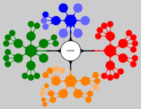
Geocomputation 1

Algorithms

Understanding algorithms in geology

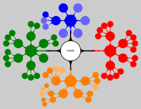
version: 30 November 2010

Chuck Connor and Laura Connor
Department of Geology
University of South Florida



What are the goals of this module?

1. Learn the definition of algorithm
2. Learn the characteristics of algorithms
3. Develop a sense of how to use algorithms in the geosciences
4. Learn to express algorithms using flowcharts



Outline

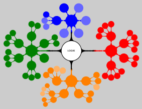
Computer programs are all about solving problems

This course is all about learning to write good code to help solve geoscience problems. There are lots of reasons to solve problems with computer codes. Here are a few, you can probably think of many more!

Some reason's geoscientists write code:

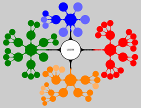
- Calculate: writing computer programs facilities all kinds of calculations
- Explore data: Exploration of geological or geophysical data usually requires writing or using a computer code. Writing your own code is worth it when it allows you to explore data in new ways
- Create models: modern geoscience often involves simulating earth processes; computer programs are one way – sometimes the only way – to do it.
- Explore models: computer codes facilitate comparison of observations and model results. This creates a powerful way to validate models (decide if they work to understand earth processes).

One, perhaps unexpected outcome, is that writing computer programs helps us think about earth processes in a quanitative way. Experience in writng computer programs is closely linked to experience in solving problems. Some examples follow!

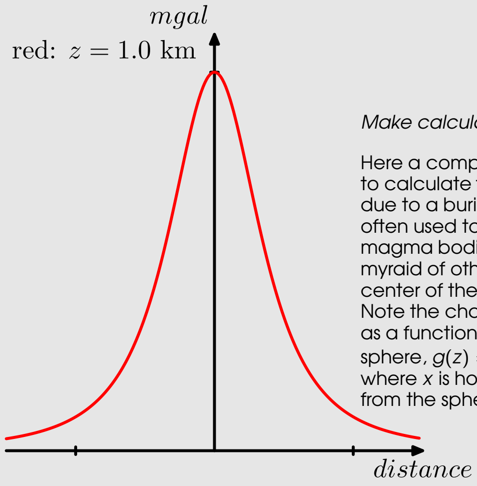


Why should geoscientists write computer programs?

Calculate: Computer programs can be used as elaborate calculators. Many tools are available to make calculations on a computer (e.g., a spreadsheet) but if the caluations are complex or it is necessary to repeat them many times, it is useful to write a program. In some cases, where quality assurance is important, a computer program is a very effective way to document the method(s) used to perform calculations. In other words, a computer program can provide a record of how problems are solved.



Why should geoscientists write computer programs?



Make calculations:

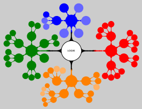
Here a computer program is used to calculate the gravity anomaly due to a buried sphere, a shape often used to approximate magma bodies, sinkholes, and a myraid of other features. The center of the sphere is at depth, z. Note the change in the anomaly as a function of depth of the sphere, $g(z) = f[z/(z^2 + x^2)^{3/2}]$, where x is horizontal distance from the sphere center.

Click on the icons below the graph to change the depth, z, of the sphere center.

Computer programs allow you to consider a range of parameters in calculations.

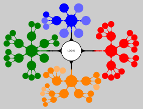
Why should geoscientists write computer programs?

Explore data: The geosciences are increasingly characterized by LOTS of data. A topographic surface may be represented by a high resolution digital elevation model (DEM); thousands of trace element analyses may be used to characterize the geochemistry of an aquifer. Computer programs are essential for exploring specific data sets in sufficient detail. The larger the data set, the more likely it needs to be visualized and explored using a computer program tailored to the task.



Why should geoscientists write computer programs?

