Finite Difference Methods Basics

Zhilin Li
Center for Research in Scientific Computation & Department of Mathematics
North Carolina State University
Raleigh, NC 27695, e-mail: zhilin@math.ncsu.edu

1 Examples of differential equations in one space dimension

- Newton cooling model:

$$\frac{du}{dt} = c(u_{sur} - u), \quad t > 0 \qquad u(0) = u_0 \quad \text{is given} \tag{1}$$

This is an ordinary differential equation (ODE) with an initial value $u(0)$ is given. It is also called an initial value problem (IVP).

- Heat diffusion and heat equations.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\beta \frac{\partial u}{\partial x}\right), \quad a < x < b, \quad t > 0, \tag{2}$$

$$= \beta \frac{\partial^2 u}{\partial x^2}, \qquad \text{if } \beta \text{ is a constant}, \tag{3}$$

$$u(a,t) = g_1(t), \qquad u(b,t) = g_2(t), \quad \text{the boundary condition}, \tag{4}$$

$$u(x,0) = u_0(x), \qquad \text{the initial condition}. \tag{5}$$

The equation is a partial differential equation (PDE) with initial and boundary conditions. It is also called an initial and boundary value problem (IBVP).

The following equation is called one dimensional linear parabolic equation with a source;

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\beta \frac{\partial u}{\partial x}\right) + \alpha(x,t)\frac{\partial u}{\partial t} + \gamma(x,t)u + f(x,t), \quad a < x < b, \quad t > 0, \tag{6}$$

with a boundary condition at two ends and an initial condition.

- The one way wave equation (first order PDE)

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad a < x < b, \quad t > 0, \tag{7}$$

$$u(a,t) = g_1(t), \qquad \text{if } c > 0, \quad \text{the boundary condition}, \tag{8}$$

$$u(x,0) = u_0(x), \qquad \text{the initial condition}, \tag{9}$$

where c is called the wave speed. Note that there is only one boundary condition.

The following equation is called one dimensional first order linear PDE

$$\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x} + \alpha(x,t)u + f(x,t), \quad a < x < b, \quad t > 0, \tag{10}$$

with a boundary condition at one end and an initial condition.

- One-dimensional elliptic equations (steady state solution of the parabolic differential equation).

$$\frac{\partial}{\partial x}\left(\beta\frac{\partial u}{\partial x}\right) + \alpha(x)\frac{\partial u}{\partial x} + \gamma(x)u = f(x), \quad a < x < b, \tag{11}$$

$$u(a) = u_a, \qquad u(b) = u_b, \qquad \text{the boundary condition} \tag{12}$$

It is an ODE and is called a two-point boundary value problem (BVP). It is an elliptic type differential equation.

1.1 General linear second order partial differential equations:

$$a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} + d(x,y)u_x + e(x,y)u_y + g(x,y)u = f(x,y), \tag{13}$$

where $(x,y) \in \Omega$ is a domain in x-y coordinates. The PDE above can be classified as one of the following:

- Elliptic equation if $b^2 - ac < 0$ for all $(x,y) \in \Omega$. Examples include the Laplace equation

$$u_{xx} + u_{yy} = 0, \tag{14}$$

and the Poisson equation

$$u_{xx} + u_{yy} = f(x,y). \tag{15}$$

- Parabolic equation if $b^2 - ac = 0$ for all $(x,y) \in \Omega$. One example is the heat equation with/without a source

$$u_t = \beta u_{xx} + f(x,t). \tag{16}$$

An example in two space dimensions is

$$u_t = \beta(u_{xx} + u_{yy}) + f(x,y,t). \tag{17}$$

- Hyperbolic equation if $b^2 - ac > 0$ for all $(x,y) \in \Omega$. Example includes the two way wave equation

$$u_{tt} = c^2 u_{xx}. \tag{18}$$

- Mixed type if $b^2 - ac$ changes sign in Ω.

2 Finite difference approximations

A finite difference method typically involves the following steps:

1. Generate a grid, for example $(x_i, t^{(k)})$, where we want to find an approximate solution.

2. Substitute the derivatives in an ODE/PDE or an ODE/PDE system of equations with finite difference schemes. The ODE/PDE then become a linear/non-linear system of algebraic equations.

3. Solve the system of algebraic equations.

4. Implement and debug the computer code.

5. Do the error analysis, both analytically and numerically.

Note that besides *Finite Difference* methods, there are other methods that can be used to solve ODE/PDEs such as *Finite element methods, spectral methods* etc. Generally finite difference are simple to use for problems defined on regular geometries, such as an interval in 1D, a rectangular domain in two space dimensions, and a cubic in three space dimensions.

