Homework 1

李青松 BA21002019

2021 年 10 月 8 日

目录

Problem 1

summarize the major models and algorithms of numerical method

1.1 插值

1.1.1 n 次多项式插值

对于一组插值点：$\{(x_i, f(x_i), i = 0, 1, \ldots, n\}$, 构造 n 次多项式 $P_n(x) = a_0 + a_1x + \cdots + a_nx^x$，将插值点带入多项式，得到：$VA = F$, 其中 V 是范德蒙行列式，A 是 a_0 等系数组成的列向量，F 是 $f(x_i)$ 组成的列向量。x_i 互异时，可解得 $A = V^{-1}F$。

1.1.2 拉格朗日插值法

构造多项式：$P_n(x) = \sum_{k=0}^{n} e_k(x)f(x_k)$, 其中 $e_k(x) = \frac{\prod_{j\neq k}(x-x_j)}{\prod_{j\neq k}(x_k-x_j)}$

1.1.3 牛顿插值法

构造多项式：$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\ldots(x - x_{n_1})$, 依次求解得：$a_0 = f(x_0), a_1 = \frac{f(x_1)-f(x_0)}{x_1-x_0}, \ldots$.

1.2 最小二乘拟合

选取候选拟合函数 $\phi_\alpha(x)$, 利用最小二乘法,求参数 α, 使得误差 Q=$\sum_{i=1}^{n}(\phi_\alpha(x_i) - y_i)^2$ 最小。

1.2.1 多项式拟合

给定一组数据 $\{x_1, y_i), i = 0, 1, \ldots, m\}$, 拟合函数 $\phi(x) = a_0 + a_1 x + \cdots + a_n x^n$, 要使得 $Q(\alpha) = \sum_{m=1}^{n}(a_0 + a_1 x_i + \cdots + a_n x_i^n - y_i)^2$ 最小，其中 $\alpha = (a_0, a_1, \ldots, a_n)$, 可以微分后给出：

$$\begin{pmatrix} m & \sum_{i=1}^m x_i & \cdots & \sum_{i=1}^m x_i^n \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 & \cdots & \sum_{i=1}^m x_i^{n+1} \\ \vdots & \ddots & & \ddots \\ \sum_{i=1}^m x_i^n & \sum_{i=1}^m x_i^{n+1} & \cdots & \sum_{i=1}^m x_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m y_1 \\ \sum_{i=1}^m x_i y_i \\ \vdots \\ \sum_{i=1}^m x_i^n y_i \end{pmatrix}. \quad (1.1)$$

对于一般的 $y_1, y_2, \ldots, y_m$ 方程组都有解。

1.2.2 矛盾方程组拟合

对于一般的拟合函数 $\phi(x) = \alpha_1\phi_1(x) + \cdots + \alpha_n\phi_n(x)$, 其中 $\phi_i(x)$ 是已知函数，α_i 是未知参数，记

$$A = \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) \cdots \phi_n(x_1) & & \\ phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_n(x_2) \\ \vdots & \vdots & \vdots & \\ \phi_1(x_m) & \phi_2(x_m) & \vdots\phi_n(x_m) & \end{pmatrix}, \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}. \quad (1.2)$$

误差 $Q(\alpha) = \sum_{i=1}^{m}(\phi(x_i) - y_i)^2 = ||A\alpha - Y||_2^2$, $\mathrm{A}\alpha = Y$ 有解时，拟合函数经过所有数据点，当无解时，成为矛盾方程组，使 $||A\alpha - Y||_2$ 最小的 α 成为矛盾方程组的最小二乘解。

1.3 非线性方程组求解

1.3.1 迭代法

1. 实根的对分法：设函数在 $[a, b]$ 上连续，且 $f(a)f(b)<0$, 则 $f(x)$ 在区间 $[a, b]$ 至少有一个零点，计算中通过对分区间，不断搜小区间范围搜索零点的位置.

2. 不动点迭代：对给定的方程 $f(x) = 0$，将其转换为 $x = \phi(x)$. 给定初值 x_0，由此构造迭代序列 $x_{k+1} = \phi(x_k)$, 如果 $\phi(x)$ 连续且迭代收敛：$\lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} \phi(x_k) = \alpha$, 则有 $\alpha = \phi(\alpha)$，α 就是方程的根。

3.Nerton 迭代法：作 $f(x)$ 在 x_0 的 Taylor 展开：

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots \tag{1.3}$$

取线性部分作为 $f(x) \approx 0$ 的近似值，则有 $x = x_0 - \frac{f(x_0)}{f'(x_0)}$, 令 $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, 再作 $f(x)$ 在 x_1 的 Taylor 展开，一直做下去可得到迭代式 $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

3. 弦截法：在 Newton 迭代格式中，用差商代替导数，并给定两个初始值 x_0, x_1, 得到弦截法迭代公式：

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, \dots \tag{1.4}$$

4. 求解非线性方程组的 Newton 方法：为方便讨论我们给出二阶方程组的公式，高阶方程组可类似给出。设二阶方程组：

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases} \tag{1.5}$$

方便起见，将方程组写成向量形式：

$$F(w) = \begin{pmatrix} f(x, y) = 0 \\ g(x, y) = 0 \end{pmatrix}, \quad w = \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.6}$$

对 $f(x, y), g(x, y)$ 在 x_0, y_0 处作二元展开，取其线性部分，令 $x - x_0 = \Delta x, y - y_0 = \Delta y$, 则有：

$$F(w) = \begin{cases} \Delta x \frac{\partial f_1(x_0, y_0)}{\partial x} + \Delta y \frac{\partial f_1(x_0, y_0)}{\partial y} = -f_1(x_0, y_0) \\ \Delta x \frac{\partial f_2(x_0, y_0)}{\partial x} + \Delta y \frac{\partial f_2(x_0, y_0)}{\partial y} = -f_2(x_0, y_0) \end{cases} \tag{1.7}$$

如果

$$det(J(x_0, y_0)) \left| \begin{matrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{matrix} \right|_{(x_0, y_0)} \neq 0 \tag{1.8}$$

解出 $\Delta x, \Delta y$,

$$w_1 = w_0 + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x_0 + \Delta x \\ y_0 + \Delta y \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{1.9}$$

