Simple example of a tectonic Inverse Problem (inspired by the rifting crisis in Afar)

Solving the problem using a standard minimization function.

Albert Tarantola, February 2006

Forward Problem

These expresions solve the problem of evaluating the displacements caused when a fissure opens in a 2D "elastic" medium (see the PDF text for details).

$$
\begin{aligned}
&Ux = x - X;\ \ Uy = y - Y;\ U = \sqrt{Ux^2 + Uy^2}\ ;\\
&\Delta = Exp[\delta];\\
&f1 = Exp\left[-\frac{U}{\Delta}\right];\\
&fx = Sin[\psi];\ fy = Cos[\psi];\\
&\beta = ArcCos\left[\frac{fx\,Ux + fy\,Uy}{U}\right];\\
&\alpha = \pi / 2 - \beta;\\
&f2 = Sin[\alpha]^2;\\
&Q = Exp[q];\\
&ux = Q\ f1\ f2\ (Ux / U);\ uy = Q\ f1\ f2\ (Uy / U);\ uH = \sqrt{ux^2 + uy^2}\ ;\ uz = uH / 7;
\end{aligned}
$$

The final expressions (for the three components of the displacements) are now used to define the functions solving the forward problem.

```
uxCAL[x_, y_, X_, Y_, δ_, ψ_, q_] = ux;
uyCAL[x_, y_, X_, Y_, δ_, ψ_, q_] = uy;
uzCAL[x_, y_, X_, Y_, δ_, ψ_, q_] = uz;
```

Generating the synthetic data

We shall have (simulated) observations at six points, whose coordinates are as follows.

```
x1 = 11.0; y1 = 24.5; x2 = 13.0; y2 = 28.5; x3 = 14.0; y3 = 21.5;
x4 = 15.0; y4 = 17.5; x5 = 07.0; y5 = 16.5; x6 = 08.0; y6 = 24.5;
```

To simulate our observations, we arbitrarily take some "true model".

```
Xtrue = 10.0; Ytrue = 20.0; δtrue = 1.0; ψtrue = 0.5; qtrue = -0.5;
```

To the computed values, we shall add some random "errors", whose "size" is $\epsilon = 0.01$.

```
ϵ = 0.01;
random := Random[Real, {-ϵ, ϵ}]
SeedRandom[123]
```

