數值分析 MA-3021

Chapter 6. Numerical Ordinary Differential Equations

§6.1 Introduction

§6.2 Taylor-Series Method

§6.3 Euler's Method

§6.4 The Runge-Kutta Method

§6.5 Collocation Method

§6.6 Finite Difference Method

§6.7 Finite Element Method

§6.1 Introduction

Definition

A differential equation is a mathematical equation that relates some unknown function with its derivatives. A differential equation is called an ***ordinary differential equation*** (ODE) if it contains an unknown function of one independent variable and its derivatives. A differential equation is called a ***partial differential equation*** (PDE) if it contains unknown multi-variable functions and their partial derivatives.

Definition

The ***order*** of a differential equation is the order of the highest-order derivatives presented in the equation. A differential equation of order 1 is called first order, order 2 second order, etc.

§6.1 Introduction

Definition

A differential equation is a mathematical equation that relates some unknown function with its derivatives. A differential equation is called an ***ordinary differential equation*** (ODE) if it contains an unknown function of one independent variable and its derivatives. A differential equation is called a ***partial differential equation*** (PDE) if it contains unknown multi-variable functions and their partial derivatives.

Definition

The ***order*** of a differential equation is the order of the highest-order derivatives presented in the equation. A differential equation of order 1 is called first order, order 2 second order, etc.

§6.1 Introduction

Definition

A differential equation is a mathematical equation that relates some unknown function with its derivatives. A differential equation is called an ***ordinary differential equation*** (ODE) if it contains an unknown function of one independent variable and its derivatives. A differential equation is called a ***partial differential equation*** (PDE) if it contains unknown multi-variable functions and their partial derivatives.

Definition

The ***order*** of a differential equation is the order of the highest-order derivatives presented in the equation. A differential equation of order 1 is called first order, order 2 second order, etc.

§6.1 Introduction

Remark:

1. In general, an n-th order (scalar) ODE has the form
$$F(t, y, y', \cdots, y^{(n)}) = 0\,,$$
where $\dfrac{\partial F}{\partial y^{(n)}} \neq 0$. By the **implicit function theorem**,
$$y^{(n)} = \varphi\left(t, y, y', \cdots, y^{(n-1)}\right) \qquad (\star)$$
for some function φ (locally).
2. Assume that y satisfies $(\star)$. Let $\boldsymbol{x} = \left(y, y', y'', \cdots, y^{(n-1)}\right)^\top$. Then $\boldsymbol{x}$ satisfies
$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})\,,$$
where the vector-valued function $\boldsymbol{f}$ is given by
$$\boldsymbol{f}(t, \boldsymbol{x}) = \boldsymbol{f}(t, x_1, \cdots, x_n) = \begin{bmatrix} x_2 \\ \vdots \\ x_n \\ \varphi(t, \boldsymbol{x}) \end{bmatrix}.$$

§6.1 Introduction

Remark:

1. In general, an n-th order (scalar) ODE has the form
$$F(t, y, y', \cdots, y^{(n)}) = 0\,,$$
where $\dfrac{\partial F}{\partial y^{(n)}} \neq 0$. By the **implicit function theorem**,
$$y^{(n)} = \varphi\left(t, y, y', \cdots, y^{(n-1)}\right) \qquad (\star)$$
for some function φ (locally).
2. Assume that y satisfies $(\star)$. Let $\boldsymbol{x} = \left(y, y', y'', \cdots, y^{(n-1)}\right)^\top$. Then $\boldsymbol{x}$ satisfies
$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})\,,$$
where the vector-valued function $\boldsymbol{f}$ is given by
$$\boldsymbol{f}(t, \boldsymbol{x}) = \boldsymbol{f}(t, x_1, \cdots, x_n) = \begin{bmatrix} x_2 \\ \vdots \\ x_n \\ \varphi(t, \boldsymbol{x}) \end{bmatrix}.$$

§6.1 Introduction

Remark (cont'd):

3. It is also possible to consider n-th order **vector-valued** ODE. For example, let $\boldsymbol{r}(t)$ denote the position of a planet (of mass m) moving around the sun (of mass M) which locates at the origin. Then Newton's second law of motion implies that

$$m\boldsymbol{r}''(t) = -\frac{GMm}{\|\boldsymbol{r}\|^3}\boldsymbol{r}(t)\,.$$

In general, when considering this kind of equation, we assume that it can be written as

$$\boldsymbol{y}^{(n)} = \varphi\left(t, \boldsymbol{y}, \boldsymbol{y}', \cdots, \boldsymbol{y}^{(n-1)}\right).$$

Then the same procedure (by letting $\boldsymbol{x} = \left(\boldsymbol{y}, \boldsymbol{y}', \cdots, \boldsymbol{y}^{(n-1)}\right)^\top$ and find an equation that $\boldsymbol{x}$ satisfies) shows that the equation above reduces to an first order ODE

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})\,.$$

§6.1 Introduction

Remark (cont'd):

3. It is also possible to consider n-th order **vector-valued** ODE. For example, let $\boldsymbol{r}(t)$ denote the position of a planet (of mass m) moving around the sun (of mass M) which locates at the origin. Then Newton's second law of motion implies that

$$m\boldsymbol{r}''(t) = -\frac{GMm}{\|\boldsymbol{r}\|^3}\boldsymbol{r}(t)\,.$$

In general, when considering this kind of equation, we assume that it can be written as

$$\boldsymbol{y}^{(n)} = \varphi\left(t, \boldsymbol{y}, \boldsymbol{y}', \cdots, \boldsymbol{y}^{(n-1)}\right).$$

Then the same procedure (by letting $\boldsymbol{x} = \left(\boldsymbol{y}, \boldsymbol{y}', \cdots, \boldsymbol{y}^{(n-1)}\right)^\top$ and find an equation that $\boldsymbol{x}$ satisfies) shows that the equation above reduces to an first order ODE

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})\,.$$

§6.1 Introduction

Initial-value problem (IVP): find $\boldsymbol{x}(t)$ such that

$$\begin{cases} \boldsymbol{x}'(t) = \boldsymbol{f}\big(t, \boldsymbol{x}(t)\big), & \text{(ordinary differential equations)} \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0, & \text{(initial condition)} \end{cases}$$

where $\boldsymbol{f}(t, \boldsymbol{x})$, $t_0 \in \mathbb{R}^1$, $\boldsymbol{x}_0 \in \mathbb{R}^n$ are given.

Example

$$x'(t) = x(t)\tan(t+3), \qquad x(-3) = 1.$$

The analytic solution of this IVP is $x(t) = \sec(t+3)$. The solution is valid only for $-\frac{\pi}{2} < t+3 < \frac{\pi}{2}$.

Example

$$x'(t) = x, \qquad x(0) = 1.$$

Try $x(t) = ce^{rt} \Rightarrow cre^{rt} = ce^{rt} \Rightarrow r = 1$, $x = ce^{t}$ general solution

Use $x(0) = 1 \Rightarrow x = e^{t}$ particular solution

§6.1 Introduction

Existence and Uniqueness of Solution: Not all IVPs have a solution. Even if there exists a solution, the solution may not be unique.

Example

There is no solution to the initial value problem

$$\exp\left(x'(t)\right) = 0, \qquad x(0) = 0$$

even if complex-valued solutions are allowed.

Example

The initial value problem

$$x'(t) = 3x(t)^{\frac{2}{3}}, \qquad x(0) = 0$$

has **infinitely many** solutions. In fact, the function

$$x_c(t) = \begin{cases} 0 & \text{if } t \leqslant c \\ (t-c)^3 & \text{if } t > c \end{cases}$$

is a solution to the given ODE for all $c \geqslant 0$.

§6.1 Introduction

Theorem (Existence and Uniqueness of Solution)

Consider the initial value problem

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \,.$$

If $\boldsymbol{f}$ and the first partial derivatives of $\boldsymbol{f}$ with respect to all its variables, possibly except t, are continuous functions in some rectangular domain R that contains the point $(t_0, \boldsymbol{x}_0)$ in the interior, then the initial value problem has a unique solution $\varphi(t)$ in some interval $I = (t_0 - h, t_0 + h)$ for some positive number h.

Remark:

1. If $\boldsymbol{f}$ is k-times continuously differentiable, then $\boldsymbol{x}$ is $(k+1)$-times continuously differentiable.
2. The length of the time interval of existence "usually" is inverse proportional to the maximum of $\|\boldsymbol{f}\| + \|\boldsymbol{f}_{\boldsymbol{y}}\|$.

§6.1 Introduction

Theorem (Fundamental Theorem of ODE)

Consider the initial value problem

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 .$$

If $\boldsymbol{f}$ and the first partial derivatives of $\boldsymbol{f}$ with respect to all its variables, possibly except t, are continuous functions in some rectangular domain R that contains the point $(t_0, \boldsymbol{x}_0)$ in the interior, then the initial value problem has a unique solution $\varphi(t)$ in some interval $I = (t_0 - h, t_0 + h)$ for some positive number h.

Remark:

1. If $\boldsymbol{f}$ is k-times continuously differentiable, then $\boldsymbol{x}$ is $(k+1)$-times continuously differentiable.
2. The length of the time interval of existence "usually" is inverse proportional to the maximum of $\|\boldsymbol{f}\| + \|\boldsymbol{f}_{\boldsymbol{x}}\|$.

§6.1 Introduction

Theorem (Fundamental Theorem of ODE)

Consider the initial value problem

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 .$$

If $\boldsymbol{f}$ and the first partial derivatives of $\boldsymbol{f}$ with respect to all its variables, possibly except t, are continuous functions in some rectangular domain R that contains the point $(t_0, \boldsymbol{x}_0)$ in the interior, then the initial value problem has a unique solution $\varphi(t)$ in some interval $I = (t_0 - h, t_0 + h)$ for some positive number h.

Remark:

1. If $\boldsymbol{f}$ is k-times continuously differentiable, then $\boldsymbol{x}$ is $(k+1)$-times continuously differentiable.
2. The length of the time interval of existence "usually" is inverse proportional to the maximum of $\|\boldsymbol{f}\| + \|\boldsymbol{f}_{\boldsymbol{x}}\|$.

§6.1 Introduction

Remark:

3. Suppose that for a constant $k \in (0,1)$ the "cube" Q centered at $(t_0, \boldsymbol{x}_0)$ with width $2k$ is a subset of the rectangular domain R, and for some $M \geqslant 1$, $\left|\boldsymbol{f}(t,y)\right| + \left|\boldsymbol{f}_{\boldsymbol{x}}(t,\boldsymbol{x})\right| \leqslant M$ for all $(t,\boldsymbol{x}) \in Q$. Then the solution $\boldsymbol{x}$ to the initial value problem

$$\boldsymbol{x}' = \boldsymbol{f}(t,\boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0\,.$$

exists at least in the interval $\left[t_0 - \frac{k}{M}, t_0 + \frac{k}{M}\right]$.

§6.1 Introduction

Definition (Informal definition)

A numerical method of solving the ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ with $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ is an **iterative scheme** which, when the ***step size*** $h > 0$ is given, generates a unique sequence of vectors $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ (for some N which in general depends on h) such that some function φ which interpolates the data $(t_0, \boldsymbol{x}_0)$, $(t_1, \boldsymbol{x}_1)$, $\cdots$, $(t_n, \boldsymbol{x}_N)$ where, $t_n = t_0 + nh$, **resembles** the solution to $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ with initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ in the time interval $[t_0, t_N]$. The function φ is called the ***numerical solution*** generated by this numerical method with step size h.

§6.1 Introduction

Definition (Informal definition (cont'd))

A numerical method of solving the ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ is called a *k*-***step method*** if it requires $\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_{n+k-1}$ to determine $\boldsymbol{x}_{n+k}$ for all $n \in \{0, \cdots, N-k\}$.

A numerical method of solving the ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ is said to be ***explicit*** if it does not require "nonlinear procedures" to obtain some $\boldsymbol{x}_n$'s, and is said to be ***implicit*** if it is not explicit.

Remark: A one-step explicit method is often (but not always) given in the form $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\,\Phi(t_n, \boldsymbol{x}_n)$ for some function Φ, while a *k*-step explicit method is often (but not always) given in the form

$$\begin{aligned}\boldsymbol{x}_{n+1} = {} & \alpha_1 \boldsymbol{x}_n + \alpha_2 \boldsymbol{x}_{n-1} + \cdots + \alpha_k \boldsymbol{x}_{n-k+1} \\ & + h\Big[\beta_1 \boldsymbol{f}(t_n, \boldsymbol{x}_n) + \cdots + \beta_k \boldsymbol{f}(t_{n-k+1}, \boldsymbol{x}_{n-k+1})\Big].\end{aligned}$$

§6.1 Introduction

Example

The forward Euler method of solving the ordinary differential equations $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ is an explicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_{n-1}, \boldsymbol{x}_{n-1}) \qquad \forall\, n \in \{1, 2, \cdots, N\},$$

while the backward Euler method is an implicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_n, \boldsymbol{x}_n) \qquad \forall\, n \in \{1, 2, \cdots, N\}\,.$$

Example

The Runge-Kutta method is an explicit one-step method given by

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\left(\frac{\boldsymbol{k}_{n1} + 2\boldsymbol{k}_{n2} + 2\boldsymbol{k}_{n3} + \boldsymbol{k}_{n4}}{6}\right),$$

where

$$\boldsymbol{k}_{n1} = \boldsymbol{f}(t_n, \boldsymbol{x}_n)\,, \qquad \boldsymbol{k}_{n2} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n1})\,,$$
$$\boldsymbol{k}_{n3} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n2})\,, \quad \boldsymbol{k}_{n4} = \boldsymbol{f}(t_n + h, \boldsymbol{x}_n + h\boldsymbol{k}_{n3})\,.$$

§6.1 Introduction

Example

The forward Euler method of solving the ordinary differential equations $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ is an explicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_{n-1}, \boldsymbol{x}_{n-1}) \qquad \forall\, n \in \{1, 2, \cdots, N\},$$

while the backward Euler method is an implicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_n, \boldsymbol{x}_n) \qquad \forall\, n \in \{1, 2, \cdots, N\}\,.$$

Example

The Runge-Kutta method is an explicit one-step method given by

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\left(\frac{\boldsymbol{k}_{n1} + 2\boldsymbol{k}_{n2} + 2\boldsymbol{k}_{n3} + \boldsymbol{k}_{n4}}{6}\right),$$

where

$$\boldsymbol{k}_{n1} = \boldsymbol{f}(t_n, \boldsymbol{x}_n)\,, \qquad \boldsymbol{k}_{n2} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n1})\,,$$
$$\boldsymbol{k}_{n3} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n2})\,, \quad \boldsymbol{k}_{n4} = \boldsymbol{f}(t_n + h, \boldsymbol{x}_n + h\boldsymbol{k}_{n3})\,.$$

§6.1 Introduction

Example

The forward Euler method of solving the ordinary differential equations $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ is an explicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_{n-1}, \boldsymbol{x}_{n-1}) \qquad \forall\, n \in \{1, 2, \cdots, N\},$$

while the backward Euler method is an implicit one-step method given by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + h\boldsymbol{f}(t_n, \boldsymbol{x}_n) \qquad \forall\, n \in \{1, 2, \cdots, N\}\,.$$

Example

The Runge-Kutta method is an explicit one-step method given by

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\left(\frac{\boldsymbol{k}_{n1} + 2\boldsymbol{k}_{n2} + 2\boldsymbol{k}_{n3} + \boldsymbol{k}_{n4}}{6}\right),$$

where

$$\boldsymbol{k}_{n1} = \boldsymbol{f}(t_n, \boldsymbol{x}_n)\,, \qquad \boldsymbol{k}_{n2} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n1})\,,$$
$$\boldsymbol{k}_{n3} = \boldsymbol{f}(t_n + \frac{1}{2}h, \boldsymbol{x}_n + \frac{1}{2}h\boldsymbol{k}_{n2})\,, \quad \boldsymbol{k}_{n4} = \boldsymbol{f}(t_n + h, \boldsymbol{x}_n + h\boldsymbol{k}_{n3})\,.$$

§6.1 Introduction

There are three fundamental sources of error of a numerical solution:

1. The iterative scheme used to produce the sequence $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ is an approximate one. In other words, at each step the numerical method does not produce the correct value of the solution at the next time step. This relates to the local/global truncation error.
2. The input data used in the iterative scheme are only approximations to the actual values of the solution at each t_k. For example, one should use $\boldsymbol{x}(t_k)$ to generate $\boldsymbol{x}_{k+1}$ but we are forced to start with $\boldsymbol{x}_k$. This relates to the global truncation error.
3. The precision of calculations of the computer is finite. In other words, at each step only a finite number of digits can be retained. This relates to the round-off error (or machine error).

