

Lesson 1: Fundamentals of proximity analysis

Topic: [Human activity, cost, and distance](#)

Concepts

[Euclidean distance and the Spatial Analyst](#)
[Euclidian distances on the grid](#)

Example

[Proving Pythagoras right](#)

Exercise

[Make a grid of straight-line distances to a single place](#)

Topic: [Multiple source cells and limited distance](#)

Concepts

[Finding the closest of many source cells](#)
[Excluding distances beyond a certain threshold](#)

Exercise

[Many cells with limited distance](#)

Topic: [Other factors that influence cost](#)

Concepts

[The CostDistance mechanism](#)
[The CostDistance request](#)
[Path calculation in EucDistance and CostDistance](#)

Example

[Understanding how total costs are calculated](#)

Exercise

[Develop a grid with CostDistance](#)

Topic: [Getting more information: Paths and allocations](#)

Concepts

[Identifying the closest source cell and how to get there](#)
[Direction and allocation grids for CostDistance](#)
[Another grid for showing least-cost paths](#)

Exercises

[Use EucDistance to create direction and allocation grids](#)
[Use CostDistance to make direction and allocation grids](#)

Calculate a least-cost path
Put the tools together: Site a regional park

Lesson summary

Lesson self test

Goals

In this lesson, you will learn:

- the importance of distance in geographic analysis and synthesis
- how to use ArcView Spatial Analyst to find distance from anywhere to places of interest
- about the relationship between distance and cost
- how to calculate the cost effects of other factors besides distance
- how to find the shortest or least expensive path from anywhere to places of interest
- how to determine which is the closest place of interest and the least-cost place of interest

Human activity, cost, and distance

Life forms on our planet may be divided into plants and animals. A major difference between them (although there are exceptions) is that plants do not move (much) and animals do. In fact, for humans in particular, moving themselves and their artifacts, such as clothing, weapons, and stereo equipment, is a major activity.

Frequently, a major concern about a given human activity is its cost--cost in terms of energy, time, money, fuel, suffering, or other parameters. Moving from one location to another is a frequent human activity and one whose cost we often want to minimize. We usually move because we want to be in a new location or leave an old one for some reason. (There are exceptions to this: a walk in the woods for soul healing; a jog for exercise.) The subject of this module is using GIS to:

- reduce the cost of going from one place to another
- lower the cost of transporting artifacts
- minimize the cost of building linear structures, such as roads, bridges, and tunnels

The costs of moving from one place to another can be complex to calculate. In cases where the distance is far, the trips are repeated frequently, or the cost critical (as in the cost in time of getting an ambulance to the scene of an accident), it may well be worth doing the computations. In this lesson, you will learn how to deal quickly and simply with the complexity of performing the "cost of moving" calculation.

What are the factors that determine the cost of moving things, including ourselves? The one that perhaps comes to mind most quickly is "distance." It costs more to go from Punxsutawney, Pennsylvania to Redlands, California than to go from your living room to your bedroom. Or from

your home to your office. So we begin with consideration of distance, distance of the simplest sort: straight-line distance (on a Cartesian plane) from any point to a given point

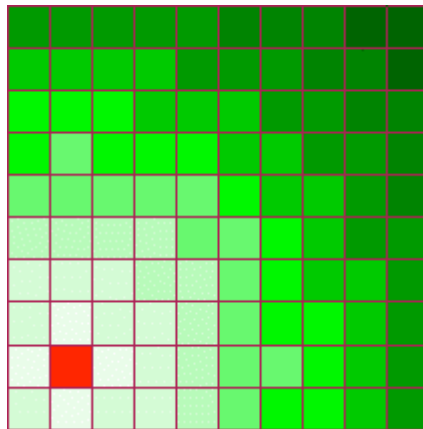
Concept

Euclidean distance and the Spatial Analyst

Proximity, the closeness of one point in space to another, is most easily expressed by the Euclidean straight-line distance between the two points. The term "Euclidean" comes from the geometry first formally developed by Euclid around 300 B.C.

ArcView Spatial Analyst allows you to automatically make a grid in which each cell in the grid contains the straight-line distance from itself to the closest cell of a set of source cells in another grid that represents the same geographic space. (Actually, because this distance is represented on the two-dimensional grid surface--differences in elevation being excluded--we could refer to the values in the cells as "Cartesian distance" after the mathematician Rene Descarte, who invented the x,y coordinate system on the 2D plane in the 1600s.)

The figure below shows a grid whose cells are composed of Euclidean distances from each cell to the red source cell.

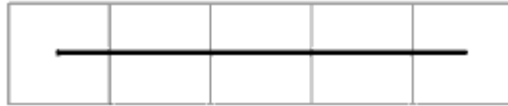


A grid of straight-line distances from each cell to a source cell, shown in red. The lighter the color, the smaller the distance.

Concept

Euclidian distances on the grid

Distance is measured between two points. The two points used when measuring distance on a grid are the centers of two specific cells. That is, in the case of the grid, the measurements are made from cell center to cell center.

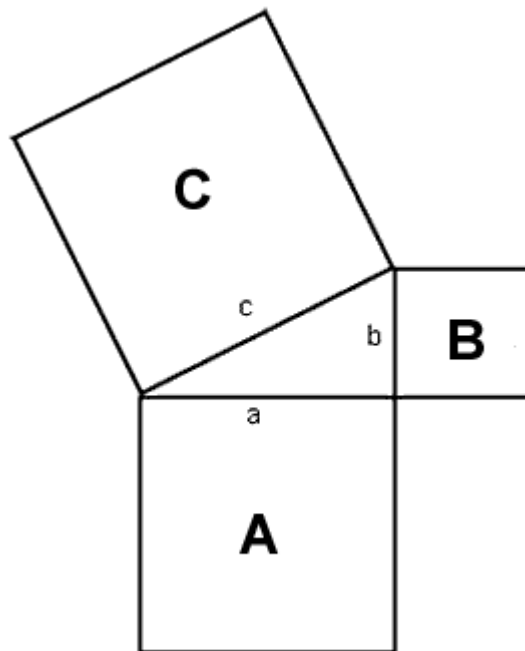


If the cell size was 50, the distance shown would be 200, spanning all or parts of five cells.

For example, if you had a grid that had cells of size 10 and 1,000 columns, the longest east-west distance measurement would be 9,990.

Measurements are made from cell center to cell center, regardless of intervening cells or the angle of the line connecting the two cells of interest. The distance is calculated by the Law of Pythagoras, who, despite having lived two hundred years before Euclid, determined that the hypotenuse of a right triangle is the square root of the sums of the squares of the other two sides.

As illustrated below, the area of the square on the hypotenuse is the sum of the areas of the squares on the other two sides.



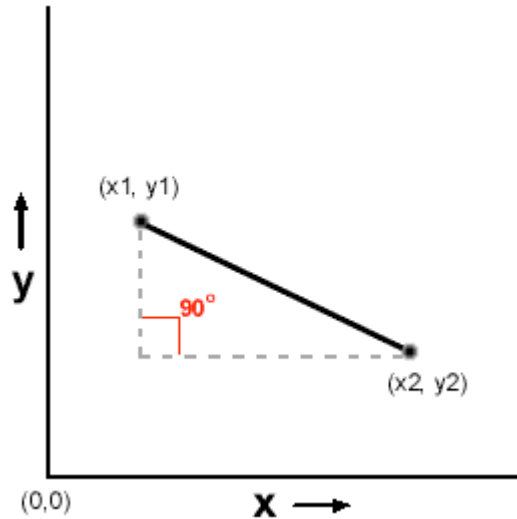
Pythagorean Theorem:
 $a^2 + b^2 = c^2$

In terms of areas, area of A plus area of B equals area of C. In terms of distances, $a^2 + b^2 = c^2$.

When you put the Law of Pythagoras (also known as the Pythagorean Theorem) together with Descartes' great invention, you can calculate the distance between two points at coordinates (x_1, y_1) and (x_2, y_2) as:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = c^2$$

where "c" is the straight-line distance between the points.