We compute the synthetic data, and, to give some realism to the values, we round them leaving only three decimals.

```
K = 1000.;
```

```
ω = uxCAL[x1, y1, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS1 = Round[K (ω + random)] / K;
Print["ux1=", %%, " , ux1(rounded)=", %]
ω = uyCAL[x1, y1, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS1 = Round[K (ω + random)] / K;
Print["uy1=", %%, " , uy1(rounded)=", %]
ω = uzCAL[x1, y1, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS1 = Round[K (ω + random)] / K;
Print["uz1=", %%, " , uz1(rounded)=", %]
```

```
ux1=0.0222763 , ux1(rounded)=0.016

uy1=0.100243 , uy1(rounded)=0.099

uz1=0.0146698 , uz1(rounded)=0.022
```

```
ω = uxCAL[x2, y2, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS2 = Round[K (ω + random)] / K;
Print["ux2=", %%, " , ux2(rounded)=", %]
ω = uyCAL[x2, y2, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS2 = Round[K (ω + random)] / K;
Print["uy2=", %%, " , uy2(rounded)=", %]
ω = uzCAL[x2, y2, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS2 = Round[K (ω + random)] / K;
Print["uz2=", %%, " , uz2(rounded)=", %]
```

```
ux2=0.0071395 , ux2(rounded)=0.016

uy2=0.0202286 , uy2(rounded)=0.015

uz2=0.0030645 , uz2(rounded)=0
```

```
ω = uxCAL[x3, y3, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS3 = Round[K (ω + random)] / K;
Print["ux3=", %%, " , ux3(rounded)=", %]
ω = uyCAL[x3, y3, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS3 = Round[K (ω + random)] / K;
Print["uy3=", %%, " , uy3(rounded)=", %]
ω = uzCAL[x3, y3, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS3 = Round[K (ω + random)] / K;
Print["uz3=", %%, " , uz3(rounded)=", %]
```

```
ux3=0.0676067 , ux3(rounded)=0.075

uy3=0.0253525 , uy3(rounded)=0.028

uz3=0.0103149 , uz3(rounded)=0.006
```

```
ω = uxCAL[x4, y4, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS4 = Round[K (ω + random)] / K;
Print["ux4=", %%, " , ux4(rounded)=", %]
ω = uyCAL[x4, y4, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS4 = Round[K (ω + random)] / K;
Print["uy4=", %%, " , uy4(rounded)=", %]
ω = uzCAL[x4, y4, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS4 = Round[K (ω + random)] / K;
Print["uz4=", %%, " , uz4(rounded)=", %]
```

```
ux4=0.000091652 , ux4(rounded)=0.003

uy4=-0.000045826 , uy4(rounded)=0.006

uz4=0.0000146386 , uz4(rounded)=-0.002
```

```
ω = uxCAL[x5, y5, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS5 = Round[K (ω + random)] / K;
Print["ux5=", %%, " , ux5(rounded)=", %]
ω = uyCAL[x5, y5, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS5 = Round[K (ω + random)] / K;
Print["uy5=", %%, " , uy5(rounded)=", %]
ω = uzCAL[x5, y5, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS5 = Round[K (ω + random)] / K;
Print["uz5=", %%, " , uz5(rounded)=", %]
```

```
ux5=-0.0693041 , ux5(rounded)=-0.077

uy5=-0.0808548 , uy5(rounded)=-0.083

uz5=0.0152132 , uz5(rounded)=0.018
```

```
ω = uxCAL[x6, y6, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uxOBS6 = Round[K (ω + random)] / K;
Print["ux6=", %%, " , ux6(rounded)=", %]
ω = uyCAL[x6, y6, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uyOBS6 = Round[K (ω + random)] / K;
Print["uy6=", %%, " , uy6(rounded)=", %]
ω = uzCAL[x6, y6, Xtrue, Ytrue, δtrue, ψtrue, qtrue];
uzOBS6 = Round[K (ω + random)] / K;
Print["uz6=", %%, " , uz6(rounded)=", %]
```

```
ux6=-0.0148412 , ux6(rounded)=-0.02

uy6=0.0333928 , uy6(rounded)=0.032

uz6=0.00522033 , uz6(rounded)=0.009
```

Plot of the Data

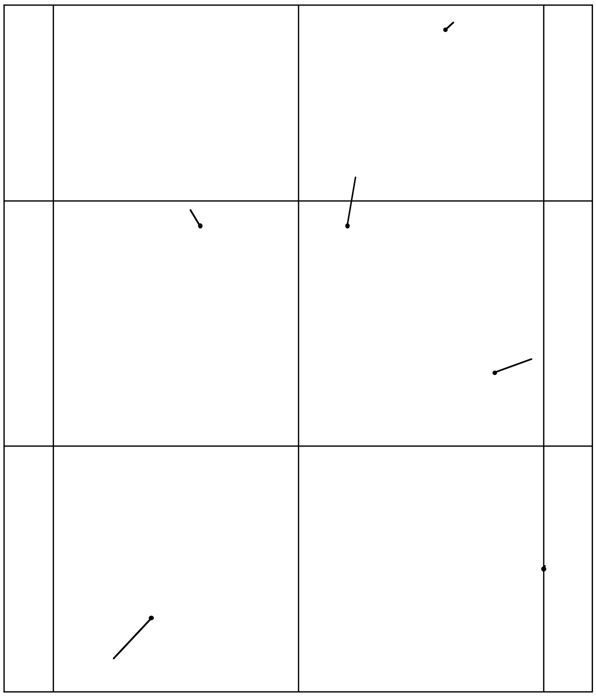
Note: the displacements have been multiplied by 10, to make them visible.

```
MeasuringPoints = Graphics[{
    Point[{x1, y1}], Point[{x2, y2}],
    Point[{x3, y3}], Point[{x4, y4}], Point[{x5, y5}], Point[{x6, y6}]
   }];
Arrows = Graphics[{
    Line[{{x1, y1}, {x1 + 10 uxOBS1, y1 + 10 uyOBS1}}],
    Line[{{x2, y2}, {x2 + 10 uxOBS2, y2 + 10 uyOBS2}}],
    Line[{{x3, y3}, {x3 + 10 uxOBS3, y3 + 10 uyOBS3}}],
    Line[{{x4, y4}, {x4 + 10 uxOBS4, y4 + 10 uyOBS4}}],
    Line[{{x5, y5}, {x5 + 10 uxOBS5, y5 + 10 uyOBS5}}],
    Line[{{x6, y6}, {x6 + 10 uxOBS6, y6 + 10 uyOBS6}}]
   }];
```

