Earth Science Applications of Space Based Geodesy

DES-7355

Tu-Th 9:40-11:05

Seminar Room in 3892 Central Ave. (Long building)

Bob Smalley
Office: 3892 Central Ave, Room 103
678-4929
Office Hours – Wed 14:00-16:00 or if I' m in my office.

http://www.ceri.memphis.edu/people/smalley/ESCI7355/ESCI_7355_Applications_of_Space_Based_Geodesy.html

Class 6

WAAS

It provides enhanced integrity, accuracy, availability, and continuity over and above GPS SPS.

The differential correction function provides improved accuracy required for "precision" instrument approaches for aircraft.

SBAS

Satellite-based augmentation systems

use satellites and networks of ground stations to provide improved accuracy for received GPS satellite signals.

SBAS

Internationally, many countries are working with the International Civil Aviation Organization (ICAO) to standardize satellite-based augmentation systems (SBAS) globally.

WAAS is an SBAS currently being implemented in the United States

How WAAS provides improved accuracy.

The Wide Area Augmentation System (WAAS) uses a network of ground stations to provide necessary corrections to received GPS SPS navigation signals.

Precisely surveyed ground reference stations are strategically positioned across the country, including in Alaska, Hawaii, and Puerto Rico, to collect GPS satellite data.

How WAAS provides improved accuracy.

Using this information, a message is developed to correct any signal errors.

These correction messages are then broadcast on the same frequency as GPS signals by communications satellites to GPS receivers (on board aircraft, but really most new GPS receivers).

How WAAS provides improved accuracy.

WAAS is designed to provide the additional accuracy, availability, and integrity necessary to enable users to rely on GPS for all phases of flight, from enroute through approach for all qualified airports within the WAAS coverage area.

How WAAS provides improved accuracy.

WAAS supplies two different sets of corrections:

How WAAS provides improved accuracy.

1: corrected GPS parameters (position, clock, etc.)

How WAAS provides improved accuracy.

2: Ionospheric parameters. The second set of corrections is user position independent (i.e., they apply to all users located within the WAAS service area).

How WAAS provides improved accuracy.

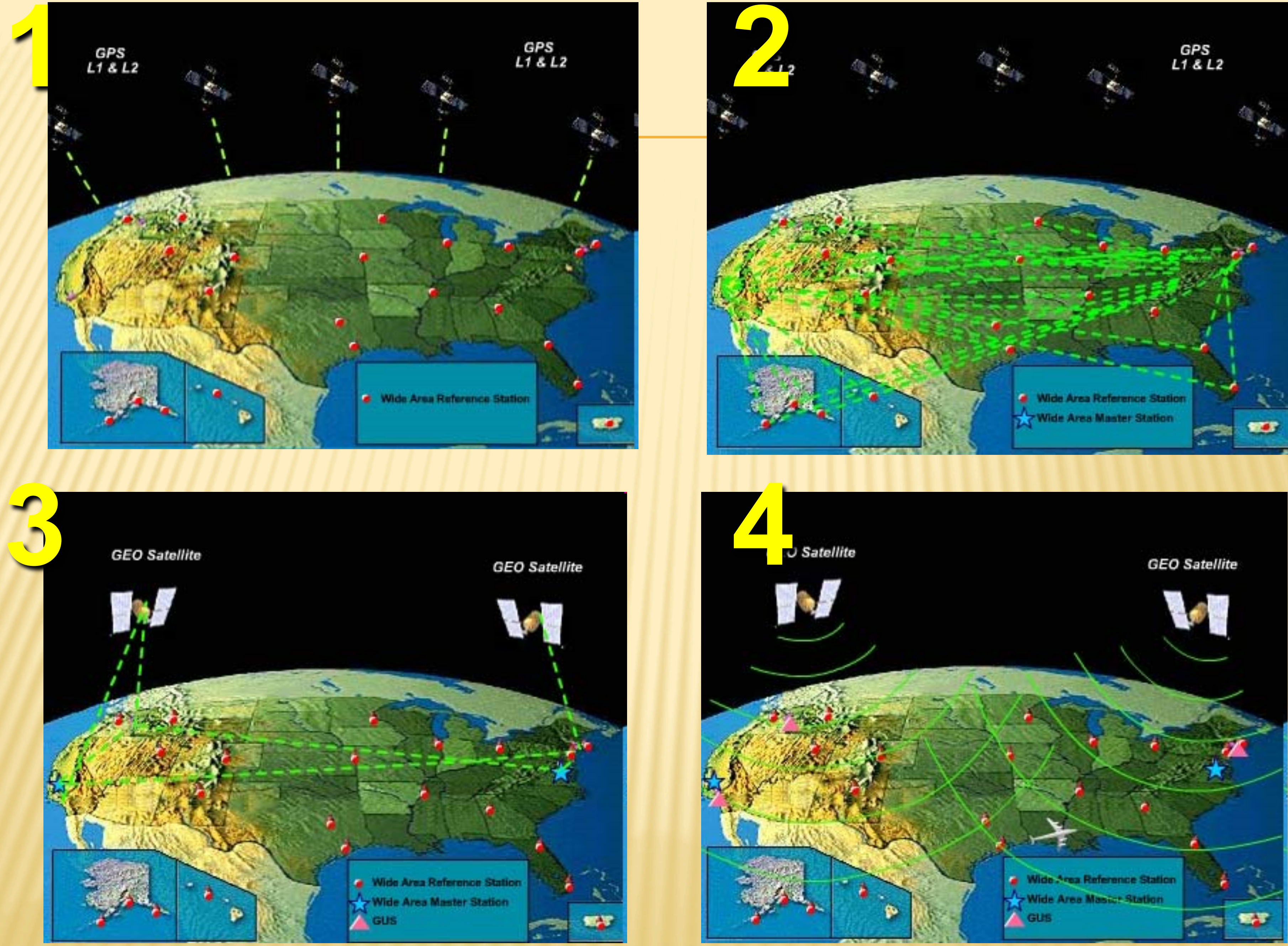
The second set of corrections is area specific. WAAS supplies correction parameters for a number of points (organized in a grid pattern) across the WAAS service area.

The user receiver computes ionospheric corrections for the received GPS signals based on algorithms which use appropriate grid points for the user location.

How WAAS provides improved accuracy.

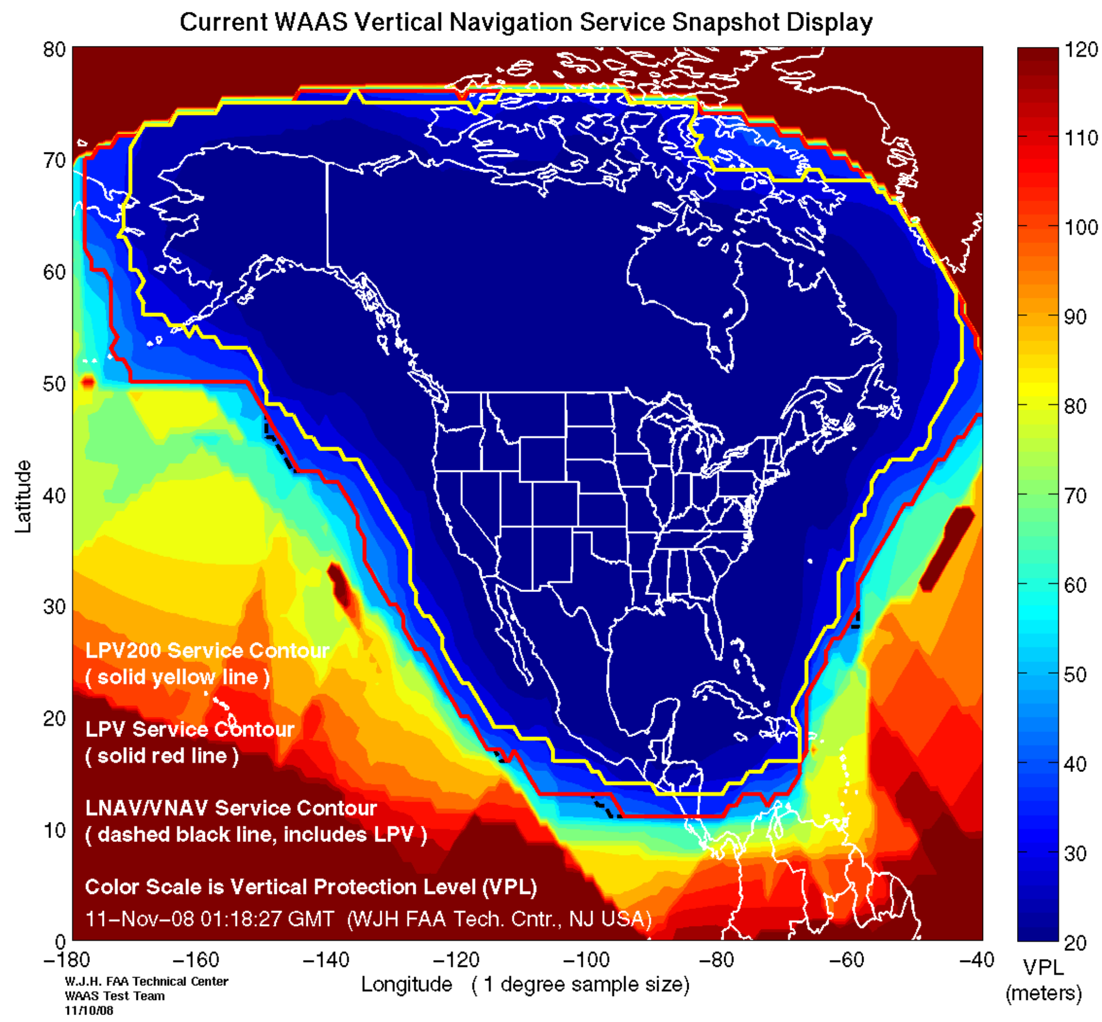
Furthermore, the appropriate grid points may differ for each GPS satellite signal received and processed by the user receiver, since GPS satellites are located at various positions in the sky relative to the user.

The combination of these two sets of corrections allows for significantly increased user position accuracy and confidence anywhere in the WAAS service area.



http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/waas/howitworks/

LPV minimums available 98%+ of time in continental NA



www.nstb.tc.faa.gov/RT_VerticalProtectionLevel.htm

Web site updates every 3 minutes

Other SBAS systems

EGNOS

European Geostationary Navigation Overlay Service

MSAS

Multi-Functional Satellite Augmentation System

(Japan)

Miraculously they are all (supposed to be) compatible!

(only WAAS functional)

WAAS not much help for high-precision, dual frequency, GPS.

(gives better starting estimate, but once station is processed one time have starting estimate to cm's.)

Carrier Phase Tracking

Used in high-precision survey work

Can generate sub-centimeter accuracy

The ~20 cm carrier is tracked by a reference receiver and a remote (user) receiver

The carrier is not subject to S/A and is a much more precise measurement than pseudoranges.

Carrier Phase Tracking

Requires bookeeping of cycles: subject to "slips" (loss of "lock" by the phase locked loop tracking each satellite)

Ionospheric delay differences must be small enough to prevent full slips

Requires remote receiver be within ~30km of base (for single frequency, short occupations)

Usually used in post-processed mode, but RealTime Kinematic (RTK) method is developing

A. Ganse, U. Washington , http://staff.washington.edu/aganse/

Receivers

Basic 12 channel receivers start at $100

Usually includes track & waypoint entry

With built-in maps start at $150

Combination GPS receiver/cell phone ~$350

"Outdoor" have barometric altimeter and electronic (flux gate) compass (hiking, camping, bicycling, flying w/o aviation database, GPS fieldwork, ...)
~$100 to $500

Receivers

Survey-quality: $1000 and up

Carrier tracking

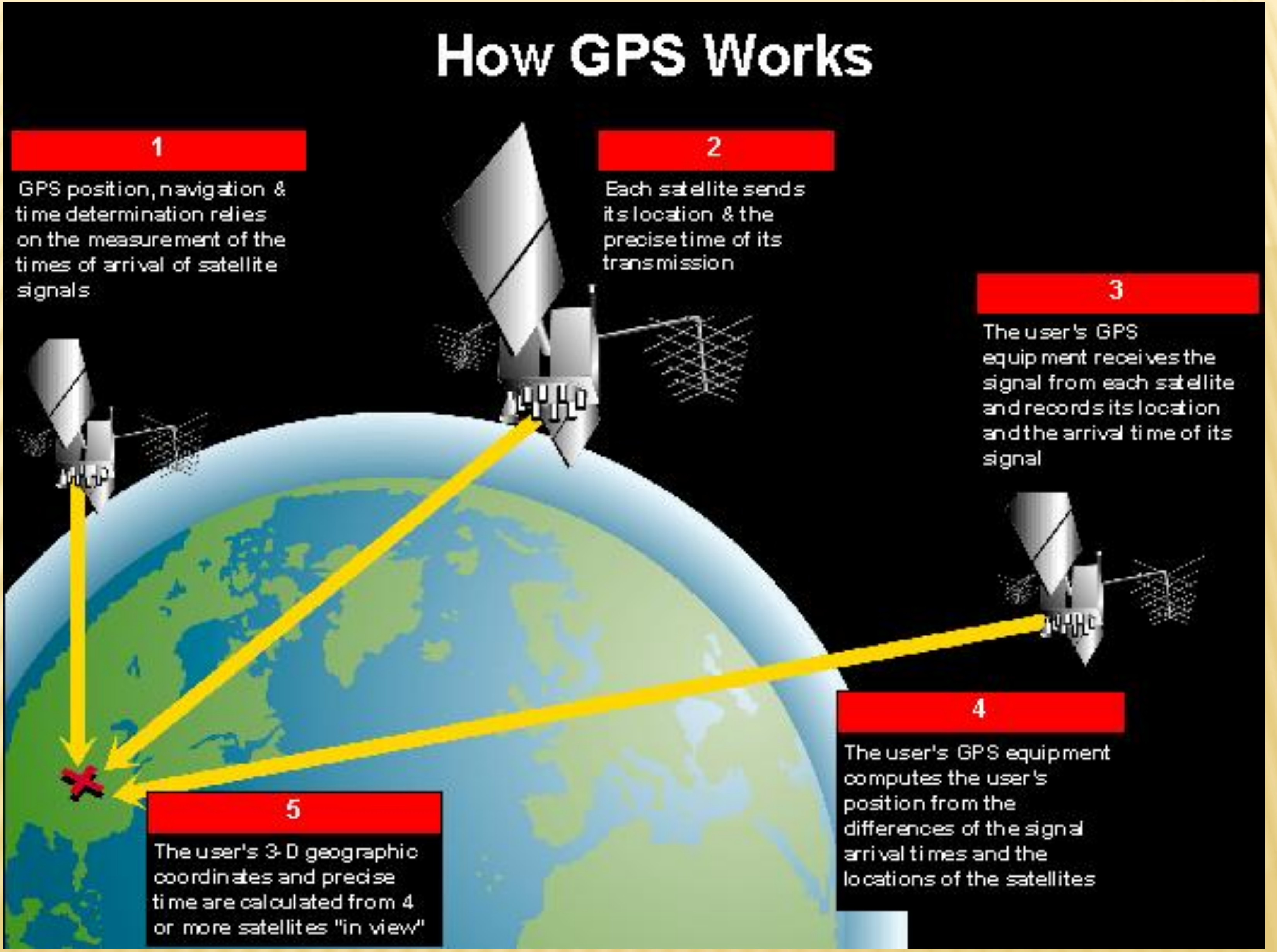
~~FM receiver for differential corrections (old – out of date)~~

WAAS

~~RS232 port to PC for realtime or post-processing (old – out of date)~~

Internet/web page interface/USB/bluetooth.

