Numerical Analysis

Chap 1. Error

目标：算出来是不是准的?

1. Components

Truncation error　截断误差

Round-off error　舍入误差

2. Significant Digits

3. Convergence & Stability

Chap 2. Solution of Equations in One Variable

目标：给定初值求解单变量函数$f(x)=0$的根。

1. Bisection Method

$$p_1 = \frac{a_1 + b_1}{2}$$

2. Fixed-point Iteration

Fixed-point Th. 存在收敛的不动点，若导函数的绝对值可被 1 约束住。

3. Newton's Method

二阶收敛，若根处导函数不为 0，否则一阶收敛。

$$p_n = p_n - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

4. Secant Method

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

牛顿法和割线法都不保证根在迭代区间内，因此有了虚位法。

5. False Position Method

虚位法在切线法上增加了一步条件判断，确保迭代以夹逼方式进行。但虚位法并不见得好。

6. Modified Newton's Method

改进牛顿法保证了在导函数为 0 的点上的收敛速度。

$$p_n = p_{n-1} - \frac{f(p_{n-1})f'(p_{n-1})}{\left(f'(p_{n-1})^2 - f(p_{n-1})f''(p_{n-1})\right)}$$

7. Order of Convergence α

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$$

8. Accelerating Convergence

类似于 Richardson 外推法的想法，用多项低阶去产生高阶精度而避免向下迭代。

Forward difference　前差Δ

$$\Delta^k p_n = \Delta^{k-1} p_{n+1} - \Delta^{k-1} p_n$$
$$\Delta p_n = p_{n+1} - p_n$$

Backward difference　后差∇类似

Centered difference　中差δ

Aitken's Δ^2 method

$$\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$$

Steffensen's Method

每次向前迭代两步，用

$$\hat{p} = p_0 - \frac{(p_1 - p_0)^2}{p_2 - 2p_1 + p_0}$$

去产生下一步的p_0

Chap 3. Interpolation & Polynomial Approximation

目标：使用多项式生成函数散点之间的值。

非按段的多项式插值都可以纳入到密切曲线的定义中。

Weierstrass Approx. Th. 连续函数一定有足够好的多项式近似。

1. Lagrange Polynomial

不等式的每个点都在被插值的点处，从$n+1$个点到一个n阶多项式。

$$P(x) = \sum_{k=0}^{n} f(x_k) L_{n,k}(x)$$
$$L_{n,k}(x) = \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}$$
$$L_{n,k}(x) = \begin{cases} 0, & x = x_i, i \neq k \\ 1, & x = x_k \end{cases}$$

误差估计：

$$f(x) = P(x) + \frac{f^{(n+1)}\big(\xi(x)\big)}{(n+1)!}(x - x_0)\cdots(x - x_n)$$

Lagrange 多项式是唯一的。

2. Neville's Method

使用 Lagrange 基生成的多项式没有办法随时停止，Neville's method 解决了这个问题。Neville's method 使用了一种可以控制的方式去生成 Lagrange 多项式，且和与 Lagrange 基表示的多项式具有不同的形式。

Neville's method 基于定理

$$P(x) = \frac{(x - x_j)P_{\hat{j}}(x) - (x - x_i)P_{\hat{\imath}}(x)}{x_i - x_j}$$

Neville 多项式的下标应当是所有参与点的序号，为了便于表示，常常采用如下符号：

$$Q_{i,j} = P_{i-j, i-j+1, \cdots, i-1, i}$$
$$Q_{i,j}(x) = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$$

3. Newton's Divided Differences

牛顿均差法同样生成 Lagrange 不等式，具有和 Neville's method 相类似的形式。

$$f[x_0,\cdots,x_k]=\frac{f[x_0,\cdots,x_{k-1}]-f[x_1,\cdots,x_k]}{x_0-x_k}$$
$$P_n(x)=f[x_0]+f[x_0,x_1](x-x_0)+a_2(x-x_0)(x-x_1)$$
$$+\cdots+a_n(x-x_0)\cdots(x-x_{n-1})$$
$$f[x_i]=f(x_i)$$

4. Hermite Interpolation

密切曲线 osculating polynomials 是 Taylor 多项式和 Lagrange 多项式的推广。

$$\frac{\mathrm{d}^k P(x_i)}{\mathrm{d}x^k}=\frac{\mathrm{d}^k f(x_i)}{\mathrm{d}x^k},\qquad i=0,1,\cdots,n;k=0,1,\cdots,m_i$$

