Unit 5.1 – Root Finding

Root finding is the identification of values that solve an equation. For

$$y = 7x + 1 \qquad (5.1\text{-}1)$$

We might want to know what *x* value corresponds to *y=15*. This means we would solve

$$15 = 7x + 1 \qquad (5.1\text{-}2)$$

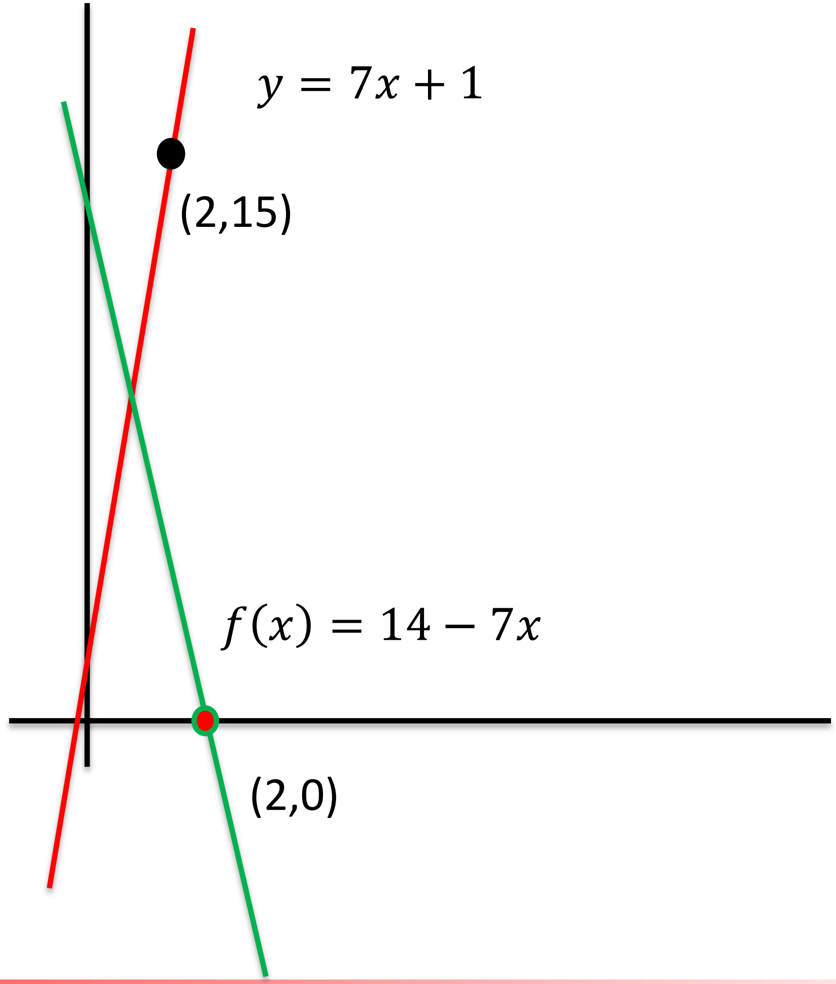
Or as finding the zero(s?) of *f(x),* where:

$$\mathrm{f}(x) = 14 - 7x = 0 \qquad (5.1\text{-}3)$$

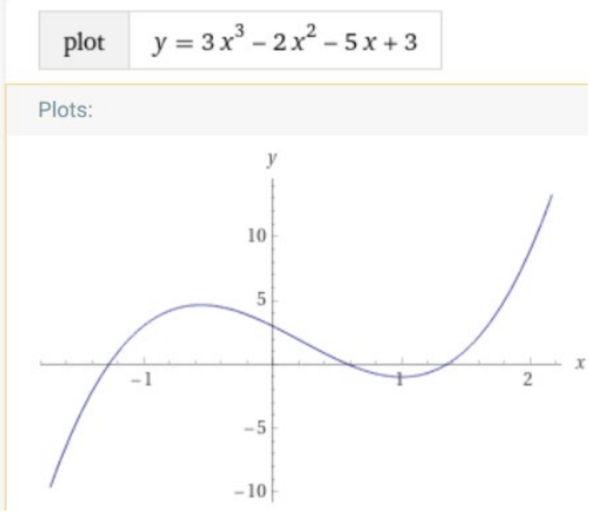
We see the solution is *x=2*.

Note the two decidedly different problems expressed by Eqs. 5.1-1 and 5.1-3 and shown graphically to the right.

The two are decidedly different descriptions of the same problem. The latter suggests an approach since *f(x)* changes sign across the zero.