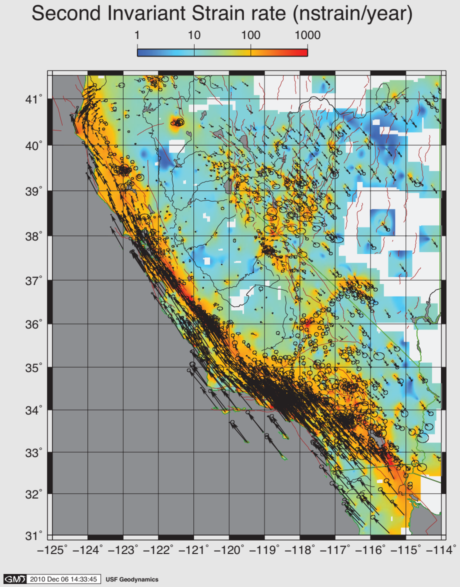
Create models: Most geological processes are not directly observable. We cannot directly observe the path water takes through an aquifer or the subduction of ocean lithosphere into the mantle. One way to learn about these processes, that happen on very long time scales, very rarely, or out of view, is to create simulations. This type of numerical modeling is a very effective method of testing ideas about how the earth functions. As in other fields, numerical simulation of dynamic phenomena has become an indepensible tool in the geosciences. Numerical simulations are applied to phenomena ranging from heat transfer, to geophysical flows (ice, magma), to seismic wave propagation. A typical path is to develop increasingly complex numerical simulations that incorporate "physical realism" or at least versions of reality.



Why should geoscientists write computer programs?

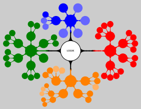
Creating models:

Geodesists use a tremendous aomount of GPS data to determine plate velocities, here for the western margin of North America. The black arrows show relative plate velocity, with respect to a fixed Sierra Nevada block (the large low velocity region in the middle of the figure), ellipses show the estimated error in these velocities, assuming rigid plate motion. The map is then colored using a model of strain rate derived from these data So, this figure illustrates two different models: a model of rigid plate motion (arrows) and a model of the strain rate (colors). Both models indicate that the rigid plate model works well, even for microplates, and deformation is mainly accomodated at the plate boundary, revealed by the huge change in relative plate velocity across the San Andreas fault system west of the Sierra Nevada block.



Why should geoscientists write computer programs?

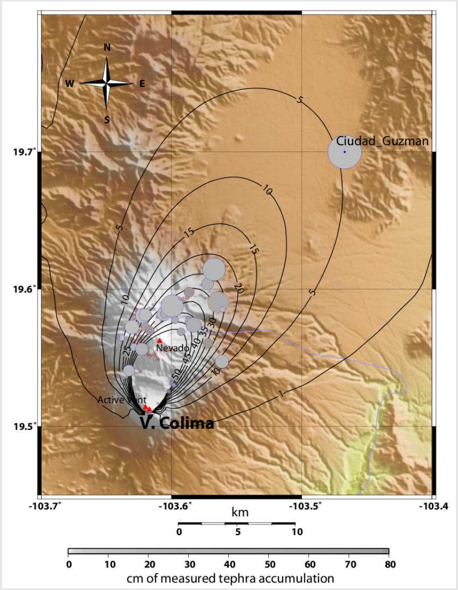
Explore models: Increasingly, geoscience models are assessed using the goodness-of-fit between observations and realizations of our mathematical approximations of reality. How well can we infer geologically important parameters from our observations? Computers provide a powerful platform for optimization and/or inversion of data.



Why should geoscientists write computer programs?

Explore models:

Volcanologists model the dispersion of tephra (volcanic ash) from explosive volcanic eruptions using fluiddynamic models. This figure shows the results of modeling the 1913 eruption of Colima volcano, Mexico, using the advection-diffusion equation. Contours show the tephra thickness (cm) based on the model and gray circles show the fit to observed tephra thickness in the stratigraphic section (bigger circles indicate poorer fit). This best-fit model suggests the mass of the 1913 eruption was $1 \times 10^{11} \pm 0.3 \times 10^{11}$ kg and the height of the volcanic eruption column was 30 km $\pm$ 5 km.



