ME 614 Spring 2017-Homework 1

Introduction to Parallel Computing

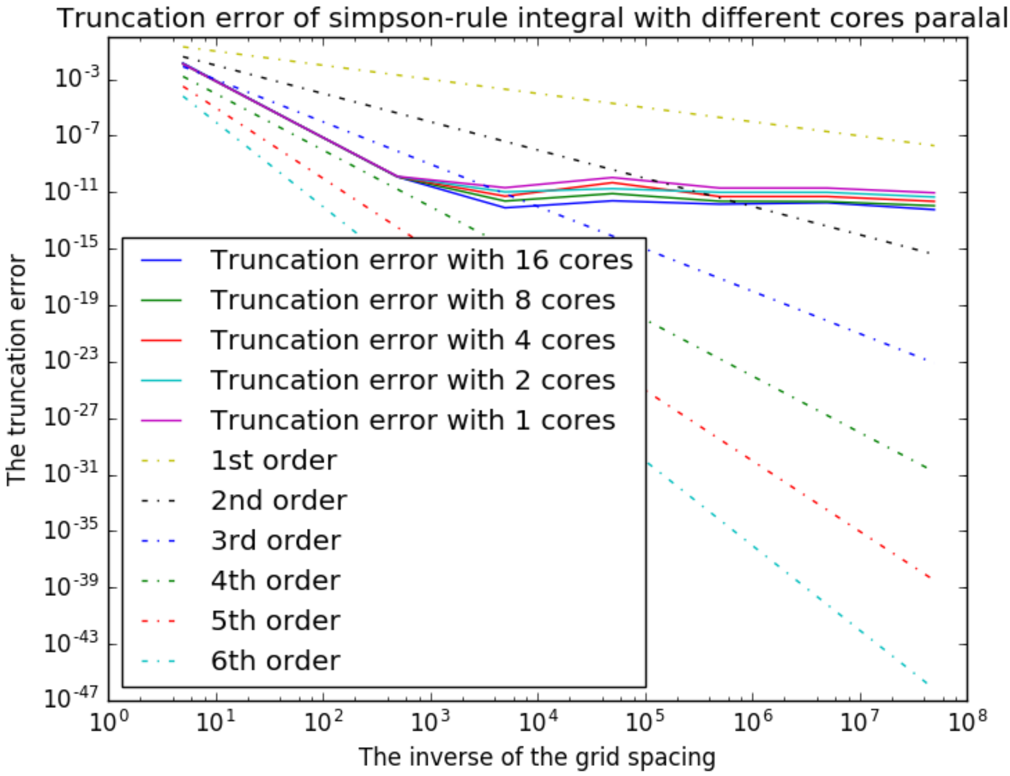
Zitao He

2/22/2017

Problem 1

(a)

Different possesses (by applying different cores) are used to compute the integral through Simpson's quadrature rule. The truncation error against the inverse grid spacing with different possesses is showed below.



The truncation error in this plot is calculated by the difference of analytical integral and numerical integral(absolute value) which comprised of truncation error from numerical method and round-off error from limited computer precision.

