MATH 2300 Assignment 3

This assignment is due Tuesday March 29, 2016 at the beginning of class. **No unstapled assignments will be accepted.**

1. Fit a low degree polynomial to the points

```
> p:=[[0,0],[1,2],[2,4],[3,6],[4,8],[5,10.001],[6,12],[7,14]];
```

$$p := [[0, 0], [1, 2], [2, 4], [3, 6], [4, 8], [5, 10.001], [6, 12], [7, 14]]$$

Solution: We use the atMostQuartic procedure in Question 2 (you can alternatively use divided differences to decide on the polynomial to use):

```
> f1:=atMostQuartic(p,1);
  f2:=atMostQuartic(p,2);
  f3:=atMostQuartic(p,3);
  f4:=atMostQuartic(p,4);
```

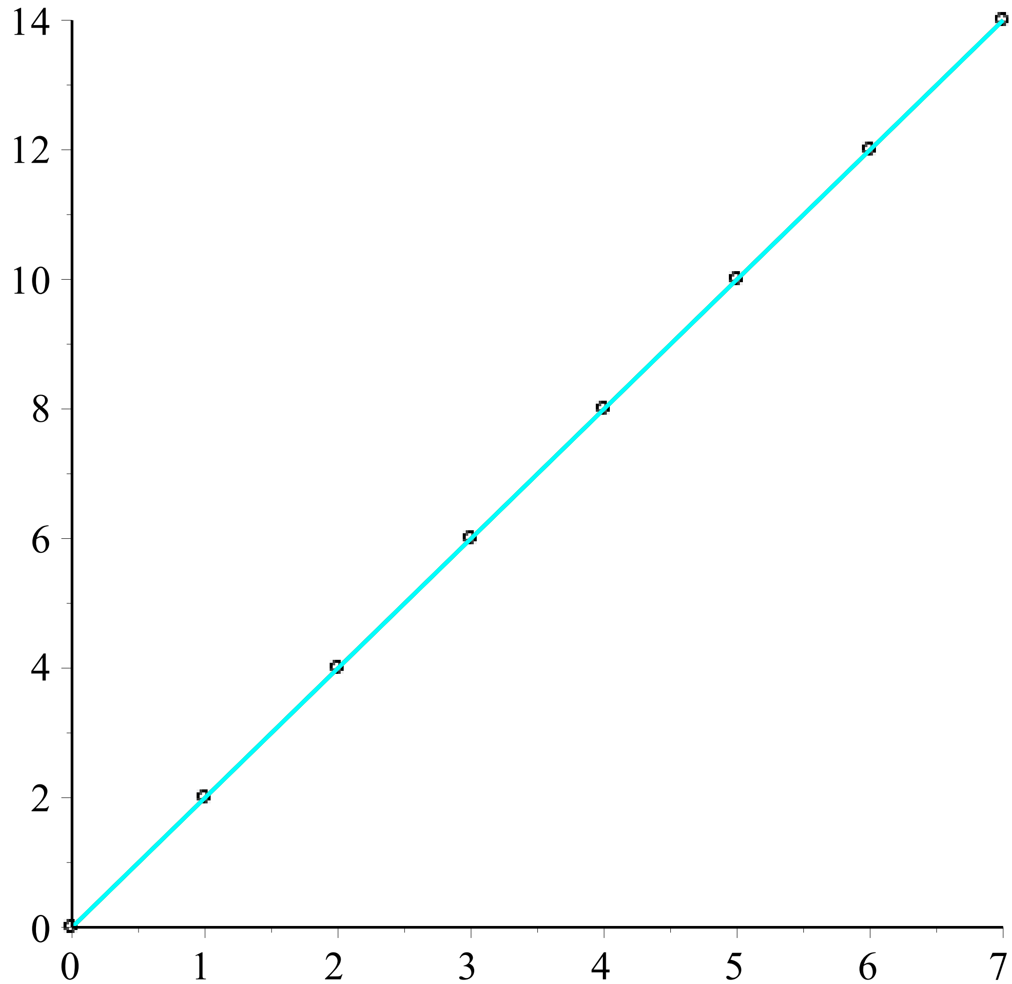
$$f1 := x \rightarrow 2.000035714\, x$$

$$f2 := x \rightarrow -0.00001785714286\, x^2 + 2.000160714\, x - 0.0001250000000$$

$$f3 := x \rightarrow -0.00001767676768\, x^3 + 0.0001677489177\, x^2 + 1.999674603\, x + 0.00006060606061$$

$$f4 := x \rightarrow -0.000002840909091\, x^4 + 0.00002209595960\, x^3 - 0.000004734848485\, x^2 + 1.999907558\, x + 0.00002651515152$$

```
> with(plots):
  plot0:=plot(p,style=point,symbol=circle,colour=black):
  plot1:=plot([f1,f2,f3,f4],0..7,color=[red,blue,green,cyan]):
  display([plot0,plot1]);
```



We see that all the functions fit the points well. We choose the one of least degree, namely the linear one.

Alternatively, we can use the divided differences:

```
> divDiff:=proc(p,n)
  local lst,newlst,j,i:
  lst:=[seq(p[i][2],i=1..nops(p))]:
  for i from 1 to n do
      newlst:=NULL:
      for j from 1 to nops([lst][i])-1 do
        newlst:=newlst,([lst][i][j+1]-[lst][i][j])/(p[j+i][1]-p[j][1])
  :
      od:
      lst:=lst,[newlst]:
  od:
  [seq(lst[i],i=2..nops([lst]))];
  end;
```

$divDiff := \mathbf{proc}(p, n)$ **(1)**

local $lst, newlst, j, i;$

$lst := [seq(p[i][2], i = 1 ..nops(p))];$

for i **to** n **do**

$newlst := NULL;$

```
        for j to nops([lst][i]) - 1 do
            newlst := newlst, ([lst][i][j + 1] - [lst][i][j]) / (p[j + i][1] - p[j][1])
        end do;
        lst := lst, [newlst]
    end do;
    [seq(lst[i], i = 2 ..nops([lst]))]
end proc
> divDiff(p,2);
```

$$[[2, 2, 2, 2, 2.001, 1.999, 2], [0, 0, 0, 0.0005000000000, -0.001000000000, 0.0005000000000]] \quad (2)$$

Here we see already that the decond divided differences are almost all 0, so we conclude that a linear function fits the data well.