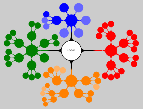
The Pólya approach to solving problems

George Pólya (1887—1985) was a Hungarian mathematician and a heroic problem solver. In the 1940's, Pólya developed a systematic approach to problem solving that still resonates with anyone interested in scientific programming. His approach appears in every worthy text on "how to write computer programs", but is generally useful, even if no computer is needed to solve the problem!

Overview of Pólya's approach

- First, *understand the problem*. What is the unknown? What are the data? What conditions link the data to the unknown?
- Second, *devise a plan for solving the problem*. Find the connection between the data and the unknown!
- Third, *carry out the plan*. Often, at this step an actual computer program is written to solve the problem.
- Fourth, *examine the solution*. Can you check the results? Can you verify that the mathematics in the solution are done correctly? Can you determine that the original problem is solved?

Note that "writing the computer program" only appears in the third step, where a plan is carried out to solve the problem. This step might be the easiest, especially if sufficient time is spent considering the other steps in Pólya's problem solving method! In the rest of this module we will concentrate on the first two steps.



Step 1: Understand the problem

Geoscience is often thought of as a discovery-driven discipline. That is, if we go out and map a quadrangle or analyze some rocks, interesting results will crop up that require explanation. This can be a fine and exciting approach to learning about the Earth, but it can be inefficient, or worse, a waste of time. At its best, geoscience is problem-driven. What is the liquidus of basalt? What is the volume of a metaliferous deposit? What is the expected run-up distance of a beach wave? Investigations can be much more productive (and computer programs much more useful) when specific questions are addressed. So the first step is to understand the problem.

Questions to ask to help understand the problem

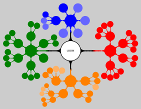
- What is the unknown?
- What are the data?
- What are the links between the data and the unknown? Are these links sufficient? Or redundant?
- What assumptions are required to solve the problem?

Step 1: Understand the problem

It is essential to get acquainted with the problem.

Get acquainted with the problem

- *Where should I start?* Write down a clear statement of the problem. Do not be fancy! This statement should be so clear that nearly anyone can read it and understand the problem to be solved.
- *Visualize the problem.* Draw a figure to illustrate the problem. Try to make the figure to scale. What parts of the problem are easiest to visualize? Which are the most difficult to visualize and why?
- *Develop notation for the problem.* Can you express the unknown as a variable? How about the data? Can you describe the functional relationship between variables? If not, can you isolate parts of the problem that can be so described?

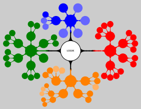


An example of “understanding the problem"

Geoscienctists are often concerned with the volume of features such as gravel deposits, lava flows, formations and basins. Consider Crater Lake (Oregon), formed in the caldera of Mt. Mazama after a 7700 BP eruption. What is the volume of the caldera? This is an interesting question if we want to compare, say, the volume of erupted products to the volume of the caldera resulting from the explosive eruption.

The unknown is the volume of the caldera (m^3).

The answer must be estimated somehow using the diameter (m) or area (m^2) and depth (m) of the caldera. The photo indicates there is a problem. Part of the caldera is above water, and part of the caldera is below water. What is the shape of the caldera below the water line? Also, the part of the caldera below water line is likely partially filled with sediment or other post-caldera eruption rocks. How can the volume of these sediments and rocks be accounted for? Getting acquainted with the problem reveals some complexities that might otherwise not be recognized!