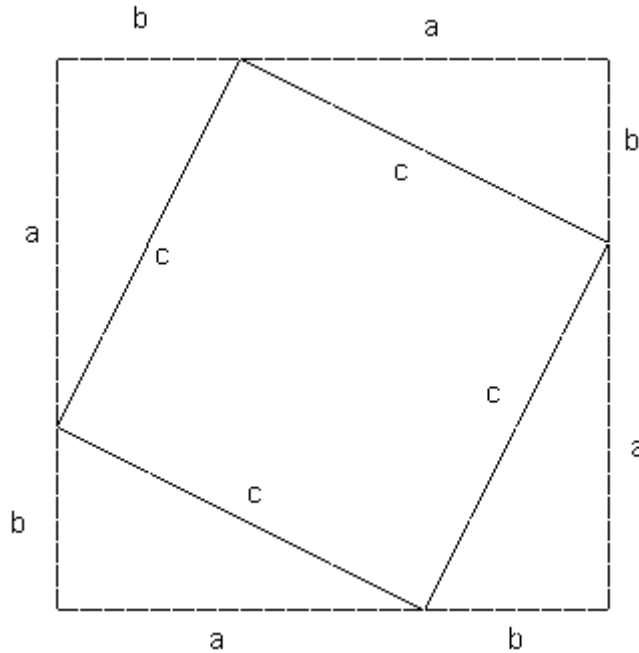
Using the coordinates of the points, the Pythagorean Theorem can be applied to calculate the distances between them.

Example

Proving Pythagoras right

If you're like most people, particularly those who are spatially-minded, you (vaguely) know the Law of Pythagoras, but probably not why it is true. So, for those who are interested, here is a proof--one that will appeal to those interested in graphical, rather than strictly algebraic, matters. (If you just want to take our word for the validity of the Law, you may skip the remainder of this concept.)

Draw a square on a piece of paper. Draw a smaller square within the first, rotated so that its corners touch the sides of the first. Call the line segment from a corner of the big square to a point where the smaller square touches it "a." Call the segment from the corner of the big square to the other touching point "b." Label all of the line segments around the big square with "a" or "b," as appropriate. Label each of the sides of the smaller square "c."



Placing one square inside another square helps illustrate the Pythagorean Theorem.

Now compute some areas. The area of each of the four triangles is $\frac{1}{2}ab$. So the total area of all the triangles is $2ab$. The area of the smaller square is, of course, c^2 . What is the area of the larger square? The length of each side is $a+b$, so the area of the square is $(a+b)$ times $(a+b)$. When you multiply it out, that's:

$$a^2 + 2ab + b^2.$$

Note now, by looking at the figure, that the area of the larger square is the same as the area of the smaller square plus the areas of the triangles, so we have:

$$a^2 + 2ab + b^2 = c^2 + 2ab.$$

Subtracting $2ab$ from each side of the equation gives the Law of Pythagoras: $a^2 + b^2 = c^2$.

Exercise

Make a grid of straight-line distances to a single place

In this exercise, you will see how ArcView Spatial Analyst makes a grid of Euclidean (straight-line) distance from each cell to a single source cell. This is accomplished with an Avenue request, `EucDistance`, which you will enter into the Map Calculator.

A simplified syntax (which shows you the components of the request for this exercise) of `EucDistance` is:

```
aSourceGrid.EucDistance( Nil, Nil, Nil)
```

where `aSourceGrid` is the grid containing the cell to which distance

is to be calculated. In a source grid the source cell has a numeric value; all other cells contain NoData. The three arguments (shown here as Nil) are explained in a later discussion.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Start ArcView

Start ArcView, choose Extensions from the File menu, and load the Spatial Analyst extension. Also load the CellTool sample extension for later use.

Note: If you are running ArcView GIS 3.1, you see a Welcome to ArcView GIS dialog. Click Cancel to close this dialog.

If ArcView is already running, close any open projects.


Step 2 Open the project

Navigate to the **proxalslesson01** directory and open **I1_ex01.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I1_ex01.apr message box. Click No to dismiss this box.

A view with a line theme called Fishnet opens. Note that the line theme appears as a 10 x 10 matrix of transparent squares. Each square is 10 units (say the units are kilometers) on a side, so the square represents a space 100 kilometers on a side. Fishnet will serve as a backdrop for the next theme.

Step 3 Add a theme to the view

Click the Add Theme button . In the Add Theme dialog, navigate to the **proxalslesson01** directory, set Data Source Type to Grid Data Source, and click on the grid named **Onecell**. Click OK.

Display Onecell by clicking its check box in the view's Table of Contents (TOC).

[VIEW RESULT](#)

Onecell is a grid consisting entirely of No Data values, except for a single cell with a value of 999.

With Onecell active, choose Properties from the Theme menu. Notice that the cell size is 10 and there are 10 rows and 10 columns. Cancel the Theme Properties dialog.

By using EucDistance, you will be able to determine the distance from the center of each cell in the grid to the center of the source cell, represented by Onecell.

Step 4 Prepare to calculate the Euclidean distance

From the Analysis menu, choose Map Calculator. In the Map Calculator, within the parentheses, enter the expression below:

```
[Onecell] . Euclidean (Nil, Nil, Nil)
```

[VIEW RESULT](#)

A tip on the Map Calculator:

- When typing an expression into the Map Calculator, a frequent problem is unbalanced parentheses. A habit that can prevent this difficulty is to always type parentheses in pairs, (). Then move the cursor back one character (using the left arrow key) so that whatever is typed, in insert mode, occurs within balanced parentheses. This way, you can avoid syntax errors.

[REVIEW CONCEPT](#)

Step 5 Generate and display the new grid

Click Evaluate and close the Map Calculator.

Spatial Analyst generates a grid theme named Map Calculation 1 and adds that theme to the view at the top of the TOC.


Turn the theme on, make it active, and change its name to **MapDist1**.

Change the order of themes, so that from top to bottom they are ordered: Fishnet, Onecell, MapDist1.

[VIEW RESULT](#)

[REVIEW CONCEPT](#)

Step 6 Explore the grid with the Identify tool

Make MapDist1 the active theme. Click the Identify tool . Run the mouse over the grid and note the cursor position given by the numbers in the upper right of the ArcView window. Now click on one of the cells.

[VIEW RESULT](#)

Notice that the cell center's coordinates are displayed in the Identify Results window. The cell's value, which is the distance from it to the source cell, is also displayed.

Step 7 Understand the grid values by checking some other cells



Using the Identify tool , query several more cells.

Each cell represents the straight-line distance from the center of that grid cell to the center of the source cell. Verify that the cell on the new grid that corresponds

geographically to the source cell on the source grid has a value of zero.

Note that if you go three cells to the right of the cell with a value of zero, the distance is 30. If you go from that cell up four cells (that is, 40 units) the (diagonal) distance to the source is 50. That is reassuring, because Pythagoras tells us that the sum of the squares of the two sides of a right triangle equals the square of the hypotenuse (900 plus 1600 equals 2500).

Step 8 Verify distances using the Measure tool

First, click on a cell and use the Identify tool  to obtain its value. Now click the Measure tool  and verify that the distance from the center of your chosen cell to the source cell is approximately the same as the value displayed in the Identify Results window.

Step 9 Determine the distance from a source cell to a diagonal neighbor

Use the Identify tool to check the distance from the cell in the lower left-hand corner to the source cell.

The distance should be 14.142136, which is approximately 10 times the square root of 2. ($10^2 + 10^2 = 200$; the square root of 200 is approximately 14.142136.)

REVIEW CONCEPT

Step 10 Save your work

From the File menu, choose Close All. Then, from the File menu again, choose Save Project As and save the project as **L1_ex02.apr** in the **proxalsallesson01** directory.

If you are continuing on with the next exercise, leave the project open. Otherwise, close the project and exit ArcView

TOPIC 2: Multiple source cells and limited distance

Many distance problems involve several source cells instead of a single one. Perhaps you are flying along in your private plane and because of an ominous new noise coming from the engine, you develop a strong interest in knowing the distance to the closest of several airports in the area. The airports might be recorded as source cells. No matter which cell you were in, the distance to the closest airport would be shown.

Another example: Suppose you have a neighborhood of houses that need to be connected to a power line. The power line is represented as a line of source cells; houses may be built on any other cells. You want to find the shortest distances from each house to the power line so you can calculate the minimum length of wire required. Also, due to electrical requirements, there is a maximum distance the wire can be strung, so you will also need to identify the cells that are more than that distance away.