In any event, we find that the function

```
> f1(x);
```

$$2.000035714\, x \quad (3)$$

is a low order polynomial that fits the data well.

2. Write a Maple procedure **atMostQuarticFit** that takes as input a list of points *p* and a positive integer i between 1 and 4 inclusive, and returns a best fitting polynomial of degree *i* to *p*. (You willl have tofind a way to find the best fitting quartic.

```
atMostQuartic:=proc(p,n)
local m,i,sumx8,sumx7,sumx6,sumx5,sumx4,sumx3,sumx2,sumx1,sumx0,sumyx4,
sumyx3,sumyx2,sumyx1,sumyx0:
  m:=nops(p):
  sumx8:=sum((p[i][1])^8,i=1..m):
  sumx7:=sum((p[i][1])^7,i=1..m):
  sumx6:=sum((p[i][1])^6,i=1..m):
  sumx5:=sum((p[i][1])^5,i=1..m):
  sumx4:=sum((p[i][1])^4,i=1..m):
  sumx3:=sum((p[i][1])^3,i=1..m):
  sumx2:=sum((p[i][1])^2,i=1..m):
  sumx1:=sum((p[i][1]),i=1..m):
  sumx0:=m:
  sumyx4:=sum(p[i][2]*(p[i][1])^4,i=1..m):
  sumyx3:=sum(p[i][2]*(p[i][1])^3,i=1..m):
  sumyx2:=sum(p[i][2]*(p[i][1])^2,i=1..m):
  sumyx1:=sum(p[i][2]*(p[i][1]),i=1..m):
  sumyx0:=sum(p[i][2],i=1..m):
  if n = 4 then
    solve({sumx8*A+sumx7*B+sumx6*C+sumx5*E+sumx4*F=sumyx4,sumx7*A+sumx6*
B+sumx5*C+sumx4*E+sumx3*F=sumyx3,sumx6*A+sumx5*B+sumx4*C+sumx3*E+sumx2*
F=sumyx2,sumx5*A+sumx4*B+sumx3*C+sumx2*E+sumx1*F=sumyx1,sumx4*A+sumx3*B+
sumx2*C+sumx1*E+sumx0*F=sumyx0},{A,B,C,E,F}):
    return(subs(%,x->A*x^4+B*x^3+C*x^2+E*x+F));
  fi:
  if n = 3 then
    solve({sumx6*A+sumx5*B+sumx4*C+sumx3*E=sumyx3,sumx5*A+sumx4*B+sumx3*
C+sumx2*E=sumyx2,sumx4*A+sumx3*B+sumx2*C+sumx1*E=sumyx1,sumx3*A+sumx2*B+
sumx1*C+sumx0*E=sumyx0},{A,B,C,E}):
    return(subs(%,x->A*x^3+B*x^2+C*x+E));
  fi:
  if n = 2 then
```

```
    solve({sumx4*A+sumx3*B+sumx2*C=sumyx2,sumx3*A+sumx2*B+sumx1*C=
sumyx1,sumx2*A+sumx1*B+sumx0*C=sumyx0},{A,B,C}):
    return(subs(%,x->A*x^2+B*x+C));
  fi:
  if n = 1 then
    solve({sumx2*A+sumx1*B=sumyx1,sumx1*A+sumx0*B=sumyx0},{A,B}):
    return(subs(%,x->A*x+B));
  fi:
end;
```

atMostQuartic := **proc**(*p*, *n*) **(4)**

```
  local m, i, sumx8, sumx7, sumx6, sumx5, sumx4, sumx3, sumx2, sumx1, sumx0, sumyx4,
  sumyx3, sumyx2, sumyx1, sumyx0;
  m := nops(p);
  sumx8 := sum(p[i][1]^8, i = 1 ..m);
  sumx7 := sum(p[i][1]^7, i = 1 ..m);
  sumx6 := sum(p[i][1]^6, i = 1 ..m);
  sumx5 := sum(p[i][1]^5, i = 1 ..m);
  sumx4 := sum(p[i][1]^4, i = 1 ..m);
  sumx3 := sum(p[i][1]^3, i = 1 ..m);
  sumx2 := sum(p[i][1]^2, i = 1 ..m);
  sumx1 := sum(p[i][1], i = 1 ..m);
  sumx0 := m;
  sumyx4 := sum(p[i][2]*p[i][1]^4, i = 1 ..m);
  sumyx3 := sum(p[i][2]*p[i][1]^3, i = 1 ..m);
  sumyx2 := sum(p[i][2]*p[i][1]^2, i = 1 ..m);
  sumyx1 := sum(p[i][2]*p[i][1], i = 1 ..m);
  sumyx0 := sum(p[i][2], i = 1 ..m);
  if n = 4 then
    solve({sumx8*A + sumx7*B + sumx6*C + sumx5*E + sumx4*F = sumyx4, sumx7
    *A + sumx6*B + sumx5*C + sumx4*E + sumx3*F = sumyx3, sumx6*A + sumx5
    *B + sumx4*C + sumx3*E + sumx2*F = sumyx2, sumx5*A + sumx4*B + sumx3
    *C + sumx2*E + sumx1*F = sumyx1, sumx4*A + sumx3*B + sumx2*C + sumx1
    *E + sumx0*F = sumyx0}, {A, B, C, E, F});
    return subs(`%`, x→A*x^4 + B*x^3 + C*x^2 + E*x + F)
  end if;
  if n = 3 then
    solve({sumx6*A + sumx5*B + sumx4*C + sumx3*E = sumyx3, sumx5*A + sumx4
    *B + sumx3*C + sumx2*E = sumyx2, sumx4*A + sumx3*B + sumx2*C + sumx1
    *E = sumyx1, sumx3*A + sumx2*B + sumx1*C + sumx0*E = sumyx0}, {A, B, C,
    E});
    return subs(`%`, x→A*x^3 + B*x^2 + C*x + E)
  end if;
  if n = 2 then
    solve({sumx4*A + sumx3*B + sumx2*C = sumyx2, sumx3*A + sumx2*B + sumx1
    *C = sumyx1, sumx2*A + sumx1*B + sumx0*C = sumyx0}, {A, B, C});
    return subs(`%`, x→A*x^2 + B*x + C)
  end if;
  if n = 1 then
    solve({sumx2*A + sumx1*B = sumyx1, sumx1*A + sumx0*B = sumyx0}, {A, B});
    return subs(`%`, x→A*x + B)
```

```
    end if
end proc
```