3 Some commonly used finite difference formulas

Below we list three commonly used finite difference formulas to approximate the first order derivative of a function $u(x)$ using the function values only.

- The forward finite difference

$$D_+u(x) = \frac{u(x+h) - u(x)}{h} = u'(x) + \frac{1}{2}u''(\xi)h. \tag{19}$$

  Therefore the error (absolute error) of the forward finite difference is proportional to h and the approximation is referred as a first order approximation.

- The backward finite difference

$$D_-u(x) = \frac{u(x) - u(x-h)}{h} = u'(x) - \frac{1}{2}u''(\xi)h. \tag{20}$$

  The approximation again is a first order approximation.

- The central finite difference

$$D_0u(x) = \frac{u(x+h) - u(x-h)}{2h} = u'(x) + \frac{h^2}{6}u'''(\xi). \tag{21}$$

The error is proportional to h^2 and the approximation is referred as a second order approximation. Note that

$$D_0 u(x) = \frac{1}{2}\left(D_+ u(x) + D_+ u(x)\right) \tag{22}$$

3.1 Approximation to high order derivatives

Usually finite difference schemes for high order derivatives can be obtained from the formulas for lower order derivatives. The central finite difference scheme for the second order derivative u_{xx} is

$$\begin{aligned} D_0^2 u(x) &= D_+ D_-\, u(x) = D_+ \frac{u(x) - u(x-h)}{h} \\ &= \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = u_{xx}(x) + \frac{h^2}{12}\, u^{(4)}(\xi). \end{aligned} \tag{23}$$

So it is a second order approximation. The third order derivative u_{xxx} can be approximated, for example

$$\begin{aligned} D^3 u(x) &= D_+ D_0^2 u(x) = \frac{u(x+2h) - 3u(x+h) + 3u(x) - u(x-h)}{h^3} \\ &= u_{xxx}(x) + \frac{h}{2} u^{(4)}(\xi). \end{aligned} \tag{24}$$

There are other approaches to construct finite difference schemes, for example, the method of un-determined coefficients, polynomial interpolations and others.

4 Finite difference approximations for model problems

4.1 Newton cooling model

The differential equation is:

$$\frac{du}{dt} = c(u_{sur} - u), \quad t > 0 \qquad u(0) = u_0 \quad \text{is given}$$

The finite difference equation is

$$\frac{u^{k+1} - u^k}{\Delta t} = c(u_{sur} - u^k), \quad u^0 = u_0 \tag{25}$$

where u^k is an approximation to the solution $u(t^k) = u(k\Delta t)$. The forward finite difference approximation to u_t is called the forward Euler method.

4.2 General 1D parabolic equation

Consider the differential equation:

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} + f(x,t), \quad a < x < b, \quad t > 0.$$

In order to derive a finite difference scheme, we need to generate a grid

$$\begin{aligned} x_i &= a + ih, \quad i = 0, 1, \cdots, m, \quad h = \frac{b-a}{m}, \\ t^k &= k\,\Delta t, \quad k = 0, 1, \cdots. \end{aligned}$$

The Δt should satisfy the time step restriction (to guarantee the stability)

$$0 < \Delta t \le \frac{h^2}{2\beta}. \tag{26}$$

The finite difference scheme is

$$\begin{aligned} \frac{u_i^{k+1} - u_i^k}{\Delta t} &= \beta \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h^2} + f_i^k, \\ k = 0, 1, \cdots, \qquad & i = 1, 2, \cdots, m-1, \quad \text{if } u(a,t) \text{ and } u(b,t) \text{ are given,} \end{aligned} \tag{27}$$

where u_i^k is an approximation to $u(x,t)$, and $f_i^k = f(x_i, t^k)$. One of finite difference schemes for 1D general parabolic equation

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} + \alpha(x,t) \frac{\partial u}{\partial t} + \gamma(x,t) u + f(x,t),$$

is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \beta \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h^2} + \alpha_i^k \, \frac{u_{i+1}^k - u_{i-1}^k}{2h} + \gamma_i^k \, u_i^k + f_i^k. \tag{28}$$

However, the time step restriction is different.

