第五章作业

问题回顾 一段时间内经由管道传输的流体总质量为

$$M=\int_{t_1}^{t_2} Q(t)c(t)\mathrm{d}t$$

其中，M 为质量(kg)，t1 为初始时间(min)，t2 为终止时间(min)，Q(t)为体积流率(m3/min)，c(t)为浓度(mg/m3)。体积流率和浓度为时间的函数：

$$Q(t)=9+4cos^2(0.4t)$$
$$c(t)=5e^{-0.5t}+2e^{0.15t}$$

试确定在$\mathrm{t}_1=2$到$\mathrm{t}_2=8$的时间内通过的质量。

目录

问题分析

该问题的积分可以解析求解，但从数值积分的角度出发，只需要令被积函数为：

$$f(x)=Qc(t)=[9+4cos^2(0.4t)]\cdot[5e^{-0.5t}+2e^{0.15t}],\ \ x=t$$

下面使用几种不同的数值方法求解该问题。

复合梯形算法

将区间分成 n 等分以后用，对每一个区间使用梯形公式，理论上当区间的划分趋向于无穷时，连续函数的近似积分值将趋向于积分真值。而本题的被积函数在被积分区间上连续，可以用这种方法。

$$M\approx\sum_{i=1}^{n} h\cdot\frac{f(x_{i-1})+f(x_i)}{2}=\frac{h}{2}[f(a)+f(b)]+h\sum_{i=1}^{n-1} f(x_i)$$

由于 MATLAB 中 m 语言向量化的特点，采用易于读写的向量化语言编写方案（并没有采用 matlab 中有关数值积分的函数）增加程序的可读性：

```
% this program is developed to demonstrate the combined trapz integration
b = 8; a = 2;       % integrating bounds
n = 4096;          % sub-interval quantities
x = linspace(a,b,n+1);
h = (b-a)/n;       % sub-interval length
I = (Qc(x(1))+Qc(x(n+1)))/2 + sum(Qc(x(2:n)));
I = I*h;
```

复合梯形法的误差可以被控制。

$$|E_T(f)| = \frac{b-a}{12}h^2 f''(\xi) = \frac{(b-a)^3}{12n^2}f''(\xi) \le E$$

区间划分段数 n 可以根据，而$\max_{2\le x\le 8}(f''(\xi)) = 87.42$ 当且仅当ξ=8。代入具体数值可知：

$$n \ge \sqrt{\frac{(8-2)^3}{12} \cdot \frac{87.42}{E}}$$

上面是一个不紧的界，事实上不需要$\left\lceil\sqrt{\frac{(8-2)^3}{12} \cdot \frac{87.42}{E}}\right\rceil$这么大的分割数就可以使得梯形积分收敛到较好的结果。

代入绝对误差 $E = 5 \times 10^{-5}$[1]，实际计算结果表明，n=5609。事实上用上述程序改变 n 的值循环求解积分问题有：

子区间数 n	子区间长度	M 估计值	绝对误差 E	相对误差 ϵ_t	子区间数	子区间长度	M 估计值	绝对误差E	相对误差 ϵ_t
1	6	322.3484	88.91258	27.58276%	1200	0.005	322.3484	4.99E-05	0.00002%
40	0.15	322.3933	0.044958	0.01395%	1800	0.003333	322.3484	2.22E-05	0.00001%
200	0.03	322.3502	0.001798	0.00056%	2000	0.003	322.3484	1.80E-05	0.00001%
400	0.015	322.3488	0.00045	0.00014%	3000	0.002	322.3484	7.98E-06	0.00000%
800	0.0075	322.3485	0.000112	0.00003%	4000	0.0015	322.3484	4.48E-06	0.00000%