§6.1 Introduction

There are three fundamental sources of error of a numerical solution:

1. The iterative scheme used to produce the sequence $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ is an approximate one. In other words, at each step the numerical method does not produce the correct value of the solution at the next time step. This relates to the local/global truncation error.
2. The input data used in the iterative scheme are only approximations to the actual values of the solution at each t_k. For example, one should use $\boldsymbol{x}(t_k)$ to generate $\boldsymbol{x}_{k+1}$ but we are forced to start with $\boldsymbol{x}_k$. This relates to the global truncation error.
3. The precision of calculations of the computer is finite. In other words, at each step only a finite number of digits can be retained. This relates to the round-off error (or machine error).

§6.1 Introduction

There are three fundamental sources of error of a numerical solution:

1. The iterative scheme used to produce the sequence $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ is an approximate one. In other words, at each step the numerical method does not produce the correct value of the solution at the next time step. This relates to the local/global truncation error.
2. The input data used in the iterative scheme are only approximations to the actual values of the solution at each t_k. For example, one should use $\boldsymbol{x}(t_k)$ to generate $\boldsymbol{x}_{k+1}$ but we are forced to start with $\boldsymbol{x}_k$. This relates to the global truncation error.
3. The precision of calculations of the computer is finite. In other words, at each step only a finite number of digits can be retained. This relates to the round-off error (or machine error).

§6.1 Introduction

Definition

Let φ be a numerical solution obtained by a specific numerical method (with step size $h > 0$ fixed) of solving ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ with initial data $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. At each time step t_n,

1. the ***global truncation error*** (associated with this numerical method) is the number $\boldsymbol{E}_n(h) = \boldsymbol{x}(t_n) - \varphi(t_n)$;
2. the ***local truncation error*** (associated with this numerical method) is the number $\tau_n(h) = \boldsymbol{x}(t_{n+1}) - \boldsymbol{x}_{n+1}$, where $\boldsymbol{x}(\cdot)$ is the exact solution and $\boldsymbol{x}_{n+1}$ is obtained according to the iterative scheme with $\boldsymbol{x}_j = \boldsymbol{x}(t_j)$ for all $j \in \{0, 1, \cdots, n\}$.
3. the ***round-off error*** or ***machine error*** (associated with this numerical method) is the number $\boldsymbol{R}_n = \varphi(t_n) - \boldsymbol{X}_n$, where $\boldsymbol{X}_n$ is the actual value computed from the numerical method.

§6.1 Introduction

Definition

Let φ be a numerical solution obtained by a specific numerical method (with step size $h > 0$ fixed) of solving ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ with initial data $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. At each time step t_n,

1. the ***global truncation error*** (associated with this numerical method) is the number $\boldsymbol{E}_n(h) = \boldsymbol{x}(t_n) - \varphi(t_n)$;
2. the ***local truncation error*** (associated with this numerical method) is the number $\boldsymbol{\tau}_n(h) = \boldsymbol{x}(t_{n+1}) - \boldsymbol{x}_{n+1}$, where $\boldsymbol{x}(\cdot)$ is the exact solution and $\boldsymbol{x}_{n+1}$ is obtained according to the iterative scheme with $\boldsymbol{x}_j = \boldsymbol{x}(t_j)$ for all $j \in \{0, 1, \cdots, n\}$.
3. the ***round-off error*** or ***machine error*** (associated with this numerical method) is the number $\boldsymbol{R}_n = \varphi(t_n) - \boldsymbol{X}_n$, where $\boldsymbol{X}_n$ is the actual value computed from the numerical method.

§6.1 Introduction

Definition

Let φ be a numerical solution obtained by a specific numerical method (with step size $h > 0$ fixed) of solving ODE $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$ with initial data $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. At each time step t_n,

1. the ***global truncation error*** (associated with this numerical method) is the number $\boldsymbol{E}_n(h) = \boldsymbol{x}(t_n) - \varphi(t_n)$;
2. the ***local truncation error*** (associated with this numerical method) is the number $\boldsymbol{\tau}_n(h) = \boldsymbol{x}(t_{n+1}) - \boldsymbol{x}_{n+1}$, where $\boldsymbol{x}(\cdot)$ is the exact solution and $\boldsymbol{x}_{n+1}$ is obtained according to the iterative scheme with $\boldsymbol{x}_j = \boldsymbol{x}(t_j)$ for all $j \in \{0, 1, \cdots, n\}$.
3. the ***round-off error*** or ***machine error*** (associated with this numerical method) is the number $\boldsymbol{R}_n = \varphi(t_n) - \boldsymbol{X}_n$, where $\boldsymbol{X}_n$ is the actual value computed from the numerical method.

§6.1 Introduction

In other words, the local truncation error measures the accuracy of the numerical method for each time step, while the global truncation error measure the errors accumulated from the beginning of this iterative scheme.

Definition

A numerical method is said to be ***consistent*** if

$$\lim_{h\to 0} \max_{0\leqslant n\leqslant N-1} \left|\frac{\tau_n(h)}{h}\right| = 0\,,$$

where $\tau_n(h)$ is the local truncation error associated with the numerical method with step size h.

A numerical method is said to **have order** k if for any sufficiently smooth solution of the initial value problem, the local truncation error is $\mathcal{O}(h^{k+1})$.

§6.1 Introduction

In other words, the local truncation error measures the accuracy of the numerical method for each time step, while the global truncation error measure the errors accumulated from the beginning of this iterative scheme.

Definition

A numerical method is said to be ***consistent*** if

$$\lim_{h\to 0}\ \max_{0\leqslant n\leqslant N-1}\left|\frac{\tau_n(h)}{h}\right| = 0\,,$$

where $\tau_n(h)$ is the local truncation error associated with the numerical method with step size h.

A numerical method is said to **have order** k if for any sufficiently smooth solution of the initial value problem, the local truncation error is $\mathcal{O}(h^{k+1})$.

§6.1 Introduction

In other words, the local truncation error measures the accuracy of the numerical method for each time step, while the global truncation error measure the errors accumulated from the beginning of this iterative scheme.

Definition

A numerical method is said to be ***consistent*** if

$$\lim_{h\to 0}\ \max_{0\leqslant n\leqslant N-1}\left|\frac{\tau_n(h)}{h}\right| = 0\,,$$

where $\tau_n(h)$ is the local truncation error associated with the numerical method with step size h.

A numerical method is said to **have order** k if for any sufficiently smooth solution of the initial value problem, the local truncation error is $\mathcal{O}(h^{k+1})$.

§6.1 Introduction

For a second order ODE

$$y'' + p(t)y' + q(t)y = g(t),$$

instead of imposing the initial condition $y(t_0) = y_0$ and $y'(t_0) = y_1$ sometimes the boundary condition $y(\alpha) = y_0$ and $y(\beta) = y_1$ are imposed and consider the two-point boundary value problem (BVP)

$$y'' + p(x)y' + q(x)y = g(x) \quad \forall\, x \in (\alpha, \beta),\ y(\alpha) = y_0,\ y(\beta) = y_1.$$

Let $z(x) = y(x) - \frac{x-\alpha}{\beta-\alpha}y_1 - \frac{x-\beta}{\alpha-\beta}y_0$. Then z satisfies

$$z'' + p(x)z' + q(x)z = G(x) \quad \forall\, x \in (\alpha, \beta),\ z(\alpha) = z(\beta) = 0, \quad (\star)$$

where $G(x) = g(x) - p(x)\frac{y_0 - y_1}{\alpha - \beta} - q(x)\left(\frac{x-\alpha}{\beta-\alpha}y_1 + \frac{x-\beta}{\alpha-\beta}y_0\right)$. $(\star)$ is called a **homogeneous** two-point boundary value problem.

§6.1 Introduction

For a second order ODE

$$y'' + p(t)y' + q(t)y = g(t),$$

instead of imposing the initial condition $y(t_0) = y_0$ and $y'(t_0) = y_1$ sometimes the boundary condition $y(\alpha) = y_0$ and $y(\beta) = y_1$ are imposed and consider the two-point boundary value problem (BVP)

$$y'' + p(x)y' + q(x)y = g(x) \quad \forall\, x \in (\alpha, \beta),\ y(\alpha) = y_0,\ y(\beta) = y_1.$$

Let $z(x) = y(x) - \frac{x-\alpha}{\beta-\alpha}y_1 - \frac{x-\beta}{\alpha-\beta}y_0$. Then z satisfies

$$z'' + p(x)z' + q(x)z = G(x) \quad \forall\, x \in (\alpha, \beta),\ z(\alpha) = z(\beta) = 0, \quad (\star)$$

where $G(x) = g(x) - p(x)\frac{y_0 - y_1}{\alpha - \beta} - q(x)\big(\frac{x-\alpha}{\beta-\alpha}y_1 + \frac{x-\beta}{\alpha-\beta}y_0\big)$. $(\star)$ is called a **homogeneous** two-point boundary value problem.

§6.1 Introduction

Remark: Even though the IVP

$$y'' + p(t)y' + q(t)y = g(t), \qquad y(t_0) = y_0\,, \quad y'(t_0) = y_1$$

looks quite similar to the boundary value problem, they actually differ in some very important ways. For example, if p, q, g are continuous, the IVP above always has a unique solution, while the BVP might have no solution or infinitely many solutions:

1. $y'' + y = 0$ with boundary condition $y(0) = y(\pi) = 0$ has infinite many solutions $y_c(x) = c \sin x$.
2. $y'' + y = \sin x$ with boundary condition $y(0) = y(\pi) = 0$ has no solution.

On the other hand, some BVPs has a unique solution. For example,

$$y'' + 2y = 0, \quad \forall\, x \in (0, \pi),\ y(0) = 1,\ y(\pi) = 0$$

has a unique solution $y(x) = \cos\sqrt{2}x - \cot\sqrt{2}\pi \sin\sqrt{2}x$.

§6.1 Introduction

Remark: Even though the IVP

$$y'' + p(t)y' + q(t)y = g(t), \qquad y(t_0) = y_0\,, \quad y'(t_0) = y_1$$

looks quite similar to the boundary value problem, they actually differ in some very important ways. For example, if p, q, g are continuous, the IVP above always has a unique solution, while the BVP might have no solution or infinitely many solutions:

1. $y'' + y = 0$ with boundary condition $y(0) = y(\pi) = 0$ has infinite many solutions $y_c(x) = c\sin x$.
2. $y'' + y = \sin x$ with boundary condition $y(0) = y(\pi) = 0$ has no solution.

On the other hand, some BVPs has a unique solution. For example,

$$y'' + 2y = 0, \quad \forall\, x \in (0, \pi),\ y(0) = 1,\ y(\pi) = 0$$

has a unique solution $y(x) = \cos\sqrt{2}x - \cot\sqrt{2}\pi \sin\sqrt{2}x$.

§6.1 Introduction

Theorem

Let α, β be real numbers and $\alpha < \beta$. Suppose that the function $f = f(t, y, p)$ is continuous on the set

$$D = \{(x, y, p) \,|\, x \in [\alpha, \beta], y, p \in \mathbb{R}\}$$

and the partial derivatives f_y and f_p are also continuous on D. If

1. *$f_y(t, y, p) > 0$ for all $(t, y, p) \in D$, and*
2. *there exists a constant $M > 0$ such that*

$$|f_p(t, y, p)| \leqslant M \quad \forall\, (t, y, p) \in D\,,$$

then the boundary value problem

$$y'' = f(t, y, y') \quad \forall\, x \in (\alpha, \beta),\ y(\alpha) = y(\beta) = 0$$

has a unique solution.

§6.1 Introduction

Theorem

Let α, β be real numbers and $\alpha < \beta$. Suppose that $p : [\alpha, \beta] \to \mathbb{R}$ is continuously differentiable, and $q : [\alpha, \beta] \to \mathbb{R}$ is continuous. Then

$$y'' + p(x)y' + q(x)y = g(x) \quad \forall\, x \in (\alpha, \beta),\ y(\alpha) = y(\beta) = 0$$

has a solution if and only if $g : [\alpha, \beta] \to \mathbb{R}$ is integrable and

$$\int_\alpha^\beta g(x)\varphi(x)\, dx = 0$$

for all function φ satisfying

$$\varphi'' - p(x)\varphi' + \big(q(x) - p'(x)\big)\varphi = 0 \quad \forall\, x \in (\alpha, \beta),\ \varphi(\alpha) = \varphi(\beta) = 0.$$

The solution is unique if the ODE $y'' + p(x)y' + q(x)y = 0$ with $y(\alpha) = y(\beta) = 0$ has only trivial solution $y \equiv 0$.

