Homework 7

```
> restart;
```

Problem 1

The differential equation is y' = f(t,y) on [0,1], where y(0)=1/3. We need bounds M and L.

```
> f := (t,y) -> -5*y+5*t^2+2*t;
```

$$f := (t, y) \mapsto -5\,y + 5\,t^2 + 2\,t \qquad \textbf{(1)}$$

Since

```
> diff(f(t,y),y);
```

$$-5 \qquad \textbf{(2)}$$

We have

```
> L:= 5;
```

$$L := 5 \qquad \textbf{(3)}$$

y"(t) is

```
> diff(f(t,y),y)*f(t,y)+diff(f(t,y),t);
```

$$-25\,t^2 + 25\,y + 2 \qquad \textbf{(4)}$$

M is the max of the absolute value of y"(t)

At any point t, a multiple of h, the bound is

```
> h:=0.1;
  B:= t -> M/100*(exp(5*t)-1);
```

$$h := 0.1$$

$$B := t \mapsto \frac{M\left(e^{5\,t} - 1\right)}{100} \qquad \textbf{(5)}$$

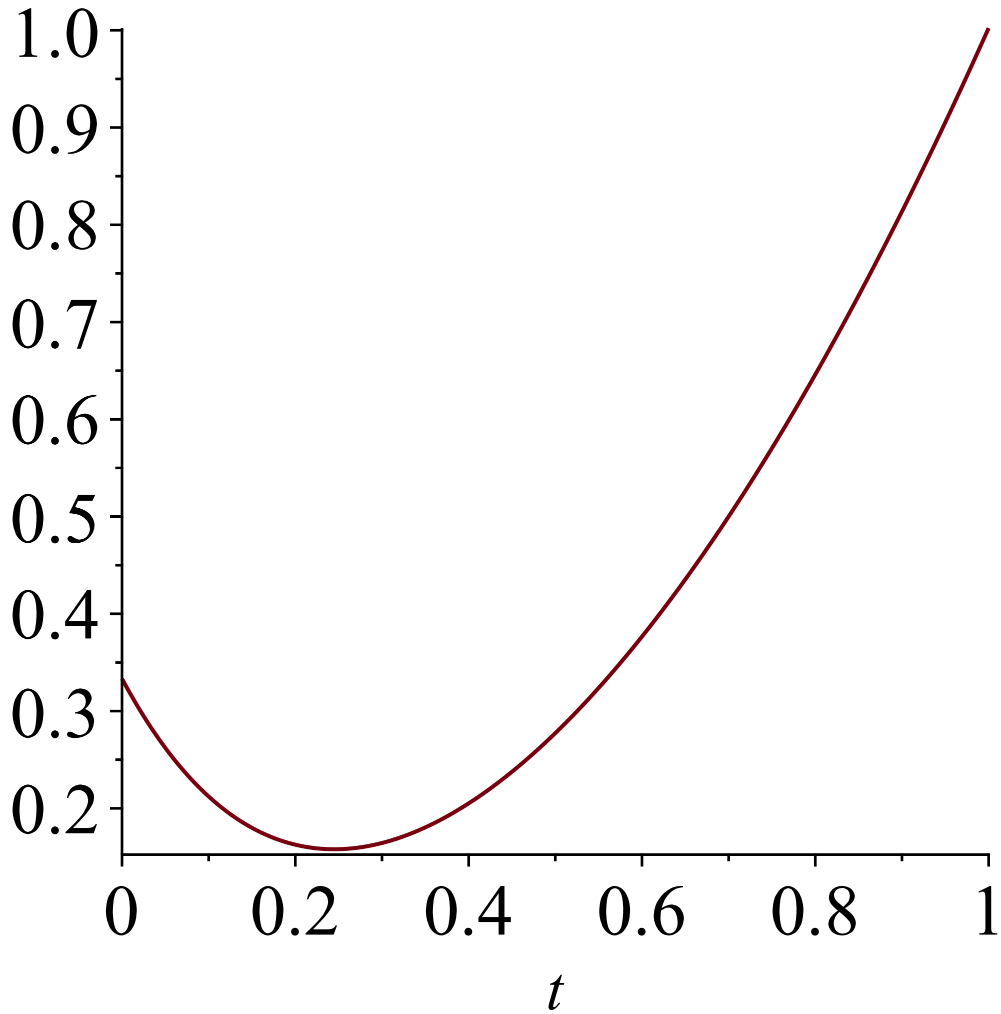
M is a problem. We can still see the growth is fast, e.g., exp(5) is around 148. To be more precise, let's use the easily computable solution

```
> g:= t-> t^2+exp(-5*t)/3;
```

