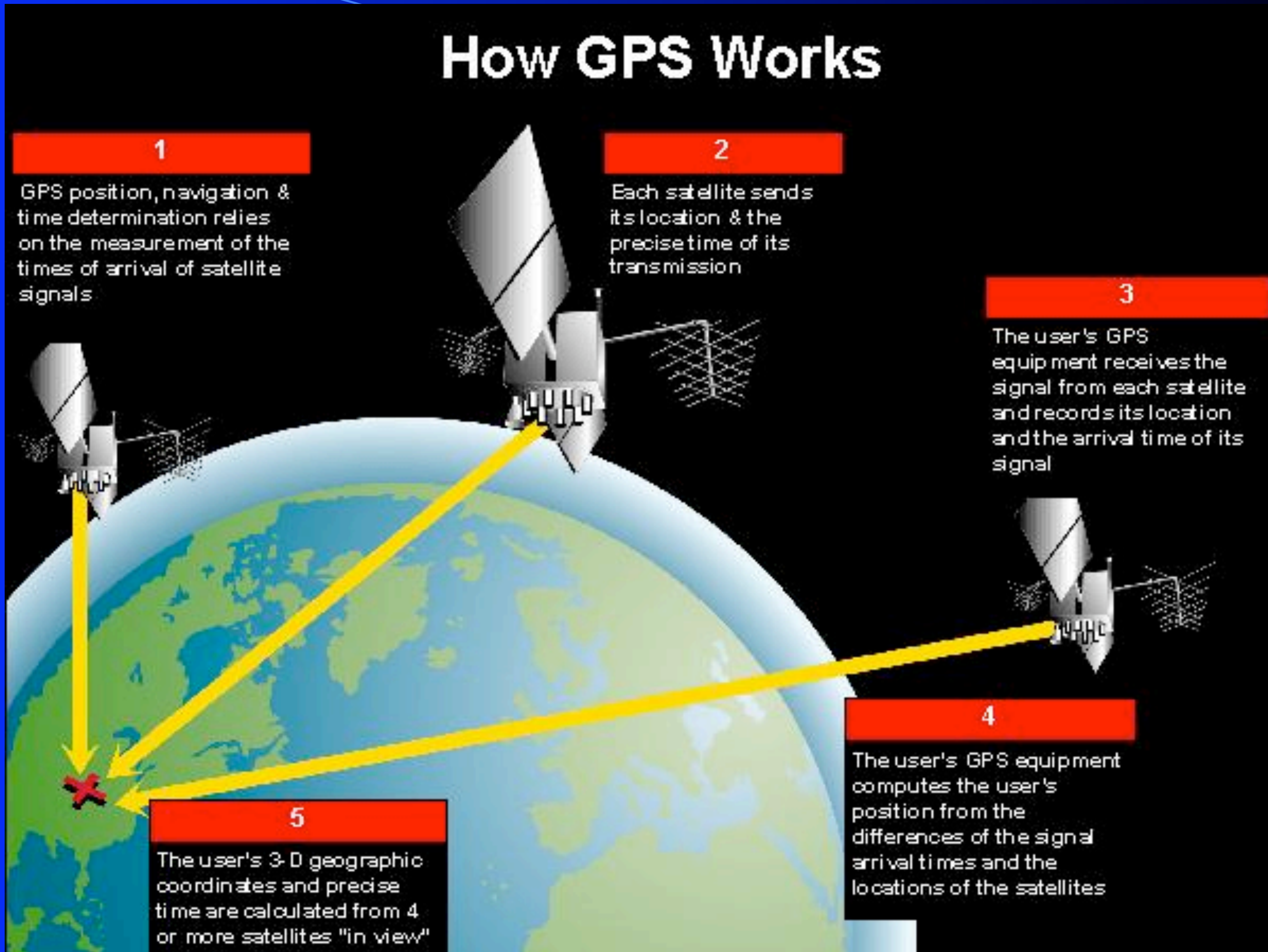
GPS point positioning

(using psuedorange from code).

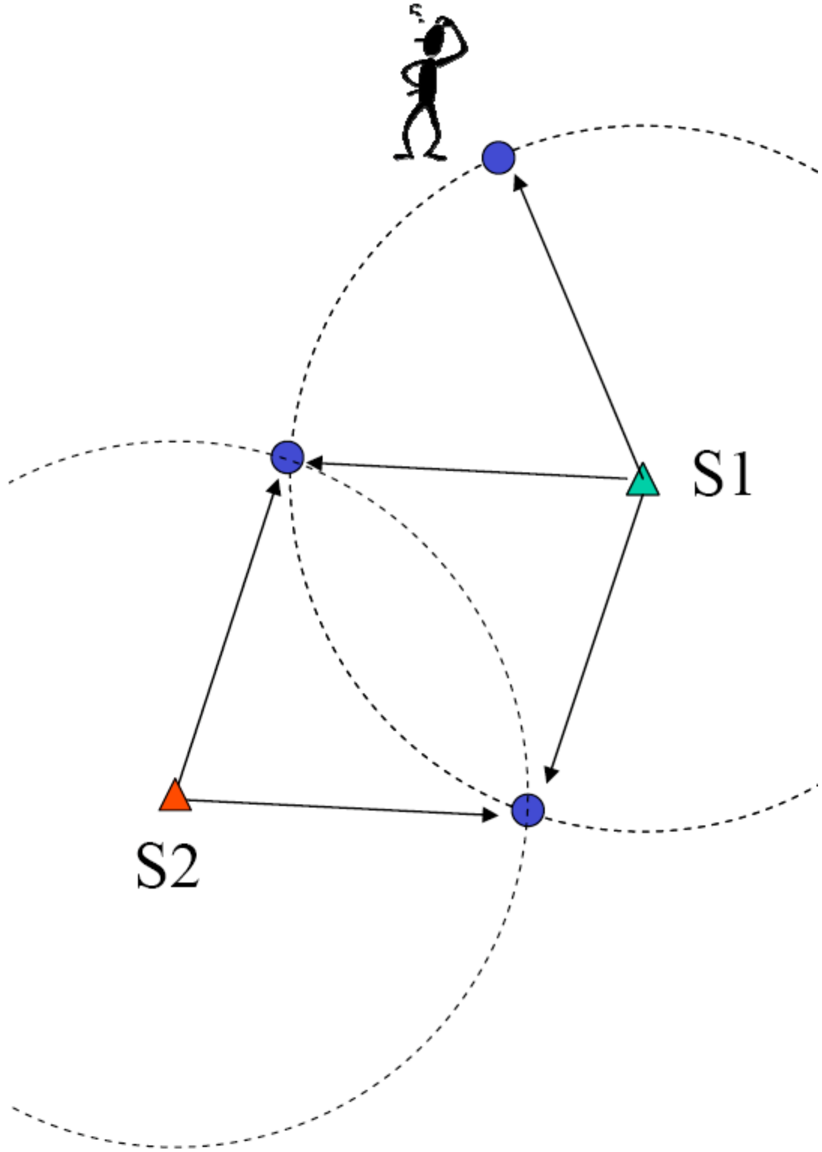
and earthquake location

(using P or S travel times).


How GPS Works
1
GPS position, navigation & time determination relies on the measurement of the times of arrival of satellite signals
2
Each satellite sends its location & the precise time of its transmission
3
The user's GPS equipment receives the signal from each satellite and records its location and the arrival time of its signal
4
The user's GPS equipment computes the user's position from the differences of the signal arrival times and the locations of the satellites
5
The user's 3-D geographic coordinates and precise time are calculated from 4 or more satellites "in view"

2D Example Continued

- Ranging to 1 station places us anywhere on a circle
- Ranging to 2 stations reduces uncertainty to only 2 points
- Could use a 3rd station to determine unique position estimate



Code Point Positioning

- Write pseudorange as a function of
  - Spacecraft position X^k,...
  - Receiver position (ECEF) X_u,...
  - Clock errors of spacecraft and receiver

$$\tau_u^k = \left[\sqrt{(x_u - x^k)^2 + (y_u - y^k)^2 + (z_u - z^k)^2} + b_u - B^k\right] + \nu_u^k$$

  - Measure pseudorange from >= 4 satellites and you can solve for x_u, y_u, z_u, b_u

$$\tau_u^k = f^k(x_u, y_u, z_u, b_u) + \nu_u^k$$

From J. HOW, MIT, GPS-RJH5.pdf

$$P_1 = \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} + b$$
$$P_2 = \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} + b$$
$$P_3 = \sqrt{(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2} + b$$
$$P_4 = \sqrt{(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2} + b$$

Position Equations

Where:

-P_i = Measured PseudoRange to the i^{th} SV, to be found (pseudorange because not measureing range – measureing time, which can e converted to range).

Position Equations

$$P_1 = \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} + b$$
$$P_2 = \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} + b$$
$$P_3 = \sqrt{(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2} + b$$
$$P_4 = \sqrt{(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2} + b$$

Where:

-X_i , Y_i , Z_i = Position of the i^{th} SV, Cartesian Coordinates, known.

-X , Y , Z = User position, Cartesian Coordinates, to be found

- b = User clock bias (in distance units), to be found.

Position Equations

$$P_1 = \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} + b$$
$$P_2 = \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} + b$$
$$P_3 = \sqrt{(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2} + b$$
$$P_4 = \sqrt{(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2} + b$$

The equations above are nonlinear in the terms we are looking for (x, y, z).

They cannot be solved directly (by simply inverting a matrix - at least mathematically) as in the case of linear equations.

Will solve a linearized approximation to these equations iteratively.