再列出同样的方程组解出 $\Delta x, \Delta y$，得到 w_2, 重复做下去，每次都是解一个类似的方程组：

$$J(x_k, y_k) = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -f(x_k, y_k) \\ -g(x_k, y_k) \end{pmatrix} \tag{1.10}$$

其中 $\Delta x = x_{k+1} - x_k, \quad \Delta y = y_{k+1} - y_k$, 直到 $\max(|\Delta x|, |\Delta y|) < \epsilon$ 为止。

1.4 求解线性方程组的直接法

线性方程组具有一形式

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \tag{1.11}$$

可以写成矩阵形式 $Ax = b$,$A = (a_{ij})$ 为系数矩阵,n 维列向量 $x = (x_1, x_2, \ldots, x_n)^T$ 为解向量，m 维列向量 $b = (b_1, b_2, \ldots, b_m)^T$ 为常数向量。当 A 是可逆方阵时，线性方程组存在唯一解，当 n 较小时，可用 Cramer 法则给出解 $x_i = \frac{\Delta_i}{\Delta}$, 其中 Δ 是 A 的行列式，Δ_i 是把 A 中第 i 列替换成 b 之后的行列式。

1.4.1 高斯消元法

当 A 是对角阵时，线性方程组的解为：$x_i = \frac{b_i}{a_{ii}}$, 当 A 为下三角阵时：

$$\begin{cases} a_{11}x_1 = b_1 \\ \vdots \\ a_{i1}x_1 + \cdots + a_{ii}x_i = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{1.12}$$

可依次解得 $x_1 = \frac{b_1}{a_{11}}, x_2 = \frac{b2 - a_{21}x_1}{a_{22}}, \ldots, x_i = \frac{b_i - a_{i1}x_1 - \ldots a_{i,i-1}x_{i-1}}{a_{ii}}, \ldots$. 该解法称为回代法。同理可得上三角阵的解法，其计算复杂度为 n^2.

1. 高斯顺序消元法：对增广矩阵 (A, b) 进行行初等变换，得到三角形线性方程组，然后用回代法求解。

2. 高斯列主元消元法：第一步在第一列元素中选取绝对值最大的元素，$max\{|a_{i1}|\} = |a_{m1}|$, 交换第一行和第 m 行，再作化简 $\{a_{21}, a_{31}, \ldots, a_{n1}\}$ 为 0 的初等变换，对于每个 k 在消元前，选出绝对值最大的元素 $a_{mk}^{(k-1)}$，对 k 和 m 行进行交换。

1.4.2 直接分解法

解方程组：$Ax = b$, 首先分解 $A = LU$, 则方程组可转化为解线性方程组 $Ly = b, Ux = y$, 分解后只需 $\mathcal{O}(n^2)$ 运算量。

特殊线性方程组:三对角方程组可用 LU 分解,对称正定矩阵可用 LDL^T 分解。

1.5 求解线性方程组的迭代法

由 $Ax = b$ 构造等价方程组，拆分 $A = N - P$, 设 N 可逆，得到同解方程组 $X = N^{-1}PX + N^{-1}b$, 令 $M = N^{-1}P, g = N^{-1}b, X = MX + g$, 构造迭代式：$X^{(k)} = MX^{(k-1)} + g$, 任取初始向量 $X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(n)})^T$ 代入迭代式中，得到序列 $\{X^{(k)}\}$, 若序列收敛，设极限为 $X^\star$，即为方程组 $Ax = b$ 的解。

1.5.1 简单 (Jacobi) 迭代

把方程组 $AX = b$ 中 $a_{ii}x_i$ 留在方程左边，其余各项挪到方程右边，得到迭代式:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - \cdots - a_{(1n)}x_n^{(k)}) \\ x_2^{(k+1)} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k)} - \cdots - a_{(2n)}x_n^{(k)}) \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{(n,n-1)}x_{n-1}^{(k)}) \end{cases} \tag{1.13}$$

1.5.2 Gauss-Seidel 迭代矩阵

设 A=D+L+U，写成矩阵表达式：$AX = (D + L + U)X = (D + L)X + UX = b$, 构造迭代式 $(D + L)X^{(k+1)} = -UX^{(k)} + b$, 有 $X^{(k+1)} = -(D+L)^{-1}UX^{(k)} + (D+L)^{-1}b$, 令 $S = -(D+L)^{-1}U, f = (D+L)^{-1}b$, 有 $X^{(k+1)} = SX^{(k)} + f$, 称 S 为 $Gauss - Seidel$ 迭代矩阵。

1.5.3 松弛迭代

令 $S_w = (I+wD^{-1}L)^{-1}[(1-w)I-wD^{-1}f], \quad f = w(I+wD^{(-1)}L)^{-1}D^{-1}b$, 则松弛因子 1 为 w 的迭代矩阵为 $X^{(k+1)} = S_wX^{(k)} + f$.

1.6 数值积分和数值微分

1.6.1 牛顿-柯特斯数值积分

定积分是黎曼和的极限，即 $\int_a^b f(x)dx = \lim_{\Delta x_i \to 0}(\sum_{i=0}^{n-1} f(x_i)\Delta x_i)$, 在数值积分公式中，用有限项的和近似上面的极限，记：

$$I(f) = \int_a^b f(x)dx, \quad I_n(f) = \sum_{i=0}^{n} \alpha_i f(x_i) \tag{1.14}$$

1. 插值型数值积分:对给定被积函数 $f(x)$ 在 [a,b] 上的点列 $\{(x_i, f(x_i)), i = 0, 1, \ldots, n\}$, 作 Langrange 插值多项式 $L_n(x)$, 有

$$\int_a^b f(x)dx \approx \int_a^b L_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x)f(x_i)dx = \sum_{i=0}^{n}[\int_a^b l_i(x)dx]f(x_i) \tag{1.15}$$

记 $\alpha_i = \int_a^b l_i(x)dx$, 则有 $I_n(f) = \int_a^b L_n(x)dx = \sum_{i=0}^{n} \alpha_i f(x_i)$.

