Worked solution - Tutorial Sheet 3

restart;read "C:/JF/Software/Maple/Start.mpl";Digits:=6:with(plots):

Question 1

$$GVFE := \frac{S - \frac{Q^2}{K^2}}{1 - \frac{\beta Q^2 B}{g A^3}} :$$

subs(beta=0,%);

$$S - \frac{Q^2}{K^2} \quad \textbf{(1.1)}$$

$$subs\left(Q = q\,B, K = \frac{k_{St} A^{\frac{5}{3}}}{P^{\frac{2}{3}}}, A = h\,B, P = B, q = k_{St}\, h_0^{\left(\frac{5}{3}\right)} S^{\frac{1}{2}}, \% \right)$$

$$S - \frac{h_0^{(10/3)}\, S\, B^{10/3}}{(h\,B)^{10/3}} \quad \textbf{(1.2)}$$

Unfortunately that was a bit clumsy, being unable to cancel common terms. We write the differential equation by hand:

diff(h(x),x)=S*(1-h[0]^(10/3)/h(x)^(10/3));

$$\frac{d}{dx}\, h(x) = S\left(1 - \frac{h_0^{(10/3)}}{h(x)^{10/3}}\right) \quad \textbf{(1.3)}$$

Introduce the numbers given:

$S := 10.^{-4} : k[St] := 50 : h[0] := 2 : \beta := 1.05 : g := 9.8 :$

Calculate discharge

q:=k[St]*h[0]^(5./3.)*sqrt(S);

$$q := 1.58741 \quad \textbf{(1.4)}$$

Now test the inertia term:

$$\frac{\beta q^2}{g h_0^3}$$

$$0.0337483 \quad \textbf{(1.5)}$$

Only 3% - we will ignore it now.

Question 2 and 3

The differential equation is

```
Diff(h,x)=S*(1-h[0]^(10/3)/h^(10/3));
```

$$\frac{\partial}{\partial x} h = 0.000100000 - \frac{0.000800000\, 2^{1/3}}{h^{10/3}} \tag{2.1}$$

Initial conditions

$h[1] := 3 : Z[1] := 0 : \eta[1] := h[1] + Z[1] :$

$\delta := 5000 :$

$x[1] := 0. :$

Euler method

```
for i to 40 do
  x[i+1] := x[i]-delta;
  h[i+1] := h[i]-delta*S*(1-h[0]^(10./(3.))/h[i]^(10./(3.))); The differential equation
  Z[i+1] := Z[i]+delta*S; Points on the bed
  print(i+1, x[i+1], h[i+1]);
  if h[i+1] < h[0]+0.1e-1 then N := i+1; break end if ;
end do:
```

$$\begin{gathered} 2,\ -5000.,\ 2.62942 \\ 3,\ -10000.,\ 2.33027 \\ 4,\ -15000.,\ 2.13068 \\ 5,\ -20000.,\ 2.03557 \\ 6,\ -25000.,\ 2.00703 \end{gathered} \tag{2.2}$$

It has always surprised me how far a backwater curve extends for gentle slopes: here 25km to get to within 1cm of normal flow

Questions 4, 5, 6

Halve the step size: $\delta := \frac{1\,\delta}{2} : x2_1 := 0 : h2_1 := h[1] : Z2_1 := Z[1] : \eta 2_1 := \eta[1] :$

$Uniform_1 := h[0] :$

Euler method, halving the steps

```
for i to 100 do
   x2[i+1] := x2[i]-delta;
   h2[i+1] := h2[i]-delta*S*(1-h[0]^(10./(3.))/h2[i]^(10./(3.)));
   Z2[i+1] := Z2[i]+delta*S;
   Uniform[i+1]:=h[0]-S*x2[i+1];
```

```
    print(i+1, x2[i+1], h2[i+1]);
    if h2[i+1] < h[0]+0.1e-1 then N2 := i+1; break end if
end do:
```