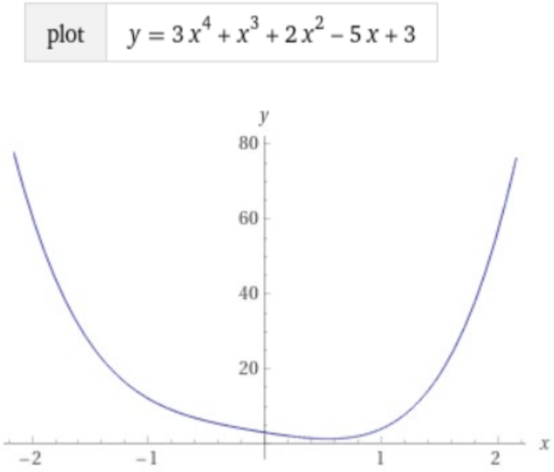


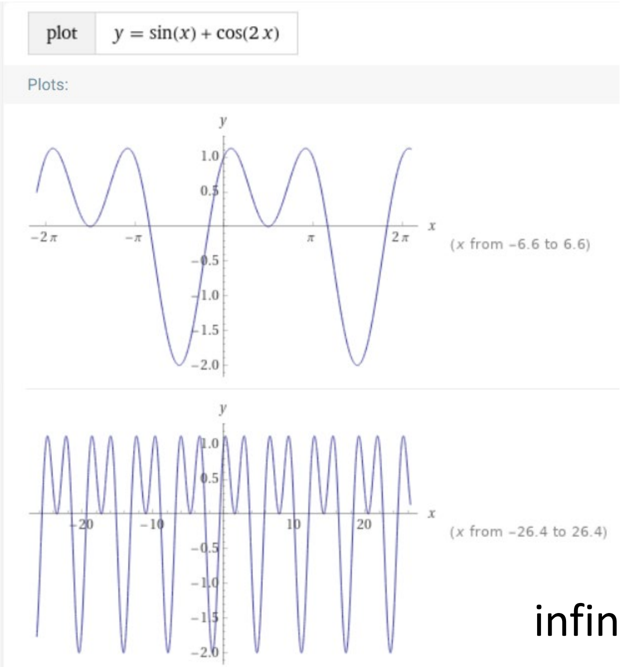
However, a simple generalization of this is not possible, because the number or existence of roots is not guaranteed except for special cases.



