计算物理第一次作业

近代物理系 张浩然 SA21004048

October 2, 2021

Question 1 ▷ Obtain a general feeling of numerical physics

Summarize the major models and algorithms in any book of "numerical method".

Solution

- 插值(略)
- 最小二乘法拟合

  给定一组数据$\{(x_i, y_i)\}_{i=1}^m$, 假设拟合函数形式为 $\varphi(x) = \sum_{i=1}^m \alpha_i \varphi_i(x)$, 令

$$A = \begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \cdots & \varphi_n(x_m) \end{pmatrix}, \ \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}, \ \boldsymbol{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}, \tag{1}$$

  则误差平方和$Q(\boldsymbol{\alpha}) = \sum_{i=1}^m (\varphi(x_i) - y_i)^2 = ||A\alpha - Y||_2^2$满足 $\frac{\partial Q(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = 0$时, 有

$$A^{\mathrm{T}} A \boldsymbol{\alpha} = A^{\mathrm{T}} \boldsymbol{Y}. \tag{2}$$

- LU分解法求解线性方程组

  求解$Ax = b$, 做分解$A = LU$, 则$Ax = b$转化为$Ly = b$和$Ux = y$. 分解后只需$O(n^2)$计算量即可解出x. L、U分别是上、下三角阵. 一般取其中一个为单位三角阵, 否则分解不唯一.

- 迭代法求解线性方程组

  求解$Ax = b$, 令$D = \mathrm{diag}\{a_{11}, a_{22}, \cdots, a_{nn}\}$, 则有

$$\begin{aligned} DX &= (D - A)X + b \\ \Rightarrow \quad X^{(k+1)} &= D^{-1}(D - A)X^{(k)} + D^{-1}b \\ &= RX^{(k)} + g, \text{ with } R = I - D^{-1}A,\ g = D^{-1}b. \end{aligned} \tag{3}$$

- 数值积分和数值微分(略)
- 常微分方程数值解

求解$\begin{cases} y'(x) = f(x,y) \\ y(a) = y_0 \end{cases}$, $a \le x \le b$ 的二阶Runge-Kutta公式

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(k_1 + k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \end{cases} \tag{4}$$

- 矩阵的特征值和特征向量

  利用幂法可从任意初始向量X_0出发迭代得到矩阵A的按模最大的特征值及对应的特征向量

$$X^{(k+1)} = AX^{(k)}. \tag{5}$$

类似地, 利用反幂法可得到A的按模最小的特征值及特征向量.

Question 2 ▷ Summary the models in physics

Summarize as many models as possible in the books you have, which need to be calculated numerically. Models from literature are also welcome. The techniques in solving these problems are the focus of this course.

Solution

- Classical Mechanics

  1. Euler-Lagrange Equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} = \frac{\partial L}{\partial \boldsymbol{q}}. \tag{6}$$

  2. Hamilton's Equation

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{\partial H}{\partial \boldsymbol{p}}, \ \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \boldsymbol{q}}. \tag{7}$$

  3. Hamilton-Jacobi Equation

$$-\frac{\partial S}{\partial t} = H\left(\boldsymbol{q}, \frac{\partial S}{\partial \boldsymbol{q}}, t\right). \tag{8}$$

- Classical Electrodynamics

  1. Poisson's Equation for the static electric field

$$\nabla^2 \varphi = -\frac{\rho}{\epsilon_0}, \tag{9}$$

  Expansion of φ in spherical coordinates,

$$\varphi(r,\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{\infty}\left(a_{lm}r^l + \frac{b_{lm}}{r^{l+1}}\right)Y_{lm}(\theta,\phi), \tag{10}$$

  in cylindrical coordinates,

$$\varphi(s,\phi) = a_0 \ln s + b_0 + \sum_{m=1}^{\infty}\left(a_m s^m + \frac{b_m}{s^m}\right)(c_m \cos m\phi + d_m \sin m\phi) \tag{11}$$