4.3 The one way wave equation

The differential equation is:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = f(x,t), \quad a < x < b, \quad t > 0.$$

A simple finite difference scheme is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + c \, \frac{u_i^k - u_{i-1}^k}{h} = f_i^k, \quad i = 0, 1, \cdots, \qquad \text{if } c > 0. \tag{29}$$

Question: Can we use the central finite difference approximation for u_x?

4.4 One dimensional elliptic equations.

The differential equation is:

$$\begin{aligned} \beta \frac{\partial^2 u}{\partial x^2} + \alpha(x) \frac{\partial u}{\partial x} + \gamma(x) u &= f(x), \quad a < x < b, \\ u(a) = u_a, \qquad & u(b) = u_b, \qquad \text{the boundary condition.} \end{aligned}$$

The second order finite difference approximation is

$$\beta \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + \alpha_i \frac{u_{i+1} - u_{i-1}}{2h} + \gamma_i u_i = f_i, \quad i = 1, 2, \cdots m - 1. \tag{30}$$

Note we need to solve a linear system of equations to get the approximate solution to the differential equation. **Question:** Can you write down the system of equations in matrix-vector form?

In order to have a finite difference scheme work, two conditions need to be satisfied: the consistence and the stability.

5 The local truncation error and consistence of a finite difference scheme

The local truncation error is defined as the difference of the differential equation and the finite difference scheme. The finite difference scheme is called *consistent* if the limit of the local truncation error is zero as h and/or Δt approach zero.

The local truncation errors for the forward, backward, and central finite difference are

$$\begin{aligned} T_h(D_+) &= f'(x) - \frac{f(x+h) - f(x)}{h} = \frac{h}{2} f''(\xi) \\ T_h(D_-) &= f'(x) - \frac{f(x) - f(x-h)}{h} = -\frac{h}{2} f''(\xi) \\ T_h(D_0) &= f'(x) - \frac{f(x+h) - f(x-h)}{2h} = \frac{h^2}{6} f^{(4)}(\xi). \end{aligned}$$

In all three cases, we have $\lim_{h \to 0} T_h = 0$. Therefore they are all consistent.

5.1 The truncation error of the finite difference method for the Newton cooling model (25)

$$T_h = \frac{u(t+h) - u(t)}{\Delta t} - c(u_{sur} - u(t)). \tag{31}$$

Therefore, the local truncation error is simply the result of the finite difference equation with exact solution plugged in. Using the extended mean value theorem, we can get

$$T_h = \frac{\Delta t}{2} u''(\xi), \quad \text{and} \quad \lim_{\Delta t \to 0} T_h = 0. \tag{32}$$

So the finite difference equation is consistent.

A Remark: In the literature, there is another definition of the truncation which is

$$\bar{T_h} = u(t+h) - u(t) - c\,\Delta t\,(u_{sur} - u(t)). \tag{33}$$

We can see that

$$\bar{T_h} = \Delta t \, T_h. \tag{34}$$

5.2 The truncation error of the finite difference method for the 1D parabolic equation (27)

$$T_h = \frac{u(x,t+h) - u(x,t)}{\Delta t} - \beta \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} - f(x,t). \tag{35}$$

Again the local truncation error is simply the result of the finite difference equation with exact solution plugged in. Using the extended mean value theorem, we can get

$$\begin{aligned}
T_h &= u_t(x,t) + \frac{\Delta t}{2}\, u_{tt}(x,\eta) - \beta \left(u_{xx}(x,t) + \frac{h^2}{6} u_{xxxx}(\xi,t) \right) - f(x,t) && (36)\\
&= \frac{\Delta t}{2}\, u_{tt}(x,\eta) - \beta \frac{h^2}{6} u_{xxxx}(\xi,t) && (37)\\
&= O(\Delta t, h^2). && (38)
\end{aligned}$$

The method therefore is called first order in time and second order in space. Since

$$\lim_{\Delta t \to 0,\, h \to 0} T_h = 0, \tag{39}$$

the finite difference scheme is consistent.

5.3 The truncation error of the finite difference method for the 1D one-way wave equation (29)

$$\begin{aligned}
T_h &= \frac{u(x,t+h) - u(x,t)}{\Delta t} + c \frac{u(x+h,t) - u(x,t)}{h} - f(x,t), \quad c > 0 && (40)\\
&= u_t(x,t) + \frac{\Delta t}{2}\, u_{tt}(x,\eta) - c u_x(x,t) - c \frac{h^2}{2}\, u_{xx}(\xi,t) && (41)\\
&= O(\Delta t, h), \qquad \lim_{\Delta t \to 0,\, h \to 0} T_h = 0. && (42)
\end{aligned}$$

Therefore the finite difference scheme is consistent and is first order in time and first order in space.