Let us also add some coordinate lines...

```
left = Line[{{4, 15}, {4, 29}}];
top = Line[{{4, 29}, {16, 29}}];
right = Line[{{16, 29}, {16, 15}}];
bottom = Line[{{16, 15}, {4, 15}}];
vert1 = Line[{{5, 15}, {5, 29}}];
vert2 = Line[{{10, 15}, {10, 29}}];
vert3 = Line[{{15, 15}, {15, 29}}];
hor1 = Line[{{16, 20}, {4, 20}}];
hor2 = Line[{{16, 25}, {4, 25}}];
Lines = Graphics[{left, top, right, bottom, vert1, vert2, vert3, hor1, hor2}];
```

```
Show[{MeasuringPoints, Arrows, Lines}, AspectRatio → Automatic]
```



Resolution of the inverse problem using an optimization technique

Introducing the vector of calculated observations

```
dcal = {uxCAL[x1, y1, X, Y, δ, ψ, q],
   uyCAL[x1, y1, X, Y, δ, ψ, q], uzCAL[x1, y1, X, Y, δ, ψ, q],
   uxCAL[x2, y2, X, Y, δ, ψ, q], uyCAL[x2, y2, X, Y, δ, ψ, q],
   uzCAL[x2, y2, X, Y, δ, ψ, q], uxCAL[x3, y3, X, Y, δ, ψ, q],
   uyCAL[x3, y3, X, Y, δ, ψ, q], uzCAL[x3, y3, X, Y, δ, ψ, q],
   uxCAL[x4, y4, X, Y, δ, ψ, q], uyCAL[x4, y4, X, Y, δ, ψ, q],
   uzCAL[x4, y4, X, Y, δ, ψ, q], uxCAL[x5, y5, X, Y, δ, ψ, q],
   uyCAL[x5, y5, X, Y, δ, ψ, q], uzCAL[x5, y5, X, Y, δ, ψ, q], uxCAL[x6, y6, X,
    Y, δ, ψ, q], uyCAL[x6, y6, X, Y, δ, ψ, q], uzCAL[x6, y6, X, Y, δ, ψ, q]};
```

Introducing the vector of observed values and the associated covariance matrix

```
dobs = {uxOBS1, uyOBS1, uzOBS1, uxOBS2, uyOBS2, uzOBS2, uxOBS3, uyOBS3, uzOBS3,
   uxOBS4, uyOBS4, uzOBS4, uxOBS5, uyOBS5, uzOBS5, uxOBS6, uyOBS6, uzOBS6};
CD = ϵ^2 IdentityMatrix[18];
ICD = Inverse[CD];
```

The a priori model and the covariance matrix describing uncertainties in the a priori model .

We assume that we have the following a priori information on the model parameters:

$X = 12 \pm 20$; $Y = 12 \pm 20$; $\delta = 2 \pm 1$; $\psi = \text{ArcTan}[1] \pm 1$; $q = 0 \pm 2$.