2. Under Lorenz Gauge, the scalar and vector potential of the electromagnetic field satisfy

$$\Box\varphi = -\frac{\rho}{\epsilon},\ \Box\boldsymbol{A} = -\mu_0\boldsymbol{j}. \tag{12}$$

For a time-independent electromagnetic field, $\Box = \nabla^2$, $\partial_t\rho = 0$, thus

$$\nabla^2\varphi = -\frac{\rho}{\epsilon_0},\ \nabla^2\boldsymbol{A} = -\mu_0\boldsymbol{j}, \tag{13}$$

$$\Rightarrow \varphi = \frac{1}{4\pi\epsilon_0}\iiint \mathrm{d}V'\,\frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|},\ \boldsymbol{A} = \frac{\mu_0}{4\pi}\iiint \mathrm{d}V'\,\frac{\boldsymbol{j}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}. \tag{14}$$

- Quantum Mechanics

1. Schrödinger Equation

$$\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right]\psi(x) = E\psi(x). \tag{15}$$

Discretization: $\psi(x) \to \psi_i = \psi(x_i)$, thus

$$-\frac{\hbar^2}{2m}\frac{\psi_{i+1}+\psi_{i-1}-2\psi_i}{2h^2} + V(x_i)\psi_i = E\psi_i. \tag{16}$$

2. Time-independent perturbation theory

$$\begin{aligned} E_n &= E_n^{(0)} + \langle n|H'|n\rangle - \sum\nolimits_m' \frac{|\langle m|H'|n\rangle|^2}{E_m^{(0)}-E_n^{(0)}} + \cdots, \\ |\psi_n\rangle &= |n\rangle - \sum\nolimits_m' \frac{\langle m|H'|n\rangle}{E_m^{(0)}-E_n^{(0)}}|m\rangle + \cdots. \end{aligned} \tag{17}$$

3. Time-dependent perturbation theory

If $H = \begin{cases} H_0, & t=0 \\ H_0 + H't(t), & t>0 \end{cases}$, to the first order approximation, we have

$$|\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}E_i t/\hbar}|i\rangle + \sum\nolimits_m \left[\frac{1}{\mathrm{i}\hbar}\int_0^t \langle m|H'(\tau)|i\rangle\,\mathrm{e}^{\mathrm{i}\omega_{mi}\tau}\mathrm{d}\tau\right]\mathrm{e}^{-\mathrm{i}E_m t/\hbar}|m\rangle. \tag{18}$$

- Statistical Mechanics

1. Canonical ensemble

$$Z = \frac{1}{N!h^{Nr}}\int \mathrm{e}^{-\beta E(\boldsymbol{q},\boldsymbol{p})}\mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p},\ U = -\frac{\partial}{\partial\beta}\ln Z,\ F = U - TS - kT\ln Z \tag{19}$$

2. Grand canonical ensemble

$$\Xi = \frac{1}{N!h^{Nr}}\sum_N \mathrm{e}^{-\alpha N}\int \mathrm{e}^{-\beta E(\boldsymbol{q},\boldsymbol{p})}\mathrm{d}\boldsymbol{q}\mathrm{d}\boldsymbol{p},\ U = -\frac{\partial}{\partial\beta}\ln\Xi,\ J = U - TS - \mu N = -kT\ln\Xi \tag{20}$$

Question 3 ▷ Analytical programming language

Summarize the major commands in MATHEMATICA from the textbooks.

Solution

- `D[f,x]` gives the partial derivative $\partial f/\partial x$.

  `D[f,{x,n}]` gives the multiple derivative $\partial^n f/\partial x^n$.

  `D[f,x,y,...]` gives the partial derivative $\cdots(\partial/\partial y)(\partial/\partial x)f$.

  `D[f,{x,n},{y,m},...]` gives the multiple partial derivative $\cdots(\partial^m/\partial y^m)(\partial^n/\partial x^n)f$.

  `D[f,{{x1,x2,...}}]` for a scalar f gives the vector derivative $(\partial f/\partial x_1, \partial f/\partial x_2, \ldots)$.