§6.2 Taylor-Series Method

Consider the ODE $x' = f(t, x)$. Then the chain rule can be used to compute $x^{(k)}(t)$:

$$
\begin{aligned}
x''(t) &= f_t(t,x) + f_x(t,x)x', \\
x'''(t) &= f_{tt}(t,x) + 2f_{tx}(t,x)x' + f_{xx}(t,x)(x')^2 + f_x(t,x)x'', \\
x^{(4)}(t) &= f_{ttt}(t,x) + 3f_{ttx}(t,x)x' + 3f_{txx}(t,x)(x')^2 + f_{xxx}(t,x)(x')^3 \\
&\quad + f_{tx}(t,x)x'' + f_{xx}(t,x)x'x'' + f_x(t,x)x''' \\
&\ \ \vdots
\end{aligned}
$$

thus Taylor's Theorem implies that

$$
\begin{aligned}
x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\big[f_t(t,x) + f_x(t,x)f(t,x)\big] \\
+ \frac{h^3}{3}\Big[f_{tt}(t,x) + 2f_{tx}(t,x)f(t,x) + f_{xx}(t,x)f(t,x)^2 \\
+ f_x(t,x)\big(f_t(t,x) + f_x(t,x)f(t,x)\big)\Big] + \cdots
\end{aligned}
$$

§6.2 Taylor-Series Method

Consider the ODE $x' = f(t, x)$. Then the chain rule can be used to compute $x^{(k)}(t)$:

$$\begin{aligned}
x''(t) &= f_t(t,x) + f_x(t,x)x', \\
x'''(t) &= f_{tt}(t,x) + 2f_{tx}(t,x)x' + f_{xx}(t,x)(x')^2 + f_x(t,x)x'', \\
x^{(4)}(t) &= f_{ttt}(t,x) + 3f_{ttx}(t,x)x' + 3f_{txx}(t,x)(x')^2 + f_{xxx}(t,x)(x')^3 \\
&\quad + f_{tx}(t,x)x'' + f_{xx}(t,x)x'x'' + f_x(t,x)x''' \\
&\vdots
\end{aligned}$$

thus Taylor's Theorem implies that

$$\begin{aligned}
x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\left[f_t(t,x) + f_x(t,x)f(t,x)\right] \\
+ \frac{h^3}{3}\Big[f_{tt}(t,x) + 2f_{tx}(t,x)f(t,x) + f_{xx}(t,x)f(t,x)^2 \\
+ f_x(t,x)\big(f_t(t,x) + f_x(t,x)f(t,x)\big)\Big] + \cdots
\end{aligned}$$

§6.2 Taylor-Series Method

Depending on the smoothness of f and how many terms one wants to keep (deleting high order terms), we can obtain various Taylor's Method:

1. Taylor's method of order one:
$$x_{n+1} = x_n + hf(t_n, x_n).$$
2. Taylor's method of order two:
$$x_{n+1} = x_n + hf(t_n, x_n) + \frac{h^2}{2}\left[f_t(t_n, x_n) + f_x(t_n, x_n)f(t_n, x_n)\right].$$
3. Taylor's method of order three:
$$\begin{aligned} x_{n+1} &= x_n + hf(t_n, x_n) + \frac{h^2}{2}\left[f_t(t_n, x_n) + f_x(t_n, x_n)f(t_n, x_n)\right] \\ &+ \frac{h^3}{3}\Big[f_{tt}(t_n, x_n) + 2f_{tx}(t_n, x_n)f(t_n, x_n) + f_{xx}(t_n, x_n)f(t_n, x_n)^2 \\ &+ f_x(t_n, x_n)\big(f_t(t_n, x_n) + f_x(t_n, x_n)f(t_n, x_n)\big)\Big]. \end{aligned}$$

§6.2 Taylor-Series Method

Example

We use a concrete example to illustrate the method. Consider the following IVP

$$x'(t) = \cos t - \sin x + t^2, \quad x(-1) = 3.$$

By the Fundamental Theorem of ODE, the solution x is infinitely many times differentiable. By the Taylor series for x, we have

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t) + \mathcal{O}(h^5).$$

Since $x'(t) = \cos t - \sin x + t^2$,

$$\begin{cases} x''(t) = -\sin t - (\cos x)x' + 2t, \\ x'''(t) = -\cos t + \sin x\,(x')^2 - (\cos x)x'' + 2, \\ x^{(4)}(t) = \sin t + (\cos x)(x'')^3 + 3(\sin x)x'x'' - (\cos x)x'''. \end{cases}$$

thus by truncating at h^4, the **local truncation error** for obtaining $x(t+h)$ is $\mathcal{O}(h^5)$. Such a method is of order 4.

§6.2 Taylor-Series Method

Example (cont'd)

Starting $t = -1$ with $h = 0.01$, we can compute the solution in $[-1, 1]$ with 200 steps:

input $M \leftarrow 200,\ h \leftarrow 0.01,\ t \leftarrow -1,\ x \leftarrow 3$

output $0, t, x$

for $k = 1$ **to** M **do**

$$
\begin{aligned}
x' &\leftarrow \cos t - \sin x + t^2 \\
x'' &\leftarrow -\sin t - (\cos x)x' + 2t \\
x''' &\leftarrow -\cos t + \sin x(x')^2 - (\cos x)x'' + 2 \\
x^{(4)} &\leftarrow \sin t + (\cos x)(x')^3 + 3(\sin x)x'x'' - (\cos x)x''' \\
x &\leftarrow x + h(x' + \frac{h}{2}(x'' + \frac{h}{3}(x''' + \frac{h}{4}x^{(4)}))) \\
t &\leftarrow t + h
\end{aligned}
$$

output k, t, x

end do

§6.2 Taylor-Series Method

Example (cont'd)

1. The local truncation error can be estimated by looking at

$$E_n = \frac{1}{(n+1)!} h^{n+1} x^{(n+1)}(t+\theta h) \quad \textit{for some } \theta \in (0,1).$$

Hence

$$E_4 = \frac{1}{5!} h^5 x^{(5)}(t+\theta h) \quad \theta \in (0,1).$$

2. Replace $x^{(5)}(t+\theta h)$ by a simple finite-difference approximation

$$E_4 \approx \frac{1}{5!} h^5 \left(\frac{x^{(4)}(t+h) - x^{(4)}(t)}{h} \right) = \frac{h^4}{120} \left(x^{(4)}(t+h) - x^{(4)}(t) \right).$$

3. Suppose that the local truncation error (LTE) is $\mathcal{O}(h^{n+1})$. The accumulation of all many LTEs gives rise the global truncation error (GTE):

$$GTE \approx \frac{T - t_0}{h} \mathcal{O}(h^{n+1}) = \mathcal{O}(h^n).$$

And we say the numerical method is of $\mathcal{O}(h^n)$.

§6.2 Taylor-Series Method

Taylor-Series method for systems:

For each $1 \leqslant i \leqslant n$, by the Taylor Theorem

$$x_i(t+h) = x_i(t) + hx_i'(t) + \frac{h^2}{2}x_i''(t) + \cdots + \frac{h^n}{n!}x_i^{(n)}(t) + \mathcal{O}(h^{n+1})$$

which implies the vector form

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{x}'(t) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1})$$

Using $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$, we find that

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1}).$$

Since $\boldsymbol{x}''(t) = \boldsymbol{f}_t(t, \boldsymbol{x}) + \sum_{k=1}^{n} \frac{\partial \boldsymbol{f}}{\partial x_k}(t, \boldsymbol{x})x_k'(t) \equiv \big[\boldsymbol{f}_t + (\boldsymbol{f} \cdot \nabla)\boldsymbol{f}\big](t, \boldsymbol{x})$, we have

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\big[\boldsymbol{f}_t + (\boldsymbol{f} \cdot \nabla)\boldsymbol{f}\big](t, \boldsymbol{x}(t)) + \mathcal{O}(h^3)$$

§6.2 Taylor-Series Method

Taylor-Series method for systems:

For each $1 \leqslant i \leqslant n$, by the Taylor Theorem

$$x_i(t+h) = x_i(t) + hx_i'(t) + \frac{h^2}{2}x_i''(t) + \cdots + \frac{h^n}{n!}x_i^{(n)}(t) + \mathcal{O}(h^{n+1})$$

which implies the vector form

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{x}'(t) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1})$$

Using $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$, we find that

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1}).$$

Since $\boldsymbol{x}''(t) = \boldsymbol{f}_t(t, \boldsymbol{x}) + \sum_{k=1}^{n} \frac{\partial \boldsymbol{f}}{\partial x_k}(t, \boldsymbol{x})x_k'(t) \equiv \big[\boldsymbol{f}_t + (\boldsymbol{f} \cdot \nabla)\boldsymbol{f}\big](t, \boldsymbol{x})$, we have

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\big[\boldsymbol{f}_t + (\boldsymbol{f} \cdot \nabla)\boldsymbol{f}\big](t, \boldsymbol{x}(t)) + \mathcal{O}(h^3)$$

§6.2 Taylor-Series Method

Taylor-Series method for systems:

For each $1 \leqslant i \leqslant n$, by the Taylor Theorem

$$x_i(t+h) = x_i(t) + hx_i'(t) + \frac{h^2}{2}x_i''(t) + \cdots + \frac{h^n}{n!}x_i^{(n)}(t) + \mathcal{O}(h^{n+1})$$

which implies the vector form

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{x}'(t) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1})$$

Using $\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x})$, we find that

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\boldsymbol{x}''(t) + \cdots + \frac{h^n}{n!}\boldsymbol{x}^{(n)}(t) + \mathcal{O}(h^{n+1}).$$

Since $\boldsymbol{x}''(t) = \boldsymbol{f}_t(t, \boldsymbol{x}) + \sum_{k=1}^{n} \frac{\partial \boldsymbol{f}}{\partial x_k}(t, \boldsymbol{x})x_k'(t) \equiv \big[\boldsymbol{f}_t + (\boldsymbol{f}\cdot\nabla)\boldsymbol{f}\big](t, \boldsymbol{x})$, we have

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + h\boldsymbol{f}(t, \boldsymbol{x}(t)) + \frac{h^2}{2}\big[\boldsymbol{f}_t + (\boldsymbol{f}\cdot\nabla)\boldsymbol{f}\big](t, \boldsymbol{x}(t)) + \mathcal{O}(h^3)$$

§6.2 Taylor-Series Method

Disadvantages:

- The method depends on repeated differentiation of the differential equation, unless we intend to use only the method of order 1. For high order methods, $\boldsymbol{f}(t, x)$ must have partial derivatives of sufficient high order in the region where are solving the problem. Such an assumption is not necessary for the existence of a solution.
- The various derivatives formula need to be programmed.

Advantages:

- Conceptually simple.
- Potential for high precision. For example, if we get 20 derivatives of $x(t)$, then the method is order 20 (that is, terms up to and including the one involving h^{20}).

§6.2 Taylor-Series Method

Disadvantages:

- The method depends on repeated differentiation of the differential equation, unless we intend to use only the method of order 1. For high order methods, $\boldsymbol{f}(t, x)$ must have partial derivatives of sufficient high order in the region where are solving the problem. Such an assumption is not necessary for the existence of a solution.
- The various derivatives formula need to be programmed.

Advantages:

- Conceptually simple.
- Potential for high precision. For example, if we get 20 derivatives of $x(t)$, then the method is order 20 (that is, terms up to and including the one involving h^{20}).

§6.2 Taylor-Series Method

Disadvantages:

- The method depends on repeated differentiation of the differential equation, unless we intend to use only the method of order 1. For high order methods, $\boldsymbol{f}(t, x)$ must have partial derivatives of sufficient high order in the region where are solving the problem. Such an assumption is not necessary for the existence of a solution.
- The various derivatives formula need to be programmed.

Advantages:

- Conceptually simple.
- Potential for high precision. For example, if we get 20 derivatives of $x(t)$, then the method is order 20 (that is, terms up to and including the one involving h^{20}).

§6.2 Taylor-Series Method

Disadvantages:

- The method depends on repeated differentiation of the differential equation, unless we intend to use only the method of order 1. For high order methods, $\boldsymbol{f}(t, x)$ must have partial derivatives of sufficient high order in the region where are solving the problem. Such an assumption is not necessary for the existence of a solution.
- The various derivatives formula need to be programmed.

Advantages:

- Conceptually simple.
- Potential for high precision. For example, if we get 20 derivatives of $x(t)$, then the method is order 20 (that is, terms up to and including the one involving h^{20}).

§6.3 Euler's Method

1. If $n = 1$, the Taylor series method reduces to (forward/explicit) Euler's method:
$$x_{n+1} = x_n + hf(t_n, x_n)\,.$$
2. Advantage of the method is not to require any differentiation of f.
3. Disadvantage of the method is that the necessity of taking small value for h to gain acceptable precision.
4. One can also consider the backward/implicit Euler's method:
$$x_{n+1} = x_n + hf(t_n + h, x_{n+1})\,.$$
To obtain x_{n+1}, it is required to solve a nonlinear equation.

§6.4 The Runge-Kutta Method

The concept of the Runge-Kutta method is to provide a general procedure of deriving higher order methods, which does **NOT** involve the derivatives of $\boldsymbol{f}$, of solving the IVP

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0.$$

To see that this is possible, we note that Taylor's method of order 2 provides that

$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\left[f_t(t,x) + f_x(t,x)f(t,x)\right] + \mathcal{O}(h^3).$$

On the other hand, Taylor's Theorem implies that

$$f\big(t+h, x+hf(t,x)\big) = f(t,x) + f_t(t,x)h + f_x(t,x)hf(t,x) + \mathcal{O}(h^2);$$

thus

$$x(t+h) = x(t) + hf(t,x) + \frac{h}{2}\left[f\big(t+h, x+hf(t,x)\big) - f(t,x)\right] + \mathcal{O}(h^3)$$

which provides "a" second order Runge-Kutta method

$$x_{n+1} = x_n + hf(t_n, x_n) + \frac{h}{2}\left[f\big(t_{n+1}, x_n + hf(t_n, x_n)\big) - f(t_n, x_n)\right].$$

§6.4 The Runge-Kutta Method

The concept of the Runge-Kutta method is to provide a general procedure of deriving higher order methods, which does **NOT** involve the derivatives of $\boldsymbol{f}$, of solving the IVP

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0.$$

To see that this is possible, we note that Taylor's method of order 2 provides that

$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\left[f_t(t,x) + f_x(t,x)f(t,x)\right] + \mathcal{O}(h^3).$$

On the other hand, Taylor's Theorem implies that

$$f\big(t+h, x+hf(t,x)\big) = f(t,x) + f_t(t,x)h + f_x(t,x)hf(t,x) + \mathcal{O}(h^2);$$

thus

$$x(t+h) = x(t) + hf(t,x) + \frac{h}{2}\left[f\big(t+h, x+hf(t,x)\big) - f(t,x)\right] + \mathcal{O}(h^3)$$

which provides "a" second order Runge-Kutta method

$$x_{n+1} = x_n + hf(t_n, x_n) + \frac{h}{2}\left[f\big(t_{n+1}, x_n + hf(t_n, x_n)\big) - f(t_n, x_n)\right].$$

§6.4 The Runge-Kutta Method

The concept of the Runge-Kutta method is to provide a general procedure of deriving higher order methods, which does **NOT** involve the derivatives of $\boldsymbol{f}$, of solving the IVP

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0.$$

To see that this is possible, we note that Taylor's method of order 2 provides that

$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\left[f_t(t,x) + f_x(t,x)f(t,x)\right] + \mathcal{O}(h^3).$$

On the other hand, Taylor's Theorem implies that

$$f\big(t+h, x+hf(t,x)\big) = f(t,x) + f_t(t,x)h + f_x(t,x)hf(t,x) + \mathcal{O}(h^2);$$

thus

$$x(t+h) = x(t) + hf(t,x) + \frac{h}{2}\left[f\big(t+h, x+hf(t,x)\big) - f(t,x)\right] + \mathcal{O}(h^3)$$

which provides "a" second order Runge-Kutta method

$$x_{n+1} = x_n + hf(t_n, x_n) + \frac{h}{2}\left[f\big(t_{n+1}, x_n + hf(t_n, x_n)\big) - f(t_n, x_n)\right].$$

§6.4 The Runge-Kutta Method

The concept of the Runge-Kutta method is to provide a general procedure of deriving higher order methods, which does **NOT** involve the derivatives of $\boldsymbol{f}$, of solving the IVP

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0.$$

To see that this is possible, we note that Taylor's method of order 2 provides that

$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}\left[f_t(t,x) + f_x(t,x)f(t,x)\right] + \mathcal{O}(h^3).$$

On the other hand, Taylor's Theorem implies that

$$f\big(t+h, x+hf(t,x)\big) = f(t,x) + f_t(t,x)h + f_x(t,x)hf(t,x) + \mathcal{O}(h^2);$$

thus

$$x(t+h) = x(t) + hf(t,x) + \frac{h}{2}\left[f\big(t+h, x+hf(t,x)\big) - f(t,x)\right] + \mathcal{O}(h^3)$$

which provides "a" second order Runge-Kutta method

$$x_{n+1} = x_n + \qquad\qquad + \frac{h}{2}\left[f\big(t_{n+1}, x_n + hf(t_n,x_n)\big) + f(t_n,x_n)\right].$$

§6.4 The Runge-Kutta Method

The 2nd-order Runge-Kutta (RK) method derived previously is often written as the following alternative form

$$x_{n+1} = x_n + \frac{h}{2}(k_1 + k_2),$$

where

$$k_1 = f(t_n, x_n), \quad k_2 = f(t_n + h, x_n + hk_1).$$

This is also known as Heun's method.