Problems of this sort may be solved with the EucDistance request, using as input a source grid with multiple source cells

Concept

Finding the closest of many source cells

The grid of source cells that serves as input to EucDistance may consist of several single cells, clumps of cells (contiguous cells), a linear structure of several cells, or any combination of these. For each cell in the output grid, the straight-line distance to the closest source cell is calculated. This happens automatically. You simply apply EucDistance as you did in the previous exercise and the newly generated grid contains the closest distance

Concept

Excluding distances beyond a certain threshold

If the problem warrants it, you may limit the search for the closest source cell so that it does not exceed a given value. To do this, you use a slightly more involved syntax for EucDistance:

```
[aSourceGrid].EucDistance( Nil, Nil, MaxDist)
```

MaxDist is a number, in the units of the grid, which says "if you have to look further than this to get to a source cell, just forget it." Such a number may be called a limit, a cap, a cutoff value, or a threshold.

In the exercise, you used Nil for this third parameter because you didn't want to limit the search distance. That is, you wanted all cells in the grid to take on a value, regardless of how far they were from the source cell

Exercise

Many cells with limited distance

In this exercise, you will apply what you learned in the previous one. The new features that you will work with are: (a) more than one source cell, and (b) a cap, or cutoff value, so that no value greater than the cap will be recorded in any cell.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Open the project

If the project is open from the previous exercise, go to Step 2. Otherwise, start ArcView if it is not already running, navigate to the **proxsa\lesson01** directory and open **I1_ex02.apr**. Load the CellTool and Spatial Analyst extensions if they are not already loaded.

Note: If you are running ArcView GIS 3.1, you see an Update l1_ex02.apr message box. Click No to dismiss this box.

You see a view containing the results of the previous exercise.

Step 2 Add a new source grid

Turn off Onecell and MapDist1. Add the grid theme Manycells to the view and turn it on.

Notice the presence of several cells containing data, while the rest of the grid shows No Data. There is still a single source cell in the southwestern portion of the grid, but there is also a line of source cells in the northeast and a clump (cluster) of source cells in the northwest.

VIEW RESULT

You will generate a grid containing distances to the nearest source cells, provided that the distance is no further than 30 units.

Step 3 Make a new distance grid

Open the Map Calculator and enter the following expression:

```
[Manycells].EucDistance (Nil,Nil,30)
```


Click Evaluate, then close the Map Calculator.

REVIEW CONCEPT


Step 4 Examine the distance grid

Turn on the new grid theme (Map Calculation 1), make it active, and change its name to **DistManycells**. Arrange the themes in the Table of Contents so that from top to bottom they are ordered Fishnet, Manycells, then DistManycells.

VIEW RESULT

Make sure that DistManycells is the active theme. Use the Identify tool  to verify that the distances shown by cells in the new theme are indeed associated with the closest source cells in Manycells.

REVIEW CONCEPT

The cutoff value of 30 on the distance resulted in some No Data cells in the new theme. Verify by using the Measure tool  that the No Data cells are more than 30 units away from any source cells. Note that 30 is the largest value in the legend of DistManycells.

Step 5 Save your work

Save the project in the **proxalsa\lesson01** directory as **L1_ex03.apr**. If you are continuing on to the next exercise, leave the project open on your screen. Otherwise, close the project and exit ArcView

TOPIC 3: Other factors that influence cost

We began this module saying our primary concern was the cost associated with moving (moving being defined in a very general sense) from one place to another. We also said that distance between points on the Cartesian plane was a major factor in the cost, and we have spent much of the preceding discussion discussing how to obtain that distance.

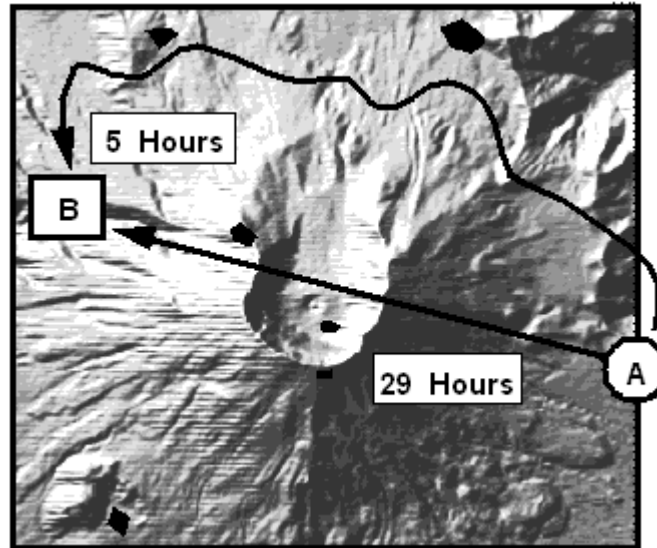
Frequently, however, distance is not the only, or even the principal, cost of moving from one place to another. An extreme example is of a person living in New York City who wants to visit a second person who lives nearby. In fact, the two people live in 33rd-floor apartments, towards the middle of parallel long block streets, and their apartments share a common back wall. They actually live only a few feet from one another. But to travel from one apartment to the other requires two elevator rides, encounters with doormen or buzzer systems, and a long walk (or cab ride). Clearly, the calculation of simple Euclidean distance is not the only tool we need to determine cost.

As a second example, suppose you want to take a hike. In between your starting and ending locations lie both swamp and dense forest. By taking a longer route you can walk through pasture. The time and difficulty of the route you take will be a function of both the distance and the difficulty of making your way through various environments.

We can generalize these problems by creating a "cost surface." We develop this cost surface so that each grid cell contains a number that indicates the cost of going through a unit of distance in that cell. If it is three times as hard to go through the forest as it is to go through the pasture, we might assign values of 1 in the pasture cells and 3 in the forest cells. If a certain place in the swamp poses a threat of people being eaten by alligators, we might assign a very high cost (say 10,000) to traversing the distances in the swamp cells.

A third example: While off-road vehicles are becoming increasingly popular, it is still less expensive to travel by automobile from one city to another using paved roads. The cost of traveling a straight line in this case, in terms of speed, danger, and possible bouts with the law, is prohibitive.

Consider the graphic below. The time to go from A to B is greatly reduced by going around the mountain rather than over it.



While the shortest path between points A and B is a straight line, it not the fastest path because the cost for crossing over the mountain is much greater than going around it.

In short, EucDistance has the limitation of assuming that the cost of crossing distances in any cell on the way to a source cell is the same as the cost of crossing distances in any other cell. You will now learn how to indicate that some cells are more expensive to cross than others. You will also learn how to find the most inexpensive route (the least-cost path) even though it may not be the shortest.

Concept

The CostDistance mechanism

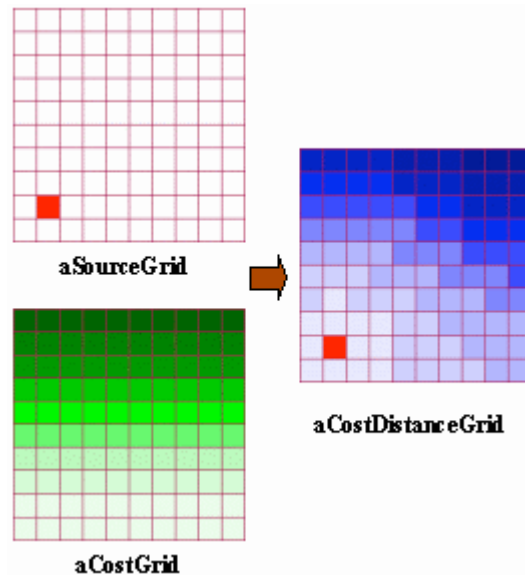
Spatial Analyst provides the Avenue request CostDistance for calculating the least cost of getting from any cell to a source cell. Two input grids are involved: the source grid and a cost surface grid, each of whose cells indicate a price of traversing each unit of distance in that cell.

The overall cost of the path from a given cell to the closest source cell is calculated on the basis of the least-cost path between the two cells. Calculating the total cost involves summing up the costs of crossing all the individual cells along the least-cost path. The cost of crossing an individual cell is found by multiplying the distance across that cell by the value found in the geographically equivalent cell in the cost-surface grid. Thus, the accumulated cost found in the output grid is the sum of the products formed by the (a) distance across each cell and (b) value (the "price per unit distance for going through") of each cell.

Concept

The CostDistance request

If you want to generate a grid similar to that made by EucDistance, but one that will place in each cell the cost of traveling from that cell to the "cheapest to get to" source cell, you can use the CostDistance request. CostDistance operates on a source grid and a cost grid to produce a cost distance grid.



A cost distance grid showing the least-cost path to a source cell is produced by CostDistance using a source grid and a cost grid as input.

Avenue syntax for CostDistance:

```
[aSourceGrid].CostDistance(aCostGrid,Nil,Nil,MaxCost)
```

You apply the CostDistance request to the grid containing the source cells, giving as the first parameter the name of the grid containing cells with costs (the cost surface grid). The resulting grid shows up as a map calculation.

