Homework #3: Hill-slope evolution and plotting **Due: 5:00 PM 09/16/16**

Please read the following questions carefully and make sure to answer the problems completely. In your MATLAB script(s), please include the problem numbers with your answers. Then use the *Publish* function in MATLAB to publish your script to a *pdf* document. For more on the *Publish* functionality within MATLAB see `http://www.mathworks.com/help/matlab/matlab_prog/publishing-matlab-code.html`. Upload your *pdf* file to Blackboard under Assignment #3. Your filename should be *GEOS397_HW3_Lastname.pdf*. Hint: You can achieve this automatically by calling your MATLAB script *GEOS397_HW3_Lastname.m*.

Background

Modern geomorphological research involves basic knowledge of 1) the landforms, processes and data that can be observed in the field, 2) the physics and mathematics of those processes, and 3) computer based modeling. These three components combined allow us to gain new insights into landform evolution processes, understand past changes, and make predictions about future landforms. In this homework you will learn some of the basics of the MATLAB computing software, the programming environment, how to build and run simple models, and to plot your results.

In this homework you will model the evolution of a hill-slope through time. All programming will take place in the MATLAB editor; you only need to make one script. I suggest you build *sections* of the script and make sure each section runs as expected before you proceed to the next section.

The Hill-slope model

After reading the papers by Hallet & Putkonen (1994) and Putkonen *et al.* (2008) and discussing them in class with the guest lecturer Mike Poulos, you should have a feel for how hill-slopes evolve through time. Let's review the equations anyway and work through the important parts we need to consider when going from equations on paper to code in a computer.

The conservation equation

The conservation equation is

$$\frac{-dz}{dt} = \frac{dq}{dx}, \tag{1}$$

where dt is the time increment in years and dx is the step size (i.e. segment length) in the horizontal direction in meters. In words, Equation 1 states that if more soil enters an area than leaves the area, it must pile up, causing the ground surface to rise. It is called a "conservation" equation because it states that the volume of material (soil, in this case) is conserved, and there are no *sources* or *sinks* of material. More specifically, it states that the rate at which the elevation of a given location decreases (dz/dt) is equal to the difference in the rates at which sediment enters (from upslope) and leaves that location (dq/dx). Formally, we can say that the rate of landscape lowering equals the divergence of the soil flux. Importantly, this is a universal idea, and the expression is valid for any process.

The transport equation

The transport equation is

$$q = -\kappa \frac{dz}{dx}, \tag{2}$$

where κ is the topographic diffusivity (or diffusion coefficient) in m^2/yr. Equation 2 states that the soil flux (q) is proportional to the hill-slope gradient (dz/dx). We use *proportional* here instead of equal because the right-hand-side is multiplied by κ. Intuitively this should make sense; all else being equal, soil should move (think *soil flux*) faster on a steep hill-slope than on a gentle one. Finally, topographic diffusivity is a function of many factors: climate, substrate, vegetation, animal activity, etc.

If we take the derivative of equation 2 with respect to x we find that

$$\frac{dq}{dx} = -\kappa \frac{d^2 z}{dx^2}, \tag{3}$$

and we can then insert this result into equation 1 to get

$$\frac{dz}{dt} = \kappa \frac{d^2 z}{dx^2}. \tag{4}$$

This equation is known as the *diffusion* equation. The left-hand-side is the rate of change in surface elevation with time. Because the right-hand-side is a second derivative of space (e.g. dx^2 in the denominator), we know that the *rate of change* of surface elevation is proportional to hill-slope curvature. We are going to use equation 4 to model how the height (z) of a hill-slope evolves (i.e. changes) with time. That means we are going to solve equation 4 numerically for the varlue of z; let's do it.

Part 1: The finite difference approximation *(30 pts.)*

The first step in solving equation 4 numerically is to write each derivative in terms of a finite difference approximation. We will do this together.