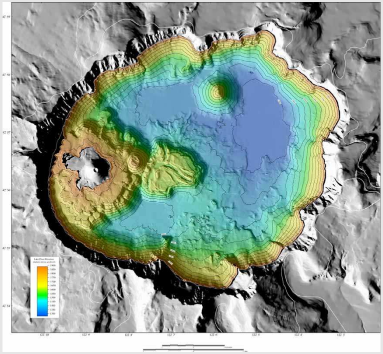


An example of “understanding the problem"

The problem is best broken into parts. What is the volume of water in Crater Lake? What is the volume of post-caldera sediment and rock within the lake? What is the volume of rock above water line removed by eruption? Here, concentrate on th volume of water in the lake, perhaps the easiest problem to solve.

What are the available data?

This figure shows a bathymetric map of Crater Lake that was created from data collected by the USGS, NPS, and University of New Hampshire. The color-shaded and countoured data were collected at a spatial resolution of 2 meters using sonar. The map is superimposed on a lower resolution (10 m) digital elevation model of the terrain (gray shading). How can the 2 m resolution bathymetric data be used to estimate lake water volume?



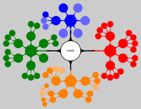
USGS, http://pubs.usgs.gov/dds/dds-72/

Step 2: Devise a plan

In devising a plan you seek to clarify the connection between the unknown and the data. There are important questions to ask to help you devise a plan.

Questions to ask when devising a plan

- Have you seen the problem before, or seen this problem is a slightly different form?
- Consider the unknown. What are the units of the unknown? Can these units help you define a functional relationship between the unknown and the data?
- Do you know of a related problem that you have solved before. Has someone else solved a similar problem? Can you use or modify their method? Look at the problem from different perspectives.
- Can part of the problem be solved? If so, could the problem be restated differently to take better advantage of the parts that can be solved?
- Does the solution use all of the available data? Would other data help?

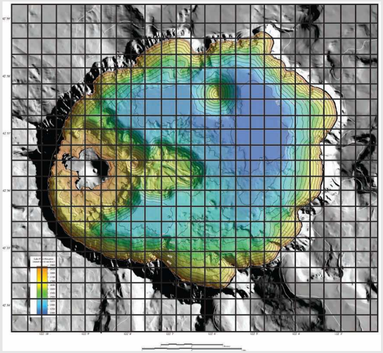


An example of “devising a plan"

For solving for the volume of water, it makes sense to consider how best to use the avialable bathymetric data.

The plan

Since the data are collected at 2 m resolution (that is, interpolated on to a 2×2 m grid, it should be possible to accurately estimate the volume of water in the lake with these data. *This problem is analogous to many integration problems.* This suggests a plan based on calculating the volume of each 2×2 m grid cell (e.g., *depth* $\times$ 2 m $\times$2 m) and summing across all cells



USGS, http://pubs.usgs.gov/dds/dds-72/

Here a coarse grid illustrates the summation of the volume of each grid cell. Carrying out the plan involves using a computer program to sum the volumes of the 2×2 m grid cells across the area of the lake.

Some definitions

For our purposes, these rough definitions work well.

Rough Definition (Algorithm)

An algorithm is a finite set of unambiguous steps that are followed in order to provide the solution to a problem, or to a specific class of problems.

An algorithmic approach is different from a heuristic approach.

Rough Definition (Heuristic)

A heuristic is a rule of thumb, strategy, or any other kind of device which drastically simplifies the search for solutions to problems.

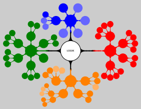
A simple example

An example may help illustrate this definition of algorithm. Here is a straightforward example:

What is the density of a rock?

- get a rock (this is the input to the algorithm)
- measure the volume of the rock
- weigh the rock
- figure the density (weight / volume)
- write down the result (this is the output of the algorithm)

Notice that the algorithm is a lot like a laboratory procedure or cooking receipe. Also note that this particular algorithm is not completely unambiguous. How might you better specify this algorithm?



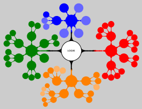
Better specification of the algorithm

Even in this simple example we should improve the algorithm so it can be implemented in an unambiguous way.

What is the density of a small rock?

