Computer Modeling and Finite Difference Methods

Andrew D. Wickert

Last updated March 30, 2017

1 Philosophy of Computer Modeling

Before starting to write a program to solve a problem, one must ask oneself important questions such as,

- Why do I want to model this system?
  - To test relationships between variables and gain some new physical insight that would be difficult to attain with reasoning or analytical solutions alone?
  - To accurately represent a real physical system, including efforts to try to match data and their associated uncertainties?
  - To make a prediction of the future and/or past?
  - Other?
- What is the best way to model it?
  - As a set of continuum equations?
  - As a number of individual agents that take actions in their virtual environment?
  - With full-realism physics or simplified rules?
  - Other?
- Should I wish to, how do I compare the "real world" and the data? Or perhaps improve the fit between data and model?

The first algorithm

The first computer program was written in 1843 (or maybe 1842) by Lady Ada Lovelace, with a numerical method to calculate the Bernoulli Numbers. She was 28!

2 Mathematics Review

2.1 Vectors

A vector is a one-dimensional matrix. It is thus a special case of the more general class of matrices. It may be a row,

$$\vec{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}, \tag{1}$$

or a column,

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}. \tag{2}$$

Vectors can define a direction. For example, in three dimensions:

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3}$$

In addition to the top arrow notation, a boldface (but italicized) notation is also common for vectors:

$$\vec{a} = \boldsymbol{a} \tag{4}$$

2.2 Two-dimensional matrices

A two-dimensional matrix of m rows and n columns can be represented in a way that is an extension of our vector notation; these are represented by boldface Roman (non-italic) characters. They are often, but not always, represented by capital letters:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{5}$$

A 3×3 matrix is often used to represent a *tensor* that can define the orientation and dimensions in a three-dimensional shape (ellipsoid). Such a matrix can also describe stress-strain relationships, mineral symmetry, or any other material anisotropy (or isotropy, if the 3×3 matrix is the identity matrix).

One special matrix is the *Identity* matrix, which is an expansion of "1" for an array:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{6}$$

2.3 Matrix operations

2.3.1 Matrix multiplication

To multiply two matrices, $\mathbf{A} \times \mathbf{B}$, follow these steps:

1. Multiply each number in the first column of the matrix on the left ($\mathbf{A}$) by each number in the first row of the matrix on the right ($\mathbf{B}$), in order: $\mathbf{A}_{1,1} \times \mathbf{B}_{1,1}$, $\mathbf{A}_{2,1} \times \mathbf{B}_{1,2}$, ..., $\mathbf{A}_{1,n} \times \mathbf{B}_{m,1}$. Note that in order for this to work, the number of rows in matrix $\mathbf{A}$ must equal the number of columns in matrix $\mathbf{B}$.

2. Sum all of these products that you have just calculated.

3. The result is the values for the resultant matrix in location (1,1).

4. Repeat this for each row in Matrix $\mathbf{A}$ and column in Matrix $\mathbf{B}$. If the sizes of these arrays are $(m_{\mathbf{A}} \times n_{\mathbf{A}})$ and $(m_{\mathbf{B}} \times n_{\mathbf{B}})$, then your resultant array will have dimensions $(m_{\mathbf{A}} \times n_{\mathbf{B}})$.

An example of matrix multiplication follows:

$$\mathbf{AB} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix} = \tag{7}$$

$$\begin{bmatrix} a_{1,1}b_{1,1} + a_{1,2}b_{2,1} + a_{1,3}b_{3,1} & a_{1,1}b_{1,2} + a_{1,2}b_{2,2} + a_{1,3}b_{3,2} \\ a_{2,1}b_{1,1} + a_{2,2}b_{2,1} + a_{2,3}b_{3,1} & 4d2, 1b_{1,2} + a_{2,2}b_{2,2} + a_{2,3}b_{3,2} \end{bmatrix} \tag{8}$$

I usually think of matrix multiplication as "over... down". But some better memonics include "karate... chop!" (the "chop" being obviously down) and the visual way of multiplying A and B, as follows:

$$\mathbf{AB} = \tag{9}$$

$$\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} \begin{bmatrix} & \\ & \end{bmatrix} \tag{11}$$

Here, the empty matrix can be filled with the values by multiplying the rows and columns that project (on a straight line) into that cell. (Thanks to Jack for both of these.)