Point positioning with Psuedorange from code


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"

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

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

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:

P_i = Measured PseudoRange to the i^th SV

X_i , Y_i , Z_i = Position of the i^th SV, Cartesian Coordinates

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

b = User clock bias (in distance units), to be solved-for

The above nonlinear equations are solved iteratively using an initial estimate of the user position, XYZ, and b

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

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

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

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

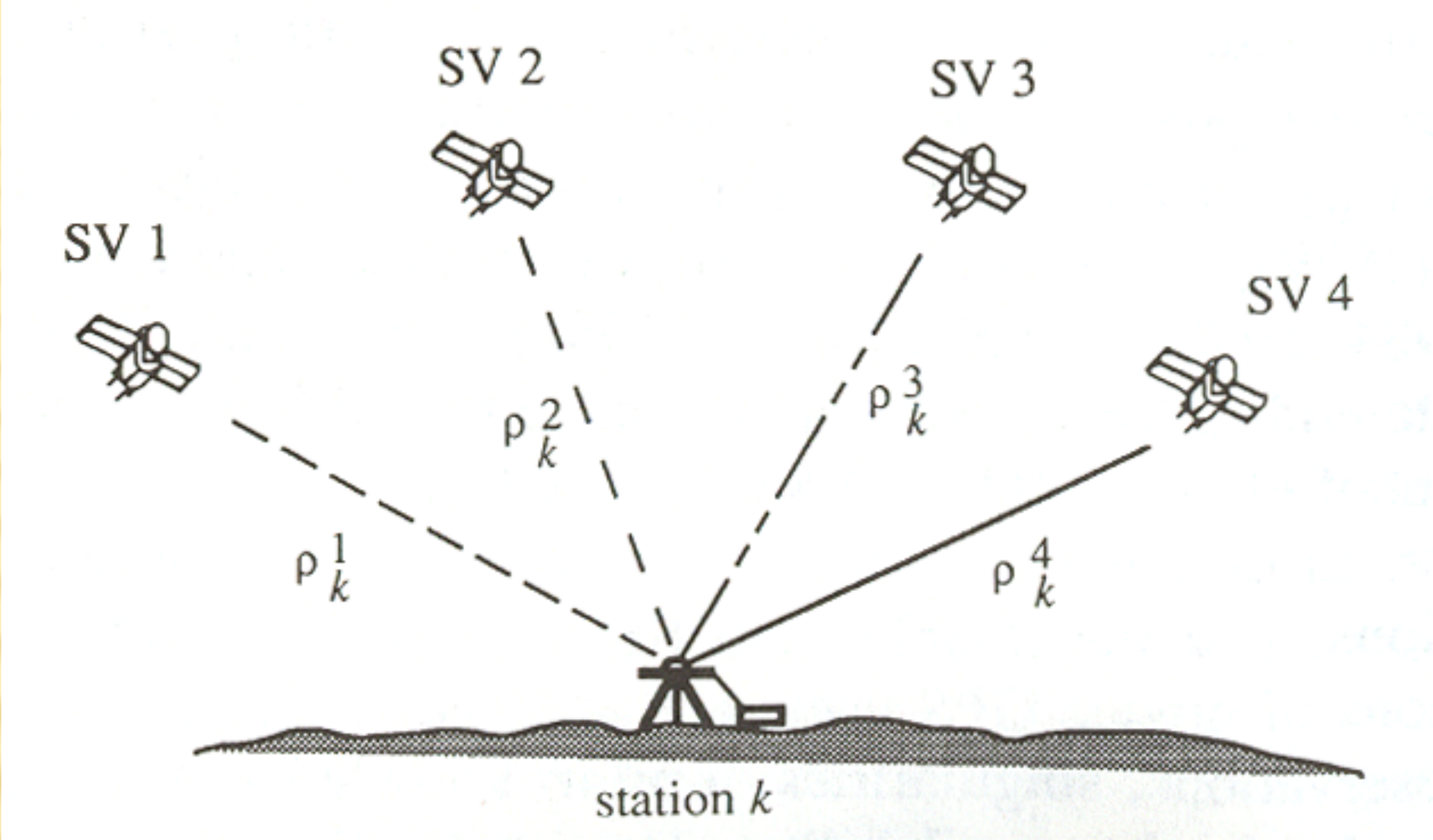
and then solved for $\delta\bar{x}$ to find the actual state

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"

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

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



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

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

Pseudo range – 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 τ^S known from satellite navigation message

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

Assume c constant along path, ignore relativity.

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

Complicating detail, satellite position has to be calculated at transmission 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 to less than a 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 exponent and which is satellite or receiver (later we will have multiple receivers also) identification.

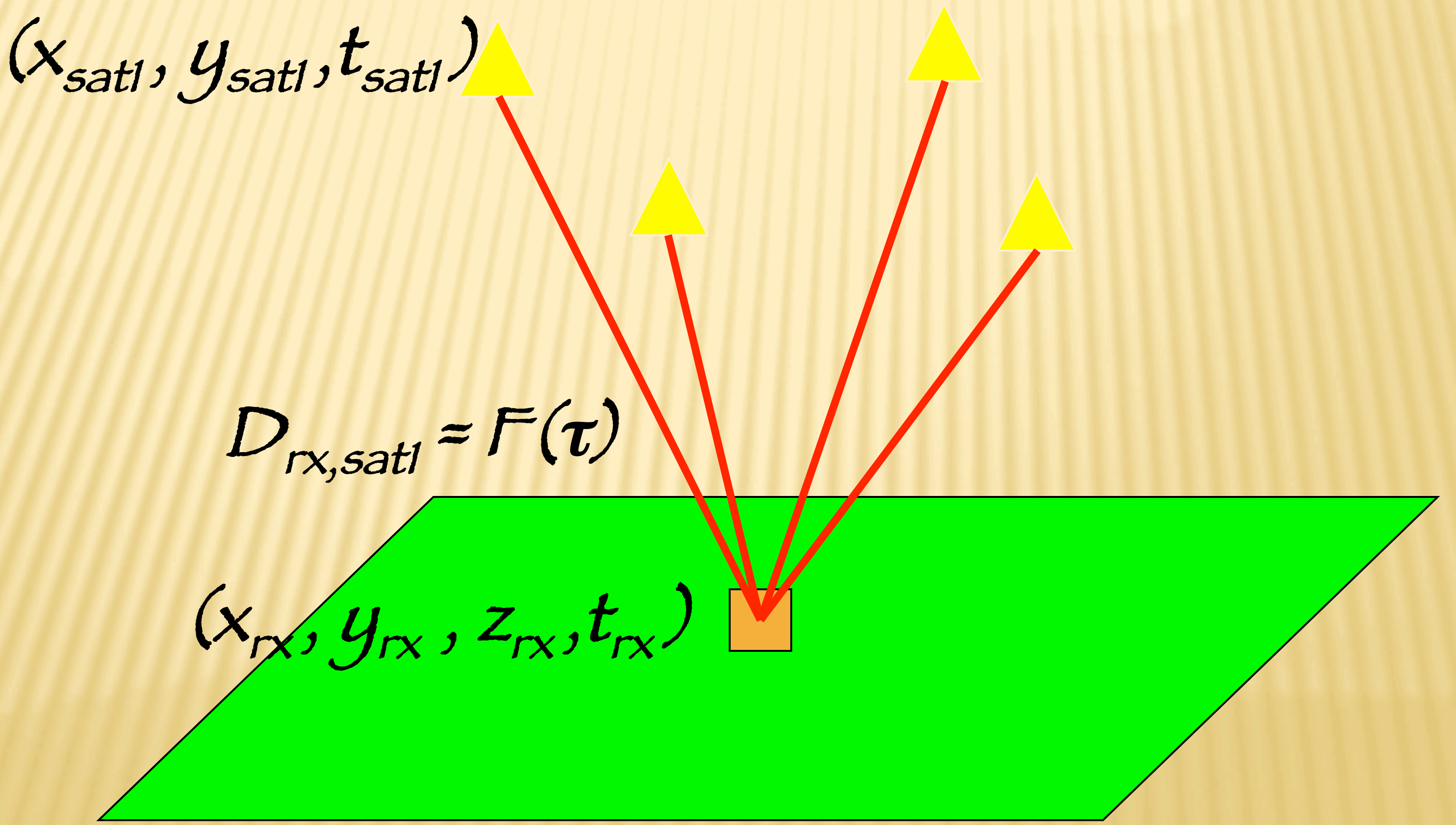
We have 4 unknowns (x^R, y^R, z^R and τ^R)
And 4 (nonlinear) equations
(later we will allow more satellites)
So we can solve for the unknowns

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

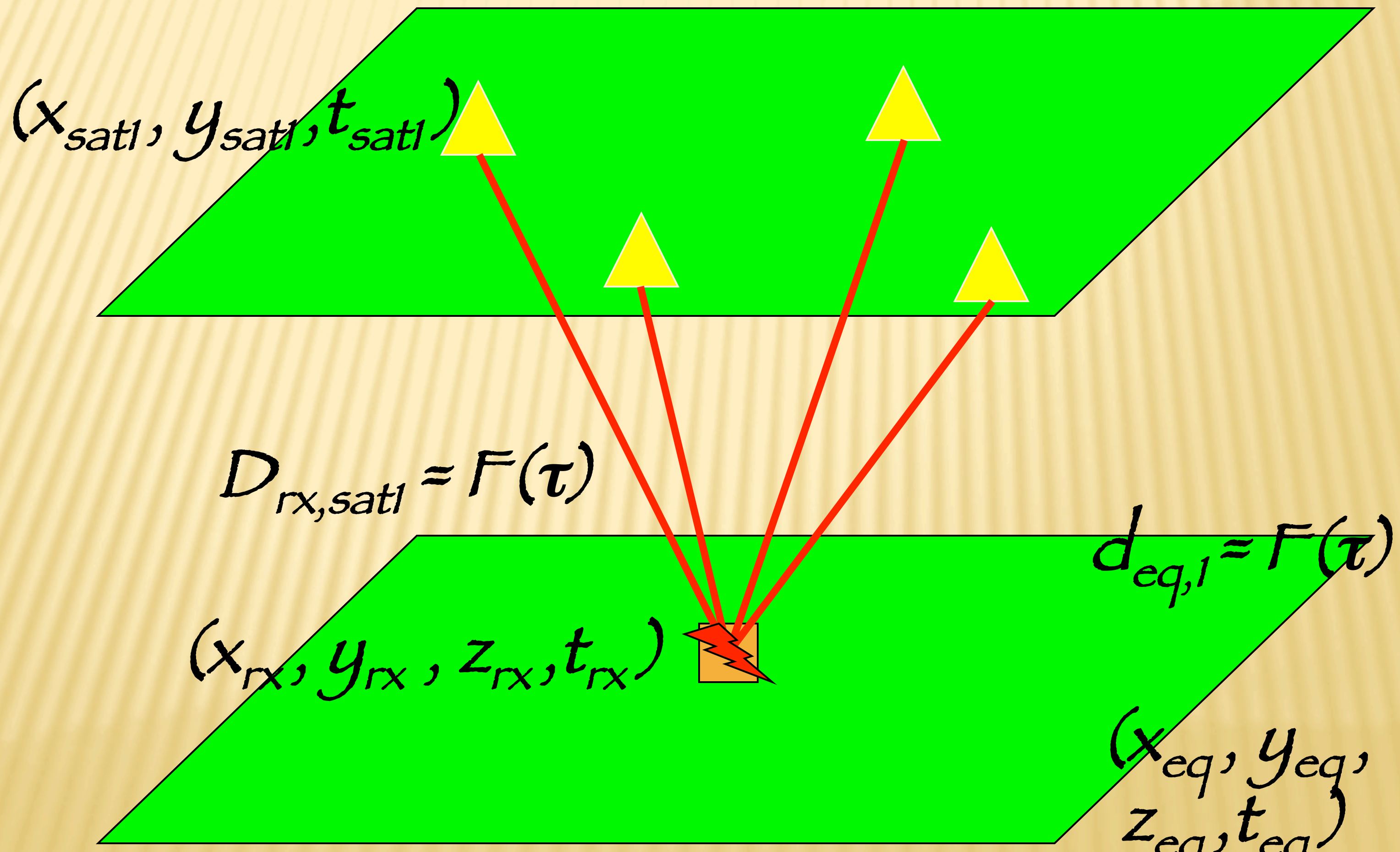
GPS geometry

Raypaths (approximately) straight lines.

