Lecture 24

Double Integrals for Rectangles

The center point method

Suppose that we need to find the integral of a function, $f(x, y)$, on a rectangle

$$R = \{(x, y) : a \leq x \leq b, c \leq y \leq d\}.$$

In calculus you learned to do this by an iterated integral

$$I = \iint_R f \, dA = \int_a^b \int_c^d f(x, y) \, dydx = \int_c^d \int_a^b f(x, y) \, dxdy.$$

You also should have learned that the integral is the limit of the Riemann sums of the function as the size of the subrectangles goes to zero. This means that the Riemann sums are approximations of the integral, which is precisely what we need for numerical methods.

For a rectangle R, we begin by subdividing into smaller subrectangles $\{R_{ij}\}$, in a systematic way. We will divide $[a, b]$ into m subintervals and $[c, d]$ into n subintervals. Then R_{ij} will be the "intersection" of the i-th subinterval in $[a, b]$ with the j-th subinterval of $[c, d]$. In this way the entire rectangle is subdivided into mn subrectangles, numbered as in Figure 24.1.

A Riemann sum using this subdivision would have the form

$$S = \sum_{i,j=1,1}^{m,n} f(x_{ij}^*) A_{ij} = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} f(x_{ij}^*) A_{ij} \right),$$

where $A_{ij} = \Delta x_i \Delta y_j$ is the area of R_{ij}, and x_{ij}^* is a point in R_{ij}. The theory of integrals tells us that if f is continuous, then this sum will converge to the same number, no matter how we choose x_{ij}^*. For instance, we could choose x_{ij}^* to be the point in the lower left corner of R_{ij} and the sum would still converge as the size of the subrectangles goes to zero. However, in practice we wish to choose x_{ij}^* in such a way to make S as accurate as possible even when the subrectangles are not very small. The obvious choice for the best point in R_{ij} would be the center point. The center point is most likely of all points to have a value of f close to the average value of f. If we denote the center points by c_{ij}, then the sum becomes

$$C_{mn} = \sum_{i,j=1,1}^{m,n} f(c_{ij}) A_{ij}.$$

Here

$$c_{ij} = \left(\frac{x_{i-1} + x_i}{2}, \frac{y_{i-1} + y_i}{2} \right).$$

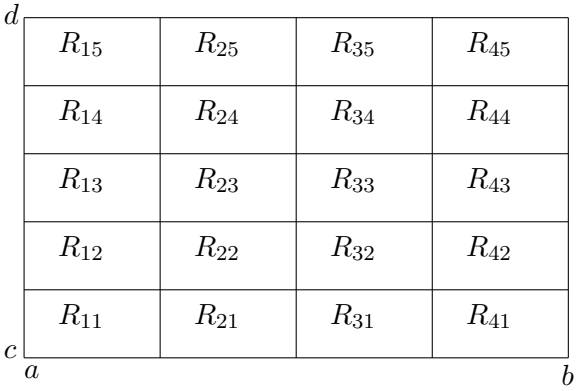

Figure 24.1: Subdivision of the rectangle $R = [a,b] \times [c,d]$ into subrectangles R_{ij}. Note that the arrangement in the x-y plane is completely different from the convention in a matrix.

Note that if the subdivision is evenly spaced then $\Delta x \equiv (b-a)/m$ and $\Delta y \equiv (d-c)/n$, and so in that case

$$C_{mn} = \frac{(b-a)(d-c)}{mn} \sum_{i,j=1,1}^{m,n} f(c_{ij}).$$

The four corners method

Another good idea would be to take the value of f not only at one point, but as the average of the values at several points. An obvious choice would be to evaluate f at all four corners of each R_{ij} then average those. If we note that the lower left corner is (x_i, y_j), the upper left is (x_i, y_{j+1}), the lower right is (x_{i+1}, y_i) and the upper right is (x_{i+1}, y_{i+1}), then the corresponding sum will be

$$F_{mn} = \sum_{i,j=1,1}^{m,n} \frac{1}{4} \left(f(x_i, y_j) + f(x_i, y_{j+1}) + f(x_{i+1}, y_i) + f(x_{i+1}, y_{j+1}) \right) A_{ij},$$

which we will call the *four-corners* method. If the subrectangles are evenly spaced, then we can simplify this expression. Notice that $f(x_i, y_j)$ gets counted multiple times depending on where (x_i, y_j) is located. For instance if (x_i, y_j) is in the interior of R then it is the corner of 4 subrectangles. Thus the sum becomes

$$F_{mn} = \frac{A}{4} \left(\sum_{\text{corners}} f(x_i, y_j) + 2 \sum_{\text{edges}} f(x_i, y_j) + 4 \sum_{\text{interior}} f(x_i, y_j) \right),$$

where $A = \Delta x \, \Delta y$ is the area of the subrectangles. We can think of this as a weighted average of the values of f at the grid points (x_i, y_j). The weights used are represented in the matrix

$$W = \begin{pmatrix} 1 & 2 & 2 & 2 & \cdots & 2 & 2 & 1 \\ 2 & 4 & 4 & 4 & \cdots & 4 & 4 & 2 \\ 2 & 4 & 4 & 4 & \cdots & 4 & 4 & 2 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 2 & 4 & 4 & 4 & \cdots & 4 & 4 & 2 \\ 1 & 2 & 2 & 2 & \cdots & 2 & 2 & 1 \end{pmatrix}. \tag{24.1}$$