2.4 Determinants

@TODO:

2.5 Matrix equations

@TODO:

2.5.1 Dot products

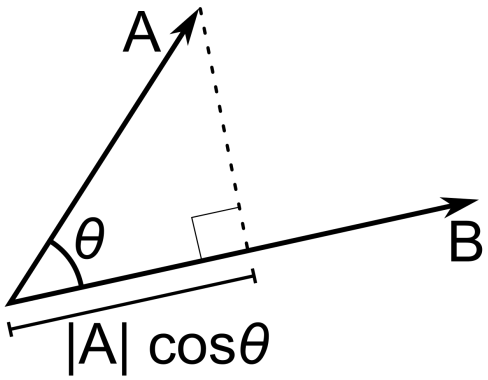

Figure 1: The dot product can be thought of as the scalar product of the magnitude of one vector, times the magnitude of another vector that has been projected onto it. https://commons.wikimedia.org/wiki/File:Dot_Product.svg.

The dot product between two arrays can be visualized as the magnitude the product of two n-dimensional vectors. It is therefore also called the "scalar product", because it gives a scalar result.

$$\vec{a} \cdot \vec{b} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \sum_{i=1}^{m(=n)} a_i b_i \tag{12}$$

If you know the angle, $\theta_{\vec{a},\vec{b}}$, between the two vectors, a dot product can also be represented as:

$$|\vec{a}| \left|\vec{b}\right| \cos\left(\theta_{\vec{a},\vec{b}}\right) \tag{13}$$

2.5.2 Cross products

A cross product, or "vector product", gives the magnitude and direction of a product of two vectors, $\vec{a}$ and $\vec{b}$, where the magnitude is the area of the parallelogram created by the two vectors and the direction is given by the right-hand rule.

The cross product can be solved in a 3-space (so for two orthogonal 2-D vectors) by solving the determinant of

The scalar value of the cross product can also be obtained as follows:

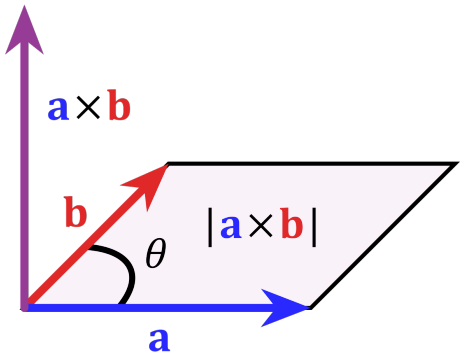

Figure 2: Graphical definition of the cross-product of two vectors. https://commons.wikimedia.org/wiki/File:Cross_product_parallelogram.svg.

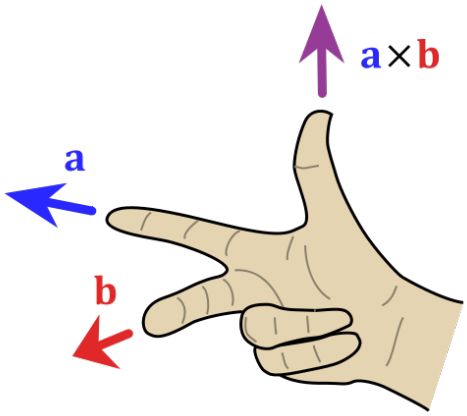

Figure 3: Direction of a cross-product of vectors $\vec{a} \times \vec{b}$ as given by the right-hand rule. https://commons.wikimedia.org/wiki/File:Right_hand_rule_cross_product.svg.

