第三次作业 报告

1400012141
邵智轩
2017年12月

1 Householder 与Givens 在QR 分解中的比较

这部分代码提交在文件”QR_Factorization.py“中。

1.1 $n \gg 1$时的渐进效率比较

1.1.1 Householder变换的渐进时间复杂度

先看对一个n维向量（$n > 1$）的Householder反射所需运算次数。第一步需要计算向量的模，约$2n$次运算。变换后的向量第一个分量就是它的模，符号与原向量第一分量符号相反。反射面的法向量$\mathbf{w}$只需将变换前后的向量相减并归一，由于变换后的向量只有一个分量，这一步约需$O(1)$次运算（归一时可以用之前计算模的结果）。共计$2n$次运算。

对于一个n阶方阵，先看第一次变换（$\mathbf{H}_1$）。对第1列向量做Householder反射需要$2n$次运算，用得到的反射面法向量$\mathbf{w}$对余下每列做变换（$\mathbf{H}\mathbf{x} = \mathbf{x} - 2(\mathbf{w}^T\mathbf{x})\mathbf{w}$）需要$4n$次运算，共计$2n + 4n(n-1) \approx 4n^2$次运算。所以对整个矩阵变换得到$\mathbf{R}$需要：

$$4[n^2 + (n-1)^2 + ... + 1] = 4\frac{n(n+1)(2n+1)}{6} \approx \frac{4}{3}n^3$$

次运算。

此外还要计算$\mathbf{Q}$。$\mathbf{Q} = \mathbf{H}_1\mathbf{H}_2...\mathbf{H}_n$，故可以从右往左乘（使得$\mathbf{H}_i$阶数从小往大乘）得到$\mathbf{Q}$。同之前的讨论，$\mathbf{H}_1$左乘$\mathbf{H}_2\mathbf{H}_3...\mathbf{H}_n$的每个列向量（$n$维，共$n$个）都需要$4n$次运算，共$4n^2$次；$\mathbf{H}_2$左乘$\mathbf{H}_3\mathbf{H}_4...\mathbf{H}_n$的每个列向量（$n-1$维，共$n$个）都需要$4(n-1)$次运算，共$4n(n-1)$次，故求出$\mathbf{Q}$一共需要

$$4n[n + (n-1) + (n-2) + ... + 1] = 4n\frac{(n+1)n}{2} \approx 2n^3$$

次运算。

故Householder变换做QR分解共需要约$\frac{10}{3}n^3$次运算。

1.1.2 Givens变换的渐进时间复杂度

对于2维向量，Givens旋转变换需要做6次浮点运算，分别计算（模长α4次，s和c各1次）。对于所有分量均非零的n维向量，要做$(n-1)$次Givens旋转，并得到$(n-1)$个Givens变换矩阵$\mathbf{G}$。任何一个Givens矩阵作用到n维向量上需要6次运算。故对矩阵第一列做Givens旋转变换总计需要：

$$6(n-1) + (n-1)6 \cdot (n-1) \approx 6n^2$$

次运算。求出$\mathbf{R}$总计需要：

$$6[n^2 + (n-1)^2 + ... + 1] = 6\frac{n(n+1)(2n+1)}{6} \approx 2n^3$$

。

再求出$\mathbf{Q}$，共计$(n-1)+(n-2)+...+1=\frac{n(n-1)}{2}$个Givens旋转变换，每次相乘需要$6n$次运算，总计

$$6n\frac{n(n-1)}{2}\approx 3n^3$$

次运算。

故Givens旋转变换总计约$5n^3$次运算。

1.1.3 小结

可以看到，两种方法虽然都是$O(n^3)$，但是领头项的系数还是有差异的。当n足够大时，对于一般的满阵（非稀疏矩阵），Householder变换显然效率更高。

1.2 编写程序

算法的正确性可以由输出结果中的"test_QR()"函数验证。

1.3 测试比较运行速度

为了随机数的可信度及效率，用"numpy"库的'"numpy.random"函数产生$\mathcal{U}(0,1)$的随机数，并使用$2X-1$转换为$\mathcal{U}(-1,1)$的随机数。用"time"库的"time.clock()"函数计时。产生20个阶数分别为6，30，100的随机矩阵，比较Householder变换和Givens变换所需时间，并与标准库函数"scipy.linalg.qr()"比较。下表是某一次试验的运行结果：