Position Equations

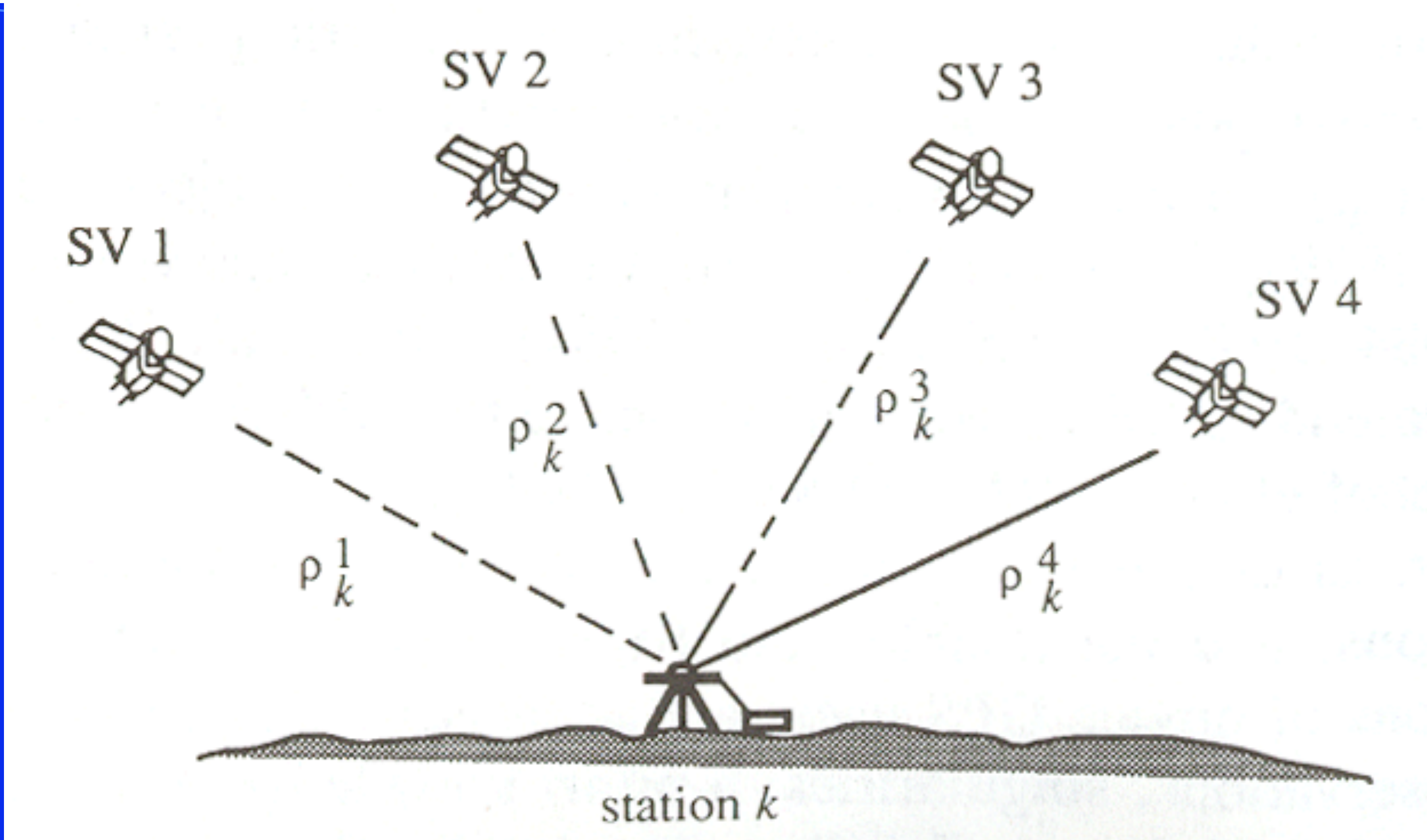
$$P_1 = \sqrt{(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2} + b$$
$$P_2 = \sqrt{(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2} + b$$
$$P_3 = \sqrt{(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2} + b$$
$$P_4 = \sqrt{(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2} + b$$

- Use an initial estimate (guess) of the user position, (X,Y,Z), and b.

- Calculate an adjustment to the location.

- Use new, adjusted location as the guess.

- Continue till converges or something bad happens (does not converge or blows up).



$$T^R = t^R + \tau^R \qquad T^S = t^S + \tau^S$$

$$P^{RS} = \left(\left(t^R + \tau^R\right) - \left(t^S + \tau^S\right)\right) c = \left(t^R - t^S\right) c + \left(\tau^R - \tau^S\right) c = \rho^{RS}\left(t^R, t^S\right) + \left(\tau^R - \tau^S\right) c$$

Pseudo range – we measure time, not range.
Calculate range from *r=ct*

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

From Pathagoras

$$\rho^{RS}\left(t^R,t^S\right)=\sqrt{\left(x^S\left(t^S\right)-x^R\left(t^R\right)\right)^2+\left(y^S\left(t^S\right)-y^R\left(t^R\right)\right)^2+\left(z^S\left(t^S\right)-z^R\left(t^R\right)\right)^2}$$

(x^S, y^S, z^S) and t^S known from satellite navigation message

(x^R, y^R, z^R) and t^R are 4 unknowns

Assume *c* constant along path, ignore relativity.

Complicating detail, satellite position has to be calculated at transmission time, not receive time.

Satellite range can change by up to 60 m during the approximately 0.07 sec travel time from satellite to receiver.

Using receive time can result in 10's m error in range.

Calculating satellite transmit time

$$t^S(0) = t^R = \left(T^R - \tau^R\right)$$

$$t^S(1) = t^R - \frac{\rho^{SR}\left(t^R, t^S(0)\right)}{c}$$

$$t^S(2) = t^R - \frac{\rho^{SR}\left(t^R, t^S(1)\right)}{c}$$

$$\vdots$$

Start w/ receiver time, need receiver clock bias

(once receiver is operating clock bias is kept less than few milliseconds)

$$P^{R1}\left(t^R,t^1\right)=\sqrt{\left(x^1\left(t^1\right)-x^R\left(t^R\right)\right)^2+\left(y^1\left(t^1\right)-y^R\left(t^R\right)\right)^2+\left(z^1\left(t^1\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^1\right)c$$

$$P^{R2}\left(t^R,t^2\right)=\sqrt{\left(x^2\left(t^2\right)-x^R\left(t^R\right)\right)^2+\left(y^2\left(t^2\right)-y^R\left(t^R\right)\right)^2+\left(z^2\left(t^2\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^2\right)c$$

$$P^{R3}\left(t^R,t^3\right)=\sqrt{\left(x^3\left(t^3\right)-x^R\left(t^R\right)\right)^2+\left(y^3\left(t^3\right)-y^R\left(t^R\right)\right)^2+\left(z^3\left(t^3\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^3\right)c$$

$$P^{R4}\left(t^R,t^4\right)=\sqrt{\left(x^4\left(t^4\right)-x^R\left(t^R\right)\right)^2+\left(y^4\left(t^4\right)-y^R\left(t^R\right)\right)^2+\left(z^4\left(t^4\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^4\right)c$$

Note, have to keep track of which superscript is an exponent and which is a satellite or receiver identification (later we will also have multiple receivers).

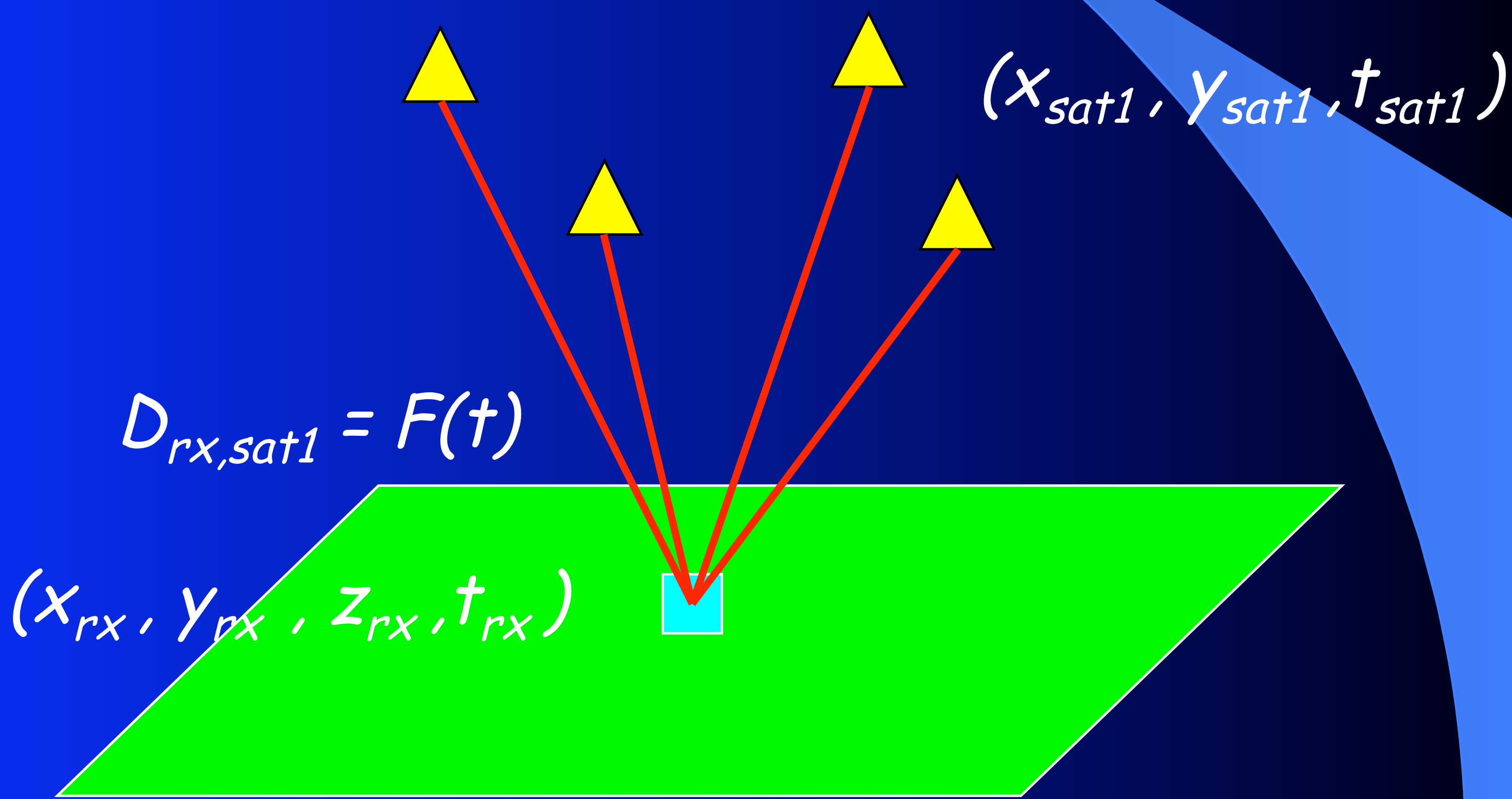
We have 4 unknowns (x^R,y^R,z^R and t^R)
And 4 (nonlinear) equations
(later we will allow more satellites)
So we can solve for the unknowns

Blewitt, Basics of GPS in “Geodetic Applications of GPS”