6 The stability of a finite difference scheme

As we see from the numerical test, whether a finite difference scheme can work depend on the choice of Δt even if the finite difference scheme is consistent. The stability condition is a requirement that the error in the computed solution would be amplified in the subsequent computations. An intuitive definition is that

$$|u_1^{k+1}| + |u_2^{k+1}| + |u_3^{k+1}| + \cdots + |u_{m-2}^{k+1}| + |u_{m-1}^{k+1}| + |u_m^{k+1}| \leq \\ |u_1^k| + |u_2^k| + |u_3^k| + \cdots + |u_{m-2}^k| + |u_{m-1}^k| + |u_m^k| \tag{43}$$

The stability condition of the finite difference (25) for Newton cooling model is

$$0 < \Delta t \leq \frac{2}{c}. \tag{44}$$

This can be proved through the von Neumann analysis by setting

$$u^{k+1} = g\, u^k \tag{45}$$

Then the scheme is stable if and only if $|g| \leq 1$.

The stability condition of the finite difference scheme (27) for the heat equation with a source term is

$$0 < \Delta t \leq \frac{h^2}{2\beta}. \tag{46}$$

This can be proved using the von Neumann analysis by setting

$$u_j^k = e^{ijh\xi}, \quad u_j^{k+1} = g(\xi) e^{ijh\xi}, \tag{47}$$

where $i = \sqrt{-1}$. Then the finite difference scheme (27) is stable if $|g| \leq 1$.

The stability condition of the finite difference scheme (29) for the one way wave equation is

$$0 < \Delta t \leq \frac{h}{c}. \tag{48}$$

using the von Neumann analysis.

The global error (overall error) of a finite difference scheme is the absolute error of the computed solution. For the solution of the finite difference scheme for the Newton cooling model, it is

$$E = u^{k_{final}} - u(k_{final}\Delta t) \tag{49}$$

For one dimensional heat equation and one way wave equation, the global error is

$$E_i = u_i^{k_{final}} - u(x_i, t_{final}), i = 1, 2, \cdots, m \tag{50}$$

Usually we use one measurement called the infinity normal of the error

$$||E||_\infty = \max\{\, |E_1|, |E_2|, \cdots, |E_{m-1}|, |E_m| \,\} \tag{51}$$

A finite difference method is *convergent* to the true solution if the global error approaches to zero as Δ and h approach to zero.

Theorem 1 *A consistent and stable finite difference method is convergent.*

7 Implicit Discretization

7.1 Backward Euler for IVP of an ODE system.

Given an initial value problem of the following form

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}, t), \quad \mathbf{y}(0) = \mathbf{y}_0.$$

the forward Euler method is

$$\frac{\mathbf{y}^{k+1} - \mathbf{y}^k}{\Delta t} = f(\mathbf{y}^k, t^k), \quad k = 0, 1, \cdots$$

Usually there is a time step constraint on Δt. For example, for the Newton's cooling law

$$y' = c\,(y_{sur} - y), \qquad c > 0.$$

The forward finite difference method is

$$\frac{y^{k+1} - y^k}{\Delta t} = c\,(y_{sur} - y^k).$$

We need to choose $0 < \Delta t \le \dfrac{1}{c}$ to get a stable algorithm so that the computed solution is a reasonable approximation to the true solution of the ODE.

If we use the backward finite difference, often called the backward Euler method, for $y'(t)$ at t^{k+1}, then we get the backward Euler method

$$\frac{\mathbf{y}^{k+1} - \mathbf{y}^k}{\Delta t} = f(\mathbf{y}^{k+1}, t^{k+1}), \quad k = 0, 1, \cdots \tag{52}$$

The method works for much larger Δt that that of implicit method. For example, for the Newton's cooling law, the backward Euler method is

$$\frac{y^{k+1} - y^k}{\Delta t} = c\,(y_{sur} - y^{k+1}).$$

We can choose any Δt and the solution will not blow-up. However, to get reasonable accuracy, we still need to choose Δt small because the global error is proportional to Δt. We do not need to worry about the stability anymore.

If we use the central finite difference scheme for $y'(t)$,

$$\frac{\mathbf{y}^{k+1} - \mathbf{y}^{k-1}}{\Delta t} = f(\mathbf{y}^k, t^k), \quad k = 0, 1, \cdots \tag{53}$$

the scheme may or may not be stable, however, we need to use a different method to compute $\mathbf{y}^1$ before we can get started.

