AMTH250 Assignment 8
Solutions

November 8, 2011

Question 1

(a)

```
octave:> f1 = @(x) (x.^3)./(exp(x)-1);
octave:> intf1 = quad(f1, 0, Inf)
intf1 =  6.49393940226683
```

(b)

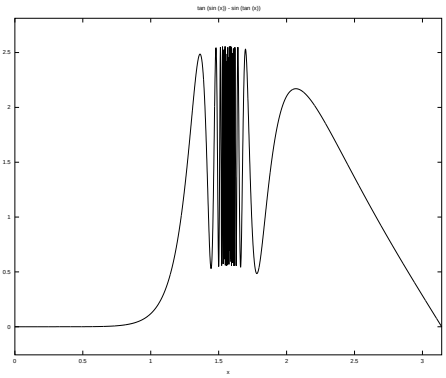
```
octave:> f2 = @(x) log(1+x).*log(1-x);
octave:> intf2 = quad(f2, -1, 1)
intf2 = -1.10155082809983
```

(c)

```
octave:> f3 = @(x) tan(sin(x)) - sin(tan(x));
octave:> intf3 = quad(f3, 0, pi)
intf3 =  2.66428197092458
```

The first two integrals are unproblematic.

The graph of the third integrand shows a singularity at $x = \pi/2$. More precisely, the second term, $\sin(\tan(x))$, is undefined at $x = \pi/2$.



The third integral may be difficult to evaluate accurately, but `quad` gives a reasonably small estimate of the error.

```
octave:> [int ier nfun err] = quad(f3, 0, pi)
int =  2.66428197092458
ier = 0
nfun =  399
err =  1.04183328630825e-12
```

We can get an accurate value for the third integral by noting that the function $\sin(\tan(x))$ is odd about $\pi/2$.

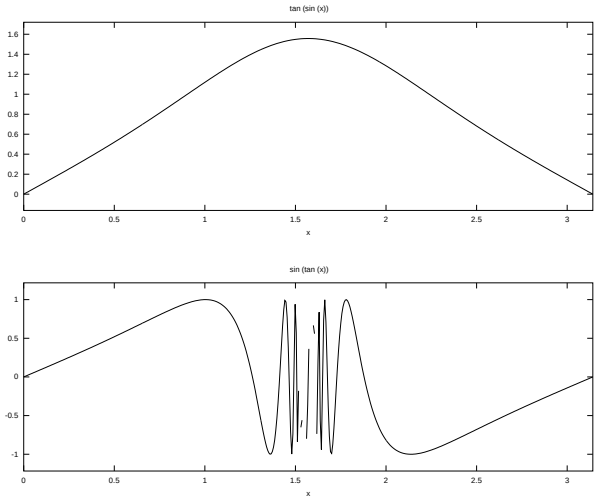


Figure 1: Graphs of $\tan(\sin(x))$ and $\sin(\tan(x))$

Therefore

$$\int_0^{\pi/2} \sin(\tan(x))\,dx = 0$$

and

$$\int_0^{\pi/2} (\tan(\sin(x) - \sin(\tan(x)))\,dx = \int_0^{\pi/2} \tan(\sin(x)\,dx$$

which can be evaluated accurately.

```
octave:> f4 = @(x) tan(sin(x));
octave:> [int ier nfun err] = quad(f4, 0, pi)
int =  2.66428197092504
ier = 0
nfun =  147
err =  2.95794718719031e-14
```

Confirming the result from `quad` for the third integral.