We test out the procedure on the points from question 1 (again!):

```
> f1:=atMostQuartic(p,1);
  f2:=atMostQuartic(p,2);
  f3:=atMostQuartic(p,3);
  f4:=atMostQuartic(p,4);
```

$$f1 := x \to 2.000035714\, x$$

$$f2 := x \to -0.00001785714286\, x^2 + 2.000160714\, x - 0.0001250000000$$

$$f3 := x \to -0.00001767676768\, x^3 + 0.0001677489177\, x^2 + 1.999674603\, x + 0.00006060606061$$

$$f4 := x \to -0.000002840909091\, x^4 + 0.00002209595960\, x^3 - 0.000004734848485\, x^2 + 1.999907558\, x + 0.00002651515152 \qquad (5)$$

3. Consider the points

```
> p:=[[20,20],[25,28],[30,40.5],[35,52.5],[40,72],[45,92.5],[50,118],
  [55,148.5],[60,182],[65,220.5],[70,266],[75,318],[80,376]];
```

$$p := [[20, 20], [25, 28], [30, 40.5], [35, 52.5], [40, 72], [45, 92.5], [50, 118], [55, 148.5], [60, 182], [65, 220.5], [70, 266], [75, 318], [80, 376]]$$

(a) Best fit a straight line to p.

```
> f1:=atMostQuartic(p,1);
```

$$f1 := x \to 5.789010989\, x - 140.6428571 \qquad (6)$$

(b) Best fit a quadratic to p.

```
> f2:=atMostQuartic(p,2);
```

$$f2 := x \to 0.08873126873\, x^2 - 3.084115884\, x + 50.12937063 \qquad (7)$$

(c) Best fit a cubic to p.

```
> f3:=atMostQuartic(p,3);
```

$$f3 := x \to 0.0005244755245\, x^3 + 0.01005994006\, x^2 + 0.5216533467\, x + 0.9597902098 \qquad (8)$$

(d) Best fit a quartic to p.

```
> f4:=atMostQuartic(p,4);
```

$$f4 := x \to 0.000004552310435\, x^4 - 0.0003859865625\, x^3 + 0.07432880845\, x^2 - 1.352923057\, x + 19.78196803 \qquad (9)$$

(e) Which model best fits the trend of the data p? Don't necessarily choose the highest degree polynomial. Explain your answer (you are graded on your reasoning).

Here are the divided differences:

```
> divDiff(p,4);
```

(10)

$$\left[\left[\frac{8}{5}, 2.500000000, 2.400000000, 3.900000000, 4.100000000, 5.100000000, 6.100000000, 6.700000000, 7.700000000, 9.100000000, \frac{52}{5}, \frac{58}{5}\right], \left[0.09000000000, -0.01000000000, 0.1500000000, 0.02000000000, 0.1000000000, 0.1000000000, 0.06000000000, 0.1000000000, 0.1400000000, 0.1300000000, \frac{3}{25}\right], [-0.006666666667, 0.01066666667, -0.008666666667, 0.005333333333, 0., -0.002666666667, 0.002666666667, 0.002666666667, -0.0006666666667, -0.0006666666667], [0.0008666666670, -0.0009666666670, 0.0007000000000, -0.0002666666666, -0.0001333333334, 0.0002666666667, 0., -0.0001666666667, 0.]\right] \quad \textbf{(10)}$$