Table 1: QR分解的运行时间(单位：s)

	20个6×6	20个500×500	2000个6×6
标准库函数	0.00147	0.943	0.1579
Householder变换	0.00515	29.46	0.3850
Givens变换	0.00600	61.40	0.4475

可以看到，在阶数比较低的时候，两种方法是差不多快的。因为领头阶后面的项的系数，乃至常数项可能很大，使得在n较小时领头阶并不显著。但当n取到500时就能看出，Householder变换用时大约是Givens变换的一半。

2 幂次法求矩阵最大模的本征值和本征矢

这部分代码提交在文件"Diatomic_Chain_Eigenvalue_Problem.py"中。

2.1 本征方程

对尝试解求二阶导，得到：

$$\ddot{\mathbf{x}}(t)=-\omega^2\mathbf{x}e^{-i\omega t}=-\omega^2\mathbf{x}(t)$$

带入经典运动方程$\ddot{\mathbf{x}}(t)=-\mathbf{A}\cdot\mathbf{x}(t)$，就得到本征方程：

$$\mathbf{A}\cdot\mathbf{x}(t)=\lambda\mathbf{x}(t),\ \lambda=\omega^2$$

考虑到所有原子的本征振动频率相同，可以分离变量消去时间的部分，得到关于振幅$\mathbf{x}$的方程：

$$\mathbf{A}\cdot\mathbf{x}=\lambda\mathbf{x},\ \lambda=\omega^2$$

2.2 程序求出的本征值和本征矢

判停条件设为$||q^{(k+1)}-q^{(k)}||_\infty > 10^{-8}$，得到本征值：3.9999999999999853，本征矢：(0.31622778, -0.31622773, 0.31622769, -0.31622768, 0.3162277, -0.31622775, 0.3162278 , -0.31622784, 0.31622785, -0.31622783)$^{\mathrm{T}}$。这符合我们预期的精确解

$$\lambda=4,\ \mathbf{x}=\frac{1}{\sqrt{10}}(1,-1,1,-1,1,-1,1,-1,1,-1)^{\mathrm{T}}$$

注：由于初始的矢量是随机的，结果在误差范围内，有一定的随机性。

3 关联函数的拟合与数据分析

3.1 估计关联函数的中心值$\bar{C}(t)$、误差$\Delta C(t)$、相对误差$\Delta C(t)/\bar{C}(t)$

样本$\bar{C}(t)$及其误差估计$\Delta C(t)$在程序中求出并输出，就不再这里列出了。其中$\Delta C(t)$用讲义(7.23)式：

$$\Delta x=\sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(x_i-\bar{x})^2}$$

估计（样本标准差S的$\frac{1}{\sqrt{n}}$）。下面列出相对误差$\Delta C(t)/\bar{C}(t)$作为t的函数，列表并作图如下。

Table 2: 相对误差$\Delta C(t)/\bar{C}(t)$作为t的函数（$\times 100\%$）

t	0	1	2	3	4	5	6	7	8	9	10
$\Delta C(t)/\bar{C}(t)$	0.11	0.17	0.40	0.59	0.73	0.84	0.93	1.00	1.05	1.07	1.09
t	11	12	13	14	15	16	17	18	19	20	21
$\Delta C(t)/\bar{C}(t)$	1.13	1.17	1.21	1.26	1.29	1.33	1.37	1.40	1.42	1.44	1.45
t	22	23	24	25	26	27	28	29	30	31	32
$\Delta C(t)/\bar{C}(t)$	1.49	1.52	1.56	1.59	1.60	1.62	1.64	1.66	1.68	1.71	1.73