Question 2

```
fc = @(t) cos(pi*t.^2/2);
fs = @(t) sin(pi*t.^2/2);
x = -10:0.01:10;
C = zeros(1,length(x));
S = zeros(1,length(x));
for i = 1:length(x)
  C(i) = quad(fc, 0, x(i));
  S(i) = quad(fs, 0, x(i));
end
```

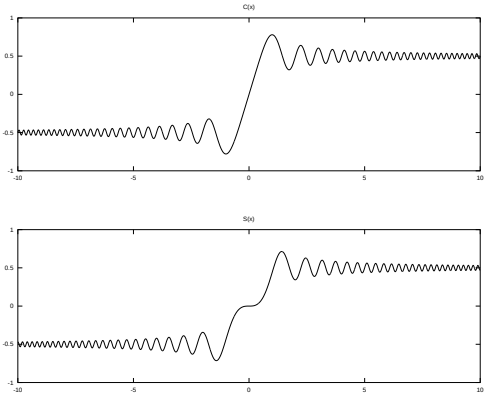


Figure 2: Graphs of Fresnel integrals $C(x)$ and $S(x)$

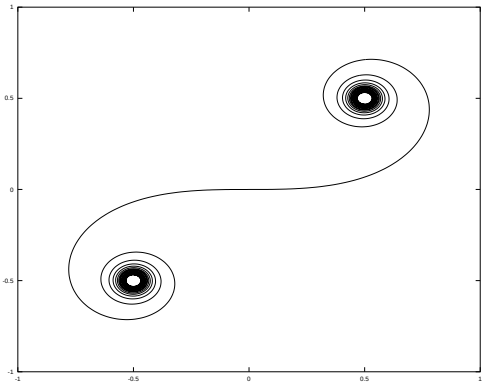


Figure 3: Cornu spiral

Question 3

(a) The ellipsoid is contained in the region

$$-1 \leq x \leq 1 \quad -2 \leq y \leq 2 \quad -4 \leq z \leq 4$$

By symmetry we can consider one of the 8 quadrants, e.g

$$0 \leq x \leq 1 \quad 0 \leq y \leq 2 \quad 0 \leq z \leq 4$$

of volume 8.

```
function v = monte3d(n)
  k = 0;
  for i = 1:n
    x = rand(1,1);
    y = 2*rand(1,1);
    z = 4*rand(1,1);
    if (16*x^2 + 4*y^2 + z^2 <= 16)
      k = k+ 1;
    end
  end
  v = 8*8*k/n;
endfunction

octave:> monte3d(1000000)
ans =  33.518
```

The exact volume is

$$\frac{32}{3}\pi = 33.5103$$

(b) We generate random points in the rectangle $0 \leq x \leq 1, -1 \leq y \leq 1$ and keep those which lie in Ω which has area $\pi/2$.

```
function ii = monte2d(n)
  k = 0;
  sumf = 0;
  while (k < n)
    x = rand(1,1);
    y = -1 + 2*rand(1,1);
    if (x^2 + y^2 <= 1)
      k = k + 1;
      sumf = sumf + exp(-sqrt(x^2+y^2));
    end
  end
  ii = (pi/2)*(sumf/n);
endfunction

octave:> monte2d(1000000)
ans =  0.83067
```

The exact answer is

$$\left(1 - \frac{2}{e}\right)\pi = 0.8301379$$

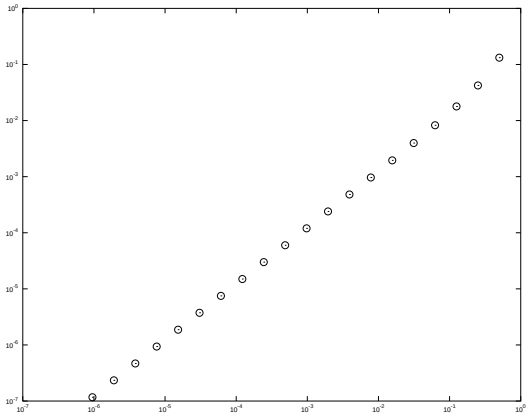
Question 4

(a) The following script computes the solution at $t = 1$ for step sizes $10^{-1}, \ldots, 10^{-8}$

```
f = @(t,y) -2*t*y;
N = 8;
h = zeros(1,N);
y1 = zeros(1,N);
for k = 1:N
  hk = 10^-k;
  h(k) = hk;
  [t y] =euler(f, 0, 1, hk, 10^k);
  y1(k) =y(end);
end
```

