created by wth

cause

Truncation：计算量有关

Round-off：bit 数相关

不动点法

收敛的充分条件：[a,b] 内所有 x 满足 |g'(x)| < k, (0 < k < 1)

收敛速度: $|p_{n+1} - p_n| <= k * |p_n - p_{n-1}|$, k 越小越快

牛顿法

用泰勒展开的前两项作为函数近似，求该近似函数的零点，在零点处再求函数近似，迭代。

$p \approx p_0 - \frac{f(p_0)}{f'(p_0)}$

牛顿法的 k 在 p 点处是 0，所以收敛非常快

迭代法误差分析

$lim \frac{p_{n+1}-p}{|p_n-p|^a} = \lambda$

pn 收敛于 p, a 越大，收敛速度越快

a = 1, linearly convergent

a = 2, quadratically convergent

$g'(p) \neq 0$ 则最少是 linear convergent，拉格朗日中值定理可退

$g'(p) = 0$ 时（如牛顿法）g 不等于 0 的最高阶导数阶数 a

牛顿法是二阶收敛的

quadratica newtom method

有重根（几重根都可以）的时候令 $\mu(x) = \frac{f(x)}{f'(x)}$

然后再做牛顿法 $g(x) = x - \frac{\mu(x)}{\mu'(x)}$

Aitken's Method (Steffensen's Method)

原理

$p_{n+2} = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$

$p_1 = g(p_0),\ p2 = g(p_1)$

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

$p_0 = p$

LU 分解

如果矩阵不需要行交换就能高斯消元成上三角矩阵则可以 LU 分解（中间有 0 的话 LU 分解会没法进行）

复杂度 $n^3/3$

1. L 对角线为 1，U 对角线为原始元素
2.
   1. 算 U 的第 i 行
   2. 算 L 的第 i 列
   3. goto 1

Pivoting Strategies

每次找一列中最大的元素，把最大元素所在的行交换到当前行

Complete Pivoting

到第 i 行时的时候在 i 右下角的矩阵里找最大的元素，然后通过交换行、交换列，换到 a_{ii}

Strictly Diagonally Dominant Matrix

对角线上的元素绝对值严格大于此行其他元素绝对值之和

Positive Define Matrix

正定矩阵所有顺序主子式 (左上角的前 k 行前 k 列构成的矩阵) 的行列式 >0

LU factorization of a positive define Matrix

正定矩阵可以分解成 $L * L^t$ 的形式，L 是下三角的 L 加上对角线替换为根号对角线

Crout Reduction of Tridiagonal LInear System

1. 先 LU 分解, $LUx = f$ 分步求解, 设 $y = Ux$
2. $Ly = f$
3. $Ux = y$

Norms 范数:

定义：满足三个条件

1. 正定性
2. 同质性
3. 满足三角不等式

在某范数下收敛于 x 即与 x 的差的范数一致小于 ϵ

实空间里所有范数等价（在任意范数下收敛则所有范数下收敛）

Vector Norm

一阶范数：绝对值之和

二阶范数：欧拉距离

无穷范数：最大的元素绝对值

负无穷范数：最小的元素绝对值

Matrix Norm

定义比 vector norm 多了一项
4. consistence:$||AB|| \leq ||A||||B||$

一般有两种

Frobenius Norm

所有元素绝对值的平方和

Natural Norm

由向量范数导出，$||A||_p = max_{||x||_p=1}||Ax||_p$

无穷范数：元素绝对值每一行的和的最大值

第一范数：元素绝对值每一列的和的最大值

第二范数：$\sqrt{\lambda_{max}(A^TA)}$ 即 A^TA 这个矩阵最大的特征值的平方根，也就是谱半径，对于方阵来说就是特征值绝对值的最大值

Spectral Radius

$\rho(A) = max|\lambda| \leq ||A||$

谱半径等于特征值模长（可能是复数）的最大值，小于等于列元素绝对值和的最大值

A 对称的时候即为第二自然范数

$|\lambda| \cdot ||x|| = ||\lambda x|| = ||Ax|| \leq ||A|| \cdot ||x||$

Jacobi Iterative Method

$Ax = b$

把 A 分为 D, -L, -U 三个矩阵相加

$(D - L - U)x = b$
$Dx = (L + U)x + b$
$x = D^{-1}(L + U)x + D^{-1}b$
$T = D^{-1}(L + U)$
$C = D^{-1}b$
$x = Tx + c$