$$g := t \mapsto t^2 + \frac{e^{-5\,t}}{3} \qquad \textbf{(6)}$$

Looking at

```
> plot(g(t),t=0..1);
```



the max occurs at 1, i.e.,

```
> M:= g(1.0);
```

$$M := 1.002245982 \quad (7)$$

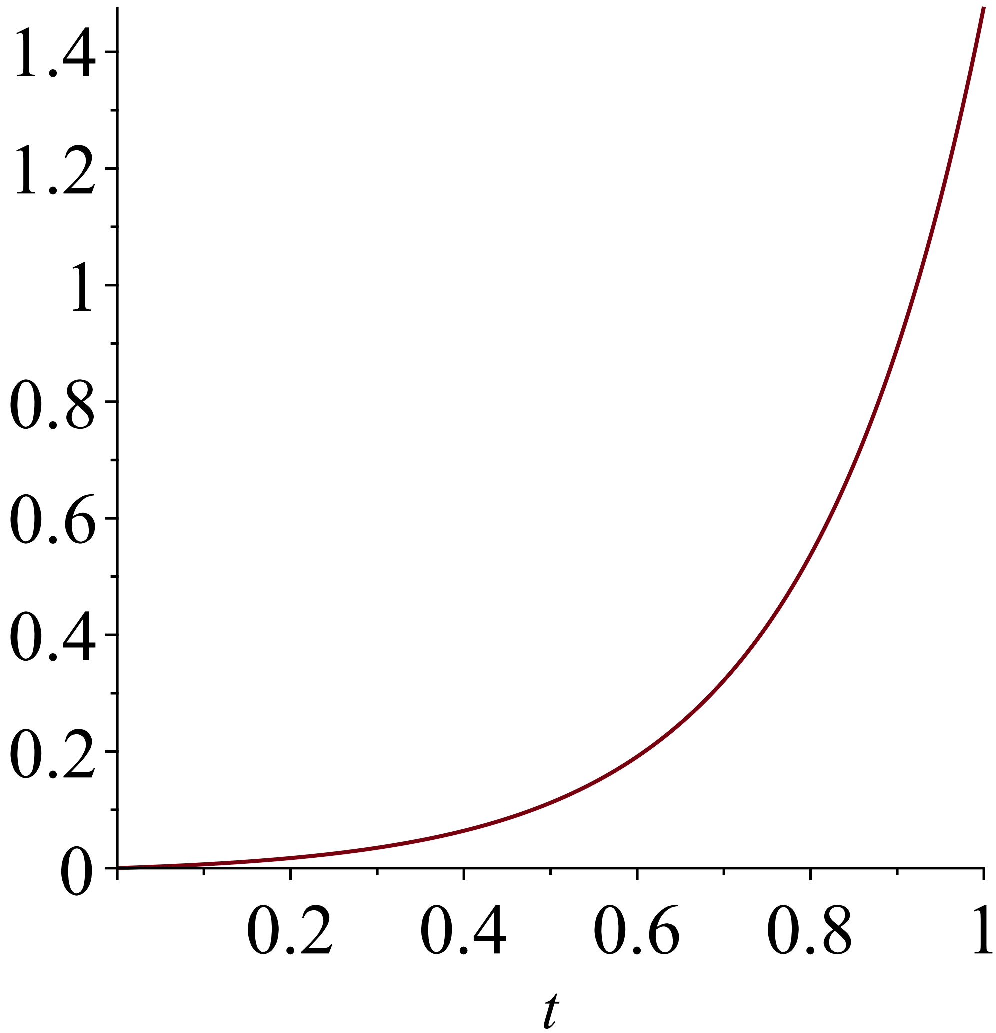
So

```
> B(t);
```

$$0.01002245982\, e^{5t} - 0.01002245982 \quad (8)$$

This is pretty bad error bound

```
> plot(B(t),t=0..1);
```



Problem 1a on page 280

```
> f:= (t,y) -> t*exp(3*t)-2*y;
  d2y:= unapply(diff(f(t,y),y)*f(t,y)+diff(f(t,y),t),t,y);
```