Remark: Heun's method can be viewed as evaluating the number $\int_{t_n}^{t_{n+1}} f(t, x(t))\, dt$ using the trapezoidal rule, while the exact value of $x(t_{n+1})$ is unknown and is replaced by the forward Euler formula $x_n + hf(t_n, x_n)$:

$$\begin{aligned} x(t_{n+1}) - x(t_n) &= \int_{t_n}^{t_{n+1}} x'(t)\, dt = \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))\right] \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_n) + hf(t_n, x_n))\right]. \end{aligned}$$

§6.4 The Runge-Kutta Method

The 2nd-order Runge-Kutta (RK) method derived previously is often written as the following alternative form

$$x_{n+1} = x_n + \frac{h}{2}(k_1 + k_2),$$

where

$$k_1 = f(t_n, x_n), \quad k_2 = f(t_n + h, x_n + hk_1).$$

This is also known as Heun's method.

Remark: Heun's method can be viewed as evaluating the number $\int_{t_n}^{t_{n+1}} f(t, x(t))\, dt$ using the trapezoidal rule, while the exact value of $x(t_{n+1})$ is unknown and is replaced by the forward Euler formula $x_n + hf(t_n, x_n)$:

$$\begin{aligned} x(t_{n+1}) - x(t_n) &= \int_{t_n}^{t_{n+1}} x'(t)\, dt = \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))\right] \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_n) + hf(t_n, x_n))\right]. \end{aligned}$$

§6.4 The Runge-Kutta Method

The 2nd-order Runge-Kutta (RK) method derived previously is often written as the following alternative form

$$x_{n+1} = x_n + \frac{h}{2}(k_1 + k_2),$$

where

$$k_1 = f(t_n, x_n), \quad k_2 = f(t_n + h, x_n + hk_1).$$

This is also known as Heun's method.

Remark: Heun's method can be viewed as evaluating the number $\int_{t_n}^{t_{n+1}} f(t, x(t))\, dt$ using the trapezoidal rule, while the exact value of $x(t_{n+1})$ is unknown and is replaced by the forward Euler formula $x_n + hf(t_n, x_n)$:

$$\begin{aligned} x(t_{n+1}) - x(t_n) &= \int_{t_n}^{t_{n+1}} x'(t)\, dt = \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))\right] \\ &\approx \frac{t_{n+1} - t_n}{2}\left[f(t_n, x(t_n)) + f(t_{n+1}, x(t_n) + hf(t_n, x_n))\right]. \end{aligned}$$

§6.4 The Runge-Kutta Method

The 2nd-order Runge-Kutta (RK) method derived previously is often written as the following alternative form

$$x_{n+1} = x_n + \frac{h}{2}(k_1 + k_2),$$

where

$$k_1 = f(t_n, x_n), \quad k_2 = f(t_n + h, x_n + hk_1).$$

This is also known as Heun's method.

Remark: Heun's method can be viewed as evaluating the number $\int_{t_n}^{t_{n+1}} f(t, x(t))\, dt$ using the trapezoidal rule, while the exact value of $x(t_{n+1})$ is unknown and is replaced by the forward Euler formula $x_n + hf(t_n, x_n)$:

$$\begin{aligned} x(t_{n+1}) - x(t_n) &= \int_{t_n}^{t_{n+1}} x'(t)\, dt = \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \\ &\approx \frac{t_{n+1} - t_n}{2}\Big[f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))\Big] \\ &\approx \frac{t_{n+1} - t_n}{2}\Big[f(t_n, x(t_n)) + f(t_{n+1}, x(t_n) + hf(t_n, x_n))\Big]. \end{aligned}$$

§6.4 The Runge-Kutta Method

In general, a 2nd order Runge-Kutta method is of to choose real numbers ω_1, ω_2 and α, β so that

$$\begin{aligned} & x(t+h) \\ &= x(t) + \omega_1 h f(t,x) + \omega_2 h \underline{f(t+\alpha h, x+\beta h f(t,x))} + \mathcal{O}(h^3), \\ &= x(t) + \omega_1 h f(t,x) + \omega_2 h \Big[\underline{f(t,x) + \alpha h f_t(t,x) + \beta h f(t,x) f_x(t,x)} \Big] \\ &\quad + \mathcal{O}(h^3) \\ &= x(t) + (\omega_1+\omega_2) h f(t,x) + h^2 \Big[\omega_2 \alpha f_t(t,x) + \omega_2 \beta f(t,x) f_x(t,x) \Big] \\ &\quad + \mathcal{O}(h^3). \end{aligned}$$

Comparing with

$$x(t+h) = x(t) + hf + \frac{h^2}{2}\big[f_t(t,x) + f_x(t,x) f(t,x)\big] + \mathcal{O}(h^3),$$

we have

$$\begin{aligned} \omega_1 + \omega_2 &= 1, \\ \omega_2 \alpha = \omega_2 \beta &= 1/2. \end{aligned}$$

§6.4 The Runge-Kutta Method

In general, a 2nd order Runge-Kutta method is of to choose real numbers ω_1, ω_2 and α, β so that

$$\begin{aligned}
&x(t+h) \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \underline{f(t+\alpha h, x+\beta h f(t,x))} + \mathcal{O}(h^3), \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \Big[\underline{f(t,x) + \alpha h f_t(t,x) + \beta h f(t,x) f_x(t,x)}\Big] \\
&\qquad + \mathcal{O}(h^3) \\
&\quad = x(t) + (\omega_1+\omega_2) h f(t,x) + h^2 \Big[\omega_2 \alpha f_t(t,x) + \omega_2 \beta f(t,x) f_x(t,x)\Big] \\
&\qquad + \mathcal{O}(h^3).
\end{aligned}$$

Comparing with

$$x(t+h) = x(t) + hf + \frac{h^2}{2}\big[f_t(t,x) + f_x(t,x) f(t,x)\big] + \mathcal{O}(h^3),$$

we have

$$\begin{aligned}
\omega_1 + \omega_2 &= 1, \\
\omega_2 \alpha = \omega_2 \beta &= 1/2.
\end{aligned}$$

§6.4 The Runge-Kutta Method

In general, a 2nd order Runge-Kutta method is of to choose real numbers ω_1, ω_2 and α, β so that

$$\begin{aligned}
&x(t+h) \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \underline{f(t+\alpha h, x+\beta h f(t,x))} + \mathcal{O}(h^3), \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \Big[\underline{f(t,x) + \alpha h f_t(t,x) + \beta h f(t,x) f_x(t,x)}\Big] \\
&\qquad + \mathcal{O}(h^3) \\
&\quad = x(t) + (\omega_1+\omega_2) h f(t,x) + h^2 \Big[\omega_2 \alpha f_t(t,x) + \omega_2 \beta f(t,x) f_x(t,x)\Big] \\
&\qquad + \mathcal{O}(h^3).
\end{aligned}$$

Comparing with

$$x(t+h) = x(t) + hf + \frac{h^2}{2}\big[f_t(t,x) + f_x(t,x) f(t,x)\big] + \mathcal{O}(h^3),$$

we have

$$\begin{aligned}
\omega_1 + \omega_2 &= 1, \\
\omega_2 \alpha = \omega_2 \beta &= 1/2.
\end{aligned}$$

§6.4 The Runge-Kutta Method

In general, a 2nd order Runge-Kutta method is of to choose real numbers ω_1, ω_2 and α, β so that

$$\begin{aligned}
&x(t+h) \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \underline{f(t+\alpha h, x+\beta h f(t,x))} + \mathcal{O}(h^3), \\
&\quad = x(t) + \omega_1 h f(t,x) + \omega_2 h \Big[\underline{f(t,x) + \alpha h f_t(t,x) + \beta h f(t,x) f_x(t,x)} \Big] \\
&\qquad + \mathcal{O}(h^3) \\
&\quad = x(t) + (\omega_1+\omega_2) h f(t,x) + h^2 \Big[\omega_2 \alpha f_t(t,x) + \omega_2 \beta f(t,x) f_x(t,x) \Big] \\
&\qquad + \mathcal{O}(h^3).
\end{aligned}$$

Comparing with

$$x(t+h) = x(t) + hf + \frac{h^2}{2}\big[f_t(t,x) + f_x(t,x) f(t,x)\big] + \mathcal{O}(h^3),$$

we have

$$\begin{aligned}
\omega_1 + \omega_2 &= 1, \\
\omega_2 \alpha = \omega_2 \beta &= 1/2.
\end{aligned}$$

§6.4 The Runge-Kutta Method

1. The previous method (Heun's method) is obtained by setting
$$\begin{cases} \omega_1 = \omega_2 = 1/2, \\ \alpha = \beta = 1. \end{cases}$$
2. The modified Euler method is obtained by setting
$$\begin{cases} \omega_1 = 0, \\ \omega_2 = 1, \\ \alpha = \beta = 1/2, \end{cases}$$
we obtain the following :
$$x(t+h) \approx x(t) + hk_2,$$
where
$$k_1 = f(t, x), \quad k_2 = f(t + \frac{h}{2}, x + \frac{h}{2}k_1).$$

§6.4 The Runge-Kutta Method

- The derivations of higher order RK methods are tedious. However, the formulas are rather elegant and easily programmed once they have been derived.
- The most popular 4th order Runge-Kutta method is based on

$$x(t+h) = x(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5),$$

where

$$\begin{cases} k_1 = f(t, x), \\ k_2 = f(t + \frac{h}{2}, x + \frac{h}{2}k_1), \\ k_3 = f(t + \frac{h}{2}, x + \frac{h}{2}k_2), \\ k_4 = f(t + h, x + hk_3). \end{cases}$$

We note that k_1, k_2, k_3, k_4 are all approximated value of $x'(t)$.

§6.4 The Runge-Kutta Method

- The derivations of higher order RK methods are tedious. However, the formulas are rather elegant and easily programmed once they have been derived.
- The most popular 4th order Runge-Kutta method is given by

$$x_{n+1} = x_n + \frac{h}{6}(k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}),$$

where

$$\begin{cases} k_{n1} = f(t_n, x_n), \\ k_{n2} = f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_{n1}), \\ k_{n3} = f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_{n2}), \\ k_{n4} = f(t_n + h, x_n + hk_{n3}). \end{cases}$$

We note that $k_{n1}, k_{n2}, k_{n3}, k_{n4}$ are all approximated value of $x'(t_n)$.

§6.4 The Runge-Kutta Method

To see that the method provided in the previous page is indeed fourth order, we apply Taylor's Theorem and obtain that

$$\begin{aligned} k_2 &= f(t+\frac{h}{2}, x+\frac{h}{2}k_1) \\ &= f(t,x) + f_t(t,x)\cdot\frac{h}{2} + f_x(t,x)\cdot\frac{h}{2}k_1 \\ &\quad +\frac{1}{2}\left[f_{tt}(t,x)\cdot\frac{h^2}{4} + 2f_{tx}(t,x)\cdot\frac{h}{2}\cdot\frac{h}{2}k_1 + f_{xx}(t,x)\cdot\frac{h^2k_1^2}{4}\right] \\ &\quad +\frac{1}{6}\left[f_{ttt}(t,x)\cdot\frac{h^3}{8} + 3f_{ttx}(t,x)\cdot\frac{h^2}{4}\cdot\frac{h}{2}k_1 + 3f_{txx}(t,x)\cdot\frac{h}{2}\cdot\frac{h^2k_1^2}{4}\right. \\ &\qquad \left. +f_{xxx}(t,x)\cdot\frac{h^3k_1^3}{8}\right] + \mathcal{O}(h^4)\,. \end{aligned}$$

The reason why we truncate at the order $\mathcal{O}(h^4)$ is that we are going to prove that

$$x(t+h) = x(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5),$$

is an order four numerical method for solving ODE.

§6.4 The Runge-Kutta Method

To see that the method provided in the previous page is indeed fourth order, we apply Taylor's Theorem and obtain that

$$\begin{aligned} k_2 &= f(t+\frac{h}{2}, x+\frac{h}{2}k_1) \\ &= f(t,x) + f_t(t,x)\cdot\frac{h}{2} + f_x(t,x)\cdot\frac{h}{2}k_1 \\ &\quad +\frac{1}{2}\left[f_{tt}(t,x)\cdot\frac{h^2}{4} + 2f_{tx}(t,x)\cdot\frac{h}{2}\cdot\frac{h}{2}k_1 + f_{xx}(t,x)\cdot\frac{h^2k_1^2}{4}\right] \\ &\quad +\frac{1}{6}\left[f_{ttt}(t,x)\cdot\frac{h^3}{8} + 3f_{ttx}(t,x)\cdot\frac{h^2}{4}\cdot\frac{h}{2}k_1 + 3f_{txx}(t,x)\cdot\frac{h}{2}\cdot\frac{h^2k_1^2}{4}\right. \\ &\qquad \left. +f_{xxx}(t,x)\cdot\frac{h^3k_1^3}{8}\right] + \mathcal{O}(h^4)\,. \end{aligned}$$

The reason why we truncate at the order $\mathcal{O}(h^4)$ is that we are going to prove that

$$x(t+h) = x(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5),$$

is an order four numerical method for solving ODE.

