

Minimization, root finding, and solutions to ODEs
PHYS 250 (Autumn 2019) – Lecture 9

David Miller

Department of Physics and the Enrico Fermi Institute
University of Chicago

October 29, 2019

Outline

- 1 *Connection between minimization and solving differential equations*
 - Reminders from Lecture 8
 - Categories of problem
 - Specific example

- 2 *Finding roots*
 - Bisection method
 - Newton's method
 - System of equations

Reminders from last time

There are many examples of optimization in physics

Examples of minimization

- **Fitting a model to data:** minimize differences between a model and data
- **Second law of thermodynamics:** minimize changes in entropy for a system in thermodynamic equilibrium
- **Conservation of momentum:** establish mechanical equilibrium by minimizing changes in momentum, $\frac{d\vec{p}}{dt} = 0$
- **Principle of least action:** obtain the equations of motion of a system by minimizing (or maximizing!) the variations of the action, S
- **Path integral formulation of quantum mechanics:** sort quantum mechanically possible trajectories by minimizing quantum action
- **Ising model:** minimization the energy of the spin configurations

Optimization and root finding

Ultimately, I want to bring us to the point of discussing **optimization in the context of modeling** (c.f. Lecture 1!).

What that means, is that we want to have the computational tools to be able to **model data and approximate complex functions computationally**, since it is often **impossible to do so analytically**.

To achieve that, let's expand our tool kit beyond the analytically solvable example from last time (the 2D linear regression for χ^2 minimization).

Categorizing the problems

As we discussed in Lecture 8, an **optimization problem** quickly turns into an attempt to solve a first-order ordinary (ODE) or partial differential equation (PDE) in order to identify the value of the parameter α for which the function F is stationary:

$$F'(\alpha) = 0 \quad (1)$$

The approaches used to do this are closely related to the methods used for finding zeroes or “**roots**” of functions (since the derivative of a function is, of course, also a function!). But we need to differentiate the problems:

Initial vs. boundary value problems

Imagine you have the ODE

$$F'' = -F \quad (2)$$

- **Initial value problem:** $F(0) = 1, F'(0) = 0$
- **Boundary value problem:** $F(0) = 1, F(\pi) = 0$

The latter is often much harder!

Explicit root finding: square well potential

The Schrodinger equation

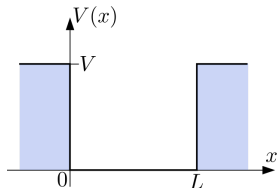
$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

Explicit root finding: square well potential

The Schrodinger equation

$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

tells us that for a finite square-well potential



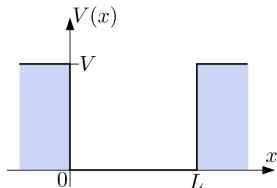
$$V(x) = \begin{cases} -V_0 & 0 < x < L \\ 0 & \text{else} \end{cases} \quad (4)$$

Explicit root finding: square well potential

The Schrodinger equation

$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

tells us that for a finite square-well potential



$$V(x) = \begin{cases} -V_0 & 0 < x < L \\ 0 & \text{else} \end{cases} \quad (4)$$

and a bound state ($E = -f < 0$, f positive definite) in the region $x < 0$, we must solve:

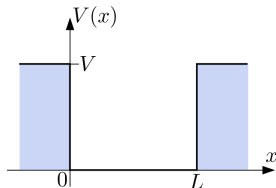
$$-\frac{\hbar^2}{2m}\psi''(x < 0) = -f\psi(x < 0) \quad (5)$$

Explicit root finding: square well potential

The Schrodinger equation

$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

tells us that for a finite square-well potential



$$V(x) = \begin{cases} -V_0 & 0 < x < L \\ 0 & \text{else} \end{cases} \quad (4)$$

and a bound state ($E = -f < 0, f$ positive definite) in the region $x < 0$, we must solve:

$$-\frac{\hbar^2}{2m}\psi''(x < 0) = -f\psi(x < 0) \quad (5)$$

The solution for $x < 0$ is

$$\psi(x < 0) = Ae^{\sqrt{z_0^2 - z^2}x}, \text{ where } z_0^2 - z^2 = \frac{2mf}{\hbar^2} \quad (6)$$

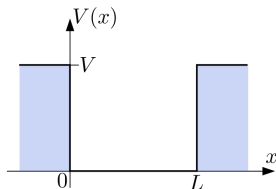
and $z_0^2 \equiv 2mV_0/\hbar^2$.