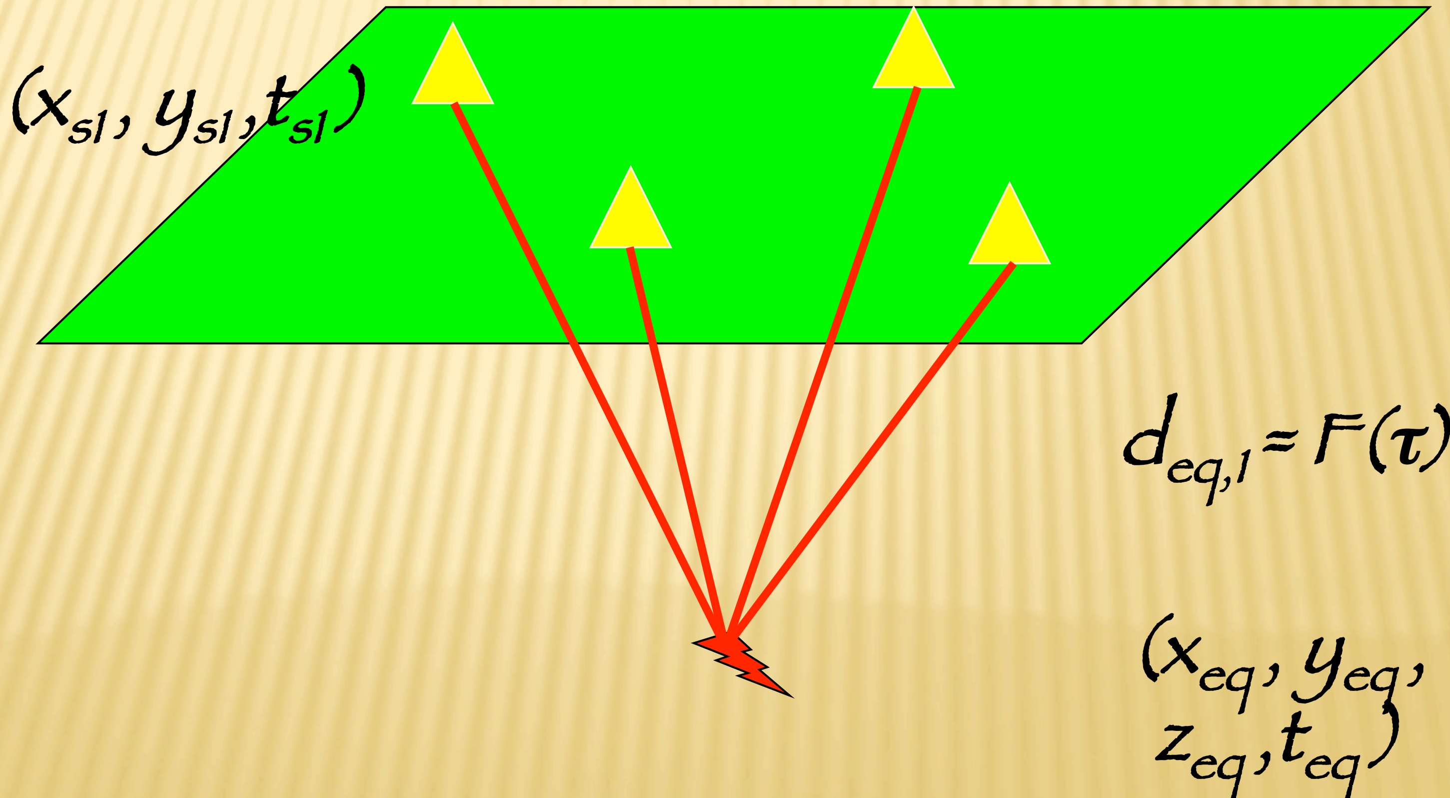
Really function of travel time (τ) but can change to psuedo-range.



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



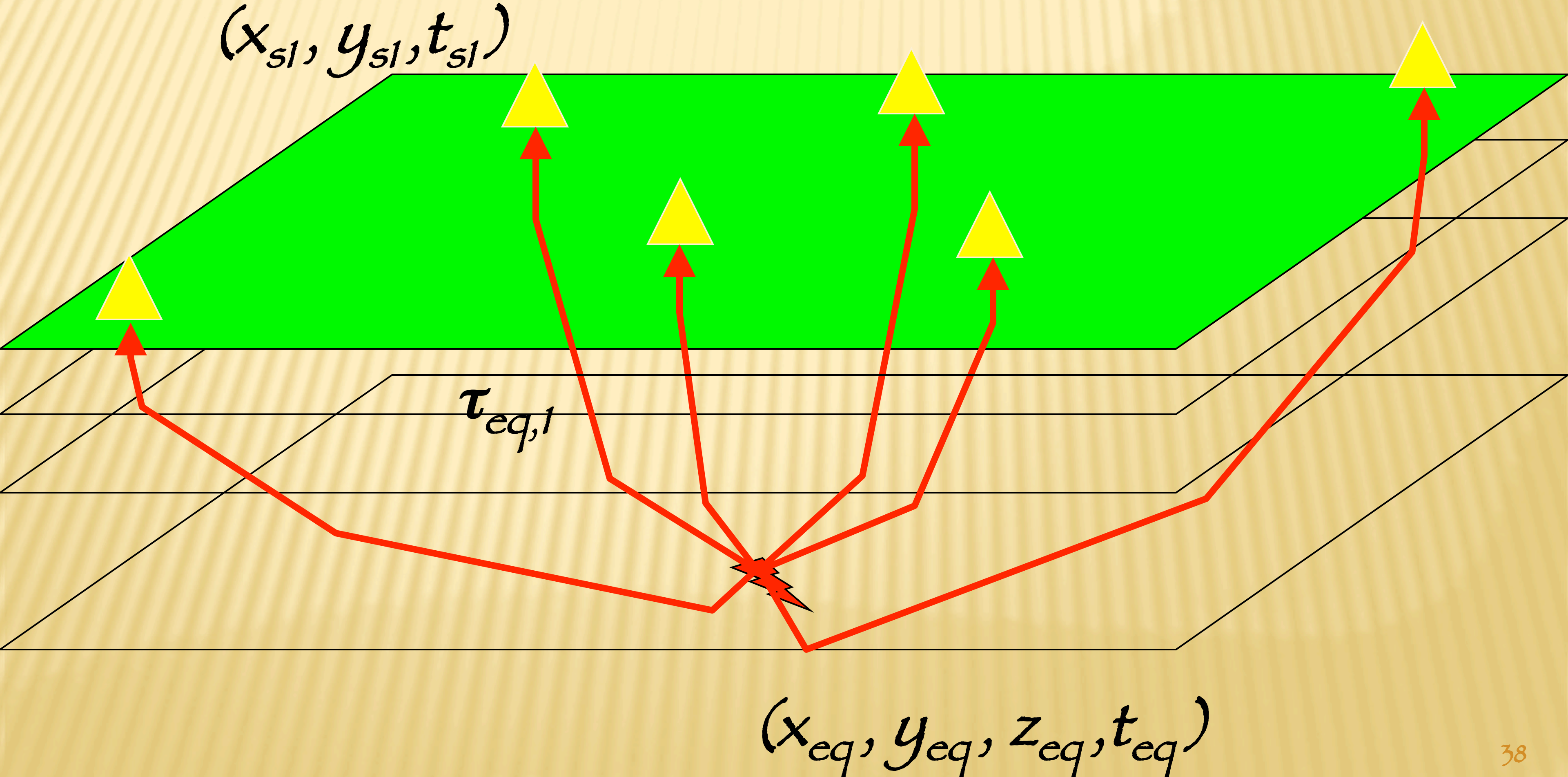
In a homogeneous half space – raypaths are straight lines, again function of travel time but can also look at distance.



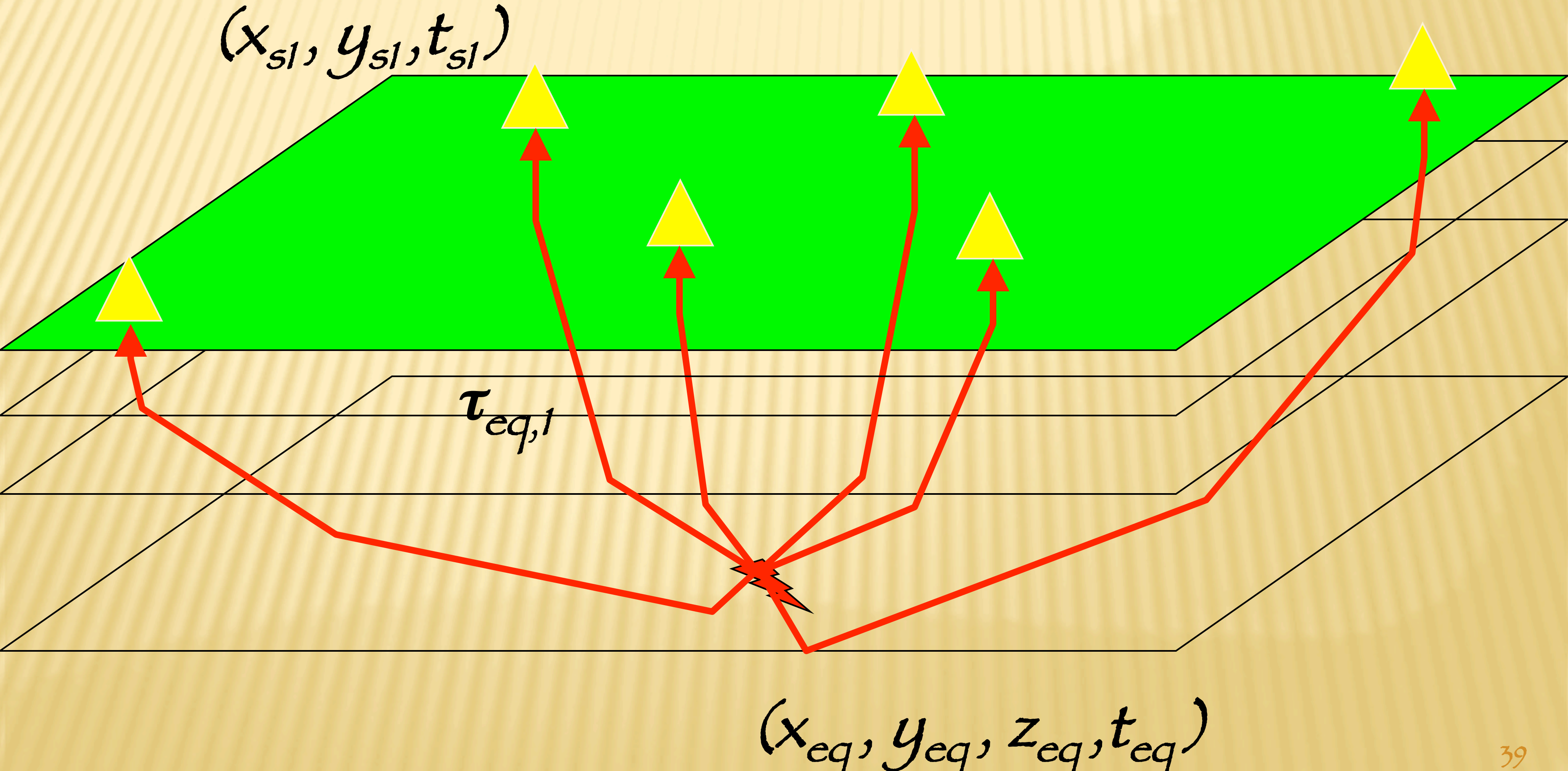
Lets look at more general problem of a layered half space.

Raypaths are now not limited to straight lines (mix of refracted and head waves shown).

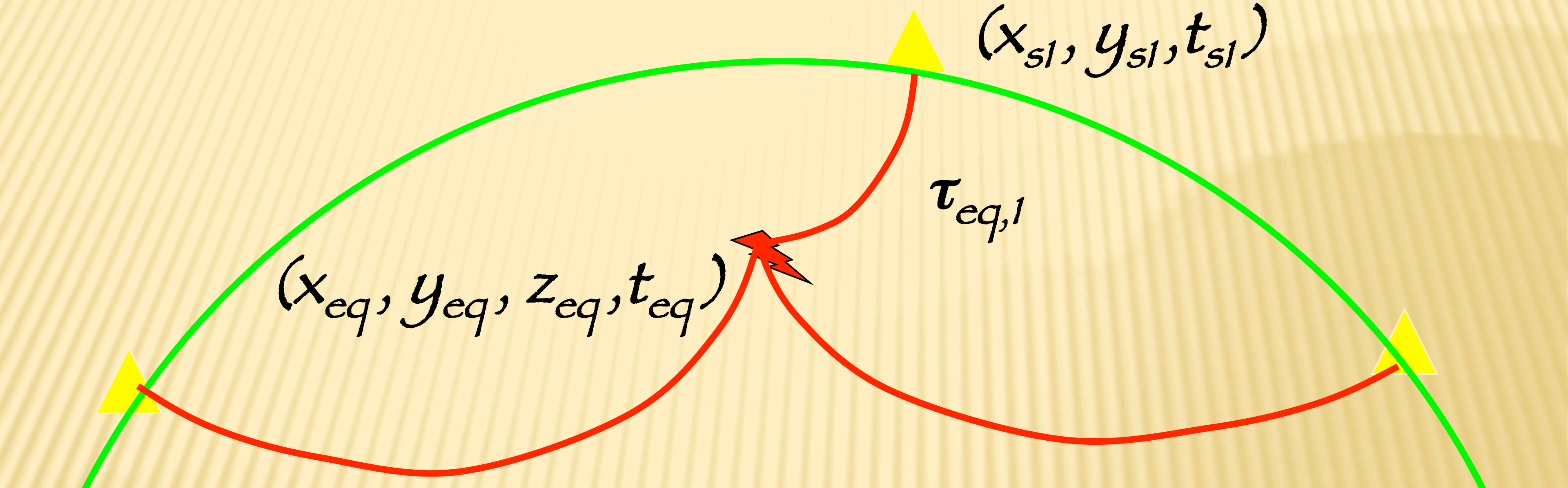
Now look at <u>travel time</u>, not distance.