§6.4 The Runge-Kutta Method

$$
\begin{aligned}
k_2 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_1 f_x(t,x)\\
&\quad+\frac{h^2}{8}\left[f_{tt}(t,x)+2k_1 f_{tx}(t,x)+k_1^2 f_{xx}(t,x)\right]\\
&\quad+\frac{h^3}{48}\left[f_{ttt}(t,x)+3k_1 f_{ttx}(t,x)+3k_1^2 f_{txx}(t,x)+k_1^3 f_{xxx}(t,x)\right]+\mathcal{O}(h^4),\\
k_3 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_2 f_x(t,x)\\
&\quad+\frac{h^2}{8}\left[f_{tt}(t,x)+2k_2 f_{tx}(t,x)+k_2^2 f_{xx}(t,x)\right]\\
&\quad+\frac{h^3}{48}\left[f_{ttt}(t,x)+3k_2 f_{ttx}(t,x)+3k_2^2 f_{txx}(t,x)+k_2^3 f_{xxx}(t,x)\right]+\mathcal{O}(h^4),\\
k_4 &= f(t,x)+hf_t(t,x)+hk_3 f_x(t,x)\\
&\quad+\frac{h^2}{2}\left[f_{tt}(t,x)+2k_3 f_{tx}(t,x)+k_3^2 f_{xx}(t,x)\right]\\
&\quad+\frac{h^3}{6}\left[f_{ttt}(t,x)+3k_3 f_{ttx}(t,x)+3k_3^2 f_{txx}(t,x)+k_3^3 f_{xxx}(t,x)\right]+\mathcal{O}(h^4).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

$$
\begin{aligned}
k_2 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_1 f_x(t,x)\\
&\quad+\frac{h^2}{8}\Big[f_{tt}(t,x)+2k_1 f_{tx}(t,x)+k_1^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{48}\Big[f_{ttt}(t,x)+3k_1 f_{ttx}(t,x)+3k_1^2 f_{txx}(t,x)+k_1^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4),\\
k_3 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_2 f_x(t,x)\\
&\quad+\frac{h^2}{8}\Big[f_{tt}(t,x)+2k_2 f_{tx}(t,x)+k_2^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{48}\Big[f_{ttt}(t,x)+3k_2 f_{ttx}(t,x)+3k_2^2 f_{txx}(t,x)+k_2^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4),\\
k_4 &= f(t,x)+hf_t(t,x)+hk_3 f_x(t,x)\\
&\quad+\frac{h^2}{2}\Big[f_{tt}(t,x)+2k_3 f_{tx}(t,x)+k_3^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{6}\Big[f_{ttt}(t,x)+3k_3 f_{ttx}(t,x)+3k_3^2 f_{txx}(t,x)+k_3^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

$$
\begin{aligned}
k_2 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_1 f_x(t,x)\\
&\quad+\frac{h^2}{8}\Big[f_{tt}(t,x)+2k_1 f_{tx}(t,x)+k_1^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{48}\Big[f_{ttt}(t,x)+3k_1 f_{ttx}(t,x)+3k_1^2 f_{txx}(t,x)+k_1^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4),\\
k_3 &= f(t,x)+\frac{h}{2}f_t(t,x)+\frac{h}{2}k_2 f_x(t,x)\\
&\quad+\frac{h^2}{8}\Big[f_{tt}(t,x)+2k_2 f_{tx}(t,x)+k_2^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{48}\Big[f_{ttt}(t,x)+3k_2 f_{ttx}(t,x)+3k_2^2 f_{txx}(t,x)+k_2^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4),\\
k_4 &= f(t,x)+hf_t(t,x)+hk_3 f_x(t,x)\\
&\quad+\frac{h^2}{2}\Big[f_{tt}(t,x)+2k_3 f_{tx}(t,x)+k_3^2 f_{xx}(t,x)\Big]\\
&\quad+\frac{h^3}{6}\Big[f_{ttt}(t,x)+3k_3 f_{ttx}(t,x)+3k_3^2 f_{txx}(t,x)+k_3^3 f_{xxx}(t,x)\Big]+\mathcal{O}(h^4).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

Therefore,

$$
\begin{aligned}
&k_1 + 2k_2 + 2k_3 + k_4 \\
&\quad = 6f(t,x) + 3hf_t(t,x) + h(k_1 + k_2 + k_3)f_x(t,x) \\
&\qquad + h^2 f_{tt}(t,x) + \frac{h^2}{2}(k_1 + k_2 + 2k_3)f_{tx}(t,x) \\
&\qquad + \frac{h^2}{4}(k_1^2 + k_2^2 + 2k_3^2)f_{xx}(t,x) + \frac{h^3}{4}f_{ttt}(t,x) \\
&\qquad + \frac{h^3}{8}(k_1 + k_2 + 4k_3)f_{ttx}(t,x) + \frac{h^3}{8}(k_1^2 + k_2^2 + 4k_3^2)f_{txx}(t,x) \\
&\qquad + \frac{h^3}{24}(k_1^3 + k_2^3 + 4k_3^3)f_{xxx}(t,x) + \mathcal{O}(h^4)\,.
\end{aligned}
$$

Next, we need to compute

1. $k_1 + k_2 + k_3$ accurate at least of order $\mathcal{O}(h^2)$.
2. $k_1 + k_2 + 2k_3$ and $k_1^2 + k_2^2 + 2k_3^2$ accurate at least of order $\mathcal{O}(h)$.
3. $k_1 + k_2 + 4k_3$, $k_1^2 + k_2^2 + 4k_3^2$ and $k_1^3 + k_2^3 + 4k_3^3$ accurate at least of order $\mathcal{O}(1)$.

§6.4 The Runge-Kutta Method

Therefore,

$$
\begin{aligned}
&k_1 + 2k_2 + 2k_3 + k_4 \\
&\quad = 6f(t,x) + 3hf_t(t,x) + h(k_1+k_2+k_3)f_x(t,x) \\
&\qquad + h^2 f_{tt}(t,x) + \frac{h^2}{2}(k_1+k_2+2k_3)f_{tx}(t,x) \\
&\qquad + \frac{h^2}{4}(k_1^2+k_2^2+2k_3^2)f_{xx}(t,x) + \frac{h^3}{4}f_{ttt}(t,x) \\
&\qquad + \frac{h^3}{8}(k_1+k_2+4k_3)f_{ttx}(t,x) + \frac{h^3}{8}(k_1^2+k_2^2+4k_3^2)f_{txx}(t,x) \\
&\qquad + \frac{h^3}{24}(k_1^3+k_2^3+4k_3^3)f_{xxx}(t,x) + \mathcal{O}(h^4)\,.
\end{aligned}
$$

Next, we need to compute

1. $k_1 + k_2 + k_3$ accurate at least of order $\mathcal{O}(h^2)$.
2. $k_1+k_2+2k_3$ and $k_1^2+k_2^2+2k_3^2$ accurate at least of order $\mathcal{O}(h)$.
3. $k_1 + k_2 + 4k_3$, $k_1^2 + k_2^2 + 4k_3^2$ and $k_1^3 + k_2^3 + 4k_3^3$ accurate at least of order $\mathcal{O}(1)$.

§6.4 The Runge-Kutta Method

Using the formula for k_2, k_3, and k_4, we find that

$$\begin{aligned}
k_1+k_2+k_3 &= 3f(t,x)+hf_t(t,x)+\frac{h}{2}(k_1+k_2)f_x(t,x)+\frac{h^2}{4}f_{tt}(t,x)\\
&\quad+\frac{h^2}{4}(k_1+k_2)f_{tx}(t,x)+\frac{h^2}{8}(k_1^2+k_2^2)f_{xx}(t,x)+\mathcal{O}(h^3),\\
k_1+k_2+2k_3 &= 4f(t,x)+\frac{3h}{2}f_t(t,x)+\frac{h}{2}(k_1+2k_2)f_x(t,x)+\mathcal{O}(h^2),\\
k_1^2+k_2^2+2k_3^2 &= 4f(t,x)^2+hf(t,x)\left[3f_t(t,x)+(k_1+2k_2)f_x(t,x)\right]\\
&\quad+\mathcal{O}(h^2),\\
k_1+k_2+4k_3 &= 6f(t,x)+\mathcal{O}(h),\\
k_1^2+k_2^2+4k_3^2 &= 6f(t,x)^2+\mathcal{O}(h),\\
k_1^3+k_2^3+4k_3^3 &= 6f(t,x)^3+\mathcal{O}(h).
\end{aligned}$$

Since $k_1+2k_2=3f(t,x)+\mathcal{O}(h)$, to continue we compute

1. k_1+k_2 accurate at least of order $\mathcal{O}(h^2)$.
2. $k_1^2+k_2^2$ accurate at least of order $\mathcal{O}(h)$.

§6.4 The Runge-Kutta Method

Using the formula for k_2, k_3, and k_4, we find that

$$\begin{aligned}
k_1+k_2+k_3 &= 3f(t,x)+hf_t(t,x)+\frac{h}{2}(k_1+k_2)f_x(t,x)+\frac{h^2}{4}f_{tt}(t,x)\\
&\quad+\frac{h^2}{4}(k_1+k_2)f_{tx}(t,x)+\frac{h^2}{8}(k_1^2+k_2^2)f_{xx}(t,x)+\mathcal{O}(h^3),\\
k_1+k_2+2k_3 &= 4f(t,x)+\frac{3h}{2}f_t(t,x)+\frac{h}{2}(k_1+2k_2)f_x(t,x)+\mathcal{O}(h^2),\\
k_1^2+k_2^2+2k_3^2 &= 4f(t,x)^2+hf(t,x)\left[3f_t(t,x)+(k_1+2k_2)f_x(t,x)\right]\\
&\quad+\mathcal{O}(h^2),\\
k_1+k_2+4k_3 &= 6f(t,x)+\mathcal{O}(h),\\
k_1^2+k_2^2+4k_3^2 &= 6f(t,x)^2+\mathcal{O}(h),\\
k_1^3+k_2^3+4k_3^3 &= 6f(t,x)^3+\mathcal{O}(h).
\end{aligned}$$

Since $k_1+2k_2=3f(t,x)+\mathcal{O}(h)$, to continue we compute

1. k_1+k_2 accurate at least of order $\mathcal{O}(h^2)$.
2. $k_1^2+k_2^2$ accurate at least of order $\mathcal{O}(h)$.

§6.4 The Runge-Kutta Method

Since

$$k_1 + k_2 = 2f(t,x) + \frac{h}{2}\big[f_t(t,x) + f(t,x)f_x(t,x)\big] + \mathcal{O}(h^2)$$
$$k_1^2 + k_2^2 = 2f(t,x)^2 + hf(t,x)\big[f(t,x) + f(t,x)f_t(t,x)\big] + \mathcal{O}(h^2)$$

we obtain that

$$\begin{aligned}
k_1+k_2+k_3 &= 3f(t,x)+h\big[f_t(t,x)+f(t,x)f_x(t,x)\big] \\
&\quad +\frac{h^2}{4}\big[f_{tt}(t,x) + f_t(t,x)f_x(t,x) + f(t,x)f_x(t,x)^2 \\
&\qquad +2f(t,x)f_{tx}(t,x) + f(t,x)^2 f_{xx}(t,x)\big]+\mathcal{O}(h^3) \\
k_1+k_2+2k_3 &= 4f(t,x)+\frac{3h}{2}\big[f_t(t,x)+f(t,x)f_x(t,x)\big]+\mathcal{O}(h^2), \\
k_1^2+k_2^2+2k_3^2 &= 4f(t,x)^2+3hf(t,x)\big[f_t(t,x)+f(t,x)f_x(t,x)\big]+\mathcal{O}(h^2), \\
k_1+k_2+4k_3 &= 6f(t,x)+\mathcal{O}(h), \\
k_1^2+k_2^2+4k_3^2 &= 6f(t,x)^2+\mathcal{O}(h), \\
k_1^3+k_2^3+4k_3^3 &= 6f(t,x)^3+\mathcal{O}(h).
\end{aligned}$$

§6.4 The Runge-Kutta Method

Therefore,

$$
\begin{aligned}
&k_1 + 2k_2 + 2k_3 + k_4 \\
&\quad = 6f(t,x) + 3h\big[f_t(t,x) + f(t,x)f_x(t,x)\big] \\
&\qquad + h^2\big[f_t(t,x) + f(t,x)f_x(t,x)\big]f_x(t,x) \\
&\qquad + \frac{h^3}{4}\big[f_{tt}(t,x) + f_t(t,x)f_x(t,x) + f(t,x)f_x(t,x)^2 \\
&\qquad\qquad + 2f(t,x)f_{tx}(t,x) + f(t,x)^2 f_{xx}(t,x)\big]f_x(t,x) + h^2 f_{tt}(t,x) \\
&\qquad + 2h^2 f(t,x)f_{tx}(t,x) + \frac{3h^3}{4}\big[f_t(t,x) + f(t,x)f_x(t,x)\big]f_{tx}(t,x) \\
&\qquad + h^2 f(t,x)^2 f_{xx}(t,x) + \frac{3h^3}{4}f(t,x)\big[f_t(t,x) + f(t,x)f_x(t,x)\big]f_{xx}(t,x) \\
&\qquad + \frac{h^3}{4}f_{ttt}(t,x) + \frac{3h^3}{4}f(t,x)f_{ttx}(t,x) + \frac{3h^3}{4}f(t,x)^2 f_{txx}(t,x) \\
&\qquad + \frac{h^3}{4}f(t,x)^3 f_{xxx}(t,x) + \mathcal{O}(h^4)\,.
\end{aligned}
$$

§6.4 The Runge-Kutta Method

Therefore,

$$
\begin{aligned}
&k_1 + 2k_2 + 2k_3 + k_4 \\
&\quad = 6f(t,x) + 3h\big[f_t(t,x) + f(t,x)f_x(t,x)\big] \\
&\qquad + h^2\big[f_{tt}(t,x) + f_t(t,x)f_x(t,x) + f(t,x)f_x(t,x)^2 + 2f(t,x)f_{tx}(t,x) \\
&\qquad\quad + f(t,x)^2 f_{xx}(t,x)\big] \\
&\qquad + \frac{h^3}{4}\big[f_{ttt}(t,x) + f_{tt}(t,x)f_x(t,x) + f_t(t,x)f_x(t,x)^2 + f(t,x)f_x(t,x)^3 \\
&\qquad\quad + f(t,x)^3 f_{xxx}(t,x) + 2f(t,x)f_x(t,x)f_{tx}(t,x) \\
&\qquad\quad + 3f(t,x)f_{ttx}(t,x) + 3f_t(t,x)f_{tx}(t,x) \\
&\qquad\quad + 3f(t,x)f_x(t,x)f_{tx}(t,x) + 3f(t,x)f_t(t,x)f_{xx}(t,x) \\
&\qquad\quad + 3f(t,x)^2 f_{txx}(t,x) + 4f(t,x)^2 f_x(t,x)f_{xx}(t,x)\big] + \mathcal{O}(h^4)
\end{aligned}
$$

§6.4 The Runge-Kutta Method

which implies that

$$
\begin{aligned}
& x(t)+\frac{h}{6}\left[k_1+2k_2+2k_3+k_4\right] \\
&= x(t)+hf(t,x)+\frac{h^2}{2}\left[f_t(t,x)+f(t,x)f_x(t,x)\right] \\
&\quad +\frac{h^3}{6}\left[f_{tt}(t,x)+f_t(t,x)f_x(t,x)+f(t,x)f_x(t,x)^2+2f(t,x)f_{tx}(t,x)\right. \\
&\qquad \left.+f(t,x)^2f_{xx}(t,x)\right] \\
&\quad +\frac{h^4}{24}\left[f_{ttt}(t,x)+f_{tt}(t,x)f_x(t,x)+f_t(t,x)f_x(t,x)^2+f(t,x)f_x(t,x)^3\right. \\
&\qquad +f(t,x)^3f_{xxx}(t,x)+2f(t,x)f_x(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_{ttx}(t,x)+3f_t(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_x(t,x)f_{tx}(t,x)+3f(t,x)f_t(t,x)f_{xx}(t,x) \\
&\qquad \left.+3f(t,x)^2f_{txx}(t,x)+4f(t,x)^2f_x(t,x)f_{xx}(t,x)\right]+\mathcal{O}(h^5) \\
&= x(t)+hx'(t)+\frac{h^2}{2}x''(t)+\frac{h^3}{6}x'''(t)+\frac{h^4}{24}x^{(4)}(t)+\mathcal{O}(h^5).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

which implies that

$$
\begin{aligned}
& x(t)+\frac{h}{6}\left[k_1+2k_2+2k_3+k_4\right] \\
&= x(t)+hf(t,x)+\frac{h^2}{2}\left[f_t(t,x)+f(t,x)f_x(t,x)\right] \\
&\quad +\frac{h^3}{6}\left[f_{tt}(t,x)+f_t(t,x)f_x(t,x)+f(t,x)f_x(t,x)^2+2f(t,x)f_{tx}(t,x)\right. \\
&\qquad \left.+f(t,x)^2f_{xx}(t,x)\right] \\
&\quad +\frac{h^4}{24}\left[f_{ttt}(t,x)+f_{tt}(t,x)f_x(t,x)+f_t(t,x)f_x(t,x)^2+f(t,x)f_x(t,x)^3\right. \\
&\qquad +f(t,x)^3f_{xxx}(t,x)+2f(t,x)f_x(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_{ttx}(t,x)+3f_t(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_x(t,x)f_{tx}(t,x)+3f(t,x)f_t(t,x)f_{xx}(t,x) \\
&\qquad \left.+3f(t,x)^2f_{txx}(t,x)+4f(t,x)^2f_x(t,x)f_{xx}(t,x)\right]+\mathcal{O}(h^5) \\
&= x(t)+hx'(t)+\frac{h^2}{2}x''(t)+\frac{h^3}{6}x'''(t)+\frac{h^4}{24}x^{(4)}(t)+\mathcal{O}(h^5).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

which implies that

$$
\begin{aligned}
& x(t)+\frac{h}{6}\left[k_1+2k_2+2k_3+k_4\right] \\
&= x(t)+hf(t,x)+\frac{h^2}{2}\left[f_t(t,x)+f(t,x)f_x(t,x)\right] \\
&\quad +\frac{h^3}{6}\left[f_{tt}(t,x)+f_t(t,x)f_x(t,x)+f(t,x)f_x(t,x)^2+2f(t,x)f_{tx}(t,x)\right. \\
&\qquad \left.+f(t,x)^2f_{xx}(t,x)\right] \\
&\quad +\frac{h^4}{24}\left[f_{ttt}(t,x)+f_{tt}(t,x)f_x(t,x)+f_t(t,x)f_x(t,x)^2+f(t,x)f_x(t,x)^3\right. \\
&\qquad +f(t,x)^3f_{xxx}(t,x)+2f(t,x)f_x(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_{ttx}(t,x)+3f_t(t,x)f_{tx}(t,x) \\
&\qquad +3f(t,x)f_x(t,x)f_{tx}(t,x)+3f(t,x)f_t(t,x)f_{xx}(t,x) \\
&\qquad \left.+3f(t,x)^2f_{txx}(t,x)+4f(t,x)^2f_x(t,x)f_{xx}(t,x)\right]+\mathcal{O}(h^5) \\
&= x(t)+hx'(t)+\frac{h^2}{2}x''(t)+\frac{h^3}{6}x'''(t)+\frac{h^4}{24}x^{(4)}(t)+\mathcal{O}(h^5).
\end{aligned}
$$