2. 牛顿-柯特斯积分：对积分区间等分，记步长 $h = \frac{b-a}{n}$, 取等分点 $\{x_i = a + ih, i = 0, 1, \ldots, n\}$ 为数值积分节点，构造 Langrange 插值多项式 $L_n(x)$,$\int_a^b f(x)dx \approx \int_a^b L_n(x)dx$. 其又主要分为梯形积分 $I(f) = \int_a^b f(x)dx \approx \frac{b-a}{2}[f(a) + f(b)]$ 和辛普森积分:$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]$

1.6.2 复化数值积分

1. 复化梯形积分:把积分区间分割成若干个小区间,在每个区间 $[x_i, x_{i+1}]$ 上用梯形积分公式, 再将这些小区间上的数值积分累加起来，称为复化梯形公式。

2. 复化辛普森积分：把积分区间分成偶数 2m 等份，记 $n = 2m$, 其中 n+1 是节点总数，m 是积分子区间的总数，记 $h = \frac{b-a}{h}$, 在每个子区间上用辛普森数值积分公式。

3. 自动控制误差的复化积分：截断误差随分点 n 的增长而减少，控制计算的精度也就是确定分点数 n. 在计算中用 $|T_{2n}(f) - T_n(f)| < \epsilon$ 作为控制，在计算中构造序列 $T_n, T_{2n}, T_{4n}, \ldots$, 直到 $|T_{2m} - T_m| < \epsilon$ 时停止计算，取 $I(f) \approx T_{2m}(f)$.

1.6.3 重积分计算

二重积分是用化为累次积分的方法计算的。二重数值积分也是累次计算数值积分。考虑矩形域上的二重积分:$\int_a^b \int_c^d f(x, y)dxdy$, 对区间 [a,b] 和 [c,d]

分别选取正整数 m 和 n, 在 x 和 y 轴上分别有步长 $h = \frac{b-a}{m}$, $k = \frac{d-c}{n}$, 则

$$\int_a^b \int_b^c f(x,y)dxdy = hk \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} f(x_i, y_j). \tag{1.16}$$

1.6.4 数值微分

1. 差商:向前差商:$f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$, 向后差商 $f'(x_0) =\approx \frac{f(x_0)-f(x_0-h}{h}$, 中心差商 $f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)-f(x_0-h)}{2h}$, 其误差为 $R(x) = \mathcal{O}(h^2)$.

2. 插值型数值微分对于给定的 f(x) 的函数表，建立插值多项式 $L(x)$, 用插值函数 $L(x)$ 的导数近似函数 $f(x)$ 的导数：

$$f(x) \approx L_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) \quad f' \approx L'_n = \sum_{i=0}^{n} l'_i(x) f(x_i). \tag{1.17}$$

1.7 常微分方程数值解

我们讨论常微分方程如下形式：

$$\begin{cases} y'(x) = f(x,y), \\ y(a) = y_0 \end{cases} \quad a \le x \le b \tag{1.18}$$

首先对区间 [a,b] 作一个剖分，确定特定点列 $\{x_n : n = 1, 2, \ldots, m\}$. 用函数在这组点列上的函数值 $\{y(x_n), n = 1, 3, \ldots, m\}$ 描述函数 y(x).

1.7.1 基于数值微商的 Euler 公式

对求解区间做等距分割的剖分，记步长为 $h = \frac{b-a}{m}$, 则 $\{x_n = a+nh, n = 0, 1, \ldots, n\}$. 向前一阶差商 $y' \approx \frac{y(x_{n+1})-y(x_n)}{h}$, 又有微分方程 $y'(x_n) = f(x_n, y(x_n))$, 于是 $f(x_n, y(x_n)) \approx \frac{y(x_{n+1})-y(x_n)}{h}$, 得到向前 Euler 公式：

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{1.19}$$

类似也有向后差商近似:$y_{n+1} = y_n + hf(x_{x+1}, y_{n+1})$ 和中心差商近似 $y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$.

1.7.2 基于数值积分的近似公式

对常微分方程 $\frac{dy}{dx} = f(X, Y)$ 两边在区间 $[X_n, x_{n+1}]$ 上积分再移项

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))dx \tag{1.20}$$

取数值积分为梯形近似公式:

$$\begin{aligned}\int_{x_n}^{x_{n+1}} f(x,y)dx &\approx \frac{1}{2}(x_{n+1}-x_n)(f(x_{n+1},y(x_{n+1}))+f(x_n,y(x_n))) \\ &= \frac{h}{2}(f(x_n,y(x_n))+f(x_{n+1},y(x_{n+1})))\end{aligned} \tag{1.21}$$

得到梯形公式：$y_{n+1}=y_n+\frac{h}{2}(f(x_n,y_n)+f(y_{n+1},y_{n+1}))$.

1.7.3 Runge-Kutta 方法

做 $y(x+h)$ 在点的 Taylor 展开：$y(x+h)=y(x)+hy'(x)+\frac{h^2}{2!}y''(x)+\cdots+\frac{h^p}{p!}y^{(p)}(x)+\frac{h^{(p+1)}}{(p+1)!}y^{(p+1)}(x+\theta h)$, 记最后一项为 $T_{(p+1)}$, 则由微分方程有

$$\begin{aligned}&y'(x_n)=f(x_n,y(x_n)) \\ &y''(x_n)=f_x\ x_n,y(x_n))+f_y(x_n,y(x_n))f(x_n,y(x_n)) \\ &\cdots\end{aligned} \tag{1.22}$$

取 $p=1$ 可得 Euler 公式，取 $p=2$，上式可写成

$$y_{n+1}=y_n+h\{f(x_n,y_n)+\frac{h}{2}[f_x(x_n,y_n)+f_y(x_n,y_n)f(x_n,y_n)]\}. \tag{1.23}$$

