NTNU

TMA4215 Numerical Mathematics — Fall 2016

Problem Set 6

1 Find an approximation to the integral

$$\int_1^{1.5} x^2 \ln x \, dx$$

by Romberg integration (see the note on the webpage). Do the first 3 rows by hand, starting with $h = 0.5$. Then rewrite the MATLAB code `extrapolation.m` for doing Romberg integration, and apply the code on the integral above. How many rows in the extrapolation table are needed before no better accuracy can be obtained?

Solution. We first note that the exact solution is $Q = \frac{1}{8}\left(9\ln(1.5) - \frac{19}{9}\right) \approx 0.192\,259\,357\,732\,796$. The Romberg integration scheme for an integral $\int_a^b f(x)\,dx$ is given as

$$\begin{cases} T_{j,1} = F\left(h/2^{j-1}\right) & \text{for } j = 1, 2, \dots; \\ T_{j,k+1} = \dfrac{4^k T_{j,k} - T_{j-1,k}}{4^k - 1} & \text{for } k = 1, 2, \dots, j-1, \end{cases}$$

where

$$F(h) = h\left(\tfrac{1}{2}f(x_0) + \sum_{i=1}^{N-1} f(x_i) + \tfrac{1}{2}f(x_N)\right)$$

is the trapezoidal rule with stepsize $h = (b-a)/N$ and $x_i = a + ih$ for $i = 0, 1, \dots, N$. Routine calculations, with $h = 0.5$, now yield the values in Table 1.

Table 1: First 3 rows in the Romberg extrapolation table of the $T_{j,k}$'s.

j	$T_{j,1}$	$T_{j,2}$	$T_{j,3}$
1	0.228 074 123 310 842		
2	0.201 202 511 387 534	0.192 245 307 413 098	
3	0.194 494 473 181 091	0.192 258 460 445 610	0.192 259 337 314 444

For example, defining $f(x) = x^2 \ln x$, we have

$$T_{1,1} = \tfrac{1}{4}\big(f(1) + f(1.5)\big), \qquad T_{2,1} = \tfrac{1}{2}T_{1,1} + \tfrac{1}{4}f(1.25) \qquad \text{and} \qquad T_{2,2} = \tfrac{1}{3}\left(4T_{2,1} - T_{1,1}\right).$$

A naïve modification of the `extrapolation.m` code is to define F as follows.

```
% Trapezoidal rule, assuming h = (b - a) / N;
F = @(h) h * (f(a)/2 + (h < b-a) * sum(f(a + (1:(b-a)/h - 1) * h)) + f(b)/2);
N = 1;              % Initial no. of eval. pts minus one in the trap. rule.
h = (b - a) / N;    % Initial stepsize.
```

This procedure, however, uses many unnecessary function evaluations. The reader is invited to implement an improved version reducing the computational cost, where we note that $T_{j,1}$ can be partly expressed by $T_{j-2,1}$ (the exception is $T_{2,1}$, as seen above).

In double precision, 6 rows are needed to achieve optimal accuracy. Table 2 displays the errors for convenience.

Table 2: Errors $Q - T_{j,k}$.

j	$T_{j,1}$	$T_{j,2}$	$T_{j,3}$	$T_{j,4}$	$T_{j,5}$	$T_{j,6}$
1	-3.581×10^{-2}					
2	-8.943×10^{-3}	1.405×10^{-5}				
3	-2.235×10^{-3}	8.973×10^{-7}	2.042×10^{-8}			
4	-5.587×10^{-4}	5.640×10^{-8}	3.448×10^{-10}	2.621×10^{-11}		
5	-1.397×10^{-4}	3.530×10^{-9}	5.507×10^{-12}	1.204×10^{-13}	1.807×10^{-14}	
6	-3.492×10^{-5}	2.207×10^{-10}	8.657×10^{-14}	5.274×10^{-16}	5.551×10^{-17}	2.776×10^{-17}

$\boxed{2}$ a) Let $F(h)$ be our numerical approximation to some solution Q, and assume we have an error expansion given by

$$F(h) = Q + C_1 h + C_2 h^2 + C_3 h^3 + \cdots,$$

where the constants C_k are independent of h. Explain how you can make an extrapolation table based on the stepsize sequence $\{h/j\}$, $j = 1, 2, \ldots$.