§6.4 The Runge-Kutta Method

Computer Project:

1. Use the most popular 4th order Runge-Kutta with $h = 1/128$, $h = 1/256$ and $h = 1/512$ to solve the following IVP for $t \in [1, 3]$ and then plot the piecewise linear approximate solution:

$$\begin{cases} x'(t) = t^{-2}(tx - x^2), \\ x(1) = 2. \end{cases}$$

2. Also plot the exact solution:

$$x(t) = (1/2 + \ln t)^{-1} t.$$

3. Find the global truncation error $E(h)$ and "verify" numerically that the 4th order Runge-Kutta is indeed a fourth order numerical method of solving IVPs.

§6.4 The Runge-Kutta Method

input $M \leftarrow 256,\ t \leftarrow 1.0,\ h \leftarrow 0.0078125,\ x \leftarrow 2.0$

define $f(t,x) = (tx - x^2)/t^2,\ u(t) = t/(1/2 + \ln t)$

$e \leftarrow |u(t) - x|$

output $0, t, x, e$

for $k = 1$ **to** M **do**

$$
\begin{aligned}
k_1 &\leftarrow f(t,x) \\
k_2 &\leftarrow f(t + \frac{h}{2}, x + \frac{h}{2}k_1) \\
k_3 &\leftarrow f(t + \frac{h}{2}, x + \frac{h}{2}k_2) \\
k_4 &\leftarrow f(t + h, x + hk_3) \\
x &\leftarrow x + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
t &\leftarrow t + h \\
e &\leftarrow |u(t) - x|
\end{aligned}
$$

output k, t, x, e

end do

§6.4 The Runge-Kutta Method

For a system of equations $\boldsymbol{x}'(t) = \boldsymbol{f}(t, \boldsymbol{x})$, we have 4th-order Runge-Kutta method:

$$\boldsymbol{x}(t+h) = \boldsymbol{x}(t) + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4) + \mathcal{O}(h^4),$$

where

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}\big(t, \boldsymbol{x}(t+h)\big), \\
\boldsymbol{k}_2 &= \boldsymbol{f}\big(t + \frac{h}{2}, \boldsymbol{x}(t) + \frac{h}{2}\boldsymbol{k}_1\big), \\
\boldsymbol{k}_3 &= \boldsymbol{f}\big(t + \frac{h}{2}, \boldsymbol{x}(t) + \frac{h}{2}\boldsymbol{k}_2\big), \\
\boldsymbol{k}_4 &= \boldsymbol{f}\big(t + h, \boldsymbol{x}(t) + h\boldsymbol{k}_3\big).
\end{aligned}$$

Other methods, they are all similar to the single equation case.

§6.5 Collocation Method

The idea of **collocation methods** is to choose a finite-dimensional space of candidate solutions (usually polynomials up to a certain degree) and a number of points in the domain (called collocation points), and to select that solution which satisfies the given equation at the collocation points.

Suppose that the ordinary differential equation

$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0$$

is to be solved over the interval $[t_0, t_0 + c_k h]$. Choose c_k from $0 \leqslant c_1 < c_2 < \ldots < c_n \leqslant 1$. The corresponding (polynomial) collocation method approximates the solution y by the polynomial p of degree n which satisfies the initial condition $p(t_0) = y_0$ and the differential equation $p'(t_k) = f(t_k, p(t_k))$.

§6.5 Collocation Method

The idea of **collocation methods** is to choose a finite-dimensional space of candidate solutions (usually polynomials up to a certain degree) and a number of points in the domain (called collocation points), and to select that solution which satisfies the given equation at the collocation points.

Suppose that the ordinary differential equation

$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0$$

is to be solved over the interval $\left[t_0, t_0 + c_k h\right]$. Choose c_k from $0 \leqslant c_1 < c_2 < \ldots < c_n \leqslant 1$. The corresponding (polynomial) collocation method approximates the solution y by the polynomial p of degree n which satisfies the initial condition $p(t_0) = y_0$ and the differential equation $p'(t_k) = f(t_k, p(t_k))$.

§6.5 Collocation Method

Example

Pick to collocation points $c_1 = 0$ and $c_2 = 1$. The collocation method is then looking for a polynomial p of degree 2 satisfying the collocation conditions

$$\begin{aligned} p(t_0) &= y_0, \\ p'(t_0) &= f\big(t_0, p(t_0)\big), \\ p'(t_1) &= f\big(t_1, p(t_1)\big). \end{aligned}$$

If $p(t) = \alpha(t-t_0)^2 + \beta(t-t_0) + \gamma$, then the collocation conditions above become

$$\begin{aligned} \gamma &= y_0, \\ \beta &= f(t_0, \gamma), \\ \alpha &= \frac{1}{2h}\Big[f\big(t_0 + h, \alpha h^2 + \beta h + \gamma\big) - f(t_0, \gamma)\Big]. \end{aligned}$$

It remains to solve for α from a nonlinear equation.

§6.5 Collocation Method

Example

Pick to collocation points $c_1 = 0$ and $c_2 = 1$. The collocation method is then looking for a polynomial p of degree 2 satisfying the collocation conditions

$$\begin{aligned} p(t_0) &= y_0, \\ p'(t_0) &= f\big(t_0, p(t_0)\big), \\ p'(t_1) &= f\big(t_1, p(t_1)\big). \end{aligned}$$

If $p(t) = \alpha(t-t_0)^2 + \beta(t-t_0) + \gamma$, then the collocation conditions above become

$$\begin{aligned} \gamma &= y_0, \\ \beta &= f(t_0, \gamma), \\ \alpha &= \frac{1}{2h}\Big[f\big(t_0+h, \alpha h^2 + \beta h + \gamma\big) - f(t_0, \gamma)\Big]. \end{aligned}$$

It remains to solve for α from a nonlinear equation.

§6.5 Collocation Method

Suppose that we have a linear differential operator L and we wish to solve the equation: given a function f, find the function u satisfying

$$Lu(t) = f(t), \quad a < t < b.$$

1. Let $\{v_1, v_2, \cdots, v_n\}$ be a set of functions that are linearly independent. Suppose that $u(t) \approx c_1 v_1(t) + c_2 v_2(t) + \cdots + c_n v_n(t)$.
2. Then solve $L\big(\sum_{j=1}^{n} c_j v_j(t)\big) = f(t)$. How to determine c_j?
3. Let $t_1, t_2, \cdots, t_n$ be n prescribed points (collocation points) in the domain of u and f. Then we require that
$$\sum_{j=1}^{n} c_j (Lv_j)(t_i) = f(t_i), \quad i = 1, 2, \cdots, n.$$
4. This is a system of n linear equations in n unknowns c_j. The functions v_j and the points t_i should be chosen so that the matrix with entries $(Lv_j)(t_i)$ is non-singular.

§6.5 Collocation Method

Example (Collocation method for Sturm-Liouville BVPs)

Consider a Sturm-Liouville two-point BVP:

$$\begin{cases} u''(t) + p(t)u'(t) + q(t)u(t) = f(t) \quad \forall\, t \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{$\square$}$$

where p, q, f are given continuous functions on $[0,1]$

1. Let $Lu := u'' + pu' + qu$. Define the vector space

$$\mathcal{V} = \left\{ u \in C^2((0,1)) \cap C([0,1]) \,\middle|\, u(0) = u(1) = 0 \right\}.$$

   If u is an exact solution of $(\square)$, then $u \in \mathcal{V}$.

2. One set of functions is given by

$$v_{jk}(t) = t^j(1-t)^k \in C^2([0,1]), \quad 1 \leqslant j \leqslant m, 1 \leqslant k \leqslant n.$$

§6.6 Finite Difference Method (FDM)

The finite-difference methods (FDM) are discretizations used for solving differential equations by approximating them with difference equations that finite differences approximate the derivatives.

Example

Again we consider the Sturm-Liouville two-point BVP:

$$\begin{cases} u''(t) + p(t)u'(t) + q(t)u(t) = f(t) \quad \forall\, t \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{$\square$}$$

Let $t_k = \dfrac{k}{n+1}$ for $k = 0, 1, \cdots, n+1$. Using the central difference approximation, ($\square$) implies that

$$\begin{aligned} &\frac{u(t_{k-1}) - 2u(t_k) + u(t_{k+1})}{h^2} + p(t_k)\frac{u(t_{k+1}) - u(t_{k-1})}{2h} + q(t_k)u(t_k) \\ &\quad = f(t_k) + \mathcal{O}(h^2) \qquad \text{for } k = 1, 2, \cdots, n, \\ &u(t_0) = u(t_{n+1}) = 0. \end{aligned}$$

§6.6 Finite Difference Method (FDM)

The finite-difference methods (FDM) are discretizations used for solving differential equations by approximating them with difference equations that finite differences approximate the derivatives.

Example

Again we consider the Sturm-Liouville two-point BVP:

$$\begin{cases} u''(t) + p(t)u'(t) + q(t)u(t) = f(t) \quad \forall\, t \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{$\square$}$$

Let $t_k = \dfrac{k}{n+1}$ for $k = 0, 1, \cdots, n+1$. Using the central difference approximation, ($\square$) implies that

$$\begin{aligned} &\frac{u(t_{k-1}) - 2u(t_k) + u(t_{k+1})}{h^2} + p(t_k)\frac{u(t_{k+1}) - u(t_{k-1})}{2h} + q(t_k)u(t_k) \\ &\quad = f(t_k) + \mathcal{O}(h^2) \qquad \text{for } k = 1, 2, \cdots, n, \\ &u(t_0) = u(t_{n+1}) = 0. \end{aligned}$$

§6.6 Finite Difference Method (FDM)

The finite-difference methods (FDM) are discretizations used for solving differential equations by approximating them with difference equations that finite differences approximate the derivatives.

Example

Again we consider the Sturm-Liouville two-point BVP:

$$\begin{cases} u''(t) + p(t)u'(t) + q(t)u(t) = f(t) \quad \forall\, t \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{$\square$}$$

Let $t_k = \dfrac{k}{n+1}$ for $k = 0, 1, \cdots, n+1$. Using the central difference approximation, ($\square$) implies that

$$\frac{u(t_{k-1}) - 2u(t_k) + u(t_{k+1})}{h^2} + \frac{p(t_k)u(t_{k+1}) + 2hq(t_k)u(t_k) - p(t_k)u(t_{k-1})}{2h}$$
$$= f(t_k) + \mathcal{O}(h^2) \qquad \text{for } k = 1, 2, \cdots, n,$$
$$u(t_0) = u(t_{n+1}) = 0.$$

§6.6 Finite Difference Method (FDM)

Example (Cont'd)

Define $A(h)$ as the $n \times n$ matrix

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix} + \frac{1}{2h}\begin{bmatrix} 2hq(t_1) & p(t_1) & 0 & \cdots & 0 \\ -p(t_2) & 2hq(t_2) & p(t_2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -p(t_{n-1}) & 2hq(t_{n-1}) & p(t_{n-1}) \\ 0 & \cdots & 0 & -p(t_n) & 2hq(t_n) \end{bmatrix}.$$

Then

$$A(h)\begin{bmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ \vdots \\ u(t_n) \end{bmatrix} = \begin{bmatrix} f(t_1) \\ f(t_2) \\ \vdots \\ \vdots \\ f(t_n) \end{bmatrix} + \mathcal{O}(h^2).$$

§6.6 Finite Difference Method (FDM)

Example (Cont'd)

In particular, if $p \equiv 0$ and $q < 0$ on $(0, 1)$, then $A(h)$ is diagonal dominant:

$$\left|\frac{-2}{h^2} + q(t_k)\right| = \frac{2}{h^2} - q(t_k) > \frac{1}{h^2} + \frac{1}{h^2}\,.$$

Therefore, $A(h)$ is invertible if $p \equiv 0$ and $q < 0$.

Remark: Let $f(t, y, y') = f(t) - q(t)y(t) - p(t)y'(t)$. Then the condition $q < 0$ on $(0, 1)$ corresponds to the condition

$$f_y > 0$$

in the statement of the existence theorem. This is an indication why we need this condition in the existence theorem.

§6.7 Finite Element Method (FEM)

We begin with considering the following two-point boundary value problem:

$$\begin{cases} -u''(x) = f(x) \quad \forall\, x \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{D}$$

where f is a given function in $C([0,1])$.

Remark: (D) has a unique classical solution.

Let $v \in C([0,1])$, piecewise differentiable, and $v(0) = v(1) = 0$.

Then integration by parts implies that

$$\begin{aligned} \int_0^1 f(x)v(x)\,dx &= -\int_0^1 u''(x)v(x)\,dx \\ &= -u'(x)v(x)\Big|_{x=0}^{x=1} + \int_0^1 u'(x)v'(x)\,dx \\ &= \int_0^1 u'(x)v'(x)\,dx. \end{aligned}$$

§6.7 Finite Element Method (FEM)

We begin with considering the following two-point boundary value problem:

$$\begin{cases} -u''(x) = f(x) \quad \forall\, x \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{D}$$

where f is a given function in $C([0,1])$.

Remark: (D) has a unique classical solution.