但是其不便于计算，在 Runbge-Kutta 方法中，用 f(x,y) 在点 $(x_n,y(x_n))$ 和它附近的点 $(x_n+ah,y(x_n)+bhf(x_n,y(x_n)))$ 上的值的线性组合逼近上式的主体，即用 $c_1f(x_n,y(x_n))+c_2f(x_n+ah,y(x_n)+bhf(x_n,y(x_n)))$ 逼近，得到数值公式:

$$y_{n+1}=y_n+h[c_1f(x_n,y_n)+c_2f(x_n+ah,y_n+bhf(x_n,y_n))]. \tag{1.24}$$

1.7.4 常微分方程组的数值解法

将 m 个一阶微分方程组写成常微分方程初值形式：

$$\begin{cases}\frac{dy_1}{dx}=f_1(x,y_1,y_2,\ldots,y_m), \\ \frac{dy_2}{dx}=f_2(x,y-1\ y_2,\ldots,y_m), \\ \vdots \\ \frac{dy_m}{dx}=f_m(x,y_1,y_2,\ldots,y_m), \\ y_1(a)=b_1, \\ y_2(a)=b_2, \\ \vdots \\ y_m(a)=b_m,\end{cases} \quad a\le x\le b \tag{1.25}$$

写成向量形式：

$$\begin{cases} \frac{dY}{dx} = F(x, Y) \\ Y(a) = b \end{cases} \tag{1.26}$$

解常微分方程的 Euler 方法，Runge-Kutta 方法等都可以平行地应用到常微分方程组的数值解中。

1.8 计算矩阵的特征值和特征向量

1.8.1 幂法

任取初始向量 $X^{(0)}$, 进行迭代计算 $X^{(k+1)} = AX^{(k)}$, 得到迭代序列 $\{X^{(k)}\}$, 分析 $X^{(k+1)}$ 与 $X^{(k)}$ 之间的关系，得到 A 的按模最大特征值以及特征向量的近似解。

1.8.2 反幂法

反幂法是计算矩阵按模最小的特征值以及相应的特认向量的数值方法。设矩阵 A 可逆，λ 和 v 分别是 A 的特征值以及相应地特征向量，对 $Av = \lambda v$ 两边同乘 A^{-1}, 得 $A^{-1}v = \frac{1}{\lambda}v$。可见 A 的按模最小的特征值与 A^{-1} 的按模最大的特征值互为倒数，所以我们可以计算 A^{-1} 模最大的特征值来得到 A 的按模最小的特征值。

1.8.3 实对称阵的 Jacobi 方法

构造一系列特殊形式的正交矩阵 Q 对 A 做正交相似变换，使矩阵的非对角线比重逐渐减小，直到每个非对角元无足轻重时，其对角元素可看作 A 的特征值。

1.8.4 QR 方法

若 A 为 n 阶实矩阵，存在正交阵 Q，上三角阵 R，使得：$A = QR$, 记 $A_1 = A$, 对 A_1 作 QR 分解，$A_1 = Q_1R_1$, 记 $A_2 = R_1Q_1$, 对 A_2 作 QR 分解，$A_2 = Q_2R_2$, 记 $A_3 = R_2Q_2$，若已有 $A_k = Q_kR_k$, 记 $A_{k+1} = R_kQ_k$, 矩阵序列 $\{A_k\}$ 为相似正交序列，记 A_k 的元素 $(a_{ij}^k)_{n\times n}$, 若 A 满足一定条件，则有 $a_{ij}^{(k)} \to 0$, 当 $k \to \infty, 1 \le i < j \le n; a_{ii}^(k) \to \lambda_i$, 当 $k \to \infty, i = 1, 2, \ldots, n. \lambda_i$ 即为 A 的特征值。

Problem 2

textSummary the models in physics

2.1 经典粒子的多体运动

考虑 N 个有相互作用的经典粒子，它们的运动由牛顿运动方程描述：

$$m_i\ddot{X}_i = F(X_1,\ldots,X_i,\ldots,X_N;\dot{X}_1,\ldots,\dot{X}_i,\ldots,\dot{X}_N;t), \quad \text{for i =1,2,..., N} \tag{2.1}$$

2.2 混沌

混沌理论涉及其行为原则上可以预测的确定性系统。混沌系统在一段时间内是可预测的，然后“似乎”变得随机。可以有效预测混沌系统行为的时间量取决于三件事：预测中可以容忍多少不确定性，可以测量其当前状态的准确度以及取决于系统动态的时间尺度，称为李雅普诺夫时间。在混沌系统中，预测的不确定性呈指数增长随着时间的流逝。因此，从数学上讲，将预测时间加倍超过预测中比例不确定性的平方。这意味着，在实践中，不能在超过李雅普诺夫时间的两到三倍的间隔内做出有意义的预测。当无法做出有意义的预测时，系统就会显得随机。

2.2.1 混沌同步

混沌同步是两个或多个耗散混沌系统耦合时可能发生的现象。由于混沌系统附近轨迹的指数发散，两个混沌系统同步演化可能会令人惊讶。然而，耦合或驱动的混沌振荡器的同步是一种通过实验很好地建立并在理论上很好理解的现象。

2.3 牛顿方程在其他方面的应用

2.3.1 群聚

群聚是当一群鸟类（称为群）正在觅食或飞行时所表现出的行为。群聚行为的基本模型由三个简单规则控制：分离——避免邻居拥挤（短程排斥）对齐——转向邻居的平均航向凝聚力——转向邻居的平均位置（远距离吸

引）通过这三个简单的规则，鸟群以极其逼真的方式移动，创建复杂的运动和交互。

2.3.2 分子动力学

分子动力学 (MD) 是一种分析原子和分子物理运动的计算机模拟方法。原子和分子被允许在一段固定的时间内相互作用，从而了解系统的动态演化。原子和分子的轨迹是通过数值求解相互作用粒子系统的牛顿运动方程来确定的，其中粒子之间的力及其势能通常使用原子间势或分子力学力场来计算。对于具有坐标的 X 和速度 V 的 N 个粒子系统，其运动可由牛顿运动方程描写：

$$\begin{aligned} F(X) &= -\nabla U(X) = M\dot{V}(t) \\ V(t) &= \dot{X}(t). \end{aligned} \tag{2.2}$$

其中势能函数 $U(x)$ 是粒子坐标的函数。

2.3.3 药物设计

药物设计是根据生物靶点的知识寻找新药物的过程。药物最常见的是有机小分子，其可以激活或抑制生物分子 (如蛋白质) 的功能。药物设计要设计出与生物分子结构相适应的分子，使之与生物分子容易结合而发挥作用。药物设计涉及到量子力学和基于牛顿方程的分子动力学，应用这些方法来预测药物分子与生物分子的结合过程。