Explicit root finding: square well potential

The Schrodinger equation

$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

tells us that for a finite square-well potential



$$V(x) = \begin{cases} -V_0 & 0 < x < L \\ 0 & \text{else} \end{cases} \quad (4)$$

and a bound state ($E = -f < 0, f$ positive definite) in the region $x < 0$, we must solve:

$$-\frac{\hbar^2}{2m}\psi''(x < 0) = -f\psi(x < 0) \quad (5)$$

The solution for $x < 0$ is

$$\psi(x < 0) = Ae^{\sqrt{z_0^2 - z^2}x}, \text{ where } z_0^2 - z^2 = \frac{2mf}{\hbar^2} \quad (6)$$

and $z_0^2 \equiv 2mV_0/\hbar^2$.

In order to solve for the bound-state energies, we need to solve the following

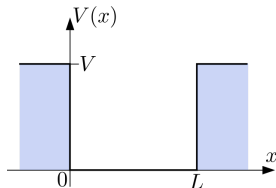
$$\tan(za) = \frac{2\sqrt{\left(\frac{z_0}{z}\right)^2 - 1}}{2 - \left(\frac{z_0}{z}\right)^2} \quad (7)$$

Explicit root finding: square well potential

The Schrodinger equation

$$\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (3)$$

tells us that for a finite square-well potential



$$V(x) = \begin{cases} -V_0 & 0 < x < L \\ 0 & \text{else} \end{cases}$$

and a bound state ($E = -f < 0, f$ positive definite) in the region $x < 0$, we must solve:

$$-\frac{\hbar^2}{2m}\psi''(x < 0) = -f\psi(x < 0) \quad (5)$$

The solution for $x < 0$ is

$$\psi(x < 0) = Ae^{\sqrt{z_0^2 - z^2}x}, \text{ where } z_0^2 - z^2 = \frac{2mf}{\hbar^2} \quad (6)$$

and $z_0^2 \equiv 2mV_0/\hbar^2$.

In order to solve for the bound-state energies, we need to solve the following

$$\tan(za) = \frac{2\sqrt{\left(\frac{z_0}{z}\right)^2 - 1}}{2 - \left(\frac{z_0}{z}\right)^2} \quad (7)$$

This is transcendental, requires numerical approaches to finding the “**root**” of

$$F(z) = \tan(za) - \frac{2\sqrt{\left(\frac{z_0}{z}\right)^2 - 1}}{2 - \left(\frac{z_0}{z}\right)^2} \quad (8)$$

Outline

- 1 *Connection between minimization and solving differential equations*
 - Reminders from Lecture 8
 - Categories of problem
 - Specific example

- 2 *Finding roots*
 - Bisection method
 - Newton's method
 - System of equations

Bisection method

The bisection method is straightforward and reliable.

Start with an interval defined by its endpoints, x_ℓ and x_r on the left and right of a starting point, such that

$$F(x_\ell)F(x_r) < 0 \quad (9)$$

That is, the function $F(x)$ has at least one root somewhere between x_ℓ and x_r . Then split that interval in half

$$x_m = \frac{1}{2}(x_\ell + x_r) \quad (10)$$

If the product $F(x_\ell)F(x_m) < 0$, then a root of $F(x)$ lies between x_ℓ and x_m , otherwise, it lies between x_m and x_r . Repeat until some tolerance ϵ is reached such that

$$|F(x_m)| \leq \epsilon \quad (11)$$

Bisection method

The bisection method is straightforward and reliable, **but it can be slow if you don't have a good initial guess for the interval.**

If there is no root in the chosen interval, then it does not matter what you do.