We could implement the four-corner method by forming a matrix (f_{ij}) of f values at the grid points, then doing entry-wise multiplication of the matrix with the weight matrix. Then the integral would be obtained by summing all the entries of the resulting matrix and multiplying that by $A/4$. The formula would be

$$F_{mn} = \frac{(b-a)(d-c)}{4mn} \sum_{i,j=1,1}^{m+1,n+1} W_{ij} f(x_i, y_j).$$

Notice that the four-corners method coincides with applying the trapezoid rule in each direction. Thus it is in fact a *double trapezoid* rule.

The double Simpson method

The next improvement one might make would be to take an average of the center point sum C_{mn} and the four corners sum F_{mn}. However, a more standard way to obtain a more accurate method is the Simpson double integral. It is obtained by applying Simpson's rule for single integrals to the iterated double integral. The resulting method requires that both m and n be even numbers and the grid be evenly spaced. If this is the case we sum up the values $f(x_i, y_j)$ with weights represented in the matrix

$$W = \begin{pmatrix} 1 & 4 & 2 & 4 & \cdots & 2 & 4 & 1 \\ 4 & 16 & 8 & 16 & \cdots & 8 & 16 & 4 \\ 2 & 8 & 4 & 8 & \cdots & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & \cdots & 8 & 16 & 4 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 2 & 8 & 4 & 8 & \cdots & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & \cdots & 8 & 16 & 4 \\ 1 & 4 & 2 & 4 & \cdots & 2 & 4 & 1 \end{pmatrix}. \tag{24.2}$$

The sum of the weighted values is multiplied by $A/9$ and the formula is

$$S_{mn} = \frac{(b-a)(d-c)}{9mn} \sum_{i,j=1,1}^{m+1,n+1} W_{ij} f(x_i, y_j).$$

MATLAB has a built in command for double integrals on rectangles: `integral2(f,a,b,c,d)`. Here is an example:

```
≫ f = @(x,y) sin(x.*y)./sqrt(x+y)
≫ I = integral2(f,0.5,1,0.5,2)
```

Note that here, as usual, operations are component-wise.

Below is a MATLAB function which will produce the matrix of weights needed for Simpson's rule for double integrals. It uses the function `mysimpweights` from Lecture 22.

```
function W = mydblsimpweights(m,n)
    % Return the matrix of weights for Simpson's rule for double integrals.
    % Inputs: m -- number of intervals in the row direction.
    %              must be even.
    %         n -- number of intervals in the column direction.
    %              must be even.
    % Output: W -- a (m+1)x(n+1) matrix of the weights
    if rem(m,2)~=0 || rem(n,2)~=0
        error('m and n must be even for Simpsons rule')
    end
    % get 1-dimensional weights
    u = mysimpweights(m);
    v = mysimpweights(n);
    W = u*v';
end
```

Exercises

24.1 Download the program `mylowerleft.m`. Modify it to make a new program `mycenter` that does the center point method. Implement the change by changing the "meshgrid" to put the grid points at the centers of the subrectangles. Look at the mesh plot produced to make sure the program is putting the grid where you want it. Use both programs to approximate the integral

$$\int_0^2 \int_1^5 \sqrt{xy}\, dy\, dx,$$

using $(m, n) = (10, 18)$. Evaluate this integral exactly (by hand) and compare the accuracy of the two programs.

24.2 Write a well-commented MATLAB **function** program `mydblsimp` that computes the Simpson's rule approximation. Let it call the program `mydblsimpweights` (above) to make the weight matrix, w (24.2). Check the program on the integral in the previous problem using $(m, n) = (10, 18)$ and $(m, n) = (100, 100)$.

24.3 Using `mysimpweights` and `mydblsimpweights` as models make well-commented MATLAB **function** programs `mytrapweights` and `mydbltrapweights` that will produce the weights for the trapezoid rule and the weight matrix for the four corners (double trapezoid) method (24.1). Use it to approximate the integral in the previous problem usings $(m, n) = (5, 6)$ and $(m, n) = (100, 100)$. Turn in the programs, the weights for a 5×6 grid, and the results of your tests on the integral.