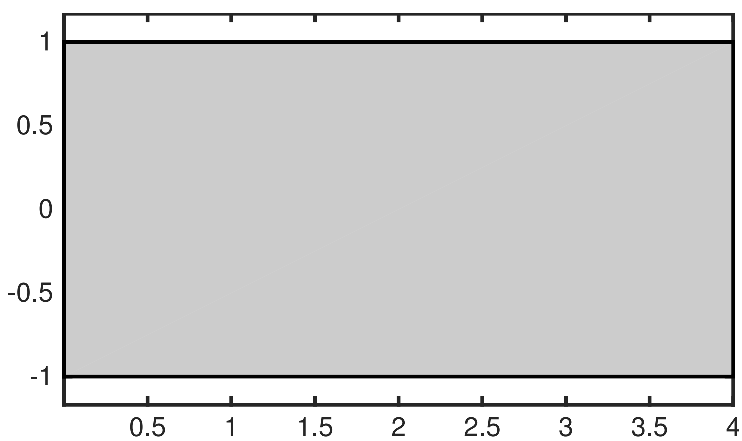
Depths of despair

Numerical integration of a function of one variable is called quadrature. The analogous situation of integrating a function of two variables is sometimes called *cubature*. In this project you will explore the extension of the trapezoid formula to this situation, and apply it to find the volume of water in a patch of the ocean.

Consider the problem $\iint_R f(x,y)\,dx\,dy$, where R is the rectangle $[x_0, x_m] \times [y_0, y_n]$. We discretize each variable using the space steps h and k, respectively, so that $x_i = x_0 + ih$ for $i = 0, \ldots, n$, and $y_j = y_0 + jk$ for $j = 0, \ldots, m$. (This requires that $h = (x_m - x_0)/m$ and $k = (y_n - y_0)/n$.) The result is a decomposition of R into rectangles with the nodes at the corners. The strategy is to compute a volume over each little rectangle and sum them over the whole domain. Here's an example of how the discretization process works in MATLAB:

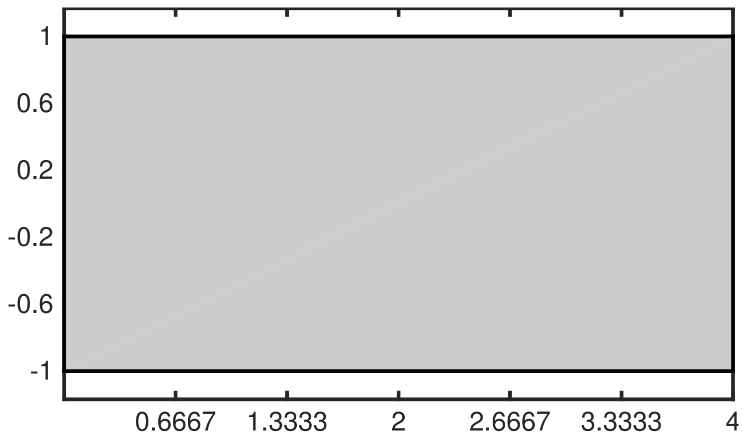
Construct the domain of integration.

```
xLeft = 0;  xRight = 4;
yLeft = -1;  yRight = 1;
fill([xLeft xRight xRight xLeft],[yLeft yLeft yRight yRight],0.8*[1 1 1])
axis equal, hold on
```



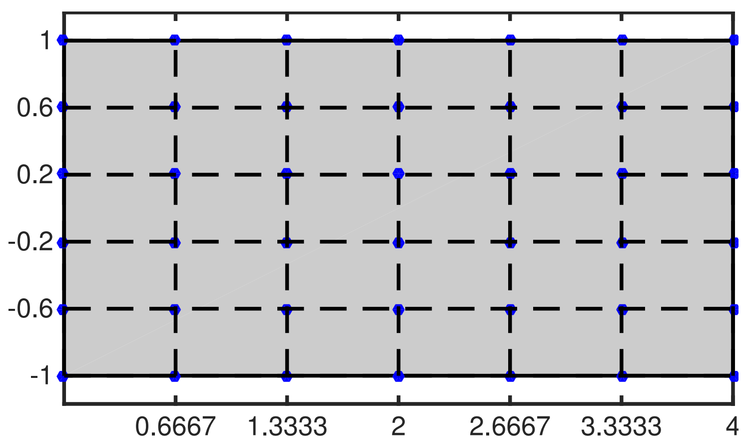
These are the individual variable discretizations.

```
m = 6;  h = (xRight-xLeft)/m;
x = xLeft + h*(0:m)';

n = 5;  eta = (yRight-yLeft)/n;
y = yLeft + eta*(0:n)';

set(gca,'xtick',x,'ytick',y)
```



Now we construct a 2D grid out of them.

```
[X,Y] = ndgrid(x,y);
plot(X,Y,'b.')
plot(X,Y,'k--')
plot(X',Y','k--')
```



Both X and Y are $(m+1)\times(n+1)$ matrices, and satisfy the identity $(x_i, y_j) = (X_{ij}, Y_{ij})$.

```
point = [x(3),y(2)]
pointAgain = [ X(3,2),Y(3,2) ]
```

```
point =
    1.3333   -0.6000
pointAgain =
    1.3333   -0.6000
```