GPS geometry

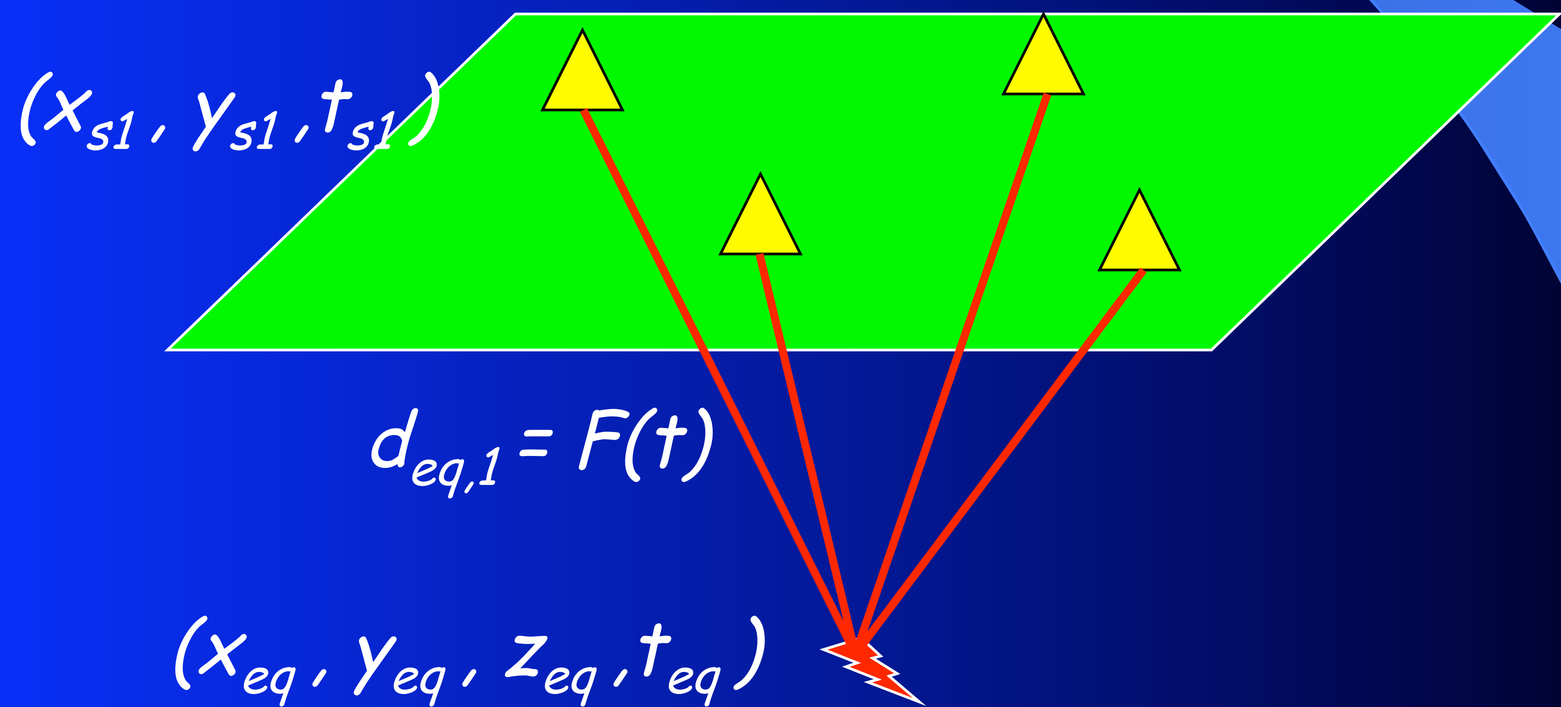
Raypaths (approximately) straight lines.

Really function of travel time (*t*) but can change to psuedo-range.

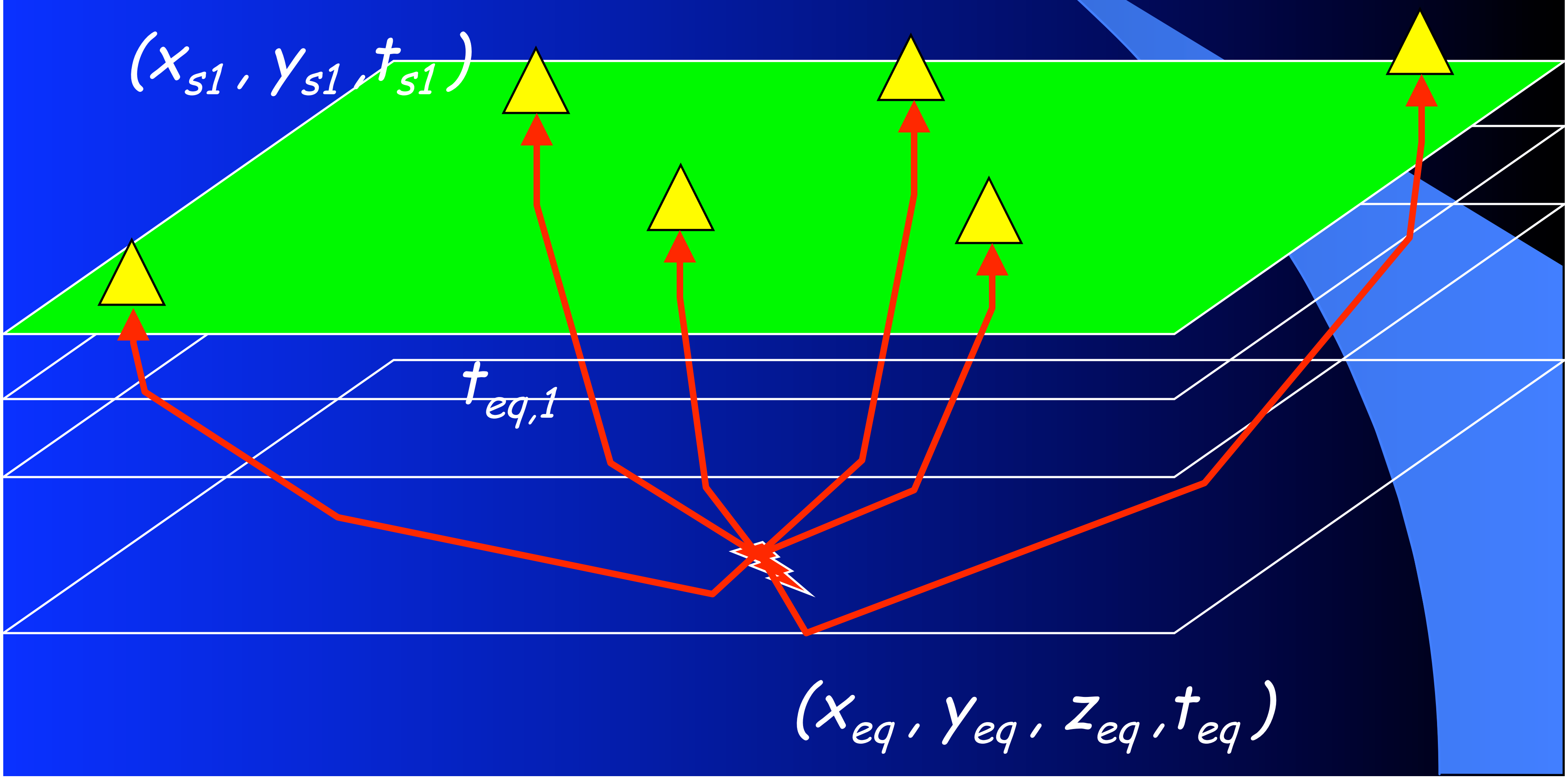


Note that GPS location is almost exactly the same as the earthquake location problem.

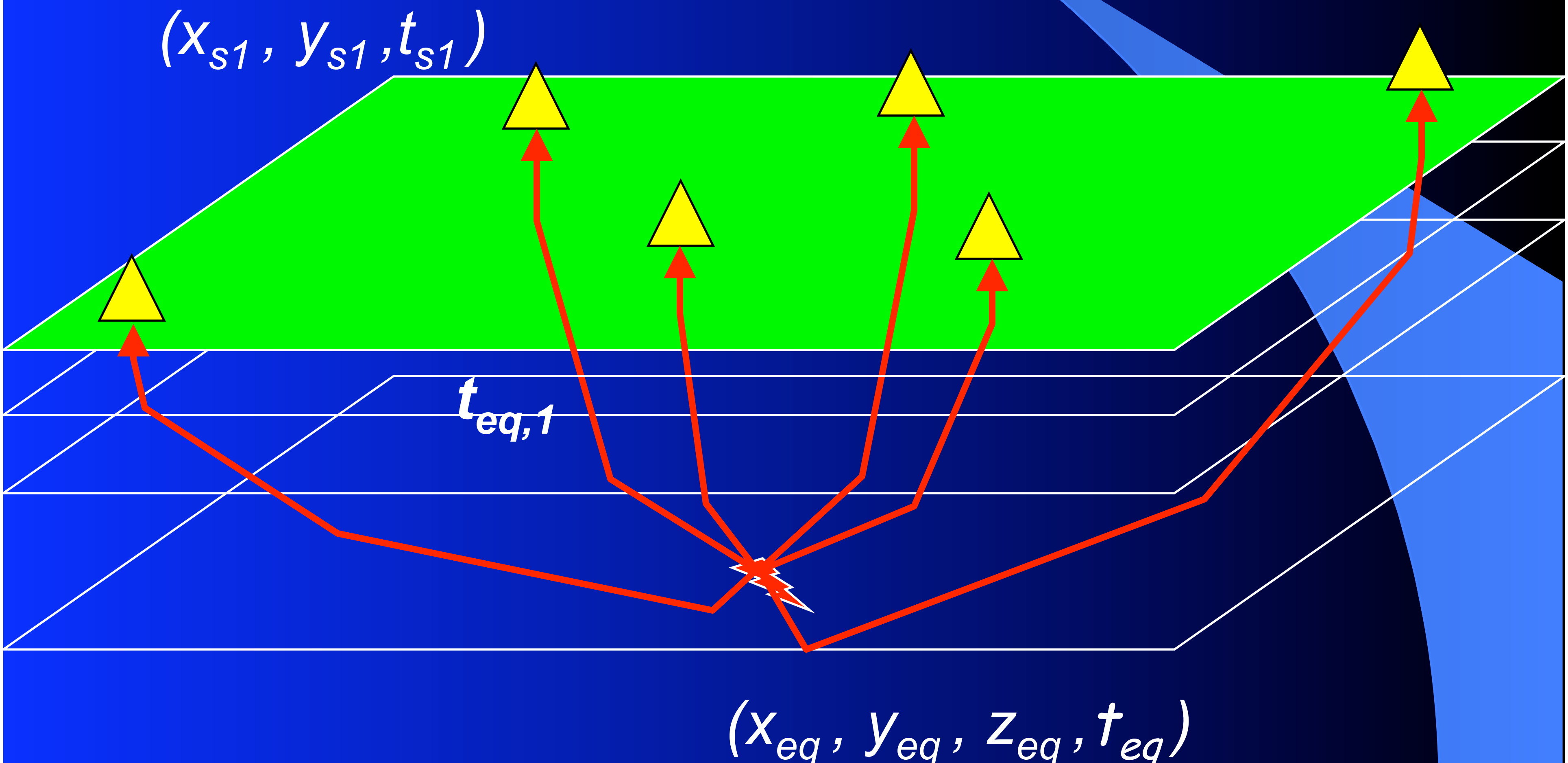
(in a homogeneous half space – raypaths are straight lines, again function of travel time but can also look at distance).