Write a MATLAB code to test your algorithm on the forward difference approximation to the derivative of $f(x)$ in some point x_0. In this case

$$F(h) = \frac{f(x_0 + h) - f(x_0)}{h} \qquad \text{and} \qquad Q = f'(x_0).$$

Solution. Somewhat embarrassing, this was in fact more complicated to solve than first expected. By brute force, let $T_{j,1} = F(h/j)$ for $j = 1, 2, \ldots$, so that

$$T_{j-1,1} = Q + C_1 \frac{h}{j-1} + \sum_{p=2}^{\infty} \frac{1}{(j-1)^p} C_p h^p;$$

$$T_{j,1} = Q + C_1 \frac{h}{j} + \sum_{p=2}^{\infty} \frac{1}{j^p} C_p h^p.$$

We can eliminate the 1st-order term h by multiplying the second equation by j and the first by j and then subtract to get a 2nd-order approximation

$$T_{j,2} = jT_{j,1} - (j-1)T_{j-1,1} = Q + \sum_{p=2}^{\infty} K_{j,2}^{(p)} C_p h^p,$$

where

$$K_{j,2}^{(p)} = \frac{1}{j^{p-1}} - \frac{1}{(j-1)^{p-1}}.$$

Now, in a general extrapolation step we have

$$T_{j-1,k} = Q + K^{(k)}_{j-1,k} C_p h^k + \sum_{p=k+1}^{\infty} K^{(p)}_{j-1,k} C_p h^p;$$

$$T_{j,k} = Q + K^{(k)}_{j,k} C_p h^k \quad + \sum_{p=k+1}^{\infty} K^{(p)}_{j,k} C_p h^p.$$

In order to get a $(k+1)$th-order approximation, we do similarly as above, but also normalize the expressions so that the coefficient in front of Q is 1. This yields the extrapolation formula

$$T_{j,k+1} = \frac{\frac{1}{K^{(k)}_{j,k}} T_{j,k} - \frac{1}{K^{(k)}_{j-1,k}} T_{j-1,k}}{\frac{1}{K^{(k)}_{j,k}} - \frac{1}{K^{(k)}_{j-1,k}}} = Q + \sum_{p=k+1}^{\infty} K^{(p)}_{j,k+1} C_p h^p,$$

which also can be written as

$$T_{j,k+1} = T_{j,k} + \frac{T_{j,k} - T_{j-1,k}}{\frac{K^{(k)}_{j-1,k}}{K^{(k)}_{j,k}} - 1} = Q + \sum_{p=k+1}^{\infty} K^{(p)}_{j,k+1} C_p h^p,$$

with

$$K^{(p)}_{j,k+1} = K^{(p)}_{j,k} + \frac{K^{(p)}_{j,k} - K^{(p)}_{j-1,k}}{\frac{K^{(k)}_{j-1,k}}{K^{(k)}_{j,k}} - 1}.$$

The only thing we need to know in the formula is the fraction $K^{(k)}_{j-1,k}/K^{(k)}_{j,k}$, which for the first few values of k equals

$$\frac{K^{(1)}_{j-1,1}}{K^{(1)}_{j,1}} = \frac{j}{j-1}, \qquad \frac{K^{(2)}_{j-1,2}}{K^{(2)}_{j,2}} = \frac{j}{j-2}, \qquad \frac{K^{(3)}_{j-1,3}}{K^{(3)}_{j,3}} = \frac{j}{j-3} \qquad \text{and} \qquad \frac{K^{(4)}_{j-1,4}}{K^{(4)}_{j,4}} = \frac{j}{j-4}.$$

This indicates that

$$\frac{K^{(k)}_{j-1,k}}{K^{(k)}_{j,k}} = \frac{j}{j-k}, \tag{$\star$}$$

and we end up with the extrapolation formula

$$\begin{cases} T_{j,1} = F(h/j) & \text{for } j = 1, 2, \dots; \\ T_{j,k+1} = T_{j,k} + \dfrac{T_{j,k} - T_{j-1,k}}{\frac{j}{j-k} - 1} & \text{for } k = 1, 2, \dots, j-1. \end{cases}$$

A reward (a bag of Twist chocolate) is given to the first one of you who can give a direct proof of ($\star$). We know from the literature that the statement is true, because there are alternative procedures to derive the extrapolation formula.

A MATLAB implementation of this algorithm applied to the test problem can be found on the website. Note that we do not achieve errors close to machine precision. This occurs because $\frac{j}{j-k} \approx 1$ for sufficiently large j and small k, which leads to numerical instability from round-off error.

b) It can be proved that, using the forward Euler method

$$y_{n+1} = y_n + hf(t_n, y_n) \qquad \text{with} \qquad n = 0, \dots, N-1$$

to solve the IVP $y' = f(t, y)$ and $y(t_0) = y_0$ from t_0 to $t_0 + H$, the global error can be expressed as

$$y_N = y(t_0 + H) + C_1 h + C_2 h^2 + C_3 h^3 + \cdots, \qquad \text{where} \qquad h = \frac{H}{N}.$$

See if you can figure out how the extrapolation algorithm developed in a) applied to this problem can be expressed as explicit Runge–Kutta methods of arbitrary high order (with a lot of stages). The stepsize used in these RK-methods is H.

