Computational Physics

Episode 1. 计算方法

Computational Physics

REF：计算方法/scherer computational physics/Hoffman 计算物理学/丁泽军讲义

涉及内容：经典力学问题/电磁学/QM本征值问题/统计力学 相变/随机过程/二次量子化/严格对角化/开放系统耗散问题/固体物理/随机矩阵

---

1 Review：数值方法

1.1 插值

给定n组数据点$(x_1, y_1), \cdots, (x_n, y_n)$，考虑存在函数$f(x)$，满足$f(x_i) = y_i$，显然在这些点处是没有误差的$e_n(x_i) = 0$，显然误差满足

$$e_n(x) \propto \prod_i^n (x - x_i) \tag{1}$$

误差阶数是n，那么$f(x)$应该是一个阶数小于n的多项式，形式上

$$f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + e_n(x) \tag{2}$$

直接的想法是求解方程组

$$y_i = f(x_i) = \sum_{j=0}^{n-1} a_j x_i^j \quad \text{for all } \ i = 1, \cdots, n \tag{3}$$

1.1.1 Lagrange插值

插值的用意：

计算积分

$$I = \int_a^b f(x) \, \mathrm{d}x \tag{4}$$

计算积分最粗糙的方式是取矩形近似，但毫无疑问精度是不太够的，此时积分值表示为$I = h \sum_i y_i$。外推这个表达式，有一般形式

$$I = (b - a) \sum_i y_i w_i \tag{5}$$

其中w_i为权重，$\sum_i w_i = 1$，例如矩形近似的等权重的。上式通常称为和积分形式，但处理高微积分时效率并不高。矩形近似的误差为$\sim \mathcal{O}(h)$，采取梯形近似的误差为$\sim \mathcal{O}(h^2)$

$$I_i = [f(x_i + h) - f(x_i)]h/2 \simeq f'(x_i)h^2/2 \tag{6}$$

实际上对应Taylor展开得不同阶项

$$f(x_i + z) = f(x_i) + f'(x_i)z + f''(x_i)z^2/2! + \cdots \tag{7}$$

的解析积分值。这么做可以不断提高积分的精度，但是前提我们需要知道$f(x)$的各阶导数，而实际情况往往不能满足。不要求导数信息的数值方法Simpson summation具有$\sim \mathcal{O}(h^3)$，记已知数据点$\{y_i(x_i) : i = 1, \cdots, n\}$

$$I = (x_2 - x_1)(f_1 + f_2)/2 + (x_3 - x_2)(f_2 + f_3)/2 + (x_4 - x_3)(f_3 + f_4)/2 + \cdots \tag{8}$$

数据点等距的情况下

$$I = \frac{h}{2}(f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n) \tag{9}$$

这种方法实际上用两点插值

$$f(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \tag{10}$$

给出的函数$f(x)$完成了解析积分运算。

进一步，考虑Lagrange插值

$$f(x) = \sum_i w_i l_i(x) = \sum_i y_i \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} \tag{11}$$

考虑等距数据点，积分表示为

$$I = \sum_i^n \int_{x_1}^{x_1 + (n-1)h} y_i l_i(x)\,\mathrm{d}x = \sum_i^n y_i \int_0^{n-1} h \frac{\prod_{j \neq i}(x' - x'_j)}{\prod_{j \neq i}(x'_i - x'_j)}\,\mathrm{d}x' = h \sum_i^n y_i w_i^{(n)} \tag{12}$$

其中第二步移动坐标原点以及rescale$x = hx'$，最后的权重定义为

$$w_i^{(n)} := \int_0^{n-1} \frac{\prod_{j \neq i}(x' - x'_j)}{\prod_{j \neq i}(x'_i - x'_j)}\,\mathrm{d}x' \tag{13}$$

可直接查表。

注意到Lagrange函数的性质

$$l_i(x_j) = \delta_{ij} \tag{14}$$

定义

$$l(x) = \sum_{i=1}^n l_i(x) - 1 \tag{15}$$

可以验证

$$\sum_i l_i(x) = 1 \tag{16}$$

同时也满足权重函数的归一性。

1.1.2 Newton-Cotes插值

对于Lagrange插值，增加一个数据点时，插值函数要完全重新计算；相对的Newton-Cotes插值在处理上简单很多，只需在插值多项式上额外增加一项即可。

考虑如下形式的插值多项式

$$f(x) = b_0 + b_1(x - x_1) + b_2(x - x_1)(x - x_2) + \cdots + b_n(x - x_n)\cdots(x - x_n) \tag{17}$$

取$x = x_i$，分别得到

$$b_0 = f(x_1),\ b_1 = \frac{f(x_1) - f(x_2)}{x_1 - x_2} := f[x_1, x_2],\ b_2 = \frac{f[x_1, x_2] - f[x_2, x_3]}{x_1 - x_3} := f[x_1, x_2, x_3],\ \cdots \tag{18}$$

1.1.3 其他插值方法

切比雪夫插值，考虑一个例子

$$\int_{-1}^{1} f(t)\,\mathrm{d}t = \int_0^{\pi} f(\cos t)\sin t\,\mathrm{d}t \tag{19}$$