$$f := (t, y) \mapsto t\, e^{3\,t} - 2\,y$$

$$d2y := (t, y) \mapsto t\, e^{3\,t} + 4\,y + e^{3\,t} \quad \textbf{(9)}$$

A single step of the order 2 taylor method starting at the approximation W at t with step h=0.5 is

```
> OneStep2:= (t,W) -> W + f(t,W)*0.5+d2y(t,W)*0.5^2/2;
```

$$OneStep2 := (t, W) \mapsto W + f(t, W)\, 0.5 + \frac{d2y(t, W)\, 0.5^2}{2} \quad \textbf{(10)}$$

```
> w[0]:= 0;
```

$$w_0 := 0 \quad \textbf{(11)}$$

```
> for j from 1 to 2 do
  w[j] := OneStep2((j-1)*0.5,w[j-1]);
  od;
```

$$w_1 := 0.1250000000$$

$$w_2 := 2.023238968 \quad \textbf{(12)}$$

Problem 3a on page 280

```
> f:= (t,y) -> t*exp(3*t)-2*y;
```

```
d2y:= unapply(diff(f(t,y),y)*f(t,y)+diff(f(t,y),t),t,y);
d3y := unapply(diff(d2y(t,y),y)*f(t,y)+diff(d2y(t,y),t),t,y);
d4y := unapply(diff(d3y(t,y),y)*f(t,y)+diff(d3y(t,y),t),t,y);
```

$$f := (t, y) \mapsto t\,e^{3t} - 2y$$

$$d2y := (t, y) \mapsto t\,e^{3t} + 4y + e^{3t}$$

$$d3y := (t, y) \mapsto 7t\,e^{3t} - 8y + 4\,e^{3t}$$

$$d4y := (t, y) \mapsto 13t\,e^{3t} + 16y + 19\,e^{3t} \quad \textbf{(13)}$$

A single step of the order 4 taylor method starting at the approximation W at t with step h=0.5 is

```
> OneStep4:= (t,W) -> W + f(t,W)*0.5 + d2y(t,W)*0.5^2/2 + d3y(t,W)*
0.5^3/3! + d4y(t,W)*0.5^4/4!;
```

$$OneStep4 := (t, W) \mapsto W + f(t, W)\,0.5 + \frac{d2y(t, W)\,0.5^2}{2} + \frac{d3y(t, W)\,0.5^3}{3!} + \frac{d4y(t, W)\,0.5^4}{4!} \quad \textbf{(14)}$$

```
> w[0]:= 0;
for j from 1 to 2 do
w[j] := OneStep4((j-1)*0.5,w[j-1]);
od;
```

$$w_0 := 0$$

$$w_1 := 0.2578125000$$

$$w_2 := 3.055294737 \quad \textbf{(15)}$$

Let's compare it to the output of the 4th order runge-kutta method

```
> rkStep := proc(f,a,b,w)
local W,K1,K2,K3,K4,h;
h := b-a;
K1 := h*f(a,w);
K2 := h*f(a+h/2,w+K1/2);
K3 := h*f(a+h/2,w+K2/2);
K4 := h*f(b,w+K3);
W := w+(K1+2*K2+2*K3+K4)/6;
return W
end proc;
```

rkStep := **proc**(*f*, *a*, *b*, *w*) **(16)**

local *W*, *K1*, *K2*, *K3*, *K4*, *h*;

$h := b - a;$

$K1 := h * f(a, w);$

$K2 := h * f(a + 1/2 * h, w + 1/2 * K1);$

$K3 := h * f(a + 1/2 * h, w + 1/2 * K2);$

$K4 := h * f(b, w + K3);$

$W := w + 1/6 * K1 + 1/3 * K2 + 1/3 * K3 + 1/6 * K4;$

return *W*

end proc

```
> w[0]:= 0;
for j from 1 to 2 do
w[j] := rkStep(f,(j-1)*0.5,j*0.5,w[j-1]);
od;
```

$$w_0 := 0$$
$$w_1 := 0.2969974622$$
$$w_2 := 3.314311778 \quad \textbf{(17)}$$

Let's makes comparisons to the actual solution

```
> ode := diff(u(t),t)=f(t,u(t));
```

$$ode := \frac{d}{dt}\, u(t) = t\, e^{3t} - 2\, u(t) \quad \textbf{(18)}$$

```
> dsolve(ode);
```

$$u(t) = \left(\frac{(5\, t - 1)\, e^{5t}}{25} + \_C1 \right) e^{-2t} \quad \textbf{(19)}$$