2, −2500, 2.81471

3, −5000, 2.64474

4, −7500, 2.49324

5, −10000, 2.36314

6, −12500, 2.25649

7, −15000, 2.17370

8, −17500, 2.11310

9, −20000, 2.07122

10, −22500, 2.04370

11, −25000, 2.02632

12, −27500, 2.01566

13, −30000, 2.00924 **(3.1)**

Accurate numerical method (not asked for) - uses in-built software

```
xx:='xx':
diff(H(xx),xx)=S*(1-h[0]^(10/3)/H(xx)^(10/3));
Sol:=dsolve({%,H(0)=3},numeric):
SS:=[]:TT:=[]:
for xx from -25000 to 0 by 1000 do
hh:=op([2,2],Sol(xx));
SS:=[op(SS),[xx,hh]];
TT:=[op(TT),[xx,hh-S*xx]];
od:
```

$$\frac{d}{dxx} H(xx) = 0.000100000 - \frac{0.000800000\, 2^{1/3}}{H(xx)^{10/3}} \qquad \textbf{(3.2)}$$

Richardson extrapolation and comparison with accurate solution

```
printf("        x  Euler 1 Euler 2 Richardson Accurate");
for i from 0 to N-1 do
Rich[i+1]:=2*h2[2*i+1]-h[i+1];
printf("\n%10.0f %6.3f %6.3f    %6.3f    %6.3f",x[i+1],h[i+1],h2[2*i+1],Rich[i+1],
op([2,2],evalf(Sol(x[i+1]),4)));
od:
```

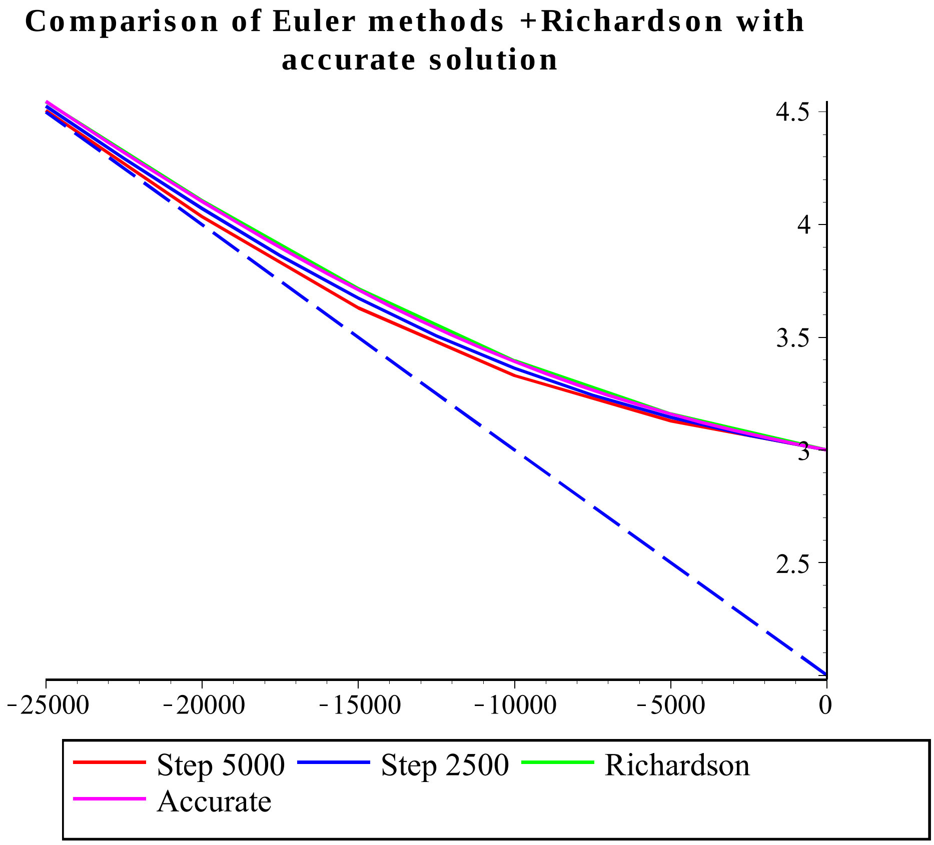
x	Euler 1	Euler 2	Richardson	Accurate
0	3.000	3.000	3.000	3.000
-5000	2.629	2.645	2.660	2.660
-10000	2.330	2.363	2.396	2.393
-15000	2.131	2.174	2.217	2.210

```
-20000  2.036  2.071    2.107    2.103
-25000  2.007  2.026    2.046    2.048
```

```
Legend:=["Step 5000","Step 2500","Richardson","Accurate",""]:
```

Plot elevation

```
plot([[[x[j],Z[j]+h[j]]$j=0..N],[[x2[j],Z2[j]+h2[j]]$j=0..N2-2],[[x[j],Z[j]+Rich[j]]$j=0..N],TT,[[x2[j],Uniform[j]]$j=0..N2-2]],title="Comparison of Euler methods + Richardson with accurate solution",legend=Legend,color=[red,blue,green,magenta,blue],linestyle=[solid$4,dash],titlefont=[Times,bold,12]);
```



Our results, with Richardson extrapolation, are very accurate!