Step 1: Taylor Series expansion

First we need the Taylor Series expansion of the function $f(x + h, t)$.

$$f(x + h, t) = f(x, t) + h\frac{\partial f(x, t)}{\partial x} + \frac{h^2}{2}\frac{\partial^2 f(x, t)}{\partial x^2} + \frac{h^3}{6}\frac{\partial^3 f(x, t)}{\partial x^3} + \dots .$$

Notice that this is a function of two variables $f(x, t)$, but that we are only computing the derivatives with respect to x. We can write a similar Taylor Series expansion with respect to time t using the variable k. Write the expansion below *(10 pts.)*.

Step 2: The *forward* difference operator for first derivatives

Recall from Lecture 05 that we can rearrange the terms in these expansions so that the desired derivative is by itself on the left-hand-side (LHS) of the equation.

$$\frac{\partial f(x,t)}{\partial x} = \frac{f(x+h,t)-f(x,t)}{h} + \frac{h}{2}\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{h^2}{6}\frac{\partial^3 f(x,t)}{\partial x^3} + \ldots = \frac{f(x+h,t)-f(x,t)}{h} + O(h),$$

What does the term $O(h)$ represent in the equation above? *(4 pts.)*

Write the first order *time* derivative using the forward difference operator. *(6 pts.)*

Step 3: The *centered* difference operator for second derivatives

Go back to the Lecture 05 notes (README.html) and look at the derivation of the second derivative with respect to x. Write the centered difference approximation. *(5 pts.)*

Step 4: Approximate the derivatives

The next step is to approximate the first order *time* and second order *space* derivatives. The approximation for the second order space derivative is

$$\frac{\partial^2 f(x,t)}{\partial x^2} \approx \frac{f(x+h,t)-2f(x,t)+f(x-h,t)}{h^2}.$$

Note the approximation symbol instead of the equal sign. This is because we have truncated the $O(h^2)$ terms. Write the approximation for the first order time derivative. *(5 pts.)*

Part 2: The finite difference solution to the diffusion equation *(20 pts.)*

Step 1: Approximate the partial differential equation

Now that we have an approximation for the first order time derivative $\frac{\partial f(x,t)}{\partial t}$ and the second order space derivative $\frac{\partial^2 f(x,t)}{\partial x^2}$, we can insert these into the diffusion equation. Rewrite equation 4 in terms of the finite difference approximations. (*Hint: Notice that we just did the derivation in the previous section using a generic function f. When writing your answers in this Part, make sure to replace f with z because in our diffusion equation we are computing the elevation $z(x,t)$.*) *(10 pts.)*

Step 2: Solve for the value of the function at time $t+k$

Now solve this equation for $z(x, t+k)$. (*Hint: Remember that k and κ are different!*) *(10 pts.)*

Reflection:

It is important to stop here and reflect on what we have just done. Ask yourself, what does the equation you just derived actually represent? That is a good question. Basically you have now derived the finite difference solution to the diffusion equation. This analysis is **very** general and can be applied to any diffusion equation (e.g. temperature, fluid flow in porous media, etc.). You just need the (partial) differential equation; then you approximate derivatives and insert into the (partial) differential equation.

In particular though, you have just solved for the value of the function at the next time step $t+k$. Let's look at what we need in order to solve the function at the next time step. We need $z(x,t)$, which is the value of the topography at the current time step t and at the observation location x. That makes sense right? You can compute the elevation in the future if you do not know the elevation now. What else? Well, you need to know the value of the topography to the left ($z(x-h,t)$) and to the right ($z(x+h,t)$) of you at time t. This also makes sense. We are computing derivatives so we should expect that we need to know the local slope of the topography around observation point x. Finally, we need to know κ, h and k. We haven't discussed h and k yet, but we will in the next section.

Part 3: Implementing the numerical solution *(35 pts.)*

In this last part, we will implement the finite difference solution to the diffusion equation that just derived. This will allow us to model the evolution of topography given two things:

1. the initial topography profile,
2. the value of topographic diffusivity (κ).

Start a new MATLAB script and begin to code based on the following steps. Make sure that you name variables following the MATLAB style guide document *MatlabStyle1p5.pdf*.

Preliminaries

A typical computer program usually follows the logical sequence or steps below:

1. Define parameters and constants,
2. Create the model space (includes time if necessary),
3. Compute calculations (using loops if necessary),
4. Plot results.

Let's do this now.