- `Series[f,{x,x0,n}]` generates a power series expansion for f about the point $x = x_0$ to order $(x - x_0)^n$, where n is an explicit integer.

  `Series[f,x->x0]` generates the leading term of a power series expansion for f about the point $x = x_0$.

- `FullSimplify[expr]` tries a wide range of transformations on *expr* involving elementary and special functions and returns the simplest form it finds.

  `FullSimplify[expr,assum]` does simplification using assumptions.

- `Solve[expr,vars]` attempts to solve the system *expr* of equations or inequalities for the variables *vars*.

  `Solve[expr,vars,dom]` solves over the domain *dom*. Common choices of *dom* are Reals, Integers, and Complexes.

  `NSolve[expr,vars]` attempts to find numerical approximations to the solutions of the system expr of equations or inequalities for the variables vars.

  `DSolve[eqn,u,x]` solves a differential equation for the function u, with independent variable x.

  `DSolve[eqn,u,{x,xmin,xmax}]` solves a differential equation for x between $x_{\min}$ and $x_{\max}$.

  `DSolve[{eqn1,eqn1,...},{u1,u2,...},...]` solves a list of differential equations.

  `DSolve[eqn,u,{x1,x2,...}]` solves a partial differential equation.

- `Dot[a,b,c]` or `a.b.c` gives products of vectors, matrices, and tensors.

  `Cross[a,b]` gives the vector cross product of a and b.

  `Inverse[m]`, `Transpose[m]`, `Tr[m]`, `Det[m]` gives the inverse, transpose, trace, or determinate of a square matrix m.

  `Norm[expr]` gives the norm of a number, vector, or matrix. `Norm[expr,p]` gives the p-norm.

  `QRDecomposition[m]` yields the QR decomposition for a numerical matrix m. The result is a list $\{q, r\}$, where q is a unitary matrix and r is an upper-triangular matrix.

  `LUDecomposition[m]` generates a representation of the LU decomposition of a square matrix m.

- `Table[expr,n]` generates a list of n copies of *expr*.

  `Table[expr,{i,imax}]` generates a list of the values of *expr* when i runs from 1 to i_{max}.

  `Table[expr,{i,imin,imax}]` starts with $i = i_{\text{min}}$.

  `Table[expr,{i,imin,imax,di}]` uses steps di.

  `Table[expr,{i,{i1,i2,...}}]` uses the successive values $i_1, i_2, \ldots$.

  `Table[expr,{i,imin,imax},{j,jmin,jmax},...]` gives a nested list. The list associated with i is outermost.

- `Do[expr,n]` evaluates *expr* n times.

  `Do[expr,{i,imax}]` evaluates *expr* when i runs from 1 to i_{max}.

  `Do[expr,{i,imin,imax}]` starts with $i = i_{\text{min}}$.

  `Do[expr,{i,imin,imax,di}]` uses steps di.

  `Do[expr,{i,{i1,i2,...}}]` uses the successive values $i_1, i_2, \ldots$.

  `Do[expr,{i,imin,imax},{j,jmin,jmax},...]` evaluates *expr* looping over different values of j etc. for each i.

- `While[test,body]` evaluates *test*, then *body*, repetitively, until *test* first fails to give `True`.

Question 4 ▷ Tricks in numerical calculations

I use the following simple, but important examples, to illustrate some importance tricks in numerical and analytical calculations, which will be frequently used in this course.

---

1. Please generate numbers x_i and y_i, where $i = 1 \sim L$ (with $L = 10000$), then calculate
$$z = \prod_{i=1}^{L} x_i + \prod_{i=1}^{N} y_i.$$

2. Please plot the following function
$$P(x) = \sum_{k=1}^{\infty} \frac{x^{k/2}}{(3k)!}, \quad x \in (0, 1000).$$

3. Determine the approximate polynomial of $P(x)$, with error less than 10^{-10} for $x \in (0, 1000)$.

4. About the Stirling's approximation for $n!$
$$n! = n^n e^{-n} \sqrt{2\pi n} F(n), \quad F(n) = 1 + \frac{1}{12n} + \frac{a_2}{n^2} + \frac{a_3}{n^3} + \frac{a_4}{n^4} + \cdots.$$
Please determine the values of $a_i \in \mathbb{Z}$ in the higher-order terms by numerical method.