The method also uses no information whatsoever about the size of the step to take with each iteration. The overall time is determined **only** by the size of the initial interval (i.e. how good your **guess is**) and the tolerance, ϵ .

Bisection method

The bisection method is straightforward and reliable, **but it can be slow if you don't have a good initial guess for the interval.**

If there is no root in the chosen interval, then it does not matter what you do.

The method also uses no information whatsoever about the size of the step to take with each iteration. The overall time is determined **only** by the size of the initial interval (i.e. how good your **guess is**) and the tolerance, ϵ .

We can do better!

Newton's method (I)

The fundamental form of the method alluded to in the last lecture is **Newton's method**. The core of this approach is to use information about the function in order to inform how **big** of a step size to take with each iteration.

Suppose a function $F(x)$ has a root $x = x_0$ and you are within δ of that root at $x = x_0 - \delta$. The question we want to answer is:

How far away from the root are we actually? What is the value of δ ?

Using a Taylor expansion (as Wah told us!)

$$F(x_0) = F(x + \delta) \approx F(x) + \delta F'(x) + \frac{1}{2}\delta^2 F''(x) + \mathcal{O}(\delta^3) \quad (12)$$

By setting $F(x_0) = 0$ by assumption, we can make the linear approximation to $\delta \approx \Delta$ as

$$\Delta = -\frac{F(x)}{F'(x)} \quad (13)$$

Newton's method (II)

Therefore, we can choose a starting point x_i and then update x according to Newton

$$x_{i+1} = x_i - \frac{F(x_i)}{F'(x_i)} \quad (14)$$

The iteration stops after j iterations when

$$|x - x_j| \leq \epsilon \quad (15)$$

Precision of Newton's method

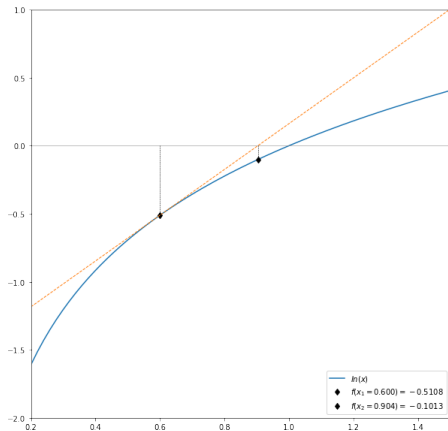
For any estimate x_i of the method, the error, E_i is the difference between the true root x and the estimate:

$$E_i = x - x_i \quad (16)$$

Merely by inspecting the design of Newton's method, you can see that the precision of the estimate for a subsequent iteration will be given by

$$E_{i+1} = E_i + \frac{F(x)}{F'(x)} \quad (17)$$

$$= -\frac{F''(x)}{2F'(x)} E_i^2 \quad (18)$$

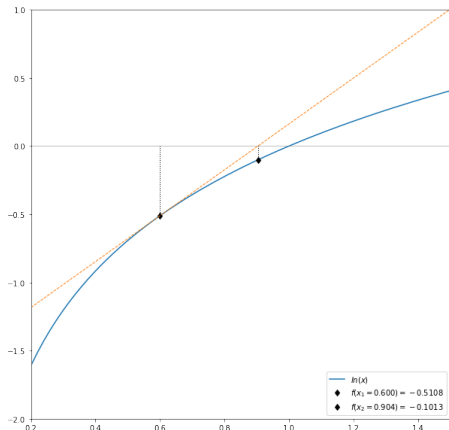


Convergence of Newton's method

Consequently, Newton's method
converges quadratically

- the error is the square of the error in the previous step)
- the number of significant figures is roughly doubled in every iteration, provided that x_i is **sufficiently** close to the root.