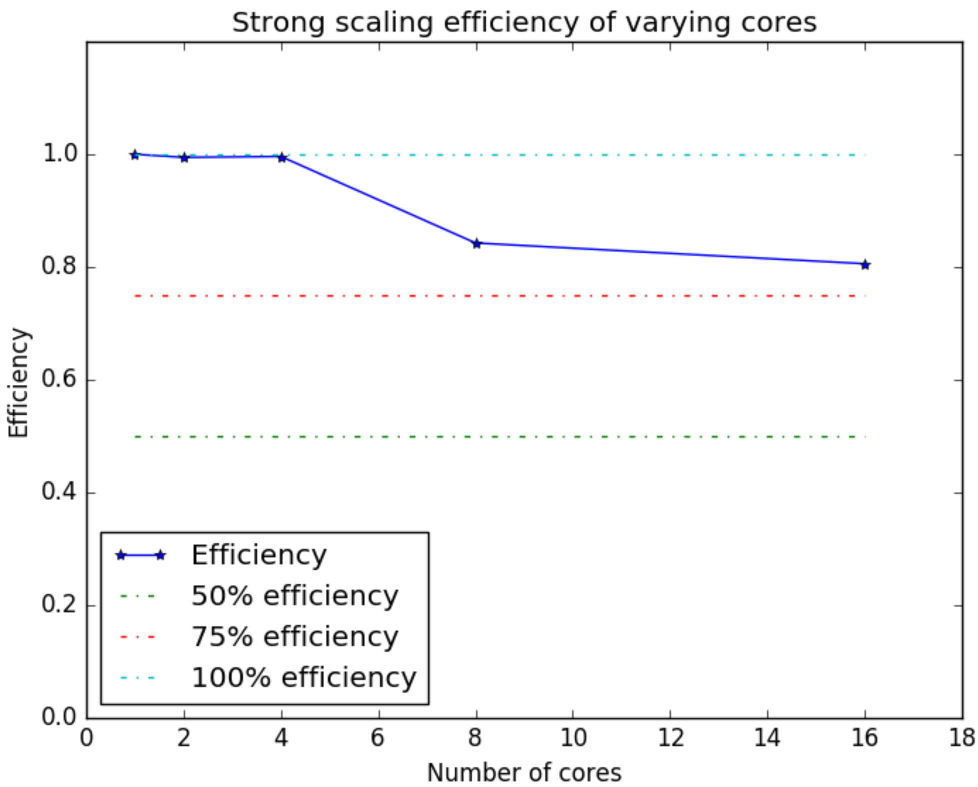
Assuming there is no round-off error, the numerical error of simpson's rule is given by:

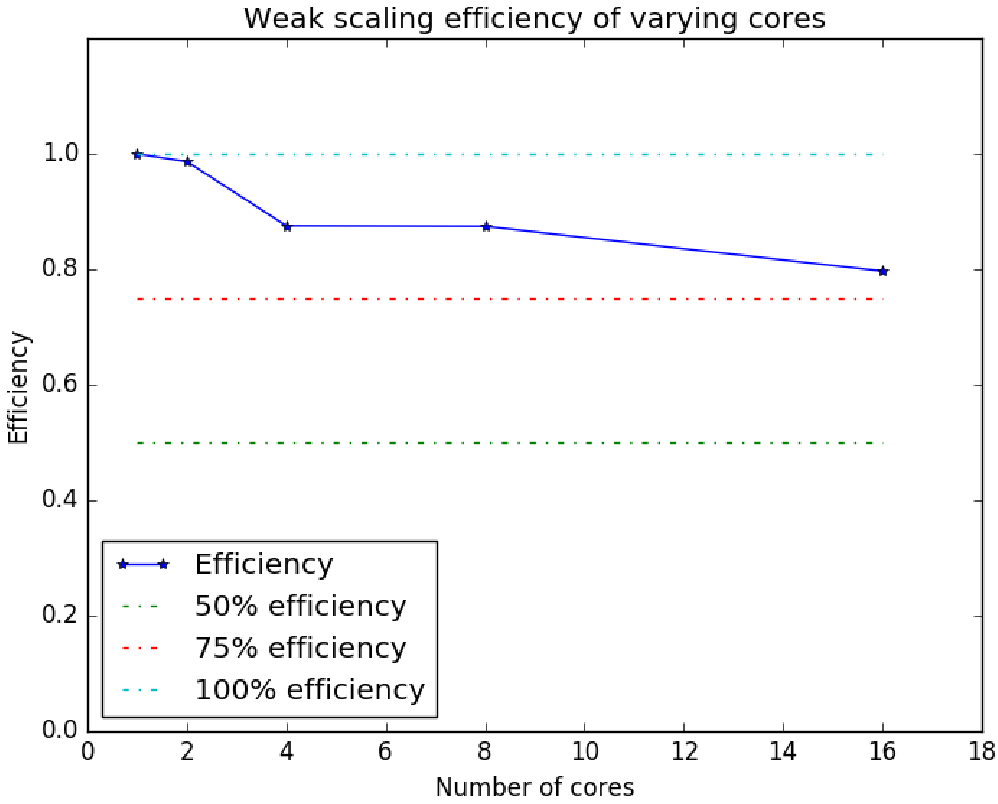
$$\frac{h^4}{180}(b-a)\max_{\xi\in[a,b]}|f^{(4)}(\xi)|,$$

where h is the "step length", given by h = (b - a)/n, and a, b is the startpoint and endpoint of integral respectively, n is the number of nodes. Ideally, simpson's rule will convergence with order of 4 as showed in the formula. So the first half of the truncation error curve is matching the reference line with order of 4.