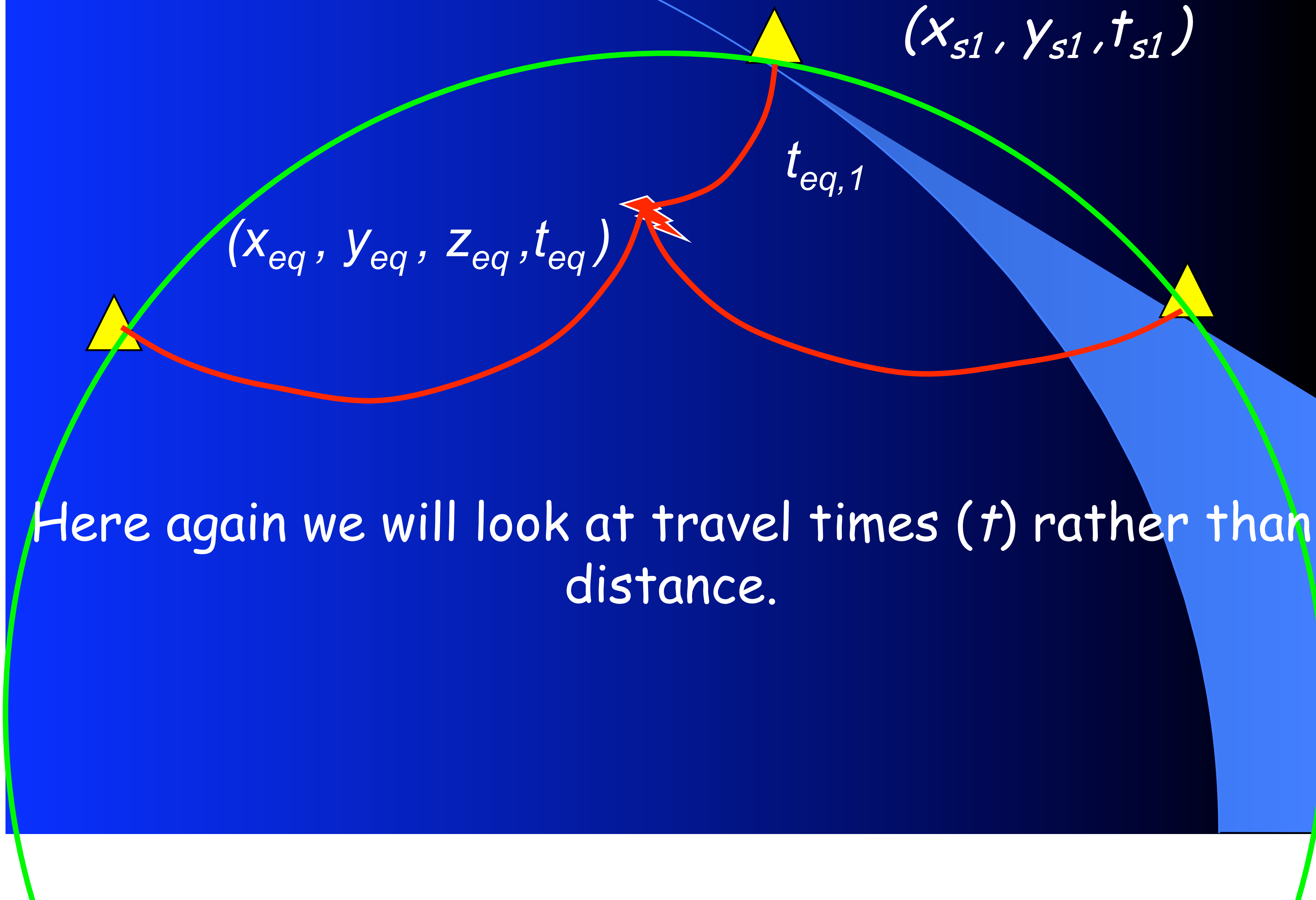
Lets look at more general earthquake location problem of a layered half space.
Raypaths are no longer restricted to straight lines (mix of refracted and head waves shown).
Now look at travel time, not distance (tt related to distance).



This view will help us see a number of problems with locating earthquakes (some of which will also apply to GPS).



This development will also work for a radially symmetric earth.



Here again we will look at travel times (t) rather than distance.

Let *c* be a vector in 4-space giving the location of the earthquake

(3 cartesian coodinates plus time)

$$\vec{\chi} = (x, y, z, t)$$

Let *X* be a vector in 3-space – location of the station

$$\vec{X}_k = (x_k, y_k, z_k)$$

Discussion follows Lee and Stewart

What data/information is available to locate an earthquake?

Arrival time of seismic waves at a number of known locations

$$\tau_{k,\text{observed}}\left(x_k, y_k, z_k\right) = \tau_{k,\text{observed}}\left(\vec{X}_k\right)$$

Plus we have a model for how seismic waves travel in the earth.

This allows us to calculate the travel time to station k

$$T_{k,\text{calculated}}\left(\vec{X}_k,\vec{\chi}\right)$$

(does not really depend on *t*, but carry it along)

from an earthquake at (location and time)

$$\vec{\chi}=\left(x,y,z,t\right)$$

So we can do the forward problem.

From the travel time plus the origin time, *t*

(when the earthquake occurred)

we can calculate the arrival time at the k[th] station

$$\tau_{k,\text{calculated}}\left(\vec{X}_k,\vec{\chi}\right)=T_{k,\text{calculated}}\left(\vec{X}_k,\vec{\chi}\right)+t$$

We want to estimate the 4 parameters of *c*

so we will need 4 data (which gives 4 equations) as a minimum

Unless the travel time – distance relationship is linear

(which it is not in general)

we cannot (easily) solve these 4 equations.

So what do we do?

One possibility is to do the forward calculation for a large number of trial solutions (usually on a grid)

and select the trial solution with the smallest difference between the predicted and measured data

This is known as a grid search (inversion!) and is expensive
(but sometimes it is the only way)

Modifications of this method use ways to cut down on the number of trial solutions

monte carlo

steepest descent

simulated annealing

other

Another approach

solve iteratively by

1) Linearizing the travel time equations

2) Assuming a location

3) Use least squares to compute an adjustment to the location (which depends on the linearization and its derivatives), which we will use to produce a new (better) location

4) Go back to step 2 using location from step 3

We do this till some convergence criteria is met

(if we're lucky)

Assuming a location

For earthquake location the initial location is typically

- the latitude and longitude of the closest station, and

- some predetermined/selected time offset before the first arrival for the origin time and

- a predetermined/selected depth.

Linearization

Choose initial state estimate:

$$\bar{x}_0 = (x_{u0}, y_{u0}, z_{u0}, b_{u0})^T$$

Assume that actual state is given by

$$\bar{x} = \bar{x}_0 + \delta\bar{x}$$

Linearize the pseudorange measurement

$$\bar{\tau} = \bar{f}(\bar{x}) + \bar{\nu}_u \approx \bar{f}(\bar{x}_0) + \left.\frac{\partial \bar{f}}{\partial \bar{x}}\right|_{\bar{x}_0} (\bar{x} - \bar{x}_0) + \bar{\nu}_u$$

which can be rewritten as

$$\bar{\tau} - \bar{f}(\bar{x}_0) = \delta\tau \approx G_{\bar{x}_0}\,\delta\bar{x} + \bar{\nu}_u$$

and then solved for $\delta\bar{x}$ to find the actual state

Pseudoranging

- Solution procedure fairly simple with 4 measurements - can just invert matrix G
- With more than 4 (normal case), must solve a least squares problem - "pseudo-inverse"

$$\delta\overline{\tau} = G\delta\overline{x} + \overline{\nu}_u \quad \Rightarrow \quad \delta\hat{\overline{x}} = (G^T G)^{-1} G^T \delta\overline{\tau}$$

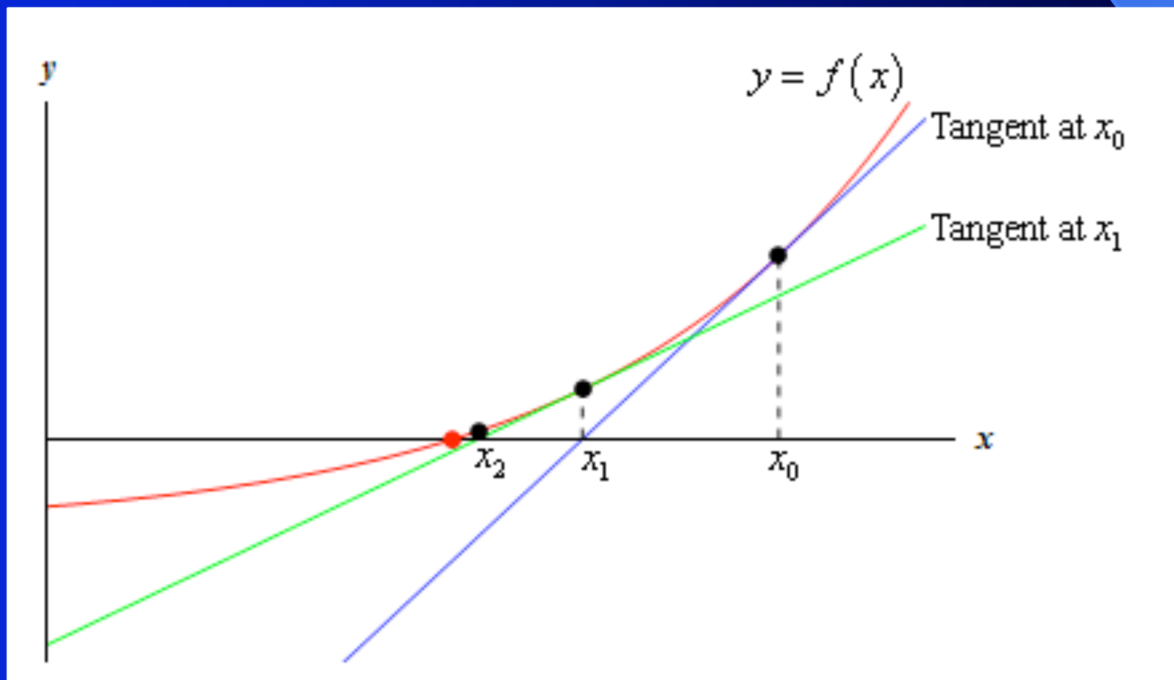
- One complication is that linearization of G depends on our current best estimate of x
  - Which is (hopefully) improving ➔ iteration might be required.

From J. HOW, MIT, GPS-RJH5.pdf

This is basically Newton's method.

If x_n is an approximation to $f(x_n)=0$ and $f'(x_n)\neq 0$, then

$$x_{n+1} \sim x_n - f(x_n)/f'(x_n).$$



Least squares "minimizes" the difference between observed and modeled/calculated data.

Assume a location (time included)

$$\vec{\chi}^{*} = \left(x^{*}, y^{*}, z^{*}, t^{*}\right)$$

and consider the difference between the calculated and measured values

Least squares minimizes the difference between observed and modeled/calculated data.

for one station we have

$$\tau_{\text{observed}} = \tau_{\text{calculated}} + \nu$$

$$\tau_{\text{observed}} = \tau\left(\vec{X}, \vec{\chi}\right) + \nu$$

noise

Did not write calculated here because I can't calculate this without knowing *c*.