Step 1: Define parameters and constants

In the *first section* of your MATLAB script define κ, dt and dx. (dt is actually k and dx is h from the previous section. It helps to think in terms of dt and dx though as these are the intervals between points in time and space. For example, when we look at the points to the left and right of us, we will look at $x - dx$ and $x + dx$, respectively.) These are the important parameters that do not change during the numerical modeling of the hill-slope evolution; set $dt = 1$ [year], $dx = 1$ [m] and $\kappa = 2e^{-3}$ [m^2/year]. *(5 pts.)*

Step 2: Make the initial model

Before we can do anything on a computer involving computations, we need to decide on a model *domain*. This is the time and/or space where this equation will be solved or studied. This means we have to 1) choose a physical space and discretize that space, 2) choose a time frame and discretize that, and 3) assign some physical properties to this space. For now, we will assume the physical properties are constant everywhere (i.e. κ is a constant.) Therefore, we need to make our initial model $z(x, t = 0)$. This means we need the initial topography model $z(x)$ at time $t = 0$.

For the initial model, let's use the model given in the (Hallet & Putkonen, 1994) paper, which is a triangular moraine. We can build a triangle function $z(x)$ with the following piece of MATLAB code.

```
z = [0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1 0 0 0 0 0]; % [m]
nNode = numel( z ); % [No] number of elements in the x-direction
xArray = ( 0 : nNode - 1 ) .* dx; % [m] make the x-position vector
```

The variable z is the elevation. The variable nNode is number of element in the z-array, meaning we have sampled the topography evenly nNode times. The variable xArray gives the relative x location of each element in z.

Based on $dx = 1$, at which x location does the maximum elevation occur in our triangular topography model? *(4 pts.)*

Plot the initial topography at time zero. Make sure to include a figure title, axes labels and a legend. Save this a .png file that has dimensions 600x600 pixels. *(6 pts.)*

Step 3: Loop through time to compute the topography at $t + dt$

Now that we have an initial model and the physical properties of this initial model, we can compute the topography at the next time step $t + dt$. This is the heart of the finite difference solution. Let's think about how we might do this. First we need to choose an end time. Make another parameter called *tMax* and set it to 100 [years]. We want to loop over time from time zero up to time *tMax*. For generality, we should also make a generic start time as well. Make a variable called *t0* and set it equal to *dt*; this means that the first observation year is 1, or that our initial model occurs at year 1.

Make a *for* loop that runs from *t0+dt:dt:tMax*. This loop starts at *t0+dt* and stops at *tMax*, incrementing by whatever we assign to *dt*. This should remind you of $z(x, t + dt)$, where we are computing the elevation at every *dt* step. *(5 pts.)*

Now what goes inside of this *for* loop? Well, it should be the right-hand-side of your equation for $z(x, t+dt)$. The only thing left to consider is the $(x-dx)$ and $(x+dx)$ terms required to compute $z(x, t+dt)$. It turns out that we need another *for* loop inside our time loop. The limits of the second *for* loop should be *2:nNode-1*. *Hint:* we are not dealing with the edges of the model (i.e. xArray(1) or xArray(nNode). We will not discuss boundary conditions here; instead should set the edges of your model to zero after each *for* loop over x.) Inside this second *for* loop, insert your finite difference equation *(10 pts.)*

Step 4: Plot your results

Using the MALTAB **hold** command, plot your final model at 100 years on top of your initial model. Make sure everything in your plot is labeled *(5 pts.)*

Part 4: Discussion *(5 pts.)*

Vary the *tMax* variable and recompute your model. Describe the differences you see when you set *tMax* equal to 1e2, 1e3, 1e4, 1e5, 1e6. Does this make sense with what you know about erosion and sediment movement? Explain. *(3 pts.)*

Vary the κ value and describe how this parameter changes the model output. *(2 pts.)*

References

Hallet, Bernard, & Putkonen, Jaakko. 1994. Surface dating of dynamic landforms: Young boulders on aging moraines. *Science (New York, N.Y.)*, **265**(5174), 937–940.

Putkonen, Jaakko, Connolly, Jon, & Orloff, Travis. 2008. Landscape evolution degrades the geologic signature of past glaciations. *Geomorphology*, **97**(1-2), 208–217.