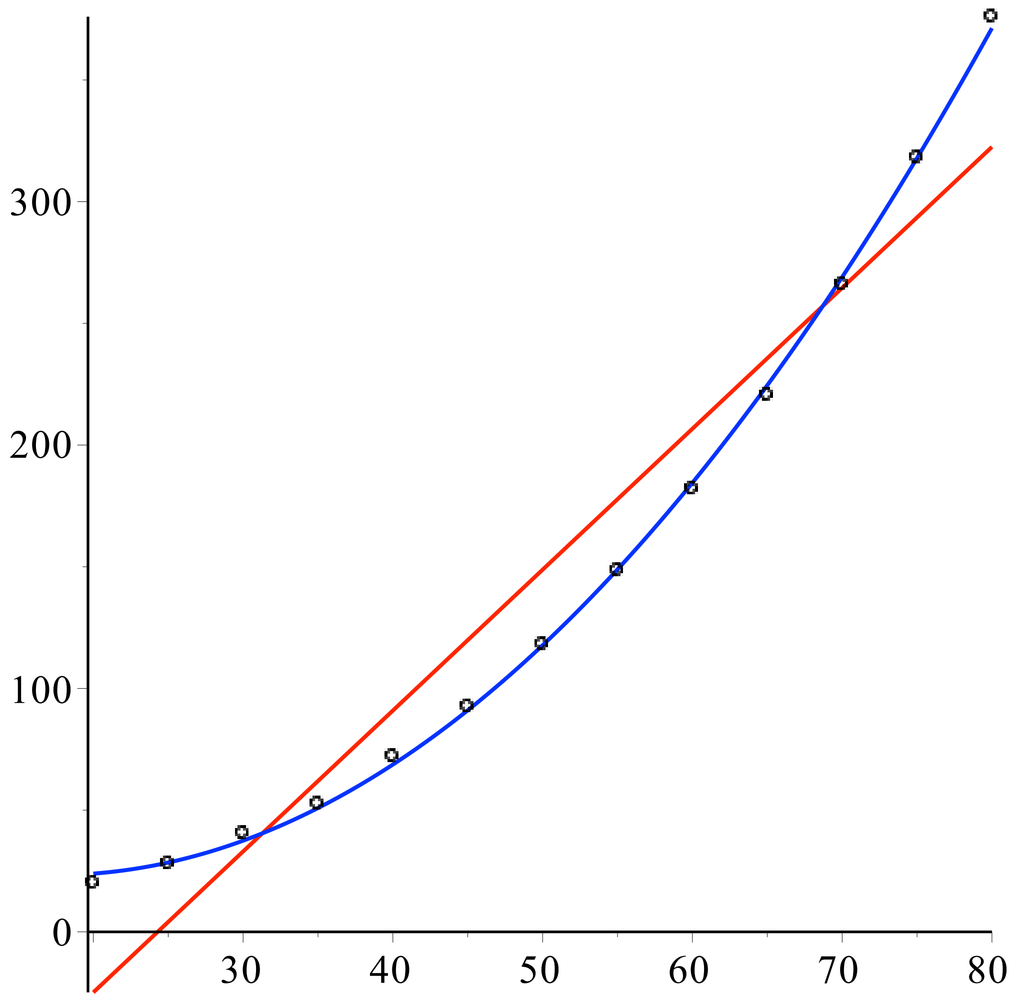
We see that the third divided difference is clsoe to 0, so I conclude that fitting a quadratic is appropriate. Thus the model that best fits the trend is

```
> f3(x);
```

$$0.0005244755245\, x^3 + 0.01005994006\, x^2 + 0.5216533467\, x + 0.9597902098$$

Here is a plot of the best fitting line and quadratic, along with the points:

```
> with(plots):
  p1:=plot(f1,20..80,color=red):
  p2:=plot(f2,20..80,color=blue):
  p3:=plot(p,20..80,style=point,symbol=circle,color=black):
  display([p1,p2,p3]);
```



4. Use a divided difference table to decide whether a low order polynomial fit the following points:

```
> p := [[0, -0.02], [1, 11.879], [2, 41.984], [3, 114.871], [4,
  255.196], [5, 487.455], [6, 836.416], [7, 1326.431], [8, 1982.444],
  [9, 2828.599], [10, 3890.080]];
```

$p := [[0, -0.02], [1, 11.879], [2, 41.984], [3, 114.871], [4, 255.196], [5, 487.455], [6, 836.416], [7, 1326.431], [8, 1982.444], [9, 2828.599], [10, 3890.080]]$

You can do a table by hand. Here I used my Maple procedure to find the columns:

```
> divDiff(p,4);
```

$[[11.899, 30.105, 72.887, 140.325, 232.259, 348.961, 490.015, 656.013, 846.155, 1061.481], [9.103000000, 21.39100000, 33.71900000, 45.96700000, 58.35100000, 70.52700000, 82.99900000, 95.07100000, 107.6630000], [4.096000000, 4.109333333, 4.082666667, 4.128000000, 4.058666667, 4.157333333, 4.024000000, 4.197333333], [0.003333333250, -0.006666666500, 0.01133333325, -0.01733333325, 0.02466666650, -0.03333333325, 0.04333333325]]$

We see that the fourth divided difference is close to 0, so we conclude that a cubic polynomial fits the point.

5. Write a Monte Carlo Maple procedure to find the area below a curve f, above a curve g, between x = a and x = b. Check your procedure on $f(x) = x^3$, $g(x) = -x^2 - 1$, between x = 0 and x = 3.

We also input a minimum value m and a maximum vaue M such that f(x) <= M and g(x) >= m.

```
> randNum:= proc(a, b)
  local t;
  t := evalf((1/1000000000000)*rand());
  evalf(a+(b-a)*t);
  end;
```

$randNum := \textbf{proc}(a, b)$
  local t; $t := evalf(1/1000000000000 * rand(\,))$; $evalf(a + (b - a) * t)$
end proc **(11)**

```
> areaMCTopBottom:=proc(f,g,a,b,M,m,n)
  local ccount,x,y,i;
  ccount:=0;
  for i to n do
    x:=randNum(a,b);
    y:=randNum(m,M);
    if y<=f(x) and y>=g(x) then
      ccount:=ccount+1;
    fi:
  od:
  evalf(ccount*(M-m)*(b-a)/n);
  end;
```

$areaMCTopBottom := \textbf{proc}(f, g, a, b, M, m, n)$ **(12)**
  local $ccount, x, y, i$;
  $ccount := 0$;
  for i **to** n **do**
    $x := randNum(a, b)$;
    $y := randNum(m, M)$;
    if $y <= f(x)$ **and** $g(x) <= y$ **then** $ccount := ccount + 1$ **end if**
  end do;
  $evalf(ccount * (M - m) * (b - a)/n)$
end proc

```
> f:=x->x^3;
  g:=x->-x^2-1;
  M:=27;
  m:=-10;
  areaMCTopBottom(f,g,0,3,27,-10,1000);
```

$$f := x \rightarrow x^3$$
$$g := x \rightarrow -x^2 - 1$$
$$M := 27$$
$$m := -10$$
$$32.41200000 \quad \textbf{(13)}$$