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 (τ) rather than distance.

Let χ 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

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

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

This allows us to calculate the travel time to station k

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

from an earthquake at (location and time, does not really depend on *t*, but carry it along)

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

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 χ

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) Assuming a location

2) Linearizing the travel time equations

3) Use least squares to compute an adjustment to the location, which we will use to produce a new (better) location

4) Go back to step 1 using location from step 3

We do this till some convergence criteria is met

(if we' re lucky)

This is basically Newton's method

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

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

noise

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

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

First – linearize the expression for the arrival time $\tau(X,\chi)$

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

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

Now can put calculated here because can calculate this using the known (assumed) χ^*, but don't know these.

Now consider the difference between the observed and linearized τ – the residual $\Delta\tau$.

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

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

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

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

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 $m \geq 4$)

$$\begin{pmatrix} \Delta\tau_1 \\ \Delta\tau_2 \\ \Delta\tau_3 \\ \vdots \\ \Delta\tau_m \end{pmatrix} = \begin{pmatrix} \frac{\partial\tau_1}{\partial x} & \frac{\partial\tau_1}{\partial y} & \frac{\partial\tau_1}{\partial z} & \frac{\partial\tau_1}{\partial t} \\ \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_2}{\partial z} & \frac{\partial\tau_2}{\partial t} \\ \frac{\partial\tau_3}{\partial x} & \frac{\partial\tau_3}{\partial y} & \frac{\partial\tau_3}{\partial z} & \frac{\partial\tau_3}{\partial t} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\tau_m}{\partial x} & \frac{\partial\tau_m}{\partial y} & \frac{\partial\tau_m}{\partial z} & \frac{\partial\tau_m}{\partial t} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta t \end{pmatrix} + \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_m \end{pmatrix}$$

Jacobian matrix

Which is usually written as

$$b = Ax + \nu$$

Evaluating the time term

$$b = Ax + \nu$$

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

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

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

Expresses linear relationship between residual observations, $\Delta\tau$, and unknown corrections, δx.

Plus unknown noise terms.

Linearized observation equations

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

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

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

Previous linear least squares discussion gives us

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

–In GPS processing we call the matrix *A* the design matrix (it goes by other names in other fields)

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

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

This is called Geiger's method

Published 1910

Not used till ~1960s!

(when geophysicists first got hold of a computer)

So far

Have not specified type of arrival.

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

Need velocity <u>model</u> 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.)

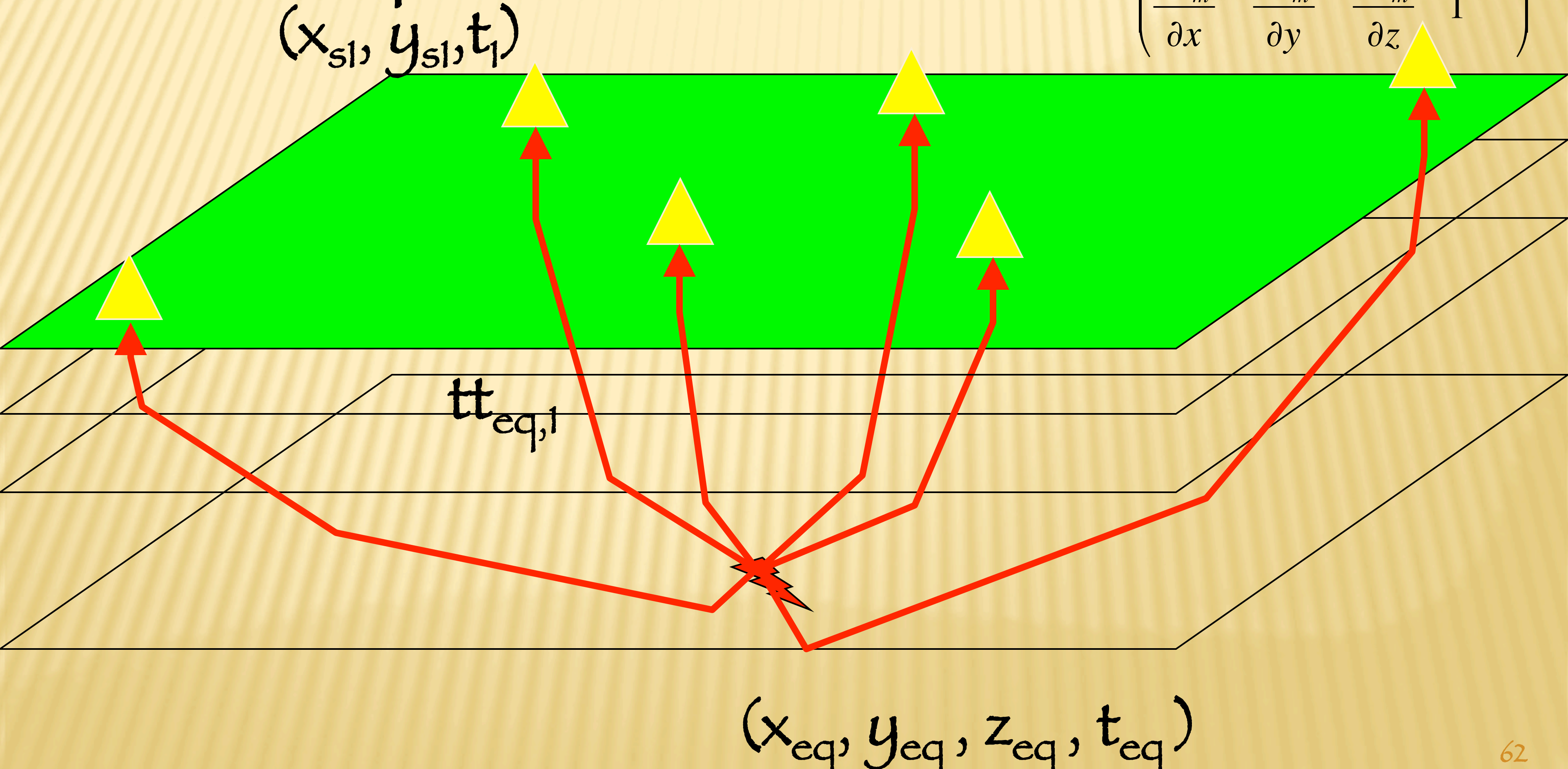
Velocity models usually laterally homogeneous.

Problems:

Column of 1's – if one of the other columns is constant the matrix is singular and can't be inverted.

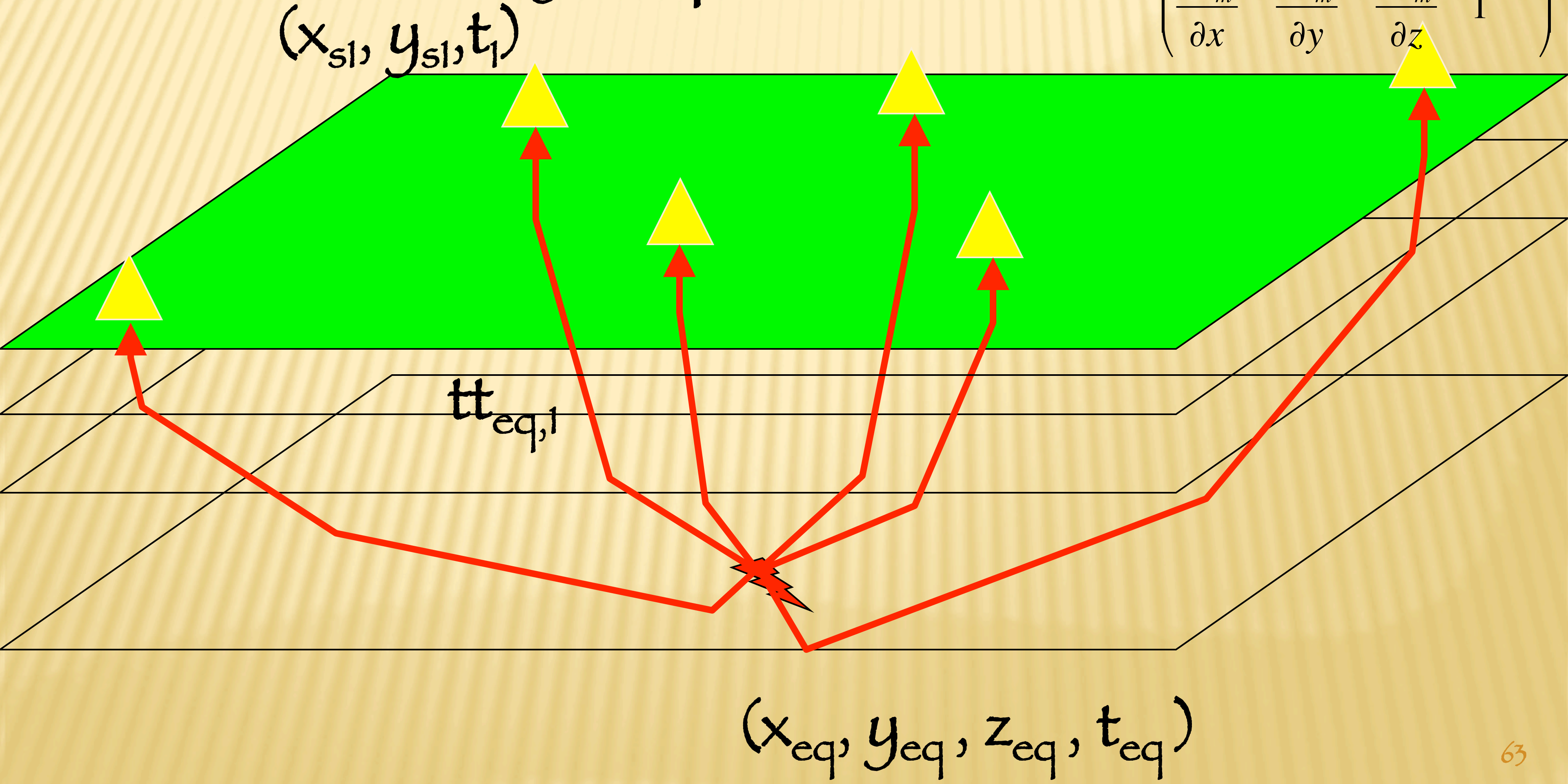
Mathematical problem.

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



Problems:
if any of the columns are approximately constant the matrix is "ill-conditioned" (looks singular in computer math) and can't be inverted by computer.

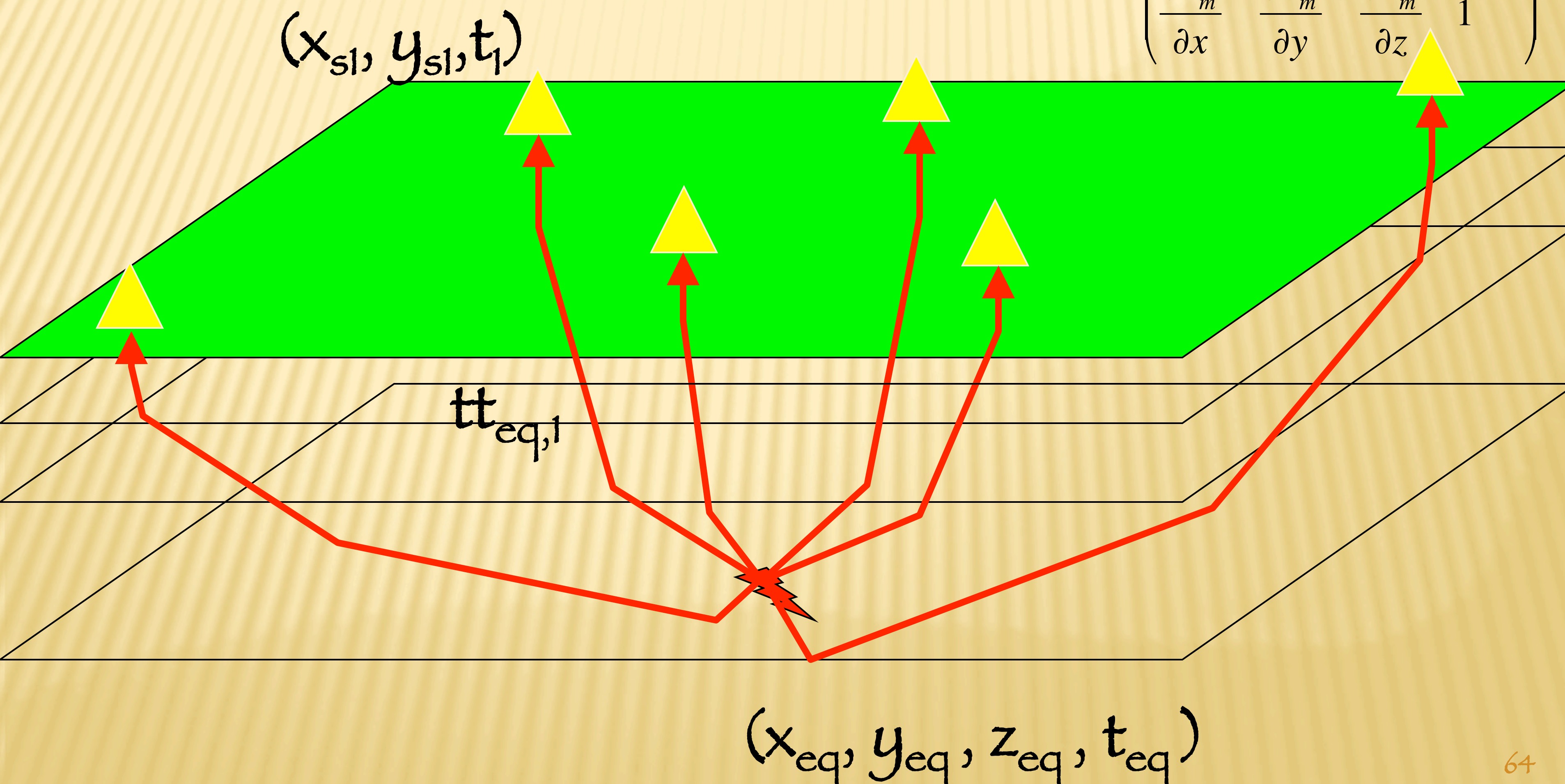
$$A = \begin{pmatrix} \frac{\partial \tau_1}{\partial x} & \frac{\partial \tau_1}{\partial y} & \frac{\partial \tau_1}{\partial z} & 1 \\ \frac{\partial \tau_2}{\partial x} & \frac{\partial \tau_2}{\partial y} & \frac{\partial \tau_2}{\partial z} & 1 \\ \frac{\partial \tau_3}{\partial x} & \frac{\partial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