2.6 Dot and cross products

The relationship between the dot and cross products is that:

$$\|\vec{a} \times vecb\|^2 = \|\vec{a}\|^2 \left\|\vec{b}\right\|^2 - (\vec{a} \cdot \vec{b})^2 \tag{14}$$

2.6.1 Trace

The trace operator is given by the sum of the diagonal elements:

$$\mathrm{Tr}\left(\mathbf{A}\right) = \sum \mathbf{AI} \tag{15}$$

2.6.2 Einstein notation

@TODO:

2.7 Derivatives

A derivative is the change in a function $(f(x))$ with respect to the change in the independent variable (x) as the interval (Δx) approaches 0:

2.7.1 Ordinary

$$\frac{d}{dx} f\left(x\right) = \lim_{\Delta x \to 0} \frac{f\left(x + \Delta x\right) - f\left(x\right)}{\Delta x} \tag{16}$$

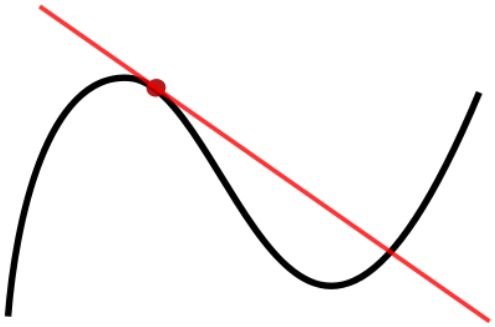

Figure 4: The graph of a function, drawn in black, and a tangent line to that function, drawn in red. The slope of the tangent line is equal to the derivative of the function at the marked point. (Text from https://en.wikipedia.org/wiki/Derivative on 2015.05.07; borrowing it because I couldn't find a better way to say it!)

@TODO: Create a better figure for derivative definition and finite difference, with shown Δx

2.7.2 Partial

@TODO:

2.8 Integrals

Antiderivatives, or integrals, provide the sum of the area under a given curve. These are very useful for solving problems involving processes that occur over space, time, or other (more mathematical, less tangible) dimensions. An example of a non-tangible dimensions would be one that relates to a degree of freedom of some equation and/or error analysis.

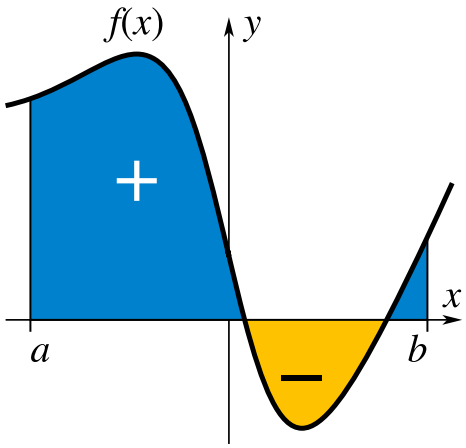

Figure 5: An integral is the sum of all area under a curve. (Contributed to Wikimedia Commons by User:KSmrq)

Indefinite integrals can be defined as:
Definite integrals can be defined as:
@TODO: Check

$$\int_a^b f(x)\,dx = F(b) - F(a) \tag{17}$$

$$\int_a^b f(x)\,dx \tag{18}$$

2.9 Taylor series

The Taylor series of any complex function $f(x)$ (i.e., $f(x) \in \mathbb{C}$) that is infinitely differentiable at a point x_0 approximates that function as a power series:

$$f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \cdots \tag{19}$$

Figure 6: This "S" in the Berlin–Wannsee station sign is written in the old style. It is used for integrals to stand for "sum", as the area under a curve can be imagined to be a sum ($\sum$) of every infinitessimally thin column of space under a curve.

Or, in the more-compact summation notation, this is:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n \tag{20}$$

where n is the number of the derivative of f.

3 Finite difference

One of the most useful functions of computers is their ability to approximate the values of derivatives at points. This allow us to directly compute approximations of the values of derivatives in differential equations and therefore solve them, typically through space and/or time.