As f is clearly always above g on [0,3], the actual value of the area is

```
> int(f(x)-g(x),x=0..3);
  evalf(%);
```

$$\frac{129}{4}$$

(14)

$$32.25000000 \tag{14}$$

6. Write a Monte Carlo Maple procedure to estimate the volume trapped between the paraboloids $z = 8 - x^2 - y^2$ and $z = x^2 + 3\,y^2$. Explain any mathematics that you need to use (e.g. solving equations) for the simulation. Also, run your simulation and compare its output values to the actual volume.

```
> volumeMCTopBottom:=proc(top,bottom,xa,xb,ya,yb,M,m,n)
  local ccount,x,y,z,i;
  ccount:=0;
  for i to n do
    x:=randNum(xa,xb);
    y:=randNum(ya,yb);
    z:=randNum(m,M);
    if z<=top(x,y) and z>=bottom(x,y) then
      ccount:=ccount+1;
    fi:
  od:
  evalf(ccount*(M-m)*(xb-xa)*(yb-ya)/n);
  end;
```

volumeMCTopBottom := **proc**(*top*, *bottom*, *xa*, *xb*, *ya*, *yb*, *M*, *m*, *n*) **(15)**
    local *ccount*, *x*, *y*, *z*, *i*;
    ccount := 0;
    for *i* **to** *n* **do**
        x := *randNum*(*xa*, *xb*);
        y := *randNum*(*ya*, *yb*);
        z := *randNum*(*m*, *M*);
        if *z* <= *top*(*x*, *y*) **and** *bottom*(*x*, *y*) <= *z* **then** *ccount* := *ccount* + 1 **end if**
    end do;
    evalf(*ccount* * (*M* − *m*) * (*xb* − *xa*) * (*yb* − *ya*) / *n*)
end proc

```
> f:=(x,y)->8-x^2-y^2;
  g:=(x,y)->x^2+3*y^2;
```

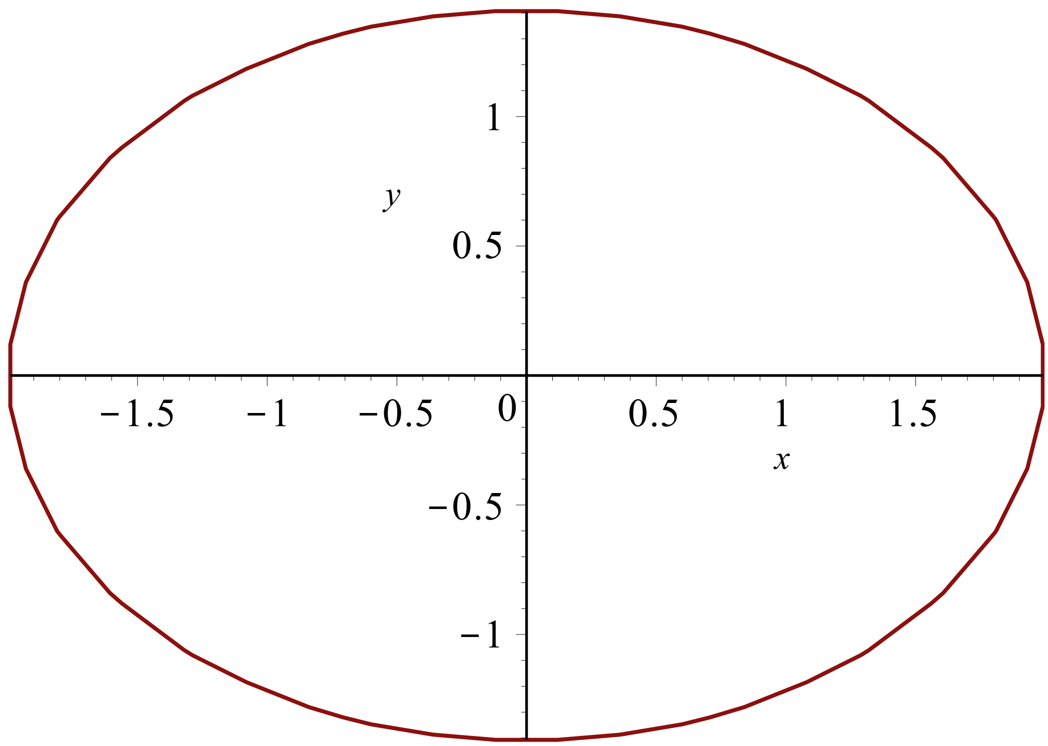
$$f := (x, y) \rightarrow 8 - x^2 - y^2$$
$$g := (x, y) \rightarrow x^2 + 3\,y^2 \tag{16}$$

We see that f is always at most 8, and g is always at least 0, so we can set M = 8, m= 0. The intersection is an ellpise that traps the volume, and a little bit of math shows that the ellipse is

```
> h:=x^2+2*y^2=4;
```