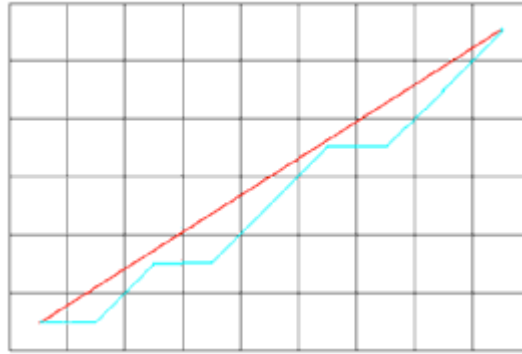
Concept

Path calculation in EucDistance and CostDistance

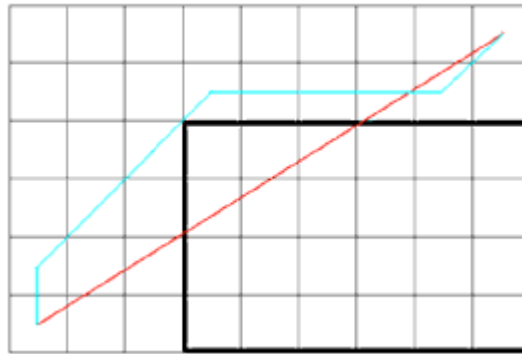
It is worth mentioning that the straight-line distance from a given cell produced by EucDistance and the least-cost path generated by CostDistance are calculated in completely different ways. EucDistance calculates the direct, straight-line distance from each cell center to the closest source cell center, regardless of how this path slices through intervening cells. The CostDistance path, however, must pass through cell centers, usually generating a sequence of line segments that change direction at the center of each cell.

Therefore, even when crossing a cost surface where each cell cost is 1, so that a straight line would be the least-cost path, CostDistance will usually produce a slightly longer path – because the path proceeds from cell center to adjacent cell center. This sort of "connect the dots" behavior will not result in a much longer path than the straight-line distance, but observing it illustrates the difference in the ways the two output grids are calculated.

In CostDistance, the path proceeds from one cell center to the center of one of the eight adjacent cells, always with the aim of minimizing the cost of getting to the source cell. In other words, except for a few special cases (e.g., due north, straight southwest), the path generated by CostDistance will not be a straight line, but will zigzag so as to take in the cell centers along the way.



The EucDistance direction is shown in red. A possible path generated by CostDistance is shown in blue, under conditions of a uniform cost grid.



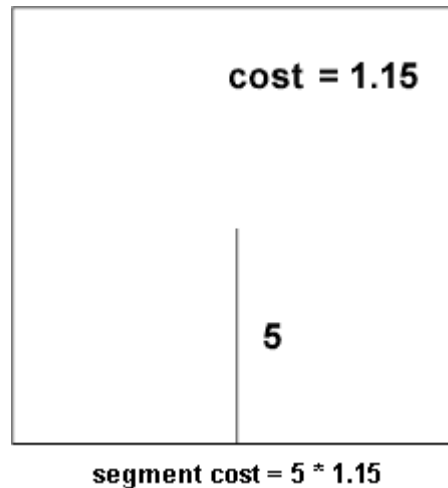
The EucDistance direction is shown in red. A possible path generated by CostDistance is shown in blue, under the condition that high costs exist in the cells within the box

Example

Understanding how total costs are calculated

Calculating cost distance is somewhat involved. Take as an example the accumulated cost of traveling directly from the north towards the source cell. The travel is assumed to take place from cell center to cell center. The total cost is calculated by adding the costs of crossing half or all of individual cells, which have different costs per unit distance.

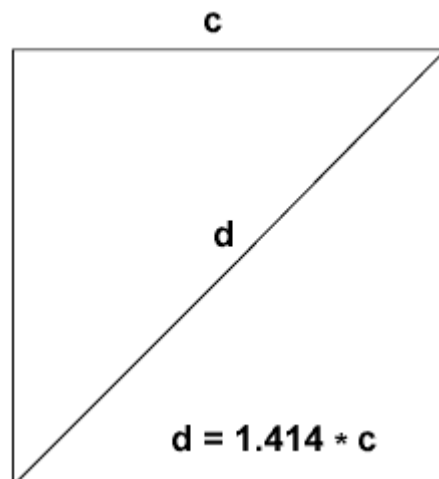
Start with the segment of the path from the center of the top cell to its lower edge. The length of this segment is 5 (half the total distance of 10 across the cell). The cost associated with the cell is 1.15 per unit of distance. The cost of this segment, then, is the product of the distance and the cost per unit distance: $5 * 1.15 = 5.75$.



The top (northernmost) cell in the path.

The next segment, heading south, goes completely across a cell whose cost is 1.05. Because the distance is 10, the cost of traversing the cell is 10.5. This amount is added to 5.75 to give a total accumulated cost so far of 16.25. This process continues until the center of the source cell is reached. The calculated cost amount is then assigned to the cell where the path originated. Actually, the accumulated costs are computed from the source cell to the other cells, but since addition is commutative (it doesn't matter in what order you add numbers, you get the same answer) you may think of obtaining the cost as described above.

When the path goes from cell corner to corner, the distance is longer than if the path goes from the side of a cell to the opposite side. How much longer? Multiply the side-to-side distance by the square root of 2: 1.41421356237.



The distance diagonally across a cell is the cell size multiplied by the square root of 2.

For example, the cost of the diagonal segment in the cell in the upper right corner (northeasternmost) cell towards the source cell would be:

$$5 * 1.414 * 1.15 = 8.1305$$

Each possible path from the source cell to each other cell is either calculated or considered in some way. That can be a lot of paths. This type of calculation may take place several times for

each cell, there can be hundreds of thousands of cells, and the computer may make many attempts for each cell to find the least-cost path.

Exercise

Develop a grid with CostDistance

This exercise will illustrate the principle of calculating cost based on both distance and a cost surface.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Open the project

If the project is open from the previous exercise, go to Step 2. Otherwise, navigate to the **proxalsa\lesson01** directory and open **I1_ex03.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I1_ex03.apr message box. Click No to dismiss this box.

You see a view containing the results of the previous exercise.

Step 2 Prepare the view

Turn off all visible themes except Fishnet. In the steps that follow, keep Fishnet at the top of the Table of Contents to delineate the cells of the displayed grids.

Step 3 Add the cost surface theme

From the **proxalsa\lesson01** directory, add the **Tolls** theme and turn it on.

This is an artificial cost surface grid theme. The surface is divided into rows of cells; all the cells in a given row have the same value. (Reminder: Move Fishnet to the top of the TOC, if you have not already done so.)

VIEW RESULT

Imagine that the center of each cell contains a toll booth. Depending on which row the cell is in, the toll booth charge varies from 25 cents to \$1.15 to pass through a unit distance of the cell. In the Tolls grid, all the cells in the bottom row have values of 0.25; the top row cell values are 1.15.

So, to cross a cell in the top row in an east-west direction would cost \$11.50 (1.15 dollars times 10 distance units). (It costs more to cross the cell in a diagonal direction because the distance is longer; more about that later.)

Make Tolls the active theme. Use the Identify tool  to examine the values of cells in

other rows.

[VIEW RESULT](#)

Step 4 Examine the source grid

Move Onecell to the top of the Table of Contents and turn it on.

Recall that Onecell was the source grid you originally worked with that contained a single source cell.

Step 5 Apply CostDistance

Open the Map Calculator and enter the following expression:

```
[Onecell].CostDistance([Tolls],Nil,Nil,Nil)
```

Click Evaluate and close the Map Calculator.


Step 6 Arrange the themes and look at the new legend

Move the result theme (Map Calculation 1) below Fishnet and Onecell, make it active, and turn it on. Change its name to **TotCostOnecell** (the total cost of the path from each cell to the source cell in the theme Onecell).

Each cell contains the cost of moving across the cost surface grid from that cell to the source cell. You can get a general idea of the relative costs by looking at the colors of the grid.

[VIEW RESULT](#)

Step 7 Examine the results

Using the Identify tool , examine the total cost of traversing the eight cells due north of the source cell (pick the cell on the top row directly above the source cell). Now look at the total cost of traversing the eight cells due east of the source cell.

It is obviously cheaper to come from the east (about \$23.50) than from the north (\$60).

The overall range of costs is from 0 (from the source cell to the source cell) to approximately \$82.