You may have wondered why I covered derivatives and Taylor series right after one another. This is because these provide two ways to generate finite difference solutions to differential equations. The definition of a derivative provides a way to think of a derivative as the linear tangent to a curve at a given point – and hence, also, to be well-approximated by a straight line over a finite distance so long as that distance is small compared to the nonlinearity of the curve that it is approximating. This is the same as the second term in the Taylor series expansion.

The Taylor series provides a way to consider higher-order equations that could fit a curve near a certain point. A straight line often is not a poor fit, but if we want to replace an arbitrary function with an infinitely differentiable polynomial, generating a Taylor series approximation is an easy way to do so – and thus, to numerically solve differential euqations.

The result of this section will be a formalilzation of the **finite difference method**, by which a set of differential equations is estimated on a grid with constant (but

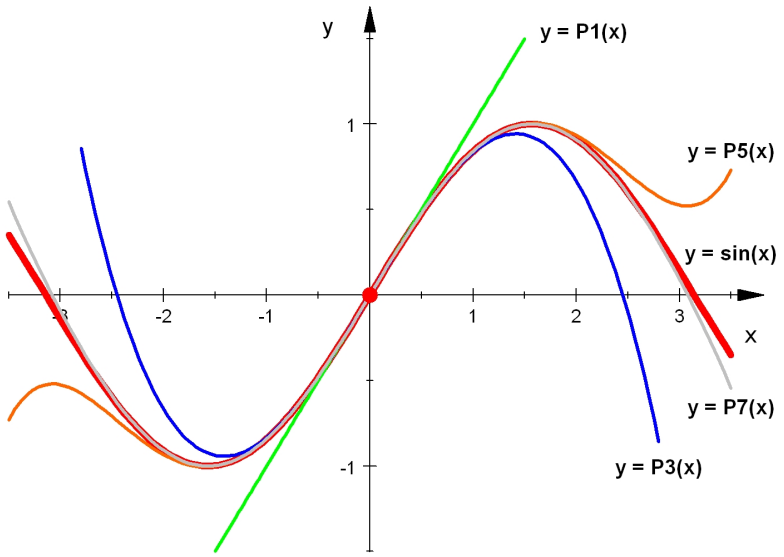

Figure 7: Increasing orders of the Taylor series expansion (and thus its derivative), denoted Pn, where $n = 1, 2, 3, \cdots$, show the increasing approximation of a sum of polynomials to the sine function.

not necessarily uniform) spacings between grid cells (hence finite difference) and then solved. By this, I mean that the distances between grid points stays constant across the varaible over which you are integrating, but may vary with respect to any other variables.

In this class, we will most likey truncate all expansions after the linear first terms: the higher-order terms are more work to include, for both you and the computer, and smaller time-steps or more clever solutiom nethods can often make up for not having them.

3.1 Where this fits in the spectrum of modeling

To provide a bit more background, finite difference is just one of several major methods for modeling with continuum equations. Other methods are finite volume (similar to finite difference except considering full cell volume, finite element (which allows a more irregular grid) and spectral (using how something varies in terms of periodic functions in order to write solutions to equations – this generally requires Fourier and/or other spectral transformations of the equations). Agent-based models that follow the actions of individual "agets" in a computational space also exist. These agents are often given simple rules and then allowed to interact with one another.

3.2 Thermal diffusion

In the example given here, we will solve the diffusion equation. First, I will present a derivation.

Heat is conducted at a rate that is proportional to the:

- **Temperature difference (directly proportional)**: hotter things more quickly make other things hotter and colder things more quickly make other things colder. The contents of a bottle placed in the freezer will become cold more quickly than will those of one placed in the refrigerator
- **Distance along which heat must be conducted (inversely proportional)**: if there is a thicker wall to a thermos, cooler, or refrigerator, it will better insulate the heat

The constant of proportionality, k, is called the "thermal conductivity" and tells one how easily heat is conducted through a medium. It is an analog to electrical conductivity.