The log-log plot shows that the error is a power function of the step-size h.

```
octave:> err= abs(y1 - exp(-1));
octave:> loglog(h, err, 'o')
```



Fitting a power function using least squares

```
octave:> cs = polyfit(log(h),log(err),1)
cs =
   1.0260  -1.8402
octave:> exp(cs)
ans =
   2.78982   0.15879
```

Gives

$$\text{Error} \approx 0.159h^{1.03}$$

This agrees with theory which tells us that the error is proportional to the stepsize.

(b) The only way determine the range of stability is by "trial-and-error" experimentation. There is some theory which helps explain the difficulty of this problem:

Instability begins once the computed solution starts to oscillate between positive to negative values. These oscillations can start when the value of y is very small so they are not apparent at first, but once started they increase in amplitude.

Mathematically, Euler's method for this problem is unstable for *all* step-sizes h with the instability starting near $t = 1/h$. In practice, however, the solution may become zero due to underflow before the instability begins.

The smallest computer representable number is about 10^{-323} so the exact solution $y = e^{-t^2}$ will underflow at about $t \approx \sqrt{323 \ln 10} \approx 27.3$. Therefore at a step-size around $h = 1/t \approx 0.037$ we would expect underflow to occur before instability sets in.

Experiment shows Euler's method is stable for $h \leq 0.035$ and unstable for $h \geq 0.036$.

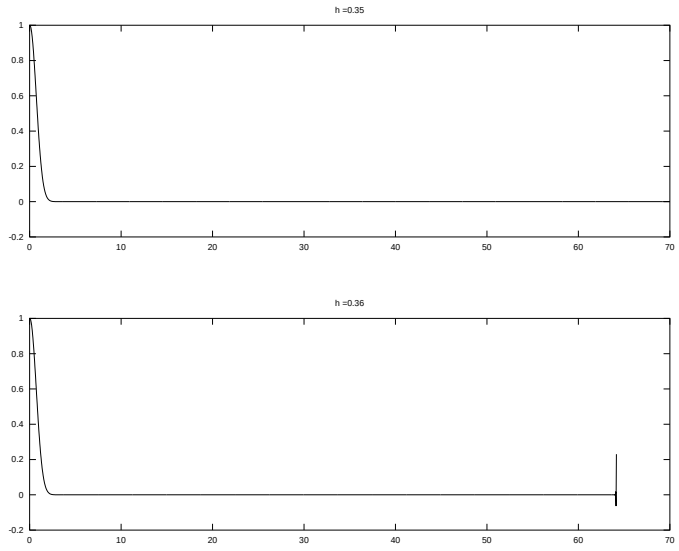


Figure 4: Solution by Euler's method for $h = 0.035$ and $h = 0.036$

n.b. there are some larger step sizes, in particular $h = 1/2, 1/4, 1/8, ...$, for which the computed solution hits zero, by numerical 'accident' as it were, which might also be considered stable.

Question 5

```
function dy = lorenz(y, t)
  s = 10;
  b = 8/3;
  r = 28;
  dy = zeros(3,1);
  dy(1) = s*(y(2) - y(1));
  dy(2) = r*y(1) - y(2) - y(1)*y(3);
  dy(3) = y(1)*y(2) - b*y(3);
endfunction

octave:> tt = 0:0.01:100;
octave:> yy = lsode(@lorenz, [0 1 0], tt);
```