由于上面的估计没有去平均导数，而去了[2,8]区间上的最大导数，使得这个估计并不准确，事实上当$n = 1201$ 时，满足小数点后 4 位有效数字的要求。

复合 simpson 方法

在区间[a, b]上等间距地取 2n-1 个点。可以分成 n 个组反复运用 Simpson1/3 法则进行积分，最后可以将公式表示为：

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(a) + f(b) + 2\sum_{k=1}^{n-1} f(x_{2k}) + 4\sum_{k=1}^{n} f(x_{2k-1})\right]$$

```
m = round(n/2)
```

[1] 这个绝对误差可以使得小数点后有 4 为有效数字。

```
b = 8; a = 2;          % integrating bounds
x = linspace(a,b,2*m+1); [2]
h = (b-a)/2/m;         % sub-interval length
I = (Qc(a)+Qc(b)...
    + 4*sum(Qc(x(2:2:2*m))) + 2*sum(Qc(x(3:2:2*m-1))));
I = I*h/3;
```

同前面的复合型梯形法则，可以得出

$$|E_T(f)| = \frac{(b-a)^5}{180n^4} f^{(4)}(\xi) \le E$$

那么区间分割数量 n 的一个送的估计界是：

$$\left\lfloor \sqrt[4]{\frac{(8-2)^5}{180} \cdot \frac{8}{E}} \right\rfloor$$

其中$\max\left(f^{(4)}(\xi)\right) = 8$。

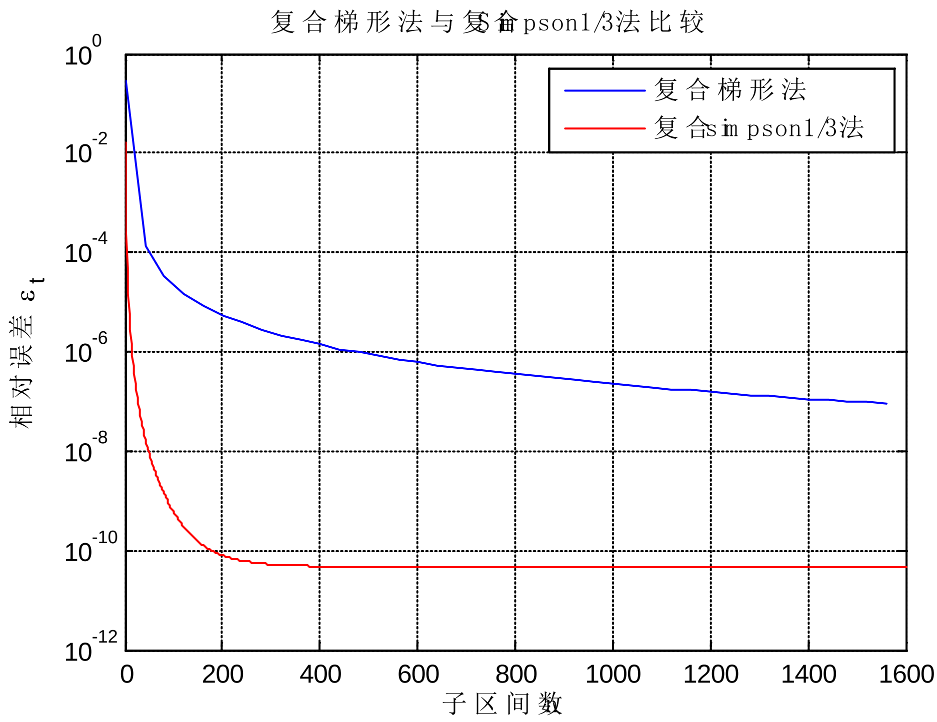
同样取$E = 10^{-5}$，与前面$n_{梯形} = 5690 \gg n_{辛普森} = 52$。这样一来通过复合辛普森 1/3 法则不仅可以大大减小计算量，并且提高精度。