具体计算:

$x_i = \frac{b_i - \sum_{j=1,j\neq i}^{n}(a_{ij}X0_j)}{a_{ii}}$

可以做并行计算

Gauss - Seidel Iterative Method

$(D - L)x = Ux + b$

不储存 X0，每次直接用更新完的计算 X_{i+1}

不能并行

SOR(Successive Over Relaxation)

$(D/\omega - L)X = ((1/\omega - 1)D + U)X + b$
$x_i = x_i - \omega\frac{r_i}{a_i i}$, $r_i = b_i - \sigma a_{ij}x_j$
$\omega > 1$X 不保存，边算边用
$\omega = 1$ 时即为 Gauss-Seidel,$\omega < 1$ 时为 Under-Relaxation
kahan 定理，只有 $0 < \omega < 2$ 时，SOR 才能收敛
Ostrowski-Reich 定理，如果 A 正定而且 $0 < \omega < 2$，SOR 对于任意初值收敛

Convergency of Iterative Methods

: The following statements are equivalent:

(1) A is a convergent matrix;

(2) $lim_{n\to\infty}||A^n|| = 0$ for some natural norm;

(3) $lim_{n\to\infty}||A^n|| = 0$ for all natural norms;

(4) $\rho(A) < 1$; 常用

(5) $lim_{n\to\infty}A^n x = 0$

error bounds:

$||x - x_k|| \approx \rho(T)^k ||x - x_0||$

T 是 Jaccobi 的 T

Relaxation Methods

$X_i^k = X_i^{k-1} - \omega(\frac{r_i^k}{a_{ii}})$

$r_i^k = b_i - \sum_{j<i} a_{ij}x_j^k - \sum_{j\geq i} a_{ij}x_j^{k-1}$

$\omega = 1$ 时即为 Gauss - Seidel Iterative Method

$T = I + \omega A$

拉格朗日基:

第 i 个基在第 i 个插值点为 1，其他插值点为 0

$Ln,i(x) = \prod_{j=0,j\neq i}^{n}\frac{x - x_j}{x_i - x_j}$

n 是次数，从 0 开始

$P_n(x) = \sum L_{n,i}(x)y_i$

Rolle's Theorem:

n 个零点所在的区间里必有一个点的 n-1 阶导数为 0

Remainder

$R(x) = f(x) - P_n(x)$

$g(t) = R(t) - K(x)\prod(t - x_i)$ 这个 x 是不等于 x_i 的任意固定值

根据 Rolle's Theorem 存在一个 ζ_x 满足 $g^{(n+1)}(\zeta_x) = 0$，带入上述两式，又因为 $P^{(n+1)}(\zeta_x) = 0$，推出

$R_n(x) = \frac{f^{(n+1)(\zeta_x)}}{(n+1)!}\prod_{i=0}^{n}(x - x_i)$

但是 ζ_x 不一定能求得，常用 $f^{(n+1)}(\zeta_x)$ 的一个上界来估算 $R_n(x)$

Condition number

$||A|| \cdot ||A^{-1}||$ is the key factor of error amplification, and is called the condition number K(A). K(A) 越大越难获得精确解

$A(x + \delta x) = b + \delta b$

$\frac{||\delta x||}{||x||} \leq K(A) \cdot \frac{||\delta b||}{||b||}$

$(A + \delta A)(x + \delta x) = b + \delta b$

$\frac{||\delta x||}{||x||} \leq \frac{K(A)}{1 - K(A)\frac{||\delta A||}{||A||}} \cdot (\frac{||\delta A||}{||A||} + \frac{||\delta b||}{||b||})$

Refinement

1. $Ax = b$
2. $r = b - Ax$
3. $Ad = r$
4. $x = x + d$

Power method

$x^k = Ax^{k-1}$

相当于把一个随机向量塞到面团里，然后拉拉面，最后这个向量会跟拉面平行，即最大特征值对应的特征向量方向

要求最大特征值唯一，不能互为相反数（但可以两个完全相同）

$\lambda \approx \frac{x_i^k}{x_i^{k-1}}$

Normalization

1. $u^{k-1} = \frac{x^{k-1}}{|x^{k}-1|}$
2. $x^k = Au^{k-1}$
3. $\lambda = max(x_i^k)$