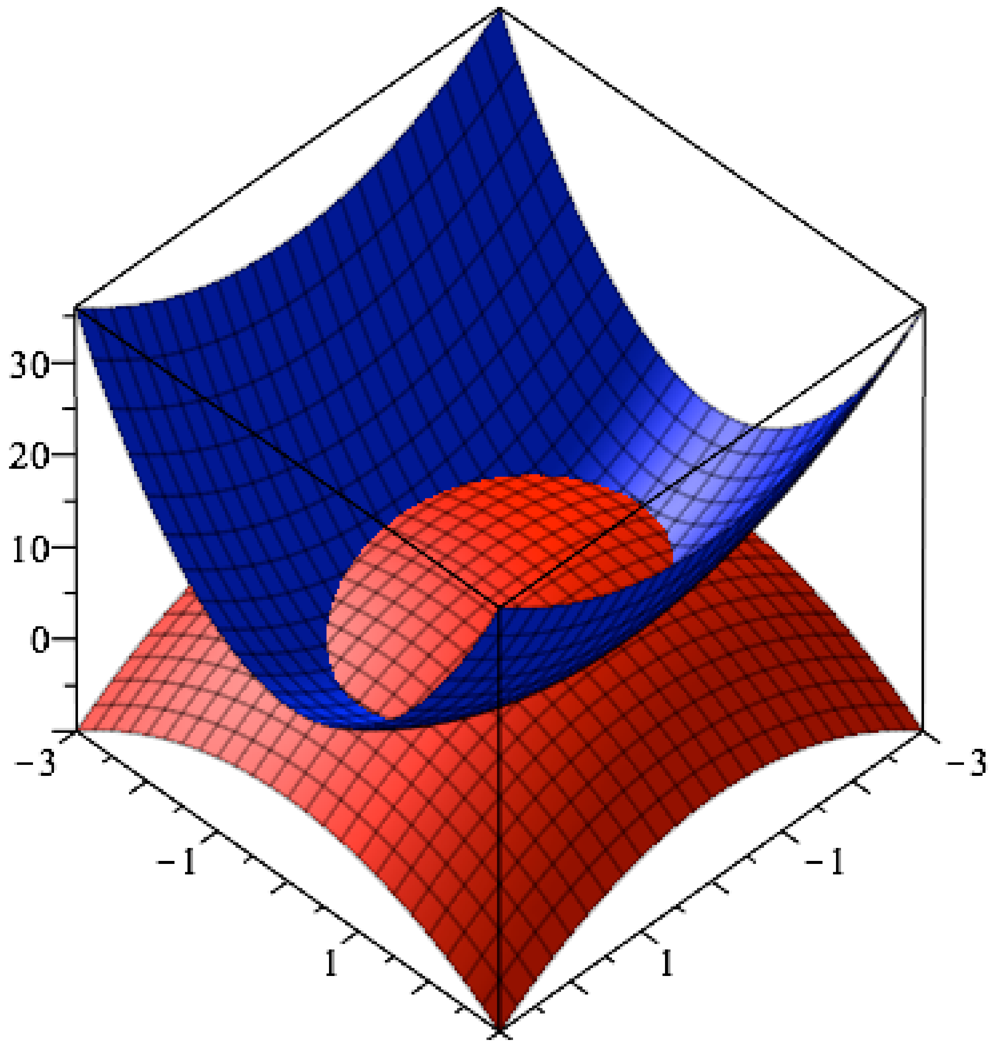
$$h := x^2 + 2\,y^2 = 4 \tag{17}$$

```
> with(plots):
  implicitplot(h,x=-3..3,y=-3..3,scaling=CONSTRAINED);
```



The range for x is -2 to 2, and y from -sqrt(2) to sqrt(2). A 3D plot gives us a better picture:

```
> plot3d([f,g],-3..3,-3..3,colour=[red,blue]);
```



We see that f is the top and g is the bottom on the volume trapped between the two curves.Using the routine we wrote, we approximate the volume:

```
> volumeMCTopBottom(f,g,-2,2,-sqrt(2),sqrt(2),8,0,100000);
```

$$35.41552798 \tag{18}$$

The actual volume is found by integrating the top function, f, mnus the bottom function, g, over the intersection region, where x goes between -2 and 2 and y goes between -sqrt(4-x^2)/sqrt(2) and sqrt(4-x^2)/sqrt(2), so we find that the true volume is

```
> int(f(x,y)-g(x,y),y=-sqrt(4-x^2)/sqrt(2)..sqrt(4-x^2)/sqrt(2),x=-2.
  .2);
```

$$8\sqrt{2}\,\pi \tag{19}$$

```
> evalf(%);
```

$$35.54306350 \tag{20}$$

7.

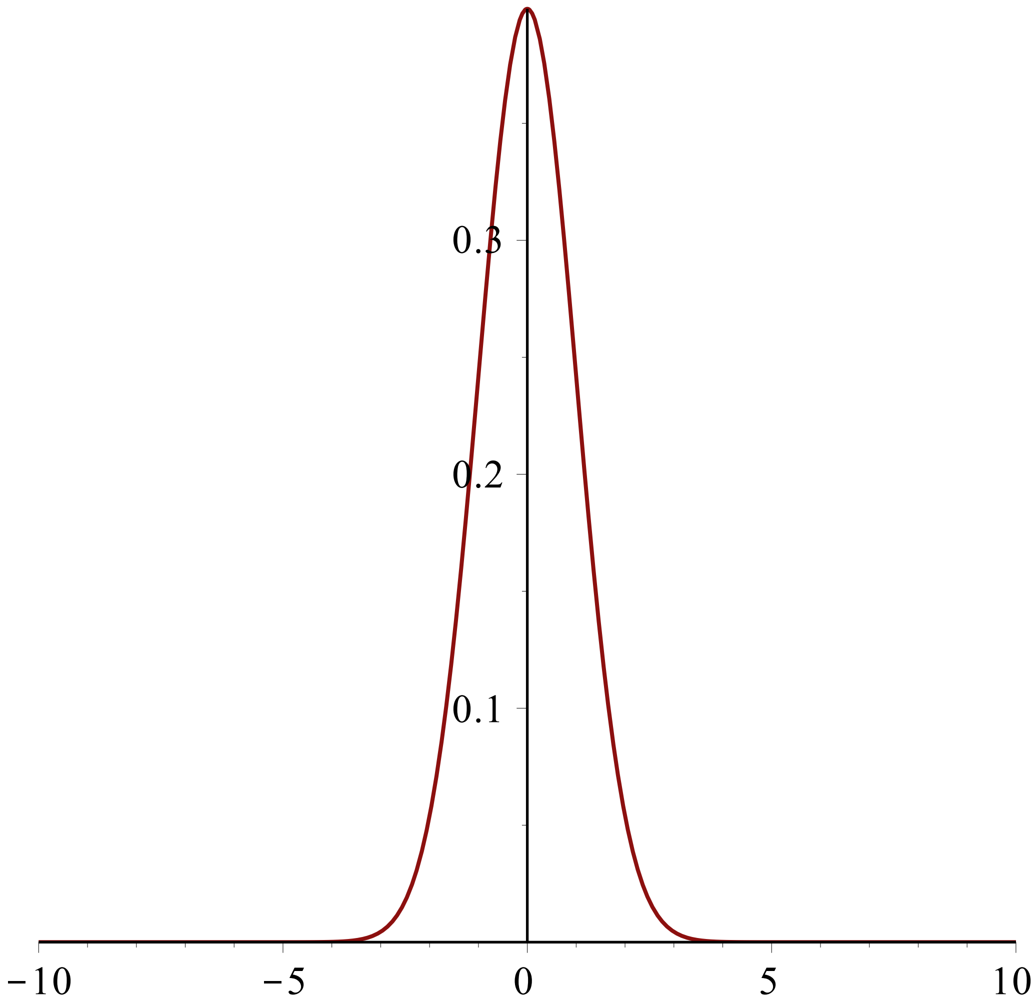
The distribution of some values are normally distributed with mean mu and standard deviation sigma if, for any number x, the probability that a value is less than x is $\int_{-infinity}^{x} \frac{e^{-\frac{(t-\text{mu})^2}{2\,\text{sigma}^2}}}{\text{sigma}\sqrt{2\,\text{Pi}}}\,dt$.