Putting all of these together, we can write (in one dimension):

$$q_x = -k\frac{dT}{dx} \tag{21}$$

Here, q_x is the heat flux (units of Watts per meter squared [W m^{-2}]). x indicates that we are solving the equation in the x-direction; we will work with a one-dimensional solution here, though you can imagine higher-dimensional solutions to require 2-D instead of 1-D arrays of values for our numerical implementation. The negative sign indicates that heat is flowing downgradient, from hotter items to colder ones.

This expression for thermal conduction (Fourier's Law of Thermal Conduction @TODO: check) can be paired with an expression that describes how the transfer of heat from one object to another results in changes in temperature. This change in temperature produces a *feedback*, by which an object that is conducting heat to another object cools (while the other object warms. This decreases the gradient in temperature between the two objects, and therefore decreases the heat flux.

As one might expect, temperature increases with heat flux into an object and decreases with heat flux out of an object. The negative spatial derivative of heat flux provides the negative divergence (i.e. convergence) of heat at the point of interest. Assuming no phase changes (e.g., solid to liquid), this occurrs at a rate that is inversely proportional to the specific heat capacity of the material and its density.

$$\frac{\partial T}{\partial t} = -\frac{1}{c_p \rho}\frac{\partial q_x}{\partial x} \tag{22}$$

We can plug Equation 21 into Equation 22 to produce a single equation for thermal conduction

$$\frac{\partial T}{\partial t} = \frac{1}{c_p \rho}\frac{\partial}{\partial x}\left(k\frac{\partial q_x}{\partial x}\right) \tag{23}$$

If we assume that the thermal conductivity, k, is spatially uniform, then we can pull it outside the derivative. This yields the most common form of the *diffusion equation*, one of the most widely-used equations in science.

$$\frac{\partial T}{\partial t} = \frac{k}{c_p \rho} \frac{\partial^2 q_x}{\partial x^2} \tag{24}$$

Redefining the coefficients on the right-hand side of the equation as κ, which is known as "thermal diffusivity" and for many Earth materials $\approx 10^{-6}$ m^2 s^{-1} yields:

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 q_x}{\partial x^2} \tag{25}$$

In addition to heat transfer, the diffusion equation cab be used to model any process in which a rate is linearly dependent on the difference in values of some state variable. This includes chemical fluxes, temperature fluxes, groundwater flow, hillslope geomorphic change, ...

3.3 Discretization

The simplest way to "discretize" an equation, or turn it into something that can represent changes over finite (i.e., $\Delta x =$ some number > 0) distances, is to replace the $\mathrm{d}x$ or ∂x terms with a finite Δx. In doing so, we need to take into account a couple of considerations:

1. What do our grids of variables look like?
2. Do we want to generate differences forwards or backwards in time? (I am focusing on changes in time here, but you can really model a function – in this case, integrating it numerically – over any variable.)

3.3.1 Grids

For our thermal diffusion example, there are two (or three) important grids, each of which I will represent as a column vector (not to waste space, but to prepare you for matrix methods to be introduced in a later section). The first is our temperatures:

$$\boldsymbol{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{bmatrix} \tag{26}$$

And the second is position:

$$\boldsymbol{T} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \tag{27}$$

3.3.2 Differences

In order to do finite difference calculations, we must make, well, finite differences! But how should we define our differences? There are a few ways. Using the example of $T(x)$:

- Left-handed or "upwind" difference scheme:

$$1 \tag{28}$$

$$\frac{\Delta T}{\Delta t} = \kappa \frac{\partial^2 q_x}{\partial x^2} \tag{29}$$

Forward difference and "implicit" on both structured and unstructured meshes

Stencils

Linearizing equations (is this the right word? turning them into a set of linear equations)

Differential equations and linear algebra review

Include examples