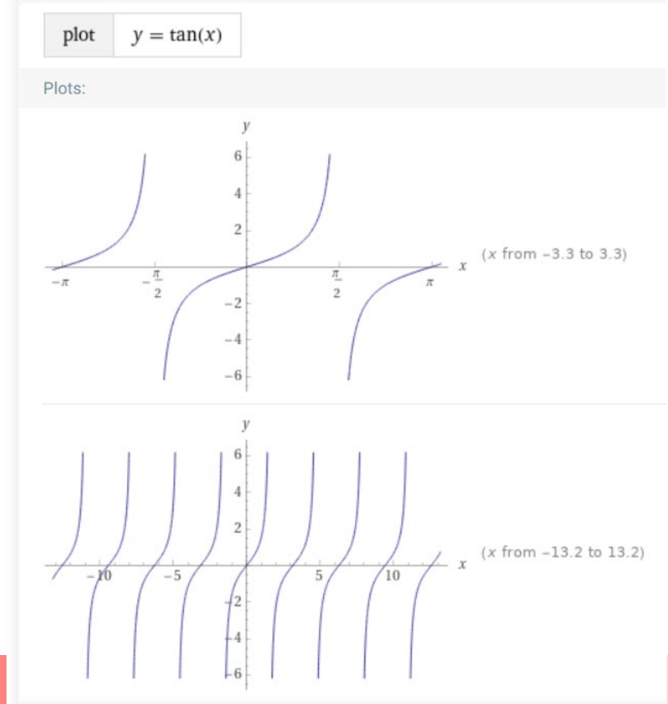
3 roots

No real roots





infinite roots



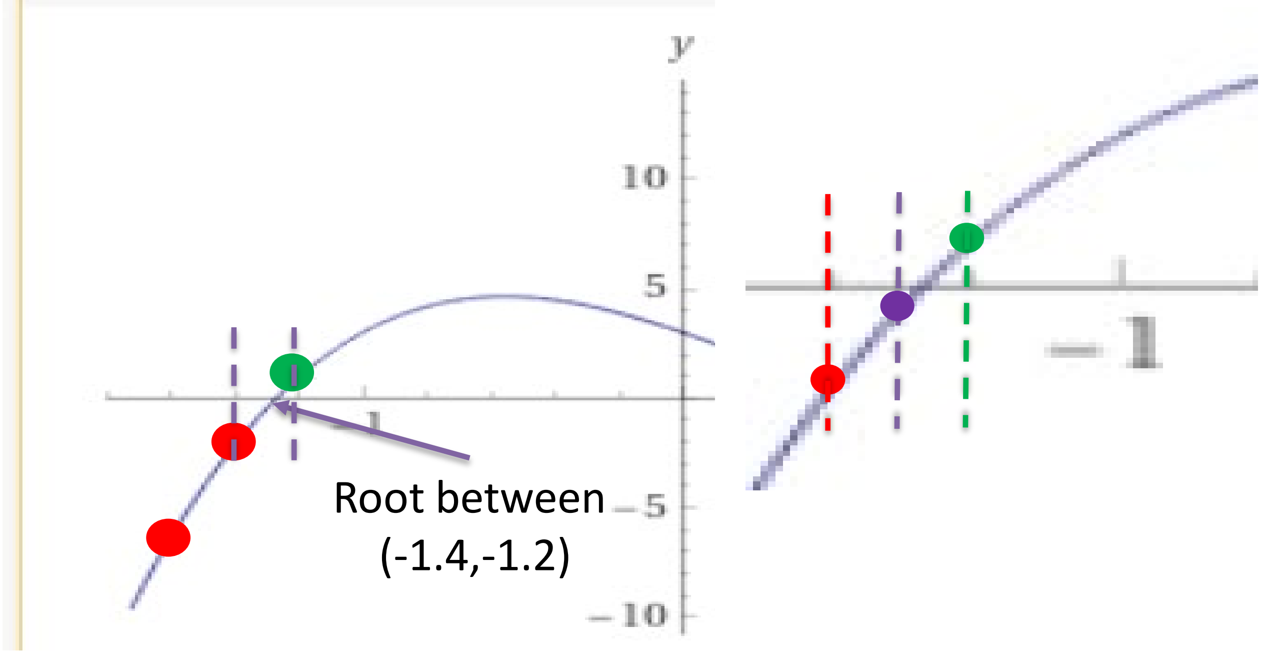
Bottom line: You need to know your function.

Methods

The first, most simple-minded approach is called the Incremental Search method. Basically, start at some point, check the sign, move to another point, check if sign changes.

Yes, root on interval

No, no root move on.



This gives an estimate of the root to at best Δx accuracy. To get better, reduce Δx and start again at beginning of interval to narrow the result.

This works but is super inefficient.

rootsearch is the code in the text.

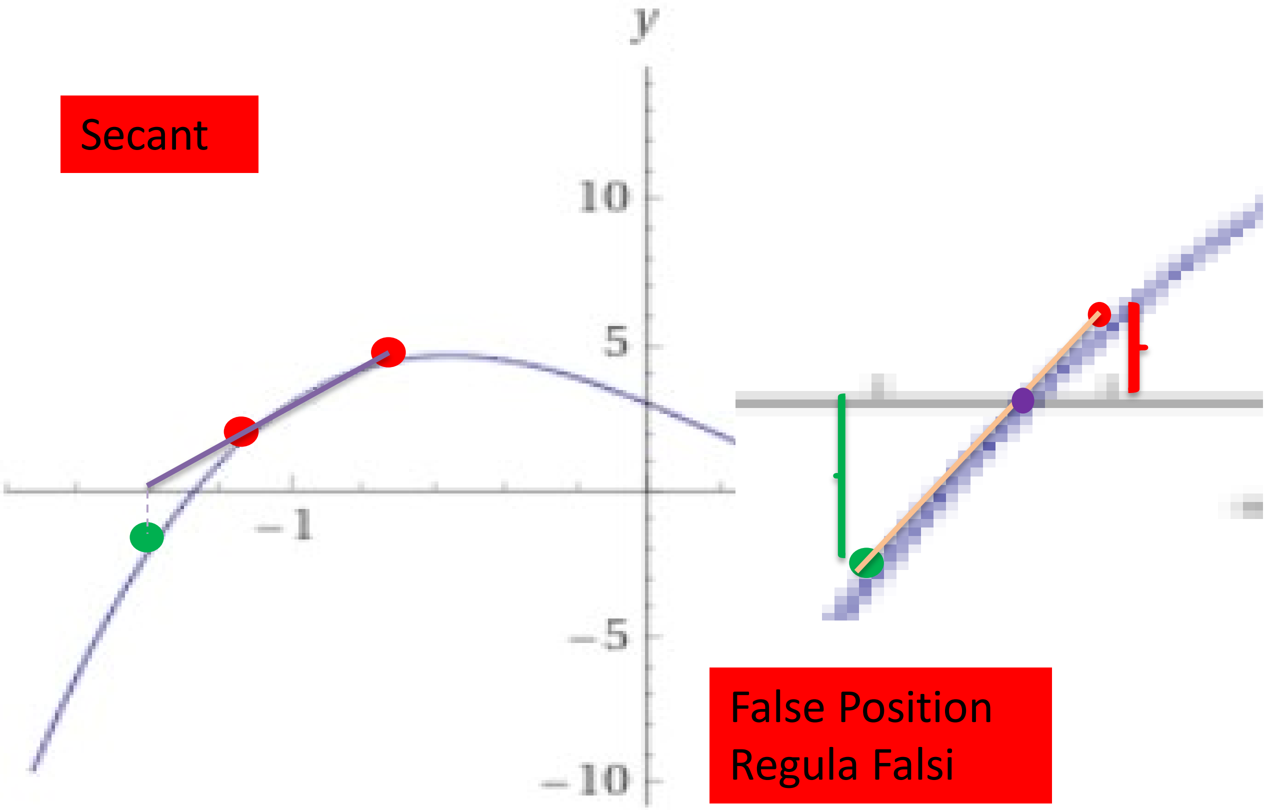
Bisection is a strategy to narrow down the root faster. Once the root is bounded, compute the value at the halfway point

- If $f(x)=0$ (to some tol) you're done
- If not, use new guess to replace previous guess with same sign

bisection is the code in the text

Linear Interpolation-Based Methods

If we have 2 guesses for a root, the Secant and False Position Methods give a new guess.