A question you might have asked: the lecturer seems preoccupied with getting an accurate solution, yet at the start he ignored the Froude number term. How would that have changed results? We use the built-in software

```
xx:='xx':
```

The differential equation is

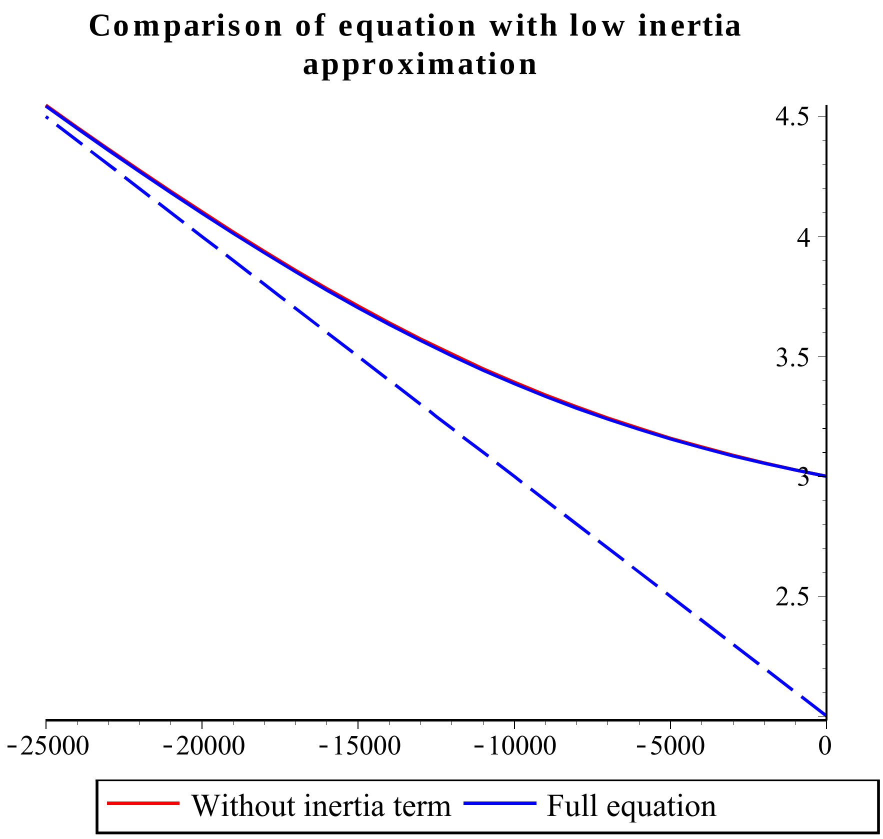
```
DE[2]:=diff(H(xx),xx)=S*(1-h[0]^(10/3)/H(xx)^(10/3))/(1-beta*q^2/g/H(xx)^3);
```

(4.1)

$$DE_2 := \frac{d}{dxx} H(xx) = \frac{0.000100000 \left(1 - \frac{8\, 2^{1/3}}{H(xx)^{10/3}}\right)}{1 - \frac{0.269986}{H(xx)^3}} \tag{4.1}$$

Use package software to solve:

```
Sol2:=dsolve({DE[2],H(0)=3},numeric):
SSS:=[]:
for xx from -25000 to 0 by 1000 do
hh:=op([2,2],Sol2(xx));
SSS:=[op(SSS),[xx,hh-S*xx]];
od:
plot([TT,SSS,[[x2[j],Uniform[j]]$j=0..N2-2]],title="Comparison of equation with low inertia approximation",legend=["Without inertia term","Full equation",""],color=[red,blue$2],linestyle=[solid$2,dash],titlefont=[Times,bold,12]);
```



The approximation of ignoring the Froude number term was quite good - it was 3%, if you remember, and the solution seems to be accurate to within that.