Rate of Convergence

收敛速率是 $|\lambda 2/\lambda 1|$，更快的收敛速率要尽量让 $\lambda 2$ 小，调整原点位置到 $(\lambda 2 + \lambda n)/2$ 可以再不产生新的 $\lambda 2$ 的情况下让 $\lambda 2$ 最小

Inverse Power Method

可以求得绝对值最小的特征值

求 p_0 附近的特征值:

1. $B = A - p_0 I$
2. $x^k = B^{-1}x^{k-1}$
3. $\frac{1}{\lambda} = x^k/x^{k-1}$

但是导致 condition number 降低

一般内插比外插要准确

拉格朗日插值如果新增加一个插值点需要全部重算

下面两种方法都更方便于添加插值点

Neville's Method:

用两个同阶的 p 合并可以得到更高阶的 p

$p_{1,2,3,4}(x) = \frac{(x - x_4)p_{1,2,3} - (x - x_1)p_{2,3,4}}{x_1 - x_4}$

Newton's Interpolation

$f[x_0 \cdots x_k] = \frac{f[x_0 \cdots x_{k-1}] - f[x_1 \cdots x_k]}{x_0 - x_k}$

$f(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, \cdots, x_n](x - x_0)\cdots(x - x_n)$

Hermite Interpolation

插值满足在 n 个点出给出的值和 m 个点处给出的斜率

$H(x) = \sum_{i=0}^{n} f(x_i)h_i(x) + \sum_{i=0}^{m} f'(x_i)\hat{h}_i(x)$

where $h_i(x_i) = (i == j)$, $h_i'(x_j) = 0$, $\hat{h}_i(x_j) = 0$, $\hat{h}_j'(x_j) = (i == j)$

推出

$h_i(x) = [1 - 2L'_{n,i}(x_i)(x - x_i)]L^2_{n,i}(x)$
$\hat{h}(x) = (x - x_i)L^2_{n,i}(x)$

Cubic Spline Interpolation

解决了随着插值点的增多插值函数并不收敛于原函数的问题

方法是用子区间三阶插值来拟合原函数

Least Squares Approximation

m 个点，用 n 阶多项式函数 P 你和 f，$n \ll m$

使得 $E = \sum_1^m [P_n(x_i) - y_i]^2$ 最小

即求 P 的系数 a 满足 E 最小，求偏导可得

Let $b_k = \sum_1^m x_i^k$，$c_k = \sum_1^m y_i x_i^k$

出题的话一般求导令导数等于 0 即可，计算机则解下列矩阵 $\begin{pmatrix} b_{0+0} & \cdots & b_{0+n} \\ \vdots & \ddots & \vdots \\ b_{n+0} & \cdots & b_{n+n} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}$ P 是非线性函数的时候使用换元法变成线性，直接求导应该也是可以的

General Least Squares Approximation

$E = \sum_1^m w_i[P_n(x_i) - y_i]^2$

$E = \int_1^m w(x)[P(x) - f(x)]^2$

如果 (f, g) 则两个基正交

$(f, g) = \begin{cases} \sum_1^m w_i f(x_i) g(x_i) \\ \int_a^b w(x) f(x) g(x) dx \end{cases}$

设 $P(x) = a_0\varphi_0(x) + \ldots + a_n\varphi_n(x)$，求 a $\begin{pmatrix} b_{ij} = (\varphi_i, \varphi_j) \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} (\varphi_0, f) \\ \vdots \\ (\varphi_0, f) \end{pmatrix}$ 取 $\varphi_j(x) = x^j$ 时，$(\varphi_i, \varphi_j) = \frac{1}{i+j+1}$，不是正交的，Hilbert Matrix，不好算，条件数大