Another interpolation-based approach is called Ridder's Method, it scales *f(x)* by an exponential function $e^{(x-x1)Q}$, so that

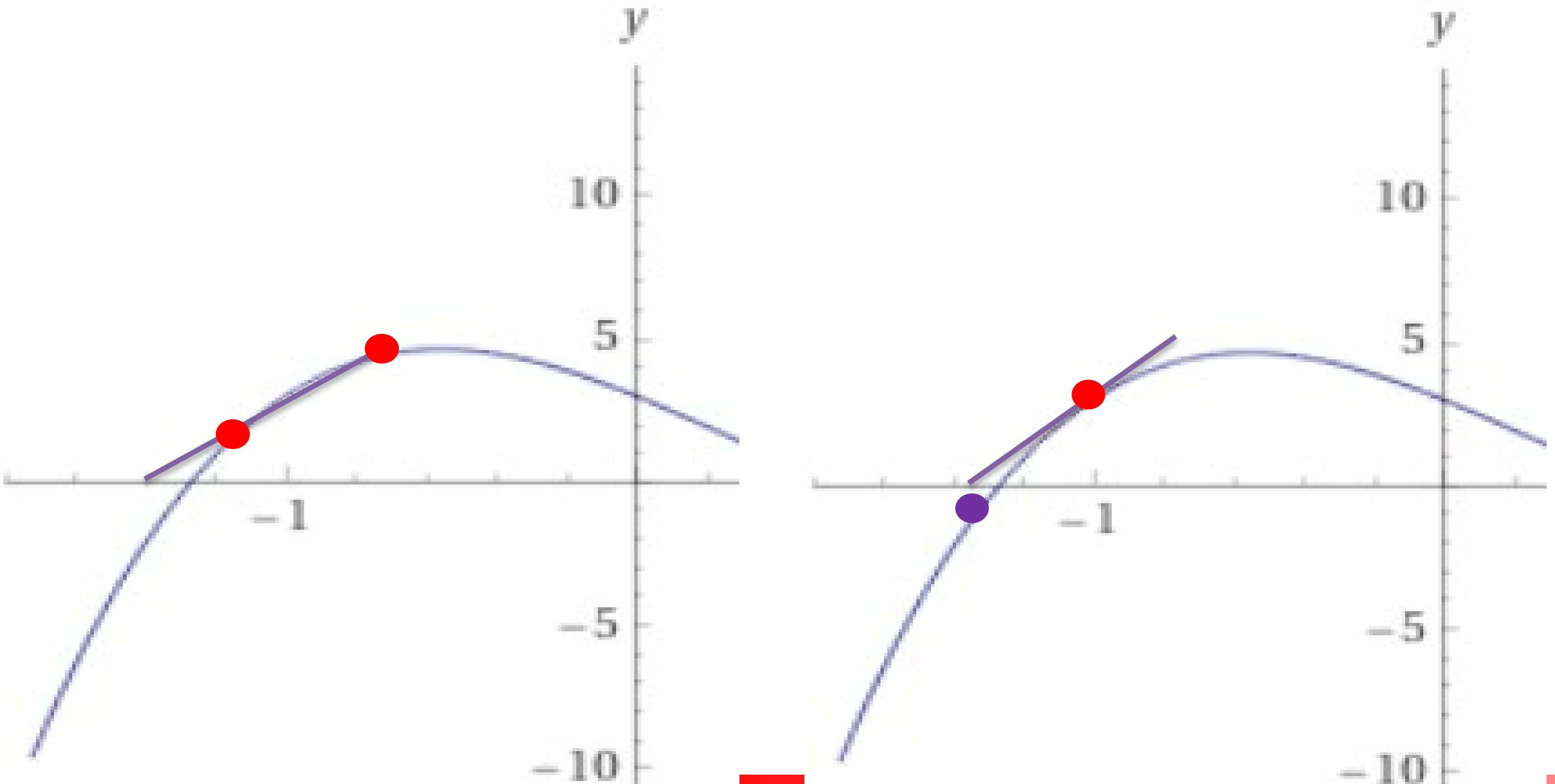
$$g(x) = f(x)e^{(x-x_1)Q} \qquad (5.1\text{-}4)$$

and then applies the false position (linear interpolation) method to the new variable.

