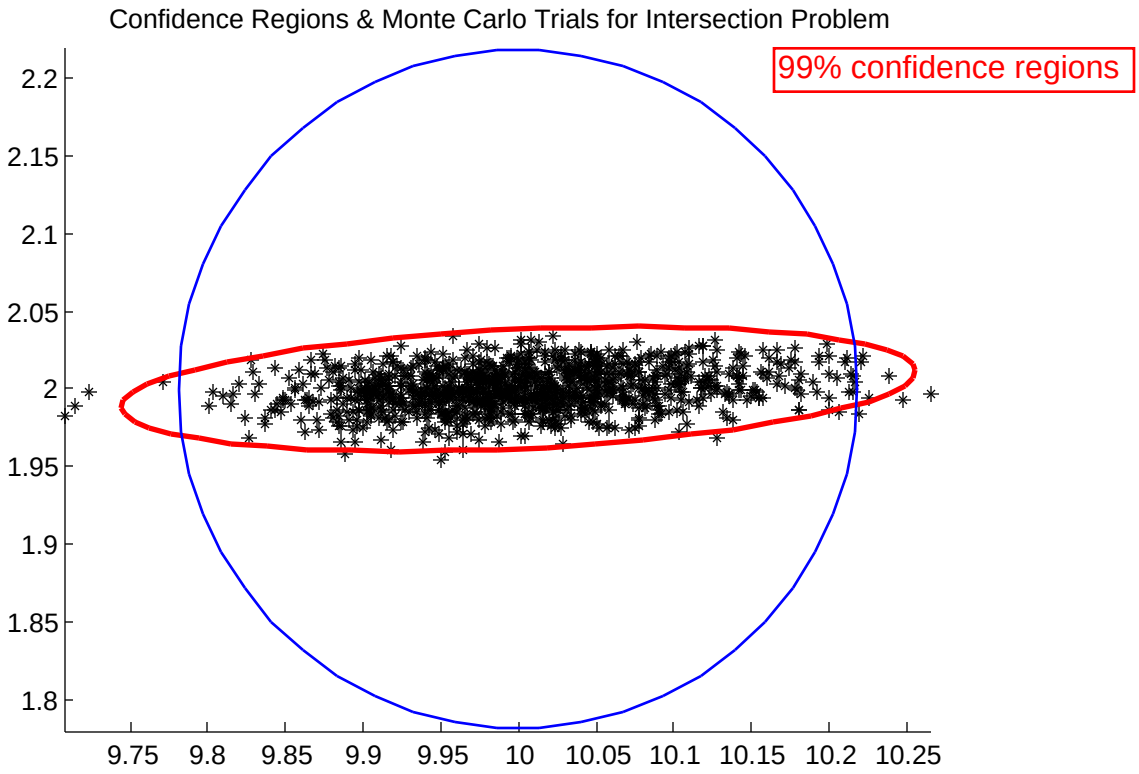
Confidence Regions & Monte Carlo Trials for Intersection Problem
99% confidence regions
2.2
2.15
2.1
2.05
2
1.95
1.9
1.85
1.8
9.75
9.8
9.85
9.9
9.95
10
10.05
10.1
10.15
10.2
10.25

```
% epex1.m  26-oct-2011
% error prop example (based on hw1 scatterplot)
% uses functions:
% function [ac Fa dFdxi dFdyi dFdxj dFdyj] =
azimuth2d(a,i,j,X,Y)
% function result=draw_ell(xorg,yorg,a,b,theta)

X=[0;0;10];
Y=[3;0;2];

dr=180/pi;
th1=84.289406862/dr;
th2=78.690067526/dr;
sig0=0.1/dr;
sig0_sq=sig0^2;
th1az=pi-th1;

for iter=1:2
  B=zeros(2,2);
  f=zeros(2,1);
  [ac Fa dFdxi dFdyi dFdxj
dFdyj]=azimuth2d(th1az,1,3,X,Y);
  B(1,:)=[dFdxj dFdyj];
  f(1)=-Fa;
  [ac Fa dFdxi dFdyi dFdxj
dFdyj]=azimuth2d(th2,2,3,X,Y);
  B(2,:)=[dFdxj dFdyj];
  f(2)=-Fa;
  W=eye(2);
  N=B'*W*B;
  t=B'*W*f;
  Ni=inv(N);
  del=Ni*t
  X(3)=X(3)+del(1);
  Y(3)=Y(3)+del(2);
  end

Qdd=Ni;
Sdd=sig0_sq*Qdd;
[V,D]=eig(Sdd);
% make D(1,1) the larger eignevalue
if(D(2,2) > D(1,1))
  temp=D(1,1);
```

```
  D(1,1)=D(2,2);
  D(2,2)=temp;
  tempv=V(:,1);
  V(:,1)=V(:,2);
  V(:,2)=tempv;
  end

P=0.99
fct=icdf('chi2',P,2);
a=sqrt(D(1,1)*fct);
b=sqrt(D(2,2)*fct);
sxy=Sdd(1,2);
sxx=Sdd(1,1);
syy=Sdd(2,2);
two_theta=atan2(2*sxy,sxx-syy);
theta=0.5*two_theta;
p3x=X(3);
p3y=Y(3);

radius=cep2(P,Sdd);


% ****************************************
% do monte carlo trials, plot outcomes
% ****************************************

for ntrials=1:1000
  e=random('norm',0,sig0,2,1);
  obs1=th1az + e(1);
  obs2=th2 + e(2);
  X(3)=10;
  Y(3)=2;
  for iter=1:4
    B=zeros(2,2);
    f=zeros(2,1);
    [ac Fa dFdxi dFdyi dFdxj
dFdyj]=azimuth2d(obs1,1,3,X,Y);
    B(1,:)=[dFdxj dFdyj];
    f(1)=-Fa;
    [ac Fa dFdxi dFdyi dFdxj
dFdyj]=azimuth2d(obs2,2,3,X,Y);
    B(2,:)=[dFdxj dFdyj];
    f(2)=-Fa;
```

```
    W=eye(2);
    N=B'*W*B;
    t=B'*W*f;
    Ni=inv(N);
    del=Ni*t;
    X(3)=X(3)+del(1);
    Y(3)=Y(3)+del(2);
    end
  if(ntrials == 1)
    hold on
    end
  plot(X(3),Y(3),'k*');
  end

% ****************************************
% plot confidence ellipse from error prop
% ****************************************