构造正交基

$\varphi_0(x) = 1$
$\varphi_k(x) = (x - B_k)\varphi_{k-1}(x) - C_k\varphi_{k-2}(x)$
$B_k = \frac{(x\varphi_{k-1}, \varphi_{k-1})}{(\varphi_{k-1}, \varphi_{k-1})}$
$C_k = \frac{(x\varphi_{k-1}, \varphi_{k-2})}{(\varphi_{k-2}, \varphi_{k-2})}$

forward backward

拉格朗日插值分析，误差级别是 O(h)

积分近似

用拉格朗日插值的积分近似

$\int_a^b f(x)dx = \sum_0^n f(x_k) \int_a^b L_k(x)dx$

对于等间距取点的情况，后面朗格朗日项的积分不依赖于 f 或者区间，所以说是 stable 的

误差估计

使用拉格朗日余项积分获得误差

$R[f] = \int_a^b \frac{f^{(n+1)(\zeta_x)}}{(n+1)!} \prod_{i=0}^n (x - x_i)$

积分准确度

对几阶多项式是完全准确的

Simpson's Rule

$\int_a^b f(x)dx = \frac{b-a}{6}[f(a) + 4f((a+b)/2) + f(b)]$

Chebyshev Polynomial

Chebyshev Polynomial 如下，两种表达方式

$T_0(x) = 1,\ T_1(x) = x$
$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$
$T_n(x) = cos(n\ arccos(x))$

该多项式是用于选择取样点的，用该多项式选取的取样点插值可以使得无穷范数最小，即最大的误差最小。最大误差处的点称为 deviation point。

原理：余项中的 $w_n(x)$ 最小 (余项中仅有此项与点选取有关)

$w_n(x) = \prod(x - x_i)$
$w_n(x) = x^n - P_{n-1}(x)$
$P_{n-1} n + 1 deviation\ points.$

目标为寻找一个 $P_{n-1}(x)$ 使得 $w_n(x)$ 最小，用下式取点

$T_n(x)\ has\ n\ root$
$x_k = cos(\frac{2k-1}{2n}\pi)$

确定点后用拉格朗日插值

Economization

减去高阶切比雪夫多项式达到降阶的目的

n 阶 mono chebyshev 多项式（最高项系数归一化）的最大值 $\frac{1}{2^{n-1}}$

Chatper 4

Composite Numerical Integration

分段过细拟合会导致过拟合（采样点太多，阶数变高）

所以要分段拟合积分

Composite Trapezoidal Rule

$\int_a^b f(x)dx \approx \frac{h}{2}[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)] = T_n$
$R[f] = -\frac{h^2}{12}(b-a)f''(\xi)$

Composite Simpson's Rule

$h = \frac{2(b-a)}{n}$
$\int_{x_k}^{x_{k+2}} f(x)dx \approx \frac{h}{6}[f(x_k) + 4f(x_{k+1}) + f(x_{k+2})]$
$\int_a^b f(x)dx \approx \frac{h}{3}[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)] + R[f]$
$R[f] = -\frac{b-a}{180}h^4 f^{(4)}(\xi)$

Romberg Integration

迭代直到 $S_{k,0}$ 与 $S_{k-1,0}$ 误差小于 ϵ

$S_{0,n} = T_n$
$S_{k,n} = \frac{4^k S_{2n} - S_n}{4^k - 1}$

计算顺序如 $T_1, T_2, T_4, T_8, S_1, S_2, S_4, C_1, C_2, R_1$

余项 $R_{2n}[f] = -(\frac{h}{2})^2 \frac{1}{12}(b-a)f''(\xi) \approx \frac{1}{4}R_n[f]$

Richardson's Extrapolation

某种插值方法 T_0 的误差为 $T_0 - I = a_1h + a_2h^2 + a_3h^3 + \cdots$，则可使用下式迭代减小误差

$T_m(h) = \frac{2^m T_{m-1}(\frac{h}{2}) - T_{m-1}(h)}{2^m - 1} = I + \delta_1 h^{m+1} + \delta_2 h^{m+2} + \cdots$

Adaptive Quadrature Methods

自动在变化剧烈的地方多采样

simpson 积分，分为两个区间后得到两个等式，假设等式中的导数相等，得到一个 error 关于整个区间的辛普森和两个半区间的辛普森的表达式。当 error 小于给定值时即可停止区间细分，否则继续细分