子区间数 n	子区间长度	M 估计值	绝对误差 E	相对误差 ϵ_t	子区间数	子区间长度	M 估计值	绝对误差E	相对误差ϵ_t
2	3	317.1553	-5.19307	-1.6110%	16	0.375	322.3481	-0.00027	-0.0001%
4	1.5	322.2557	-0.09268	-0.0288%	18	0.333333	322.3482	-0.00017	-0.0001%
6	1	322.3329	-0.01546	-0.0048%	20	0.3	322.3483	-0.00011	0.0000%
8	0.75	322.3437	-0.00462	-0.0014%	22	0.272727	322.3483	-0.00008	0.0000%
10	0.6	322.3465	-0.00185	-0.0006%	24	0.25	322.3483	-0.00005	0.0000%
12	0.5	322.3475	-0.00088	-0.0003%	26	0.230769	322.3483	-0.00004	0.0000%
14	0.428571	322.3479	-0.00047	-0.0001%	28	0.214286	322.3483	-0.00003	0.0000%

可以看到当 n=22 时，就有了小数点后 4 位有效数字。

比较上面两种方法如下图：

[2] 此处用的 linspace()函数将区间等分成 2n 个子区间，共 2n+1 个端点



可以看到复合 Simpson1/3 法则在本文题上比复合梯形算法好了 2 个数量级以上（在相对误差的意义上）。但在子区间划分到达一定数量后，对精度的改进就不明显了。

误差的更精确估计

前面使用$|E_T(f)| = \frac{b-a}{12}h^2f''(\xi)$ 和 $|E_T(f)| = \frac{(b-a)^5}{180n^4}f^{(4)}(\xi)$ 来估计误差，如果使用导数在区间上极值效果并不理想，并且要事先知道导数的形式，这在实际问题求解中代价太大。所以需要一种迭代的求解方法通过判断两次迭代值的差的收敛情况来估计计算值收敛到真值的程度。

由于这些复合插值型的积分方法都采用了分区间的策略，那么一种自然的方法就是对二分区间，利用原有计算的一些区间端点值来进行进一步计算。这就是所谓的自适应步长的估计。

经过公式推导有梯形公式递推：

$$T_{2n} = \frac{1}{2}T_n + \frac{b-a}{2n}\sum_{i=1}^{n} f\left(a + \frac{(2i-1)(b-a)}{2n}\right)$$

$$= \frac{1}{2}T_n + \frac{1}{2}h_n \cdot \sum(\text{新加分割点的函数值})$$

具体求解编程在后面进行

Romberg 积分

通过上例可以想到我们仍然可以对 Simpson1/3 法则进行递推计算，但是从 Romberg 积分的角度可以解决所有的问题。即通过 Romberg 的关系式，可以从最初的梯形递推，推出 Simpson1/3 法则积分值，以至于更高的精度。

一般用如下式子表示：

$$M(j,k) = M(j,k-1) + \frac{M(j,k-1) - M(j-1,k-1)}{4^k - 1}$$

一般推至 K=3 或者 4 即可终止。

而在边界上有：

$$M(j,0) = T_{2^j}$$

T_{2^j} 的求解利用前面的递推公式得到。

下面给出 Romberg 积分的 MATLAB 程序，这个程序的终止条件事先设定，没有自适应的收敛机制。并且在数据结构上，直接构造了 Romberg 积分表，便于调试和现实实验结果，事实上这种方法浪费了存储空间，但是由于 Romberg 积分的存储量本身不大，这种浪费并不致命。需要注意的是由于 MATLAB 中的矩阵下标不能出现 0，程序中的 t-1 就表示了 romberg 积分的阶数，同样地程序中的结果 M(j,k) 就是理论推导中的 M(j-1,k-1)：

```
% this program is developed to demonstrate the Romberg's rule
j=0; t=5; % t-1 is the order of Romberg's rule
M=[];
b = 8; a=2;
h = (b-a)/2^j;        % sub-interval length
M(1,1) = h*(Qc(a)+Qc(b))/2;
for j=2:t;      % iteratively solve the trapezoidal integrating.
h=h/2;
M(j,1) = 0.5*M(j-1,1) + h*sum(Qc(a+h*(1:2:2^(j-1))));
end
for k=2:t       % Romberg's rule
   M(k:t,k) = M(k:t,k-1) + (M(k:t,k-1)-M(k-1:t-1,k-1))/(4^(k-1)-1);
end
```