The text provides only the code for this method called ridder.

Newton-Raphson Method

The secant method provides an estimate of the derivate between the two red points and suggests that an even better approach would be to use a single point together with its derivative called the Newton-Raphson method, text code newtonRaphson.



Since

$$f'(x_i) \approx \frac{f(x_i) - f(x)}{x_i - x}$$

& we want $f(x) = 0$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (5.1\text{-}5)$$

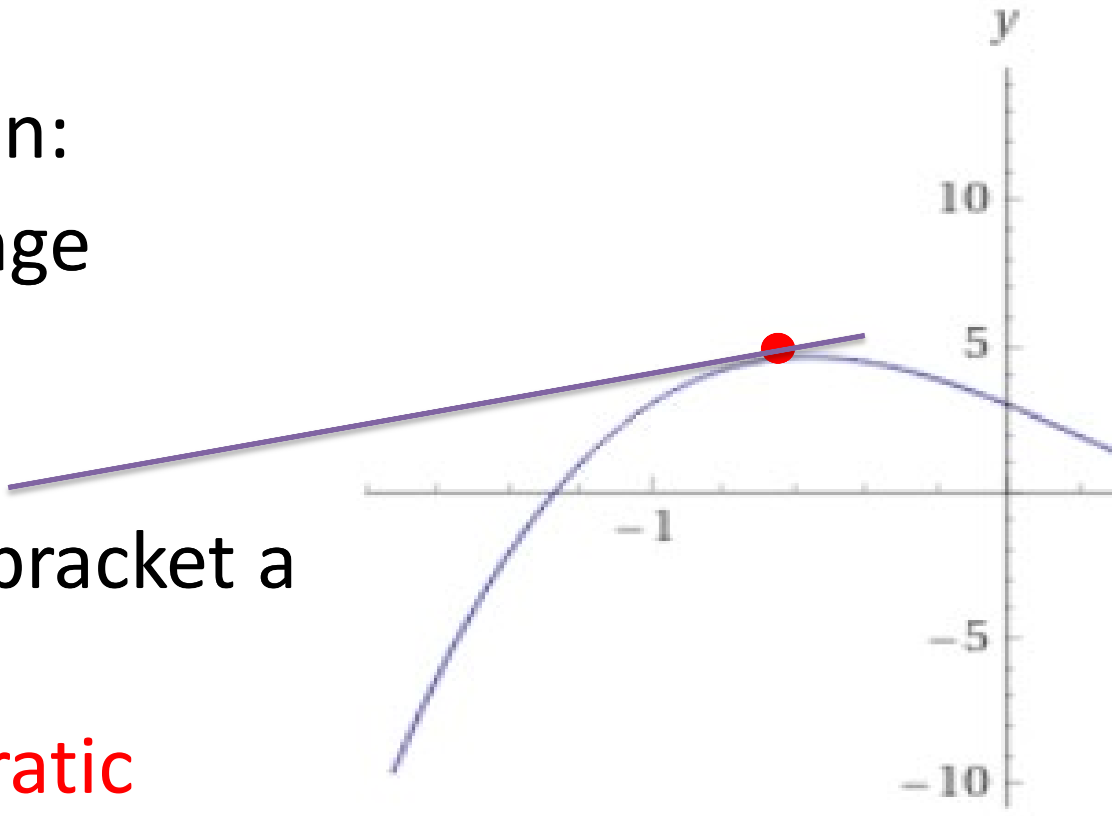
Limitations and Advantages

Slope type methods are limited when:

- slopes are small, causing *x* to change greatly,
- rapid solution changes occur,

The best approach, in general, is to bracket a root.

A major advantage of NR is its quadratic convergence, meaning the number of significant digits effectively doubles each iteration, when close to the root.



Shock-Deflection-Mach Equation

$$tan\theta = 2cot\beta\left[\frac{M_1^2 sin^2\beta - 1}{M_1^2(\gamma + cos2\beta) + 2}\right]$$