Quite often, the Crank-Nicolson method is an ideal method because it has second order accuracy and it is implicit. The Crank-Nicolson method for the IVP-ODE is

$$\frac{\mathbf{y}^{k+1} - \mathbf{y}^{k-1}}{\Delta t} = \frac{1}{2}\left(f(\mathbf{y}^k, t^k) + f(\mathbf{y}^{k+1}, t^{k+1})\right). \tag{54}$$

The global error is $O((\Delta t)^2)$.

7.2 The 1D heat equations

For the heat diffusion equation

$$u_t = \beta u_{xx} + f(x,t), \qquad \beta > 0,$$

the forward Euler method (FW-CT, meaning forward in time, central in space) is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \beta \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h^2} + f_i^k.$$

It is first order in time and second order in space ($O(\Delta t + h^2)$). There is a *severe* time step size constraint $0 < \Delta t \le h^2/(2\beta)$ which potentially makes the method very slow.

Usually we prefer to use $\Delta t \sim h$ so that we can advance quickly. The backward Euler method (BW-CT) scheme is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \beta \frac{u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}}{h^2} + f_i^{k+1}. \tag{55}$$

The truncation error is $O(\Delta t + h^2)$. It is unconditionally stable meaning that we can take any Δt.

However, we need to solve a linear system of equations below

$$\begin{bmatrix} \frac{1}{\Delta t} + \frac{2\beta}{h^2} & -\frac{\beta}{h^2} & 0 & \cdots & 0 \\ -\frac{\beta}{h^2} & \frac{1}{\Delta t} + \frac{2\beta}{h^2} & -\frac{\beta}{h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -\frac{\beta}{h^2} & \frac{1}{\Delta t} + \frac{2\beta}{h^2} & -\frac{\beta}{h^2} \\ 0 & \cdots & 0 & -\frac{\beta}{h^2} & \frac{1}{\Delta t} + \frac{2\beta}{h^2} \end{bmatrix} \begin{bmatrix} u_1^{k+1} \\ u_2^{k+1} \\ \vdots \\ u_{m-2}^{k+1} \\ u_{m-1}^{k+1} \end{bmatrix} = \begin{bmatrix} f_1^{k+1} + \frac{\beta}{h^2} u(0, t^{k+1}) \\ f_2^{k+1} \\ \vdots \\ f_{m-2}^{k+1} \\ f_{m-1}^{k+1} + \frac{\beta}{h^2} u(L, t^{k+1}) \end{bmatrix}$$

to get an approximate solution of the differential equation at $t = t^{k+1}$ The system of equations is a tridiagonal and it is very to solve, in Matlab, we can simply use the command $\mathbf{u} = A \backslash \mathbf{b}$, see *heat_im.m* for example.

7.3 Central discretization for u_t?

The central finite difference of u_t for heat diffusion equations is unconditionally unstable (no matter what Δt we choose).

7.4 The Crank-Nicolson scheme for 1D heat equation.

If we want to take $\Delta t \sim h$ and keep second order accuracy, then we can use the Crank-Nicolson scheme

$$\frac{u_i^{k+1}-u_i^k}{\Delta t} = \frac{\beta}{2}\left(\frac{u_{i-1}^{k+1}-2u_i^{k+1}+u_{i+1}^{k+1}}{h^2}+\frac{u_{i-1}^{k}-2u_i^{k}+u_{i+1}^{k}}{h^2}\right)+f_i^{k+\frac{1}{2}} \tag{56}$$

The local truncation error is $O((\Delta t)^2+h^2)$ and it is unconditionally unstable (no matter what Δt we choose). We usually take $\Delta t = h$. The global error is $O(h^2+\Delta t)^2)$. However, we need to solve the system of equations $A\mathbf{u}^{k+1} = B\mathbf{u}^k + \mathbf{b}^k$ to get the solution at time $t^{k+1} = (k+1)\Delta t$, where A, B, and $\mathbf{b}^k$ are given below:

$$A = \begin{bmatrix} \frac{1}{\Delta t}+\frac{\beta}{h^2} & -\frac{\beta}{2h^2} & 0 & \cdots & 0 \\ -\frac{\beta}{2h^2} & \frac{1}{\Delta t}+\frac{\beta}{h^2} & -\frac{\beta}{2h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -\frac{\beta}{2h^2} & \frac{1}{\Delta t}+\frac{\beta}{h^2} & -\frac{\beta}{2h^2} \\ 0 & \cdots & 0 & -\frac{\beta}{2h^2} & \frac{1}{\Delta t}+\frac{\beta}{h^2} \end{bmatrix}, \quad \mathbf{b}^k = \begin{bmatrix} f_1^{k+\frac{1}{2}}+\frac{\beta}{h^2}\,u(0,t^{k+\frac{1}{2}}) \\ f_2^{k+\frac{1}{2}} \\ \vdots \\ f_{m-2}^{k+\frac{1}{2}} \\ f_{m-1}^{k+\frac{1}{2}}+\frac{\beta}{h^2}\,u(L,t^{k+\frac{1}{2}}) \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{\Delta t}-\frac{\beta}{h^2} & \frac{\beta}{2h^2} & 0 & \cdots & 0 \\ \frac{\beta}{2h^2} & \frac{1}{\Delta t}-\frac{\beta}{h^2} & \frac{\beta}{2h^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \frac{\beta}{2h^2} & \frac{1}{\Delta t}-\frac{\beta}{h^2} & \frac{\beta}{2h^2} \\ 0 & \cdots & 0 & \frac{\beta}{2h^2} & \frac{1}{\Delta t}-\frac{\beta}{h^2} \end{bmatrix}.$$

If we denote $\mathbf{F}^k = B\mathbf{u}^k + \mathbf{b}^k$, then the approximate solution the differential equation at t^{k+1} is the solution of $A\mathbf{u}^{k+1} = \mathbf{F}^k$ which can be solved easily, for example, in Matlab, by $\mathbf{u} = A\backslash\mathbf{F}$, see *heat_im.m*.

7.5 Implicit method for the 1D one-way wave equations

The 1D one-way wave equation

$$u_t + cu_x = f(x,t), \quad 0 < x < L, \quad u(0,t) = g(t), \quad u(x,0) = u_0(x),$$

is a hyperbolic type equation. The FW-BW (forward in time and backward in space) finite difference scheme is

$$\frac{u_i^{k+1}-u_i^k}{\Delta t} + c\,\frac{u_i^k-u_{i-1}^k}{h} = f_i^k, \quad i = 0,1,\cdots, \qquad \text{if } c > 0,$$

whose truncation error is $O(\Delta t + h)$ (first order in time and first order in space). The time step restriction $\Delta t \le h/c$ is often acceptable (not like the explicit finite difference method for the heat equation where $\Delta t \le h^2/(2\beta)$).

The implicit finite difference scheme is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + c\,\frac{u_i^{k+1} - u_{i-1}^{k+1}}{h} = f_i^{k+1}, \quad i = 0, 1, \cdots, \qquad \text{if } c > 0,$$

We can write the scheme as a matrix-vector form $A\mathbf{u}^{k+1} = B\mathbf{u}^k + \mathbf{b}^k$,

$$A = \begin{bmatrix} d & 0 & 0 & \cdots & 0 \\ \alpha & d & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \alpha & d & 0 \\ 0 & \cdots & 0 & \alpha & d \end{bmatrix}, \quad B = I, \quad \mathbf{b}^k = \begin{bmatrix} \Delta t\, f_1^{k+1} - \alpha\, u(0, t^{k+1}) \\ \Delta t\, f_2^{k+1} \\ \vdots \\ 0 \\ \Delta t\, f_n^{k+1} \end{bmatrix},$$

where $d = 1 + \dfrac{c\,\Delta t}{h}$, $\alpha = -\dfrac{c\,\Delta t}{h}$. The system can be easily solved using a *forward substitution*

$$\begin{aligned}
u_1^{k+1} &= \frac{u_1^k + \Delta t\, f_1^{k+1} - \alpha\, u(0, t^{k+1})}{d} \\
u_2^{k+1} &= \frac{u_2^k + \Delta t\, f_2^{k+1} - \alpha\, u_1^{k+1}}{d} \\
\cdots &= \cdots \\
u_i^{k+1} &= \frac{u_i^k + \Delta t\, f_i^{k+1} - \alpha\, u_{i-1}^{k+1}}{d} \\
\cdots &= \cdots \\
u_n^{k+1} &= \frac{u_n^k + \Delta t\, f_n^{k+1} - \alpha\, u_{n-1}^{k+1}}{d}.
\end{aligned}$$