2.4 麦克斯韦方程组

麦克斯韦方程组的微分形式是:

$$\begin{aligned} \nabla \cdot E &= \frac{\rho}{\epsilon_0} \\ \nabla \cdot B &= 0 \\ \nabla \times E &= -\frac{\partial B}{\partial t} \\ \nabla \times B &= \mu_0(J + \epsilon_0 \frac{\partial E}{\partial t}). \end{aligned} \tag{2.3}$$

理论求解复杂系统的麦克斯韦方程组几乎不可能，通常是离散化求数值解。

2.5 统计物理中的配分函数

对于微整则系综，其配分函数是：$Z_{NVE} = \int \delta[H(q,p) - E]d\Omega = \Omega'(E)$, 正则系综的配分函数为:$Z_{NVT} = \int exp(-\beta H) Z_{NVE} dE$, 巨正则系综的配分函数是 $Z_{\mu VT} = \sum_N \frac{1}{N!} \int \{-\beta[\mu N + H(q,p)]\} d\Omega$.

2.6 哈密顿量本征值

一般量子体系的哈密顿量写成 $H = -\frac{\hbar^2}{2m} + V(r)$，只有少数体系，例如氢原子，简谐振子可以理论求解，大部分体系都需要数值求解，一般是把波函数离散化，把哈密顿量写成矩阵形式进行求解：或者利用打靶法求解。在多体模式下一般不能精确求解，有许多近似方法求解多体问题，例如化学上的 Hartee Fock 方法，Confifuration Interaction，Coupled Cluster theory 等，凝聚态物理常用的密度泛函方法，还有格林函数方法，蒙特卡洛模拟等。

2.7 随机过程

在建立物理的实在的模型时，可以暂时只考虑一个小的物理子系统，这个子系统可以受系统外部的力，场和碰撞的影响。为了避免描述大的物理系统，我们用一个适当选择的偶然的力，场和碰撞来替代外部对子系统施加的影响。

2.7.1 布朗运动

布朗颗粒的运动是大量碰撞的涨落的结果，是一种完全无规则的随机运动。设其所受的阻力为 $-\alpha v$, 所受的涨落力为 F, 则其运动方程为：

$$m\ddot{r} = F - \alpha\dot{r} \tag{2.4}$$

这就是 Langevin 提出的布朗运动方程。

2.7.2 扩散过程

扩散是由于粒子浓度梯度的存在 $\Delta\rho$ 形成粒子往低密度区域迁移的趋势，单位时间内通过某一方向垂直截面的粒子数即为粒子流密度，$J = -D\Delta\rho$, 由粒子数守恒的连续性方程 $\partial\rho/\partial t + \Delta \cdot J = 0$ 可得扩散方程：

$$\partial p/\partial t = D\Delta^2 p \tag{2.5}$$

其中 $p(r,t)dv$ 是粒子在 t 时刻处于位置 r 附近体积 dv 之间的概率。

2.8 多体相互作用求解基态能量

2.8.1 变分法

对于一个哈密顿量为 H 的体系。考虑其处于态 $|\psi\rangle$, 则能量为：$E = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$, 根据变分原理，体系基态能量总是 $E_0 \leq E$, 因此我们的目标就是寻找使得 E 最小的态，即为基态。我们可以先假设一个试探态 $\psi(\theta)$，然后求取能量，根据能量的梯度更新参数，期望找到基态或者接近基态的态。

2.8.2 密度泛函理论

在 DFT 中，关键变量是电子密度 $n(r)$，对于归一化的 Ψ，其由下式给出:

$$n(r) = N \int d^3r_2 \cdots \int d^3r_N \Psi^*(r, r_2, \ldots, r_N)\Psi(r, r_2, \ldots, r_N). \tag{2.6}$$

对于给定的基态密度 $n_0(r)$,原则上可以计算相应的基态波函数 $\Psi_0(r_1, \ldots, r_N)$, 基态能量是 n_0 的函数：

$$E_0 = E[n_0] = \langle\Psi[n_0]|\, T + V + U \,|\Psi[n_0]\rangle\,. \tag{2.7}$$

2.9 平均场论

平均场理论 (又名或自洽场理论) 通过研究一个更简单的模型来研究高维随机模型的行为，该模型通过对自由度进行平均处理来近似原来的场。其平均场由下式给出：

$$h_i^{MF}(\xi) = \sum_{j|(i,j)\in P} Tr_j V_{i,j}(\xi_i, \xi_j) P_0^{(j)}(\xi_j) \tag{2.8}$$

其中 $P_0^{(i)}(\xi_i) = \frac{1}{Z_0} e^{-\beta h_i^{MF}(\xi_i)}$.

2.10 蒙克卡罗模拟

2.10.1 经典蒙特卡洛模拟

经典的蒙特卡罗方法可用于模拟随机行走，粒子输运，多粒子系统的配分函数计算，具体的有 Ising 模型，$E = -\sum_{\langle i,j|i,j\rangle} J_{ij}\sigma_i\sigma_j - \mu_B H \sum_{i=1} \sigma_i$.

2.10.2 量子蒙特卡罗模拟

按照 Feynman 的思想，微观粒子取各种可能的路径，各条路径对传播子的贡献是相同的，但是由于每条路径给出的几率幅的相位不同，而发现粒子在终点的几率将因各条路径的相位叠加造成微观粒子的干涉效果。

$$\Psi(r'',t'') = \int dr' G(r'',t'';r',t')\Psi(r',t'), \quad G(r'',t'';r',t') = Z^{-1}dr_1 \dots dr_{n-1} exp[iS/\hbar]. \tag{2.9}$$

除了路径积分，蒙特卡洛方法还可用于前面提到的变分法。

Problem 3

Analytical programming language 以下为具体总结，在 mathematica 的练习在 P3.nb 文件

3.1 基本运算

3.1.1 四则运算

b+c 加; a-b 减; a b c 或 a*b*c 乘; a/b 除; -a 负号; $a \wedge b$ 次方;