Question 7

Now solve the problem for slope 0.001 - but with smaller depths (on a steep slope it is probably a bit much if we were to use the same normal depth of 2m)

$S := 0.001; h[0] := 1; h[1] := 1.5$

$$S := 0.001$$
$$h_0 := 1$$
$$h_1 := 1.5 \quad \textbf{(5.1)}$$

$q := k[St]\, h_0^{\left(\frac{5.}{3.}\right)} \sqrt{S}$

$$q := 1.58114 \quad \textbf{(5.2)}$$

Check inertia term: $\dfrac{\beta q^2}{g h_0^3}$

$$0.267857 \quad \textbf{(5.3)}$$

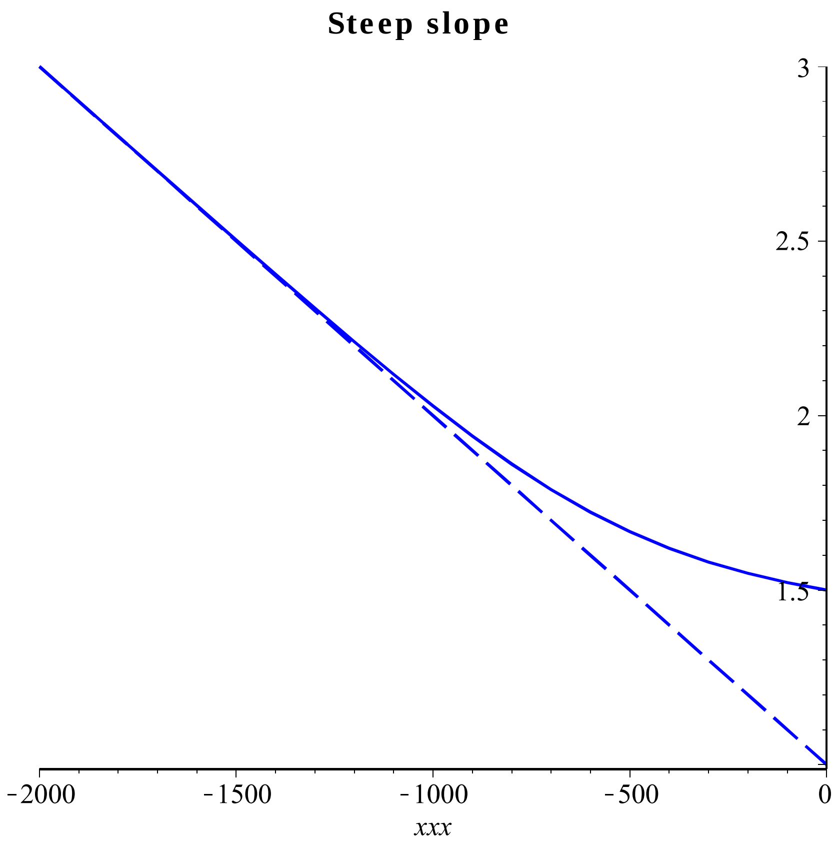
Now we can't neglect it!

xx:='xx':

$$DE[3] := \frac{d}{dxx} H(xx) = \frac{S\left(1 - \dfrac{h_0^{\left(\frac{10}{3}\right)}}{H(xx)^{\frac{10}{3}}}\right)}{1 - \dfrac{\beta q^2}{g H(xx)^3}}:$$

The lecturer is feeling lazy: he will just use the package software

```
Sol3:=dsolve({DE[3],H(0)=h[1]},numeric):
SSS:=[]:
for xx from -2000 to 0 by 100 do
hh:=op([2,2],Sol3(xx));
SSS:=[op(SSS),[xx,hh-S*xx]];
od:
plot([SSS,1-xxx*S],xxx=-2000..0,title="Steep slope",legend=["",""],color=[blue],
linestyle=[solid,dash],titlefont=[Times,bold,12]);
```



Now the length of the backwater curve is only about 1km, compared with 25km for the gentle slope case. In lectures we had the formula for the decay constant $\gamma = \frac{3\,S}{h[0]}$: Hence, the "length" of the backwater curve is proportional to $\frac{h[0]}{S}$: In our first example this was 2/10^(-4) = 20000: and in the second, 1/10^(-3)=1000, both of which are roughly right.