How can this happen:

- All first arrivals are head waves from same refractor
- Earthquake outside the network

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

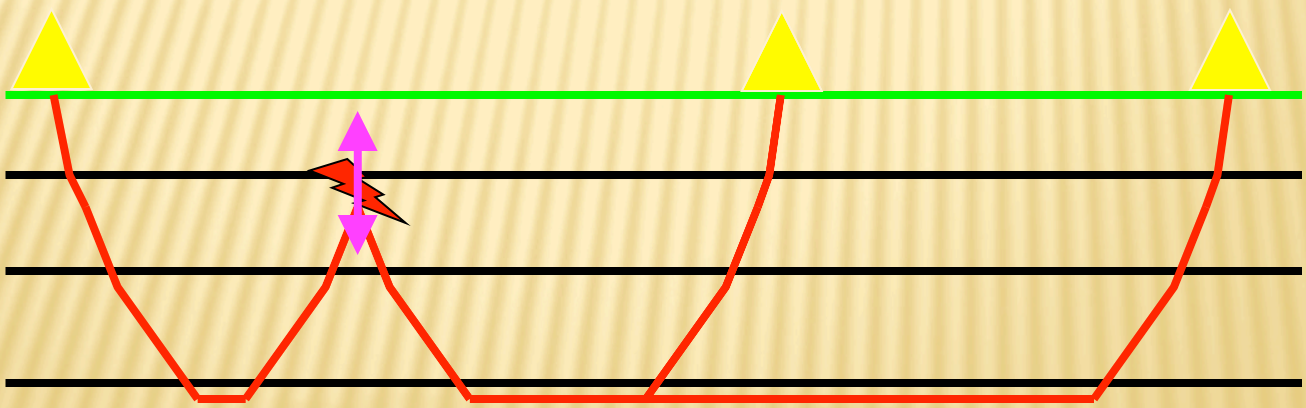


All first arrivals are head waves from same refractor

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

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

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



Earthquake outside the network

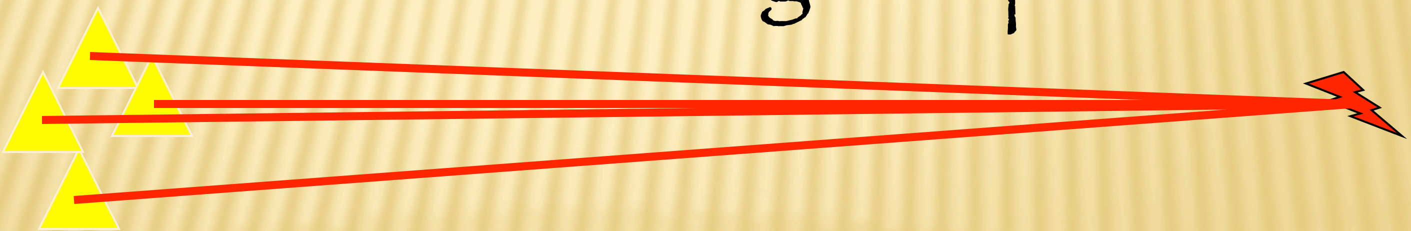
$$\frac{\partial \tau_k}{\partial x} \approx \text{constant } \forall\, k$$

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

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

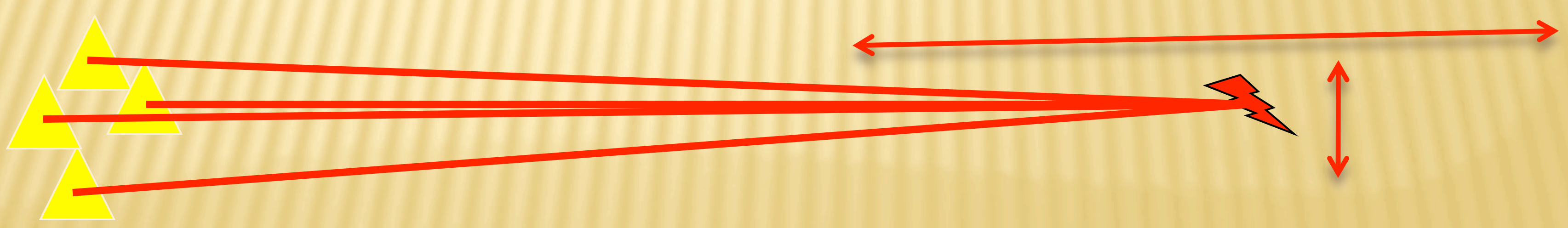
In this case only the azimuth is constrained.

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 since all first arrivals could be head waves.

Problem gets worse with addition of noise (changes length of red lines – intersection point moves left/right – change of distance - much more than in perpendicular direction – change of azimuth.)

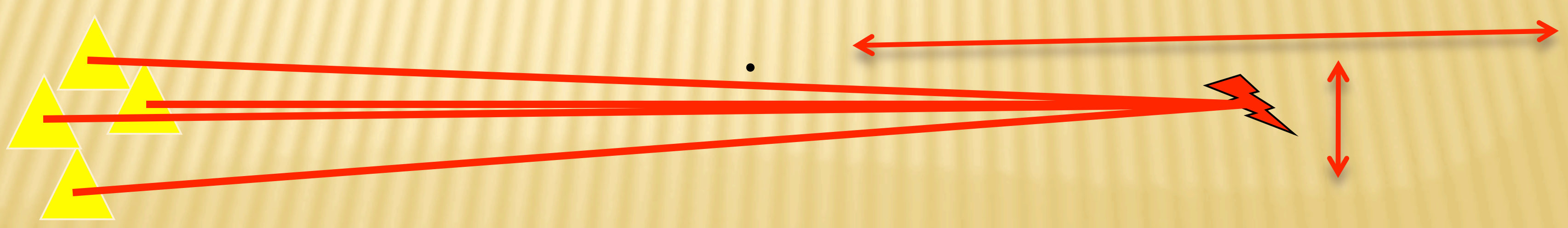


Similar problems with depth.

d/dz column ~equal, so almost linearly dependent on last column

and

gets worse with addition of noise (changes length of red lines – intersection point moves left/right [depth, up/down {drawn sideways}] much more than in perpendicular direction [position].)



Other problems:

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

When earthquake "crosses" a layer boundary (as iterate), 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 – especially in depth. Location tends to move horizontally at depth of boundary.

Other problems:

Related to earthquake location, but bigger problem for 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.

Since head wave usually very weak, oftentimes don't actually see head wave. Measure P arrival time, but location program treats it as P_n.

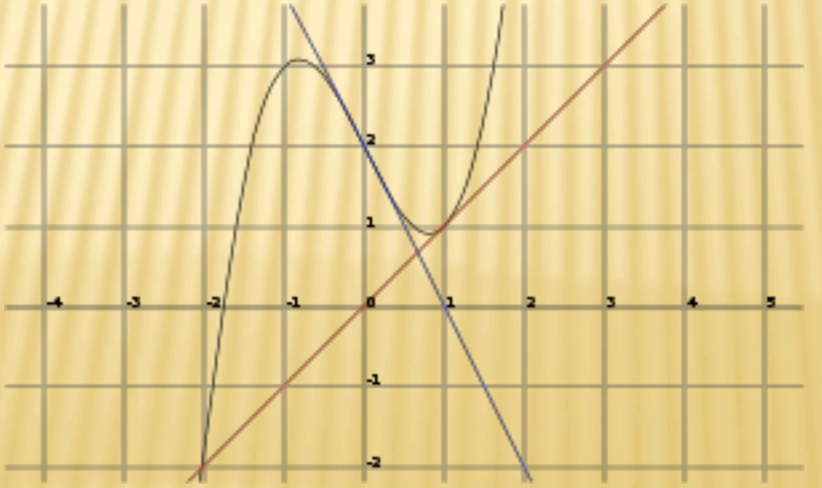
A look at Newton's method

Want to solve for zero(s) of F(x)

Start with guess, x_0.

Calculate $F(x_0)$ (probably not zero, unless VERY lucky).

Find intercept $x_1 = x_0 - F(x_0)/F'(x_0)$



Newton's method

Want to solve for zero(s) of F(x)

Now calculate $F(x_1)$.

See how close to zero.

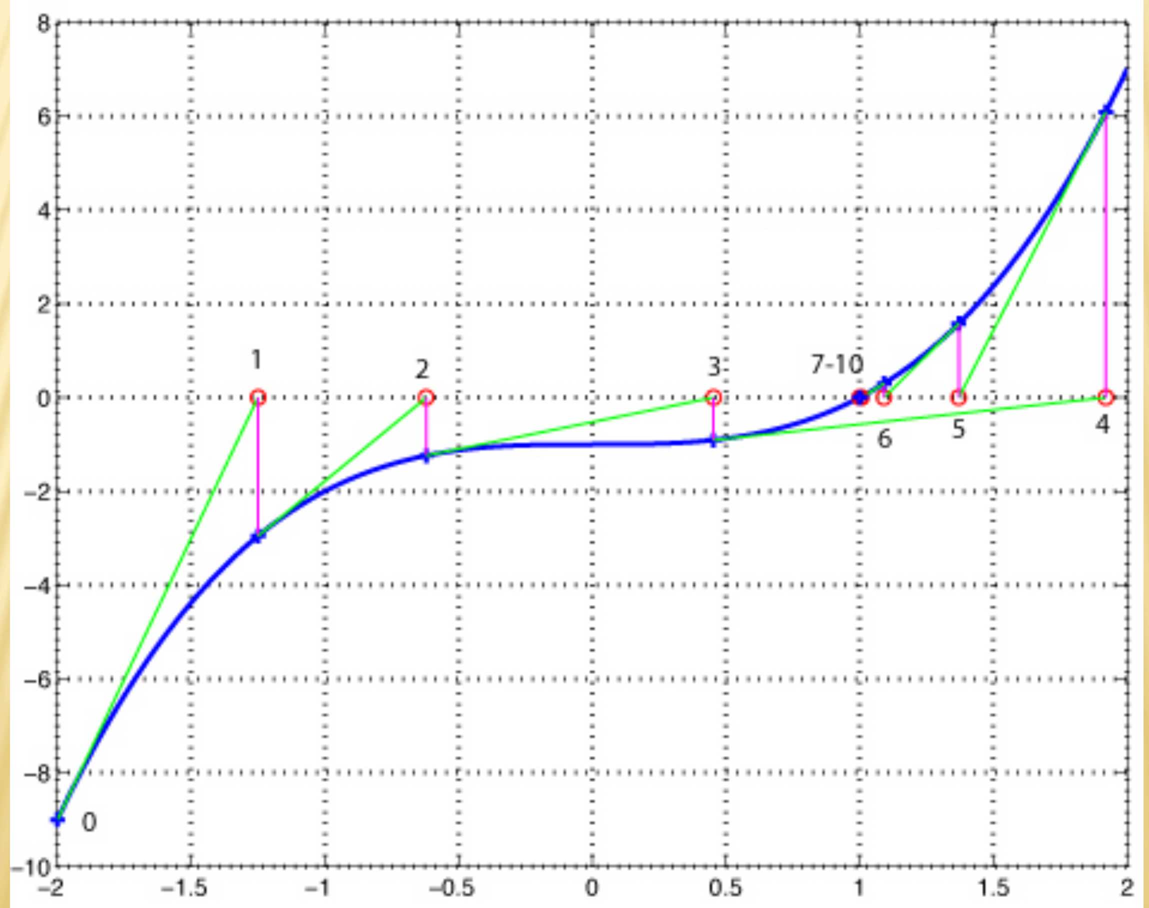
If close enough – done.

Newton's method

If not "close enough", do again

Find intercept $x_2 = x_1 - F(x_1)/F'(x_1)$

If close enough, done, else – do again.