Taylor 多项式即只使用的一个点，去向高阶展开。

Lagrange 多项式即在每个点处都只保证 0 阶导数贴合。

Hermite 多项式是保证每个点处的 1 阶贴合，生成的多项式至多是$2n+1$阶的。

$$H_{2n+1}(x)=\sum_{j=0}^{n}f(x_j)H_{n,j}(x)+\sum_{j=0}^{n}f'(x_j)\hat{H}_{n,j}(x)$$
$$H_{n,j}(x)=[1-2(x-x_j)L'_{n,j}(x_j)]L^2_{n,j}(x)$$
$$\hat{H}_{n,j}(x)=(x-x_j)L^2_{n,j}(x)$$

C^{2n+2}上的误差估计

$$f(x)=H_{2n+1}(x)+\frac{(x-x_0)^2\cdots(x-x_n)^2}{(2n+2)!}f^{(2n+2)}\big(\xi(x)\big)$$

5. Cubic Spline Interpolation

三次样条是一种按段插值的方式，满足：

(a) $S_j(x)$插值区间$[x_j,x_{j+1}],j=0,1,\cdots,n-1$
(b) 区间端点处贴合
(c) 区间端点处重合
(d) 区间端点处一阶导数重合
(e) 区间端点处二阶导出重合
(f) 边界导数值满足：
  i. Natural boundary: 0
  ii. Clamped boundary: 给定导数值

$$S_j(x)=a_j+b_j(x-x_j)+c_j(x-x_j)^2+d_j(x-x_j)^3$$
$$h_j=x_{j+1}-x_j$$
$$a_j=f(x_j)$$
$$b_j=(a_{j+1}-a_j)/h_j-h_j(c_{j-1}+2c_j)/3$$
$$d_j=(c_{j+1}-c_j)/(3h_j)$$

求解c_j的矩阵参见 3.5 (149)

Chap 4. Numerical Differentiation & Integration

1. Numerical Differentiation

差分

$$f'(x_0)=\frac{f(x_0+h)-f(x_0)}{h}-\frac{h}{2}f''(\xi)$$

利用 Lagrange 多项式

$$f'(x)=\sum_{k=0}^{n}f(x_k)L'_k(x_j)+\frac{f^{(n+1)}\big(\xi(x_j)\big)}{(n+1)!}\prod_{k=0,k\neq j}^{n}(x_j-x_k)$$

二阶中点

$$f''(x_0)=\frac{1}{h^2}\big(f(x_0-h)-2f(x_0)+f(x_0+h)\big)-\frac{h^2}{12}f^{(4)}(\xi)$$

2. Richardson's Extrapolation (182)

Richardson 外推法的思想是，对余项形式已知且为多项式的函数，带入不同倍率的因变量，得到两个近似，利用作差消元，将某高阶项消去，从而留下低阶的计算项和高阶的余项，实现减少计算量加快收敛的效果。

3. Numerical Integration

不按段，以下几种方法其一般形式是 Newton-Cotes 公式

Trapezoidal Rule　梯形法

$$\int_a^b f(x)\mathrm{d}x=\frac{h}{2}\big(f(x_0)+f(x_1)\big)-\frac{h^3}{12}f''(\xi)$$

Simpson's Rule　　抛物线法

$$\int_{x_0}^{x_2} f(x)\mathrm{d}x=\frac{h}{3}\big(f(x_0)+4f(x_1)+f(x_2)\big)-\frac{h^5}{90}f^{(4)}(\xi)$$

精确度的衡量：degree of accuracy/precision，即保证数值积分是准确的被积函数最高阶数。

Closed Newton-Cotes Formulas

闭合 Newton-Cotes 使用边界和内部的若干点做插值多项式并积分。

$$\int_a^b f(x)\mathrm{d}x=\sum_{i=0}^{n}a_i f(x_i)+o$$
$$a_i=\int_{x_0}^{x_n}L_i(x)\mathrm{d}x=\int_{x_0}^{x_n}\prod_{j=0,j\neq i}^{n}\frac{x-x_j}{x_i-x_j}\mathrm{d}x$$