First - linearize the expression for the arrival time *t(X,c)*

$$\tau\left(\vec{X},\vec{\chi}\right) \approx \tau_{\text{calculated}}\left(\vec{X},\vec{\chi}^{*}\right) + \left(x - x^{*}\right)\frac{\partial\tau}{\partial x}\bigg|_{\chi^{*}} + \left(y - y^{*}\right)\frac{\partial\tau}{\partial y}\bigg|_{\chi^{*}} + \left(z - z^{*}\right)\frac{\partial\tau}{\partial z}\bigg|_{\chi^{*}} + \left(t - t^{*}\right)\frac{\partial\tau}{\partial t}\bigg|_{\chi^{*}}$$

$$\tau\left(\vec{X},\vec{\chi}\right) \approx \tau_{\text{calculated}}\left(\vec{X},\vec{\chi}^{*}\right) + \frac{\partial\tau}{\partial x}\Delta x + \frac{\partial\tau}{\partial y}\Delta y + \frac{\partial\tau}{\partial z}\Delta z + \frac{\partial\tau}{\partial t}\Delta t$$

Now I can put calculated here because I can calculate this using the known *c**, but I don't know these.

Now consider the difference between the observed and linearized t – the residual

$$\Delta\tau = \tau_{\text{observed}} - \tau_{\text{calculated}}$$

$$\Delta\tau = \tau\left(\vec{X},\vec{\chi}\right) + \nu - \tau_{\text{calculated}}$$

$$\Delta\tau = \left(\tau_{\text{calculated}}\left(\vec{X},\vec{\chi}\right) + \frac{\partial\tau}{\partial x}\Delta x + \frac{\partial\tau}{\partial y}\Delta y + \frac{\partial\tau}{\partial z}\Delta z + \frac{\partial\tau}{\partial t}\Delta t\right) + \nu - \tau_{\text{calculated}}\left(\vec{X},\vec{\chi}\right)$$

$$\Delta\tau = \frac{\partial\tau}{\partial x}\Delta x + \frac{\partial\tau}{\partial y}\Delta y + \frac{\partial\tau}{\partial z}\Delta z + \frac{\partial\tau}{\partial t}\Delta t + \nu$$

We have the following for one station

$$\Delta\tau = \frac{\partial\tau}{\partial x}\Delta x + \frac{\partial\tau}{\partial y}\Delta y + \frac{\partial\tau}{\partial z}\Delta z + \frac{\partial\tau}{\partial t}\Delta t + \nu$$

Which we can recast in matrix form

$$\Delta\tau = \begin{pmatrix} \frac{\partial\tau}{\partial x} & \frac{\partial\tau}{\partial y} & \frac{\partial\tau}{\partial z} & \frac{\partial\tau}{\partial t} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta t \end{pmatrix} + \nu$$

For *m* stations (where *m*≥4)

$$
\begin{pmatrix} \Delta\tau_1 \\ \Delta\tau_2 \\ \Delta\tau_3 \\ \vdots \\ \Delta\tau_m \end{pmatrix} = \begin{pmatrix} \frac{\partial\tau_1}{\partial x} & \frac{\partial\tau_1}{\partial y} & \frac{\partial\tau_1}{\partial z} & \frac{\partial\tau_1}{\partial t} \\ \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_2}{\partial z} & \frac{\partial\tau_2}{\partial t} \\ \frac{\partial\tau_3}{\partial x} & \frac{\partial\tau_3}{\partial y} & \frac{\partial\tau_3}{\partial z} & \frac{\partial\tau_3}{\partial t} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\tau_m}{\partial x} & \frac{\partial\tau_m}{\partial y} & \frac{\partial\tau_m}{\partial z} & \frac{\partial\tau_m}{\partial t} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta t \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_m \end{pmatrix}
$$

Jacobian matrix

Which is usually written as

$$b = Ax + v$$

Evaluating the time term

$$b = Ax + v$$

$$\Delta\vec{\tau} = A\delta\vec{x} + v$$

$$A = \begin{pmatrix} \frac{\partial\tau_1}{\partial x} & \frac{\partial\tau_1}{\partial y} & \frac{\partial\tau_1}{\partial z} & 1 \\ \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_2}{\partial z} & 1 \\ \frac{\partial\tau_3}{\partial x} & \frac{\partial\tau_3}{\partial y} & \frac{\partial\tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\tau_m}{\partial x} & \frac{\partial\tau_m}{\partial y} & \frac{\partial\tau_m}{\partial z} & 1 \end{pmatrix}$$

$$b = Ax + v$$

$$\Delta\vec{\tau} = A\delta\vec{x} + v$$

Expresses linear relationship between residual observations and unknown corrections.

Plus unknown noise terms.

Linearized observation equations

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

Next use least squares to minimize the sum of the squares of the residuals for all the stations.

$$\Delta\vec{\tau} = A\delta\vec{x} + v$$

$$F\left(\chi^*\right) = \sum_{k=1}^{m}\left[\Delta\tau_k\left(\chi^*\right)\right]^2$$

Previous linear least squares discussion gives us

$$A^T\Delta\vec{\tau} = A^T A\delta\vec{x}$$

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

Design matrix – *A*

Coefficients
Partial derivatives of each observation
With respect to each parameter
Evaluated at provisional parameter values

A has 4 columns (for the 4 parameters)
and
As many rows as satellites (need at least 4)

Can calculate derivatives from the model for the observations

This is called Geiger's method

Published 1910

Not used till 1960

(when geophysicists first got hold of a computer)

So far

Have not specified type of arrival.

Can do using P only, S only (?!), P and S together, or S-P.

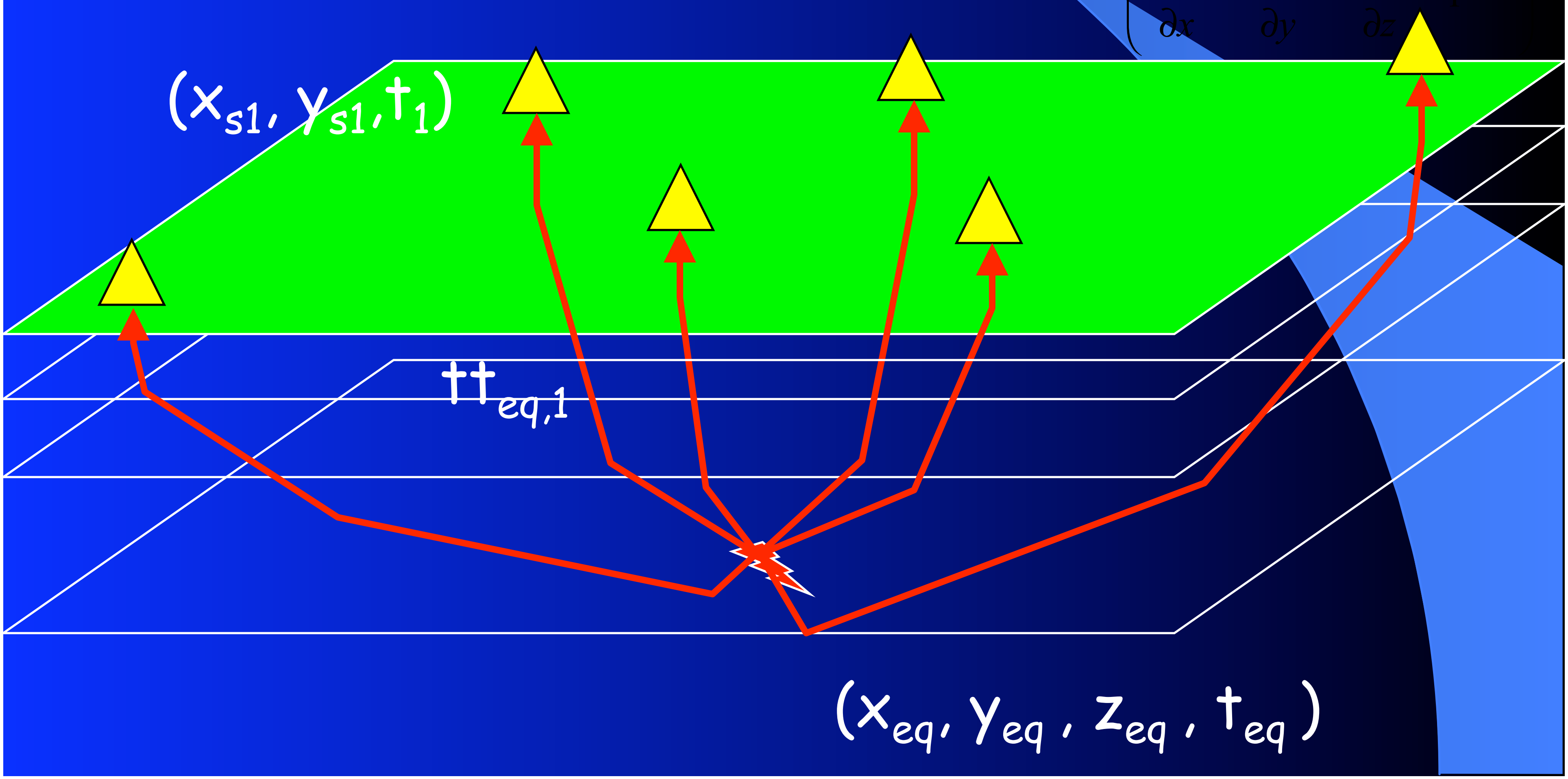
Need velocity model to calculate travel times and travel time derivatives
(so earthquakes are located with respect to the assumed velocity model, not real earth.
Errors are "formal", i.e. with respect to model.)

Velocity models usually laterally homogeneous.

Problems:

Column of 1's - if one of the other columns is constant (or approximately constant) matrix is singular and can't be inverted.

$$A = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & \frac{\partial \tau_1}{\partial z} & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & \frac{\partial \tau_2}{\partial z} & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial \tau_3}{\partial y} & \frac{\partial \tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tau_m}{\partial x} & \frac{\partial \tau_m}{\partial y} & \frac{\partial \tau_m}{\partial z} & 1 \end{pmatrix}$$