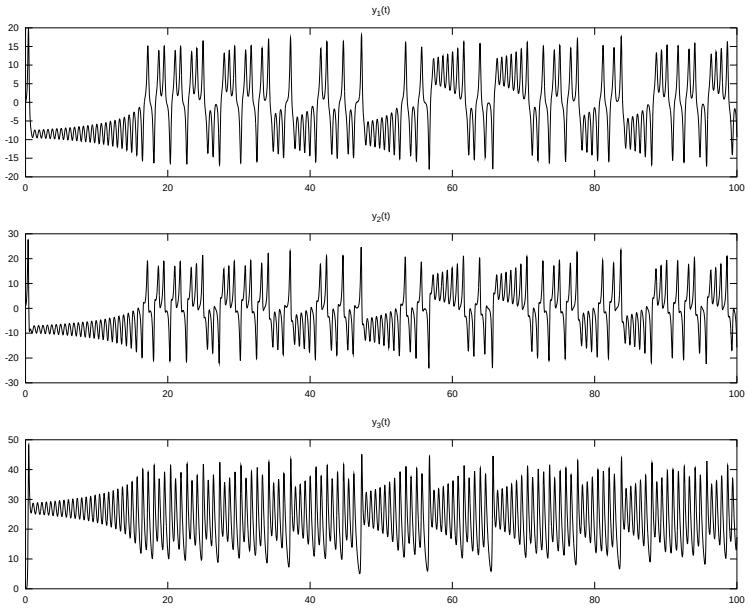


Figure 5: Components of the solution

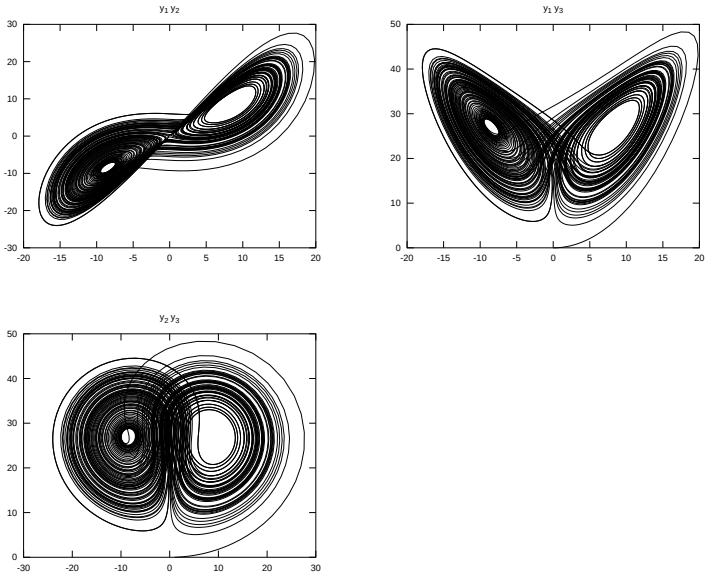


Figure 6: Phase plane plots

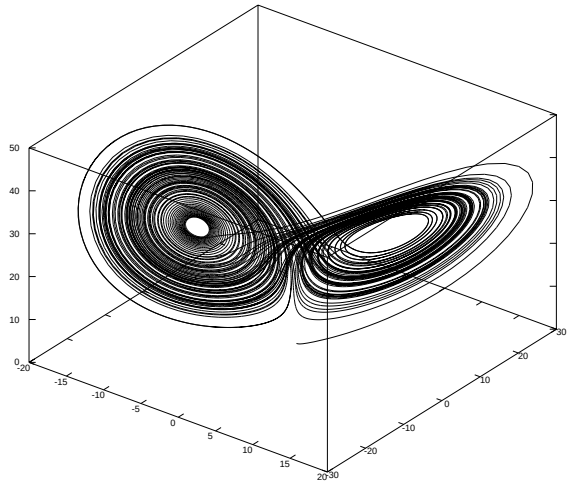


Figure 7: 3D phase plot

Now change the initial values slightly:

```
octave:> yy1 = lsode(@lorenz, [0 1 0] + 1e-8*randn(1,3), tt);
```