Let's not forget the initial condition (to fix the constant _C1

```
> ic := u(0)=0;
```

$$ic := u(0) = 0 \quad \textbf{(20)}$$

```
> dsolve({ic,ode});
```

$$u(t) = \left(\frac{(5\, t - 1)\, e^{5t}}{25} + \frac{1}{25} \right) e^{-2t} \quad \textbf{(21)}$$

```
> g:= t-> ((1/25)*(5*t-1)*exp(5*t)+1/25)*exp(-2*t);
```

$$g := t \mapsto \left(\frac{(5\, t - 1)\, e^{5t}}{25} + \frac{1}{25} \right) e^{-2t} \quad \textbf{(22)}$$

```
> g(0);
  g(0.5);
  g(1.0);
```

$$0$$
$$0.2836165219$$
$$3.219099319 \quad \textbf{(23)}$$

Why is the 4th order runge-kutta better than the 4th order taylor method (which is the gold standard for 4th order methods)?
Basically because runge-kutta methods contain (typically) some terms which are fifth order, which can at least in this case
give a slightly better approximation than expected.

Problem 10a on page 280

```
> f := (t,y) -> 1/t^2-y/t-y^2;
  d2y:= unapply(diff(f(t,y),y)*f(t,y)+diff(f(t,y),t),t,y);
  OneStep2:= (t,W) -> W + f(t,W)*0.05+d2y(t,W)*0.05^2/2;
```

$$f := (t, y) \mapsto \frac{1}{t^2} - \frac{y}{t} - y^2$$

$$d2y := (t, y) \mapsto \left(-\frac{1}{t} - 2\, y \right) \left(\frac{1}{t^2} - \frac{y}{t} - y^2 \right) - \frac{2}{t^3} + \frac{y}{t^2}$$

$$OneStep2 := (t, W) \mapsto W + f(t, W)\, 0.05 + \frac{d2y(t, W)\, 0.05^2}{2} \quad \textbf{(24)}$$

```
> w[0] := -1;
```

$$w_0 := -1 \quad \textbf{(25)}$$

```
> for j from 1 to 20 do
```

```
w[j]:= OneStep2(1+(j-1)*0.05,w[j-1]):
err[j]:= abs(w[j]+1/(1+0.05*j)):
print(1+j*0.05,w[j],err[j]);
od:
```

1.05, −0.9525000000, 0.0001190476
1.10, −0.9093137897, 0.0002228806
1.15, −0.8698798963, 0.0003146789
1.20, −0.8337301905, 0.0003968572
1.25, −0.8004712717, 0.0004712717
1.30, −0.7697701344, 0.0005393652
1.35, −0.7413430111, 0.0006022704
1.40, −0.7149465996, 0.0006608853
1.45, −0.6903711003, 0.0007159279
1.50, −0.6674346434, 0.0007679767
1.55, −0.6459787921, 0.0008175018
1.60, −0.6258648873, 0.0008648873
1.65, −0.6069710557, 0.0009104496
1.70, −0.5891897450, 0.0009544509
1.75, −0.5724256807, 0.0009971093
1.80, −0.5565941630, 0.0010386074
1.85, −0.5416196401, 0.0010790996
1.90, −0.5274345051, 0.0011187156
1.95, −0.5139780792, 0.0011575664
2.00, −0.5011957462, 0.0011957462 **(26)**

Let's program the modified Euler and midpoint methods for y'=f(t,y)

```
> ModEulerStep := proc(f,a,b,w)
local W,h;
h := b-a;
W := w+(f(a,w)+f(b,w+f(a,w)*h))*h/2;
return W
end proc;
```

ModEulerStep := **proc**(*f*, *a*, *b*, *w*) **(27)**

local *W*, *h*;

$h := b - a$; $W := w + 1/2 * (f(a, w) + f(b, w + f(a, w) * h)) * h$; **return** *W*

end proc

```
> MidpointStep := proc(f,a,b,w)
local W,h;
h := b-a;
W := w+(f((a+b)/2,w+f(a,w)*h/2))*h;
return W
end proc;
```

MidpointStep := **proc**(*f*, *a*, *b*, *w*) **(28)**

local *W*, *h*;