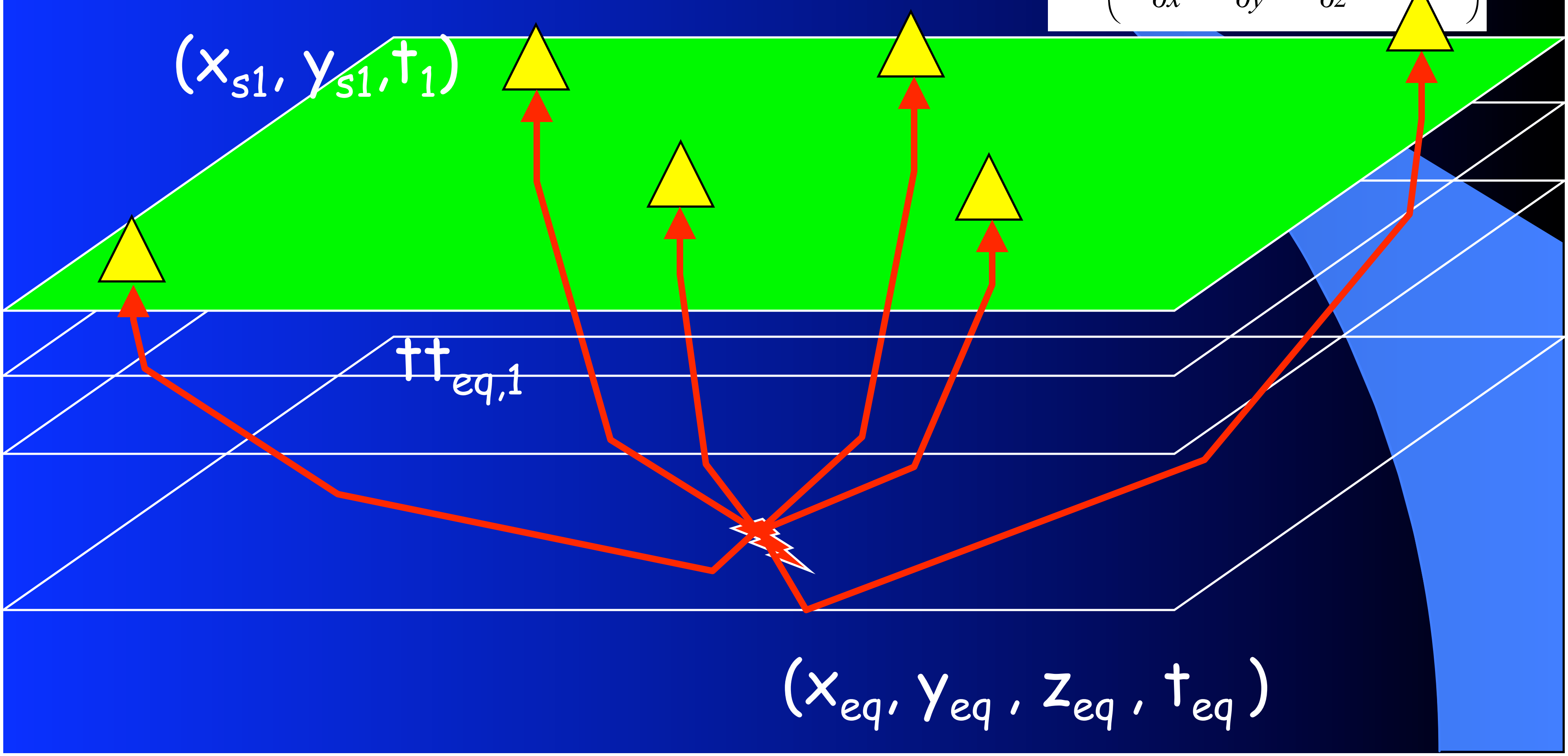


How can this happen:

-All first arrivals are head waves from same refractor (so all have same derivative).

- Earthquake outside the network

$$A = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & \frac{\partial \tau_1}{\partial z} & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & \frac{\partial \tau_2}{\partial z} & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial \tau_3}{\partial y} & \frac{\partial \tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tau_m}{\partial x} & \frac{\partial \tau_m}{\partial y} & \frac{\partial \tau_m}{\partial z} & 1 \end{pmatrix}$$

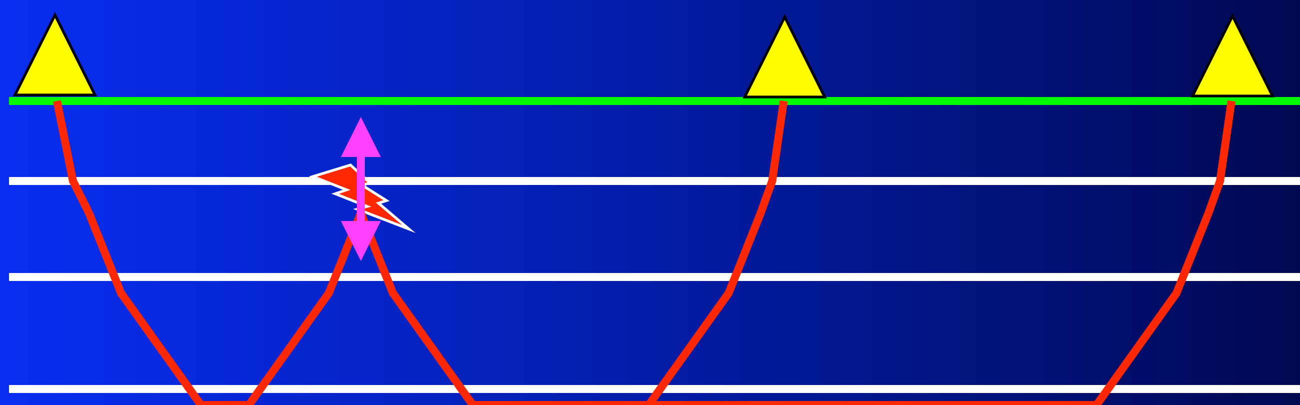


All first arrivals are head waves from same refractor (all have same slope).

$$\frac{\partial \tau_k}{\partial z} = \text{constant} \ \forall \ k$$

$$A = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & \frac{\partial \tau_1}{\partial z} & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & \frac{\partial \tau_2}{\partial z} & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial \tau_3}{\partial y} & \frac{\partial \tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tau_m}{\partial x} & \frac{\partial \tau_m}{\partial y} & \frac{\partial \tau_m}{\partial z} & 1 \end{pmatrix} = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & c & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & c & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial \tau_3}{\partial y} & c & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tau_m}{\partial x} & \frac{\partial \tau_m}{\partial y} & c & 1 \end{pmatrix}$$

In this case we cannot find the depth and origin time independently



Earthquake outside the network

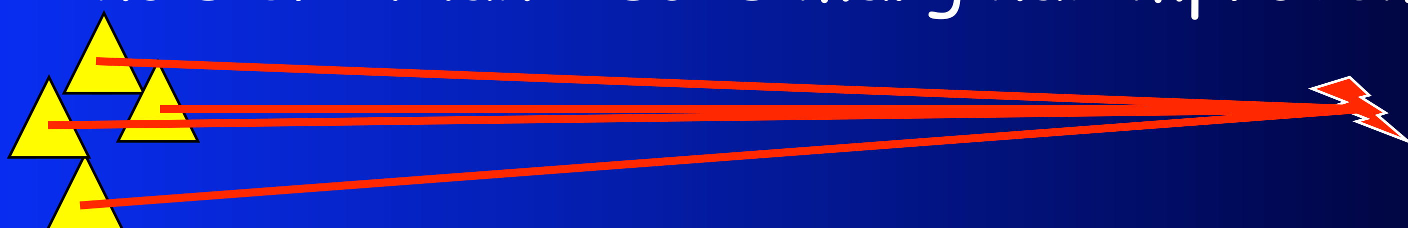
$$\frac{\partial \tau_k}{\partial x} \approx \text{constant } \forall\ k$$

$$\frac{\partial \tau_k}{\partial y} \approx \text{constant } \forall\ k$$

$$A = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & \frac{\partial \tau_1}{\partial z} & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & \frac{\partial \tau_2}{\partial z} & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial \tau_3}{\partial y} & \frac{\partial \tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tau_m}{\partial x} & \frac{\partial \tau_m}{\partial y} & \frac{\partial \tau_m}{\partial z} & 1 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & \frac{\partial \tau_1}{\partial z} & 1 \\ c_1 & c_2 & \frac{\partial \tau_2}{\partial z} & 1 \\ c_1 & c_2 & \frac{\partial \tau_3}{\partial z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ c_1 & c_2 & \frac{\partial \tau_m}{\partial z} & 1 \end{pmatrix}$$

In this case only the azimuth is constrained.

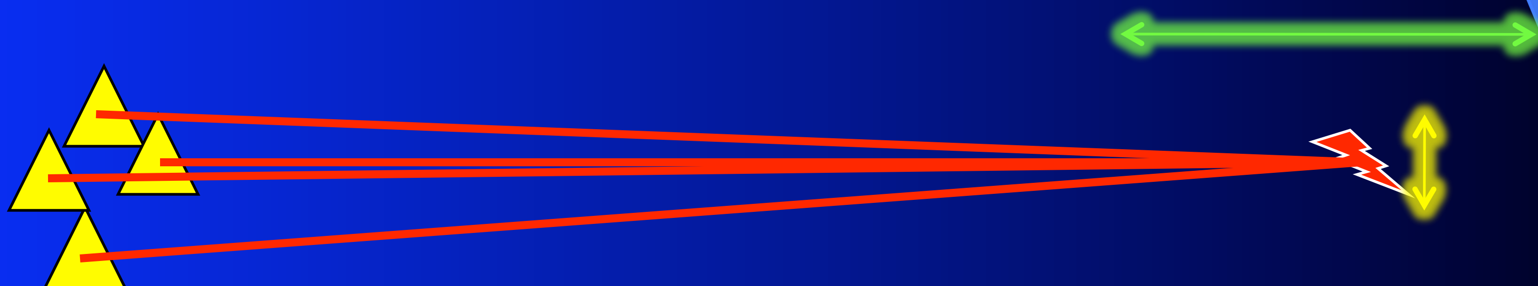
If using both P and S, can also get range, but S "noisier" than P so is marginal improvement.



Probably also suffering from depth-origin time coupling

Similar problems with depth.

Gets worse with addition of noise (changes length of red lines - intersection point moves left/right or up/down [green] much more than in perpendicular direction [yellow].)



Other problems:

Earthquake locations tend to “stick-on” layers in velocity model.

When earthquake crosses a layer boundary, or the depth change causes the first arrival to change from direct to head wave (or vice verse or between different head waves), there is a discontinuity in the travel time derivative (Newton’s method). May move trial location a large distance.

Solution is to “damp” (limit) the size of the adjustments.

Other problems:

Not related to earthquake location, but focal mechanism determination.

Raypath for first arrival from solution may not be actual raypath, especially when first arrival is head wave.
Results in wrong take-off angle.

More inversion pitfalls

Bill and Ted's misadventure.

Bill and Ted are geo-chemists who wish to measure the number of grams of each of three different minerals *A*,*B*,*C* held in a single rock sample.

Let
a be the number of grams of *A*,
b be the number of grams of *B*,
c be the number of grams of *C*
d be the number of grams in the sample.

From Todd Will

By performing complicated experiments Bill and Ted are able to measure four relationships between *a,b,c,d* which they record in the matrix below:

$$\begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \\ 23.2134 & -86.3925 & 44.693 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 34.7177 \\ 70.9241 \\ 82.9271 \\ -26.222 \end{pmatrix}$$

$$Ax = b$$

Now we have more equations than we need

What to do?

From Todd Will

One (relatively unsophisticated) thing to do is throw out one of the equations
(in reality only a Mathematician is naïve enough to think that three equations is sufficient to solve for three unknowns – but lets try it anyway).

So throw out one - leaving

$$\begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 34.7177 \\ 70.9241 \\ 82.9271 \end{pmatrix}$$

$$Ax = b$$

(different *A* and *b* from before)

From Todd Will

Remembering some of their linear algebra the know that the matrix is not invertible if the determinant is zero, so they check that

$$\begin{vmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{vmatrix} \approx -2$$

OK so far

(or "fat, dumb and happy")

From Todd Will

So now we can compute

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{pmatrix}^{-1} \begin{pmatrix} 34.7177 \\ 70.9241 \\ 82.9271 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.8 \\ 0.7 \end{pmatrix}$$

$$x = A^{-}b$$

So now we're done.