Let $v \in C([0,1])$, piecewise differentiable, and $v(0) = v(1) = 0$. Then integration by parts implies that

$$\begin{aligned} \int_0^1 f(x)v(x)\,dx &= -\int_0^1 u''(x)v(x)\,dx \\ &= -u'(x)v(x)\Big|_{x=0}^{x=1} + \int_0^1 u'(x)v'(x)\,dx \\ &= \int_0^1 u'(x)v'(x)\,dx. \end{aligned}$$

§6.7 Finite Element Method (FEM)

Note that if u is a solution to (D), then u also satisfy the identity

$$\int_0^1 u'(x)v'(x)\,dx = \int_0^1 f(x)v(x)\,dx \qquad \forall\, v \in \mathcal{V}\,, \tag{V}$$

where $\mathcal{V}$ is the collection of all continuous, piecewise differentiable functions on $[0,1]$ that vanish at $x = 0$ and $x = 1$. Therefore, instead of solving for (D), we first look for a function $u \in \mathcal{V}$ satisfies (V). (V) is called the **variational form** of (D).

We also note that if (V) has a solution u, the function u might not satisfy (D) because u may not be twice differentiable. However, if $u \in C^2((0,1)) \cap \mathcal{V}$ satisfies (V), then for all $v \in \mathcal{V}$,

$$\int_0^1 \big[f(x) + u''(x)\big]v(x)\,dx = \int_0^1 \big[f(x)v(x) - u'(x)v'(x)\big]\,dx = 0;$$

thus $-u''(x) = f(x)$ for all $x \in (0,1)$. A solution of (V) is called a **weak solution** of (D), and (V) is also called the **weak form** of (D).

§6.7 Finite Element Method (FEM)

Note that if u is a solution to (D), then u also satisfy the identity

$$\int_0^1 u'(x)v'(x)\,dx = \int_0^1 f(x)v(x)\,dx \qquad \forall\, v \in \mathcal{V}\,, \tag{V}$$

where $\mathcal{V}$ is the collection of all continuous, piecewise differentiable functions on $[0,1]$ that vanish at $x = 0$ and $x = 1$. Therefore, instead of solving for (D), we first look for a function $u \in \mathcal{V}$ satisfies (V). (V) is called the **variational form** of (D).

We also note that if (V) has a solution u, the function u might not satisfy (D) because u may not be twice differentiable. However, if $u \in C^2((0,1)) \cap \mathcal{V}$ satisfies (V), then for all $v \in \mathcal{V}$,

$$\int_0^1 \big[f(x) + u''(x)\big]v(x)\,dx = \int_0^1 \big[f(x)v(x) - u'(x)v'(x)\big]\,dx = 0;$$

thus $-u''(x) = f(x)$ for all $x \in (0,1)$. A solution of (V) is called a **weak solution** of (D), and (V) is also called the **weak form** of (D).

§6.7 Finite Element Method (FEM)

Note that if u is a solution to (D), then u also satisfy the identity

$$\int_0^1 u'(x)v'(x)\,dx = \int_0^1 f(x)v(x)\,dx \qquad \forall\, v \in \mathcal{V}\,, \tag{V}$$

where $\mathcal{V}$ is the collection of all continuous, piecewise differentiable functions on $[0,1]$ that vanish at $x = 0$ and $x = 1$. Therefore, instead of solving for (D), we first look for a function $u \in \mathcal{V}$ satisfies (V). (V) is called the **variational form** of (D).

We also note that if (V) has a solution u, the function u might not satisfy (D) because u may not be twice differentiable. However, if $u \in C^2((0,1)) \cap \mathcal{V}$ satisfies (V), then for all $v \in \mathcal{V}$,

$$\int_0^1 \big[f(x) + u''(x)\big]v(x)\,dx = \int_0^1 \big[f(x)v(x) - u'(x)v'(x)\big]\,dx = 0;$$

thus $-u''(x) = f(x)$ for all $x \in (0,1)$. A solution of (V) is called a **weak solution** of (D), and (V) is also called the **weak form** of (D).

§6.7 Finite Element Method (FEM)

Note that if u is a solution to (D), then u also satisfy the identity

$$\int_0^1 u'(x)v'(x)\,dx = \int_0^1 f(x)v(x)\,dx \qquad \forall\, v \in \mathcal{V}\,, \tag{V}$$

where $\mathcal{V}$ is the collection of all continuous, piecewise differentiable functions on $[0,1]$ that vanish at $x = 0$ and $x = 1$. Therefore, instead of solving for (D), we first look for a function $u \in \mathcal{V}$ satisfies (V). (V) is called the **variational form** of (D).

We also note that if (V) has a solution u, the function u might not satisfy (D) because u may not be twice differentiable. However, if $u \in C^2((0,1)) \cap \mathcal{V}$ satisfies (V), then for all $v \in \mathcal{V}$,

$$\int_0^1 \big[f(x) + u''(x)\big]v(x)\,dx = \int_0^1 \big[f(x)v(x) - u'(x)v'(x)\big]\,dx = 0;$$

thus $-u''(x) = f(x)$ for all $x \in (0,1)$. A solution of (V) is called a **weak solution** of (D), and (V) is also called the **weak form** of (D).

§6.7 Finite Element Method (FEM)

Let $F : \mathcal{V} \to \mathbb{R}$ be defined by

$$F(v) = \frac{1}{2}\int_0^1 \left|v'(x)\right|^2 dx - \int_0^1 f(x)v(x)\,dx$$

and consider the problem of finding $u \in \mathcal{V}$ such that

$$F(u) \leqslant F(v) \qquad \forall\, v \in \mathcal{V}\,. \tag{M}$$

In the following, we prove that "if $u \in \mathcal{V}$, then u satisfies (V) if and only u satisfies (M). Moreover, (V) has at most one solution in $\mathcal{V}$."

Before proceeding, for the purpose of simplifying the notation, we define an "inner product" on $\mathcal{V}$:

$$\langle f, g\rangle \equiv \int_0^1 f(x)g(x)\,dx.$$

Using this notation, (V) can be rewritten as

$$\langle u', v'\rangle - \langle f, v\rangle = 0 \qquad \forall\, v \in \mathcal{V}\,.$$

§6.7 Finite Element Method (FEM)

Let $F : \mathcal{V} \to \mathbb{R}$ be defined by

$$F(v) = \frac{1}{2}\int_0^1 \left|v'(x)\right|^2 dx - \int_0^1 f(x)v(x)\,dx$$

and consider the problem of finding $u \in \mathcal{V}$ such that

$$F(u) \leqslant F(v) \qquad \forall\, v \in \mathcal{V}\,. \tag{M}$$

In the following, we prove that "if $u \in \mathcal{V}$, then u satisfies (V) if and only u satisfies (M). Moreover, (V) has at most one solution in $\mathcal{V}$."

Before proceeding, for the purpose of simplifying the notation, we define an "inner product" on $\mathcal{V}$:

$$\langle f, g\rangle \equiv \int_0^1 f(x)g(x)\,dx.$$

Using this notation, (V) can be rewritten as

$$\langle u', v'\rangle - \langle f, v\rangle = 0 \qquad \forall\, v \in \mathcal{V}\,.$$

§6.7 Finite Element Method (FEM)

Let $F : \mathcal{V} \to \mathbb{R}$ be defined by

$$F(v) = \frac{1}{2}\int_0^1 \left|v'(x)\right|^2 dx - \int_0^1 f(x)v(x)\,dx$$

and consider the problem of finding $u \in \mathcal{V}$ such that

$$F(u) \leqslant F(v) \qquad \forall\, v \in \mathcal{V}\,. \tag{M}$$

In the following, we prove that "if $u \in \mathcal{V}$, then u satisfies (V) if and only u satisfies (M). Moreover, (V) has at most one solution in $\mathcal{V}$."

Before proceeding, for the purpose of simplifying the notation, we define an "inner product" on $\mathcal{V}$:

$$\langle f, g\rangle \equiv \int_0^1 f(x)g(x)\,dx.$$

Using this notation, (V) can be rewritten as

$$\langle u', v'\rangle - \langle f, v\rangle = 0 \qquad \forall\, v \in \mathcal{V}\,.$$

§6.7 Finite Element Method (FEM)

1. (V) $\Rightarrow$ (M): Let u be a solution of problem (V). Let $v \in \mathcal{V}$ and $w = v - u \in \mathcal{V}$. Then $v = u + w$ and

$$\begin{aligned} F(v) = F(u+w) &= \tfrac{1}{2}\langle (u+w)', (u+w)' \rangle - \langle f, u+w \rangle \\ &= \tfrac{1}{2}\langle u', u' \rangle + \langle u', w' \rangle + \tfrac{1}{2}\langle w', w' \rangle - \langle f, u \rangle - \langle f, w \rangle \\ &= \tfrac{1}{2}\langle u', u' \rangle + \tfrac{1}{2}\langle w', w' \rangle - \langle f, u \rangle \geqslant \tfrac{1}{2}\langle u', u' \rangle - \langle f, u \rangle = F(u). \end{aligned}$$

2. (M) $\Rightarrow$ (V): Let u be a solution of problem (M). Note that if $v \in \mathcal{V}$, then $u + \varepsilon v \in \mathcal{V}$ for all $\varepsilon > 0$. Therefore,

$$F(u) \leqslant F(u + \varepsilon v) \qquad \forall\, v \in \mathcal{V} \text{ and } \varepsilon \in \mathbb{R}. \qquad (\diamond)$$

Define $g(\varepsilon) = F(u + \varepsilon v)$. Then $(\diamond)$ implies that g attains its global minimum at $\varepsilon = 0$. Therefore,

$$0 = g'(0) = \langle u', v' \rangle - \langle f, v \rangle$$

§6.7 Finite Element Method (FEM)

1. (V) $\Rightarrow$ (M): Let u be a solution of problem (V). Let $v \in \mathcal{V}$ and $w = v - u \in \mathcal{V}$. Then $v = u + w$ and

$$\begin{aligned} F(v) = F(u+w) &= \tfrac{1}{2}\langle (u+w)', (u+w)' \rangle - \langle f, u+w \rangle \\ &= \tfrac{1}{2}\langle u', u' \rangle + \langle u', w' \rangle + \tfrac{1}{2}\langle w', w' \rangle - \langle f, u \rangle - \langle f, w \rangle \\ &= \tfrac{1}{2}\langle u', u' \rangle + \tfrac{1}{2}\langle w', w' \rangle - \langle f, u \rangle \geqslant \tfrac{1}{2}\langle u', u' \rangle - \langle f, u \rangle = F(u). \end{aligned}$$

2. (M) $\Rightarrow$ (V): Let u be a solution of problem (M). Note that if $v \in \mathcal{V}$, then $u + \varepsilon v \in \mathcal{V}$ for all $\varepsilon > 0$. Therefore,

$$F(u) \leqslant F(u + \varepsilon v) \qquad \forall\, v \in \mathcal{V} \text{ and } \varepsilon \in \mathbb{R}. \qquad (\diamond)$$

Define $g(\varepsilon) = F(u + \varepsilon v)$. Then $(\diamond)$ implies that g attains its global minimum at $\varepsilon = 0$. Therefore,

$$0 = g'(0) = \langle u', v' \rangle - \langle f, v \rangle$$

§6.7 Finite Element Method (FEM)

3. (V) has at most one solution in $\mathcal{V}$: Suppose that $u_1, u_2 \in \mathcal{V}$ both satisfy (V). Then

$$\langle u_1', v' \rangle - \langle f, v \rangle = \langle u_2', v' \rangle - \langle f, v \rangle = 0 \qquad \forall\, v \in \mathcal{V}\,.$$

Note that $u_1 - u_2 \in \mathcal{V}$; thus letting $v = u_1 - u_2$ in the equality above we find that

$$\langle u_1', (u_1 - u_2)' \rangle - \langle f, u_1 - u_2 \rangle = 0\,,$$
$$\langle u_2', (u_1 - u_2)' \rangle - \langle f, u_1 - u_2 \rangle = 0\,,$$

Therefore, forming the difference of the two equations above shows that

$$\int_0^1 \big|\big[u_1(x) - u_2(x)\big]'\big|^2\, dx = \langle (u_1 - u_2)', (u_1 - u_2)' \rangle = 0\,.$$

The identity above implies that $\big[u_1(x) - u_2(x)\big]' = 0$; thus $u_1(x) - u_2(x) = 0$ for all $x \in [0, 1]$ which concludes that $u_1 = u_2$.

§6.7 Finite Element Method (FEM)

FEM for the model problem with piecewise linear functions:

Construct a finite-dimensional space $\mathcal{V}_h$ (finite element space) as follows: let $0 = x_0 < x_2 < \cdots < x_M < x_{M+1} = 1$ be a partition of $[0, 1]$, and set

- $I_j := [x_{j-1}, x_j], \quad j = 1, 2, \cdots, M+1.$
- $h_j := x_j - x_{j-1}, \quad j = 1, 2, \cdots, M+1.$
- $h := \max\limits_{j=1,2,\cdots,M+1} h_j$, a measure of how fine the partition is.

Define

$$\mathcal{V}_h := \Big\{ v_h \in \mathcal{V} \;\Big|\; v_h \text{ is linear on each subinterval } I_j, \; v_h(0) = v_h(1) = 0 \Big\}.$$

Notice that $\mathcal{V}_h \subseteq \mathcal{V}$.

§6.7 Finite Element Method (FEM)

For $j = 1, 2, \cdots, M$, we define $\varphi_j \in \mathcal{V}_h$ by

$$\varphi_j(x_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Then every $f \in \mathcal{V}_h$ can be expressed as

$$f(x) = \sum_{j=1}^{M} f(x_j)\varphi_j(x);$$

thus $\{\varphi_j\}_{j=1}^{M}$ is a basis of the finite-dimensional vector space $\mathcal{V}_h$.

Two numerical methods for approximating the solution of (D):

1. Ritz method:

   Find $u_h \in \mathcal{V}_h$ such that $F(u_h) \leqslant F(v_h)$ for all $v_h \in \mathcal{V}_h$. $(\mathrm{M_h})$

2. Galerkin method (finite element method):

   Find $u_h \in \mathcal{V}_h$ such that $\langle u_h', v_h' \rangle = \langle f, v_h \rangle$ for all $v_h \in \mathcal{V}_h$. $(\mathrm{V_h})$

Similar to the proof of (M) $\Leftrightarrow$ (V), one can show that $(\mathrm{M_h}) \Leftrightarrow (\mathrm{V_h})$.

§6.7 Finite Element Method (FEM)

For $j = 1, 2, \cdots, M$, we define $\varphi_j \in \mathcal{V}_h$ by

$$\varphi_j(x_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Then every $f \in \mathcal{V}_h$ can be expressed as

$$f(x) = \sum_{j=1}^{M} f(x_j)\varphi_j(x);$$

thus $\{\varphi_j\}_{j=1}^{M}$ is a basis of the finite-dimensional vector space $\mathcal{V}_h$.

Two numerical methods for approximating the solution of (D):

1. Ritz method:

   Find $u_h \in \mathcal{V}_h$ such that $F(u_h) \leqslant F(v_h)$ for all $v_h \in \mathcal{V}_h$. $(\mathrm{M_h})$

2. Galerkin method (finite element method):

   Find $u_h \in \mathcal{V}_h$ such that $\langle u_h', v_h' \rangle = \langle f, v_h \rangle$ for all $v_h \in \mathcal{V}_h$. $(\mathrm{V_h})$

Similar to the proof of (M) $\Leftrightarrow$ (V), one can show that $(\mathrm{M_h}) \Leftrightarrow (\mathrm{V_h})$.

§6.7 Finite Element Method (FEM)

Next we focus on solving $(\mathrm{V_h})$. We first claim that

Find $u_h \in \mathcal{V}_h$ such that $\langle u_h', v_h' \rangle = \langle f, v_h \rangle$ for all $v_h \in \mathcal{V}_h$.