Make the Tolls theme active and again use the Identify tool to display the values of the Toll cells that are traversed.

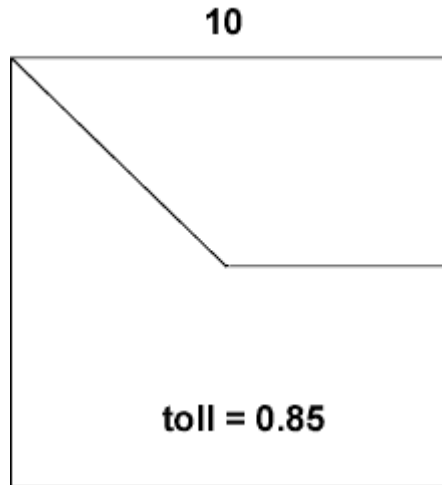
To come from the east, one need only traverse cells with costs of 0.35 or less. To come from the north, the cell costs are 1.15, 1.05, 0.95, and so on. The most expensive trip is from the northeasternmost cell--it is furthest away and in an area where the cost per cell is highest.

Step 8 Save the project

Save the project as **L1_ex04.apr**. If you are continuing on to the next exercise, leave the project open. Otherwise, close the project and exit ArcView

Challenge:
How are costs calculated?

What would the cost be for the path going through a cell (cell size 10; cell cost 0.85) as in the figure below?



Solution to challenge

$$10.2595 ((5 + 5 \cdot 1.414) \cdot 0.85)$$

TOPIC 4: Getting more information: Paths and allocations

You learned that you can calculate the shortest distance between two places using the EucDistance request and you can find the least-cost path with the CostDistance request, but there are still a couple of major questions: Which source cell is closest to (or least expensive to reach from) each given cell in the grid, and what is the direction or path from each given cell?

To answer these questions, first for straight-line distance, you can use the two parameters in the EucDistance request that were specified as Nil in previous examples

Concept

Identifying the closest source cell and how to get there

Below is a more complete version of the Avenue syntax for EucDistance:

```
[SourceGrid].EucDistance(DirGrid, AlloGrid, MaxDist)
```

The parameters of an Avenue request entered into the Map Calculator usually provide input. But here, the names used for DirGrid ("direction grid") and AlloGrid ("allocation grid") refer to output shapefiles that are automatically generated when the request is executed.

Also, when you use Avenue requests in the Map Calculator, the output usually is named by the system (e.g., Map Calculation 8) and automatically added to the current view. Not so with DirGrid and AlloGrid. These must be added to the view if you want them there. Additionally, if you want them stored as permanent grids for later use, you have to convert them to grids.

When EucDistance is applied, DirGrid's cells each contain the direction to the nearest source cell, in degrees clockwise, based on due north as 360 degrees (or 0 degrees).

AlloGrid's cells contain the value of the nearest source cell

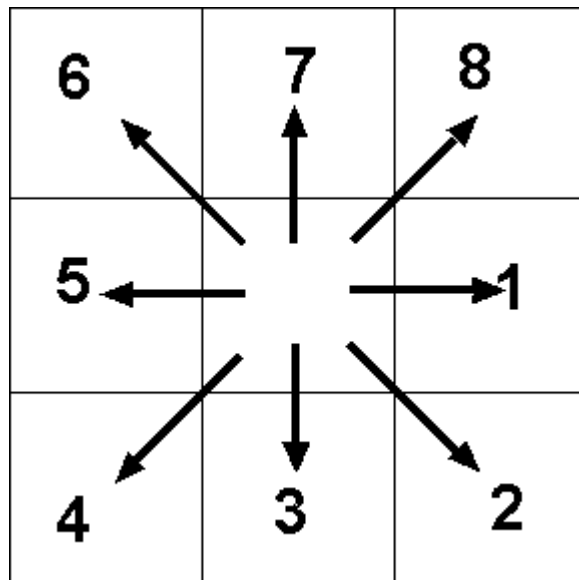
Concept

Direction and allocation grids for CostDistance

The CostDistance request is similar to the EucDistance request in that both can be used to generate direction and allocation grids. Both requests allow caps to be placed on the maximum values in the primary grids they generate. CostDistance differs from EucDistance in three major ways:

- it uses a distance-weighting grid
- the way in which the direction grid points along the path
- it calculates the least-cost path rather than the shortest path

Because the path generated by CostDistance goes from cell center to cell center, each cell in the direction grid contains information that indicates to which of the eight neighbors the path goes next. This information is coded according to the following diagram:



Direction code for the next cell in the path.

Or, in words:

- 1 means the next cell in the path is to the east (90 degrees)
- 2 means the next cell in the path is to the southeast (135 degrees)

- 3 means the next cell in the path is to the south (180 degrees)
- 4 means the next cell in the path is to the southwest (225 degrees)
- 5 means the next cell in the path is to the west (270 degrees)
- 6 means the next cell in the path is to the northwest (315 degrees)
- 7 means the next cell in the path is to the north (360 degrees)
- 8 means the next cell in the path is to the northeast (45 degrees)

Concept

Another grid for showing least-cost paths

With the direction grid that you can generate with `CostDistance`, all you have in a given cell is the identification of the next neighboring cell on the least-cost path. Under some circumstances, this direction might be 180 degrees from the general direction in which the source cell lies. The question arises, therefore: What is the least-cost path from a given cell to the appropriate source cell? You can answer this question most easily by using the `CostPath` request, specifying as inputs:

1. A destination grid containing cells with values that form the other end of the path from the source cells
2. The accumulated cost grid calculated by `CostDistance`
3. The direction (sometimes called backlink) grid also calculated by `CostDistance`

The syntax of `CostPath` is:

```
[aDestinationGrid].CostPath(distanceGrid, directionGrid, byZone)
```

The Boolean variable `byZone` controls whether a single path for multiple source cells is calculated or whether multiple paths for multiple source cells are calculated. See the `CostPath` help topic for more information.

The least-cost paths are displayed on a new grid. Most of the cells in the grid will have No Data values. Cells with integer values represent the different paths linking the source cells to the destination cells.

Exercise

Use `EucDistance` to create direction and allocation grids

In this exercise, you will again use the `EucDistance` request. Because you will specify parameters with output filenames, the request will generate two additional grids.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Open the project

Start ArcView, if necessary. If the previous exercise project is open, go on to Step 2. Otherwise, navigate to the **proxalsalesson01** directory and open **I1_ex04.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I1_ex04.apr message box. Click No to dismiss this box.

You see a view containing the results of the previous exercise.

Step 2 Set up the view

Add the theme **Twocells** to the view and turn it on. Move Fishnet to the top of the Table of Contents.

[VIEW RESULT](#)

Step 3 Use EucDistance to make three grids

You will calculate the distance from each cell to the closer of the source cells in the Twocells theme.

Open the Map Calculator and enter the following expression, noting the use of quotes and the AsFileName request:

```
( [Twocells].EucDistance("Dir".AsFileName,"Allo".AsFileName,50))
```

Click Evaluate and close the Map Calculator.

Note: If you have worked through this exercise before, you will have to use the Source Manager to delete or rename the existing Dir and Allo grids. To get to the Source Manager click on Manage Data Sources in the File menu.

Step 4 Add the new themes to the view

Map Calculation 1 contains the distances to the closest source cells and is automatically added as a theme to the view. Change its name to **CostTwocells**. Also add to the view the two new grids created by EucDistance: **Dir** and **Allo**. Arrange the themes in the TOC this way:

Fishnet
Twocells
Allo
Dir
EucTwoCells

Turn on Fishnet, Twocells, and Allo; turn all other themes off. Make Dir the only active theme so that you can examine it later with the Identify tool.


[VIEW RESULT](#)

Step 5 Examine the allocation of cells to the source cell

Now look at the resulting display.


Using the colors of Allo and the legends of Allo and Twocells, notice that like cells are allocated to (clustered around) the nearest source cell, and that the allocated cells take on the value of that source cell. Cells greater than 50 distance units away from a source are set to No Data. Note that there are several of these.

Step 6 Examine the direction grid

Examine Dir, the active grid, with the Identify tool . Verify that the direction in each cell appears to be the compass direction you would travel to reach the source cell. The direction 360 is used to indicate north; the value 0 is placed in those cells that correspond to the source cells.

Step 7 Use the CellTool extension

If the Cell Tool extension isn't already loaded from a previous exercise, load it now. You may have to resize the ArcView window to see the new tools.

Click the CellDirection tool  on the toolbar. Do not save the graphics. Use the CellDirection tool to put arrows in several cells of Dir.