n为偶，$f\in C^{n+2}$余项

$$\frac{h^{n+3}f^{(n+2)}(\xi)}{(n+2)!}\int_0^n t^2(t-1)\cdots(t-n)\mathrm{d}t$$

n为奇，$f\in C^{n+1}$余项

$$\frac{h^{n+2}f^{(n+1)}(\xi)}{(n+1)!}\int_0^n t(t-1)\cdots(t-n)\mathrm{d}t$$

Open Newton-Cotes Formulas

不包含边界点，其余没有太多不同。

4. 按段方法

5. Romberg Integration (217)

利用 Richardson 外推法，首先生成一系列按段梯形法的近似，再利用迭代式生成更精确的结果。（这一做法类似于 Neville 多项式的生成）

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1}-1}(R_{k,j-1} - R_{k-1,j-1}),$$
$$k = j, j+1, \cdots$$

6. Adaptive Quadrature Methods (220)

自适应方法将误差二分，在每个区间估计相邻两次迭代的误差，当达到可接受范围时，不再迭代；否则继续二分，均分误差，从而实现有误差控制的自适应积分。

7. Gaussian Quadrature (232)

Gauss 积分使用中间点做插值，最终将积分式写为中间点加权求和的形式（这一思想类似于 Runge-Kutta）：

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=1}^{n} c_i f(x_i)$$

生成c_i的思想是，利用正交多项式求解函数在区间上的最佳近似，对多项式求积分，再将所得定积分转变为函数值的线性组合，正交多项式在这里具有足够好的性质。对一般的定积分，通常的做法是将区间映射到$[-1,1]$使用其上的恒权重正交多项式，即 Legendre 多项式，某一基根处的值，作为x_i，其对应的系数：

$$c_i = \int_{-1}^{1} \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}\mathrm{d}x$$

Chap 5. Initial-Value Problems for ODE

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y), \qquad a \le t \le b, y(a) = \alpha$$

这章中几乎所有的单步迭代方法都是广义 Runge-Kutta 方法的特例。

1. 基本概念

Lipschitz 条件：$f(t,y)$对定义域上的所有点都对y一致连续（函数值差分可以被限制住）

Well-posed 适定性：初值问题存在唯一的解。对于任何不超过某一限度的扰动，被扰动的初值问题的解和原来的解之间的偏差，与这一扰动的限度同阶。

初值问题走的每一步所踩到的点称为 mesh point 节点。

Local truncation error 局部截断误差约定为，这一步按照真实趋势走和按照迭代产生的直线走，走出的差距，除以这一步的步长。

2. Euler's Method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + hf\big(t_i, y(t_i)\big)$$

余项，注意局部截断误差在教材中约定要除以步长：

$$\frac{h^2}{2}y''(\xi_i)$$

3. High-Order Taylor Method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i)$$

$$T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2}f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, w_i)$$

$$y(t_{i+1}) = w_{i+1} + \frac{h^{n+1}}{(n+1)!}f^{(n)}\big(\xi_i, y(\xi_i)\big)$$

4. Runge-Kutta Methods (281)

思想是，每步都在中间取若干点（偏移了$a_i h$处），增加在这个点之前已经求出的点对整体的贡献。

$w_0 = \alpha$
$w_{i+1} = w_i + h(\lambda_1 K_1 + \lambda_2 K_2 + \cdots + \lambda_m K_m)$
$K_1 = f(t_i, w_i)$
$K_2 = f(t_i + \alpha_2 h, w_i + \beta_{21} h K_1)$
$K_3 = f(t_i + \alpha_3 h, w_i + \beta_{31} h K_1 + \beta_{32} h K_2)$
$\cdots$
$K_m = f(t_i + \alpha_m h, w_i + \beta_{m1} h K_1 + \beta_{m2} h K_2 + \cdots + \beta_{m\ m-1} h K_{m-1})$

$$\sum_{j=1}^{m-1} \beta_{m,j} = \alpha_m, \qquad \sum_{i=1}^{m} \lambda_i = 1$$

$$\begin{array}{c|ccccc}
0 & & & & & \\
\alpha_2 & \beta_{21} & & & & \\
\alpha_3 & \beta_{31} & \beta_{32} & & & \\
\cdots & \cdots & \cdots & \ddots & & \\
\alpha_m & \beta_{m1} & \beta_{m2} & \cdots & \beta_{m,m-1} & \\
\hline
 & \lambda_1 & \lambda_2 & \cdots & \lambda_{m-1} & \lambda_m
\end{array}$$