As we can observe on the figure, truncation error are perfectly overlap at the beginning. This can be explained by: when discretization is relatively rough, the error mainly comes from the "true" truncation error of the numerical method(as shown in the formula). However, when the discretization is finer, which means the nodes of discretization increases, the round-off error become non-trivial and numerical truncation error become smaller(4th order of grid spacing h). The order of convergence is no longer 4. It shows oscillation which is depending on the computer instead of convergence with specific regulation. Someone got the result that all the lines are perfectly overlap but others got results with very trivial overlap after grid is fine enough. My guess is that because each one uses different codes which might have different precision (for example arithmetic operations are different which will bring different precision on each case).

(b) Strong scalling efficieny and Weak scalling efficiency.





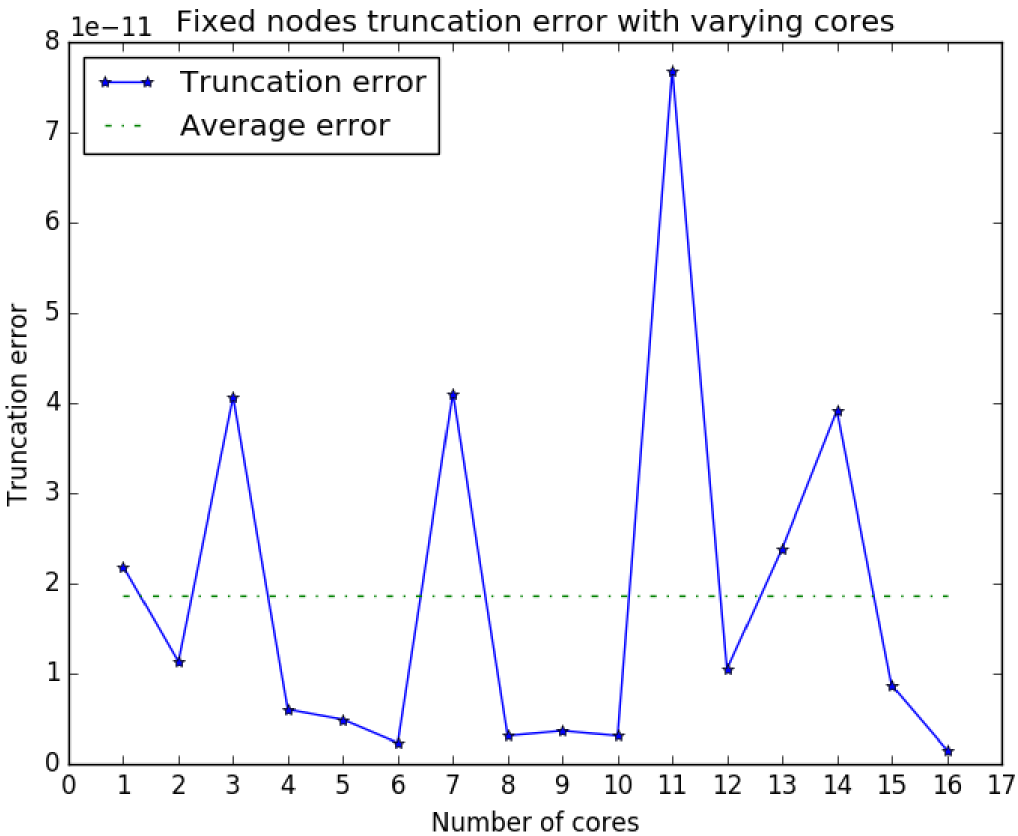
Comments:

The first three points in the strong scaling efficiency plot are almost 100%, this is explained by Prof. Scalo on piazza:

“It means that timing for a single process is off, probably due to the machine slowing down (due to background processes) when running with a single process (it's totally random and can not be controlled). ”

(c)

With fixed number of points, N=10e5+1 and varying the number of processes between 1 and 16 in increment of 1. The absolution truncation error plot is shown below:



Ideally, the truncation error with N = 10e5+1 points with different processes should be the same. However, from my results they are slightly different and some oscillation are recognizable. This might be caused by limited precision from my code and computer and it’s very case-sensitive. Everyone will get the different results regarding this part. The error seems to be distributed randomly.