

Spectral Analysis of finite difference method for convcetion-diffusion problems

Contents

1	Introduction	2
2	Spectral Analysis	2
2.1	Model problem in 1D	2
2.2	Finite difference methods	3
2.3	Spectral analysis	4
2.4	Amplification factors	4
2.5	Significance of the Amplification factor	5
3	Error Dynamics	6
4	Numerical Properties	8
5	Description of the code:	8

1 Introduction

Many physical phenomena are governed by the partial differential equations (PDEs) those model convection and/or diffusion processes. Presence of non-linear terms in such PDEs make the numerical analysis of their discrete models difficult. It is common to analyze numerical methods for a linear model problem, for example, linear convection or linear diffusion equation. Availability of exact solution and linearity of the equation enables one to quantify the performance of the numerical methods upto certain depth.

In this report, we write the spectral analysis of finite difference methods by following the approaches in [1, 2, 3]. The analysis written here is applicable to the following types of methods in the context in linear convection, diffusion and convection-diffusion equations:

- Explicit and implicit finite difference methods (for equi-spaced grids)
- Explicit and implicit time-integration methods
- Two and three level time-integration methods

This analysis identifies some quantities those determine the accuracy of the numerical method. A small code `afd_1d` is written in Fortran 90 that computes these quantities for some well known finite difference methods. However, any finite difference method that falls into above categories can be analyzed for its performance by easily modifying the code `afd_1d`.

2 Spectral Analysis

2.1 Model problem in 1D

For generality, we consider is the 1D linear convection-diffusion equation with infinite domain as a model problem for the analysis.

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad -\infty < x < \infty \quad (1)$$

$$u(x, t = 0) = f_0(x) \quad (2)$$

where, c is the convection speed and $\nu (> 0)$ is the rate of diffusion. The analysis can be reduced to the convection and diffusion equations by taking $\nu = 0$ and $c = 0$ respectively.

The dispersion relation for 1D advection-diffusion is obtained by substituting Fourier-Laplace transform $u(x, t) = \int \int \hat{U}(k, \omega) e^{i(kx - \omega t)} dk d\omega$ in Eq. 1.

$$\omega = kc - i\nu k^2 \quad (3)$$

Representing the variable u by Fourier transform $u(x, t) = \int \hat{U}(k, t) e^{ikx} dk$, Eq. 1 and 2 can be written as

$$\frac{d\hat{U}}{dt} + ick\hat{U} = -\nu k^2 \hat{U}$$

$$u(x, t = 0) = \int \hat{U}_0(k) e^{ikx} dk$$

Above ordinary differential equation can be solved to obtain

$$\hat{U}(k, t) = \hat{U}_0 e^{-ikct} e^{-\nu k^2 t} = \hat{U}_0 e^{-i\omega t}$$

Physically, e^{-ikct} propagates the initial condition with the speed c and $e^{-\nu k^2 t}$ damps the initial condition at the rate ν (assuming $\nu > 0$) for each wavenumber k . The suitable numerical methods are expected to follow these two physical aspects. Spectral analysis of finite difference methods given in [1, 2] provides the insight about how effectively finite differences follow these two aspects.

2.2 Finite difference methods

The spectral representation of the first and second derivatives can be written as:

$$u' = \int_{-\infty}^{\infty} ik \hat{U} e^{ikx} dk$$

$$u'' = - \int_{-\infty}^{\infty} k^2 \hat{U} e^{ikx} dk$$

Finite difference methods naturally arise from the Taylor series expansion. For higher resolution, implicit finite difference methods based on the Padé's approximation are also widely used. Assuming that a 1D domain is discretized to obtain an equi-spaced grid of N nodes, the numerical derivative u'_j at any node can be written in terms of the discrete function u_j as

$$[A]\{u'_j\} = \frac{1}{h}[B]\{u_j\}$$

where entries of $[A]$ and $[B]$ comes from the coefficients of the finite difference method and h denotes the grid-spacing. For explicit finite difference methods $[A]$ is an identity matrix. Inverting $[A]$,

$$\{u'_j\} = \frac{1}{h}[C]\{u_j\}$$

Alternatively

$$u'_j = \frac{1}{h} \sum_{l=1}^N C_{jl} u_l$$

Again, by Fourier representation $u_l = \int \hat{U}(k) e^{ikx_l} dk$ and little algebra, we can write

$$u'_j = \frac{1}{h} \int \sum_{l=1}^N C_{jl} \hat{U} e^{ik(x_l - x_j)} e^{ikx_j} dk \quad (4)$$

Similarly, a finite difference method for the second order derivative can be written as

$$u''_j = \frac{1}{h^2} \int \sum_{l=1}^N D_{jl} \hat{U} e^{ik(x_l - x_j)} e^{ikx_j} dk \quad (5)$$

The similar representation can be written for the derivative of any order with appropriate finite difference method. We restrict here upto two derivatives on the account of the model problem considered.