This is a quite weak information, but it is useful for two reasons: i) sometimes - when we have low quality data - it may help obtaining reasonable solutions; a ii) the algorithm uses a metric in the model space, and the convergence is more rapid when this metric is reasonable.

```
Xprior = 12.; Yprior = 12.; δprior = 2.; ψprior = ArcTan[1.]; qprior = 0;
σX = 20.; σY = 20.; σδ = 1.; σψ = 1.; σq = 2;
```

Note that I use the von Mises distribution for the angle ψ.

$$\text{prior} = \text{Exp}\left[-\frac{1}{2}\ \frac{(\text{X}-\text{Xprior})^2}{\sigma\text{X}^2}\right]\ \text{Exp}\left[-\frac{1}{2}\ \frac{(\text{Y}-\text{Yprior})^2}{\sigma\text{Y}^2}\right]$$
$$\text{Exp}\left[-\frac{1}{2}\ \frac{(\delta-\delta\text{prior})^2}{\sigma\delta^2}\right]\ \text{Exp}\left[\frac{\text{Cos}[\psi-\psi\text{prior}]}{\sigma\psi^2}\right]\ \text{Exp}\left[-\frac{1}{2}\ \frac{(\text{q}-\text{qprior})^2}{\sigma\text{q}^2}\right];$$

The misfit function (with a priori information) .

Warning: for some strange reason, the developers of *Mathematica* use the same symbol (a dot) to designate a product of two matrices and the (elementary) scalar product of two vectors. The expression below should be written S2=-Transpose[(tcal-tobs)].Inverse[CD].(tcal-tobs) , but *Mathematica* would misunderstand the expression.

$$\text{likelihood} = \text{Exp}\left[-\frac{1}{2}\ (\text{dcal}-\text{dobs}).\text{Inverse}[\text{CD}].(\text{dcal}-\text{dobs})\right];$$

```
posterior = prior * likelihood;
```

Using the FindMinimum function in Mathematica.

Defining the misfit function:

```
S = -Log[posterior];
```

:et us check the value of S at the "true point":

```
S /. {X → Xtrue, Y → Ytrue, δ → δtrue, ψ → ψtrue, q → qtrue}
```

```
1.92519
```

When initializing the algorithm at the prior point mprior there is an Overflow:

```
{Smin, replace} =
 FindMinimum[S, {X, Xprior}, {Y, Yprior}, {δ, δprior}, {ψ, ψprior}, {q, qprior}]
```

```
General::unfl : Underflow occurred in computation. More…

FindMinimum::ngnum : The gradient is not a vector of
   numbers at {X, Y, δ, ψ, q} = {-25.5584, ≪3≫, ≪18≫}. More…
```

```
{116.057, {X → 11.8416, Y → 13.2029, δ → 2.16241, ψ → 6.1307, q → -1.72943}}
```

After some trial and error I take an starting point form which the algorithm converges:

```
{Smin, replace} = FindMinimum[S, {X, 12.}, {Y, 18.}, {δ, 1.3}, {ψ, 0.7}, {q, -0.7}]
```

```
{1.29492,
 {X → 10.0682, Y → 19.8434, δ → 0.994468, ψ → 0.488357, q → -0.441343}}
```

Of course, the convergence point is independent of the starting point:

```
{Smin, replace} = FindMinimum[S, {X, 11.}, {Y, 20.}, {δ, 1.4}, {ψ, 0.8}, {q, -0.8}]
```

```
{1.29492,
 {X → 10.0682, Y → 19.8434, δ → 0.994468, ψ → 0.488357, q → -0.441343}}
```