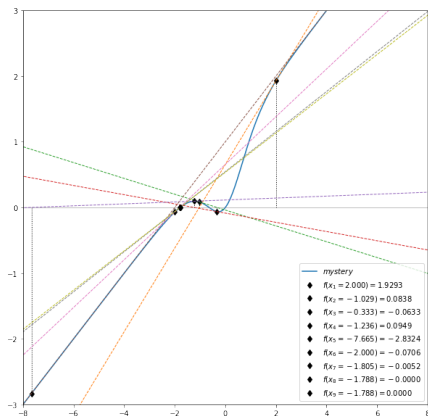
However, a critical assumption is that $F'(x) \neq 0$; for all $x \in I$, where I is the interval $[x - r, x + r]$ for some $r \geq |x - x_0|$ and x is the true root and x_0 was the starting point.



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I cooked up (since you need to do something similar on your homework):

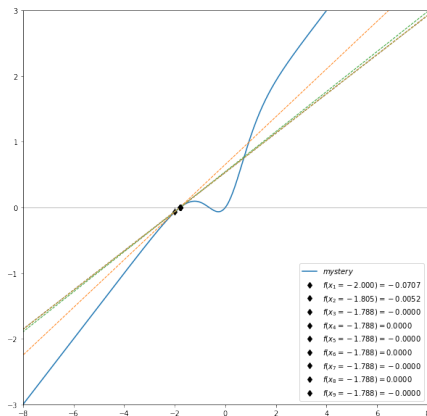
- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I
cooked up (since you need to do
something similar on your homework):

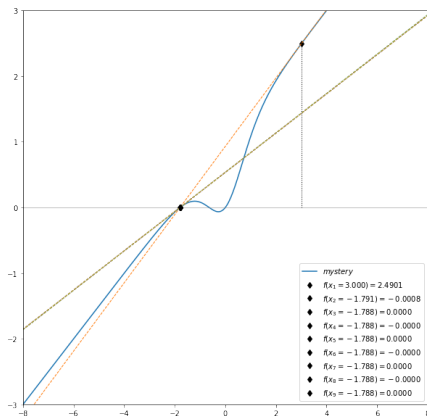
- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I
cooked up (since you need to do
something similar on your homework):

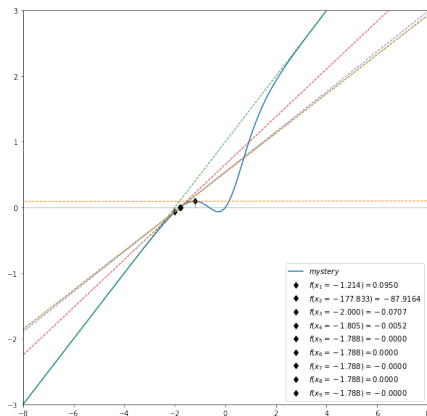
- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I
cooked up (since you need to do
something similar on your homework):

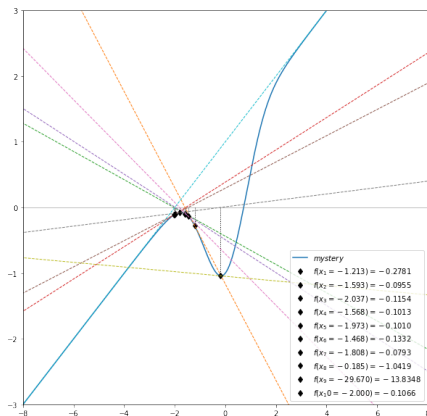
- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I
cooked up (since you need to do
something similar on your homework):

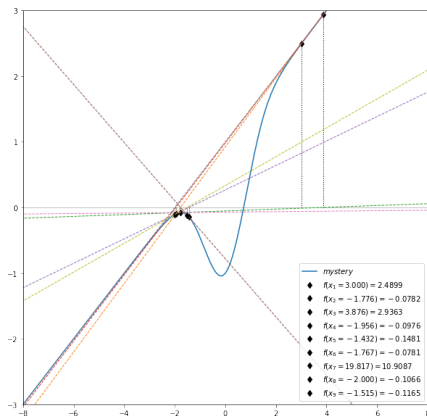
- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Pathologies and divergent scenarios

That is definitely not always the case.
Let's look at a pathological example.
Here is a fun mystery function that I
cooked up (since you need to do
something similar on your homework):

- $x_0 = 2.000$, **7 iterations**
- $x_0 = -2.000$, **2 iterations**
- $x_0 = 3.000$, **2 iterations**
- $x_0 = -1.214$, **4 iterations**
- Slight modification:
 $x_0 = -1.213$, **no convergence**
- Slight modification: $x_0 = 3.000$,
no convergence



Backtracking

In the last examples above we have a case where the search falls into the pathology of a situation where the initial guess was not **sufficiently close** to the root. an “infinite” loop without ever getting there.