下面给出一个 4 阶的求解结果：

Romberg	M(j,k=0)	M(j,k=1)	M(j,k=2)	M(j,k=3)	M(j,k=4)
0	411.2610				
1	340.6817	317.1553			
2	326.8622	322.2557	322.5957		
3	323.4734	322.3437	322.3496	322.3457	
4	322.6294	322.3481	322.3484	322.3484	322.3484
截断误差	O(h^2)	O(h^4)	O(h^6)	O(h^8)	O(h^10)

可以看到最后的结果精度不高

$$\epsilon = \frac{M(4,4) - I}{I} = 2.16e - 8$$

如果要提高精度那么就要增加阶数。

利用上表最后一行对误差阶数的估计结果可以得到：

$$\left(\frac{b-a}{2^j}\right)^{2j+2} = h_j^{2j+2} \approx E$$

两边取对数后有

$$(2j+2)(\ln(b-a) - j\cdot\ln 2) = \ln E$$

解出这个方程得到：

$$j = \frac{-(2\ln(\mathrm{b}-\mathrm{a}) - 2\ln 2) + \sqrt{(-2\ln 2 + 2\ln(\mathrm{b}-\mathrm{a}))^2 + 4\times 2\ln 2\left((2\ln(\mathrm{b}-\mathrm{a})) - \ln E\right)}}{4\ln 2}$$

但这个基于阶的估计只能作为一个阶数 j 下界的估计，在计算中并不能保证收敛。比如取

$$\mathrm{E} = 5\times 10^{-5}$$

$$\mathrm{J} = 3$$

J 是 Romberg 方法的阶。但此时的绝对误差为|E| =0.0027，并没有小数点后 4 位准确有效数字。

事实上准确的截断误差估计为：

$$|E_T(f)| = \frac{\left(j\cdot h_j\right)^{2j+3} j\cdot h_j}{C_k} f^{(2j+2)}(\xi)$$

其中$\mathrm{C_n}$为 n 阶 cotes 系数和，但是$f^{(2j+2)}(\xi)$的估计仍然比较困难，且还要求取 cotes 系数。

一种可行的办法是利用迭代求 Romberg 方法的迭代误差。即迭代停止条件是：

$$|M(J,J) - M(J-1,J-1)| < E$$

J 就是 Romberg 方法的阶。

改进 Romberg 积分

那么下面一个程序可以根据绝对误差要求自适应确定步长，从算法上说就是每计算一个新的梯形积分就应用 Romberg 关系算出所有新的高阶积分（Simpson 积分及其更高阶）。

```
% adaptive Romberg order -- Romberg 2.0
E = 5e-5;          % absolute error tolerance
j = 1;             % initializing
M=[];
b = 8; a=2;
h = (b-a);         % initial sub-interval length
M(1) = h*(Qc(a)+Qc(b))/2;
M(2) = inf;           % initializing
while (abs(M(1)-M(2))>E)
   j = j+1;        % trapezoidal integrating
   h = h/2;
   M(j) = 0.5*M(j-1) + h*sum(Qc(a + h*(1:2:2^(j-1))));
   for k = j-1:-1:1
   M(k) = M(k+1) + (M(k+1) - M(k))/(4^(j-k) - 1);
   end
end
```

在 E = 5e-5 或者小数点后 4 位有效数字的要求下，Romberg 积分阶数 j=4（程序中的 j=5）就可以得到积分值：

$$M = 322.348374$$

$$\text{真值 } Q = 322.348367$$

符合预期结果。

小结

本次作业从两种最常见的复合积分方法——梯形积分、Simpson1/3 法则出发，编写了相应的求解程序，并且突出了两者求解计算量以及计算精度的比较。从减少计算量出发，导出了递归的复合梯形法的求解公式，这将最终启发 romberg 方法的产生。

在讨论 romberg 算法的过程中讨论了最基本的形式和自适应步长的算法。