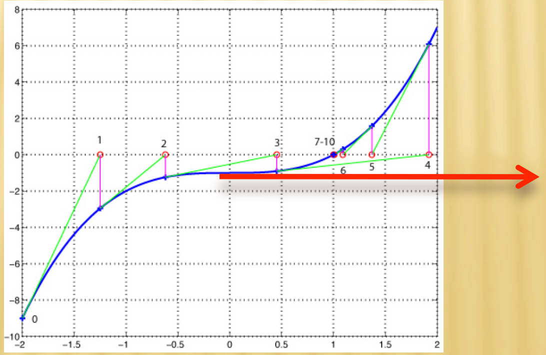
Newton's method

$$X_{n+1} = x_n - F(x_n)/F'(x_n)$$

What happens when $F'(x_n)=0$?

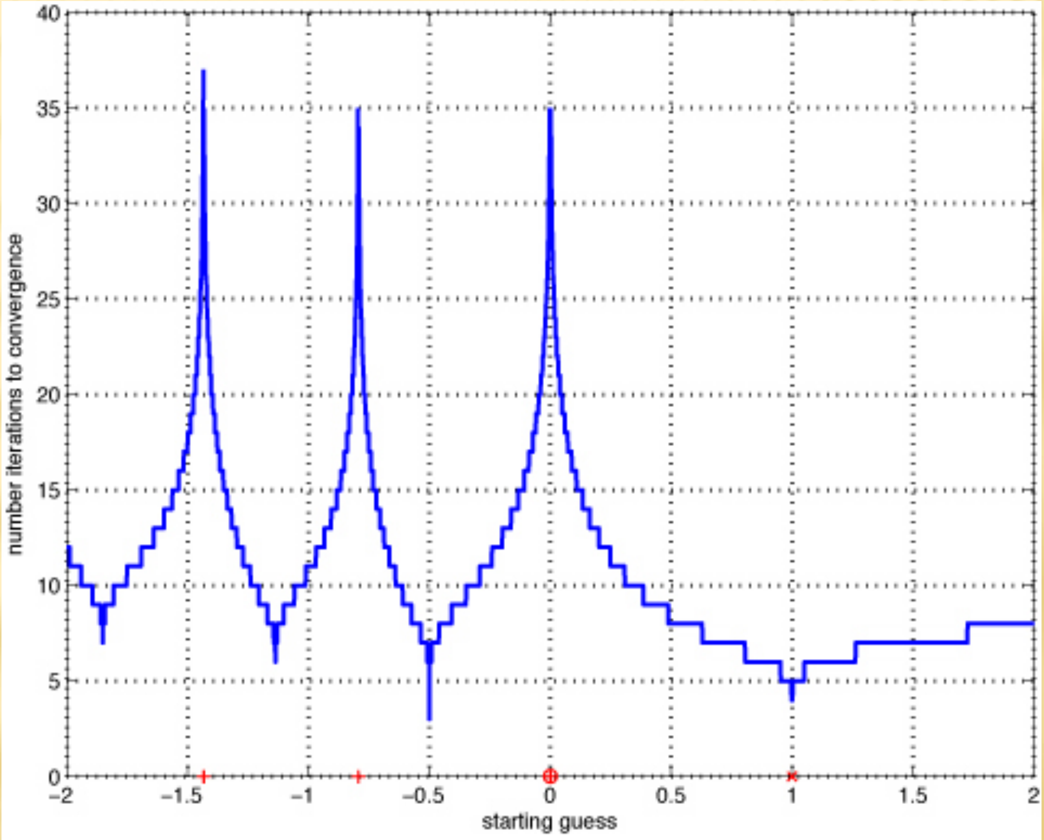
Geometrically, you get sent off to infinity – method fails.

(Mathematically can't divide by zero – method fails.)



Newton's method

How does convergence depend on starting value?



Some starting values iterate through x_n=0 and therefore do no converge (limited calculation to 35 iterations).

Newton's method

Other problems

Point is "stationary" (gives back itself x_n -> x_n…).

Iteration enters loop/cycle: x_n -> x_{n+1} -> x_{n+2} -> or = x_n …

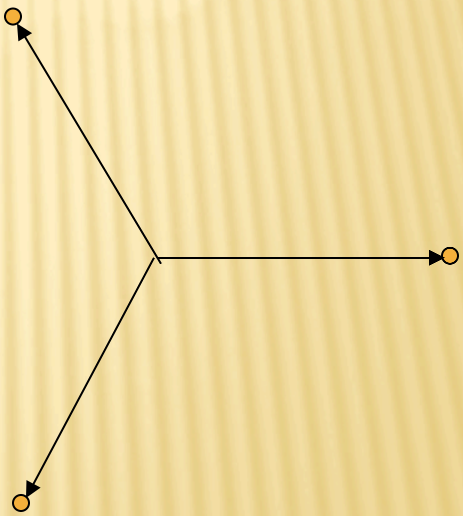
Derivative problems (does not exist, eg. absolute value at 0).

Discontinuous derivative.

Newton's method applied to solution of non-linear, complex valued, equations

Consider

$Z^3-1=0.$



Newton's method applied to solution of non-linear, complex valued, equations

Consider

$Z^3-1=0.$

Solutions

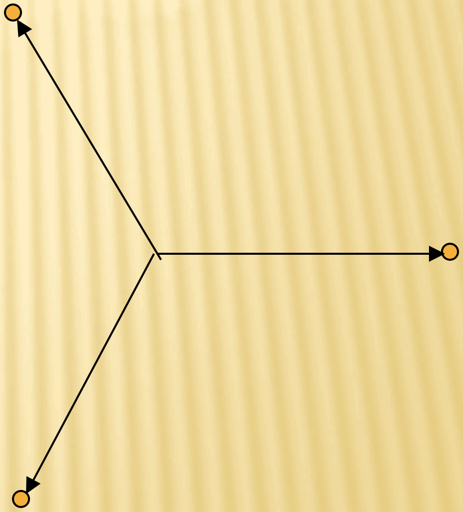
Three of them

$1\,e^{(i2\pi n/3)}$

n=0, 1, 2

Distance = 1

Every 120 degrees



Newton's method applied to solution of non-linear, complex valued, equations

Consider

$Z^3 - 1 = 0$

Solutions

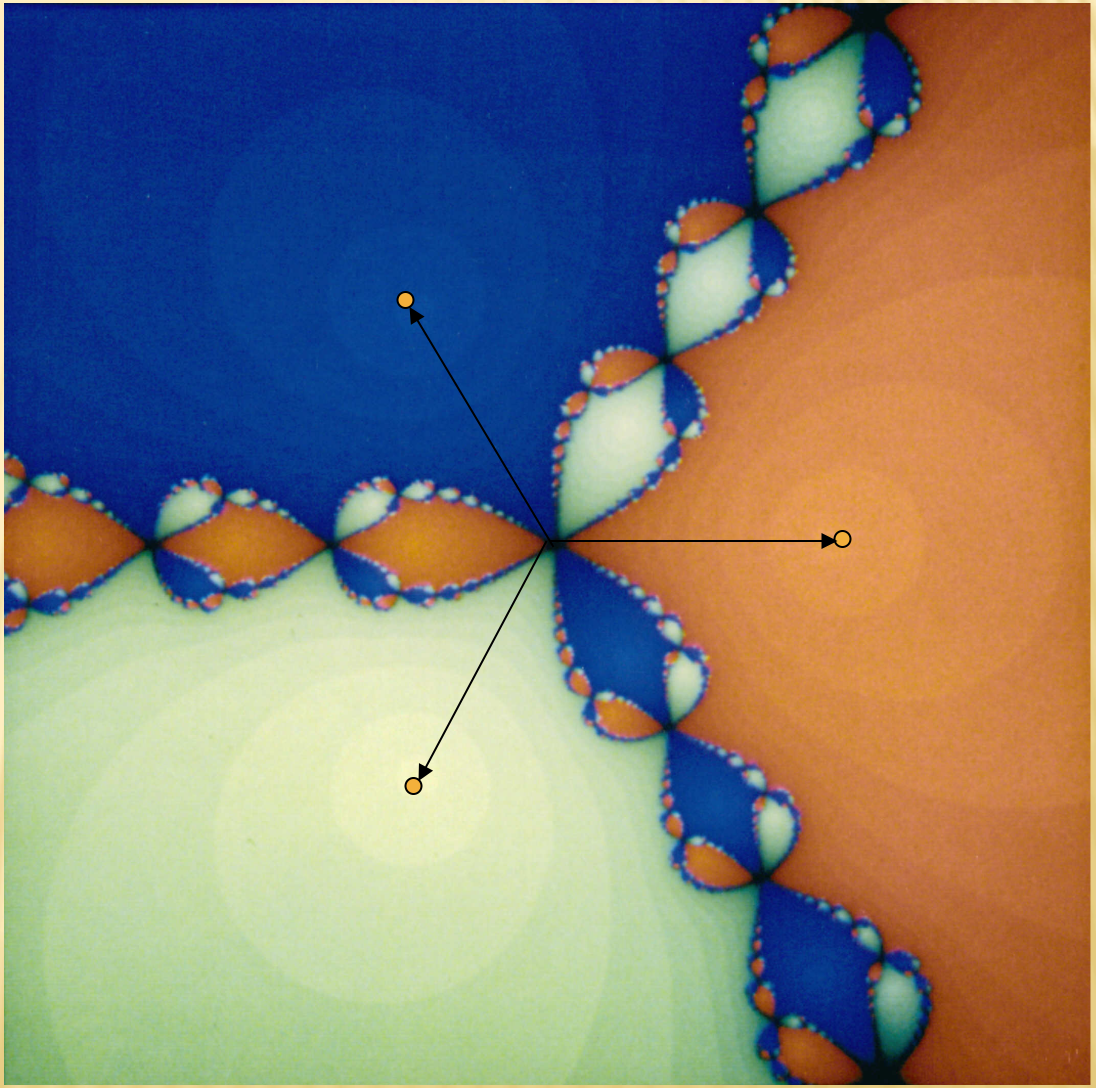
Three of them

$1\, e^{(i2\pi n/3)}$

n=0, 1, 2

Distance = 1

Every 120 degrees

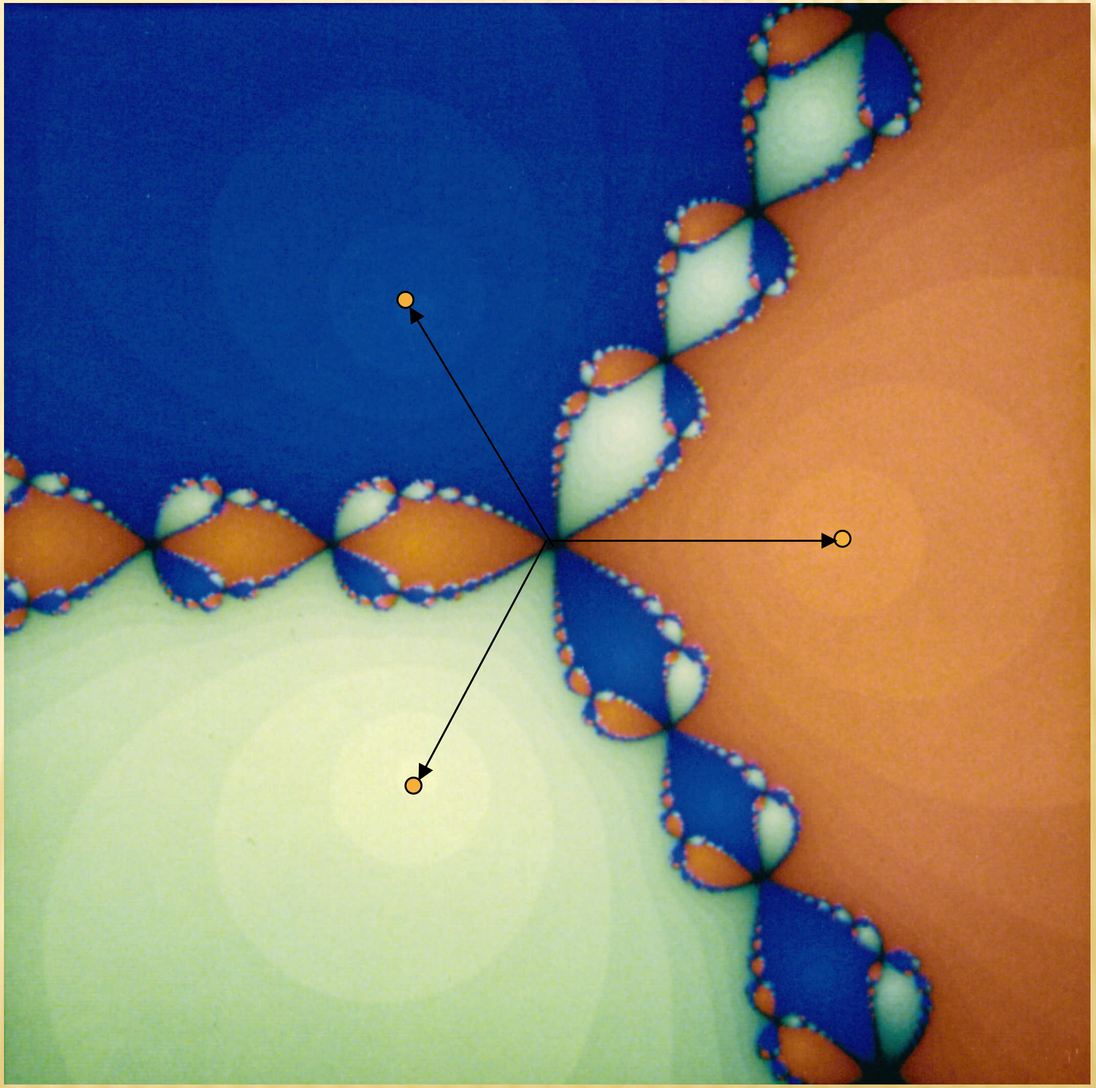


Take each point in the complex plane as a starting guess and apply Newton's method.