3.1.2 四种括号

(term) 圆括号，括号内的 term 先计算；f[x] 方括号，内放函数的引数；{x,y,z} 大括号或串列括号，内放串列的元素；p[[i]] 或 Part[p,i] 双方括号，p 的第 i 项元素；p[[i,j]] 或 Part[p,i,j] p 的第 i 项第 j 个元素；

3.1.3 关系运算子

a==b 等于 a>b 大于 a>=b 大于等于 a<b 小于 a<=b 小于等于 a!=b 不等于

3.1.4 Mathematica 数字的形式

256 整数; 2.56 实数; 11/35 分数; 2+6I 复数;

3.1.5 常用的数学常数

Pi 圆周率; =3.141592654…; E 尤拉常数，e=2.71828182…; Degree 角度转换弧度的常数，Pi/180 ; I 虚数，其值为 $\sqrt{-1}$; Infinity 无限大;

3.1.6 指定之前计算结果的方法

% 前一个运算结果; %% 前二个运算结果; %%…%(n 个%) 前 n 个运算结果; %n 或 Out[n] 前 n 个运算结果;

3.1.7 复数的运算指令

a+bI 复数; Conjugate[a+bI] 共轭复数; Re[z], Im[z] 复数 z 的实数/虚数部分; Abs[z] 复数 z 的大小或模数 (Modulus) ; Arg[z] 复数 z 的幅角 (Argument) ;

3.1.8 Mathematica 输出的控制指令

expr1; expr2; expr3 做数个运算，但只印出最后一个运算的结果; expr1; expr2; expr3; 做数个运算，但都不印出结果; expr; 做运算，但不印出结果;

3.1.9 常用数学函数

Sin[x],Cos[x],Tan[x],Cot[x],Sec[x],Csc[x] 三角函数，其引数的单位为弧度; Sinh[x],Cosh[x],Tanh[x],…双曲函数; ArcSin[x],ArcCos[x],ArcTan[x] 反三角函数; ArcCot[x],ArcSec[x],ArcCsc[x] ; ArcSinh[x],ArcCosh[x],ArcTanh[x], …反双曲函数;

Sqrt[x] 根号; Exp[x] 指数; Log[x] 自然对数; Log[a,x] 以 a 为底的对数; Abs[x] 绝对值; Round[x] 最接近 x 的整数; Floor[x] 小于或等于 x 的最大整数; Ceiling[x] 大于或等于 x 的最小整数; Mod[a,b] a/b 所得的馀数; n! 阶乘; Random[] 0 至 1 之间的随机数; Max[a,b,c,...]，Min[a,b,c,…] a,b,c,…的极大/极小值

3.1.10 数值设定

x=a 将变数 x 的值设为 a ; x=y=b 将变数 x 和 y 的值均设为 b ; x=. 或 Clear[x] 除去变数 x 所存的值;

变数使用的一些法则: xy 中间没有空格，视为变数 xy ; x y x 乘上 y ; 3x 3 乘上 x ; x3 变数 x3 ; x∧2y 为 x∧2 y 次方运算子比乘法的运算子有较高的处理顺序;

3.2 其他运算

3.2.1 四个处理代数指令

Expand[expr] 将 expr 展开;Factor[expr] 将 expr 因式分解;Simplify[expr] 将 expr 化简成精简的式子；FullSimplify[expr] Mathematica 会尝试更多的化简公式，将 expr 化成更精简的式子；

3.2.2 多项式/分式转换函数

ExpandAll[expr] 把算式全部展开；Together[expr] 将 expr 各项通分在并成一项；Apart[expr] 把分式拆开成数项分式的和；Apart[expr,var] 视 var 以外的变数为常数，将 expr 拆成数项的和；Cancel[expr] 把分子和分母共同的因子消去；

3.2.3 分母/分子的运算

Denominator[expr] 取出 expr 的分母；Numerator[expr] 取出 expr 的分子；ExpandDenominator[expr] 展开 expr 的分母；ExpandNumerator[expr] 展开 expr 的分子；

3.2.4 多项式二种转换函数

Collect[expr,x] 将 expr 表示成 x 的多项式，如 Collect[expr,{x,y,⋯}] 将 expr 分别表示成 x,y,⋯的多项式；FactorTerms[expr] 将 expr 的数值因子提出，如 4x+2=2(2x+1) ；FactorTerms[expr,x] 将 expr 中把所有不包含 x 项的因子提出；FactorTerms[expr,{x,y,⋯}] 将 expr 中把所有不包含 {x,y,...} 项的因子提出；

3.2.5 函数和指数运算

TrigExpand[expr] 将三角函数展开；TrigFactor[expr] 将三角函数所组成的数学式因式分解；TrigReduce[expr] 将相乘或次方的三角函数化成一次

方的基本三角函数之组合；ExpToTrig[expr] 将指数函数化成三角函数或双曲函数；TrigToExp[expr] 将三角函数或双曲函数化成指数函数；

3.2.6 复数、次方乘积之展开

ComplexExpand[expr] 假设所有的变数都是实数来对 expr 展开；ComplexExpand[expr,{x,y,⋯}] 假设 x,y,.. 等变数均为复数来对 expr 展开；PowerExpand[expr] ；

3.2.7 项次、系数与最高次方

Coefficient[expr,form] 于 expr 中 form 的系数；Exponent[expr,form] 于 expr 中 form 的最高次方；Part[expr,n] 或 expr[[n]] 在 expr 项中第 n 个项；

3.2.8 代换运算子

expr/.x->value 将 expr 里所有的 x 均代换成 value ；expr/.{x->value1,y->value2,⋯} 执行数个不同变数的代换；expr/.{{x->value1},{x->value2},⋯} 将 expr 代入不同的 x 值；expr//.{x->value1,y->value2,⋯} 重复代换到 expr 不再改变为止；

3.2.9 求解方程式的根

Solve[lhs==rhs,x] 解方程式 lhs==rhs，求 x ；Nsolve[lhs==rhs,x] 解方程式 lhs==rhs 的数值解；Solve[{lhs1==rhs1,lhs2==rhs2,⋯},{x,y,⋯}] 解联立方程式，求 x,y,⋯；NSolve[{lhs1==rhs1,lhs2==rhs2,⋯},{x,y,⋯}] 解联立方程式的数值解；FindRoot[lhs==rhs,{x,x0}] 由初始点 x0 求 lhs==rhs 的根；