The direction arrows will point at the nearest source. You get a graphic picture, signposts as it were, of how to proceed from each cell to its nearest source cell.

[VIEW RESULT](#)

Step 8 Save the project

Save the project as **L1_ex05.apr**. If you are continuing on to the next exercise, leave the project window open. Otherwise, close the project window and exit ArcView

Exercise

Use CostDistance to make direction and allocation grids

In this exercise, you will expand on what you have learned. You should be able to solve the following problem in a new view. Basically, you are repeating the previous exercise using CostDistance instead of EucDistance.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Open the project

Start ArcView, if necessary. If your project from the previous exercise is open, go to Step 2. Otherwise, navigate to the **proxalsallesson01** directory and open **I1_ex05.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I1_ex05.apr message

box. Click No to dismiss this box.

You see the view from the previous exercise.

Step 2 Create a new view

Close View1 and, from the project window, create a new view: View2.

Step 3 Generate the accumulated cost, direction, and allocation grids

Open the Map Calculator and use the CostDistance request, inputting a source grid named TryThis and the weighting grid Tolls, to produce these three new grids:

- Map Calculation 1--represents the cost of going most cheaply from any cell to a source cell in the source grid named TryThis.
- Dd--the direction grid, sometimes called a backlink grid, because it can be used to find the least-cost path from any cell back to the source cell.
- Aa--the allocation grid.

Cap the cost at a maximum of **30**.

What you are being asked to do here is similar to what you did previously with EucDistance. Try to make the new grids without looking first at the hints below.



[Hints](#)

[VIEW RESULT](#)

Step 4 Add themes to the view

Add the grid themes Aa and Dd to the view. Also add the arc component feature theme Fishnet to the view.

To add the arc component rather than the polygon component, in the Add Theme dialog, click on the file folder icon next to fishnet, then click on **arc** and click OK. If you click on polygon you will get an opaque fishnet.

Step 5 Examine the accumulated cost grid

Place the themes in the TOC in the following order:

Fishnet
TryThis
Aa
Dd
Map Calculation 1 (which you should rename to **Cc**)
Tolls

Turn off all themes except Fishnet and TryThis. Make Cc the active theme and turn it on. Use the Identify tool to verify that the cost from the cell that is 9 from the top and 9 from the left is about 21.

The top row is row 0 and the left-most column is column 0, so the cell identifier (row,column) is 8,8.

[VIEW RESULT](#)

Turn off Cc.

Step 6 Examine the allocation grid

Make Aa active and turn it on. Verify that cell 8,8 is allocated to (generates the least-cost path to) a source cell whose value is 50. Determine where the cells of value 50 are.

[VIEW RESULT](#)

Turn Aa off.

Step 7 Compare the allocation grid (Aa) with the source grid (TryThis)

Display TryThis with a color ramp by double-clicking on it in the TOC to open the Legend Editor. In the Color Schemes dropdown list, choose Warm Tones. Click Apply, then close the Legend Editor. Repeat the process for grid Aa.

Turn both themes on. Now, by toggling Aa off and on, you can see the cells in Aa that are clustered around the source cells in TryThis.

Because the two themes have the same color scheme, Aa seems to "grow out of" TryThis. This is the essence of the allocation operation.

[VIEW RESULT](#)

Turn off Aa.

Step 8 Examine the direction grid

Make Dd active and turn it on. Verify that to head towards the source cell from cell 8,8 you would first go to cell 7,9. That is, cell 8,8 will have the value 4, indicating a direction of southwest.

Trace the rest of the path from cell 8,8. Notice how it takes a southern route despite the fact that the shortest distance (the straight-line distance) is due west. Why is this?

Step 9 Reclassify the direction code to degrees

As you probably noticed, the colors and the coding of Dd are not much help in getting a picture of the path from 8,8 to its source cell 8,1. So you will put some arrows in the cells.


Make sure that Dd is the active theme. From the Analysis menu, choose Reclassify.

Click the Unique button to show unique values. Type in the new values listed below, which are in degrees, based on the compass direction north as 360. Press Enter after each new entry. It may be that not all the Old Values listed below will be represented; just type values for those that are.

Old Value	New Value
0	No Data

1	90
2	135
3	180
4	225
5	270
6	315
7	360
8	45
No Data	No Data

Click OK to generate a new theme called Reclass of Dd. Turn Reclass of Dd on and make it active.

Click the CellDirection tool  on the toolbar. Do not save graphics. Use the CellDirection tool to examine Reclass of Dd.

The direction that the arrow points will be the same as the direction indicated by the number in grid Dd. You will get a graphic picture of which cell is next on the least-cost path.

Trace the path from 8,8 to its source; you should get a graphic picture of the process you did in Step 8 above.

[VIEW RESULT](#)

Trace some other paths with the CellDirection tool.

Step 10 Save the project

Save the project as **L1_ex06.apr** in the **proxalsallesson01** directory. Leave the project window open if you are continuing on with the next exercise. Otherwise, close the project and exit ArcView

Exercise

Calculate a least-cost path

In this exercise, you will use the Avenue request CostPath to show the least-cost path between a source and a destination.

The CostPath request can handle multiple cells in both the destination grid and the source grid, but this exercise will demonstrate the request using only one cell in each.

If it has not become clear by this point, we should state the obvious: the cost (and/or the path) of going from A to B, as calculated by the grid manipulations you have been directing,

is the same as the cost (and/or the path) of going from B to A.

There are certainly instances in which this assumption is not warranted. One example would be if A is at the bottom of a mountain and B is at the top. Another example is driving in a city that has one-way streets.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Open the project

Start ArcView, if necessary. If the previous exercise project is open, go to Step 2. Otherwise, navigate to your **proxalsallesson01** directory and open **I1_ex06.apr**.

Note: If you are running ArcView GIS 3.1, you see an Update I1_ex06.apr message box. Click No to dismiss this message.

Step 2 Open a new view and add the necessary themes

Open a new view. Add the themes **Fishnet** (the arc feature type from Feature Data Source), and the grids **OneCell**, **Tolls**, and **CPDestination**. Turn all the themes on and arrange them in the following order: Fishnet, OneCell, CPDestination, Tolls.

VIEW RESULT

The goal of this exercise is to use CostPath to create a new grid that shows the least-cost path between the single cell in CPDestination and the cell in OneCell. You could, of course, determine this path by exploring the destination grid, but CostPath generates a grid with the path delineated.

Step 3 Make a least-cost grid and direction grid for OneCell

CostPath requires several inputs. The request itself operates on the destination grid, but CostPath has parameters that must have been generated previously by the CostDistance request. These parameters are the costs grid and the direction grid. So, you need to first use the CostDistance request.

Open the Map Calculator and enter the following expression:

```
[OneCell].CostDistance([Tolls], "CPDirection".AsFileName, Nil, Nil)
```

Click Evaluate, then close the Map Calculator. Make Map Calculation 1 active and change its name to **CPCosts**.

Add CPDirection as a theme to the view. Move the Fishnet theme to the top of the TOC. Turn off the Tolls grid.

VIEW RESULT

Step 4 Make a grid containing cells constituting the least-cost path

Use the Map Calculator to evaluate the following Avenue request:

```
[CPDestination].CostPath([CPCosts], [CPDirection], True)
```

Step 5 Examine the new theme

Make the new theme, Map Calculation 1, active, turn it on, and rename it to **LeastCostPath**. Move Fishnet and LeastCost Path to the top of the TOC. Turn them on; turn all other themes off.

[VIEW RESULT](#)

The value of the source cell is 1. The values of the cells in the path are all 3s. If there were another path, it would have the value 4, the next one 5, and so on.

What happened to 2? It is reserved for cells that lie on two or more paths--a possibility when you have multiple source or destination cells. Note the circuitous path that was calculated. Is it surprising in light of the Tolls cost surface grid?

Step 6 Save your work

Save the project to the **proxsa\lesson01** directory with the name **L1_ex_yi**, where "yi" represents your initials. Close the project. Exit ArcView

Exercise

Put the tools together: Site a regional park

The power of GIS is most evident when you use a number of tools in concert. This is also the stage where the creativity comes in--and sometimes the frustration as well, because frequently, even if datasets are available, they are not in the form you can most easily use.

This exercise uses the CostDistance tool, which calculates and tries to minimize costs according to the "price you pay" in traveling between a source and a destination. This price has many names, among them: cost surface, friction, resistance to flow, resistance to travel, and impedance.