$\int_a^b f(x)dx = S(a,b) + \frac{h^5}{90} f^{(4)}(\xi_1)$
$\int_a^b f(x)dx = S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b) + \frac{1}{16}\frac{h^5}{90} f^{(4)}(\xi_2)$
$Assuming\ f^{(4)}(\xi_1) \approx f^{(4)}(\xi_2)$
$\epsilon = \frac{1}{15}|S(a,b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)|$

Gaussian Quadrature

目标：用尽量少的点得到 2n+1 阶的精确积分拟合（在 w(x) 意义下的积分），注意是积分，而不是函数，只能得到一个精确的值而不是函数表达式。即

$\int w(x)f(x) = \sum A_k f(x_k)$

高斯积分只用 n+1 个点的采样即可做到。

如何得到这 n 个点：

以这 n+1 个点为根的多项式 $W(x) = \prod_0^n (x - x_i)$，满足与所有小于等于 n 阶的多项式正交。利用正交化可解出 $W(x)$

证明：

精确 => 正交

$q \le 2n+1,\ m \le n$
$\int w(x)Q_q(x) = \int w(x)P_m(x)W(x) = \sum A_k P_m(x)W(x_k) = 0$

正交 => 精确

$Q(x) = W(x)Q(x) + r(x)$
$\int w(x)Q(x) = \int w(x)W(x)q(x) + \int w(x)r(x) = 0 + \int w(x)r(x)$
$= \sum_0^n A_k r(x_k) = \sum_0^n [A_k W(x_k)q(x_k) + A_k r(x_k)] = \sum_0^n A_k Q(x_k)$

Euler method

$w_{i+1} = w_i + hf(t_i, w_i)$

误差 $|y_i - w_i| \le \frac{hM}{2L}[e^{L(ti-a)} - 1]$

yi 是精确值，wi 是数值值，M 是 $max(|y''(x)|)$，L 是 lipschitz 常数，ti 是第 i 个点 x，a 是起点

有 round off error 的时候，δ 是误差的界

误差 $|y_i - w_i| \le \frac{1}{L}(\frac{hM}{2} + \frac{\delta}{h})[e^{L(ti-a)} - 1] + |\delta_0|e^{L(t_i-a)}$

truncation error

$\tau_{i+1} = \frac{y_{i+1}-y_i}{h} - f(t_i, y_i) = \frac{h}{2}y''(\xi)$

Implicit Euler's Method

$\tau_{i+1} = \frac{y_{i+1}-y_i}{h} - f(t_i, y_i) = -\frac{h}{2}y''(\xi)$

Trapezoidal Method (modified Euler's Method)

$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_{i+1})]$

local truncation error $O(h^2)$

Double-Step Method

$w_{i+1} = w_{i-1} + 2hf(t_i, w_i)$

local truncation error $O(h^2)$

Runge-Kutta Method

$w_{i+1} = w_i + (\lambda_1 + \lambda_2)hy'(t_i) + \lambda_2 ph^2 y''(t_i) + O(h^3)$

$\lambda_1 + \lambda_2 = 1, \lambda_2 p = \frac{1}{2}$

local truncation error $O(h^2)$

$w_{i+1} = w_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K4)$
$K_1 = f(t_i, w_i)$
$K_2 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_1)$
$K_3 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}K_2)$
$K_4 = f(t_i + h, w_i + hK_3)$

Multi-Step Method

$w_{i+1} = a_1 w_i + a_2 w_{i-1} + h(b_1 f_i + b_2 f_{i-1})$
求 a_1, a_2, b_1, b_2，使用泰勒展开，w_j 全部展开成从 w_i 出发，f_j 全部展开成从 f_i 出发。展开后带入原式，比较系数

Stability

假设某种迭代方法如下

$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(t_i, h, w_{i+1}, w_i, \cdots, w_{i+1-m})$

则该积分的特征多项式 $P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - \cdots - a_1\lambda - a_0 = 0$

解得特征根，如果所有 $|\lambda_i| \le 1$，并且模长为 1 的根都是单根，则称为满足 root condition

1. 满足 root condition 并且模长为 1 的特征根只有 1 的方法，称为 strongly stable
2. 满足 root condition 并且有多个模长为 1 的特征根的方法，称为 weakly stable
3. 不满足 root condition 的方法称为 unstable