3.2.10 缩短输出指令

expr//Short 显示一行的计算结果; Short[expr,n] 显示 n 行的计算结果; Command; 执行 command，但不列出结果;

3.2.11 查询物件

?Command 查询 Command 的语法及说明; ??Command 查询 Command 的语法和属性及选择项; ?Aaaa* 查询所有开头为 Aaaa 的物件;

3.2.12 函数定义、查询与清除

f[x_]= expr 立即定义函数 f[x] ; f[_]:= expr 延迟定义函数 f[x] ; f[x_,y_,…] 函数 f 有两个以上的引数; ?f 查询函数 f 的定义; Clear[f] 或 f=. 清除 f 的定义; Remove[f] 将 f 自系统中清除掉; 含有预设值的 Pattern : a_+b_. b 的预设值为 0，即若 b 从缺，则 b 以 0 代替; x_ y_ y 的预设值为 1 ; x_∧y_ y 的预设值为 1 ; 条件式的自订函数: lhs:=rhs/;condition 当 condition 成立时，lhs 才会定义成 rhs ;

3.2.13 If 指令

If[test,then,else] 若 test 为真，则回应 then，否则回应 else ;

If[test,then,else,unknow] 同上，若 test 无法判定真或假时，则回应 unknow

3.2.14 极限

Limit[expr,x->c] 当 x 趋近 c 时,求 expr 的极限; Limit[expr,x->c,Direction->1] ; Limit[expr,x->c,Direction->-1] ;

3.2.15 微分

D[f,x] 函数 f 对 x 作微分; D[f,x1,x2,…] 函数 f 对 x1,x2,…作微分; D[f,x,n] 函数 f 对 x 微分 n 次; D[f,x,NonConstants->y,z,…] 函数 f 对 x 作微分，将 y,z,…视为 x 的函数;

3.2.16 全微分

Dt[f] 全微分 df ; Dt[f,x] 全微分; Dt[f,x1,x2,…] 全微分; Dt[f,x,Constants->c1,c2,…] 全微分，视 c1,c2,…为常数;

3.2.17 不定积分

Integrate[f,x] 不定积分 $\int f\,dx$;

3.2.18 定积分

Integrate[f,{x,xmin,xmax}] 定积分; Integrate[f,{x,xmin,xmax},{y,ymin,ymax}] 定积分;

3.2.19 数列之和与积

Sum[f,{i,imin,imax}] 求和; Sum[f,{i,imin,imax,di}] 求数列和，引数 i 以 di 递增; Sum[f,{i,imin,imax},{j,jmin,jmax}] ;

Product[f,{i,imin,imax}] 求积; Product[f,{i,imin,imax,di}] 求数列之积，引数 i 以 di 递增; Product[f,{i,imin,imax},{j,jmin,jmax}];

3.2.20 函数之泰勒展开式

Series[expr,{x,x0,n}] 对 expr 于 x0 点作泰勒级数展开至 (x-x0)n 项; Series[expr,{x,x0,m},{y,y0,n}] 对 x0 和 y0 展开;

3.2.21 逻辑运算子

!p not ; p||q||…or ; p&&q&&…and ; Xor[p,q,…] exclusive or ; LogicalExpand[expr] 将逻辑表示式展开;

3.3 绘图

3.3.1 基本二维绘图指令

Plot[f,x,xmin,xmax] 画出 f 在 xmin 到 xmax 之间的图形; Plot[f1,f2,…,x,xmin,xmax] 同时画出数个函数图形; Plot[f,x,xmin,xmax,option->value] 指定特殊的绘图选项，画出函数 f 的图形;

3.3.2 Plot 几种指令

选项预设值说明 AspectRatio 1/GoldenRatio 图形高和宽之比例,高/宽; Axes True 是否把坐标轴画出; AxesLabel Automatic 为坐标轴贴上标记，若设定为 AxesLabel->?ylabel?，则为 y 轴之标记。若设定为 AxesLabel->?xlabel?,?ylabel? ，则为 x 轴,y 轴的标记; AxesOrigin Automatic 坐标轴的相交的点;

DefaultFont $DefaultFont 图形里文字的预设字型; Frame False 是否将图形加上外框; FrameLabel False 从 x 轴下方依顺时针方向加上图形外框的标记; FrameTicks Automatic (如果 Frame 设为 True) 为外框加上刻度, None 则不加刻度; GridLines None 设 Automatic 则于主要刻度上加上网格线;

PlotLabel None 整张图之图名; PlotRange Automatic 指定 y 方向画图的范围;

Ticks Automati 坐标轴之刻度，设 None 则没有刻度记号出现;

※“Automatic、None、True、False”为 Mathmatica 常用的选项设定，其代表意义分别为“使用内部设定、不包含此项、作此项目、不作此项目”;

3.3.3 串列绘图

ListPlot[{y1,y2,⋯}] 画出 {1,y1},{2,y2},⋯的点; ListPlot[{{x1,y1},{x2,y2},⋯}] 画出 {x1,y1},{x2,y2},⋯的点; ListPlot[{{x1,y1},{x2,y2},⋯},PlotJoined->True] 把画出来的点用线段连接;

3.3.4 绘图颜色的指定

Plot[{f1,f2,⋯},{x,xmin,xmax}, PlotStyle->{RGBColor[r1,g1,b1],RGBColor[r2,g2,b2],⋯}] ; 彩色绘图

Plot[{f1,f2,⋯},{x,xmin,xmax}, PlotStyle->{GrayLevel,GrayLevel[j],⋯}] ; 灰阶绘图

3.3.5 图形处理指令

Show[plot] 重画一个图; Show[plot1,plot2,⋯] 将数张图并成一张; Show[plot,option->opt] 加入选项;

3.3.6 图形之排列

Show[GraphicsArray[{plot1,plot2,⋯}]] 将图形横向排列; Show[GraphicsArray[{„⋯}]] 将图形垂直排列; Show[GraphicsArray[{{plot1,plot2,⋯},⋯}]] 将图形成二维矩阵式排列;

二维参数图: ParametricPlot[{f1,f2},{t,tmin,tmax}] ;