Suppose that the county of San Bernardino, California would like to build a regional park somewhere near the city of Redlands. One of the criteria for the park location is that it must be accessible to those who might drive some distance using freeways or interstate highways.

For this exercise, accessibility is defined very specifically: the less time it takes to drive from any freeway offramp, the more accessible the park is deemed to be. The source grid, therefore, will consist of cells containing offramps.

Your approach will be to create a cost surface, expressed in

minutes required to travel. You will derive travel cost weights from speed limits on the non-freeway roads in the area. The cost surface, or weighting grid, will be the inverse of the speed (divide the speed into 1.0, so you get, say, hours per mile, rather than miles per hour). The grid will be based on the time it will take a car to traverse each given cell in the study area.

If you have not downloaded the exercise data for this module, you should [download the data now](#).

Step 1 Start ArcView and open the project file

Start ArcView. If it is already running, exit it and restart. Load the Spatial Analyst extension and the CellTools extension.

Note: If you are running ArcView GIS 3.1, you see a Welcome to ArcView GIS dialog. Click Cancel to close this dialog.

From the File menu, open the project **sitepark.apr** located in your **proxals\lesson01\spatial** directory .

Note: If you are running ArcView GIS 3.1, you see an Update SitePark.apr message box. Click No to dismiss this message.

From the File menu, choose Set Working Directory and set the project's working directory to **proxals\lesson01\spatial**.

[VIEW RESULT](#)


Step 2 Examine the themes and the options

Once the project opens, you see the Distance to Off Ramps view (which is what the view will become). It contains two themes: Study Area, which is the area around Redlands, and Roads.

A couple of issues might come to mind. First, how are you going to identify the offramps? Second, because your experience has been with grids in which all cells could be traversed, how are you going to exclude cells that have no roads?

You can select the ramps based on values in the Roads table. And you can set a high impedance, or cost, on cells that do not have roads through them. Once you have identified cells with road segments, you can set their impedance based on speed limits, which will also be available in the Roads table. You'll have to be pretty ingenious to modify the data so it will work for this project.

Step 3 Examine the Roads table

Make Roads the active theme and open the Attributes of Roads theme table by clicking the Open Theme Table button .

[VIEW RESULT](#)

Note that some of the data massaging has already been done for you. The roads have been clipped to the study area, and the number of roads has been reduced to 12 sizable networks (from 29,326 road segments in the valley). The 12 road networks were derived by merging similar road types based on their Census Feature Classification Codes (CFCCs). Roads is a feature-based theme, not a grid theme.

The assumption has been made that all roads of the same CFCC will have the same speed limit. (The speed limit data is in a table that you will later join to the Attributes of Roads table).

Step 4 Look at more road information

Make the project window (sitepark.apr) active and click the Tables icon. Open the **cfcc.dbf** table.


Notice the different road classifications and the speed limits associated with each.

Make the Cfcc field active.

[VIEW RESULT](#)

Step 5 Join the two tables

In the Attributes of Roads table, make the Cfcc field active. (The idea here is to join the tables by making the key field active in each.)


Click the Join button .

The data contained in cfcc.dbf is added to the Attributes of Roads table and cfcc.dbf closes. Resize the table so you can see all its fields.

[VIEW RESULT](#)

The Roads table identifies the offramps with a code (A63). You want to make a grid containing source cells that represent road segments with the A63 designation; you want to assign No Data to all other cells.

Step 6 Select the offramps

With the Attributes of Roads table active, click the Select Record tool  and select the Offramp record (the 10th record down).

[VIEW RESULT](#)

Next, make the Distance to Offramps view active. You see the offramps selected in the view, highlighted in yellow.

Step 7 Prepare to create the source grid

With the Distance to Off Ramps view active, from the Analysis menu, choose

Properties.

The goal here is to define the extent of the source grid you are about to generate and set its cell size. You want the Analysis Extent to be the Same as Study Area.

To set the Cell Size, type **150** (the units are feet) and press Enter. You should see the number of rows and columns defined as 390 and 508.

[VIEW RESULT](#)

Click OK.

Step 8 Create the source grid

Make Roads the active theme. From the Theme menu, choose Convert to Grid.

Convert to Grid creates a grid with values in those cells that contain selected features; other cells in the grid contain No Data. (That is, if a feature from the linear shapefile Roads runs through a cell, the cell has a value in the resulting grid.)

In the Convert Roads dialog, navigate to your **proxalsa\lesson01\spatial** directory and name the new grid **Ramps**.

Click OK.

In the Conversion Field: Roads dialog, choose Cfcc. (The field you select here will show up as the "S_value" in the Attributes of Ramps table. You won't actually use the value, because you want only the ramp locations, but you must select one field to satisfy the input requirements of Convert to Grid.)

Click OK to start the conversion. When prompted, add the grid as a theme to the view. (Turn off the Study Area theme if its background interferes with seeing colors of other themes.)

[VIEW RESULT](#)

Step 9 Examine the offramps

Make the Ramps theme active. Use the Identify tool to check the values for some of the ramps. (You may have to zoom in to click on the ramps.)

[VIEW RESULT](#)

Note that the S_value (select value) of the ramp cells is A63 and that there are 391 of these cells. All the other cells are set to No Data.

Step 10 Prepare to create a cost surface


You have just made the source grid. The next step is to create a cost surface grid that represents "resistance to travel" by assigning travel costs to every cell in the study area.

In those cells where there are roads, you will base the travel cost on speed limits. In areas where there are no roads, you will simply assign a high cost based on a very low speed limit (5 miles per hour) primarily to keep cars on the roads. It would be

reasonable to ask, "Why not just assign a very large impedance to off-road cells?" You won't do this because you want to allow every cell in the study area the possibility of access. If you assign an impossibly high impedance to cells that don't have roads in them, you exclude those cells from consideration for the park because they could not be accessed unless a road ran through them. Presumably, if a group of cells is to become a park, a low-speed access road will be built.

You do, however, want to completely exclude freeways to force travel on local roads and to prevent a path from crossing the freeways. In the following steps, you will select highways, county roads, and local roads to make a grid based on their speed limit values. You will then reclassify the non-road cells to give them a speed limit of 5 mph.

Step 11 Select the proper local roads

Make the Attributes of Roads table active. Clear any selections. (To clear selections, press the Select None button  or choose Select None from the Edit menu.)

Now select the US Highway, County Road, and Local Road records. (Use the Select Record tool and the Shift key.)

[VIEW RESULT](#)

Step 12 Create the speed theme

Make the Distance to Off Ramps view and the Roads theme active. From the Theme menu, choose Convert to Grid. When you are prompted to specify the location of the new grid, navigate to the **proxalsallesson01\spatial** directory and name the new grid **Speed**.

Click OK.

Step 13 Put Speedlimit in the cells

In the Conversion Field: Roads dialog, choose Speedlimit from the scrolling list. Click OK, and answer Yes to add the new theme to the view.

Make the Attributes of Roads table active, clear all selections, then close it. Make the Roads theme active, turn the Study Area, Ramps, and Roads themes off if they are still on and turn the Speed theme on.

[VIEW RESULT](#)

The Speed theme is a raster representation of the selected roads; the values in the cells are the speed limits. Notice from the map and the legend that the cells between the roads were assigned the No Data value. You want to change the No Data cells to 5 mph, using the Reclassify tool.

Step 14 Convert the No Data values to slow speeds

Make the Speed theme active. From the Analysis menu, choose Reclassify. Click the Unique button to show the unique values.

A table showing the values of speed limits appears. The Old Values are 30, 45, 50, and No Data. The suggested new values are 1, 2, 3, and No Data. You do not want

the suggested values. You want the old values, except for the No Data value, which you want to be 5.

Type in **30, 45, 50**, and **5** (pressing Enter after each) for the new values.

[VIEW RESULT](#)

Click OK.

Step 15 Verify that all cells have speed values

Turn off the Speed, Ramps, and the Study Area themes, if they are on. Move the Roads theme to the top of the TOC and turn it on. Turn on the Reclass of Speed theme and make it active.

Next, zoom in on an area that includes part of the interstate where the ramps are.

Notice how the speed limit cells follow along with the local roads. Notice also that all the cells between the roads now have the 5 mph value.

Zoom in some more to see the relationship between the feature theme Roads and the cell theme Reclass of Speed. When you are finished looking, choose Full Extent from the View menu.