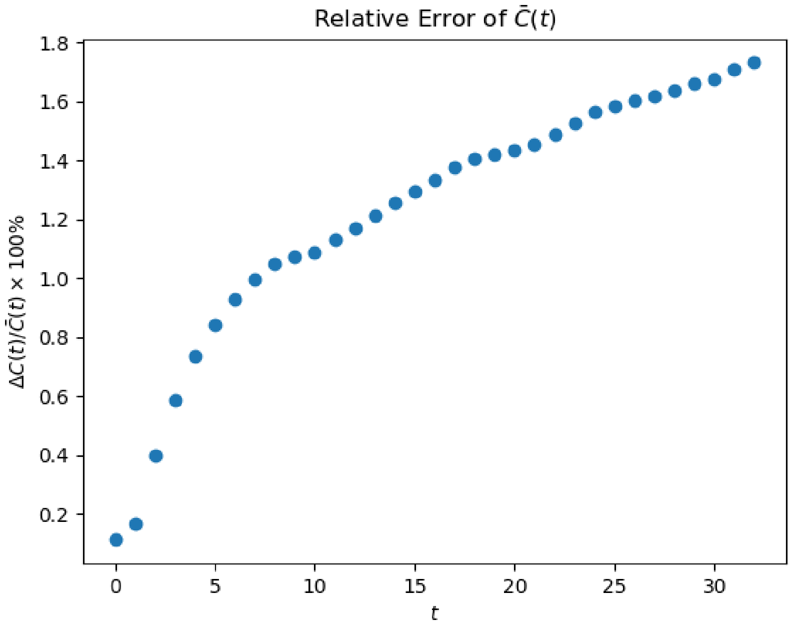

Figure 1: 相对误差$\Delta C(t)/\bar{C}(t)$作为t的函数

相对误差随t增大，信噪比随t减小。

3.2 用Jackknife方法估计有效质量$m_{eff}(t)$及其误差$\Delta m_{eff}(t)$

用全样本（$N = 200$）的平均值得到有效质量$m_{eff}(t)$的估计值。显示在Figure 2中的蓝点。

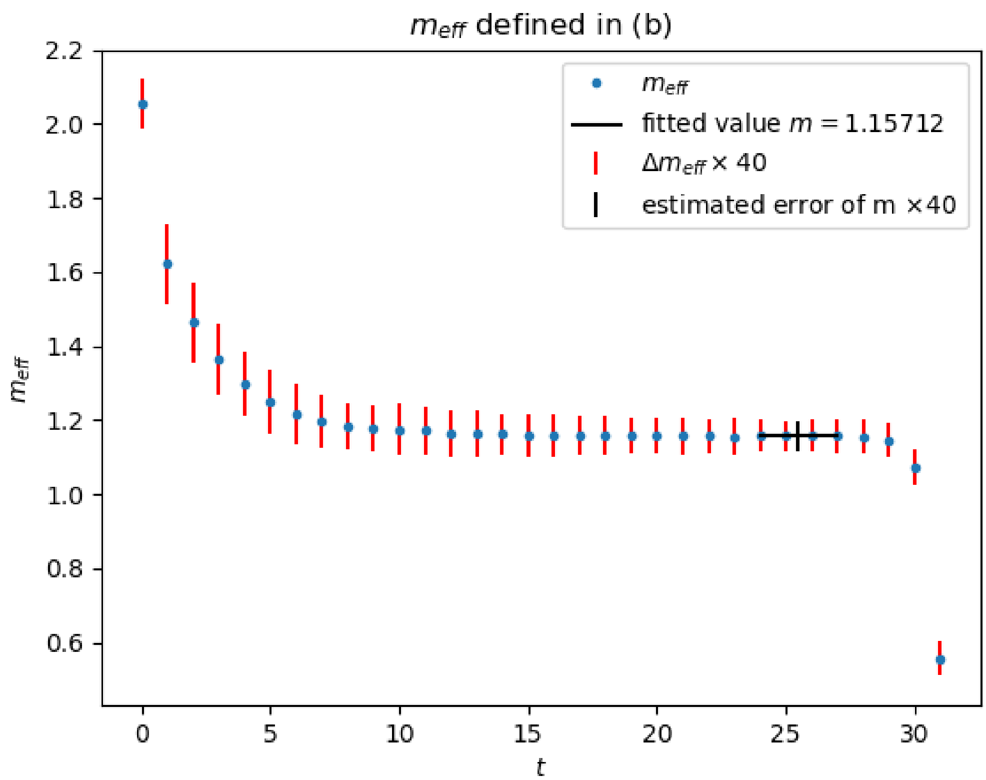

Figure 2: (b)问定义的有效质量函数$m_{eff}(t)$

用Jackknife方法每次剔除第i个组态，用其他所有组态的平均值$C^{(i)}(t)$通过公式(9)：计算$m_{eff}^{(i)}$，再用讲义上的(7.45)式:

$$\delta f = \sqrt{\frac{M}{M-1}\sum_{i=1}^{M}(f_{(i)} - \langle f \rangle)^2}$$

得到误差的估计$\Delta m_{eff}(t)$。显示在Figure 2中红色的误差棒。由于误差太小不容易看见，将误差放大了40倍作图。并将每一个t的（绝对）误差$\Delta m_{eff}(t)$用柱状图作图在Figure 4中。另外截取m_{eff}的平台区，按照1:1的误差作了一张图，见Figure 3。