参数绘图 ParametricPlot[{{f1,f2},{g1,g2},⋯},{t,tmin,tmax}] ;

同时绘数个参数图 ParametricPlot[f1,f2,t,tmin,tmax,AspectRatio->Automatic] ; 保持曲线的真正形状，即 x,y 坐标比为 1：1 ;

等高线图: ContourPlot[f,x,xmin,xmax,y,ymin,ymax] 于指定范围之内画出 f 的等高线图;

3.3.7 ContourPlot 选项

选项预设值说明 ColorFunction Automatic 上色的预设值为灰阶，选 Hue 则为系列色彩; Contours 10 等高线的数目。设 Contours->{z1,z2,⋯} 则指定等高值为 z1,z2,⋯; ContourShading True Contour 的上色，选 False 则不上色; PlotRange Automatic 高度 z 值的范围，也可指定 {zmin,zmax} ;

Problem 4

Tricks in numerical calculation

4.1 problem 4_1

计算:

$$z = \prod_{i=2}^{L} x_i + \prod_{i=1}^{N} y_i \tag{4.1}$$

取 L=10000，N=8000,10000,12000; 先取随机数的范围是 (0,1) 我们发现最后的结果太小，无法计算，但是我们可以先对每个随机数 × 10, 最后除以

$10^N(10^L)$. 计算得

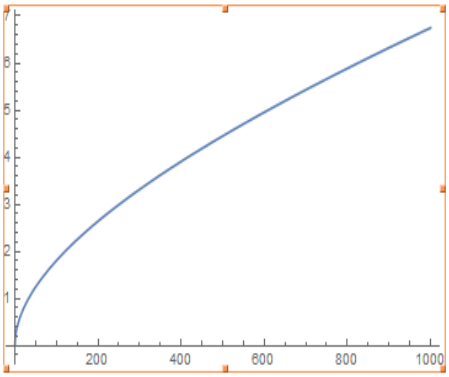
$$\begin{aligned}
&\prod_{i=1}^{10000} x_i = 4.39730003856529 \times 10^{5617}/10^{10000} = 4.39730003856529 \times 10^{-4383} \\
&\prod_{i=1}^{8000} y_i = 5.03090425290087 \times 10^{4528}/10^{8000} = 5.03090425290087 \times 10^{-3472} \\
&\prod_{i=1}^{10000} y_i = 9.1027680863848 \times 10^{5612}/10^{10000} = 9.1027680863848 \times 10^{-4388} \\
&\prod_{i=1}^{12000} y_i = 2.17775099860717 \times 10^{6886}/10^{12000} = 2.17775099860717 \times 10^{-5114}
\end{aligned} \tag{4.2}$$

可以看到当 N=8000 时，$\sum_{i=1}^{N} y_i$ 远大于 $\sum_{i=1}^{L} x_i$, 可以近似认为 $z = \sum_{i=1}^{N} y_i = 5.03090425290087 \times 10^{-3472}$，当 N=10000 时，两者有可比性，此时 $z = 4.39739106624616 \times 10^{-4383}$. 当 N=12000 时，$\sum_{i=1}^{L} x_i$ 远大于 $\sum_{i=1}^{N} y_i$, 可以近似认为 $z = \sum_{i=1}^{L} x_i = 4.39730003856529 \times 10^{-4383}$.

接下来讨论随机数范围取 (0,10) 的情况，此时只考虑 L=N=10000 的情况，因为 N，L 不同时，乘积的值相差过大。计算得 $z = 8.8403188120049 \times 10^{5656}$ 此时的 z 值大约是之前取 (0,1) 范围的 10^{1039} 倍，接近 10^{10000}. 代码在文档 p4_1.nb 里面

4.2 problem 4_2

画出函数图：其中代码在文档 p4_2.nb



4.3 problem 4_3

因为误差随着 x 单增，我们只需要使得展开误差在 $x = 1000$ 小于 10^{-10} 即可，代码在 p4_3.nb，结果如下：

$$\frac{\sqrt{x}}{6} + \frac{x}{720} + \frac{x^{3/2}}{362\,880} + \frac{x^2}{479\,001\,600} + \frac{x^{5/2}}{1\,307\,674\,368\,000} + \frac{x^3}{6\,402\,373\,705\,728\,000} + \frac{x^{7/2}}{51\,090\,942\,171\,709\,440\,000} + \frac{x^4}{620\,448\,401\,733\,239\,439\,360\,000} + O[x]^{25/6}$$

4.4 problem 4_4

首先反解出 $F(n) = \frac{n!e^n}{n^n\sqrt{2\pi n}}$, 又知道 $F(n) = 1 + \frac{1}{12n} + \frac{a_2}{n^2} + \frac{a_3}{n_3} + \frac{a_4}{n^4} + \ldots$, 把 $1 + \frac{1}{12n}$ 移到左边，方程两边同乘 n^2，然后取极限 $n \to \infty$ 可得 a_2; 然后把 $1 + \frac{1}{12n} + \frac{a_2}{n^2}$ 移到左边，方程两边同乘 n^3，然后取极限 $n \to \infty$ 可得 a_3, 依次类推可得 $a_4, a_5, \ldots$. 可以计算得到：$a_2 = \frac{1}{288}, a_3 = \frac{-139}{51840}, a_4 = \frac{-571}{2488320}$. 代码在文档 p4_4.nb 中。

4.5 problem 4_5

平衡位置势能导数为零，可得 $exp(-a(r-r_e))^2 - exp(-a(r-r_e)) + \frac{F}{2aU} = 0$, 解二元一次方程得:$exp(-a(r-r_e)) = \frac{1}{2} \pm \sqrt{\frac{1}{4} - \frac{F}{2aU}}$, 由于 $r - r_e$ 是小量，因此上式取加号合理，得到 $r = r_e - \frac{1}{a}\log\left(\frac{1}{2} + \sqrt{\frac{1}{4} - \frac{F}{2aU}}\right)$, 对 F 在 0 出做展开可得 $a_1 = \frac{1}{2a^2U}, a_2 = \frac{3}{8a^3U^2}, a_3 = \frac{5}{12a^4U^3}$. 代码在文档 p4_5.nb 中。