$h := b - a$; $W := w + f(1/2 * a + 1/2 * b, w + 1/2 * f(a, w) * h) * h$; **return** *W*

end proc

Problems 1d, 5d, 13d on page 291

```
> f:= (t,y) -> cos(2*t)+sin(3*t);
  g:= t -> sin(2*t)/2-cos(3*t)/3+4/3;
```

$$f := (t, y) \mapsto \cos(2\,t) + \sin(3\,t)$$

$$g := t \mapsto \frac{\sin(2\,t)}{2} - \frac{\cos(3\,t)}{3} + \frac{4}{3} \quad \textbf{(29)}$$

```
> w[0]:=1;
  for j from 1 to 4 do
  w[j]:= ModEulerStep(f,(j-1)*0.25,j*0.25,w[j-1]):
  err[j]:= abs(w[j]-g(j*0.25)):
  print(j*0.25,w[j],err[j]);
  od:
```

$$w_0 := 1$$

0.25, 1.319902665, 0.009247148
0.50, 1.707029992, 0.023459766
0.75, 2.005355953, 0.036116081
1.00, 2.077078899, 0.040900646 **(30)**

```
> w[0]:=1;
  for j from 1 to 4 do
  w[j]:= MidpointStep(f,(j-1)*0.25,j*0.25,w[j-1]):
  err[j]:= abs(w[j]-g(j*0.25)):
  print(j*0.25,w[j],err[j]);
  od:
```

$$w_0 := 1$$

0.25, 1.333796238, 0.004646425
0.50, 1.742285354, 0.011795596
0.75, 2.059637390, 0.018165356
1.00, 2.138555951, 0.020576406 **(31)**

```
> w[0]:=1;
  for j from 1 to 4 do
  w[j]:= rkStep(f,(j-1)*0.25,j*0.25,w[j-1]):
  err[j]:= abs(w[j]-g(j*0.25)):
  print(j*0.25,w[j],err[j]);
  od:
```

$$w_0 := 1$$

0.25, 1.329165048, 0.000015235
0.50, 1.730533568, 0.000043810
0.75, 2.041543579, 0.000071545
1.00, 2.118063602, 0.000084057 **(32)**

```
> restart;
```

Problem 1a on page 337: we will use the l1 norm

```
> f1 := (t,u1,u2) -> 3*u1+2*u2-(2*t^2+1)*exp(2*t);
  f2 := (t,u1,u2) -> 4*u1+u2+(t^2+2*t-4)*exp(2*t);
  g1:= t -> exp(5*t)/3-exp(-t)/3+exp(2*t);
  g2:= t -> exp(5*t)/3+2*exp(-t)/3+t^2*exp(2*t);
```

$$f1 := (t, u1, u2) \mapsto 3\,u1 + 2\,u2 - \left(2\,t^2 + 1\right)\,e^{2\,t}$$

$$f2 := (t, u1, u2) \mapsto 4\,u1 + u2 + \left(t^2 + 2\,t - 4\right)\,e^{2\,t}$$

$$g1 := t \mapsto \frac{e^{5t}}{3} - \frac{e^{-t}}{3} + e^{2t}$$
$$g2 := t \mapsto \frac{e^{5t}}{3} + \frac{2\,e^{-t}}{3} + t^2\,e^{2t} \quad \mathbf{(33)}$$

Lets check

```
> expand(diff(g1(t),t)-f1(t,g1(t),g2(t)));
  expand(diff(g2(t),t)-f2(t,g1(t),g2(t)));
```