A solution to this problem is called **backtracking**.

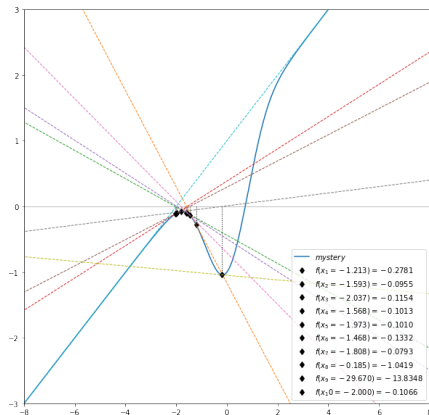
Backtracking

In cases where the new guess $x_0 + \Delta x$ leads to an increase in the magnitude of the function, $|f(x_0 + \Delta x)|^2 > |f(x_0)|^2$, you should backtrack somewhat and try a smaller guess, say, $x_0 + \Delta x/2$. If the magnitude of f still increases, then you just need to backtrack some more, say, by trying $x_0 + \Delta x/4$ as your next guess, and so forth.

Pathological case fixed with backtracking

Fixing the pathological example with backtracking:

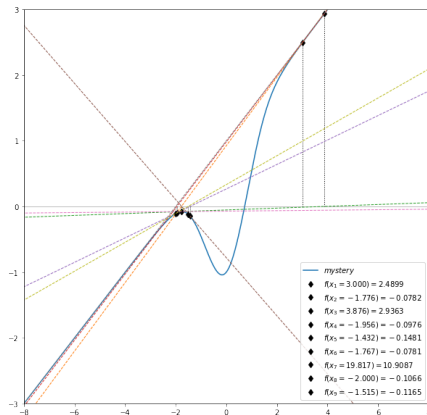
- $x_0 = -1.213$, **no convergence**
- $x_0 = 3.000$, **no convergence**
- $x_0 = 1.500$, **3 iterations**



Pathological case fixed with backtracking

Fixing the pathological example with backtracking:

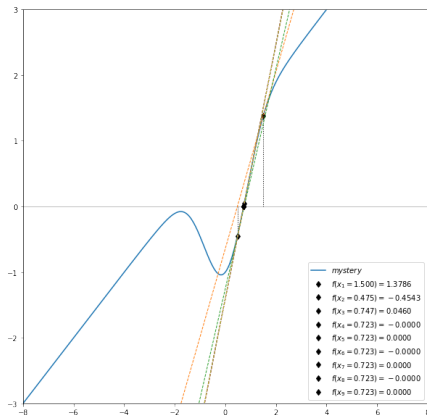
- $x_0 = -1.213$, **no convergence**
- $x_0 = 3.000$, **no convergence**
- $x_0 = 1.500$, **3 iterations**



Pathological case fixed with backtracking

Fixing the pathological example with backtracking:

- $x_0 = -1.213$, **no convergence**
- $x_0 = 3.000$, **no convergence**
- $x_0 = 1.500$, **3 iterations**



Multidimensional problems

Up to this point, we have confined our attention to solving the single equation $F(x) = 0$. Let us now consider the n -dimensional version of the same problem, namely

$$\vec{F}(\vec{x}) = 0 \quad (19)$$

where we allow for a vector of functions $\vec{F} = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})\}$, and $\vec{x} = \{x_1, x_2, \dots, x_n\}$.