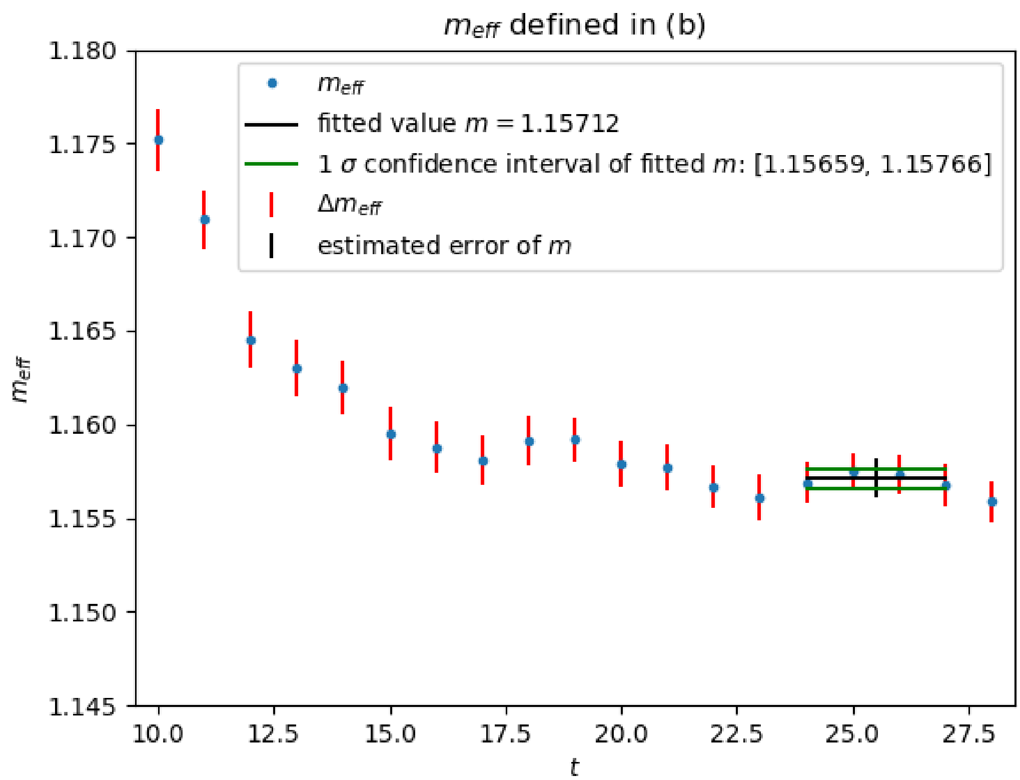

Figure 3: (b)问定义的有效质量的误差$\Delta m_{eff}(t)$（平台区域）

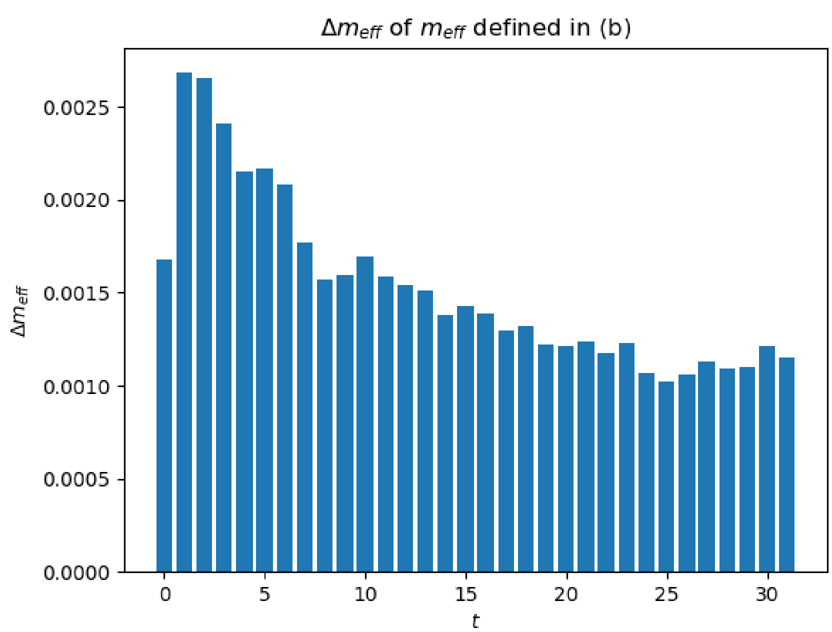

Figure 4: (b)问定义的有效质量的误差$\Delta m_{eff}(t)$

3.3 单参数拟合m

对于任意一个给定的区间$[t_{min}, t_{max}]$和其上的数据$m_{eff}(t)$，由于χ^2是关于参数m的开口向上二次函数，它取极大值的充要条件为对m的一阶导为零，即：

$$m = \left[\sum_{t=t_{min}}^{t_{max}} \frac{1}{\Delta m_{eff}(t)^2}\right]^{-1} \left[\sum_{t=t_{min}}^{t_{max}} \frac{m_{eff}(t)}{\Delta m_{eff}(t)^2}\right]$$

让计算机扫描所有$[t_{min}, t_{max}]$区间，用上式的m计算χ^2，以最小的$\chi^2/d.o.f = \chi^2/(t_{max} - t_{min})$为标准确定合适的区间。拟合结果如下，并将拟合结果画在了Figure 2与Figure 3中：

$$t_{min} = 24,\ t_{max} = 27,\ m = 1.157123872,\ \chi^2_{min} = 0.327841179,\ p = 0.045289983$$

拟合出的m的误差由下式估计：

$$\Delta m = \sqrt{\frac{\sum_{t=t_{min}}^{t_{max}} \Delta m_{eff}(t)^2}{t_{max} - t_{min} + 1}}$$