5. Consider the Morse potential

$$V(r) = U\left(1 - \mathrm{e}^{-a(r-r_e)}\right)^2 - Fr, \quad U, a, r_e > 0.$$

Assume that $F \to 0$, find analytically the minimal position of $V(r)$, to the accuracy of F^3, that is, $V'(x) = 0$, with

$$x = r_e + a_1 F + a_2 F^2 + a_3 F^3.$$

Then determine a_i analytically using perturbation theory; and verify them numerically.

Solution

1.

为避免出现趋于零的数, 提高浮点数运算精度, 可利用

$$z = \prod_{i=1}^{L} x_i + \prod_{i=1}^{N} y_i = \exp\left\{\sum_{i=1}^{L} \ln x_i\right\} + \exp\left\{\sum_{i=1}^{N} \ln y_i\right\} \tag{21}$$

进行计算. 不失一般性地, 令 $x_i, y_j \overset{\text{i.i.d.}}{\sim} U(0,1)$, $L = N = 10000$.
$X = \sum_{i=1}^{L} \ln x_i$ 和 $Y = \sum_{i=1}^{N} \ln y_i$ 可分别利用

```
N[Sum[Log[RandomReal[]], {i, 10000}]]
```

计算, 得 $X = -9996.15, Y = -9914.19$, 则

$$z = \prod_{i=1}^{L} x_i + \prod_{i=1}^{N} y_i = \mathrm{e}^{-9996.15} + \mathrm{e}^{-9914.19} \tag{22}$$

2.

使用Mathematica计算求和$\sum_{k=1}^{\infty} \frac{x^{k/2}}{(3k)!}$:

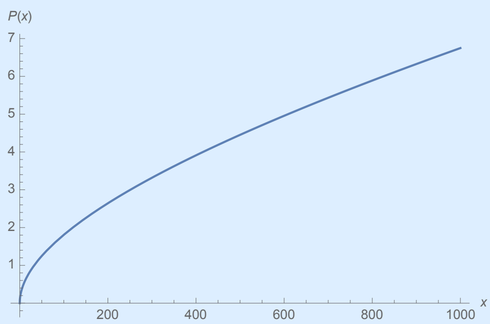
```
Sum[x^(k/2)/(3*k)!, {k, 1, Infinity}]
```

输出结果为

$$\frac{1}{3}\mathrm{e}^{-\frac{x^{1/6}}{2}}\left(-3\mathrm{e}^{\frac{x^{1/6}}{2}} + \mathrm{e}^{\frac{3x^{1/6}}{2}} + 2\cos\left(\frac{\sqrt{3}}{2}x^{1/6}\right)\right)$$

画出$P(x)$的图像:

```
p[x_] := Sum[x^(k/2)/(3*k)!, {k, 1, Infinity}]
Plot[p[x], {x, 0, 1000}]
```



3.

给出$P(x)$的一个近似

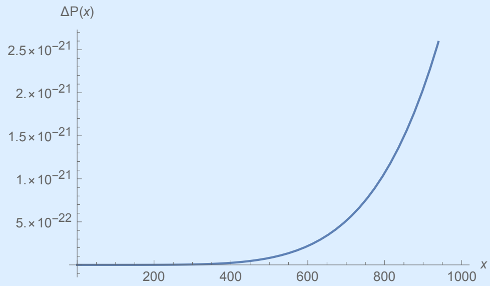
$$\tilde{P}(x)=\sum_{k=1}^{10}\frac{x^{k/2}}{(3k)!}. \tag{23}$$

为验证其误差$< 10^{-10}$, 可将其误差写为

$$\Delta P(x)=P(x)-\tilde{P}(x)=\sum_{k=11}^{\infty}\frac{x^{k/2}}{(3k)!}. \tag{24}$$

作出$\Delta P(x)$在$x\in(0,1000)$的图像:

```
\[CapitalDelta]p[x_] := Sum[x^(k/2)/(3 k)!, {k, 11, Infinity}];
Plot[\[CapitalDelta]p[x], {x, 0, 1000}]
```



可以看到该近似的误差$< 10^{-20}$, 满足要求.