Now you have the raw material (speeds at which those going to the park may drive) for building a cost surface. What numbers do you put in the cost surface cells? You cannot put miles per hour (mph), because then higher speeds would imply greater resistance to flow--the opposite of what you want. You could use the reciprocal, hours per mile. 50 mph is 0.02 hours per mile; 25 mph is 0.04 hours per mile. 0.04 is a larger amount than 0.02, and that larger amount is what you want to see in your cost grid.

There is another consideration: it is important to express the cell costs in terms of the units of the grid and the problem statement. You are interested in drive times in minutes. The grid is measured in feet. The numbers you have been given to work with are miles and hours, but what you want, strange as it may seem, is minutes per foot!

How do you convert hours per mile (hpm) to minutes per foot? Multiply hpm by a conversion factor: 0.0113636, which is the number of minutes it takes to travel a foot at the rate of one hour per mile.

One hour per mile is 60 minutes per 5280 feet. $60/5280$ equals 0.0113636--your conversion factor. So, for example, 50 miles per hour is $1/50$ hours per mile. $1/50 * 0.0113636 = 0.000227272$ minutes per foot. If a cell is 150 feet across, say from east to west, it therefore takes $150 * 0.00022727$ or 0.034 minutes to cross it at 50 mph.

Step 16 Create the travel cost surface

From the Analysis menu, open the Map Calculator. Enter the expression that will convert mph into minutes per foot for every cell in the grid. It is:

```
(1.AsGrid.Float / [Reclass of Speed] * 0.0113636.AsGrid)
```

This may seem like a lot of trouble just to divide the speed limits into 1.0 and multiply

by a constant, but the arithmetic operations have to take place over the entire study area, composed of 198,120 cells ($390 * 508 = 198,120$). You get a lot of computation for your expression.

Click Evaluate and close the Map Calculator.

Step 17 Examine the cost theme

Turn off the Reclass of Speed theme. Make the new theme active, rename it to **Minutes per Foot**, and turn it on.

[VIEW RESULT](#)

In the TOC for the new theme are travel time costs expressed in minutes per foot. It doesn't take much of a minute to travel a foot, so the numbers are very small. They are so small, in fact, that three decimal places is not enough to represent them precisely enough. The solution to increasing the precision is to edit the legend.

Step 18 Change the displayed precision of the new theme

In the TOC, double-click on Minutes per Foot to open the Legend Editor. Click the Classify button and change the "Round values at" field to d.dddddd, so that six significant digits will be shown in the legend.

Click OK, then Apply. Close the Legend Editor.

[VIEW RESULT](#)

Step 19 Create the accumulated travel cost grid

You now have the ingredients necessary to run the CostDistance request. The source grid is called Ramps; the cost surface is Minutes per Foot. Before you enter the actual request, recall the general syntax for CostDistance:

```
aSourceGrid.CostDistance  
(aCostGrid,directionFN,AllocationFN,MaxCost)
```

Now enter the following expression into the Map Calculator:

```
([Ramps].CostDistance([Minutes per Foot],Nil,Nil,Nil))
```

Click Evaluate.

The evaluation may take a few moments. Why does it take so long? Realize that almost one-fifth of a million cells must have values calculated for them.

You will know that CostDistance is finished when the new theme (Map Calculation 1) is added to the view.

When the calculations are completed, close the Map Calculator.

Step 20 Examine the drive time theme

Turn on the new Map Calculation, make it active, and rename it to **Drive Time**. Move the Roads theme above it in the TOC. Turn all other themes off.

[VIEW RESULT](#)

Look carefully at the Drive Time theme values. While many areas are within a 10-minute drive of an offramp, others are more than an hour away. And of course, drive time is only one factor to consider in siting the park. Probably the areas closest to the offramps are the most developed and therefore least suitable for a park.

Step 21 Load an improved drive time legend

You can get a better idea of travel times by redoing the legend of Drive Time. The current legend simply breaks the drive times up into equal intervals, resulting in such unwieldy ranges as 9.065 minutes to 18.131 minutes. A better legend has been created for you.

In the TOC, double-click the Drive Time theme to open the Legend Editor. Click Load and navigate to the **proxalsallesson01\spatial** directory. Select the file named **time2off.avl**.

Click OK to load the entire legend (choose All in the next dialog), then click Apply in the Legend Editor. Close the Legend Editor.

[VIEW RESULT](#)

The new legend uses more appropriate divisions of driving time. It is now apparent that the mathematical massaging you did earlier (calculating minutes per foot) to make the ultimate results show up as minutes was worthwhile.

Step 22 Save the project

Save the project with a new name: **sitepark_yi.apr**, where "yi" represents your initials. Close the project window. Exit ArcView.

Calculating costs (or cost-weighted distance) has applications in many fields. If you were siting a business that needed to be accessible, you could use the ArcView Spatial Analyst CostDistance tool to measure distance to offramps, as you just did.

If you were doing animal migration studies, you could use it to measure migration routes. You could also use cost distance calculations to determine sites for new roads, pipe lines, or transmission lines. In fact, this approach has been used in Alaska as part of a project to determine the impact of building pipelines.

Before closing this discussion on CostDistance, it is important to mention another ArcView tool designed specifically for moving across distances to points connected by linear features, such as roads or railroads. The ArcView Network Analyst allows you to consider additional factors, including one-way streets and costs of turning at intersections. Network Analyst is very powerful and, if your problem or application is specific to street or road networks, you should become familiar with this extension

Summary


In this lesson, you learned how to use several Spatial Analyst tools. The Avenue request EucDistance builds a grid that stores the distances from each cell to the nearest source cell. Distances are calculated from cell center to source cell center and EucDistance calculates the straight-line distance between these two points. You can exclude cells (assign them No Data values) based on a maximum distance away from a source cell.

The cost of moving from one place to another depends not only on distance, but is influenced by other factors. If these factors can be represented as the costs of crossing individual grid cells, you can use the Avenue request CostDistance, which assigns an accumulated cost to each cell, the cost of moving from that cell to the source cell that can be reached least expensively.

EucDistance and CostDistance may optionally generate grids that show the directions one would need to travel to arrive at the closest source cell. EucDistance computes compass directions. CostDistance computes the directions along a least-cost path, cell by cell. Each cell contains a code that points to one of its eight neighbors as the next step in getting to the appropriate source cell. Another Avenue request, CostPath, generates a grid consisting of the least-cost path from cells in a destination grid to those in a source grid using an accumulated cost grid.

Both EucDistance and CostDistance can also generate a grid showing the cluster of cells around source cells. Those cells are allocated to the nearest source cells (or "cheapest to get to" source cells) and their cell values are calculated to be the same as the source cells to which they are allocated

This is the **Using ArcView Spatial Analyst Proximity and Hydrologic Tools - Lesson 1 Self test**.

 **Please watch your time—you have 2 hours to complete this test.**

Use the knowledge you have gained in *Using ArcView Spatial Analyst Proximity and Hydrologic Tools* to answer the following questions. You will need to correctly answer 7 of the following questions to pass.

Netscape Users: Do not resize this browser window. This can cause the page to reload and generate new questions.

GOOD LUCK!

-
1. Usually the first consideration in determining the cost of going from "a" to "b" is:
 - ☐ Impedance
 - ☐ Weight grid
 - ☐ Friction
 - ☐ Distance
 2. When you use the Identify tool, the value displayed in the coordinate window is the cell center.
 - ☐ True

☐ False

3. Using the EucDistance request, there could be an instance when a path would come into a cell directly from the west and exit directly to the south.

☐ True

☐ False

4. Using the CostDistance request, there could be an instance when a path would come into a cell directly from the west and exit directly to the south.

☐ True

☐ False

5. It costs twice as much to cross a cell diagonally (e.g., from northeast to southwest) as it does to cross it vertically (e.g., north to south).

☐ True

☐ False

6. A difference between the EucDistance and CostDistance requests is:

☐ The number of arguments

☐ The way the path is calculated

☐ CostDistance allows the use of a weight grid

☐ All of the above

7. EucDistance calculates the distance from all cells to:

☐ The closest source cell, unless there is a weight grid

☐ The closest source cell

☐ The source cell farthest away

☐ All cells

8. The direction grid for CostDistance contains the compass directions of the least expensive path to a source cell.

☐ True

☐ False

9. The direction grid for EucDistance contains the compass directions of the least expensive path to a source cell.

☐ True

☐ False

10. If the distance to a closest neighboring cell is 2, the distance to a diagonal neighbor cell is:

- ☐ 1.414
- ☐ 2
- ☐ 2.828
- ☐ 4

Calculate My Grade