2.3 Spectral analysis

Substituting Eq. 4 and 5 in 1 and again using Fourier representation

$$\int \left[\frac{\partial \hat{U}}{\partial t} + \frac{c}{h} C_{jl} \hat{U} e^{ik(x_l - x_j)} - \frac{\nu}{h^2} D_{jl} \hat{U} e^{ik(x_l - x_j)} \right] e^{ikx_j} dk = 0$$

Since the integrand must be zero for any k ,

$$\frac{d\hat{U}}{dt} \Big|_j = \left[-\frac{c}{h} C_{jl} e^{ik(x_l - x_j)} + \frac{\nu}{h^2} D_{jl} e^{ik(x_l - x_j)} \right] \hat{U}$$

Using separation of variables,

$$\frac{d\hat{U}}{\hat{U}} \Big|_j = \left[-\frac{cdt}{h} C_{jl} e^{ik(x_l - x_j)} + \frac{\nu dt}{h^2} D_{jl} e^{ik(x_l - x_j)} \right]$$

Here $\frac{cdt}{h}$ is the CFL number Nc and $\frac{\nu dt}{h^2}$ is the Peclet number Pe . Hence

$$\frac{d\hat{U}}{\hat{U}} \Big|_j = - \left[Nc C_{jl} e^{ik(x_l - x_j)} - Pe D_{jl} e^{ik(x_l - x_j)} \right] = -A_j \quad (6)$$

In above equation, A_j contains the information of finite difference approximations. For any choice of the time-integration method, above equation can be simplified to give amplification factor G for the combination of the finite difference method for space derivatives and time integration method.

2.4 Amplification factors

The amplification factor is defined as the ratio of the amplitudes at two successive time levels.

$$G = \frac{\hat{U}^{n+1}}{\hat{U}^n}$$

where superscript denotes the index for time discretization. Amplification factors for some well-known two and three level methods are written below.

Explicit Euler Method

For Euler's forward method, Eq. 6 becomes

$$\frac{\hat{U}^{n+1} - \hat{U}^n}{\hat{U}^n} = \frac{\hat{U}^{n+1}}{\hat{U}^n} - 1 = G_j - 1 = -A_j$$

Hence,

$$G_j = 1 - A_j$$

The subscript j is retained as this amplification factor is specific to the node j .

Implicit Euler Method

$$G_j = \frac{1}{1 - A_j}$$

Classical Runge-Kutta second order method

$$G_j = 1 - A_j + \frac{A_j^2}{2}$$

Classical Runge-Kutta third order method

$$G_j = 1 - A_j + \frac{A_j^2}{2} - \frac{A_j^3}{6}$$

Classical Runge-Kutta forth order method

$$G_j = 1 - A_j + \frac{A_j^2}{2} - \frac{A_j^3}{6} + \frac{A_j^4}{24}$$

Crank-Nicholson Method

$$G_j = \frac{1 + 0.5A_j}{1 - 0.5A_j}$$

Richardson Method

$$G_j = A_j \pm \sqrt{A_j^2 + 1}$$

Gears or BDF2 Method

2.5 Significance of the Amplification factor

The amplification factor acts as a numerical dispersion relation. Therefore, it governs the dynamics of the numerical solution. Let us consider the evolution of the solution through first time step i.e. from $t = 0$ to $t = \Delta t$, which corresponds to the amplification factor G_j^1 .