Now

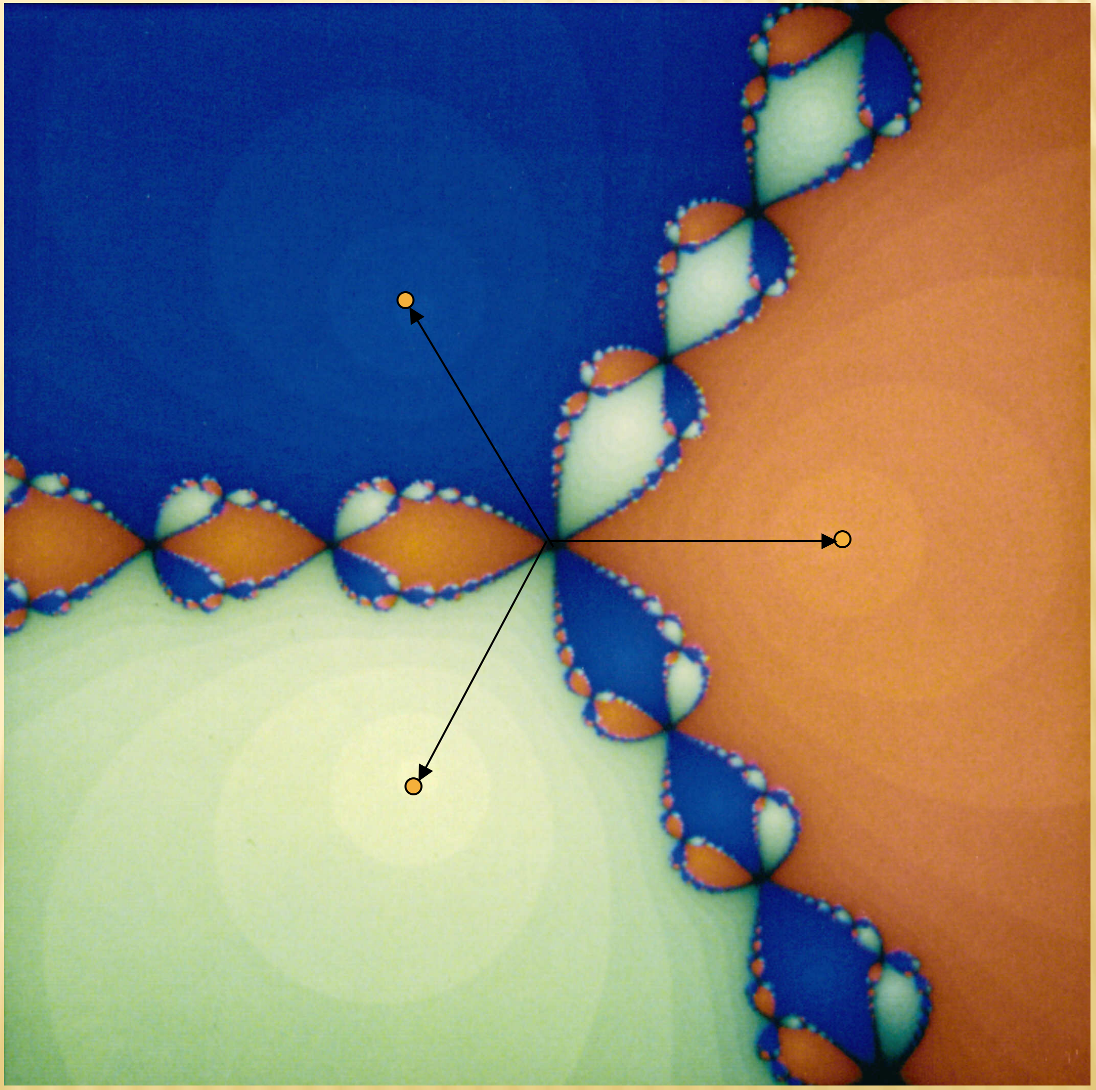
Color the starting points to identify which of the three roots each starting point converges to using Newton's method.

eg. all the red points converge to the root at 1.



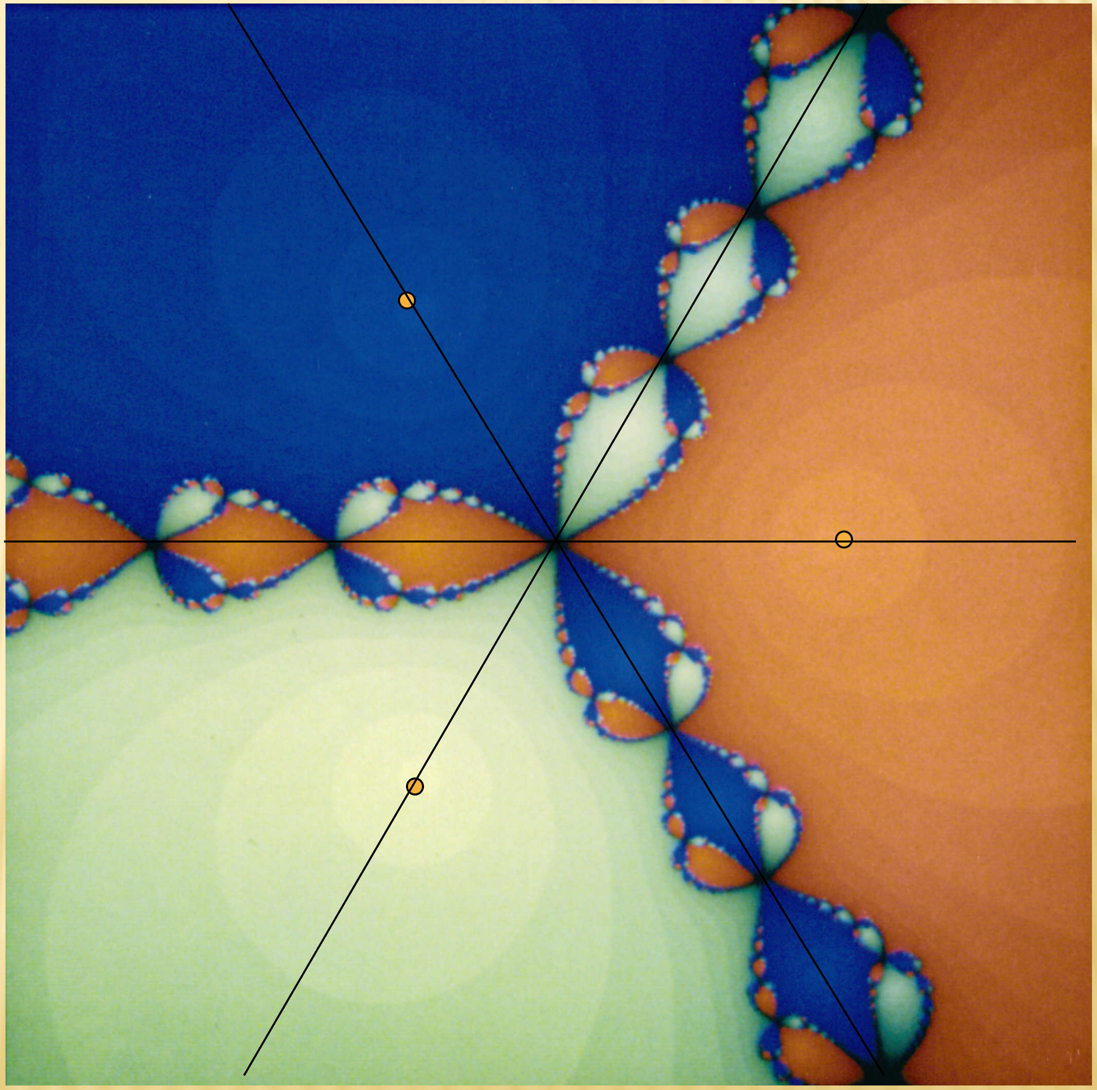
Let the intensity of each starting point's color be related to the number of steps to converge to that root

(brighter - converges faster, darker – converges slower)

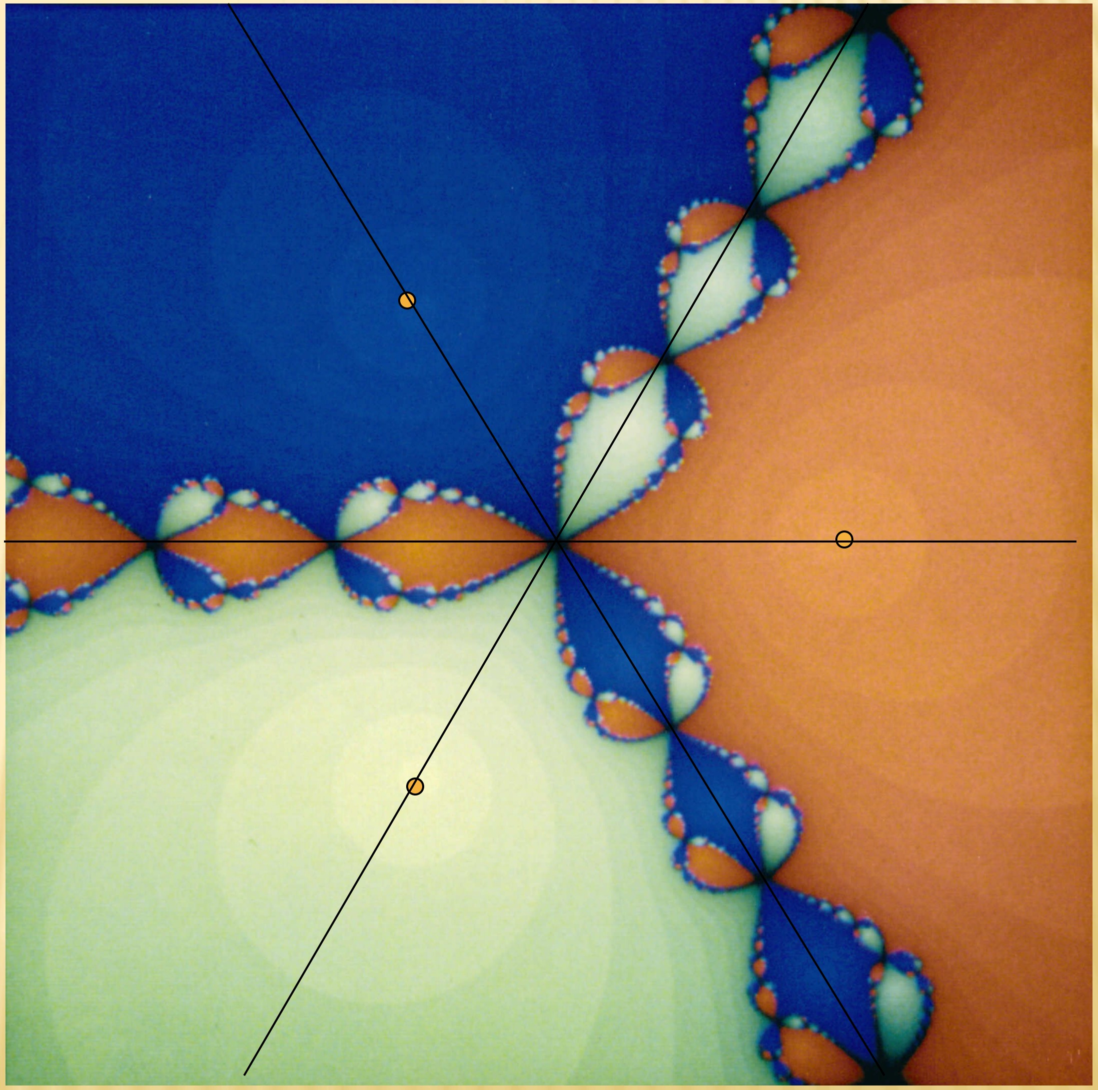


Notice that any starting point on the real line converges to the root at 1

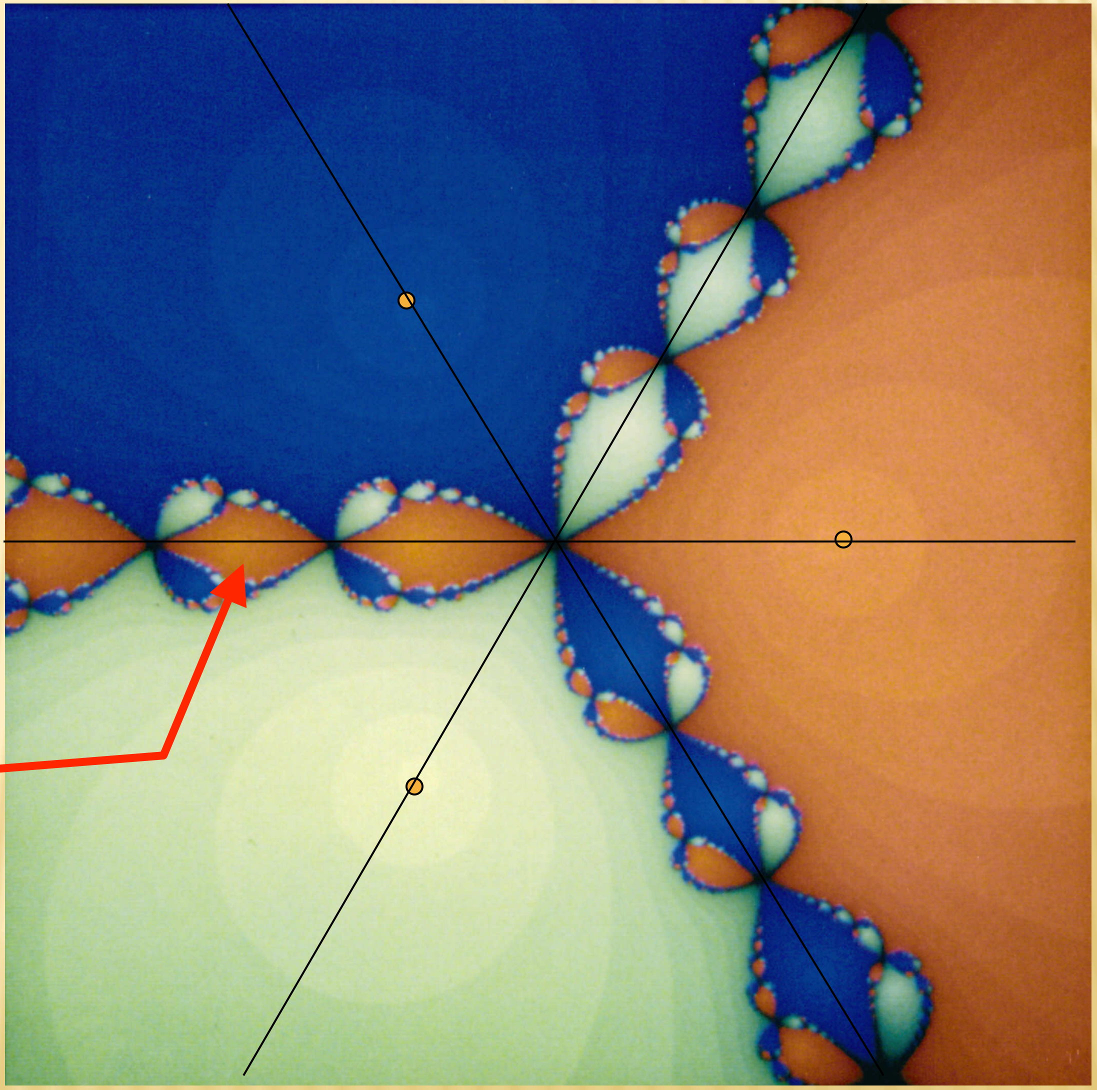
Similarly points on line sloping 60 degrees converge to the other 2 roots.