The solution of n simultaneous, nonlinear equations is a much more formidable task than finding the root of a single equation. The trouble is the lack of a reliable method for bracketing the solution vector \vec{x} . Therefore, we cannot always provide the solution algorithm with a good starting value of x , unless such a value is suggested by the physics of the problem.

Newton's method is the workhorse here!

Reminder of the general problem

Start by considering each one of the n functions, $f_n(x)$, separately:

$$f_i(\vec{x}) = f_i(\vec{a}) + \sum_j^n \frac{\partial f_i}{\partial x_j} \Big|_{\vec{x}} \Delta x_j + \frac{1}{2!} \sum_j^n \sum_k^n \frac{\partial^2 f_i}{\partial_j \partial_k} \Big|_{\vec{x}} \Delta x_j \Delta x_k \quad (20)$$

$$= f_i(\vec{a}) + \vec{\nabla} f_i(\vec{a}) \cdot \vec{\Delta x} + \frac{1}{2!} \vec{\Delta x}^T \mathbf{H}(\vec{a}) \vec{\Delta x} \quad (21)$$

where \mathbf{H} is the **Hessian matrix**, describing the **curvature** of $f_i(\vec{x})$ by

$$\mathbf{H}_{j,k} = \frac{\partial^2 f(\vec{a})}{\partial x_j \partial x_k} \quad (22)$$

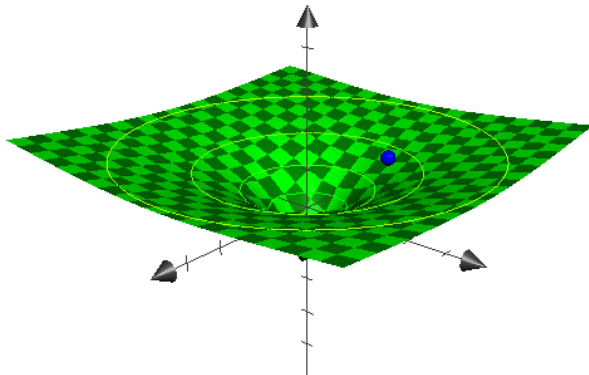
(The determinant of \mathbf{H} is also sometimes referred to as **the Hessian**) and $\vec{\Delta x}$ is a vector that describes the distance from the point about which we are expanding the function $f_i(\vec{x})$.

Specific example (I)

Let's take a very explicit example of a function:

$$f(\vec{x}) = f(x, y) = \ln \sqrt{x^2 + y^2} \quad (23)$$

and compute the gradient and Hessian matrix at $x = -2, y = 1$ (see point on graph below).



Specific example (II)

First, we calculate the gradient:

$$\frac{\partial f}{\partial x} = \frac{1}{\sqrt{x^2 + y^2}} \left(\frac{1}{2} \frac{2x}{\sqrt{x^2 + y^2}} \right) \quad (24)$$