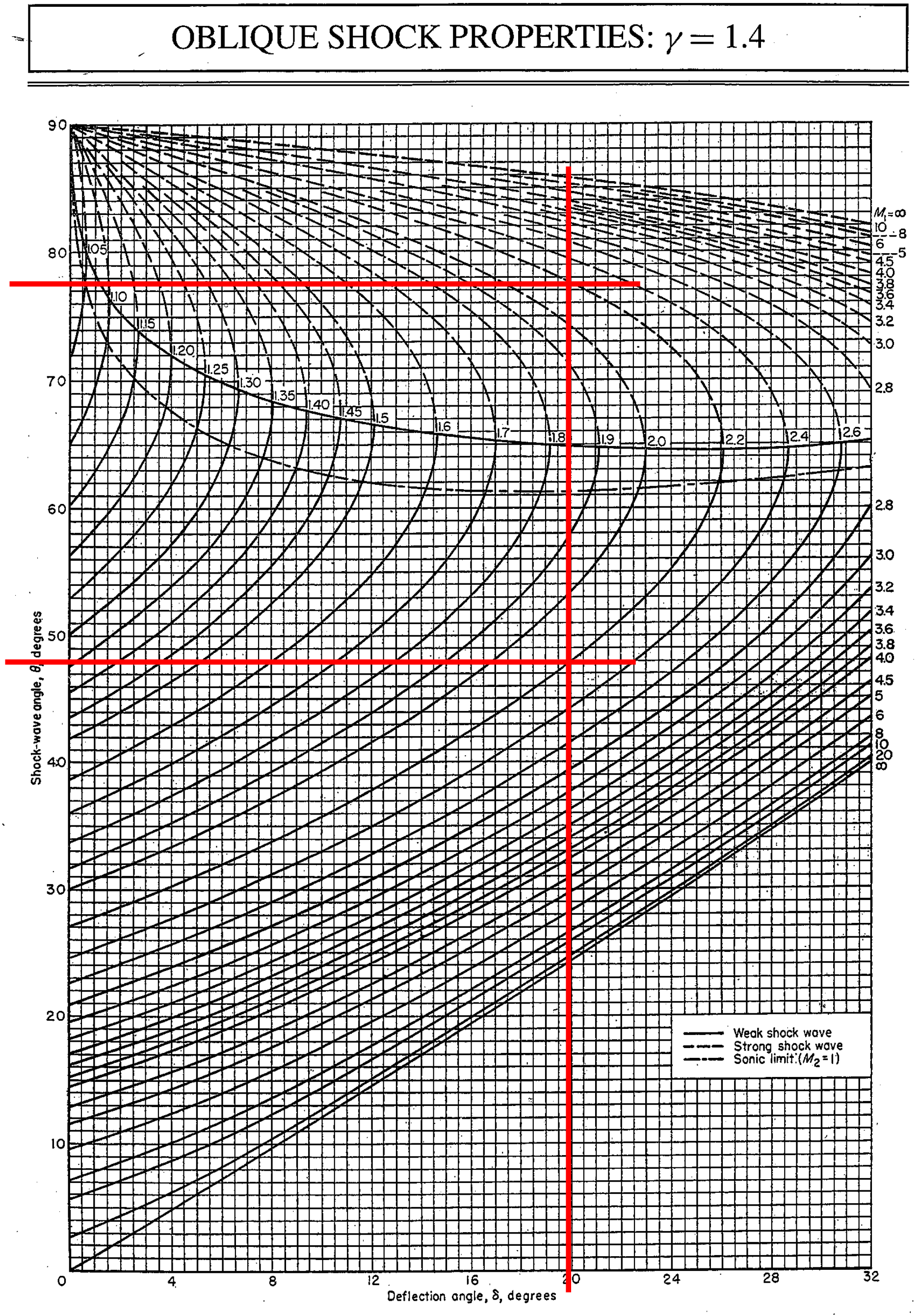
$\theta - \beta - M$ Equation

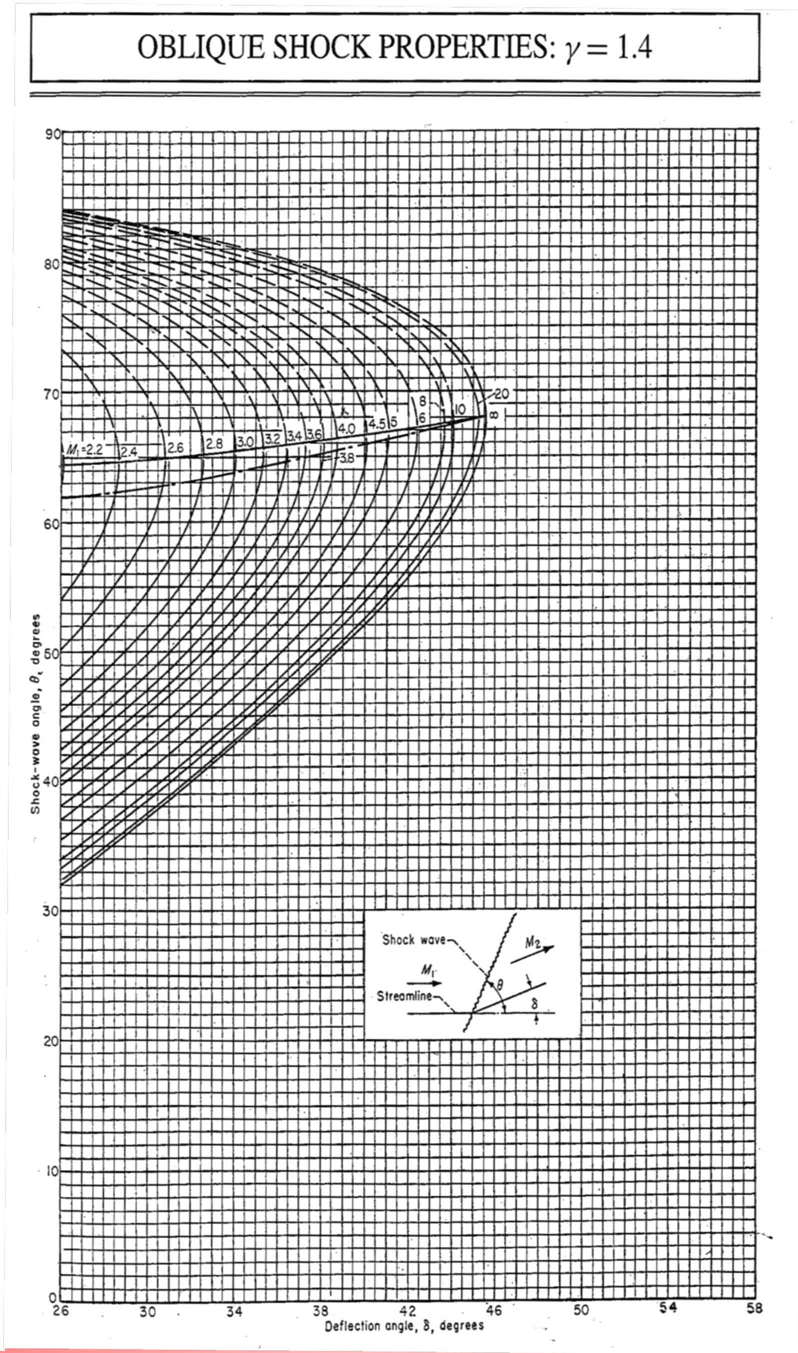
(5.1-6)