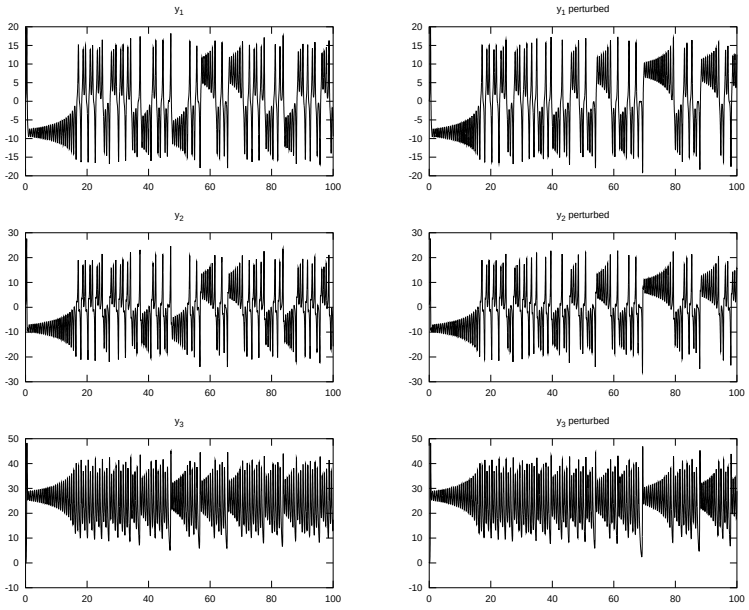


Figure 8: Components of the solution, original and perturbed initial values

With the change in initial values the "qualitative" features of the solution remain the same (also see the phase plots below) but the details of the solution change.

We see that up to about $t = 30$ the original and perturbed solution remain very close, but after that they are quite distinct. This sensitivity to small changes in initial values is characteristic of *chaotic systems*.

The final values of $y(t)$ are completely different

```
octave:> yy(end,:)
ans =
   -9.4133  -15.7383   17.3386

octave:> yy1(end,:)
ans =
    2.73995    0.98530   23.77475
```

