Analysis Methods in Atmospheric and Oceanic Science

AOSC 652

Root Finding & Function Minimization

Week 7, Day 2

12 Oct 2016

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Root is place where $f(x) = 0$

$$0 \approx f(x_0) + f'(x_0)\,\Delta x = f(x_0) + f'(x_0)\,(x_1 - x_0)$$

- If the derivative is evaluated at x_0 then the "next" value x_1 is found by:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

- Convergence if either:
  - $f(x_i) = 0$ or $f(x_i)$ lies within a certain distance of 0 –or–
  - update to $|x_i| < x_{Tolerance}$

- If not converged, next value x_{i+1} is found from:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Very powerful convergence method provided:
  - derivative is continuous and non-zero near root
  - initial guess for root is "close" to root
  - should use with iteration counter and check to see if root (or function minima) was indeed found
- Can read about Newton Raphson iteration on the web:
  http://en.wikipedia.org/wiki/Newton-Raphson_method

Numerical algorithm semantics:

- **Newton-Raphson or Newton's method: derivative can be found analytically**
- **Secant method: derivative can only be found numerically**

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #3: Newton-Raphson

- **Assignment for today**
- Copy file ~rjs/aosc652/week_07/newton_raphson.f
  into your work area
- Have a look at the code and compare / contrast to *subroutine rtnewt* on page 358 of Press et al.
- Compile and run
- Find the root of $cos(x) - x^2$ to an accuracy of 1.E–2
- Repeat for accuracies of 1.E–4 and 1.E–6
  - How many calls to the function were required to find the root for each call?
  - Prepare a table noting the efficiency of the three methods for finding the root of $cos(x) - x^2$ for accuracies of 1.E–2, 1.E–4, and 1.E–6

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Find the root of $cos(x)–x^2$ to an accuracy of 1.E–2
- Repeat for accuracies of 1.E–4 and 1.E–6
  - How many calls to the function were required to find the root for each call?
  - Prepare a table noting the efficiency of the three methods for finding the root of $cos(x)–x^2$ for accuracies of 1.E–2, 1.E–4, and 1.E–6

	1×10^{-2}	1×10^{-4}	1×10^{-6}
Direct Search			
Bisection			
Newton-Raphson			
Root Newton-Raphson			
F(Root) Newton-Raphson			

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #3´: Secant Method

- Similar to Newton-Raphson method except derivative is evaluated numerically:
  - Newton-Raphson method:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

  - Secant Method:

$$x_{i+1} = x_i - \frac{f(x_i)\,[x_i - x_{i-1}]}{f(x_i) - f(x_{i-1})}$$

    - Some functions difficult or impossible to differentiate
    - Initial guess for x_1 and x_0 can be crucial for whether algorithm converges

Root Finding and Function Minimization

4. Function Minimization

- We will not delve much into the details of minimization
- Please read 3 paragraphs, page 375 of Press *et al*., that compares and contrasts root finding versus function minimization
- Chapter 10 of Press *et al*. deals with function minimization and maximization
  - many of the routines similar to those developed to find roots
  - "Simulated annealing" (section 10.9) is a very powerful method for finding minima of a set of multi-dimensional vectors

Root Finding and Function Minimization

4. Function Minimization

- We must describe the **Jacobian Matrix, *J***
- Suppose we are seeking the minima of an array ***F(x)***,
  where F has m dimensions and $\boldsymbol{x}$ has $\boldsymbol{n}$ dimensions
- Present iteration: $x_1, x_2, x_3, \ldots x_n \Rightarrow x_{\text{old}}$
- Next iteration: x_{new}

$$x_{\text{new}} = x_{\text{old}} + \delta x$$

$$\delta x = - J^{-1} \cdot F_{\text{old}}$$

Root Finding and Function Minimization

4. Function Minimization

- We must describe the **Jacobian Matrix, *J***
- Suppose we are seeking the minima of an array ***F(x)***,
  where F has m dimensions and ***x*** has ***n*** dimensions
- Present iteration: $x_1, x_2, x_3, \ldots x_n \Rightarrow x_{old}$
- Next iteration: x_{new}

$$x_{new} = x_{old} + \delta x$$

$$\delta x = -J^{-1} \cdot F_{old}$$

What is J ???