Comparison between the "observed" and the computed data (and evaluation of the final residuals).

```
m = {X, Y, δ, ψ, q};
{Xpost, Ypost, δpost, ψpost, qpost} = m /. replace;
dfinal = {
   a01 = uxCAL[x1, y1, Xpost, Ypost, δpost, ψpost, qpost],
   a02 = uyCAL[x1, y1, Xpost, Ypost, δpost, ψpost, qpost],
   a03 = uzCAL[x1, y1, Xpost, Ypost, δpost, ψpost, qpost],
   a04 = uxCAL[x2, y2, Xpost, Ypost, δpost, ψpost, qpost],
   a05 = uyCAL[x2, y2, Xpost, Ypost, δpost, ψpost, qpost],
   a06 = uzCAL[x2, y2, Xpost, Ypost, δpost, ψpost, qpost],
   a07 = uxCAL[x3, y3, Xpost, Ypost, δpost, ψpost, qpost],
   a08 = uyCAL[x3, y3, Xpost, Ypost, δpost, ψpost, qpost],
   a09 = uzCAL[x3, y3, Xpost, Ypost, δpost, ψpost, qpost],
   a10 = uxCAL[x4, y4, Xpost, Ypost, δpost, ψpost, qpost],
   a11 = uyCAL[x4, y4, Xpost, Ypost, δpost, ψpost, qpost],
   a12 = uzCAL[x4, y4, Xpost, Ypost, δpost, ψpost, qpost],
   a13 = uxCAL[x5, y5, Xpost, Ypost, δpost, ψpost, qpost],
   a14 = uyCAL[x5, y5, Xpost, Ypost, δpost, ψpost, qpost],
   a15 = uzCAL[x5, y5, Xpost, Ypost, δpost, ψpost, qpost],
   a16 = uxCAL[x6, y6, Xpost, Ypost, δpost, ψpost, qpost],
   a17 = uyCAL[x6, y6, Xpost, Ypost, δpost, ψpost, qpost],
   a18 = uzCAL[x6, y6, Xpost, Ypost, δpost, ψpost, qpost]};
dobs
Round[K (dfinal)] / K
Round[K (dfinal - dobs)] / K
```

```
{0.016, 0.099, 0.022, 0.016, 0.015, 0, 0.075, 0.028, 0.006,
 0.003, 0.006, -0.002, -0.077, -0.083, 0.018, -0.02, 0.032, 0.009}
```

```
{0.02, 0.1, 0.015, 0.007, 0.02, 0.003, 0.074, 0.031,
 0.011, 0, 0, 0, -0.076, -0.083, 0.016, -0.015, 0.034, 0.005}
```

```
{0.004, 0.001, -0.007, -0.009, 0.005, 0.003, -0.001, 0.003, 0.005,
 -0.003, -0.006, 0.002, 0.001, 0, -0.002, 0.005, 0.002, -0.004}
```

Evaluation of the a posteriori uncertainties

We now evaluate the posterior covariance matrix.

```
W = ( D[D[S, X], X]  D[D[S, X], Y]  D[D[S, X], δ]  D[D[S, X], ψ]  D[D[S, X], q]
      D[D[S, Y], X]  D[D[S, Y], Y]  D[D[S, Y], δ]  D[D[S, Y], ψ]  D[D[S, Y], q]
      D[D[S, δ], X]  D[D[S, δ], Y]  D[D[S, δ], δ]  D[D[S, δ], ψ]  D[D[S, δ], q]
      D[D[S, ψ], X]  D[D[S, ψ], Y]  D[D[S, ψ], δ]  D[D[S, ψ], ψ]  D[D[S, ψ], q]
      D[D[S, q], X]  D[D[S, q], Y]  D[D[S, q], δ]  D[D[S, q], ψ]  D[D[S, q], q] );
Wnum = Chop[W /. replace];
CMpost = Inverse[Wnum];
MatrixForm[CMpost]
```

$$\begin{pmatrix} 0.0337852 & -0.011168 & -0.00248696 & -0.00890535 & 0.009144 \\ -0.011168 & 0.0626093 & -0.00413022 & 0.0146149 & 0.00175911 \\ -0.00248696 & -0.00413022 & 0.0699174 & 0.00584384 & -0.121647 \\ -0.00890535 & 0.0146149 & 0.00584384 & 0.00830845 & -0.0128961 \\ 0.009144 & 0.00175911 & -0.121647 & -0.0128961 & 0.215879 \end{pmatrix}$$