We often know θ and M and need to find β, in which case the equation is transcendental and has two values.

In cases like this it is often a good idea to start with initial guesses at the extremes, i.e., 0 or 90 degrees depending on whether we want a weak or strong shock solution.

Example for $\theta = 20^o, M = 2.2$





Unit 5.2 – Root Finding for Systems of Equations

Can our root finding techniques generalize to systems of nonlinear equations of a form like

$$[A]\vec{x} = \vec{b} \qquad (2.1\text{-}14)$$

but where *[A]* and *b* both functions of $\vec{x}$.

$$\vec{f}(\vec{x}) = [A(\vec{x})]\vec{x} - \vec{b}(\vec{x}) = 0 \qquad (5.2\text{-}1)$$

The vector analog of the scalar Newton-Raphson method is obtained by using Taylor series expansions

$$f_i(\vec{x} + \Delta\vec{x}) = f_i(\vec{x}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} \Delta x_j + O(\Delta x^2) \qquad (5.2\text{-}2)$$

Approximate by

$$\vec{f}(\vec{x} + \Delta\vec{x}) \approx \vec{f}(\vec{x}) + \vec{J}(\vec{x})\Delta\vec{x} \qquad (5.2\text{-}3)$$

Where $\vec{J}(\vec{x})$ is called the Jacobian matrix defined by

$$J_{ij} = \frac{\partial f_i}{\partial x_j} \qquad (5.2\text{-}4)$$

Eq. 5.2-4 can be calculated analytically or by finite differences. More importantly, since we seek $\vec{f}(\vec{x} + \Delta\vec{x}) = 0$

We can solve

$$\vec{J}(\vec{x})\Delta\vec{x} = -\vec{f}(\vec{x}) \qquad (5.2\text{-}5)$$

And update $\vec{x}$ via the change vector $\Delta\vec{x}$.

The text provides code newtonRaphson2 for systems of equations and employs a finite difference form of the Jacobian. That code can be easily modified using analytical functions to write the derivatives of function $\vec{f}(\vec{x})$ with respect to $\vec{x}$ if we prefer.

Once again, the tremendous advantage of this technique is quadratic convergence as measured by a norm of the $\Delta\vec{x}$ vector.

An example of this comes from several publications by your professor where he applied the technique to the Navier-Stokes equations and demonstrated quadratic convergence.