(Here mu is any real number and sigma is any positive number).

(a) Plot the integrand with mu = 0 and sigma = 1 for x = -10 to 10. **(21)**

```
> f:=t->exp(-(t^2)/2)/sqrt(2*Pi);
  plot(f,-10..10);
```

$$f := t \rightarrow \frac{e^{-\frac{1}{2}t^2}}{\sqrt{2\,\pi}}$$



(b) Why is the integrand always positive?

The integrand is always positive as the exponential function is always positive, and the denominator is sigma times sqrt(2*Pi), which are clearly both positive (as sigma is positive).

(c) Set mu = 0 and sigma = 1. Use a Monte Carlo simulation to complete the following table: **(22)**

Spreadsheet(1)

	A	B	C	D	E
1	*x*	*probability*			
2	-3	0.0012			
3	-2.5	0.0065			
4	-2	0.0238			
5	-1.5	0.0679			
6	-1	0.1610			
7	-0.5	0.3208			
8	0	0.4900			
9	0.5	0.7062			
10	1	0.8172			
11	1.5	0.9506			
12	2	0.9764			
13	2.5	0.9702			
14	3	0.9984			

We can start at the left with say x = -5, as the function is so small to the left of that value. We sse that the integrand is always positive (by part (b)) and the largest it can be is when t = 0, when it is

```
> evalf(f(0));
```

$$0.3989422802 \qquad (23)$$

so we can use M = 0.4. We run our program and fill in the table above with approximations:

```
> areaMC:=proc(f,a,b,M,n)
  local ccount,x,y,i;
  ccount:=0;
  for i to n do
    x:=randNum(a,b);
```

```
    y:=randNum(0,M);
    if y<=evalf(f(x)) then
      ccount:=ccount+1;
    fi:
  od:
  evalf(ccount*M*(b-a)/n);
  end;

  areaMC(f,-5,-3,0.4,10000);
  areaMC(f,-5,-2.5,0.4,10000);
  areaMC(f,-5,-2,0.4,10000);
  areaMC(f,-5,-1.5,0.4,10000);
  areaMC(f,-5,-1,0.4,10000);
  areaMC(f,-5,-0.5,0.4,10000);
  areaMC(f,-5,0,0.4,10000);
  areaMC(f,-5,0.5,0.4,10000);
  areaMC(f,-5,1,0.4,10000);
  areaMC(f,-5,1.5,0.4,10000);
  areaMC(f,-5,2,0.4,10000);
  areaMC(f,-5,2.5,0.4,10000);
  areaMC(f,-5,3,0.4,10000);
```

areaMC := **proc**(*f*, *a*, *b*, *M*, *n*)
  local *ccount*, *x*, *y*, *i*;
  ccount := 0;
  for *i* **to** *n* **do**
    x := *randNum*(*a*, *b*);
    y := *randNum*(0, *M*);
    if *y* <= *evalf*(*f*(*x*)) **then** *ccount* := *ccount* + 1 **end if**
  end do;
  evalf(*ccount* * *M* * (*b* − *a*) / *n*)
end proc

$$0.001200000000$$
$$0.006500000000$$
$$0.02376000000$$
$$0.06790000000$$
$$0.1609600000$$
$$0.3207600000$$
$$0.4900000000$$
$$0.7062000000$$
$$0.8172000000$$
$$0.9505600000$$
$$0.9763600000$$
$$0.9702000000$$
$$0.9984000000 \quad \textbf{(24)}$$

8. Suppose we have the following matrix for a Markov chain with 5 states:

```
> with(linalg):
  m:=matrix([[1/5,1/5,1/5,1/5,1/5],[1/10,1/10,2/5,1/5,1/5],[2/5,1/5,
  1/5,1/10,1/10],[1/5,2/5,1/10,1/5,1/10],[1/5,1/5,1/5,3/10,1/10]]);
```

$$m := \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{10} & \frac{1}{10} & \frac{2}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{2}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{5} & \frac{2}{5} & \frac{1}{10} & \frac{1}{5} & \frac{1}{10} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{3}{10} & \frac{1}{10} \end{bmatrix} \tag{25}$$

(a) For the initial state

```
> s:=matrix([[0,1,0,0,0]]);
```

$$s := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

What is the probability you are in state 4 after 4 moves?