$\Leftrightarrow$ Find $u_h \in \mathcal{V}_h$ such that $\langle u_h', \varphi_j' \rangle = \langle f, \varphi_j \rangle$ for all $1 \leqslant j \leqslant M$.

Proof.

($\Rightarrow$) Trivial! (Choosing test function $v = \varphi_j$).

($\Leftarrow$) For any $v_h \in \mathcal{V}_h$, $v_h = \sum_{j=1}^{M} \eta_j \varphi_j$ for some $(\eta_1, \eta_2 \cdots, \eta_M) \in \mathbb{R}^M$.

Therefore,

$$\langle u_h', v_h' \rangle = \Big\langle u_h', \sum_{j=1}^{M} \eta_j \varphi_j' \Big\rangle = \sum_{j=1}^{M} \eta_j \langle u_h', \varphi_j' \rangle$$

$$= \sum_{j=1}^{M} \eta_j \langle f, \varphi_j \rangle = \Big\langle f, \sum_{j=1}^{M} \eta_j \varphi_j \Big\rangle = \langle f, v_h \rangle. \qquad \square$$

§6.7 Finite Element Method (FEM)

Note that a solution $u_h \in \mathcal{V}_h$, if exists, can be written as $u_h(x) = \sum_{j=1}^{M} \xi_j \varphi_j(x)$ for some $(\xi_1, \xi_2, \cdots, \xi_M) \in \mathbb{R}^M$. Therefore,

$$\text{Find } u_h \in \mathcal{V}_h \text{ such that } \langle u_h', \varphi_j' \rangle = \langle f, \varphi_j \rangle \text{ for all } 1 \leqslant j \leqslant M$$

$$\Leftrightarrow \langle \sum_{k=1}^{M} \xi_k \varphi_k', \varphi_j' \rangle = \langle f, \varphi_j \rangle \text{ for all } 1 \leqslant j \leqslant M$$

$$\Leftrightarrow \sum_{k=1}^{M} \langle \varphi_k', \varphi_j' \rangle \xi_k = \langle f, \varphi_j \rangle \text{ for all } 1 \leqslant j \leqslant M$$

$$\Leftrightarrow A\boldsymbol{\xi} = \boldsymbol{b} \text{ for some matrix } M \times M \text{ matrix } A.$$

Here, the matrix $A = [a_{ij}]_{M\times M}$ is defined by $a_{ij} = \langle \varphi_i', \varphi_j' \rangle$ and is called the **stiffness matrix**, while the vector $\boldsymbol{b} = [b_i]_{M\times 1}$ is defined by $b_i = \langle f, \varphi_i \rangle$ and is called the **load vector**.

§6.7 Finite Element Method (FEM)

$$A\xi = \boldsymbol{b}$$

$$\Leftrightarrow \begin{bmatrix} \langle \varphi_1', \varphi_1' \rangle & \langle \varphi_2', \varphi_1' \rangle & \cdots & \langle \varphi_M', \varphi_1' \rangle \\ \langle \varphi_1', \varphi_2' \rangle & \langle \varphi_2', \varphi_2' \rangle & \cdots & \langle \varphi_M', \varphi_2' \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \varphi_1', \varphi_M' \rangle & \langle \varphi_2', \varphi_M' \rangle & \cdots & \langle \varphi_M', \varphi_M' \rangle \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_M \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

Remark:

1. A is symmetric.
2. A is tri-diagonal: $\langle \varphi_i', \varphi_j' \rangle = 0$ if $|i-j| > 1$.
3. A is positive-definitie: if $\boldsymbol{\eta} \neq \mathbf{0}$, then

$$\boldsymbol{\eta}^{\mathrm{T}} A \boldsymbol{\eta} = \Big\langle \sum_{j=1}^{M} \eta_i \varphi_i', \sum_{j=1}^{M} \eta_j \varphi_j' \Big\rangle = \int_0^1 \Big| \Big(\sum_{j=1}^{M} \eta_j \varphi_j(x) \Big)' \Big|^2 \, dx > 0.$$

4. Since A is SPD, $A\xi = \boldsymbol{b}$ has a unique solution.

§6.7 Finite Element Method (FEM)

For $j = 1, 2, \cdots, M$, we have

$$\begin{aligned}\langle \varphi_j', \varphi_j' \rangle &= \int_{x_{j-1}}^{x_j} \varphi_j'(x)^2 dx + \int_{x_j}^{x_{j+1}} \varphi_j'(x)^2 dx \\ &= \int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx + \int_{x_j}^{x_{j+1}} \frac{1}{h_{j+1}^2} dx = \frac{1}{h_j} + \frac{1}{h_{j+1}},\end{aligned}$$

$$\langle \varphi_j', \varphi_{j-1}' \rangle = \langle \varphi_{j-1}', \varphi_j' \rangle = -\int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx = -\frac{1}{h_j}.$$

For uniform partition: $h_j = h = \frac{1-0}{M+1}$. Then $A\xi = \boldsymbol{b}$ becomes

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \vdots \\ \xi_M \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

§6.7 Finite Element Method (FEM)

For $j = 1, 2, \cdots, M$, we have

$$\begin{aligned}\langle \varphi_j', \varphi_j' \rangle &= \int_{x_{j-1}}^{x_j} \varphi_j'(x)^2 dx + \int_{x_j}^{x_{j+1}} \varphi_j'(x)^2 dx \\ &= \int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx + \int_{x_j}^{x_{j+1}} \frac{1}{h_{j+1}^2} dx = \frac{1}{h_j} + \frac{1}{h_{j+1}},\end{aligned}$$

$$\langle \varphi_j', \varphi_{j-1}' \rangle = \langle \varphi_{j-1}', \varphi_j' \rangle = -\int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx = -\frac{1}{h_j}.$$

For uniform partition: $h_j = h = \dfrac{1-0}{M+1}$. Then $A\boldsymbol{\xi} = \boldsymbol{b}$ becomes

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \vdots \\ \xi_M \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

§6.7 Finite Element Method (FEM)

For $j = 1, 2, \cdots, M$, we have

$$\begin{aligned}\langle \varphi_j', \varphi_j' \rangle &= \int_{x_{j-1}}^{x_j} \varphi_j'(x)^2 dx + \int_{x_j}^{x_{j+1}} \varphi_j'(x)^2 dx \\ &= \int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx + \int_{x_j}^{x_{j+1}} \frac{1}{h_{j+1}^2} dx = \frac{1}{h_j} + \frac{1}{h_{j+1}},\end{aligned}$$

$$\langle \varphi_j', \varphi_{j-1}' \rangle = \langle \varphi_{j-1}', \varphi_j' \rangle = -\int_{x_{j-1}}^{x_j} \frac{1}{h_j^2} dx = -\frac{1}{h_j}.$$

For uniform partition: $h_j = h = \dfrac{1-0}{M+1}$. Then $A\boldsymbol{\xi} = \boldsymbol{b}$ becomes

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ \vdots \\ u_h(x_M) \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

§6.7 Finite Element Method (FEM)

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ \vdots \\ u_h(x_M) \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

Taking into account that $u_h(x_0) = u_h(x_{M+1}) = 0$, the system above shows that

$$-\frac{1}{h^2}\left[u_h(x_{j-1}) - 2u_h(x_j) + u_h(x_{j+1})\right] = \frac{1}{h}\langle f, \varphi_j \rangle \qquad \forall\, 1 \leqslant j \leqslant M.$$

Since we expect that u_h approximates the solution u, the equality above implies that

$$-\frac{1}{h^2}\left[u(x_j - h) - 2u(x_j) + u(x_j + h)\right] \approx \frac{1}{h}\int_{x_j - h}^{x_j + h} f(x)\varphi_j(x)\, dx \quad \forall\, 1 \leqslant j \leqslant M.$$

§6.7 Finite Element Method (FEM)

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ \vdots \\ u_h(x_M) \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \vdots \\ \langle f, \varphi_M \rangle \end{bmatrix}.$$

Taking into account that $u_h(x_0) = u_h(x_{M+1}) = 0$, the system above shows that

$$-\frac{1}{h^2}\big[u_h(x_{j-1}) - 2u_h(x_j) + u_h(x_{j+1})\big] = \frac{1}{h}\langle f, \varphi_j \rangle \qquad \forall\, 1 \leqslant j \leqslant M.$$

Since we expect that u_h approximates the solution u, the equality above implies that

$$-\frac{1}{h^2}\big[u(x_j-h)-2u(x_j)+u(x_j+h)\big] \approx \frac{1}{h}\int_{x_j-h}^{x_j+h} f(x)\varphi_j(x)\,dx \quad \forall\, 1 \leqslant j \leqslant M.$$

§6.7 Finite Element Method (FEM)

Recall that if $u \in C^4([x_j - h, x_j + h])$,

$$u''(x_j) = \frac{1}{h^2}\big[u(x_j + h) - 2u(x_j) + u(x_j - h)\big] - \frac{1}{12}h^2 u^{(4)}(\xi_j)$$

for some $\xi_j \in (x_j - h, x_j + h)$. Moreover, if $f \in C^2([x_j - h, x_j + h])$, by Taylor's Theorem there exists $\eta_j \in (x_j - h, x_j + h)$ such that

$$f(x) = f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2;$$

thus if $f \in C^2([x_j - h, x_j + h])$, the fact that $\frac{1}{h}\int_{x_j-h}^{x_j+h} \varphi_j(x)\,dx = 1$ shows

$$\begin{aligned}
&\frac{1}{h}\int_{x_j-h}^{x_j+h} f(x)\varphi_j(x)\,dx \\
&\quad = \frac{1}{h}\int_{x_j-h}^{x_j+h} \left[f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2\right]\varphi(x)\,dx \\
&\quad = f(x_j) + \frac{1}{2h}\int_{x_j-h}^{x_j+h} f''(\eta_j)(x - x_j)^2\varphi(x)\,dx
\end{aligned}$$

§6.7 Finite Element Method (FEM)

Recall that if $u \in C^4([x_j - h, x_j + h])$,

$$u''(x_j) = \frac{1}{h^2}\big[u(x_j + h) - 2u(x_j) + u(x_j - h)\big] - \frac{1}{12}h^2 u^{(4)}(\xi_j)$$

for some $\xi_j \in (x_j - h, x_j + h)$. Moreover, if $f \in C^2([x_j - h, x_j + h])$, by Taylor's Theorem there exists $\eta_j \in (x_j - h, x_j + h)$ such that

$$f(x) = f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2;$$

thus if $f \in C^2([x_j - h, x_j + h])$, the fact that $\frac{1}{h}\int_{x_j-h}^{x_j+h} \varphi_j(x)\,dx = 1$ shows

$$\begin{aligned}
&\frac{1}{h}\int_{x_j-h}^{x_j+h} f(x)\varphi_j(x)\,dx \\
&\quad = \frac{1}{h}\int_{x_j-h}^{x_j+h} \left[f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2\right]\varphi(x)\,dx \\
&\quad = f(x_j) + \frac{1}{2h}f''(\zeta_j)\int_{x_j-h}^{x_j+h} (x - x_j)^2\varphi(x)\,dx
\end{aligned}$$

§6.7 Finite Element Method (FEM)

Recall that if $u \in C^4([x_j - h, x_j + h])$,

$$u''(x_j) = \frac{1}{h^2}\big[u(x_j + h) - 2u(x_j) + u(x_j - h)\big] - \frac{1}{12}h^2 u^{(4)}(\xi_j)$$

for some $\xi_j \in (x_j - h, x_j + h)$. Moreover, if $f \in C^2([x_j - h, x_j + h])$, by Taylor's Theorem there exists $\eta_j \in (x_j - h, x_j + h)$ such that

$$f(x) = f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2;$$

thus if $f \in C^2([x_j - h, x_j + h])$, the fact that $\frac{1}{h}\int_{x_j-h}^{x_j+h} \varphi_j(x)\,dx = 1$ shows

$$\begin{aligned}
&\frac{1}{h}\int_{x_j-h}^{x_j+h} f(x)\varphi_j(x)\,dx \\
&\quad = \frac{1}{h}\int_{x_j-h}^{x_j+h} \left[f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2\right]\varphi(x)\,dx \\
&\quad = f(x_j) + \frac{1}{2h}f''(\zeta_j)\cdot\frac{h^3}{2}
\end{aligned}$$

§6.7 Finite Element Method (FEM)

Recall that if $u \in C^4([x_j - h, x_j + h])$,

$$u''(x_j) = \frac{1}{h^2}\big[u(x_j + h) - 2u(x_j) + u(x_j - h)\big] - \frac{1}{12}h^2 u^{(4)}(\xi_j)$$

for some $\xi_j \in (x_j - h, x_j + h)$. Moreover, if $f \in C^2([x_j - h, x_j + h])$, by Taylor's Theorem there exists $\eta_j \in (x_j - h, x_j + h)$ such that

$$f(x) = f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2;$$

thus if $f \in C^2([x_j - h, x_j + h])$, the fact that $\frac{1}{h}\int_{x_j-h}^{x_j+h} \varphi_j(x)\,dx = 1$ shows

$$\begin{aligned}
&\frac{1}{h}\int_{x_j-h}^{x_j+h} f(x)\varphi_j(x)\,dx \\
&\quad = \frac{1}{h}\int_{x_j-h}^{x_j+h} \left[f(x_j) + f'(x_j)(x - x_j) + \frac{f''(\eta_j)}{2}(x - x_j)^2\right]\varphi(x)\,dx \\
&\quad = f(x_j) + \frac{h^2}{4}f''(\zeta_j) \text{ for some } \zeta_j \in (x_j - h, x_j + h).
\end{aligned}$$

§6.7 Finite Element Method (FEM)

Therefore, we conclude that if $f \in C^2([0,1])$ (so that $u \in C^4([0,1])$), the finite element method introduced above indeed is a second order approximation of the equation

$$-u''(x) = f(x)$$

at each node x_j.

§6.7 Finite Element Method (FEM)

FDM different from the FEM methods in few aspects:

1. In the FDM methods, the discretization of the domain is done as a set of nodes at which the results are determined, while in the FEM method the results are known in every point of the domain as the approximation is done with functions defined on small triangular (or quadrilateral) areas in 2D.
2. Because of that, the algorithms used in FDM require generally less computational power to solve the equations, but it results also in less "refines" results (only at nodes).

§6.7 Finite Element Method (FEM)

The main difference between FEM and FDM (in simple terms):

1. FDM is an older method than FEM that requires less computational power but is also less accurate in some cases where higher-order accuracy is required.
2. FEM permits to get a higher order of accuracy, but requires more computational power and is also more exigent on the quality of the mesh.

§6.7 Finite Element Method (FEM)

Computer project

Consider the following one-dimensional convection-diffusion problem:

$$\begin{cases} -\varepsilon u''(x) + u'(x) = 0 \quad \text{for } x \in (0,1), \\ u(0) = 1, \ u(1) = 0. \end{cases} \qquad (*)$$

Write the computer codes for numerical solution of problem $(*)$ by using the finite difference methods on the uniform mesh of $[0,1]$ with mesh size h:

1. Replace $u''(x_i) \approx \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2}$ and $u'(x_i) \approx \frac{U_{i+1} - U_{i-1}}{2h}$ and consider $(\varepsilon, h) = (0.01, 0.1)$, $(\varepsilon, h) = (0.01, 0.01)$. Plot u_h.
2. Replace $u''(x_i) \approx \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2}$ and $u'(x_i) \approx \frac{U_i - U_{i-1}}{h}$ (upwinding) and consider $(\varepsilon, h) = (0.01, 0.1)$, $(\varepsilon, h) = (0.01, 0.01)$. Plot u_h.