From Todd Will

Next they realize that the measurements are really only good to 0.1

So they round to 0.1 and do it again

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{pmatrix}^{-1} \begin{pmatrix} 34.7 \\ 70.9 \\ 82.9 \end{pmatrix} = \begin{pmatrix} -1.68294 \\ 8.92282 \\ -3.50254 \end{pmatrix}$$

$$x = A^{-}b$$

Now they notice a small problem –

They get a very different answer

(and they don't notice they have a bigger problem that they have negative weights/amounts!)

From Todd Will

So what's the problem?

First find the SVD of *A*.

$$A = \begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{pmatrix} =$$

$$A = \left(\vec{h}_1\ \vec{h}_2\ \vec{h}_3\right) \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0.0002 \end{pmatrix} \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{pmatrix}$$

Since there are three non-zero values on the diagonal
A is invertible

From Todd Will

BUT, one of the singular values is much, much less than the others

$$A = \begin{pmatrix} 93.477 & 10.202 & -28.832 \\ 1.93 & 32.816 & 62.414 \\ 26.821 & 36.816 & 57.234 \end{pmatrix} =$$

$$A = \begin{pmatrix} \vec{h}_1 & \vec{h}_2 & \vec{h}_3 \end{pmatrix} \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0.0002 \end{pmatrix} \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{pmatrix}$$

So the matrix is "almost" rank 2

(which would be non-invertible)

From Todd Will

We can also calculate the SVD of A^{-1}

$$A^{-1} = \left(\vec{a}_1 \; \vec{a}_2 \; \vec{a}_3\right) \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 5000 \end{pmatrix} \begin{pmatrix} \vec{h}_1 \\ \vec{h}_2 \\ \vec{h}_3 \end{pmatrix}$$

From Todd Will

So now we can see what happened

(why the two answers were so different)

Let y be the first version of b

Let y' be the second version of b (to 0.1)

$$\left|A^{-1}y - A^{-1}y'\right| = \left|A^{-1}\left(y - y'\right)\right| =$$

$$\left|\begin{pmatrix}\vec{a}_1 & \vec{a}_2 & \vec{a}_3\end{pmatrix}\begin{pmatrix}0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 5000\end{pmatrix}\begin{pmatrix}\vec{h}_1 \\ \vec{h}_2 \\ \vec{h}_3\end{pmatrix}\left(y - y'\right)\right|$$

So A^{-1} stretches vectors parallel to h_3 and a_3 by a factor of 5000.

From Todd Will

Returning to GPS

$$P^{R1}\left(t^R,t^1\right)=\sqrt{\left(x^1\left(t^1\right)-x^R\left(t^R\right)\right)^2+\left(y^1\left(t^1\right)-y^R\left(t^R\right)\right)^2+\left(z^1\left(t^1\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^1\right)c$$

$$P^{R2}\left(t^R,t^2\right)=\sqrt{\left(x^2\left(t^2\right)-x^R\left(t^R\right)\right)^2+\left(y^2\left(t^2\right)-y^R\left(t^R\right)\right)^2+\left(z^2\left(t^2\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^2\right)c$$

$$P^{R3}\left(t^R,t^3\right)=\sqrt{\left(x^3\left(t^3\right)-x^R\left(t^R\right)\right)^2+\left(y^3\left(t^3\right)-y^R\left(t^R\right)\right)^2+\left(z^3\left(t^3\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^3\right)c$$

$$P^{R4}\left(t^R,t^4\right)=\sqrt{\left(x^4\left(t^4\right)-x^R\left(t^R\right)\right)^2+\left(y^4\left(t^4\right)-y^R\left(t^R\right)\right)^2+\left(z^4\left(t^4\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^4\right)c$$

We have 4 unknowns (x^R,y^R,z^R and t^R)

We cannot solve this directly

Will solve interatively by

1) Assuming a location

2) Linearizing the range equations

3) Use least squares to compute new (better) location

4) Go back to 1 using location from 3

We do this till some convergence criteria is met (if we're lucky)

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

Linearize

for one satellite we have

$$P_{\text{observed}} = P_{\text{model}} + \vec{v}$$

$$P_{\text{observed}} = P(x, y, z, \tau) + \vec{v}$$

linearize

$$P(x,y,z,\tau) \approx P(x_0,y_0,z_0,\tau_0) + (x - x_0)\left.\frac{\partial P}{\partial x}\right|_{(x_0,y_0,z_0,\tau_0)}$$

$$+(y - y_0)\left.\frac{\partial P}{\partial y}\right|_{(x_0,y_0,z_0,\tau_0)} + (z - z_0)\left.\frac{\partial P}{\partial z}\right|_{(x_0,y_0,z_0,\tau_0)} + (\tau - \tau_0)\left.\frac{\partial P}{\partial \tau}\right|_{(x_0,y_0,z_0,\tau_0)}$$

$$P(x,y,z,\tau) \approx P_{\text{computed}} + \frac{\partial P}{\partial x}\Delta x + \frac{\partial P}{\partial y}\Delta y + \frac{\partial P}{\partial z}\Delta z + \frac{\partial P}{\partial \tau}\Delta \tau$$

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

Residual
Difference between observed and calculated (linearized)

$$P_{\text{observed}} = P(x, y, z, \tau) + \vec{v}$$

$$\Delta P = P_{\text{observed}} - P_{\text{computed}}$$

$$\Delta P = P(x, y, z, \tau) + \vec{v} - P_{\text{computed}}$$

$$\Delta P = P_{\text{computed}} + \frac{\partial P}{\partial x}\Delta x + \frac{\partial P}{\partial y}\Delta y + \frac{\partial P}{\partial z}\Delta z + \frac{\partial P}{\partial \tau}\Delta \tau + \vec{v} - P_{\text{computed}}$$

$$\Delta P = \frac{\partial P}{\partial x}\Delta x + \frac{\partial P}{\partial y}\Delta y + \frac{\partial P}{\partial z}\Delta z + \frac{\partial P}{\partial \tau}\Delta \tau + \vec{v}$$

So we have the following for one satellite

$$\Delta P = \frac{\partial P}{\partial x}\Delta x + \frac{\partial P}{\partial y}\Delta y + \frac{\partial P}{\partial z}\Delta z + \frac{\partial P}{\partial \tau}\Delta \tau + \vec{v}$$

Which we can recast in matrix form

$$\Delta P = \begin{pmatrix} \frac{\partial P}{\partial x} & \frac{\partial P}{\partial y} & \frac{\partial P}{\partial z} & \frac{\partial P}{\partial \tau} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \tau \end{pmatrix} + \vec{v}$$

Blewitt, Basics of GPS in “Geodetic Applications of GPS”

For *m* satellites (where $m \geq 4$)

$$\begin{pmatrix} \Delta P^1 \\ \Delta P^1 \\ \Delta P^1 \\ \vdots \\ \Delta P^m \end{pmatrix} = \begin{pmatrix} \frac{\partial P^1}{\partial x} & \frac{\partial P^1}{\partial y} & \frac{\partial P^1}{\partial z} & \frac{\partial P^1}{\partial \tau} \\ \frac{\partial P^2}{\partial x} & \frac{\partial P^2}{\partial y} & \frac{\partial P^2}{\partial z} & \frac{\partial P^2}{\partial \tau} \\ \frac{\partial P^3}{\partial x} & \frac{\partial P^3}{\partial y} & \frac{\partial P^3}{\partial z} & \frac{\partial P^3}{\partial \tau} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial P^m}{\partial x} & \frac{\partial P^m}{\partial y} & \frac{\partial P^m}{\partial z} & \frac{\partial P^m}{\partial \tau} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \tau \end{pmatrix} + \begin{pmatrix} v^1 \\ v^2 \\ v^3 \\ \vdots \\ v^n \end{pmatrix}$$

Which is usually written as

$$\vec{b} = A\vec{x} + \vec{v}$$

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

Calculate the derivatives

$$\frac{\partial P^{RS}}{\partial x^R} = \sqrt{\left(x^S\left(t^S\right)-x^R\left(t^R\right)\right)^2+\left(y^S\left(t^S\right)-y^R\left(t^R\right)\right)^2+\left(z^S\left(t^S\right)-z^R\left(t^R\right)\right)^2}+\left(\tau^R-\tau^1\right)c$$

$$\frac{\partial P^{RS}}{\partial x^R} = \frac{\left(\left(-1\right)\left(\frac{1}{2}\right)2\right)\left(x^S\left(t^S\right)-x^R\left(t^R\right)\right)}{\sqrt{\left(x^S\left(t^S\right)-x^R\left(t^R\right)\right)^2+\left(y^S\left(t^S\right)-y^R\left(t^R\right)\right)^2+\left(z^S\left(t^S\right)-z^R\left(t^R\right)\right)^2}}$$

$$\frac{\partial P^{RS}}{\partial x^R} = \frac{\left(x^R\left(t^R\right)-x^S\left(t^S\right)\right)}{\rho^R}$$, similarly for y and z

$$\frac{\partial P^{RS}}{\partial \tau^R} = c$$

So we get

$$A = \begin{pmatrix} \frac{x_0 - x^1}{\rho^1} & \frac{y_0 - y^1}{\rho^1} & \frac{z_0 - z^1}{\rho^1} & c \\ \frac{x_0 - x^2}{\rho_2} & \frac{y_0 - y^2}{\rho_2} & \frac{z_0 - z^2}{\rho_2} & c \\ \frac{x_0 - x^3}{\rho_3} & \frac{y_0 - y^3}{\rho_3} & \frac{z_0 - z^3}{\rho_3} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_0 - x^m}{\rho_m} & \frac{y_0 - y^m}{\rho_m} & \frac{z_0 - z^m}{\rho_m} & c \end{pmatrix}$$

Is function of direction to satellite

Note last column is a constant

Consider some candidate solution *x'*

Then we can write

$$\vec{v} = b - A\vec{x}'$$

Where x' is the old candidate solution

Ax' is the new candidate solution

b are the observations

n hat are the residuals

We would like to find the *x'* that minimizes the *n hat*

Blewitt, Basics of GPS in "Geodetic Applications of GPS"

So the question now is how to find this *x*′

One way, and the way we will do it,

Least Squares

Since we have already done this – we'll go fast

Use solution to linearized form of observation equations to write estimated residuals

$$\hat{v} = \vec{b} - A\hat{x}'$$

Vary value of x to minimize

$$J(\vec{x}) = \sum_{i=1}^{m} \vec{v}_i^2 = \vec{v}^T \vec{v} = (\vec{b} - A\vec{x})^T (\vec{b} - A\vec{x})$$

$$\delta J(\hat{x}) = 0$$

$$\delta\left(\left(\vec{b} - A\hat{x}\right)^T \left(\vec{b} - A\hat{x}\right)\right) = 0$$

$$\left\{\delta\left(\vec{b} - A\hat{x}\right)^T\right\}\left(\vec{b} - A\hat{x}\right) + \left(\vec{b} - A\hat{x}\right)^T \left\{\delta\left(\vec{b} - A\hat{x}\right)\right\} = 0$$

$$\left(-A\delta\,\hat{x}\right)^T \left(\vec{b} - A\hat{x}\right) + \left(\vec{b} - A\hat{x}\right)^T \left(-A\delta\,\hat{x}\right) = 0$$

$$\left(-A\delta\,\hat{x}\right)^T \left(\vec{b} - A\hat{x}\right) + \left(\vec{b} - A\hat{x}\right)^T \left(-A\delta\,\hat{x}\right) = 0$$

$$\left(A\delta\,\hat{x}\right)^T \left(\vec{b} - A\hat{x}\right) = 0$$

$$\left(\delta\,\hat{x}^T A^T\right)\left(\vec{b} - A\hat{x}\right) = 0$$

$$\delta\,\hat{x}^T \left(A^T\vec{b} - A^T A\hat{x}\right) = 0$$

$$A^T\vec{b} = A^T A\hat{x}$$

$$\hat{x} = \left(A^T A\right)^{-1} A^T\vec{b}$$

Solution to normal equations

$$\hat{x} = \left(A^T A\right)^{-1} A^T \vec{b}$$

Assumes

Inverse exists

(Generalized inverse always exists – solution may not make sense but exists.)