The standard deviations are:

```
σ1 = √CMpost[[1, 1]]
σ2 = √CMpost[[2, 2]]
σ3 = √CMpost[[3, 3]]
σ4 = √CMpost[[4, 4]]
σ5 = √CMpost[[5, 5]]
```

```
0.183807
```

```
0.250219
```

```
0.264419
```

```
0.0911507
```

```
0.464628
```

The basic results are:

$X = 10.1 \pm 0.2$; $Y = 19.8 \pm 0.3$; $\delta = 1.0 \pm 0.3$; $\psi = 0.49 \pm 0.1$; $q = -0.44 \pm 0.5$.

This is to be compared with the "true model", that was (remember that we contaminated the "true obseved values" with some errors):

$X = 10.$; $Y = 20.$; $\delta = 1.$; $\psi = 0.5$; $q = -0.5$.

Remember also the the a priori model was

$X = 12. \pm 20.$; $Y = 12. \pm 20.$; $\delta = 2. \pm 1.$; $\psi = 0.8. \pm 1.$; $q = 0. \pm 2.$.

We see that all the a posteriori uncertainties are smaller that the a priori ones, so, our data has actually brought information on all the parameters.

The covariances are:

$$\rho 12 = \frac{\text{CMpost[[1, 2]]}}{\sigma 1\ \sigma 2};\ \rho 13 = \frac{\text{CMpost[[1, 3]]}}{\sigma 1\ \sigma 3};$$
$$\rho 14 = \frac{\text{CMpost[[1, 4]]}}{\sigma 1\ \sigma 4};\ \rho 15 = \frac{\text{CMpost[[1, 5]]}}{\sigma 1\ \sigma 5};\ \rho 23 = \frac{\text{CMpost[[2, 3]]}}{\sigma 2\ \sigma 3};$$
$$\rho 24 = \frac{\text{CMpost[[2, 4]]}}{\sigma 2\ \sigma 4};\ \rho 25 = \frac{\text{CMpost[[2, 5]]}}{\sigma 2\ \sigma 5};\ \rho 34 = \frac{\text{CMpost[[3, 4]]}}{\sigma 3\ \sigma 4};$$
$$\rho 35 = \frac{\text{CMpost[[3, 5]]}}{\sigma 3\ \sigma 5};\ \rho 45 = \frac{\text{CMpost[[4, 5]]}}{\sigma 4\ \sigma 5};$$
$$\text{CorrelationMatrix} = \begin{pmatrix} 1 & \rho 12 & \rho 13 & \rho 14 & \rho 15 \\ \rho 12 & 1 & \rho 23 & \rho 24 & \rho 25 \\ \rho 13 & \rho 23 & 1 & \rho 34 & \rho 35 \\ \rho 14 & \rho 24 & \rho 34 & 1 & \rho 45 \\ \rho 15 & \rho 25 & \rho 35 & \rho 45 & 1 \end{pmatrix};$$

```
MatrixForm[CorrelationMatrix]
```

$$\begin{pmatrix} 1 & -0.242824 & -0.0511697 & -0.53153 & 0.10707 \\ -0.242824 & 1 & -0.0624254 & 0.640789 & 0.015131 \\ -0.0511697 & -0.0624254 & 1 & 0.242463 & -0.990152 \\ -0.53153 & 0.640789 & 0.242463 & 1 & -0.304505 \\ 0.10707 & 0.015131 & -0.990152 & -0.304505 & 1 \end{pmatrix}$$

We see that some correlations are strong: most notably, the correlation between the two parameters δ and q is -0.99 . This means that our best estimate for these two parameters, if wrong, must we wrong in a stronly anti-correlated manner: if the actual value of δ is larger (resp. smaller) than its posterior value , then, the actual value of q must be smaller (resp. larger) than its posterior value .