$$= \frac{x}{x^2 + y^2} \quad (25)$$

$$\frac{\partial f}{\partial y} = \frac{y}{x^2 + y^2} \quad (26)$$

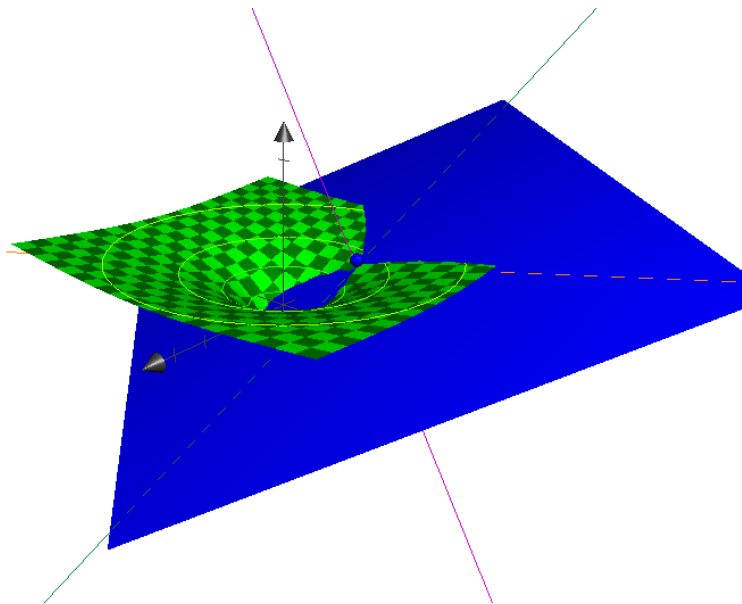
Therefore

$$\vec{\nabla} f(x, y) = \left(\frac{x}{x^2 + y^2}, \frac{y}{x^2 + y^2} \right) \quad (27)$$

Or, in matrix notation

$$\vec{\nabla} f(x, y) = \begin{bmatrix} \frac{x}{x^2 + y^2} \\ \frac{y}{x^2 + y^2} \end{bmatrix}, \quad \vec{\nabla} f(-2, 1) = \begin{bmatrix} -0.4 \\ 0.2 \end{bmatrix} \quad (28)$$

Specific example (III)



Specific example (IV)

Next, we calculate the Hessian matrix:

$$\frac{\partial^2 f}{\partial x^2} = \frac{(x^2 + y^2) - x(2x)}{(x^2 + y^2)^2} \quad (29)$$

$$= \frac{-x^2 + y^2}{(x^2 + y^2)^2} \quad (30)$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{x^2 - y^2}{(x^2 + y^2)^2} \quad (31)$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial x \partial y} = \frac{-2xy}{(x^2 + y^2)^2} \quad (32)$$

Therefore

$$\mathbf{H} = \begin{bmatrix} -x^2 + y^2 & -2xy \\ -2xy & x^2 - y^2 \end{bmatrix} \frac{1}{(x^2 + y^2)^2}, \mathbf{H} = \begin{bmatrix} -0.12 & 0.16 \\ 0.16 & 0.12 \end{bmatrix} \quad (33)$$

Jacobian matrix

In the case that the function F is a vector of functions,
 $\vec{F} = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})\}$ (i.e. a system of equations) then the gradient **also**
has a name: Jacobian.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \vec{f}}{\partial x_1} & \cdots & \frac{\partial \vec{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (34)$$

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j} \quad (35)$$

The Jacobian matrix is important because if the function \vec{f} is differentiable at a point \vec{x} , then \mathbf{J} defines a linear map which is the best (pointwise) linear approximation of the function \vec{f} near the point \vec{x} .

That's exactly what we want!

Newton's method for a system of equations

The following steps constitute Newton's method for simultaneous, nonlinear equations:

- 1 Estimate the solution vector \vec{x}
- 2 Evaluate $\vec{f}(\vec{x})$
- 3 Compute the Jacobian matrix \mathbf{J} (J_{ij})
- 4 Setup the simultaneous equations $\mathbf{J}(\vec{x})\vec{\Delta x} = -\vec{f}(\vec{x})$ and solve for \vec{x}
- 5 Let $\vec{x} \leftarrow \vec{x} + \vec{\Delta x}$ and repeat steps 2-5

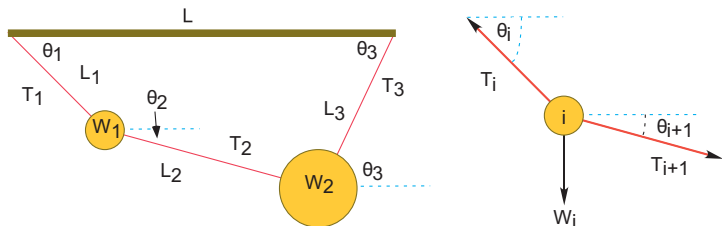
Comments on matrix computing and manipulations

Physical systems are often modeled by systems of simultaneous equations, and it is very convenient to write these matrix form. In fact, several physical systems can be expressed this way very conveniently.