$$0$$
$$0 \quad \mathbf{(34)}$$

using vectors, we can write a general runge-kutta, but let's for now just do it quickly

```
> rkVect2Step := proc(f1,f2,a,b,w1,w2)
  local j, K11,K21,K31,K41,K12,K22,K32,K42,h,W1,W2;
  h := b-a;

  K11 := h*f1(a,w1,w2);
  K12 := h*f2(a,w1,w2);

  K21 := h*f1(a+h/2,w1+K11/2,w2+K12/2);
  K22 := h*f2(a+h/2,w1+K11/2,w2+K12/2);

  K31 := h*f1(a+h/2,w1+K21/2,w2+K22/2);
  K32 := h*f2(a+h/2,w1+K21/2,w2+K22/2);

  K41 := h*f1(b,w1+K31,w2+K32);
  K42 := h*f2(b,w1+K31,w2+K32);

  W1 := w1+(K11+2*K21+2*K31+K41)/6;
  W2 := w2+(K12+2*K22+2*K32+K42)/6;

  return( [W1,W2])
  end proc;
```

```
rkVect2Step := proc(f1, f2, a, b, w1, w2)                                      (35)
    local j, K11, K21, K31, K41, K12, K22, K32, K42, h, W1, W2;
    h := b - a;
    K11 := h*f1(a, w1, w2);
    K12 := h*f2(a, w1, w2);
    K21 := h*f1(a + 1/2*h, w1 + 1/2*K11, w2 + 1/2*K12);
    K22 := h*f2(a + 1/2*h, w1 + 1/2*K11, w2 + 1/2*K12);
    K31 := h*f1(a + 1/2*h, w1 + 1/2*K21, w2 + 1/2*K22);
    K32 := h*f2(a + 1/2*h, w1 + 1/2*K21, w2 + 1/2*K22);
    K41 := h*f1(b, w1 + K31, w2 + K32);
    K42 := h*f2(b, w1 + K31, w2 + K32);
    W1 := w1 + 1/6*K11 + 1/3*K21 + 1/3*K31 + 1/6*K41;
    W2 := w2 + 1/6*K12 + 1/3*K22 + 1/3*K32 + 1/6*K42;
    return [W1, W2]
end proc
```

```
> w1:=1; w2:=1;hh:=0.2;
  for j from 1 to 5 do
```

```
a:=(j-1)*hh;
b:=j*hh;
A:=rkVect2Step(f1,f2,a,b,w1,w2):
w1:=A[1];
w2:=A[2];
err:= abs(w1-g1(b))+abs(w2-g2(b)):
print(j*0.2,w1,w2,err);
od:
```

$w1 := 1$

$w2 := 1$

$hh := 0.2$

0.2, 2.120365827, 1.506991852, 0.009238144

0.4, 4.441227756, 3.242240208, 0.047636929

0.6, 9.739133288, 8.163416998, 0.186103887

0.8, 22.67655978, 21.34352779, 0.65142860

1.0, 55.66118088, 56.03050298, 2.15116088 **(36)**

Let's halve the step size and see if the error goes down by a factor near 16

```
> w1:=1; w2:=1;hh:=0.1;
for j from 1 to 10 do
a:=(j-1)*hh;
b:=j*hh;
A:=rkVect2Step(f1,f2,a,b,w1,w2):
w1:=A[1];
w2:=A[2];
err:= abs(w1-g1(b))+abs(w2-g2(b)):
print(j*0.2,w1,w2,err);
od:
```

$w1 := 1$

$w2 := 1$

$hh := 0.1$

0.2, 1.469229954, 1.164879865, 0.000266953

0.4, 2.124579344, 1.511161418, 0.000855061

0.6, 3.068042652, 2.150738426, 0.002060535

0.8, 4.462901900, 3.263777000, 0.004425993

1.0, 6.572461606, 5.140295661, 0.008934544

1.2, 9.823671697, 8.247630720, 0.017351756

1.4, 14.91172652, 13.34019237, 0.03282548

1.6, 22.97214911, 21.63843306, 0.06093400

1.8, 35.86404564, 35.12124819, 0.11151216

2.0, 56.63652553, 57.00449690, 0.20182231 **(37)**

Clearly we are not in the region where the error goes down as expected from theory, so let's halve againt

```
> w1:=1; w2:=1;hh:=0.05;
for j from 1 to 10 do
a:=(j-1)*hh;
b:=j*hh;
A:=rkVect2Step(f1,f2,a,b,w1,w2):
w1:=A[1];
```

```
w2:=A[2];
err:= abs(w1-g1(b))+abs(w2-g2(b)):
print(j*0.2,w1,w2,err);
od:
```

$$w1 := 1$$

$$w2 := 1$$

$$hh := 0.05$$

0.2, 1.216098879, 1.064920348, $8.038\ 10^{-6}$
0.4, 1.469353842, 1.165002607, 0.000020323
0.6, 1.768603469, 1.309824590, 0.000038576
0.8, 2.124975724, 1.511554959, 0.000065140
1.0, 2.552516927, 1.785641786, 0.000103202
1.2, 3.068997032, 2.151687501, 0.000157080
1.4, 3.696941038, 2.634561591, 0.000232599
1.6, 4.464950509, 3.265816769, 0.000337615
1.8, 5.409397298, 4.085492715, 0.000482669
2.0, 6.576594841, 5.144415058, 0.000681912 **(38)**

The difference between being in the range where the error is like O(h^4) and where we are not!