MATH 335: Numerical Analysis

Problem Set 4, Solutions

⋆3 (a) This equation follows from taking 4 times equation (??) and subtracting (??), then solving for \( f'(x) \). One also must use the relationships \( d_4 = -c_4/4 \) and \( d_6 = -5c_6/16 \).

(b) Multiplying equation (??) by 16 and subtracting (??), we get

\[
15f'(x) = 16D_{3.2} - D_{2.2} - \frac{3}{4}d_6h^6 - \cdots ,
\]

or

\[
f'(x) = \frac{16}{15}D_{3.2} - \frac{1}{15}D_{2.2} - \frac{1}{20}d_6h^6 - \cdots \\
= D_{3.3} + O(h^6).
\]

9.2.10 Let us denote \( \int_a^b f(x) \, dx \) by \( J \). Then we have

\[
J = I(h) + K_1h + K_2h^3 + K_3h^5 + \cdots \quad (1)
\]

\[
J = I(h/3) + K_1\frac{h}{3} + K_2\frac{h^3}{27} + K_3\frac{h^5}{243} + \cdots \quad (2)
\]

\[
J = I(h/9) + K_1\frac{h}{9} + K_2\frac{h^3}{729} + K_3\frac{h^5}{59049} + \cdots \quad (3)
\]

Multiplying (2) by 3 and subtracting (1) we get (after solving again for \( J \))

\[
J = \frac{3}{2}I(h/3) - \frac{1}{2}I(h) + c_2h^3 + c_3h^5 + \cdots \quad (4)
\]

where \( c_2 = (-4/9)K_2 \) and \( c_3 = (-40/81)K_3 \). Multiplying (3) by 3 and subtracting (2) gives

\[
J = \frac{3}{2}I(h/9) - \frac{1}{2}I(h/3) + \frac{1}{27}c_2h^3 + \frac{1}{243}c_3h^5 + \cdots \quad (5)
\]

Equations (4) and (5) show that \([3I(h/3) - I(h)]/2\) and \([3I(h/9) - I(h/3)]/2\) both yield \( O(h^3) \) approximations to \( J \). But we can do even better by multiplying (5) by 27, subtracting (4), and then solving for \( J \) to get

\[
J = \frac{81}{52}I(h/9) - \frac{15}{26}I(h/3) + \frac{1}{52}I(h) + O(h^5).
\]

Thus, \( J \approx (81/52)(2.9795) - (15/26)(2.9263) + (2.3965)/(52) \approx 2.9990 \).

CP 9.2.2 Modification of the book’s routine doesn’t entail much—mostly commenting out the lines that do the printing (unless you still want it to be printed), and indicating that \( D \) is to be returned. Something like
function D = derive(f,h,a,n)
  % Approximate the derivative of a function at x=a
  %disp(' Derivative table')
  %disp('_________________________________________________________
  %disp(' i  h  Di,1  Di,2  Di,3  ... ')
  %disp('_________________________________________________________
  D(1,1)=(feval(f,a+h)-feval(f,a-h))/(2*h);
  %fprintf('%2.0f %8.4f %12.4f
',1,h,D(1,1));
  for i=1:n-1
    h=h/2;
    D(i+1,1)=(feval(f,a+h)-feval(f,a-h))/(2*h);
    % fprintf('%2.0f %8.4f %12.4f',i+1,h,D(i+1,1));
    for k=1:i
      D(i+1,k+1)=D(i+1,k)+(D(i+1,k)-D(i,k))/((4ˆk)-1);
      % fprintf('%12.4f',D(i+1,k+1));
    end
    % fprintf('
');
  end
end

(a) With format set to short, the output looks like

    34.96000  0.00000  0.00000  0.00000
    34.87000  34.84000  0.00000  0.00000
    34.84750  34.84000  34.84000  0.00000
    34.84187  34.84000  34.84000  34.84000

Here, as below, the best approximation to the derivative appears in the bottom
of the far-right column.

(b)  

    0.77850  0.00000  0.00000  0.00000
    0.78917  0.79273  0.00000  0.00000
    0.79177  0.79263  0.79262  0.00000
    0.79241  0.79262  0.79262  0.79262

(c)  

   -4.20387  0.00000  0.00000  0.00000
   -4.20821  -4.20966  0.00000  0.00000
   -4.20930  -4.20966  -4.20966  0.00000
   -4.20957  -4.20966  -4.20966  -4.20966

(d)  

    1.69655  0.00000  0.00000  0.00000

PS4—Solutions
AP 9.3 We are all going to guess at values here. Here are some possible values for the amounts at times surrounding $t = 4$:

<table>
<thead>
<tr>
<th>time</th>
<th>amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>4</td>
<td>755</td>
</tr>
<tr>
<td>6</td>
<td>730</td>
</tr>
<tr>
<td>7</td>
<td>700</td>
</tr>
<tr>
<td>16</td>
<td>250</td>
</tr>
<tr>
<td>12</td>
<td>480</td>
</tr>
<tr>
<td>10</td>
<td>590</td>
</tr>
<tr>
<td>9</td>
<td>650</td>
</tr>
</tbody>
</table>

These yield the Richardson table (with values approximating the rate of change at $t = 8$):

$$ D_{i,1} = \frac{f(x+h) - f(x-h)}{2h} $$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$h$</th>
<th>$D_{i,1}$</th>
<th>$D_{i,2}$</th>
<th>$D_{i,3}$</th>
<th>$D_{i,4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>-34.375</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>-34.375</td>
<td>-34.375</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>-35.000</td>
<td>-35.208</td>
<td>-35.264</td>
<td></td>
</tr>
</tbody>
</table>

3.1.8 The function has a double root, so its derivative $f'(x) = 4x^3 - 25.8x^2 - 71.02x + 464.4$ should have a single root at the same location. Using graphs of $f, f'$, it appears they have a common zero in the interval $[4, 4.5]$. Calling the routine bisect.m (and giving it $f'$ instead of $f$), we get

octave:382> function y = fprime(x)
> y = 4 * x.^3 - 25.8 * x.^2 - 71.02 * x + 464.4;
> endfunction

octave:383> bisect(@fprime, 4, 4.5, .000001, 1000)

| iter | a    | b    | c    | f(c)   | |b-a|/2 |
|------|------|------|------|--------|-----|-----|
|      |      |      |      |        |     |     |

PS4—Solutions
So, this zero of \( f \) is approximately 4.3. Similarly,

```matlab
octave:382> function y = f(x)
> y = x.^4 - 8.6 * x.^3 - 35.51 * x.^2 + 464.4 * x - 998.46;
> endfunction
```

```
octave:396> bisect(@f, -7.5, -7, 0.000001, 1000)
```

| iter | a     | b     | c     | f(c)     | |b-a|/2 |
|------|-------|-------|-------|----------|------|------|
| 10   | -7.3486 | -7.3481 | -7.348389 | -0.160641 | 0.000244 |
| 11   | -7.3486 | -7.3484 | -7.348511 | 0.082787  | 0.000122  |
| 15   | -7.3485 | -7.3485 | -7.348473 | 0.006714  | 0.000008  |
| 16   | -7.3485 | -7.3485 | -7.348469 | -0.000893 | 0.000004  |
| 17   | -7.3485 | -7.3485 | -7.348471 | 0.002911  | 0.000002  |

3.1.13 (a) We have that \( f \) is continuous on \([1,2]\) with \( f(1) = 1 > 0 \) and \( f(2) = -\ln 2 < 0 \). By the IVT there is at least one zero of \( f \) in \([1,2]\). To see that there is at most one, we note that \( f'(x) = 2(x - 2) - 1/x \) and \( f''(x) = 2 + 1/x^2 \). The latter shows that on \([1,2]\), \( f''(x) \) is always of the same sign, so \( f'(x) \) increases on \([1,2]\) reaching its maximum value (on that interval) at \( f'(2) = -1/2 \). Thus, \( f(x) \) decreases on \([1,2]\) and can have at most one zero there.

(b) As we know the root lies between 1 and 2, 6 decimal digit accuracy would be accurate to with 0.000005.
octave:382> function y = f(x)
> y = (x - 2).^2 - log(x);
> endfunction

octave:399> bisect(@f, 1, 2, 0.000005, 1000)

| iter |   a   |   b   |   c   | f(c)   | |b-a|/2 |
|------|-------|-------|-------|--------|------|------|
|  0   | 1.0000| 2.0000| 1.5000| -0.155465| 0.500000 |      |
|  1   | 1.0000| 1.5000| 1.2500| 0.339356  | 0.250000 |      |
| 12   | 1.4124| 1.4126| 1.412476| -0.000159 | 0.000122 |      |
| 13   | 1.4124| 1.4125| 1.412415| -0.000044 | 0.000061 |      |
| 14   | 1.4124| 1.4124| 1.412384| 0.000013  | 0.000031 |      |
| 15   | 1.4124| 1.4124| 1.412399| -0.000015 | 0.000015 |      |
| 16   | 1.4124| 1.4124| 1.412392| -0.000001 | 0.000008 |      |

(c) According to the formula on p. 51, we must solve

\[
\frac{2 - 1}{2^{n+1}} \leq \frac{1}{10^4} \quad \Rightarrow \quad n \geq 13.
\]