Hint: Do one extrapolation step at a time starting out with the explicit Euler method and find the ERK formulation of the 2nd-order approximation, then of the 3rd-order, *etc*. Identify the function evaluations, since

$$k_i = f\Big(t_0 + c_i H, y_0 + H \sum_{j=1}^{i-1} a_{ij} k_j\Big).$$

Solution. Let us demonstrate the idea for some small values of N. Every time an evaluation of f appears, that is, a stage derivative, we label it $k_{N,i}$ with index i in a natural manner. This yields

$$N = 1: \qquad y_1^{(1)} = y_0 + Hf(t_0, y_0) = y_0 + Hk_{1,1},$$

$$N = 2: \quad \begin{cases} y_1^{(2)} = y_0 + \frac{1}{2}Hf(t_0, y_0) = y_0 + \frac{1}{2}Hk_{2,1}, \\ y_2^{(2)} = y_1^{(2)} + \frac{1}{2}Hf\left(t_0 + \frac{1}{2}H, y_1^{(2)}\right) = y_0 + \frac{1}{2}H\left(k_{2,1} + k_{2,2}\right), \end{cases}$$

$$N = 3: \quad \begin{cases} y_1^{(3)} = y_0 + \frac{1}{3}Hf(t_0, y_0) = y_0 + \frac{1}{3}Hk_{3,1}, \\ y_2^{(3)} = y_1^{(3)} + \frac{1}{3}Hf\left(t_0 + \frac{1}{3}H, y_1^{(3)}\right) = y_0 + \frac{1}{3}H\left(k_{3,1} + k_{3,2}\right), \\ y_3^{(3)} = y_2^{(3)} + \frac{1}{3}Hf\left(t_0 + \frac{2}{3}H, y_2^{(3)}\right) = y_0 + \frac{1}{3}H\left(k_{3,1} + k_{3,2} + k_{3,3}\right), \end{cases}$$

Hence, the general pattern is

$$y_n^{(N)} = y_0 + \frac{H}{N}\sum_{i=1}^{n} k_{N,i} \qquad \text{with} \qquad k_{N,i} = f\Big(t_0 + (i-1)\frac{H}{N}, y_0 + \frac{H}{N}\sum_{j=1}^{i-1} k_{N,j}\Big).$$

Notice that $k_{N,1} = k_{1,1} = f(t_0, y_0)$ for all N.

Now we can construct higher-order methods with help of the extrapolation formula from a), by defining $T_{N,1} = y_N^{(N)}$ (replace j by N), to get

$$\begin{aligned} T_{2,2} &= 2T_{2,1} - T_{1,1} &&= y_0 + Hk_{2,1}, \\ T_{3,2} &= 3T_{3,1} - 2T_{2,1} &&= y_0 + H\left(-k_{2,2} + k_{3,2} + k_{3,3}\right), \\ T_{3,3} &= \tfrac{3}{2}T_{3,2} - \tfrac{1}{2}T_{2,2} &&= y_0 + H\left(-2k_{2,2} + \tfrac{3}{2}k_{3,2} + \tfrac{3}{2}k_{3,3}\right). \end{aligned}$$

Convince yourself that

$$T_{4,4} = y_0 + H\left[2k_{2,2} - \tfrac{9}{2}\left(k_{3,2} + k_{3,3}\right) + \tfrac{8}{3}\left(k_{4,2} + k_{4,3} + k_{4,4}\right)\right].$$

In conclusion, we observe that $T_{2,2}$, $T_{3,3}$ and $T_{4,4}$ can be considered as a solution after one step of an explicit RK-method of order 2, 3 and 4, respectively. Their corresponding Butcher tableux are

$$T_{2,2}: \quad \begin{array}{c|cc} 0 & 0 & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

$$T_{3,3}: \quad \begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & & & \\ 1/3 & 1/3 & 0 & & \\ 2/3 & 1/3 & 0 & 1/3 & \\ \hline & 0 & -2 & 3/2 & 3/2 \end{array}$$

$$T_{4,4}: \quad \begin{array}{c|ccccccc} 0 & 0 & & & & & & \\ 1/2 & 1/2 & & & & & & \\ 1/3 & 1/3 & 0 & & & & & \\ 2/3 & 1/3 & 0 & 1/3 & & & & \\ 1/4 & 1/4 & 0 & 0 & & & & \\ 2/4 & 1/4 & 0 & 0 & 0 & 1/4 & & \\ 3/4 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 & \\ \hline & 0 & 2 & -9/2 & -9/2 & 8/3 & 8/3 & 8/3 \end{array}$$

3 Let $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$.

a) Show that the spectral radius

$$\rho(A) = \max_{i=1}^{n} |\lambda_i|,$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A, is not a norm on $\mathbb{R}^{n\times n}$.