4.

由

$$n!=n^n\mathrm{e}^{-n}\sqrt{2\pi n}F(n), \tag{25}$$

得

$$\begin{aligned}\ln\left[\sqrt{2\pi}F(n)\right]&=\ln\left(n!\times\mathrm{e}^n\times n^{-n}\times\frac{1}{\sqrt{n}}\right)\\&=\sum_{k=1}^{n}\ln k+n-\left(n+\frac{1}{2}\right)\ln n,\end{aligned} \tag{26}$$

故

$$F(n)=\frac{1}{\sqrt{2\pi}}\exp\left[\sum_{k=1}^{n}\ln k+n-\left(n+\frac{1}{2}\right)\ln n\right]. \tag{27}$$

利用上式计算 $n=1,2,\cdots,100$ 时的 $F(n)$, 并用 $\left\{1,\frac{1}{x},\frac{1}{x^2},\cdots\right\}$ 进行拟合:

```
f[n_] := Exp[N[ Sum[Log[k],{k,n}] + n - (n+1/2)Log[n] ]] / Sqrt[2 Pi]
data = Table[{n, f[n]}, {n, 100}];
Fit[data, {1, 1/n, 1/n^2, 1/n^3, 1/n^4}, n]
```

输出结果为

$$1.+\frac{0.0833297}{n}+\frac{0.00353375}{n^2}-\frac{0.00303262}{n^3}+\frac{0.000606732}{n^4}.$$

其中$a_1 = \dfrac{1}{12} \simeq 0.0833297$, $a_2 \simeq 0.00353375$, $a_3 \simeq 0.00303262$, $a_4 \simeq 0.000606732$.

5.

首先求解$V(r)$的极小值点:

```
V[r_] := U (1 - Exp[-a (r - re)])^2 - F*r
Assuming[U > 0 && a > 0 && re > 0, sol = Simplify[ Solve[V'[r] == 0 && V''[r] > 0, r, Reals] ]]
```

$$\left\{\left\{r \to \boxed{\mathrm{re} + \frac{\log\left(\frac{aU - \sqrt{aU(aU-2F)}}{F}\right)}{a} \text{ if } aU > 2F \land F > 0}\right\}, \left\{r \to \boxed{\mathrm{re} + \frac{\log\left(\frac{aU - \sqrt{aU(aU-2F)}}{F}\right)}{a} \text{ if } F < 0}\right\}\right\}$$

得到两种情况下的极小值点.

对于第一种情况 $F > 0$ 且 $aU > 2F$,

```
x1[F_] := Normal[sol[[1, 1, 2]]]
Series[x1[F], {F, 0, 3}, Assumptions -> {U > 0 && a > 0 && re > 0 && F > 0 && a*U > 2 F}]
```

输出结果为

$$r_e + \frac{F}{2a^2U} + \frac{3F^2}{8a^3U^2} + \frac{5F^3}{12a^4U^3} + O\left(F^4\right) \tag{28}$$

对于第二种情况 $F < 0$,

```
x2[F_] := Normal[sol[[2, 1, 2]]]
Series[x2[F], {F, 0, 3}, Assumptions -> {U > 0 && a > 0 && re > 0 && F < 0}]
```

输出结果同样为

$$r_e + \frac{F}{2a^2U} + \frac{3F^2}{8a^3U^2} + \frac{5F^3}{12a^4U^3} + O\left(F^4\right) \tag{29}$$