unitary matrices $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$, such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The diagonal elements of $\boldsymbol{\Sigma}$ are $\{\sigma_j\}_j$, ordered such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$, are called the singular values of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is an $M \times M$ matrix, then $\|\boldsymbol{A}\|_2 = \sigma_1$, where σ_1 is the largest singular value of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is symmetric, then there exists an orthogonal matrix $\boldsymbol{U}$ and a diagonal matrix $\boldsymbol{D}$, both real-valued, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is both symmetric and invertible, then the diagonal elements of $\boldsymbol{D}$ are non-zero, and

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is symmetric, then

$$|\lambda_j| = \sigma_j,$$

where $\{\sigma_j\}_j$ are the non-decreasing singular values of $\boldsymbol{A}$, and $\{\lambda_j\}_j$ are the eigenvalues of $\boldsymbol{A}$, ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_M|$.

A linear algebraic interlude

We now review the following facts from linear algebra:

- If $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is any matrix, then it admits a singular value decomposition: there exist two unitary matrices $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$, such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The diagonal elements of $\boldsymbol{\Sigma}$ are $\{\sigma_j\}_j$, ordered such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$, are called the singular values of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is an $M \times M$ matrix, then $\|\boldsymbol{A}\|_2 = \sigma_1$, where σ_1 is the largest singular value of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is symmetric, then there exists an orthogonal matrix $\boldsymbol{U}$ and a diagonal matrix $\boldsymbol{D}$, both real-valued, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is both symmetric and invertible, then the diagonal elements of $\boldsymbol{D}$ are non-zero, and

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is symmetric, then

$$|\lambda_j| = \sigma_j,$$

where $\{\sigma_j\}_j$ are the non-decreasing singular values of $\boldsymbol{A}$, and $\{\lambda_j\}_j$ are the eigenvalues of $\boldsymbol{A}$, ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_M|$.

A linear algebraic interlude

We now review the following facts from linear algebra:

- If $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is any matrix, then it admits a singular value decomposition: there exist two unitary matrices $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$, such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The diagonal elements of $\boldsymbol{\Sigma}$ are $\{\sigma_j\}_j$, ordered such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots$, are called the singular values of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is an $M \times M$ matrix, then $\|\boldsymbol{A}\|_2 = \sigma_1$, where σ_1 is the largest singular value of $\boldsymbol{A}$.
- If $\boldsymbol{A}$ is symmetric, then there exists an orthogonal matrix $\boldsymbol{U}$ and a diagonal matrix $\boldsymbol{D}$, both real-valued, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is both symmetric and invertible, then the diagonal elements of $\boldsymbol{D}$ are non-zero, and

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T.$$

- If $\boldsymbol{A}$ is symmetric, then

$$|\lambda_j| = \sigma_j,$$

  where $\{\sigma_j\}_j$ are the non-decreasing singular values of $\boldsymbol{A}$, and $\{\lambda_j\}_j$ are the eigenvalues of $\boldsymbol{A}$, ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_M|$.

Stability, III

We can now finish our stability verification:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \qquad\qquad (\lambda_j, \boldsymbol{v}_j) = \left(\frac{4}{h^2} \sin^2\left(\pi j h/2\right), \sqrt{2}\sin(\boldsymbol{x} j \pi) \right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_M)^T$.

Since $\boldsymbol{A}$ is invertible and symmetric:

$$\|\boldsymbol{A}^{-1}\|_2 = \sigma_1(\boldsymbol{A}^{-1}) = \max_j |\lambda_j(\boldsymbol{A}^{-1})| = \max_j \frac{1}{|\lambda_j(\boldsymbol{A})|} = \frac{1}{\min_j |\lambda_j(\boldsymbol{A})|}$$

$$= \frac{1}{\frac{4}{h^2}\sin^2(h\pi/2)} = \frac{h^2}{4\sin^2(h\pi/2)}$$

We are interested in the $h \downarrow 0$ behavior of this quantity. Since,

$$\sin(x) \approx x \quad \text{as } x \downarrow 0,$$

we conclude that

$$\|\boldsymbol{A}^{-1}\|_2 \sim \frac{h^2}{4h^2\pi^2/4} = \frac{1}{\pi^2}$$

hence our scheme is stable since $\|\boldsymbol{A}^{-1}\|_2 \leqslant C$ for small h.

Stability, III

We can now finish our stability verification:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \qquad\qquad (\lambda_j, \boldsymbol{v}_j) = \left(\frac{4}{h^2} \sin^2(\pi j h/2), \sqrt{2}\sin(\boldsymbol{x} j \pi) \right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_M)^T$.

