Computational Fluid Dynamics
Homework 1

Francisco Jose Castillo Carrasco

January 28, 2018

Problem 1

Solution: By doing some problems on my own, including the examples in class and more complex ones, I came to the general "rule" that if we use the Taylor table to obtain the approximation of the $k-th$ derivative using n nodes, $n > k$, we will obtain an approximation with a leading order error term of order $n-k$ and we will need up to the $n-th$ derivative in the Taylor table. For our case in this problem, since we want to approximate the second derivative at a point x_i using three nodes, we will obtain a first order approximation and we will need up to the third derivative in the Taylor table. We will check that this is true at the end of the problem. We start by showing the Taylor expansions of the function f at the three points of interest,

$$
\begin{aligned}
&f_{i-2} = f(x_i - 2h) = f_i - 2hf_i' + \frac{4h^2}{2}f_i'' - \frac{8h^3}{6}f_i''' + ...\\
&f_i = f(x_i) = f_i\\
&f_{i+1} = f(x_i + h) = f_i + hf_i' + \frac{h^2}{2}f_i'' + \frac{h^3}{6}f_i''' + ...
\end{aligned}
$$

Using these expansion we can easily fill the following Taylor table.

	f_i	f_i'	f_i''	f_i'''
f_i''	0	0	1	0
af_{i-2}	a	$-2ah$	$2ah^2$	$-8a\frac{h^3}{6}$
bf_i	b	0	0	0
cf_{i+1}	c	ch	$\frac{1}{2}ch^2$	$c\frac{h^3}{6}$

Table 1: Taylor table.

From the table above we can obtain the following system of equations,

$$
\begin{aligned}
a + b + c &= 0,\\
c - 2a &= 0,\\
1 + 2ah^2 + \frac{1}{2}ch^2 &= 0.
\end{aligned}
$$

From the second equation we know that $c = 2a$, and inserting it in the third one get

$$1 + 2ah^2 + ah^2 = 1 + 3ah^2 = 0,$$

which gives us

$$a = -\frac{1}{3h^2} \quad \Rightarrow \quad c = -\frac{2}{3h^2}.$$

Finally, plugging this values in the first equation we get

$$-\frac{1}{3h^2} + b - \frac{2}{3h^2} = b - \frac{1}{h^2} = 0,$$

which gives us

$$b = \frac{1}{h^2}.$$

Inserting the constants in the equation

$$f_i'' + af_{i-2} + bf_i + cf_{i+1} = (a+b+c)f_i + (c-2a)hf_i' + (1 + 2ah^2 + \frac{1}{2}ch^2)f_i'' + (c-8a)\frac{h^3}{6}f_i''' + ...$$

we obtain the desired approximation

$$f_i'' = \frac{f_{i-2} - 3f_i + 2f_{i+1}}{3h^2} + \frac{h}{3}f_i''' + ...$$

where the terms of f_i, f_i' and f_i'' of the right side of the equation have vanished. Thus, the approximation is of first order, since the leading order error term $\frac{h}{3}f_i'''$ is proportional to h. The aproximation can then be expressed as

$$f_i'' = \frac{f_{i-2} - 3f_i + 2f_{i+1}}{3h^2} + \mathcal{O}(h).$$

Problem 2

Solution: In this problem we are asked approximate the first derivative of the function

$$f(x) = \frac{1 + 2x^2 \cos x}{x^{2.4}},$$

at a given point $x_0 = 33.3$. We can calculate the derivative analytically,

$$f'(x) = \frac{2x^2(2\cos x - x \sin x) - 2.4(1 + 2x^2 \cos x)}{x^{3.4}},$$

and its value at x_0 is $f'(x_0) = -0.466342049195729$ using double precision in MATLAB. We are asked to plot, in logarithmic scale, the truncation error of three different approximations versus the step h. The approximations to use are

- First order forward diffrences:

$$f_i' = \frac{f_{i+1} - f_i}{h} + \mathcal{O}(h).$$

- Second order central diffrences:

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + \mathcal{O}(h^2).$$

- Sixth order central differences:

$$f'_i = \frac{-f_{i-3} + 9f_{i-2} - 45f_{i-1} + 45f_{i+1} - 9f_{i+2} + f_{i+3}}{60h} + \mathcal{O}(h^6).$$

Once the value of each approximation is obtained, we can compute the error for every value of the step h and show it in the next figure.