$$u_j^1 = \int \hat{U}_0(k) G_j^1 e^{ikx_j} dk$$

Now using above numerical solution as an initial condition, we can write the evolution of the numerical solution through second time step i.e. from $t = \Delta t$ to $t = 2\Delta t$, which corresponds to the amplification factor G_j^2 . Therefore, the numerical solution at $t = 2\Delta t$ can be written as:

$$u_j^2 = \int \hat{U}_0(k) G_j^1 G_j^2 e^{ikx_j} dk$$

Therefore after n time steps,

$$u_j^n = \int \hat{U}_0(k) \left[\prod_{m=1}^n G_j^m \right] e^{ikx_j} dk$$

Since the amplification factor is a complex quantity,

$$u_j^n = \int \hat{U}_0(k) \left[\prod_{m=1}^n |G_j^m| \right] \left[\sum_{m=1}^n e^{i\beta_j^m} \right] e^{ikx_j} dk$$

where $\beta^m = \tan^{-1}(G_{ji}^m/G_{jr}^m)$. The product term in above equation represents the cumulative numerical amplification/damping while the summation term represents the cumulative numerical

phase speed. Each quantity in the product term can be modeled as the numerical amplification while each quantity in the summation term can be modeled as numerical phase speed. Therefore,

$$u_j^n = \int \hat{U}_0(k) \left[e^{-k^2 \sum_{m=1}^n (\nu_N^m \Delta t^m)} \right] \left[e^{-ik \sum_{m=1}^n (c_N^m \Delta t^m)} \right] e^{ikx_j} dk$$

Assuming that the time-step is kept constant,

$$u_j^n = \int \hat{U}_0(k) \left[e^{-k^2 \Delta t \sum_{m=1}^n \nu_N^m} \right] \left[e^{-ik \Delta t \sum_{m=1}^n c_N^m} \right] e^{ikx_j} dk$$

In the analogy with the physical dispersion relation, these two terms can be modeled as $e^{-\nu_N k^2 \Delta t}$ and $e^{-ik c_N \Delta t}$ respectively. Here, ν_N and c_N represents the average damping and phase speed, respectively due to all the steps in the computations. In the ideal situation, $\Delta t \sum_{m=1}^n \nu_N^m$ and $\Delta t \sum_{m=1}^n c_N^m$ should be equal to their physical counterparts νt and ct respectively. If time-step is variable then, $\sum_{m=1}^n (\nu_N^m \Delta t^m)$ and $\sum_{m=1}^n (c_N^m \Delta t^m)$ should be equal to their physical counterparts νt and ct respectively. It should be noted that the each term ν_N^m and c_N^m is the function of wavenumber k . For a constant time-step Δt above equation can be written as:

$$u_N = \int \hat{U}_0(k) e^{-\nu_N k^2 \Delta t} e^{-ik c_N \Delta t} e^{ikx} dk \quad (7)$$

3 Error Dynamics

Following the steps in [1], equation for the error dynamics of 1D convection-diffusion equation can be obtained. Let us define error e as the difference between exact and numerical solution $u - u_N$. Therefore,

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} - \nu \frac{\partial^2 e}{\partial x^2} = -\frac{\partial u_N}{\partial t} - c \frac{\partial u_N}{\partial x} + \nu \frac{\partial^2 u_N}{\partial x^2} \quad (8)$$

Taking first and second derivative of Eq. 7 w.r.t. x ,

$$\begin{aligned} \frac{du_N}{dx} &= i \int k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-ik c_N t} e^{ikx} dk \\ \frac{d^2 u_N}{dx^2} &= - \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-ik c_N t} e^{ikx} dk \end{aligned}$$

Similarly, differentiating Eq. 7 w.r.t. t ,

$$\frac{du_N}{dt} = \int -(\nu_N k^2 + i c_N k) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-ik c_N t} e^{ikx} dk$$

Substituting these derivatives in the Eq. 8,

$$\begin{aligned}
& \frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} - \nu \frac{\partial^2 e}{\partial x^2} \\
&= \int (\nu_N k^2 + i c_N k) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - c \int i k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&- \nu \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&= \int (\nu_N k^2 + i c_N k) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - \int (\nu k^2 + i k c) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk
\end{aligned}$$

This equation highlights error does not follow the same dynamics as the underlying PDE, but has the source term which is a consequence of numerical dispersion relation which is different than physical dispersion relation. With some algebra, above equation can be further explored:

$$\begin{aligned}
& \frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} - \nu \frac{\partial^2 e}{\partial x^2} \\
&= \int (\nu_N k^2 + i c_N k) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - \int (\nu k^2 + i k c) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&+ \nu_N \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - \nu_N \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&+ c_N \int i k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - c_N \int i k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&= (\nu_N - \nu) \frac{\partial^2 u_N}{\partial x^2} + (c_N - c) \frac{\partial u_N}{\partial x} + \int (\nu_N k^2 + i c_N k) \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&- \nu_N \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - c_N \int i k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&= (\nu_N - \nu) \frac{\partial^2 u_N}{\partial x^2} + (c_N - c) \frac{\partial u_N}{\partial x} \\
&+ \int \nu_N k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - \nu_N \int k^2 \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk \\
&+ \int i c_N k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk - c_N \int i k \hat{U}_0(k) e^{-\nu_N k^2 t} e^{-i k c_N t} e^{i k x} dk
\end{aligned}$$