Since $\boldsymbol{A}$ is invertible and symmetric:

$$\begin{aligned}
\|\boldsymbol{A}^{-1}\|_2 = \sigma_1(\boldsymbol{A}^{-1}) &= \max_j |\lambda_j(\boldsymbol{A}^{-1})| = \max_j \frac{1}{|\lambda_j(\boldsymbol{A})|} = \frac{1}{\min_j |\lambda_j(\boldsymbol{A})|} \\
&= \frac{1}{\frac{4}{h^2}\sin^2(h\pi/2)} = \frac{h^2}{4\sin^2(h\pi/2)}
\end{aligned}$$

We are interested in the $h \downarrow 0$ behavior of this quantity. Since,

$$\sin(x) \approx x \quad \text{as } x \downarrow 0,$$

we conclude that

$$\|\boldsymbol{A}^{-1}\|_2 \sim \frac{h^2}{4h^2\pi^2/4} = \frac{1}{\pi^2}$$

hence our scheme is stable since $\|\boldsymbol{A}^{-1}\|_2 \leqslant C$ for small h.

Convergence, I

We are finally in a position to consider our original question: is our scheme accurate? The answer to this question will quantify how large the error e is:

$$\boldsymbol{e} = (e_1, \ldots, e_M)^T, \qquad\qquad e_j := u_j - u(x_j).$$

Definition

A scheme is *convergent* if $\lim_{h\downarrow 0} \|\boldsymbol{e}\|_2 = 0$.

This is a rather strong statement, since as $h \downarrow 0$ we require small error at a larger number of spatial points.

We can now show the power of *linearity* for this problem. Define a vector containing evaluations of the exact solution:

$$\boldsymbol{U} = (u(x_1), \ldots, u(x_M))^T.$$

Now note that,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{u} &= \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M && \text{(Definition of the scheme)} \\ \boldsymbol{A}\boldsymbol{U} &= \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M + \boldsymbol{\tau}, && \text{(Consistency)} \end{aligned}$$

Convergence, I

We are finally in a position to consider our original question: is our scheme accurate? The answer to this question will quantify how large the error e is:

$$\boldsymbol{e} = (e_1, \ldots, e_M)^T, \qquad\qquad e_j := u_j - u(x_j).$$

Definition

A scheme is *convergent* if $\lim_{h \downarrow 0} \|\boldsymbol{e}\|_2 = 0$.

This is a rather strong statement, since as $h \downarrow 0$ we require small error at a larger number of spatial points.

We can now show the power of *linearity* for this problem. Define a vector containing evaluations of the exact solution:

$$\boldsymbol{U} = (u(x_1), \ldots, u(x_M))^T.$$

Now note that,

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M \qquad \text{(Definition of the scheme)}$$

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{f} + \frac{g_0}{h^2}\boldsymbol{e}_1 + \frac{g_1}{h^2}\boldsymbol{e}_M + \boldsymbol{\tau}, \qquad \text{(Consistency)}$$

Convergence, II

Therefore:

$$\boldsymbol{A}\boldsymbol{e} = \boldsymbol{A}(\boldsymbol{u} - \boldsymbol{U}) = -\boldsymbol{\tau},$$

and so,

$$\|\boldsymbol{e}\|_2 = \|\boldsymbol{A}^{-1}\boldsymbol{\tau}\|_2 \leqslant \|\boldsymbol{A}^{-1}\|_2 \|\boldsymbol{\tau}\|_2 \leqslant C\mathcal{O}(h^2),$$

where the last inequality uses both stability and consistency.

We have just proven the following:

Theorem

The second-order difference scheme is convergent, and in particular is second-order convergent.

In this particular case, the order of the LTE coincides with the order of convergence. This is not always the case.

Convergence, II

Therefore:

$$\boldsymbol{A}\boldsymbol{e} = \boldsymbol{A}(\boldsymbol{u} - \boldsymbol{U}) = -\boldsymbol{\tau},$$

and so,

$$\|\boldsymbol{e}\|_2 = \|\boldsymbol{A}^{-1}\boldsymbol{\tau}\| \leqslant \|\boldsymbol{A}^{-1}\| \, \|\boldsymbol{\tau}\| \leqslant C\mathcal{O}(h^2),$$

where the last inequality uses both stability and consistency.

We have just proven the following:

Theorem

The second-order difference scheme is convergent, and in particular is second-order convergent.

In this particular case, the order of the LTE coincides with the order of convergence. This is not always the case.