Root Finding and Function Minimization

4. Function Minimization

- Two examples where the **Jacobian Matrix, *J* :** is often used

$$J_{ik} = \frac{\partial F_i}{\partial x_k}$$

Root Finding and Function Minimization

4. Function Minimization

- Two examples where the **Jacobian Matrix, *J* :** is often used $J_{ik} = \frac{\partial F_i}{\partial x_k}$

  a) retrievals of atmospheric composition from a measured spectra

  Here, the difference between the measured spectra and a "fit" to spectra calculated using a radiative transfer model is the *m* dimensional vector *F* and the abundance of various species that affect the spectra at various heights in the atmosphere is the *n* dimensional vector *x* (sometimes called the **state vector**)

AOSC 652: Analysis Methods in AOSC

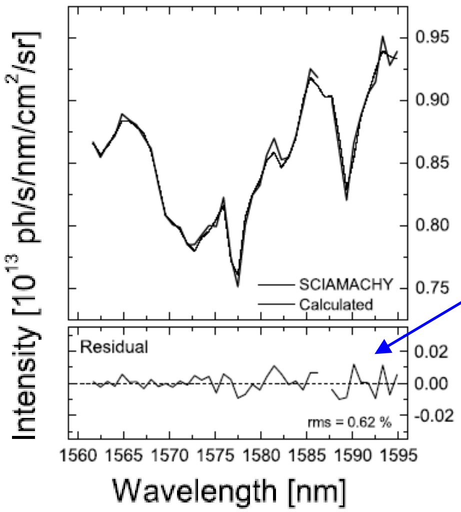
Root Finding and Function Minimization

4. Function Minimization

- Two examples where the **Jacobian Matrix, *J* :** $J_{ik} = \frac{\partial F_i}{\partial x_k}$ is often used

  a) retrievals of atmospheric composition from a measured spectra

  Here, the difference between the measured spectra and a "fit" to spectra calculated using a radiative transfer model is the *m* dimensional vector *F* and the abundance of various species that affect the spectra at various heights in the atmosphere is the *n* dimensional vector *x* (sometimes called the **state vector**)



CO_2 & H_2O profiles adjusted to obtain "best fit" to measured spectra

The integral of CO_2 profile from "best fit" (minim. residual) ⇨ column CO_2, quantity "measured" by instrument

Bösch et al., JGR, 2006

Root Finding and Function Minimization

4. Function Minimization

- Two examples where the **Jacobian Matrix, *J* :** is often used

$$J_{ik} = \frac{\partial F_i}{\partial x_k}$$

a) retrievals of atmospheric composition from a measured spectra

b) ???

Root Finding and Function Minimization

4. Function Minimization

- Two examples where the **Jacobian Matrix, *J* :** is often used $J_{ik} = \frac{\partial F_i}{\partial x_k}$

  a) retrievals of atmospheric composition from a measured spectra

  b) ???

Frequently asked question in retrieval theory and numerical modeling: how did you calculate your Jacobian Matrix: i.e. analytically or numerically?

Analytically: generally preferred for convergence, but real pain to update when new terms are added to the forward model

Numerically: how our photochemical model works (too much trouble updating an analytic derivative)

Root Finding and Function Minimization

5. Simulated Annealing

- Method of finding function minimization first implemented in 1953 !
- Based on analogy with thermodynamic behavior of crystals:
  - If a liquid is cooled slowly, thermal mobility is lost. Atoms line themselves up and form a pure crystal with order that is completely ordered over a distance billions of times the size of individual atoms
  - Amazing that nature is able to find this "minimum energy state"
  - If a system cools quickly (i.e., is quenched), it does not reach this state but rather enters into an amorphous (polycrystalline) state

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

5. Simulated Annealing

- Method of finding function minimization first implemented in 1953 !
- Based on analogy with thermodynamic behavior of crystals
- Although analogy is not perfect, but prior minimization algorithms (i.e., Newton-Raphson) correspond to rapid cooling (we try to find the nearby solution quickly)
- Nicholas Metropolis and colleagues (including Edward Teller) put forth a method whereby iterations were performed such that:
  - Let $E_1(\bar{x})$ by the present energy (cost function) of a system for state vector $\bar{x}$
  - Compute $E_2(\bar{x}+\Delta\bar{x})$
  - If $E_2 < E_1$, accept change always
  - If $E_2 > E_1$, accept change with probability $\exp[-(E_2-E_1)/kT]$

- What is the physical analogy in nature to $\Delta E/kT$? To $\exp(-\Delta E/kT)$?
- What ***characteristic*** does this process for function minimization offer that is distinctly different than Newton Raphson ?

Root Finding and Function Minimization

5. Simulated Annealing

- To implement the Metropolis algorithm, need:
  - A description of possible system configurations (state vector)
  - A generator of the random changes to the system (i.e., state vector options)
  - A measure of the energy (cost function) of the system
  - An analog of kT (usually this is adjustable) as well as a control of step size (also adjustable)

- Chapter 10 of Press et al. describes Simulated Annealing in some detail; this is not assigned, but please read if you are interested

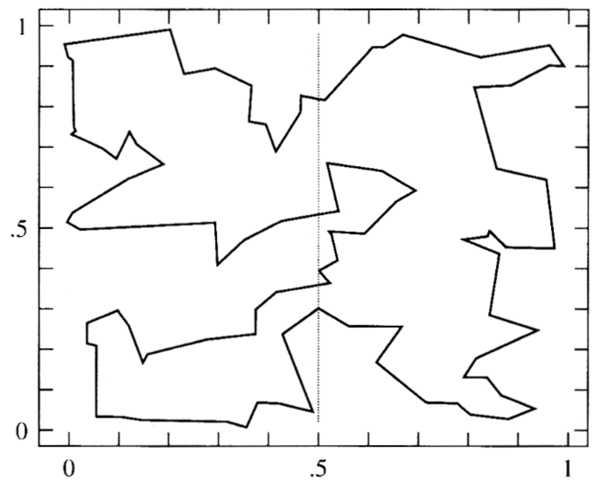
AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

5. Simulated Annealing

- Some examples from Press et al.