Euler's Implicit

$$\begin{array}{c|c}
1 & 1 \\
\hline
 & 1
\end{array}$$

Midpoint

$$\begin{array}{c|c}
0 & \\
1/2 & 1/2 \\
\hline
 & 0
\end{array}$$

Modified Euler

$$\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & 1/2 & 1/2
\end{array}$$

RK4 (288)

5. 多步法 (5.6, 302)

下一步是前面若干步的线性组合。

$$
\begin{aligned}
w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\
+ h\Big(b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \cdots \\
+ b_0 f(t_{i+1-m}, w_{i+1-m})\Big)
\end{aligned}
$$

显式的含义是，$b_m = 0$

6. 高阶与线性系统 (5.7, 327)

7. 稳定性

Consistent 一致，即局部截断误差保证了解的趋势不会差太远

$$\lim_{h\to 0} \max_{1\le i\le N} |\tau_i(h)| = 0$$

Convergent 收敛，即保证了解不会差太远

$$\lim_{h\to 0} \max_{1\le i\le N} |w_i - y(t_i)| = 0$$

对多步法，一致条件为

$$\lim_{h\to 0} |\tau_i(h)| = 0$$
$$\lim_{h\to 0} |\alpha_i - y(t_i)| = 0$$

特征多项式

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - \cdots a_1\lambda - a_0 = 0$$

所有特征值模不超过 1，且为 1 的特征值都是单根，称为满足根条件 root condition

满足根条件且仅有$\lambda = 1$是模长为 1 的特征值时，称作强稳定 strongly stable

满足根条件且有不止一个模长为 1 的特征值时，称作弱稳定 weekly stable

不满足根条件的称为不稳定 unstable

Chap 6. Direct Methods for Solving Linear Systems

目标：求解线性系统。

1. Gaussian Elimination & Backward Substitution

2. Pivoting Strategies

Partial Pivoting

向下查找主元所在列绝对值最大的，交换行。

Scale Partial Pivoting

查找绝对值除以一行中最大值最大的，交换行。

Complete Pivoting

既交换行又交换列。

3. 矩阵分解 (6.5, 400)

$$O(n^3/3) \to O(2n^2)$$

LU 分解 (406)

正定阵：二次型将非零向量都映射到正值。

正定阵：Cholesky (418)

主对角占优阵：Crout (422)

Chap 7. Iterative Techniques in Matrix Algebra

目标：利用迭代方法求解线性系统。

1. 向量与矩阵范数

向量范数满足：

1. 正性：将所有向量映射到非负实数
2. 定性：只将$\mathbf{0}$映射到 0
3. 齐性：向量数乘可以拿出来变成实数乘法
4. 三角不等式

矩阵范数满足：

1. 正性：将所有矩阵映射到非负实数
2. 定性：只有全零阵被映射到 0
3. 齐性：矩阵数乘可以拿出来变成实数乘法
4. 三角不等式：$\|A+B\| \le \|A\| + \|B\|$
5. 一致性：$\|AB\| \le \|A\|\|B\|$

距离的定义是差的范数。

矩阵的自然范数使用向量的自然范数定义：

$$\|A\| = \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|$$

直观的说，就是矩阵可以将某范数的单位圆所能带到的这个范数意义下最远的地方。

矩阵的自然范数有如下性质：

$$\|A\boldsymbol{x}\| \le \|A\| \cdot \|\boldsymbol{x}\|$$

矩阵的 1-范数是绝对值最大的列和（列和范数）

矩阵的 2-范数是$A^{\mathrm{T}}A$最大特征值的平方根（谱范数）

矩阵的无穷范数是绝对值最大的行和（行和范数）

矩阵的谱半径定义为其最大特征值。

谱半径不大于矩阵的任一自然范数。

矩阵的收敛定义为其矩阵幂随次数增高趋于零矩阵。

矩阵的收敛性有如下等价条件：

- (i) 矩阵收敛
- (ii) 矩阵幂在某自然范数下收敛
- (iii) 矩阵幂在所有自然范数下收敛
- (iv) 矩阵的谱半径严格比 1 小
- (v) 矩阵幂作用在某向量上收敛

2. Jacobi Iterative

由

$$x_i = \sum_{j=1, j\ne i}^{n} \left(-\frac{a_{ij}x_j}{a_{ii}}\right) + \frac{b_i}{a_{ii}}$$

产生迭代

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(\sum_{j=1, j\ne i}^{n} \left(-a_{ij}x_j^{(k-1)}\right) + b_i \right)$$

