Exploring Parallelization with a RasPi Cluster

Taylor Powell

2021 CS RasPi Contest

Project Overview

1. Building a Raspberry Pi cluster
   - Physically assembling the cluster
   - Get the cluster networked together and communicating through passwordless-SSH

1. Exploring parallelization
   - Develop a few test programs with parallelization through MPI
   - Examine the quantitative benefits of parallelization

Building the Cluster

Assembly

Assembly

- Install Operating Systems

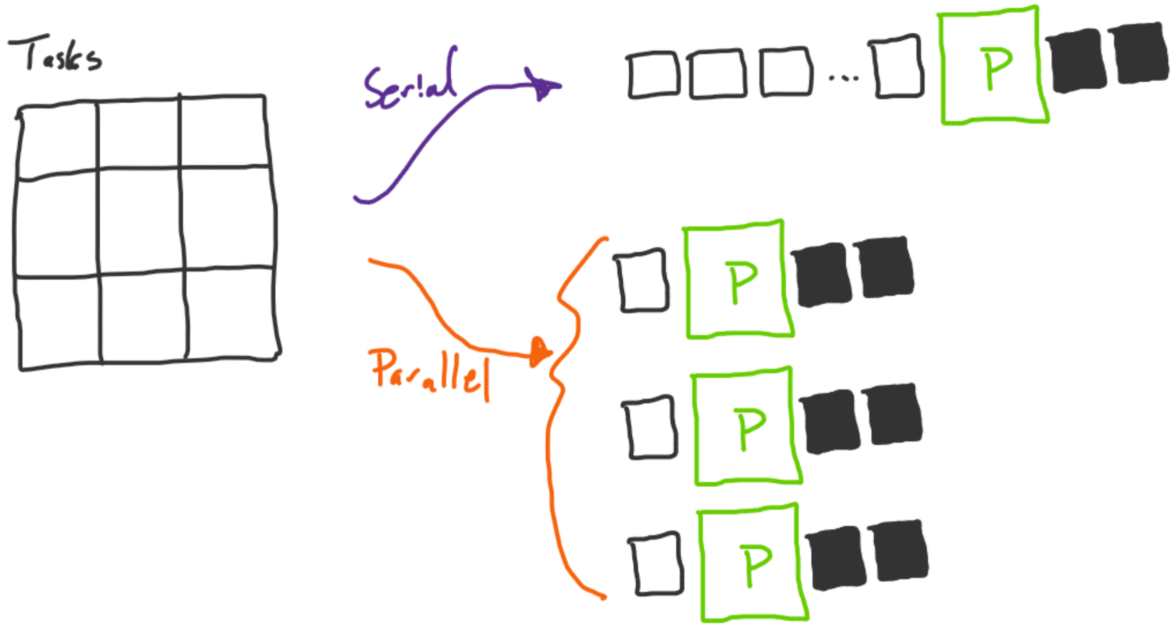
- System Updates
- Cable Management

Networking

- Set static IP addresses
- Generate RSA keys and share between nodes
- Enable SSH and ensure master node can access all slave nodes passwordlessly
- Did the same process to get my Windows PC to communicate with the master node

RasPi Cluster - Conclusions

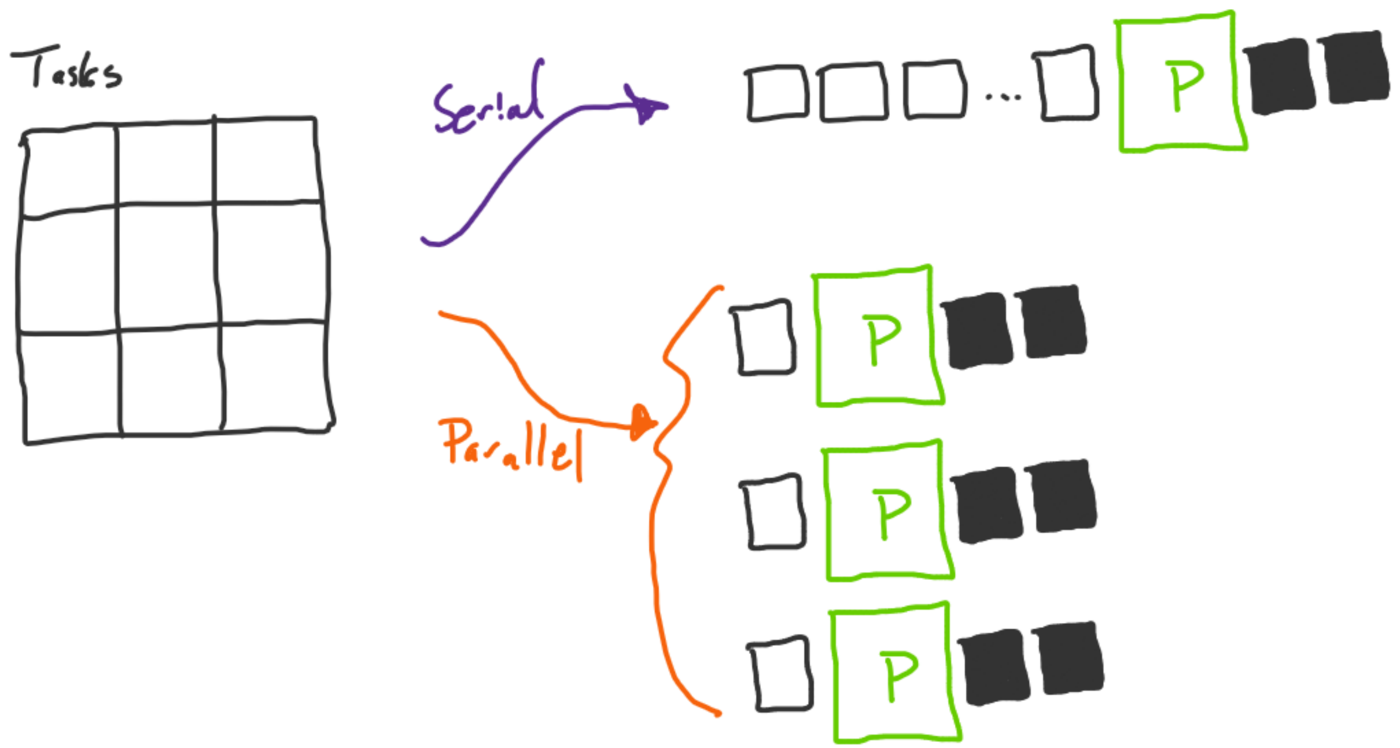
- Raspberry Pis are cheap and accessible tools for practicing developing more complicated computer architecture.
- Steep learning curve - but teaches all the foundational concepts which are essential to larger clusters.
  - Power, heat, memory, communication, data access, etc
- Plenty of opportunity for growth
  - More nodes, sophisticated data-sharing, alternate networking setups, etc