Random generation of posterior models

At this point, my suggestion is as follows. We use the approximation that the posterior probability distribution for the parameters {X,Y,δ,ψ,q} is a Gaussian distribution, of which we know the mean vector, { 10.1 , 19.8 , 0.99 , 0.49, -0.44 } , and the covariance matrix,

$$\begin{pmatrix} 0.0337852 & -0.011168 & -0.00248696 & -0.00890535 & 0.009144 \\ -0.011168 & 0.0626093 & -0.00413022 & 0.0146149 & 0.00175911 \\ -0.00248696 & -0.00413022 & 0.0699174 & 0.00584384 & -0.121647 \\ -0.00890535 & 0.0146149 & 0.00584384 & 0.00830845 & -0.0128961 \\ 0.009144 & 0.00175911 & -0.121647 & -0.0128961 & 0.215879 \end{pmatrix}$$

. We can use this Gaussian distribution to generate some random solutions, and we can represent the correponding results (i.e., draw the randomly generated fissures). This should convey the actual information that we really have, much better that any examination of numbers. [Note: it would have been better to use the actual posterior distribution, not the Gaussian approximation.]

Here is the computation. See the drawing in the PDF document.

I first define the posterior probability density (note that it is normalized so that its maximum value is 1).

```
m = {X, Y, δ, ψ, q};
mpost = m /. replace;
```

$$\text{posteriorPDF[X\_, Y\_, }\delta\text{\_, }\psi\text{\_, q\_] = Exp}\left[-\frac{1}{2}\ (\ \text{(m - mpost).Wnum.(m - mpost)}\ )\ \right];$$

Then, I use the simple rejection method to obtain sample points.

Sampling (and printing the results)

```
SeedRandom[321]
k = 3.;
Do[{
  RX = Random[Real, {Xpost - k σ1, Xpost + k σ1}],
  RY = Random[Real, {Ypost - k σ2, Ypost + k σ2}],
  Rδ = Random[Real, {δpost - k σ3, δpost + k σ3}],
  Rψ = Random[Real, {ψpost - k σ4, ψpost + k σ4}],
  Rq = Random[Real, {qpost - k σ5, qpost + k σ5}],
  α = posteriorPDF[RX, RY, Rδ, Rψ, Rq],
  TestValue = Random[Real, {0, 1}],
  If[α > TestValue,
   {Print["{X,Y,δ,ψ,q} = ", {RX, RY, Rδ, Rψ, Rq}, " , α = ", α]}]
 }, {25000}]
```

```
{X,Y,δ,ψ,q} =
 {10.0909, 19.6632, 0.774745, 0.448043, -0.0231928} , α = 0.436797

{X,Y,δ,ψ,q} =
 {9.86834, 20.0848, 0.88124, 0.606836, -0.306953} , α = 0.287708

{X,Y,δ,ψ,q} =
 {9.70666, 20.2892, 1.13745, 0.573528, -0.790382} , α = 0.0102387

{X,Y,δ,ψ,q} = {9.91912, 19.5134, 1.2302, 0.497, -0.868714} , α = 0.139157

{X,Y,δ,ψ,q} =
 {10.3134, 19.7116, 1.08668, 0.436908, -0.580539} , α = 0.348497

{X,Y,δ,ψ,q} =
 {9.83981, 19.9763, 0.882641, 0.583398, -0.156653} , α = 0.0187163

{X,Y,δ,ψ,q} = {10.0504, 20.0129, 1.30324, 0.491916, -1.00889} , α = 0.1728

{X,Y,δ,ψ,q} =
 {9.88419, 19.2355, 1.24728, 0.518022, -0.912791} , α = 0.00206815

{X,Y,δ,ψ,q} =
 {10.3263, 19.6883, 0.891467, 0.493942, -0.144367} , α = 0.0167688

{X,Y,δ,ψ,q} =
 {10.1487, 19.9548, 0.558699, 0.446514, 0.347309} , α = 0.193263

{X,Y,δ,ψ,q} =
 {10.0525, 20.1809, 0.929147, 0.532193, -0.292803} , α = 0.224602

{X,Y,δ,ψ,q} =
 {9.98947, 19.9341, 0.824007, 0.462444, -0.23285} , α = 0.175195

{X,Y,δ,ψ,q} =
 {10.3248, 19.6463, 0.897017, 0.413951, -0.116591} , α = 0.0335831

{X,Y,δ,ψ,q} =
 {9.82674, 19.4076, 0.972539, 0.38994, -0.404232} , α = 0.026328
```

```
{X,Y,δ,ψ,q} =
 {10.0141, 19.4169, 0.974945, 0.429759, -0.402893} , α = 0.159222

{X,Y,δ,ψ,q} =
 {9.82088, 19.8799, 1.04336, 0.549192, -0.471635} , α = 0.116289

{X,Y,δ,ψ,q} =
 {10.0683, 20.5021, 1.45749, 0.681703, -1.32964} , α = 0.0028071

{X,Y,δ,ψ,q} =
 {9.95844, 19.9411, 1.05503, 0.522048, -0.598833} , α = 0.632009

{X,Y,δ,ψ,q} = {10.004, 20.0533, 1.12711, 0.55698, -0.686286} , α = 0.59345

{X,Y,δ,ψ,q} =
 {9.82675, 19.7317, 1.0137, 0.502666, -0.498797} , α = 0.302691

{X,Y,δ,ψ,q} =
 {10.1156, 19.9189, 1.05012, 0.491818, -0.568373} , α = 0.727002

{X,Y,δ,ψ,q} =
 {10.13, 19.8856, 1.00352, 0.535448, -0.500114} , α = 0.497702

{X,Y,δ,ψ,q} =
 {9.77948, 19.9876, 0.892334, 0.567618, -0.236168} , α = 0.110308

{X,Y,δ,ψ,q} =
 {10.1075, 20.0678, 0.795451, 0.526308, -0.0740446} , α = 0.381208
```