Using integration by parts, the PDE for error dynamics can be written as

$$\begin{aligned}
& \frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} - \nu \frac{\partial^2 e}{\partial x^2} = (\nu_N - \nu) \frac{\partial^2 u_N}{\partial x^2} + (c_N - c) \frac{\partial u_N}{\partial x} \\
&- \int \frac{d\nu_N}{dk} \left[\int k'^2 \hat{U}_0(k') e^{-\nu_N k'^2 t} e^{-i k' c_N t} e^{i k' x} dk' \right] dk \\
&- \int \frac{dc_N}{dk} \left[\int i k' \hat{U}_0(k') e^{-\nu_N k'^2 t} e^{-i k' c_N t} e^{i k' x} dk' \right] dk
\end{aligned}$$

The source term contains:

- $(c_N - c)$: phase speed error, arises due to approximate modeling of the first derivative.
- $(\nu_N - \nu)$: diffusion error, arises due to approximate modeling of the second derivative.

- $\frac{dc_N}{dk}$ and $\frac{d\nu_N}{dk}$: dispersion error, arise due to approximate modeling of the PDE.

In the continuum limit where $\nu_N = \nu$ and $c_N = c$, the source term in the equation of error dynamics vanish and we recover the von Neumann's prediction that the solution and computational error follows the same dynamics. For practical situations von Neumann's prediction may not hold.

4 Numerical Properties

Since, expression of G_j can be found from the choice of numerical scheme, amplification factor $|G_j|$, phase speed β_j are readily available. Although all the information is contained in G_j , we consider following quantities as the numerical properties of the finite difference methods:

- When diffusion is present, numerical amplification factor $|G_j|$ should model physical amplification factor $G_{ex} = e^{-Pe(kh)^2}$. For convection problems $|G_j|$ should be equal to 1, since pure convection problem does not amplify or attenuate the solution.
- For the quantification of phase error, we choose to compute the quantity $\frac{c_N}{c} = \frac{\beta_j}{(kh)(Nc)}$. Deviation of this quantity from 1 represent the phase error.
- For the quantification of diffusion error, we choose to compute the quantity $\frac{\nu_N}{\nu} = \frac{-\log(|G_j|)}{(Pe)(kh)^2}$. Deviation of this quantity from 1 represent the diffusion error i.e. solution is either over or under damped depending on whether this quantity is > 1 or < 1 respectively.
- For the quantification of the dispersion error, we choose to (numerically) differentiate $\frac{c_N}{c}$ and $\frac{\nu_N}{\nu}$ w.r.t. k . Their expressions are written as:

$$\frac{d}{d(kh)} \left[\frac{c_N}{c} \right] = \frac{1}{(Nc)(kh)} \left[\frac{d\beta_j}{d(kh)} - \frac{\beta_j}{(kh)} \right]$$

$$\frac{d}{d(kh)} \left[\frac{\nu_N}{\nu} \right] = \frac{-1}{(Pe)(kh)^2} \left[\frac{d}{d(kh)} \log|G_j| - 2 \frac{\log|G_j|}{(kh)} \right]$$

5 Description of the code:

- Go to src directory and type `make clean` to remove previous object and executable files.
- Then run the command `make` to build the executable `afd_1d`.
- Edit `input.in` to select the model problem and available finite-difference schemes. Finite-difference schemes need to be added in the code if not available.
- Now run the code using command `./afd_1d input.in`.

References

- [1] T. K. SENGUPTA, *High accuracy computing methods: fluid flows and wave phenomena*, Cambridge University Press, New York, USA (2013).
- [2] CHARLES HIRSCH, *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*, John Wiley & Sons, Inc. New York, NY, USA (1988).

- [3] V. K. SUMAN, T. K. SENGUPTA, C. JYOTHI DURGA PRASAD, K. SURYA MOHAN AND D. SANWALIA, *Spectral analysis of finite difference schemes for convection diffusion equation*, Comput. Fluids, 150 (2017), pp. 95–114.