We need to flip both matrices. We'll use the Linear Algbera package and write them out again; M here is the transpose of the matrix above.

```
> with(LinearAlgebra):
  M:=<<1/5,1/5,1/5,1/5,1/5|1/10,1/10,2/5,1/5,1/5|2/5,1/5,1/5,1/10,
  1/10|1/5,2/5,1/10,1/5,1/10|1/5,1/5,1/5,3/10,1/10>>;
  X0:=<<0,1,0,0,0>>;
```

$$M := \begin{bmatrix} \frac{1}{5} & \frac{1}{10} & \frac{2}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{10} & \frac{1}{5} & \frac{2}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{2}{5} & \frac{1}{5} & \frac{1}{10} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{10} & \frac{1}{5} & \frac{3}{10} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

$$X0 := \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{27}$$

```
> evalm(M^4 &* X0);
```

$$\begin{bmatrix} \frac{2243}{10000} \\ \frac{2171}{10000} \\ \frac{1117}{5000} \\ \frac{961}{5000} \\ \frac{143}{1000} \end{bmatrix} \quad (28)$$

Thus we see that the probabilitiy you are in state 4 after 4 moves is 961/5000, or about

```
> 961./5000;
```

$$0.1922000000 \quad (29)$$

(b) What is the steady state matrix?

To find the steady state matrix, you find the equilibrium vector, and that becomes each column of the matrix (the matrix has an equillibrium vector as it is regular, that is, some power has all positive entries - the first power will do!). We take a general vector x and solve Mx = x, where the entries of x sum to 1.

```
> X:=<<a,b,c,d,e>>;
```

$$X := \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} \quad (30)$$

```
> Y:=evalm(M &* X);
  Y = X;
```

$$Y := \begin{bmatrix} \frac{1}{5}a + \frac{1}{10}b + \frac{2}{5}c + \frac{1}{5}d + \frac{1}{5}e \\ \frac{1}{5}a + \frac{1}{10}b + \frac{1}{5}c + \frac{2}{5}d + \frac{1}{5}e \\ \frac{1}{5}a + \frac{2}{5}b + \frac{1}{5}c + \frac{1}{10}d + \frac{1}{5}e \\ \frac{1}{5}a + \frac{1}{5}b + \frac{1}{10}c + \frac{1}{5}d + \frac{3}{10}e \\ \frac{1}{5}a + \frac{1}{5}b + \frac{1}{10}c + \frac{1}{10}d + \frac{1}{10}e \end{bmatrix}$$

$$Y = \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} \tag{31}$$

So here are the equations we need to solve:

```
> eq1:=Y[1,1]=a;
  eq2:=Y[2,1]=b;
  eq3:=Y[3,1]=c;
  eq4:=Y[4,1]=d;
  eq5:=Y[5,1]=e;
  eq6:=a+b+c+d+e=1;
```

$$\begin{gathered} eq1 := \frac{1}{5}\,a + \frac{1}{10}\,b + \frac{2}{5}\,c + \frac{1}{5}\,d + \frac{1}{5}\,e = a \\ eq2 := \frac{1}{5}\,a + \frac{1}{10}\,b + \frac{1}{5}\,c + \frac{2}{5}\,d + \frac{1}{5}\,e = b \\ eq3 := \frac{1}{5}\,a + \frac{2}{5}\,b + \frac{1}{5}\,c + \frac{1}{10}\,d + \frac{1}{5}\,e = c \\ eq4 := \frac{1}{5}\,a + \frac{1}{5}\,b + \frac{1}{10}\,c + \frac{1}{5}\,d + \frac{3}{10}\,e = d \\ eq5 := \frac{1}{5}\,a + \frac{1}{5}\,b + \frac{1}{10}\,c + \frac{1}{10}\,d + \frac{1}{10}\,e = e \\ eq6 := a + b + c + d + e = 1 \end{gathered} \tag{32}$$

```
> solve({eq1,eq2,eq3,eq4,eq5,eq6},{a,b,c,d,e});
```

$$\left\{a = \frac{451}{2021},\, b = \frac{438}{2021},\, c = \frac{453}{2021},\, d = \frac{388}{2021},\, e = \frac{291}{2021}\right\} \tag{33}$$

So the steady state matrix is

```
> S:<<451/2021,438/2021,453/2021,388/2021,291/2021>|<451/2021,
  438/2021,453/2021,388/2021,291/2021>|<451/2021,438/2021,453/2021,
  388/2021,291/2021>|<451/2021,438/2021,453/2021,388/2021,
  291/2021>|<451/2021,438/2021,453/2021,388/2021,291/2021>>;
```

$$\begin{bmatrix} \frac{451}{2021} & \frac{451}{2021} & \frac{451}{2021} & \frac{451}{2021} & \frac{451}{2021} \\ \frac{438}{2021} & \frac{438}{2021} & \frac{438}{2021} & \frac{438}{2021} & \frac{438}{2021} \\ \frac{453}{2021} & \frac{453}{2021} & \frac{453}{2021} & \frac{453}{2021} & \frac{453}{2021} \\ \frac{388}{2021} & \frac{388}{2021} & \frac{388}{2021} & \frac{388}{2021} & \frac{388}{2021} \\ \frac{291}{2021} & \frac{291}{2021} & \frac{291}{2021} & \frac{291}{2021} & \frac{291}{2021} \end{bmatrix} \tag{34}$$

9. Suppose a markov chain has a transition matrix with a row or a column of 0's. Prove that the markov chain is <u>not</u> regular.

Suppose one row, say the i-th consists of all 0's. Then in the associated picture (the directed graph) there are no arcs out of state i. It is clear then no matter what k is, there will never be a walk with k arcs from state i to any other state, and so the markov chain is not regular. A similar argument shows that if a column is entirely 0's in the traisition matrix, the markov chain is not regular as well.