



DREXEL UNIVERSITY

# Electrical and Computer Engineering

*College of Engineering*

Drexel University  
Electrical and Computer Engineering Dept.  
ECEC-413

## Numerical Integration with OpenMP

Chris Kasper  
Prof. Naga Kandasamy  
DATE: 05/2/19

## Multi-threaded Design (OpenMP)

To extend the design of the numerical integration (using the trapezoid rule) to a multi-threaded design, the work has to be divided among the number of threads. This was done as such:

```
/* Integral calculation using trapezoid rule and OpenMP */
double
compute_using_omp (float a, float b, int n, float h, int num_threads)
{
    double integral = 0.0;
    int k;

    integral = (f(a) + f(b)) / 2.0;

    // Set up parallel execution
    #pragma omp parallel num_threads(num_threads) shared(integral)
    {
        /* Private variable to compute sum */
        double thisIntegral = 0.0;

        #pragma omp for schedule(static, 1)
        for (k = 1; k <= n-1; k++) {
            thisIntegral += f(a+k*h);
        }

        #pragma omp critical /* Mutual Exclusion */
        integral += thisIntegral;
    } /* End of parallel region */

    integral = integral * h;

    return integral;
}
```

The first thing that is done before splitting the work up, is to calculate the values for the start and end point of our total area. Then, each thread gets allocated a certain number of trapezoids being used to calculate the rest of the integral of the equation. Utilizing OpenMP, we set the computation to be done in parallel, split up given the number of threads, and set our “integral” variable to be shared amongst all threads. However, each thread has a private variable, “thisIntegral”, where they can store their result for the computations it performed. Once the trapezoid computations for a thread are finished, each thread adds on their result to the shared variable “integral”, using mutual exclusion. Finally, once all the threads finish, the “integral” variable is multiplied by the trapezoid base value to give us our final estimation.

## Speedup

**Table 1: Speedup of integration on xunil-05**

Number of Trapezoids	Lower Limit	Upper Limit	Single Thread	2 Threads	4 Threads	8 Threads	16 Threads
100000	0	10	0.0052s	0.0028s	0.0017s	0.0010s	0.0012s
100000	0	100	0.0052s	0.0030s	0.0017s	0.0011s	0.0010s
1000000	0	10	0.0455s	0.0250s	0.0134s	0.0077s	0.0056s
1000000	0	100	0.0455s	0.0250s	0.0133s	0.0070s	0.0045s