draw_ell(p3x,p3y,a,b,theta);
draw_cir(p3x,p3y,radius);
axis equal
title('Confidence Regions & Monte Carlo Trials for
Intersection Problem');
```

```
% azimuth2d.m  04-oct-2011
function [ac Fa dFdxi dFdyi dFdxj dFdyj] =
azimuth2d(a,i,j,X,Y)
xi=X(i);
yi=Y(i);
xj=X(j);
yj=Y(j);
dx=xj-xi;
dy=yj-yi;
Dij_sq=(dx)^2 + (dy)^2;
dFdxi=dy/Dij_sq;
dFdyi=-dx/Dij_sq;
dFdxj=-dFdxi;
dFdyj=-dFdyi;
ac=atan2(xj-xi,yj-yi);
if(ac < 0)
  ac=ac + 2*pi;
  end

% ac
% degrad=180/pi;
% ac*degrad

Fa=a - ac;
```

```
% draw_ell.m  22-oct-08
% function to draw ellipse

function result=draw_ell(xorg,yorg,a,b,theta)

th=theta;
x0=a;
y0=0;
nseg=50;
dalpha=2*pi/nseg;
for i=1:nseg
  alpha=i*dalpha;
  x1=a*cos(alpha);
  y1=b*sin(alpha);
  px0=xorg + cos(th)*x0 - sin(th)*y0;
  py0=yorg + sin(th)*x0 + cos(th)*y0;
  px1=xorg + cos(th)*x1 - sin(th)*y1;
  py1=yorg + sin(th)*x1 + cos(th)*y1;
  plot([px0 px1],[py0 py1],'r-','linewidth',2);
  if(i == 1)
    hold on
    end
  x0=x1;
  y0=y1;
  end
result=0;
```

```
% draw_cir.m  13-oct-08
function result=draw_cir(x0,y0,r)
xi=x0+r;
yi=y0;
n=50
degrad=180/pi;
dth=2*pi/n;
rth=0;
for i=1:n
  rth=rth+dth;
  costh=cos(rth);
  sinth=sin(rth);
  xip1=x0 + r*costh;
  yip1=y0 + r*sinth;
  plot([xi xip1],[yi yip1],'b-','linewidth',1);
  if(i==1)
    hold on
    end
  xi=xip1;
  yi=yip1;
  end
result=0;
```

```
% cep2.m 11-nov-04
% for given 2x2 covariance and probability P,
% compute radius yielding P under bivariate normal
% syntax res=cep2(P,cov);
% original in d:\classes\ce603_03\

function res=cep2(P,cov)
sx2=cov(1,1);
sy2=cov(2,2);
sxy=cov(1,2);
sx=sqrt(sx2);
sy=sqrt(sy2);
long=max([sx sy]);
dr=long/100;
t1=2*pi*sqrt(det(cov));
term1=1/t1;
covi=inv(cov);
X=zeros(2,1);
degrad=180/(pi);
nth=300;
dth=pi/nth;
accumP=0;
rr=0;
while(accumP < 0.5*P)
  rp=rr + 0.5*dr;
  tt=0;
  for j=1:nth
    thp=tt + 0.5*dth;
    X(1)=rp*cos(thp);
    X(2)=rp*sin(thp);
    term2=-0.5*(X'*covi*X);
    f=term1*exp(term2);
    dens=f;
    %mu=[0 0];
    %XX=[X(1) X(2)];
    %dens=mvnpdf(XX,mu,cov);
    da=rp*dth*dr;
    accumP=accumP + da*dens;
    tt=tt + dth;
    end
  rr=rr + dr;
  end
res=rr;
```