- get a small rock (this is the input to the algorithm)
- measure the volume, V, of the rock by measuring the amount of water the rock displaces in a beaker
- determine the mass, M, of the rock using a balance accurate to 0.001 kg
- figure the density, ρ, where $\rho = M/V$
- write down the result using the units $\mathrm{kg\,m^{-3}}$ (this is the output of the algorithm)

The algorithm is improved in two ways. First, the word small is added because the algorithm certainly cannot be used to measure the density of really big rocks (boulders). In other words, the scope of the algorithm is better specified. Second, the individual steps are much more specific about how they are to be done. This makes implementation of the algorithm much easier and hopefully less prone to error.

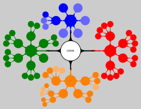


The logician's definition

Now that we have seen an algorithm, consider the definition from David Berlinaki (*The Advent of the Algorithm*, Harcourt, Inc., 2000):

An Algorithm is:

- a finite procedure,
- written in a fixed symbolic vocabulary,
- governed by precise instructions
- moving in discrete steps: 1,2,3...
- ...whose execution requires no insight, cleverness, intuition, intelligence, or perspicuity,
- and that sooner or later comes to an end.

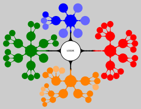


The nature of algorithms

All algorithms have these characteristics:

- each step of an algorithm is precisely defined and the actions to be carried out are rigorously specified for each case
- an algorithm achieves a problem solution after a finite number of steps – although some algrothims may have a variable number of steps, a problem solution is eventually reached
- an algorithm provides one or more outputs
- it is preferable that the algorithm be applicable to any member of a class of problems rather than only to a single problem.

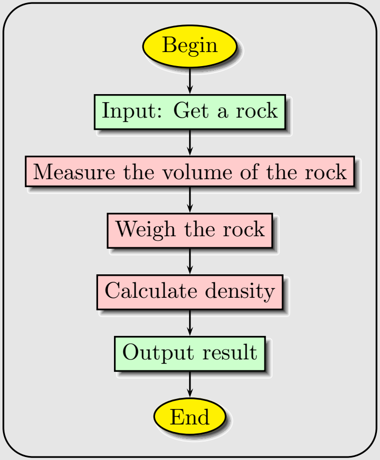
Take a moment and evaluate the “density of a rock sample" algorithm discussed on the previous slide in terms of these characteristics.



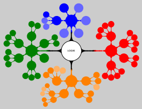
Making a flowchart of an algorithm

It is usually useful to visualize your algorithm using a flowchart. The flowchart at right illustrates the simple algorithm described on previous slides.

- the yellow ovals indicate the beginning and the end of the algorithm
- the light green boxes indicate inputs or outputs
- the light red boxes indicate a process or instruction



We will continue to use this format for illustrating algorithms with flowcharts, and add some additional complexity as we need it.

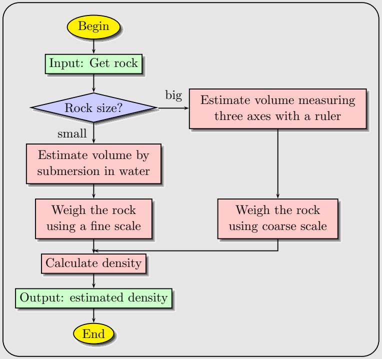


Making a flowchart of an algorithm (2)

Here is a slightly more elaborate version of the previous algorithm

The decision structure

The flowchart now handles a decision structure – a different procedure is followed depending on whether the rock is small enough to fit in a graduated flask to determine the amount of water displaced, or is so big that its dimensions should be measured and volume approximated by some heuristic. Different scales might also be used to weigh a rock if it is big or small.



Begin
Input: Get rock
Rock size?
big
Estimate volume measuring three axes with a ruler
small
Estimate volume by submersion in water
Weigh the rock using a fine scale
Weigh the rock using coarse scale
Calculate density
Output: estimated density
End

Decision structures are a key component of many algorithms