Convergence, II

Therefore:

$$\boldsymbol{A}\boldsymbol{e} = \boldsymbol{A}(\boldsymbol{u} - \boldsymbol{U}) = -\boldsymbol{\tau},$$

and so,

$$\|\boldsymbol{e}\|_2 = \|\boldsymbol{A}^{-1}\boldsymbol{\tau}\| \leqslant \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\tau}\| \leqslant C\mathcal{O}(h^2),$$

where the last inequality uses both stability and consistency.

We have just proven the following:

Theorem

The second-order difference scheme is convergent, and in particular is second-order convergent.

In this particular case, the order of the LTE coincides with the order of convergence. This is not always the case.

Convergence for linear FD methods

We have drawn an outline for how to establish convergence for FD schemes.

Many details are specific to the problem + discretization at hand, but the broad strokes are somewhat general:

- *Consistency*: The local truncation error is small relative to mesh spacing h.
- *Stability*: The scheme behaves in a well-behaved way for small mesh spacing h.
- *Linearity*: The scheme residual when the global error is plugged in is equal to the local truncation error.

Thus, the following idea is true for linear FD schemes:

$$\text{Stability} + \text{Consistency} = \text{Convergence}$$

This is called the *Lax Equivalence Theorem*, or the *Lax-Richtmyer Theorem*.

One might really consider this a "meta-theorem", as the practitioner must decide on the precise definition of what consistency and stability mean.

Convergence for linear FD methods

We have drawn an outline for how to establish convergence for FD schemes.

Many details are specific to the problem + discretization at hand, but the broad strokes are somewhat general:

- *Consistency*: The local truncation error is small relative to mesh spacing h.
- *Stability*: The scheme behaves in a well-behaved way for small mesh spacing h.
- *Linearity*: The scheme residual when the global error is plugged in is equal to the local truncation error.

Thus, the following idea is true for linear FD schemes:

$$\text{Stability} + \text{Consistency} = \text{Convergence}$$

This is called the *Lax Equivalence Theorem*, or the *Lax-Richtmyer Theorem*.

One might really consider this a "meta-theorem", as the practitioner must decide on the precise definition of what consistency and stability mean.

Convergence for linear FD methods

We have drawn an outline for how to establish convergence for FD schemes.

Many details are specific to the problem + discretization at hand, but the broad strokes are somewhat general:

- *Consistency*: The local truncation error is small relative to mesh spacing h.
- *Stability*: The scheme behaves in a well-behaved way for small mesh spacing h.
- *Linearity*: The scheme residual when the global error is plugged in is equal to the local truncation error.

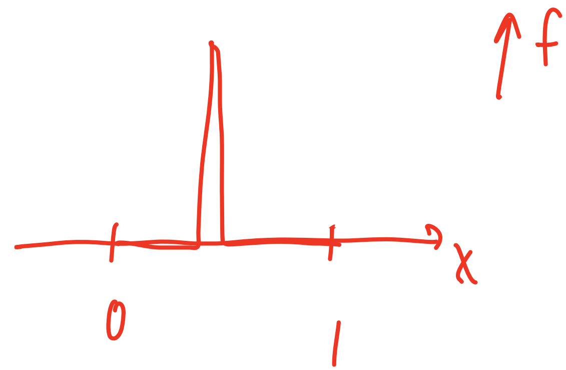
Thus, the following idea is true for linear FD schemes:

$$\text{Stability} + \text{Consistency} = \text{Convergence}$$

This is called the *Lax Equivalence Theorem*, or the *Lax-Richtmyer Theorem*.

One might really consider this a "meta-theorem", as the practitioner must decide on the precise definition of what consistency and stability mean.

Some generalizations



$-u'' = f$

Much of previous technique can be generalized to more complicated setups in 1D:

- Non-uniform grids (derive non-uniform versions of $D_{\pm,0}$
- Neumann/Robin boundary conditions (discretization of boundary conditions)
- Different error norms (e.g., L^{∞} norm error)
- Non-homogeneous diffusion: $(\kappa(x)u'(x))' = f(x)$

$u(0) = g_0$ (Dirichlet)

$u'(0) = h_0$ (Neumann)

$\alpha u(0) + \beta u'(0) = j_0$ (Robin)

References I

Lax, P. D. and R. D. Richtmyer (1956). "Survey of the Stability of Linear Finite Difference Equations". In: *Communications on Pure and Applied Mathematics* 9.2, pp. 267–293. ISSN: 1097-0312. DOI: 10.1002/cpa.3160090206. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160090206.

LeVeque, Randall J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM. ISBN: 978-0-89871-783-9.