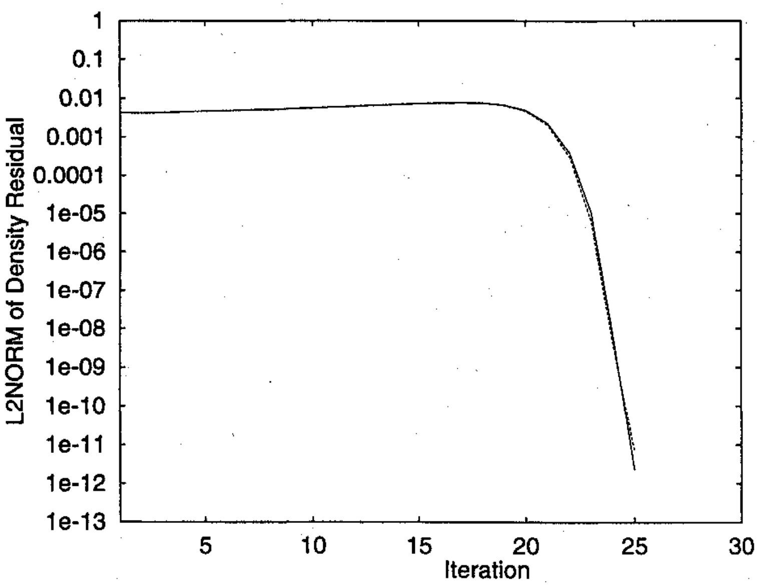


Fig. 1 40 × 40 flat plate at Mach = 2.0, second-order spatial discretization, and initial time step of 1000: ——, analytical and - - - -, numerical.

6400x6400 matrix

Unit 5.3 –Finding Zeros of Polynomials

The problem of finding the roots of an equation involves determining when the function obtains zero values. There is a rich set of literature for understanding this problem when applied to polynomials of the form:

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad (5.3\text{-}1)$$

Where the a_i coefficients could be real or complex numbers. An *n-th* degree polynomial can be shown to have *n* roots, however, they can be real, complex, or repeated.

Real roots

$$f(x) = 1 - x^2 = (x-1)(x+1) \quad (5.3\text{-}2)$$

$x_1 = 1$ and $x_2 = -1$.

Complex roots

$$f(x) = 1 + x^2 = (x-i)(x+i) \quad (5.3\text{-}3)$$

$x_1 = i$ and $x_2 = -i$.

Repeated roots

$$f(x) = 1 - 2x + x^2 = (x-1)^2 \quad (5.3\text{-}4)$$

$x_1 = 1$ and $x_2 = 1$.

If we find a root of P_n we can reduce the order of the polynomial by employing synthetic division, a technique you probably learned in high school at some point.

Polynomial Deflation

Suppose we have

$$f(x) = x^4 + x^3 - 4x^2 - 5x + 2$$

and use the *x=2* root to deflate:

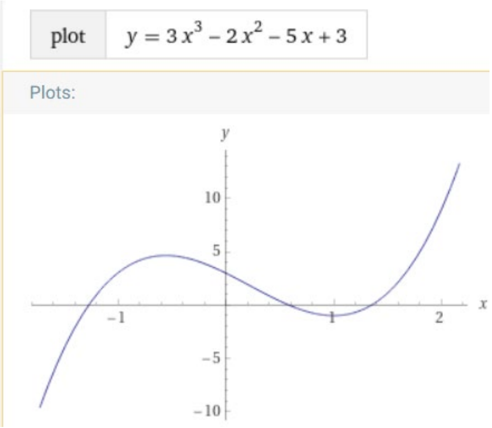
Rule of Descartes

You can estimate the number of positive and negative real roots (but not find them) by the Rule of Descartes.

- *n* roots – could be real or complex
- Complex come in conjugate pairs $(x_r + ix_i, x_r - ix_i)$
- # +ve real roots = # sign changes in $P_n(x)$ or less by an even #
- # -ve real roots = # sign changes in $P_n(-x)$ or less by an even #

Rule of Descartes

Using our previous example polynomial, we see 2 sign changes for $P_n(x)$.



$$P_n(-x) = 3(-x)^3 - 2(-x)^2 - 5(-x) + 3$$

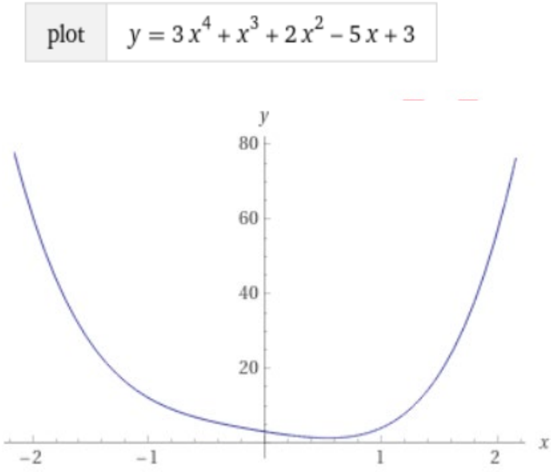
Or

$$P_n(-x) = -3(x)^3 - 2(x)^2 + 5(x) + 3$$

So only 1 sign change. Therefore, we might expect:

- 2 or 0 real +ve roots
- 1 real –ve root

The other example



2 or 0 real +ve roots

$$P_n(-x) = 3x^4 - x^3 + 2x^2 + 5x + 3$$

2 or 0 real –ve roots

So possibilities for (real+,real-,complex) are

(2,2,0),(2,0,2*),(0,2,2*),(0,0,4*)

A reasonable strategy for polynomial root search is

1. Plot the function to bracket roots.
2. Find a root and deflate the polynomial.
3. Use quadratic and cubic equations once the polynomial is reduced.