- Physical optics (lenses, mirrors, etc)
- Electromagnetic waves propagating near boundaries and in matter
- Quantum mechanical systems
- Classical mechanical dynamical systems
- etc, etc

As the models are made more realistic, the matrices often become large, and computers become an excellent tool for solving such problems.

Example: weights and strings (I)



Two weights $(W_1, W_2) = (10, 20)$ are hung from three pieces of string with lengths $(L_1, L_2, L_3) = (3, 4, 4)$ and a horizontal bar of length $L = 8$.

We want to find the angles assumed by the strings and the tensions exerted by the strings.

Example: weights and strings (II)

The first 5 equations implement the geometric constraints that the horizontal length of the structure is L and that the strings begin and end at the same height.

$$L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_3 \cos \theta_3 = L \quad (36)$$

$$L_1 \sin \theta_1 + L_2 \sin \theta_2 - L_3 \sin \theta_3 = 0 \quad (37)$$

$$\sin^2 \theta_1 + \cos^2 \theta_1 = 1 \quad (38)$$

$$\sin^2 \theta_2 + \cos^2 \theta_2 = 1 \quad (39)$$

$$\sin^2 \theta_3 + \cos^2 \theta_3 = 1 \quad (40)$$

The last three equations include trigonometric identities as independent equations because we are treating $\sin \theta$ and $\cos \theta$ as independent variables; this makes the search procedure easier to implement.

Example: weights and strings (III)

The basics physics says that since there are no accelerations, the sum of the forces in the horizontal and vertical directions must equal zero.

$$T_1 \sin \theta_1 - T_2 \sin \theta_2 - W_1 = 0 \quad (41)$$

$$T_1 \cos \theta_1 - T_2 \cos \theta_2 = 0 \quad (42)$$

$$T_2 \sin \theta_2 + T_3 \sin \theta_3 - W_2 = 0 \quad (43)$$

$$T_2 \cos \theta_2 - T_3 \cos \theta_3 = 0 \quad (44)$$

Here W_i is the weight of mass i and T_i is the tension in string i . Note that since we do not have a rigid structure, we cannot assume the equilibrium of torques.

These equations represent nine simultaneous nonlinear equations. While linear equations can be solved directly, nonlinear equations cannot.

Example: weights and strings (IV)

We apply to our set the same Newton's method algorithm as used to solve a single equation by renaming the nine unknown angles and tensions as the components of our vector \vec{x} and placing the variables together as a vector:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \sin \theta_1 \\ \sin \theta_2 \\ \sin \theta_3 \\ \cos \theta_1 \\ \cos \theta_2 \\ \cos \theta_3 \\ T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

Example: weights and strings (V)

The nine equations to be solved are written in a general form with zeros on the right-hand sides and placed in a vector:

$$f(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ f_3(\vec{x}) \\ f_4(\vec{x}) \\ f_5(\vec{x}) \\ f_6(\vec{x}) \\ f_7(\vec{x}) \\ f_8(\vec{x}) \\ f_9(\vec{x}) \end{bmatrix} = \begin{bmatrix} L_1x_4 + L_2x_5 + L_3x_6 - 8 \\ L_1x_1 + L_2x_2 - L_3x_3 \\ x_7x_1 - x_8x_2 - W_1 \\ x_7x_4 - x_8x_5 \\ x_8x_2 + x_9x_3 - W_2 \\ x_8x_5 - x_9x_6 \\ x_1^2 + x_4^2 - 1 \\ x_2^2 + x_5^2 - 1 \\ x_3^2 + x_6^2 \end{bmatrix} = \begin{bmatrix} 3x_4 + 4x_5 + 4x_6 - 8 \\ 3x_1 + 4x_2 - 4x_3 \\ x_7x_1 - x_8x_2 - 10 \\ x_7x_4 - x_8x_5 \\ x_8x_2 + x_9x_3 - 20 \\ x_8x_5 - x_9x_6 \\ x_1^2 + x_4^2 - 1 \\ x_2^2 + x_5^2 - 1 \\ x_3^2 + x_6^2 \end{bmatrix}$$

And this is then solved by computing the Jacobian and using Newton's method as usual!