作业二(2020年11月)
严禁抄袭作业

第一题 分别使用以下方法计算$f(x)=\sin(\cos(\sin(\cos(x))))$在$[-1,1]$上的积分

(a) （10分）使用自动控制误差的复化梯形公式，给出一个你认为计算得最精确的结果并作图展示误差（使用`semilogy`）是如何随着迭代次数变化的。

(b) （10分）使用Richardson外推方法，给出一个你认为计算得最精确的结果并作图展示误差（使用`semilogy`）是如何随着迭代次数变化的。

(c) （10分）使用Gauß积分，给出一个你认为计算得最精确的结果并作图展示误差（使用`semilogy`）是如何随着原函数的采样点的个数，即积分点的个数变化的。Gauß积分点和积分权重使用公布在课程主页上的程序计算即可。

(d) （10分）在数值计算中，对被积分函数采样，即求被积分函数在积分点上的值并在内存中保留这些值通常被认为是计算的主要负担，因此我们用对被积分函数的采样次数，即积分点的个数来衡量算法的效率。给出上述三种方法在达到能达到的最高精度的情况下（这三种方法的计算精度在相对误差的意义下应该都能够达到或者接近[1]计算机的机器精度[2]）的采样总数目，进而为三种方法的效率排序。

第二题 **函数求导与微分矩阵(differentiation matrix)**

当我们用$n+1$个插值点$\{x_j\}_{j=0}^n$对给定函数$f(x)$做Lagrange插值时，得到Lagrange插值多项式

$$p(x)=\sum_{j=0}^{n} f_j \ell_j(x)$$

此处，Lagrange插值基函数(Lagrange polynomial)

$$\ell_j(x)=\frac{1}{\pi_j}\prod_{\substack{k=0\\k\neq j}}^{n}(x-x_k), \qquad \pi_j=\prod_{\substack{k=0\\k\neq j}}^{n}(x_j-x_k)$$

满足

$$\ell_j(x_k)=\begin{cases}1 & k=j\\ 0 & k\neq j\end{cases}$$

[1]注意这里只是说接近，并不是说非要严格达到。例如，$\mathcal{O}(10^{-15})$可以被看作是接近计算机的机器精度。

[2]在双精度（double precision）的计算环境下，计算机的机器精度约为2.22×10^{-16}。

同时为了方便，我们令$f_j = f(x_j)$。计算$f(x)$的导数的一种最自然的想法是用$p(x)$的导数，即

$$p'(x) = \sum_{j=0}^{n} f_j \ell_j'(x)$$

近似$f'(x)$。

(a) （10分）证明$\ell_j'(x)$可以用$\ell_j(x)$简洁地表示：

$$\ell_j'(x) = \ell_j(x) \sum_{\substack{k=0 \\ k\neq j}}^{n} (x - x_k)^{-1}$$

因而

$$p'(x) = \sum_{j=0}^{n} \left(f_j \ell_j(x) \sum_{\substack{k=0 \\ k\neq j}}^{n} (x - x_k)^{-1} \right) \tag{1}$$

(b) （10分）取$n = 15$，即基于16个等距插值点，使用(1)计算$f(x) = \sin(x)$在$[-1, 1]$上的导数：在$[-1, 1]$上1001个等距点上计算所得的导函数多项式逼近关于真实解$f'(x) = \cos(x)$的误差，并用`semilogy`画出这些逐点误差的绝对值。

(c) （10分）公式(1)提供给了我们一个当给定函数在某些插值点（或叫做取样点）上的函数值已知的情况下求其导数的方法。某些情况下，我们只关心给定函数在这些插值点上的导数的情况。这时我们就可以由(1)得到

$$p'(x_i) = \sum_{j=0}^{n} \left(f_j \ell_j(x_i) \sum_{\substack{k=0 \\ k\neq j}}^{n} (x_i - x_k)^{-1} \right) \tag{2}$$

写成矩阵的形式就是

$$\mathbf{p}' = \mathbf{D}\mathbf{f}$$

此处

$$\mathbf{p}' = \begin{pmatrix} p'(x_0) \\ p'(x_1) \\ \vdots \\ p'(x_n) \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$

根据(2)，证明微分矩阵$\mathbf{D}$的元素

$$D_{ij} = \begin{cases} \dfrac{1}{\pi_j} \displaystyle\prod_{\substack{k=0 \\ k\neq i,j}}^{n} (x_i - x_k) = \dfrac{\pi_i}{\pi_j (x_i - x_j)} & i \neq j \\ \displaystyle\sum_{\substack{k=0 \\ k\neq j}}^{n} (x_j - x_k)^{-1} & i = j \end{cases}$$

(d) （10分）用微分矩阵D计算函数$f(x)=\sin(3x^2)$在插值点的导数：依次令$n=1,3,5,...,57,59$，取$[-1,1]$上$n+1$个等距点，用这$n+1$个点上的函数值计算原函数的在这些点上的导数值，并取这$n+1$个点上逐点误差的绝对值的最大值，再将这个值随n变化的情况用`semilogy`图画出（横轴为n，纵轴为误差最大值）。重复同样的实验，但改用Chebyshev点

$$x_j=\cos(j\pi/n) \qquad j=0,1,...,n$$

并将逐点误差绝对值的最大值随n变化的情况画到同一幅图内。对比两种点所带来的误差，阐述你观测到的现象，并给出解释。

第三题 (a) （10分）推导出如下格式的多步法公式：

$$y_{n+1}=y_{n-1}+\alpha f_{n+1}+\beta f_n+\gamma f_{n-1}+\mu f_{n-2}$$

(b) （10分）推导此格式的局部截断误差，并由此指明此格式的阶数。

(c) （20分）选取合适的步长值，用此格式在$[0,2]$上解如下的初值问题：

$$y'=xe^{-5x}-5y, \quad y(0)=0 \tag{3}$$

使用该格式时使用三阶Runge-Kutta方法起步，并解释为什么三阶的Runge-Kutta方法作为三阶格式并不会影响使用该多步法时的精度。

(d) （10分）推导出(3)的精确解。对比该精确解，用`log-log`图验证你推断的阶数是否正确。