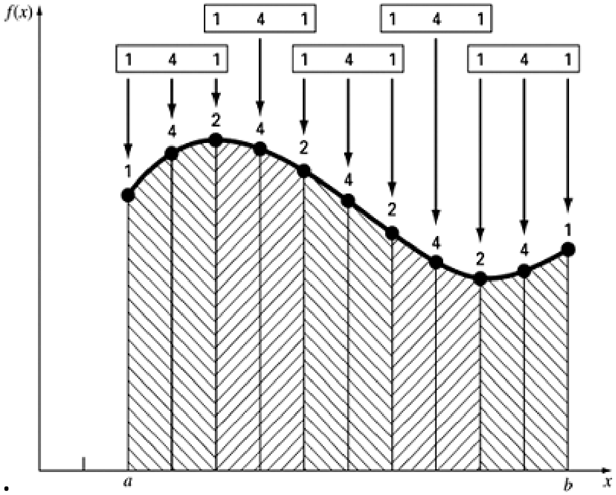
Homework 23

Numerical Integration Formulas

- Simpson's 1/3 rule can be used on a set of subintervals in the same way the trapezoidal rule was, except there must be **an odd number of points**, that is, we must have **an even number of intervals**. In the following formulas, the subscript ranges from 0 to n, so the number of data points is n+1.



$$I = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] + \frac{h}{3}[f(x_2) + 4f(x_3) + f(x_4)] + \ldots$$

$$+ \frac{h}{3}[f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

$$I = \frac{h}{3}\left[f(x_0) + 4\sum_{\substack{i=1 \\ i,odd}}^{n-1} f(x_i) + 2\sum_{\substack{j=2 \\ j,even}}^{n-2} f(x_j) + f(x_n) \right]$$

In the left formulas, i and j are the index of data points (from 0 to n). n should be an even number. n+1 is the number of data points (odd)

In the slides of Lecture 23, I wrote j=1, it is a mistake. I have corrected it.

In this formula, i=0 is the 1st point, i=1 is the 2nd point. In MATLAB, the indices of elements begins with 1. Therefore, in this summation, i should be even numbers. If n is used as the number of data points in MATLAB (odd number), we should write the summation as 4*sum(2:2:n-1).

For the same reason, j=2 is the 3rd point. In MATLAB, the indices of elements in this summation should be odd numbers. We should write it as 2*sum(3:2:n-2). n is the number of data points.

Numerical Integration Formulas

In Homework 23, n is the number of intervals, n should be an even number. Then, the number of data points is n+1. The main script and functional program are given as following

```
% ME 261 HW23 Solution
clc
clear

f = @(x) (x.^3).*exp(x);

a = input('Enter lower limit of range: ');
b = input('Enter upper limit of range: ');
n = input('Enter number of subdivisions (must be greater than 1): ');

ar = cal_area_sim(a,b,n,f);

fprintf('\n\nThe value of the integral from a = %d to b = %d\n',a,b)
fprintf('using %d equally spaced divisions is: %.5f\n\n',n,ar)
```

```
function ar = calc_area_sim(a,b,n,f)

h = (b-a)/n;
x = linspace(a,b,n+1)';
y = f(x);

ar = (h/3)*(y(1) + 4*sum(y(2:2:n)) + 2*sum(y(3:2:n-1)) + y(n+1));
```

In the last page, we said if n is the number of data points, the first summation is written as 4*sum(2:2:n-1). If n is the number of intervals, the summation should be written as 4*sum(2:2:n)

If n is the number of data points, this summation is written as 2*sum(3:2:n-2). If n is the number of intervals, the summation should be written as 2*sum(3:2:n-1)

n is the number of intervals, so the index of the last point in MATLAB is n+1