(m greater than or equal to 4, necessary but not sufficient condition)

Can have problems similar to earthquake locating (two satellites in "same" direction for example – has effect of reducing rank by one)

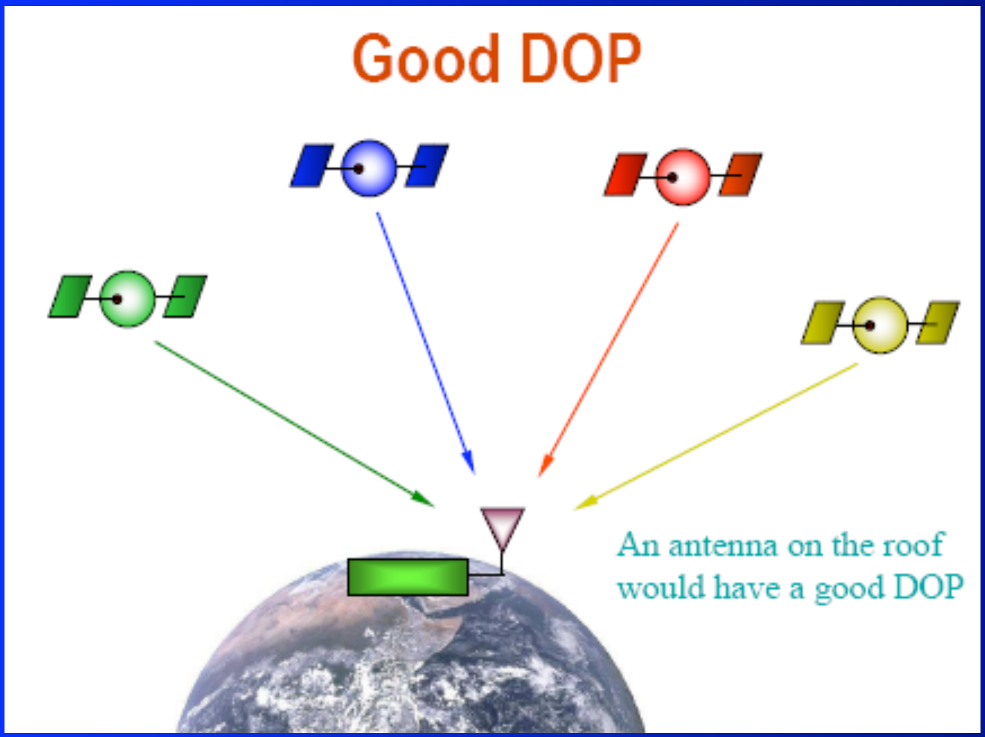
$$\hat{x} = \left(A^T A\right)^{-1} A^T \vec{b}$$

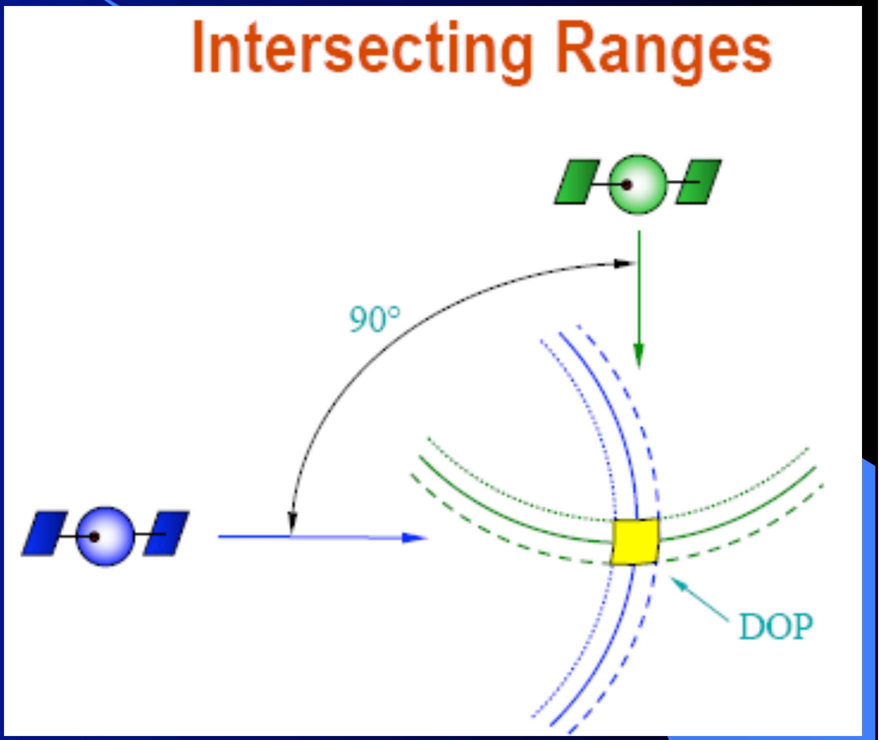
Assumes

Solution is unique (i.e. earthquake or GPS is in one place. Can see for flat earth example – there are actually two solutions, one above and one below the plane.

But – no guarantee mathematically, and may be a number of local minimums into which one can get "stuck"

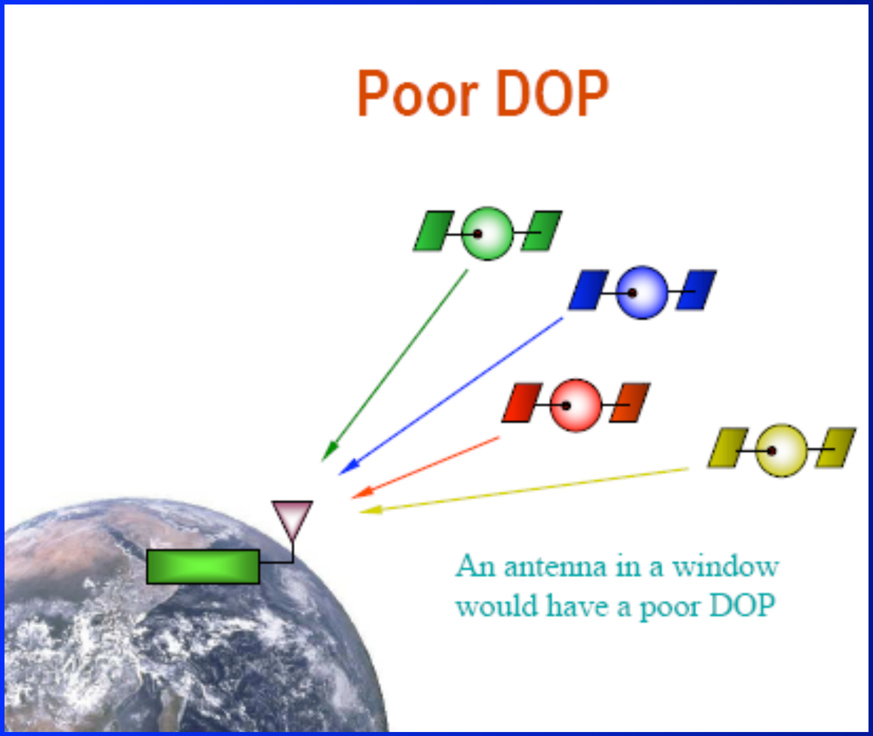
GPS tutorial Signals and Data

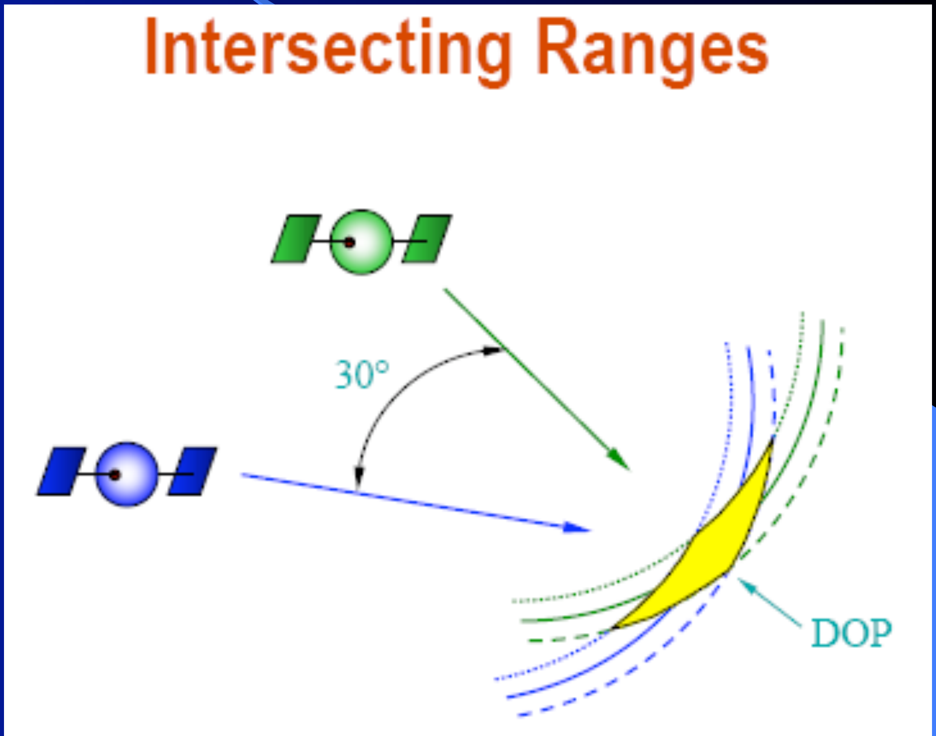




http://www.unav-micro.com/about_gps.htm

GPS tutorial Signals and Data





http://www.unav-micro.com/about_gps.htm