3. Gauss-Seidel Method

由

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(-\sum_{j=1}^{i-1}\left(a_{ij}x_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}x_j^{(k-1)}\right) + b_i\right)$$

一般地，

$$\boldsymbol{x}^{(k)} = T\boldsymbol{x}^{(\boldsymbol{k-1})} + \boldsymbol{c}$$

Chap 8. Approximation Theory

目标：用简单的函数近似复杂的。

1. Discrete Least Squares Approximation

$$E = \min \sum_{i=1}^{m} w_i(P(x_i) - y_i)^2$$

$$P(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x)$$

$$\frac{\partial E}{\partial a_k} = 0 \to \sum_{j=0}^{n}\langle\varphi_k, \varphi_j\rangle a_j = \langle\varphi_k, f\rangle, \qquad k = 0, 1, \cdots, n$$

使用时：

$$(b_{ij} = \langle\varphi_i, \varphi_j\rangle)(a_0 \ \cdots \ a_n)^{\mathrm{T}} = (\langle\varphi_0, f\rangle \ \cdots \ \langle\varphi_n, f\rangle)^{\mathrm{T}}$$

$$\varphi_0(x) \equiv 1, \varphi_1(x) = x - B_1$$

$$\varphi_k(x) = (x - B_k)\varphi_{k-1}(x) - C_k\varphi_{k-2}(x)$$

$$B_k = \frac{\langle x\varphi_{k-1}, \varphi_{k-1}\rangle}{\langle\varphi_{k-1}, \varphi_{k-1}\rangle}, \qquad C_k = \frac{\langle x\varphi_{k-1}, \varphi_{k-2}\rangle}{\langle\varphi_{k-2}, \varphi_{k-2}\rangle}$$

$$a_k = \frac{\langle\varphi_k, y\rangle}{\langle\varphi_k, \varphi_k\rangle}$$

这一过程基于 Gram-Schidt 正交化，可以得到一组正交基。每个系数即在基向量上的投影除以基向量的模长。

2. Orthogonal Polynomials and Least Squares Approximation

$$E = \min \int_a^b w(x)\big(P(x) - f(x)\big)^2$$

Legendre 多项式：$(-1, 1)$，权函数恒1

3. Chebyshev Polynomials

$(-1, 1)$权函数为$w(x) = \frac{1}{\sqrt{1-x^2}}$，迫使靠近边界贴合得好。

$$T_n(x) = \cos(n \arccos x)$$

$$T_0(x) = 1, \qquad T_1(x) = x,$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Leading coefficient 最高项系数2^{n-1}

（前几项 Chebyshev 多项式参见 519）

Th. Chebyshev 多项式$T_n(x)$在$[-1, 1]$之间有n个单根。

$$\langle T_n, T_m\rangle = \begin{cases} \pi, & n = m = 0 \\ \dfrac{\pi}{2}, & n = m \neq 0 \end{cases}$$

4. Economization of Power Series (522)

$$\frac{1}{2^{n-1}} = \max_{x\in[-1,1]}\left|\tilde{T}_n(x)\right| \leq \max_{x\in[-1,1]}|P_n(x)|, \forall P_n(x) \in \widetilde{\Pi}_n$$

Chebyshev 总是最矮的那个 monic

降低多项式阶数，而导致的精度损失最小，利用到了 Chebyshev 的一个技巧，$P(x) = a_nx^n + \cdots + a_1x + a_0$是一个多项式近似：

$$\max_{[-1,1]}|f(x) - P_{n-1}(x)| \leq \max_{[-1,1]}|f(x) - P_n(x)| + \max_{[-1,1]}|Q_n(x)|$$

为了最小化精度损失，$Q_n(x)$必须是$a_n \cdot \frac{T_n(x)}{2^{n-1}}$

Chap 9. Approximating Eigenvalues

目标：求解特征值。

幂法：从一个初始特征向量开始迭代乘幂，每轮在无穷范数意义下做单位化，最终会被带到最大特征值下的特征向量。

反幂法：收敛到距离初值最近的特征值，或利用$\frac{\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$的构造，收敛到距离初始向量最近的特征向量。每轮迭代需要求解方程$(A - q\boldsymbol{I})\boldsymbol{y}^{(m)} = \boldsymbol{x}^{(m-1)}$，其中$q$是给定的初值。