440 *Chapter 10. Minimization or Maximization of Functions*



(a)

Image from Press et al.

Traveling salesman problem: what was the cost function ?

Root Finding and Function Minimization

5. Simulated Annealing

- Some examples from Press et al.

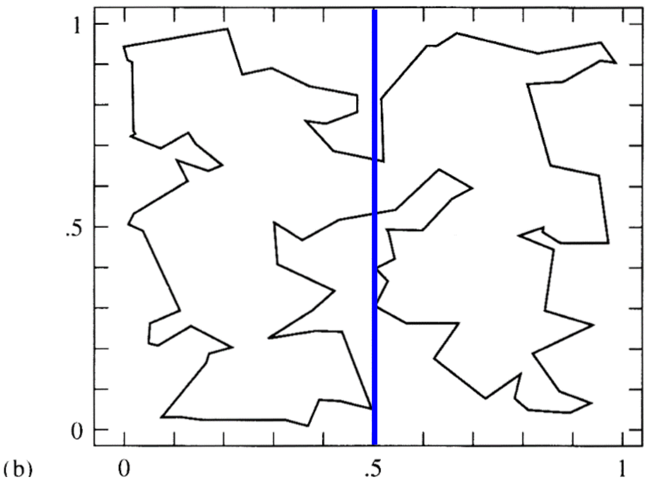


Image from Press et al.

How has the cost function been modified ?

Root Finding and Function Minimization

5. Simulated Annealing

- Some examples from Press et al.

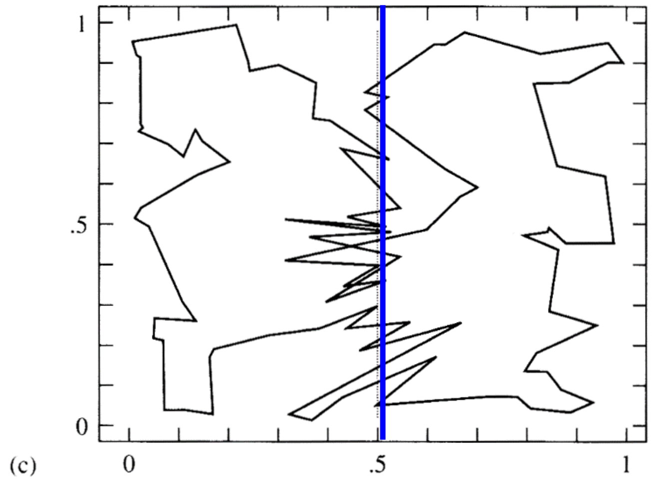


Image from Press et al.

How has the cost function been modified ?

Root Finding and Function Minimization

5. Simulated Annealing

- I've used *simulated annealing* for one geophysical problem in my career
  - We were trying to get a better estimate of oceanic circulation (especially deep water formation in the North Atlantic) based on profiles of oceanic composition conducted by the GEOSECS program
  - We began with a canonical circulation, developed a model of ocean T, salinity, salt content, etc and compared to GEOSECS data
  - What do you think our cost function was ?

Root Finding and Function Minimization

6. Singular Value Decomposition

- Suppose you have M equations and N unknowns, but M < N
- Is there a unique solution ?

Root Finding and Function Minimization

6. Singular Value Decomposition

- Suppose you have M equations and N unknowns, but M < N
- If X is our cost function, the Singular Value Decomposition of Matrix X

$$\underset{m\times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m\times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r\times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r\times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

can provide critical information for finding solutions
(see Chapter 2 of Press et al.)

$$\overline{Solution} = \overline{V} \bullet [diag(1/s_{jj})] \bullet (\overline{U}^{T} \bullet \overline{b})$$

Root Finding and Function Minimization

6. Singular Value Decomposition

- SVDs are used extensively in **Numerical Weather Prediction**
- SVDs may be expensive to compute but, once in hand, are quite powerful
- Eugenia Kalnay describes use of SVD for ensemble forecasts in her data assimilation class
- SVDs are particularly amenable to solution of problems that are either over-determined or undetermined:
  – See http://websites.uwlax.edu/twill/svd/svd/index.html for an ***excellent*** description

- I've used *SVD* to solve to solve one numerical problem in my life … it is rather interesting, especially if you are a sports fan ☺