计算得$\Delta m = 0.001050653$。

我们还可以通过$\Delta\chi^2 = \chi^2 - \chi^2_{min}$构造$m$的置信区间。由于是单参数估计，$\Delta\chi^2$服从自由度为1的$\chi^2$分布（即标准正态分布的平方）。构造$m$的$1\sigma$置信区间，对应的$\chi^2(m) = \chi^2_{min} + \Delta\chi^2 = \chi^2_{min} + 1.00 = 1.327841179$。计算得$1\sigma$置信区间为：$[1.156590, 1.157658]$。将$m$的误差估计和$m$的置信区间也画在Figure 2和Figure 3中。

3.4 重新定义m_{eff}，重复之前的做法

拟合结果如下，并将拟合结果画在了Figure 5和Figure 6中：

$$t_{min} = 25,\ t_{max} = 29,\ m = 1.157099908,\ \chi^2 = 0.359150289,\ p = 0.014317336$$

m的误差估计$\Delta m = 0.001116479$，1σ置信区间为：$[1.156600, 1.157599]$。

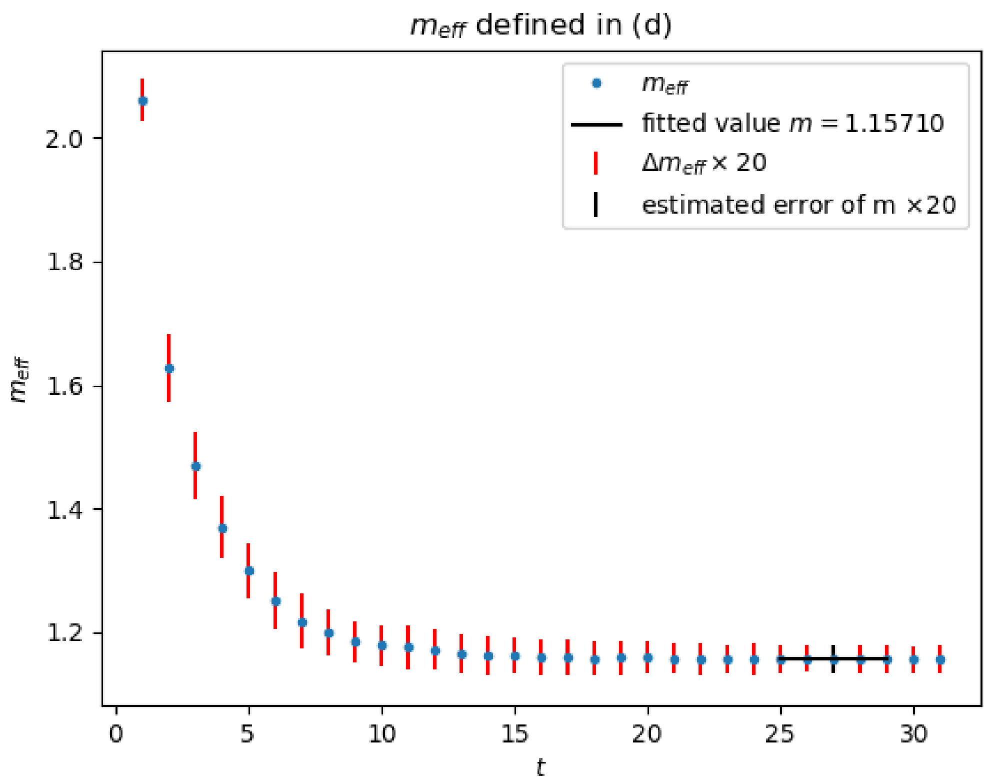

Figure 5: (d)问定义的有效质量函数$m_{eff}(t)$

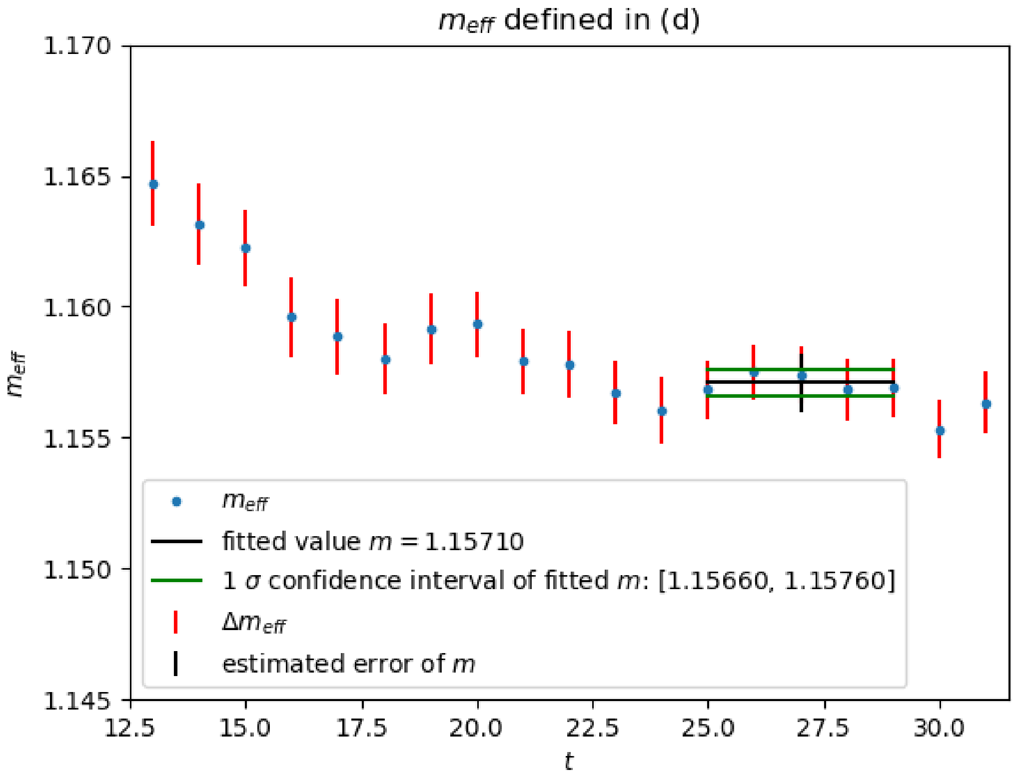

Figure 6: (d)问定义的有效质量的误差$\Delta m_{eff}(t)$（平台区域）

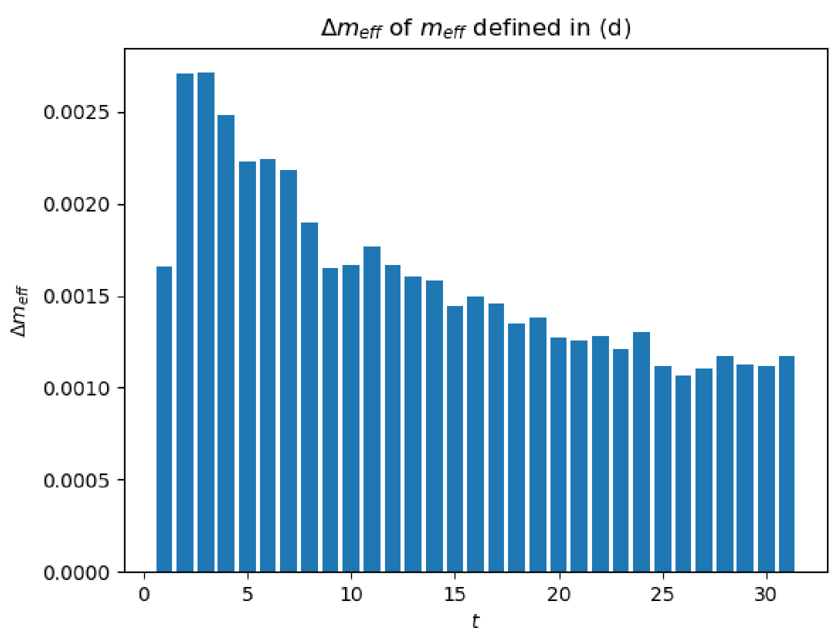

Figure 7: (d)问定义的有效质量的误差$\Delta m_{eff}(t)$

3.5 用Bootstrap方法估计$\rho_{3,5}$和$\rho_{4,5}$的中心值和误差

用Bootstrap方法从$N = 200$个样本有放回地中抽取200个，用它们代入公式(13)和(14)计算$\rho_{3,4}$和$\rho_{3,5}$。