Sampling (and drawing the results)

```
SeedRandom[321]
k = 3.;
Fissure = Table[0, {100}];
i = 0;
Do[{
  RX = Random[Real, {Xpost - k σ1, Xpost + k σ1}],
  RY = Random[Real, {Ypost - k σ2, Ypost + k σ2}],
  Rδ = Random[Real, {δpost - k σ3, δpost + k σ3}],
  Rψ = Random[Real, {ψpost - k σ4, ψpost + k σ4}],
  Rq = Random[Real, {qpost - k σ5, qpost + k σ5}],
  α = posteriorPDF[RX, RY, Rδ, Rψ, Rq],
  TestValue = Random[Real, {0, 1}],
  If[α > TestValue,
   {i = i + 1,
    center = {RX, RY},
    RΔ = Exp[Rδ],
    point1 = {RX + (RΔ / 2) Cos[Rψ], RY - (RΔ / 2) Sin[Rψ]},
    point2 = {RX - (RΔ / 2) Cos[Rψ], RY + (RΔ / 2) Sin[Rψ]},
    RQ = Exp[Rq],
    pointQ = {RX + RQ Sin[Rψ], RY + RQ Cos[Rψ]},
    lineP = Line[{point1, point2}],
    lineQ = Line[{center, pointQ}],
    Fissure[[i]] = Graphics[{lineP, lineQ}]
   }]
 }, {25000}]
```

```
FissureSet = {
   Fissure[[01]], Fissure[[02]], Fissure[[03]], Fissure[[04]], Fissure[[05]],
   Fissure[[06]], Fissure[[07]], Fissure[[08]], Fissure[[09]], Fissure[[10]],
   Fissure[[11]], Fissure[[12]], Fissure[[13]], Fissure[[14]]};
Show[{MeasuringPoints, Arrows, FissureSet, Lines},
 PlotRange → All, AspectRatio → Automatic]
Show[{FissureSet}, PlotRange → All, AspectRatio → Automatic]
```