其中

$$f(\cos t) = \frac{1}{2M}C_0 + \frac{1}{M}\sum_j C_j\cos(jt) + \frac{1}{2M}C_M\cos(Mt) \tag{20}$$

$j = 0, \cdots, M$，系数可以通过FFT计算。

1.2 Runge-Kutta方法

从一个ODE出发

$$\mathrm{d}y\,/\,\mathrm{d}x = f(x, y) \tag{21}$$

这个问题通常可以应用于两种场合：(1)ODE求解；(2)作为高精度的积分器
$y_{n+1} - y_n = \int_{x_n}^{x_{n+!}} f(x, y)\,\mathrm{d}x$。一个trivial的处理方式是

$$\begin{aligned} y_{n+1} &= y_n + \int_{x_n}^{x_{n+1}} f(x, y)\,\mathrm{d}x \\ &\simeq y_n + (x_{n+1} - x_n)f(x_n, y_n) + \mathcal{O}(h^2) \\ &\simeq y_n + \frac{x_{n+1} - x_n}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] + \mathcal{O}(h^3) \end{aligned} \tag{22}$$

第三行的由于y_{n+1}的计算要调用自身的值，计算方式收敛性会相对好，通常我们会对y_{n+1}取一个预估值$\tilde{y}_{n+1}$（可以用第二行的表达式算），然后在此基础上反复迭代计算y_{n+1}的值。

换个角度考虑上式的积分

$$\begin{aligned} y_{n+1} &= y_n + \int_0^h f(x_n + z, y(x_n + z))\,\mathrm{d}z \\ &= y_n + \int_0^h f(x_n + z, y_n + y'_n z + y''_n z^2/2)\,\mathrm{d}z \\ &= y_n + \int_0^h f + f_x z + f_{xx} z^2/2 + f_y(y'_n z + y''_n z^2/2) + \cdots \mathrm{d}z \end{aligned} \tag{23}$$

$$
\begin{aligned}
&= y_n + f(x_n, y_n)h + f_x h^2/2 + f_y y'_n h^2/2 + \mathcal{O}(h^3) \\
&= y_n + f(x_n, y_n)h + h^2[f_x(x_n, y_n) + f_y(x_n, y_n) f(x_n, y_n)]/2 + \mathcal{O}(h^3)
\end{aligned}
$$

这种算法需要计算$f(x, y)$的各阶偏导数，实际仍然不便利。如何避开导数的运算？Runge-Kutta方法在保证精度的情况下，避开了导数的运算（证明较复杂）。

Runge-Kutta 23

由于

$$
\begin{aligned}
y_{n+1} &= y_n + f(x_n, y_n)h + h^2[f_x(x_n, y_n) + f_y(x_n, y_n) f(x_n, y_n)]/2 + \mathcal{O}(h^3) \\
&= y_n
\end{aligned} \tag{24}
$$

则

$$
y_{n+1} = y_n + h[af(x_n, y_n) + bf(x_n + h, y_n + h)] + \mathcal{O}(h^3) \tag{25}
$$

这个计算方法有3阶精度。

Runge-Kutta 45

令

$$
\begin{aligned}
k_1 &= f(x_n, y_n) \\
k_2 &= f(x_n + h/2, y_n + hk_1/2) \\
k_3 &= f(x_n + h/2, y_n + hk_2/2) \\
k_4 &= f(x_n + h, y_n + k3)
\end{aligned} \tag{26}
$$

则

$$
y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5) \tag{27}
$$

上述数值方法也称为ODE45。

1.3 求解二阶ODE

物理学问题中常常有二阶微分方程，例如单摆的运动方程

$$
ml\ddot{\theta} = mg\sin\theta \tag{28}
$$

这种二阶方程可以通过设$v = \dot{\theta}$，reduce至一阶：

$$
\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\theta \\ v\end{pmatrix} = \begin{pmatrix} v \\ g\sin\theta/l \end{pmatrix} \tag{29}
$$

值得注意的是，Runge-Kutta方法也支持诸如$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$的向量情形。

HW
：Lagrange力学模拟问题

a.双摆问题

b. 引力模型$U=-A/r^{\alpha}$

c. 竖杆倾倒问题

Notes on the Runge-Kutta method

示例：MMA求解经典力学问题

二维中心势场$U=-A/r^{\alpha}$下的经典粒子运动问题。保守力场下拉格朗日量为

$$\mathcal{L}=\dot{x}^{2}(t)/2+\dot{y}^{2}/2-U \tag{30}$$

拉格朗日方程

$$\begin{cases} 0= & \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right)-\frac{\partial\mathcal{L}}{\partial x} \\ 0= & \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{y}}\right)-\frac{\partial\mathcal{L}}{\partial y} \end{cases} \tag{31}$$

以及初始条件$x(0),\dot{x}(0);y(0),\dot{y}(0)$。

```

```

2 TEST

2.1 TEST

2.1.1 TEST