执行$N_B = 1000$次，获得1000套数据，取平均（(7.46)式）得到$\rho_{3,4}$和$\rho_{3,5}$的中心值的估计，再用讲义(7.48)式和(7.50)式两种方法得到误差的估计。某一次Bootstrap结果如下：

$$\langle\rho_{3,4}\rangle = 0.957092297,\ \delta\rho_{3,4} = 0.006137477,\ 68\%\ \text{interval}: (0.950823322, 0.963301192)$$

$$\langle\rho_{3,5}\rangle = 0.901141069,\ \delta\rho_{3,5} = 0.016393360,\ 68\%\ \text{interval}: (0.884654012, 0.917089774)$$

可用直方图直观地显示Bootstrap的结果：

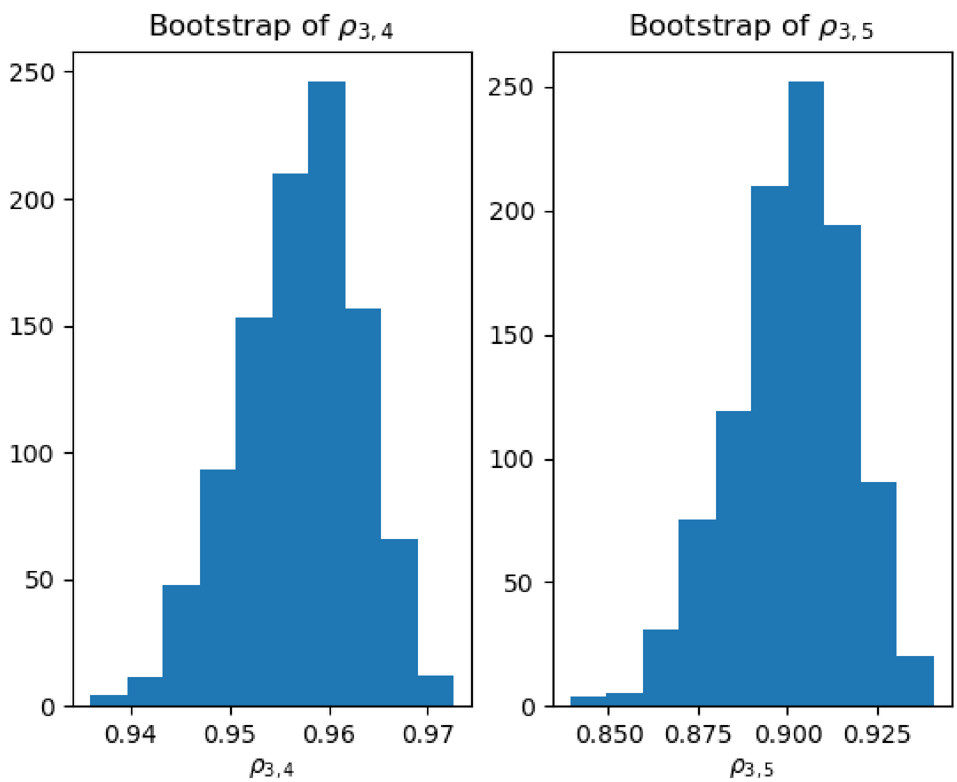

Figure 8: 用Bootstrap方法估计$\rho_{3,5}$和$\rho_{4,5}$的中心值和误差