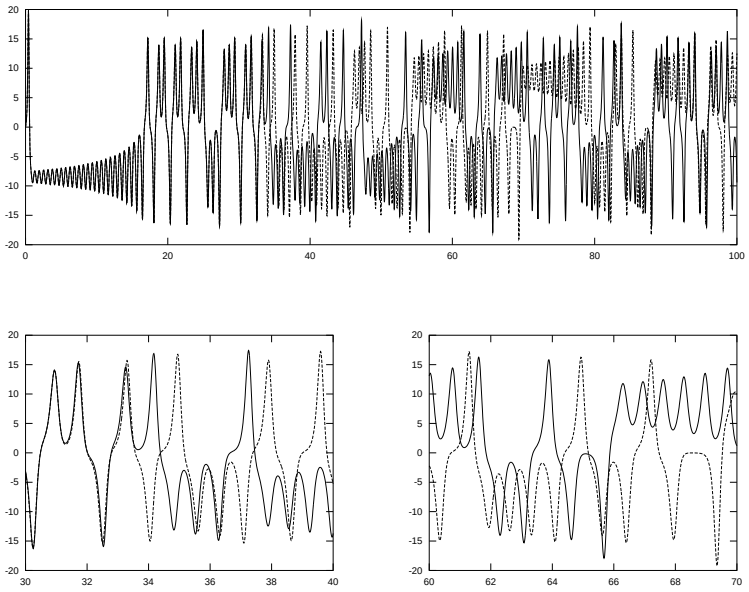


Figure 9: $y_1(t)$, original and perturbed initial values

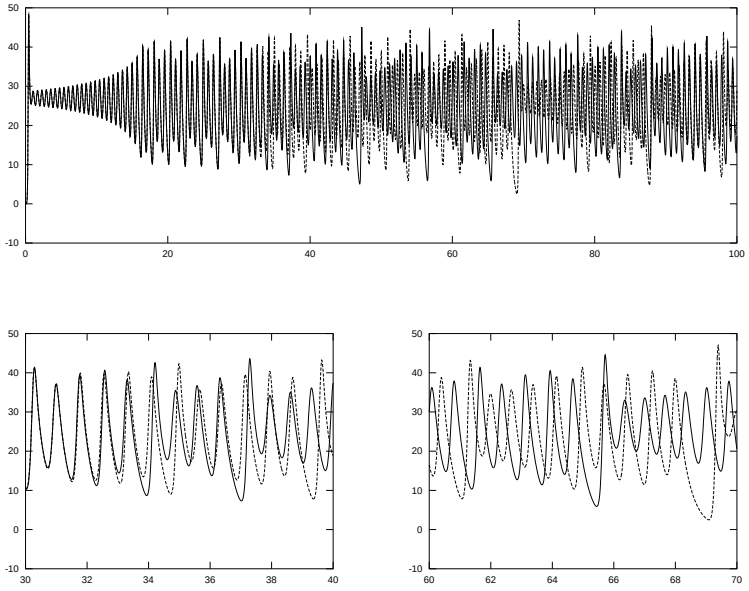


Figure 10: $y_3(t)$, original and perturbed initial values

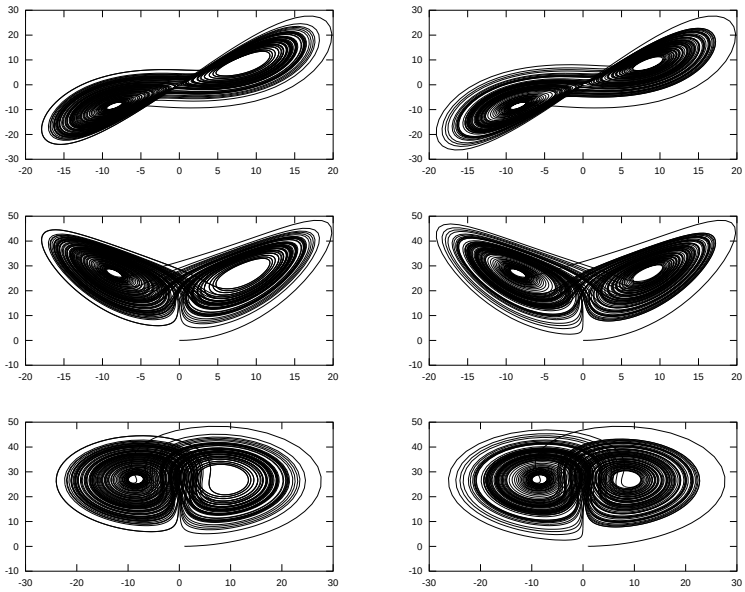


Figure 11: Phase plots, original and perturbed initial values

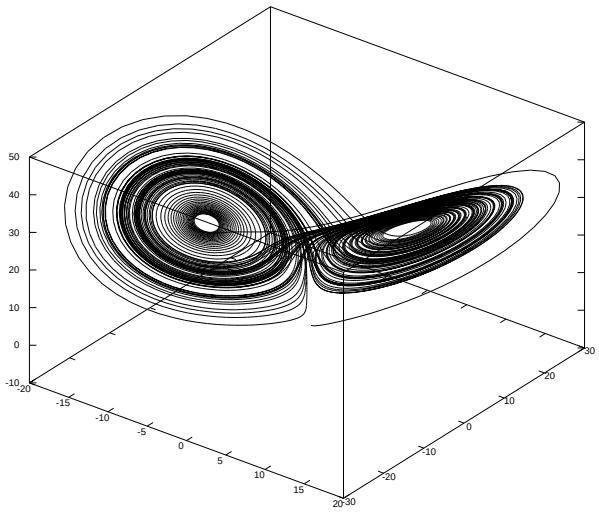


Figure 12: 3D phase plots, perturbed initial values