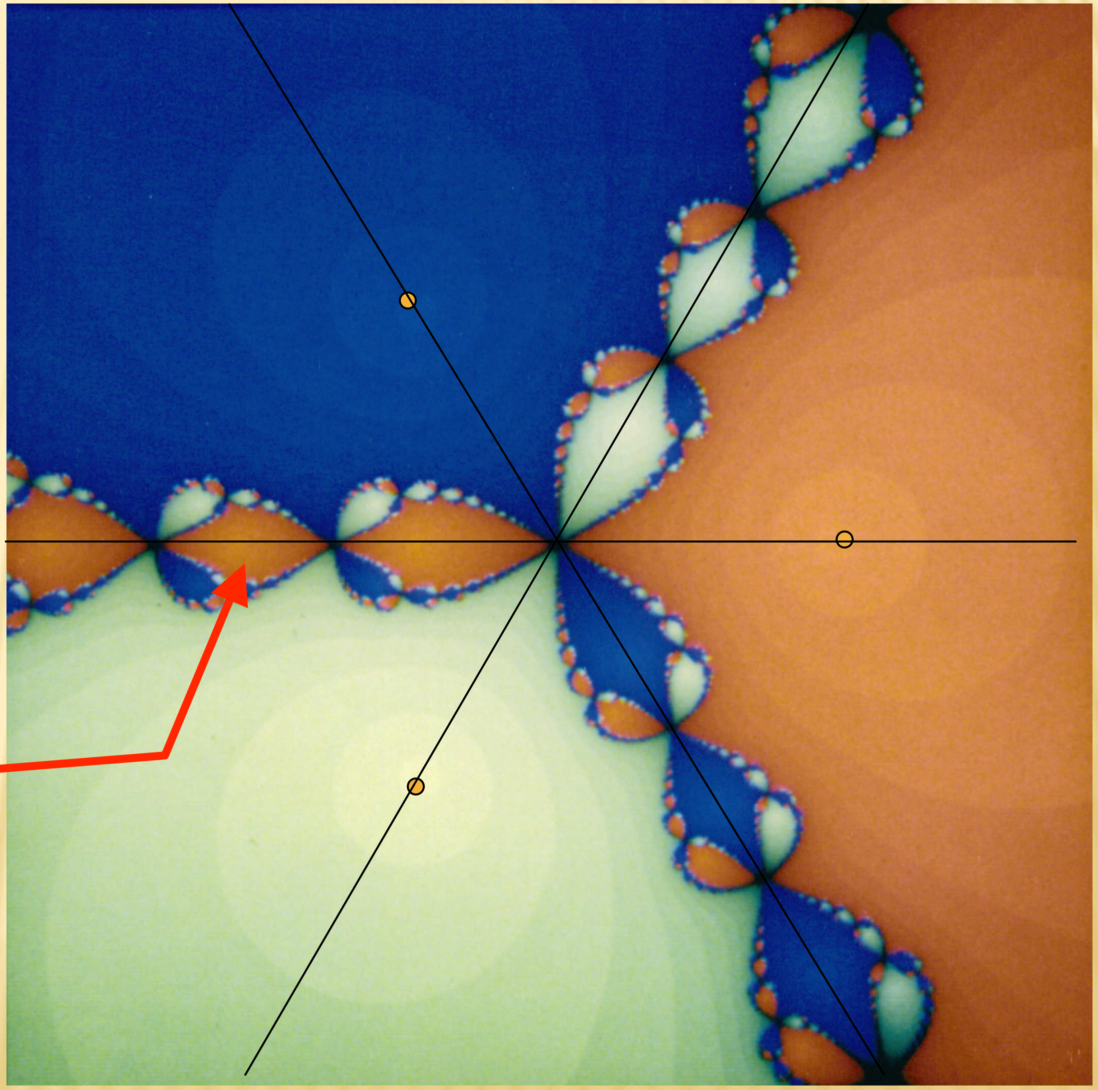
Notice that in the ~third of the plane that contains each root things are pretty well behaved.



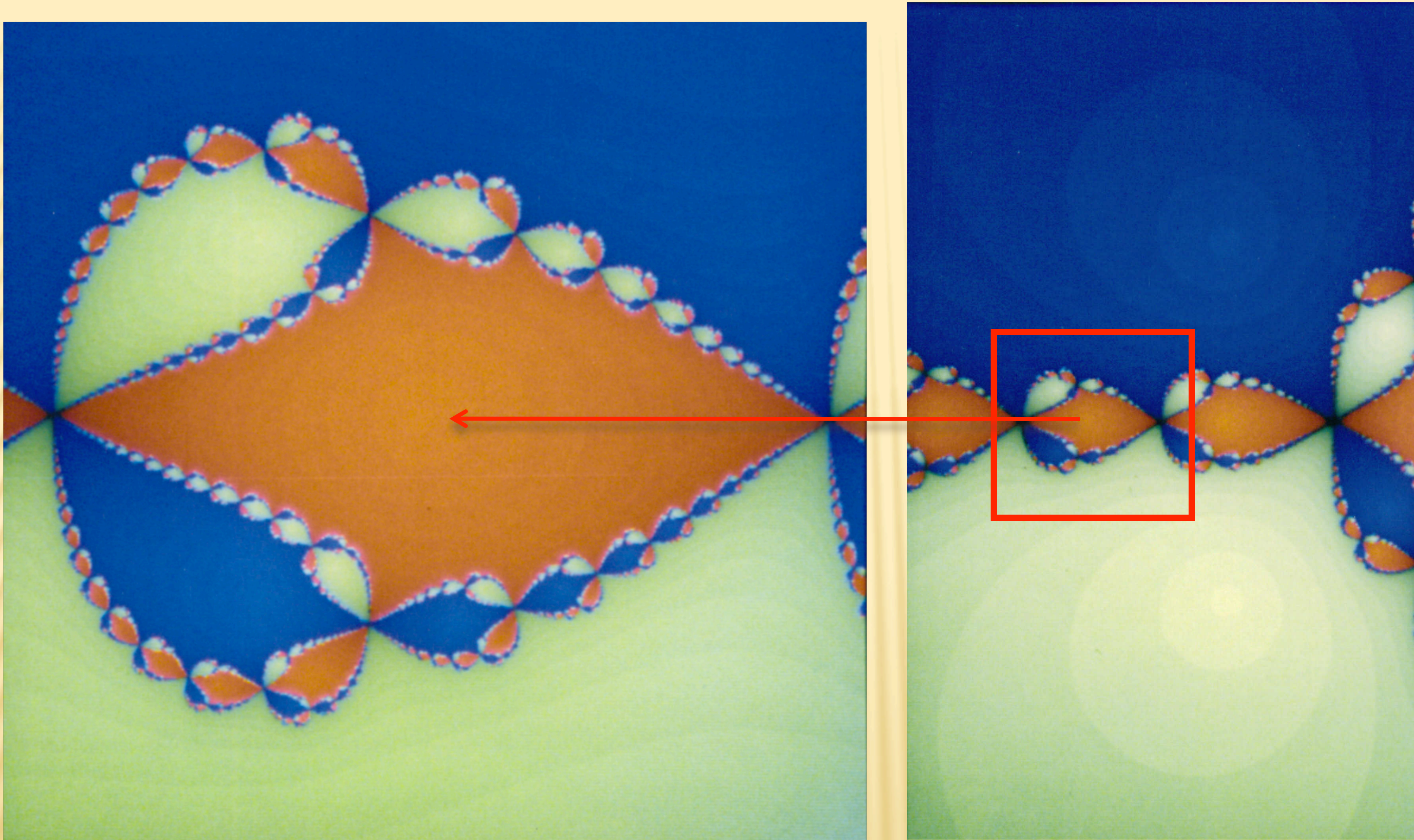
Notice that where any two domains of convergence meet, it looks a little complicated.



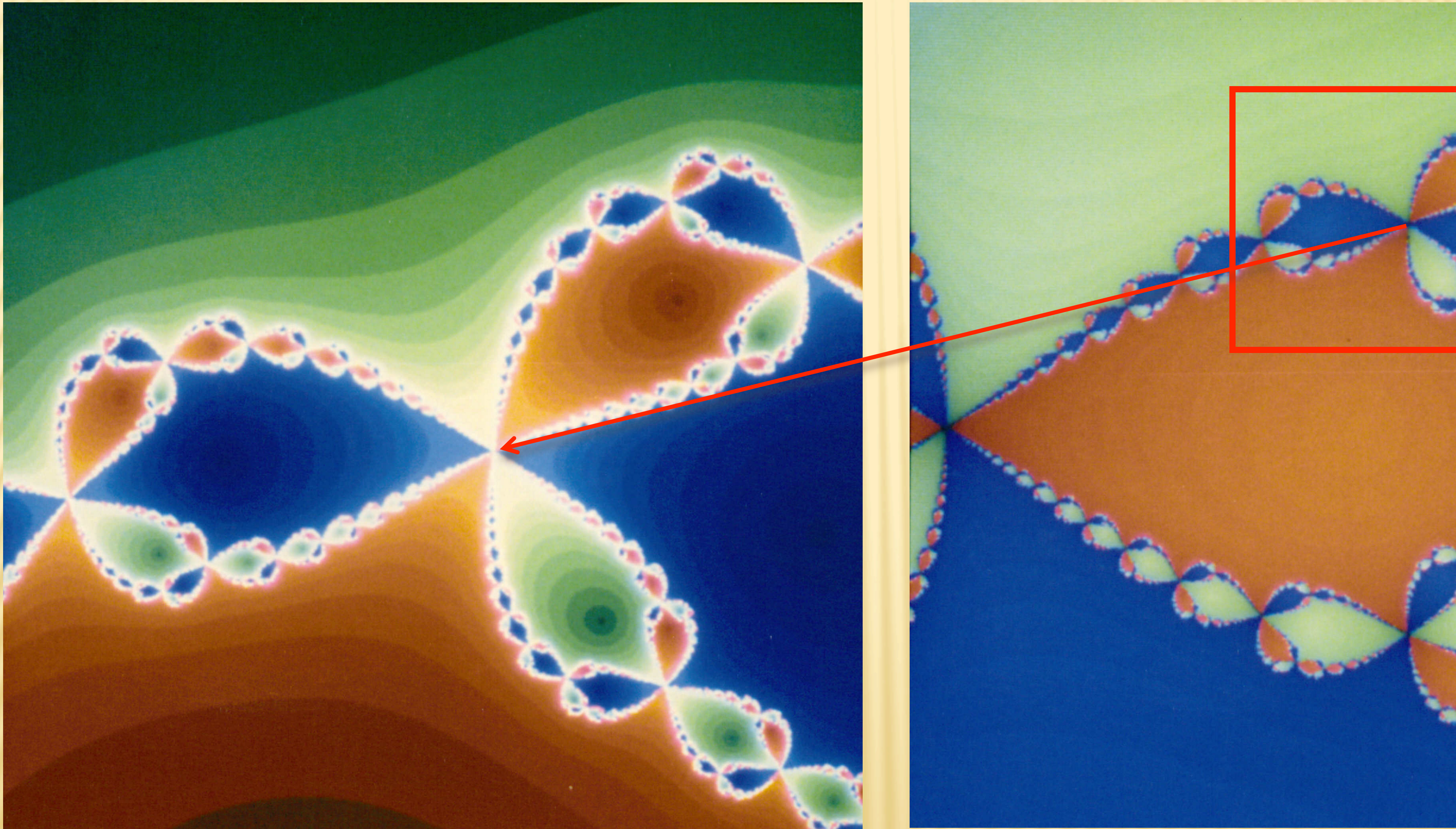
Basically – the division between any two colors is always separated by the third color.



AT ALL SCALES!



Zoom in



Zoom in again

If you keep doing this (zoom in) the “triple” junctions start to look like

Mandlebrot sets!

and you will find points that either never converge or converge very slowly.

Quick implication –

linear iteration to solve non-linear inversion problems

(Newton’s method, non-linear least squares, etc.)

may be unstable and not work.