Solution. For example, let

$$A = \begin{bmatrix} 0 & \alpha \\ 0 & 0 \end{bmatrix},$$

where $\alpha \neq 0$. Then $\rho(A) = 0$, but $A \neq 0$.

b) Prove that the *Frobenius norm*

$$\|A\|_{\mathrm{F}} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2} = \sqrt{\operatorname{tr}(AA^\top)}$$

is a norm on $\mathbb{R}^{n\times n}$.

Solution. We prove the triangle inequality; the other properties are straightforward. To this end, let $a_{i\cdot} = (a_{i1}, \dots, a_{in})$ denote the ith row of A. Then we use the triangle and Cauchy-Schwarz inequalities in the vectorial ($\mathbb{R}^n$) 2-norm and find that

$$
\begin{aligned}
\|A+B\|_{\mathrm{F}}^2 = \sum_{i,j=1}^{n} |a_{ij} + b_{ij}|^2 &= \sum_{i=1}^{n} \|a_{i\cdot} + b_{i\cdot}\|_2^2 \\
&\leq \sum_{i=1}^{n} \left(\|a_{i\cdot}\|_2^2 + 2\|a_{i\cdot}\|_2 \|b_{i\cdot}\|_2 + \|b_{i\cdot}\|_2^2 \right) \\
&= \|A\|_{\mathrm{F}}^2 + 2\sum_{i=1}^{n} \|a_{i\cdot}\|_2 \|b_{i\cdot}\|_2 + \|B\|_{\mathrm{F}}^2 \\
&\leq \|A\|_{\mathrm{F}}^2 + 2\left(\sum_{i=1}^{n} \|a_{i\cdot}\|_2^2 \right)^{1/2} \left(\sum_{i=1}^{n} \|b_{i\cdot}\|_2^2 \right)^{1/2} + \|B\|_{\mathrm{F}}^2 \\
&= (\|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}})^2 .
\end{aligned}
$$

Now take square roots on both sides. (In order to better see the application of Cauchy-Schwarz' inequality, define the two vectors

$$
\overline{a} = \left(\|a_{1\cdot}\|_2, \dots, \|a_{n\cdot}\|_2 \right) \qquad \text{and} \qquad \overline{b} = \left(\|b_{1\cdot}\|_2, \dots, \|b_{n\cdot}\|_2 \right).
$$

Then

$$
\sum_{i=1}^{n} \|a_{i\cdot}\|_2 \|b_{i\cdot}\|_2 = \langle \overline{a}, \overline{b} \rangle_2 \leq \|\overline{a}\|_2 \, \|\overline{b}\|_2 = \left(\sum_{i=1}^{n} \|a_{i\cdot}\|_2^2 \right)^{1/2} \left(\sum_{i=1}^{n} \|b_{i\cdot}\|_2^2 \right)^{1/2},
$$

as desired.)

Comment: We have essentially shown that the 2-norm of a finite collection of 2-norms is a norm.

c) Establish that $\|\cdot\|_{\mathrm{F}}$ is submultiplicative, that is,

$$
\|AB\|_{\mathrm{F}} \leq \|A\|_{\mathrm{F}} \|B\|_{\mathrm{F}},
$$

and consistent/compatible with the Euclidean vector norm (2-norm), that is,

$$
\|Ax\|_2 \leq \|A\|_{\mathrm{F}} \|x\|_2.
$$

Solution. Let $a_{i\cdot} = (a_{i1}, \dots, a_{in})$ and $a_{\cdot j} = (a_{1j}, \dots, a_{nj})$ denote the ith row and jth column of A, respectively. Cauchy-Schwarz' inequality for the vectorial 2-norm yields

$$
\begin{aligned}
\|AB\|_{\mathrm{F}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left| \sum_{k=1}^{n} a_{ik} b_{kj} \right|^2 &= \sum_{i=1}^{n} \sum_{j=1}^{n} \left| \langle a_{i\cdot}, b_{\cdot j} \rangle_2 \right|^2 \\
&\leq \sum_{i=1}^{n} \sum_{j=1}^{n} \|a_{i\cdot}\|_2^2 \, \|b_{\cdot j}\|_2^2 = \sum_{i=1}^{n} \|a_{i\cdot}\|_2^2 \sum_{j=1}^{n} \|b_{\cdot j}\|_2^2 = \|A\|_{\mathrm{F}}^2 \, \|B\|_{\mathrm{F}}^2.
\end{aligned}
$$

As regards compatibility, we have

$$\|Ax\|_2^2 = \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right|^2 = \sum_{i=1}^n \left| \langle a_{i\cdot}, x \rangle_2 \right|^2 \leq \sum_{i=1}^n \|a_{i\cdot}\|_2^2 \, \|x\|_2^2 = \|A\|_{\mathrm{F}}^2 \, \|x\|_2^2,$$

again by Cauchy-Schwarz' inequality.