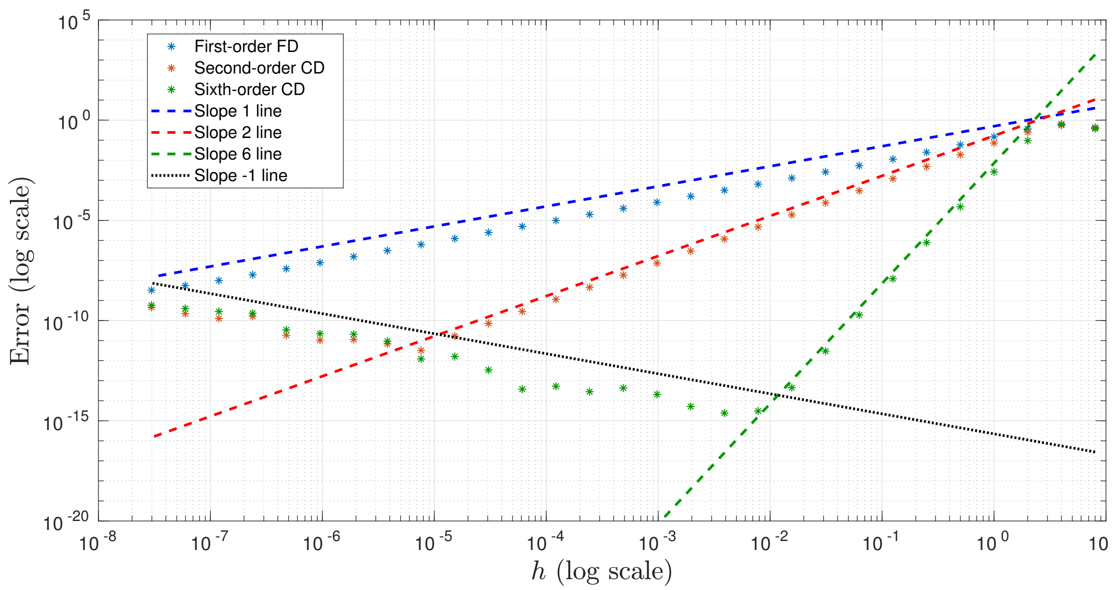

Figure 1: Error of the different approximations and reference lines.

The fact that the error describes a straight line of slope m in logarithmic scale proves that that approximation is of order m (since $\log{(kh^m)} = \log k + m \log h$, a straight line of slope m). In our case, it is clear in the figure that the approximations are, from top to bottom, indeed of order one, two and six, respectively. However, this linear dependence is broken in two areas of the figure.

First, when h is sufficiently small, the *rounding-error* is dominant and we see it as a cloud of points that seem parallel to the line of slope negative one. This is due to the fact that the rounding error is of the order of $\frac{\epsilon}{h}$, where ϵ is the well known machine epsilon. An error of that order is shown in logarithmic scale as a straight line of slope negative one (since $\log{(\frac{\epsilon}{h})} = \log \epsilon - \log h$ and ϵ is constant). The point where the rounding error becomes dominant depends, of course, of the order of approximation. The higher the order, as we make h small, the powers of h are even smaller and when they get of the order of ϵ the computer cannot differ well enough between numbers.

On the other hand, when h is large, the terms of the error that are not the leading error term become more and more important as h increases, reaching the point of not being neglectable. We can reach the point where the leading error term is not so anymore since we need h to be small in order to assume that the remaining terms of the series are considerably smaller. We can see that effect in the top right corner of the figure.

Matlab code for the problem

HOMEWORK 1 - FRANCISCO CASTILLO

Contents

Problem 2

```
format long;clear all;close all;clc
x_0=33.3;
f = @(x) (1+2*x^2*cos(x))/(x^(2.4));
fp = @(x) (2*x^2*(2*cos(x)-x*sin(x))-2.4*(1+2*x^2*cos(x)))/(x^3.4);
for k = -3:1:25
    H(k+4) = 2^(-k);
end
for k = 1:length(H)
    h = H(k);
    df1(k) = (f(x_0+h)-f(x_0))/h;
    df2(k) = (f(x_0+h)-f(x_0-h))/(2*h);
    df6(k) = (45*(f(x_0+h)-f(x_0-h))-9*(f(x_0+2*h)-f(x_0-2*h))...
        +(f(x_0+3*h)-f(x_0-3*h)))/(60*h);
end
linewidth=2;
darkgreen=[0 0.6 0];
figure('units','normalized','outerposition',[0 0 1 1])
loglog(H,abs(df1-fp(x_0)),'*',H,abs(df2-fp(x_0)),'*')
hold on
loglog(H,abs(df6-fp(x_0)),'*','Color',darkgreen)
loglog(H,H/2,'b--',H,H.^(2)/6,'r--','linewidth',linewidth)
loglog(H,H.^(6)/140,'--','Color',darkgreen,'linewidth',linewidth)
loglog(H,eps./H,'k:','linewidth',linewidth)
set(gca,'fontsize',14)
axis([1e-8 1e1 1e-20 1e5])
grid on
xlabel('$h$ (log scale)','fontsize',20,'interpreter','latex')
ylabel('Error (log scale)','fontsize',20,'interpreter','latex')
saveas(gcf,'IMAGES/problem2_1','epsc')
saveas(gcf,'IMAGES/problem2_1','fig')
```