At the corners of each rectangle we have values of z according to $Z_{ij} = f(x_i, y_j)$. We can model the surface $z = f(x, y)$ over one of these rectangles using the function $L_{ij}(x, y) = a + b(x - x_i) + c(y - y_j) + d(x - x_i)(y - y_j)$, where a, b, c, d are determined by interpolation conditions at the four corners of the rectangle. (These coefficients depend on i and j, but that is left out to make the notation clearer.) We get a solid defined over the rectangle looking like this:

Here is a single rectangle.

```
h = 0.1;   x = [0 h]';
eta = 0.2;   y = [-1 -1+eta]';
[X,Y] = ndgrid(x,y)
```

```
X =
         0         0
    0.1000    0.1000
Y =
   -1.0000   -0.8000
   -1.0000   -0.8000
```

These are made-up values at the corners of the rectangle.

```
Z = [ 1.1, 0.9; 0.75, 2 ];
```

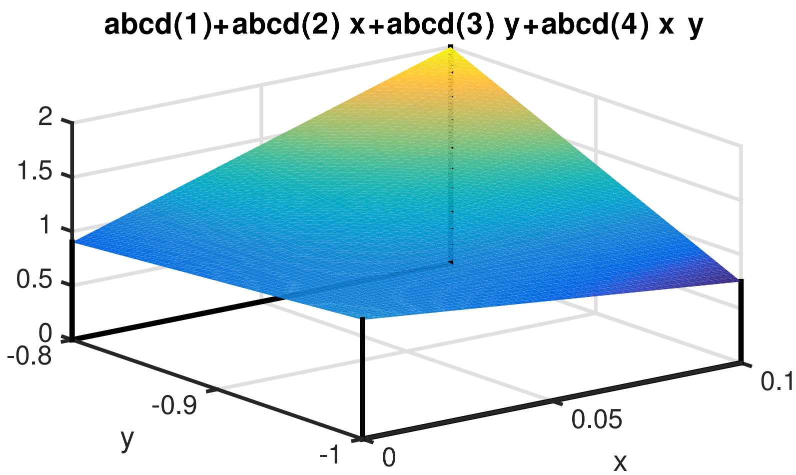
Here are coefficients of the interpolant.

```
A = [ ones(4,1), X(:), Y(:), X(:).*Y(:) ];
abcd = A \ Z(:)
```

```
abcd =
    0.1000
   69.0000
   -1.0000
   72.5000
```

And here is the resulting surface and the outlines of the solid. (Don't worry about the fancy plotting commands here.)

```
ezsurf(@(x,y) abcd(1) + abcd(2)*x + abcd(3)*y + abcd(4)*x.*y, [x;y] )
hold on, shading interp
plot3(X,Y,0*Z,'k','linew',2)
plot3([1;1]*X(:)',[1;1]*Y(:)',[zeros(1,4);Z(:)'],'k','linew',2)
axis tight
```



Objectives

Mathematical content:

1. Using a symbolic math package if you want, derive exact expressions for the interpolant L_{ij} over the rectangle whose lower left corner is (x_i, y_j), in terms of h, k, and four values Z_{ij}, $Z_{i+1,j}$, $Z_{i,j+1}$, $Z_{i+1,j+1}$. (Show the steps involved, whether you use a computer or not.)

2. Integrate the interpolant over the rectangle to get another expression in terms of the same quantities. This is V_{ij}, the volume used to approximate the integral of f over the rectangle.

3. Show that if the V_{ij} are added up over all i and j, the result is identical to applying the trapezoid formula in each dimension sequentially over the domain (i.e., as an iterated integral).

Computational content:

1. Download and install the GeoMapApp at `www.geomappapp.org`.

2. Start the app and agree to the license. You should get a window with a projected map of the world, in latitude-longitude corrdinates as measured in degrees of arc, and some GUI widgets. Use the magnifying glass tool to select a small rectangle over the ocean. You should try to get it to include about 1-2 degrees of latitiude (the y axis).

3. Click on the icon that looks like a grid. After a few seconds another window should open with a histogram in it. Make sure the dropdown menu at the top of this window says "GMRT Grid" with some version number. Click on the floppy disk icon, select "Grid: .XYZ (ascii format)", and save the file to your computer. At this point you can close GeoMapApp.

4. You can import this data into MATLAB by double-clicking on it in the "Current Folder" tab of the MATLAB Desktop, and using the resulting dialog. What you will get is three columns of x, y, and z data on a regular rectangular grid, where x and y are reported in degrees of arc, and z will be negative to represent depths below sea level.

5. **After converting all the data into km units**, use it and the formula you derived above to estimate the amount of water in your patch of ocean.

Submission

Submit a function

```
function volume = ocean(x,y,z)
```

that will compute an approximation to the volume using three vectors of depth data as described in the previous section.
Prepare a report that includes the following.

1. The required mathematical content.
2. Details needed to acquire the same raw data from the app and compute your volume result.
3. Some form of quantitative check on whether your answer is reasonable, in the sense of being at least the right order of magnitude.
4. A brief statement of the contributions of each team member to the project. **Be as specific as possible.**

Your report should read like a document understandable to anyone else in the class. All of the mathematical expressions in your document should be typeset properly as mathematical notation. You have several options: Word, OpenOffice, LaTeX, `texmacs.org`, Mathematica. Submit your report as a single PDF document, along with your M-file.