

Lecture 25

Double Integrals for Non-rectangles

In the previous lecture we considered only integrals over rectangular regions. In practice, regions of interest are rarely rectangles and so in this lecture we consider two strategies for evaluating integrals over other regions.

Redefining the function

One strategy is to redefine the function so that it is zero outside the region of interest, then integrate over a rectangle that includes the region.

For example, suppose we need to approximate the value of

$$I = \iint_T \sin^3(xy) dx dy$$

where T is the triangle with corners at $(0,0)$, $(1,0)$ and $(0,2)$. Then we could let R be the rectangle $[0,1] \times [0,2]$ which contains the triangle T . Notice that the hypotenuse of the triangle has the equation $2x + y = 2$. Then make $f(x) = \sin^3(xy)$ if $2x + y \leq 2$ and $f(x) = 0$ if $2x + y > 2$. In MATLAB we can make this function with the command

```
f = @(x,y) sin(x.*y).^3.*(2*x + y <= 2)
```

In this command `<=` is a *logical* command. The term in parentheses is then a *logical statement* and is given the value 1 if the statement is true and 0 if it is false. We can then integrate the modified `f` on $[0,1] \times [0,2]$ using the command

```
I = integral2(f,0,1,0,2)
```

As another example, suppose we need to integrate $10000 + x^2 \exp(xy)$ inside the circle of radius 2 centered at $(1,2)$. The equation for this circle is $(x - 1)^2 + (y - 2)^2 = 4$. Note that the inside of the circle is $(x - 1)^2 + (y - 2)^2 \leq 4$ and that the circle is contained in the rectangle $[-1,3] \times [0,4]$. Thus we can create the right function, plot it, and integrate it by

```
f = @(x,y) (10+(x-1).*(y-2)).*((x-1).^2 + (y-2).^2 <= 4)
[X Y] = meshgrid(-4:0.01:4, -3:0.01:5);
Z = f(X,Y);
mesh(X,Y,Z)
I = integral2(f,-1,3,0,4)
```

Integration Based on Triangles

The second approach to integrating over non-rectangular regions, is based on subdividing the region into triangles. Such a subdivision is called a *triangulation*. On regions where the boundary consists of line segments, this can be done exactly. Even on regions where the boundary contains curves, this can be done approximately. This is a very important idea for several reasons, the most important of which is that the finite elements method is based on it. Another reason this is important is that often the values of f are not given by a formula, but from data. For example, suppose you are surveying on a construction site and you want to know how much fill will be needed to bring the level up to the plan. You would proceed by taking elevations at numerous points across the site. However, if the site is irregularly shaped or if there are obstacles on the site, then you cannot make these measurements on an exact rectangular grid. In this case, you can use triangles by connecting your points with triangles. Many software packages will even choose the triangles for you (MATLAB will do it using the command `delaunay`).

The basic idea of integrals based on triangles is exactly the same as that for rectangles; the integral is approximated by a sum where each term is a value times an area

$$I \approx \sum_{i=1}^n f(x_i^*) A_i,$$

where n is the number of triangles, A_i is the area of the triangle and x^* a point in the triangle. However, rather than considering the value of f at just one point people often consider an average of values at several points. The most convenient of these is of course the corner points. We can represent this sum by

$$T_n = \sum_{i=1}^n \bar{f}_i A_i,$$

where \bar{f} is the average of f at the corners.

If the triangle has vertices (x_1, y_1) , (x_2, y_2) and (x_3, y_3) , the formula for area is

$$A = \frac{1}{2} \left| \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|. \quad (25.1)$$

The function `mythreecorners` below implements this method.

```
function I = mythreecorners(f,V,T)
% Integrates a function based on a triangulation, using three corners
% Inputs: f -- the function to integrate
%         V -- the vertices.
%         Each row has the x and y coordinates of a vertex
%         T -- the triangulation.
%         Each row gives the indices of three corners
% Output: the approximate integral

x = V(:,1); % extract x and y coordinates of all nodes
y = V(:,2);
I=0;
p = size(T,1);
for i = 1:p
```

```

x1 = x(T(i,1)); % find coordinates and area
x2 = x(T(i,2));
x3 = x(T(i,3));
y1 = y(T(i,1));
y2 = y(T(i,2));
y3 = y(T(i,3));
A = .5*abs(det([x1, x2, x3; y1, y2, y3; 1, 1, 1]));
z1 = f(x1,y1); % find values and average
z2 = f(x2,y2);
z3 = f(x3,y3);
zavg = (z1 + z2 + z3)/3;
I = I + zavg*A; % accumulate integral
end
end

```

Another idea would be to use the center point (centroid) of each triangle. If the triangle has vertices (x_1, y_1) , (x_2, y_2) and (x_3, y_3) , then the centroid is given by the simple formulas

$$\bar{x} = \frac{x_1 + x_2 + x_3}{3} \quad \text{and} \quad \bar{y} = \frac{y_1 + y_2 + y_3}{3}. \quad (25.2)$$

Exercises

- 25.1 Download the programs `mywedge.m` and `mywasher.m` from the web site. Plot $f(x, y) = \frac{x+y}{x^2+y^2}$ on the region produced by `mywasher.m` and $f(x, y) = \sin(x) + \sqrt{y}$ on the region produced by `mywedge.m`.
- 25.2 Modify the program `mythreecorners.m` to a new program `mycenters.m` that does the centerpoint method for triangles. Run the program on the region produced by `mywasher.m` with the function $f(x, y) = \frac{x+y}{x^2+y^2}$ and on the region produced by `mywedge.m` with the function $f(x, y) = \sin(x) + \sqrt{y}$.