Exploring Parallelization

Parallelization

- Subdivide large task and distribute across a network of processors instead of a single processor.

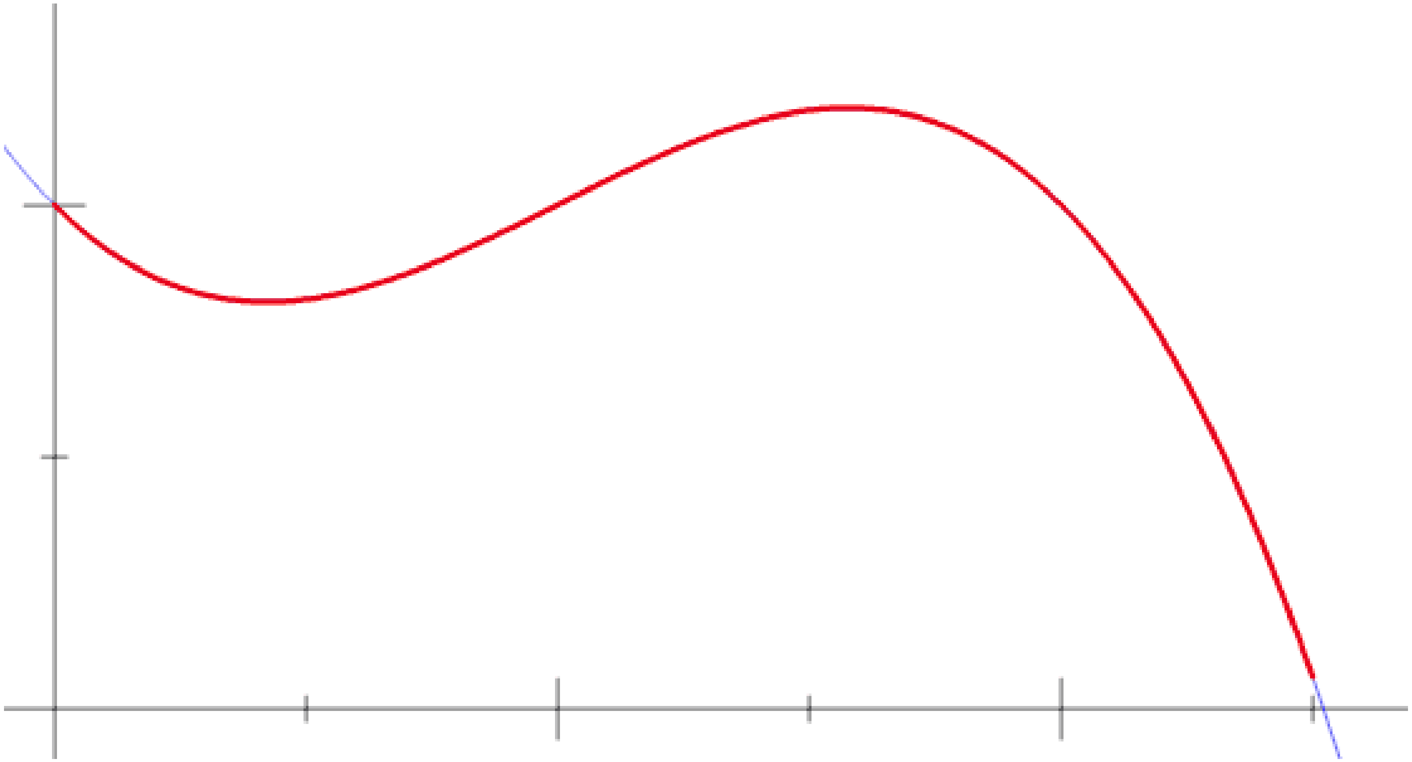


Simple Integration

- The value of an integral for a continuous function over an interval a<x<b can be approximated using the Riemann integral formulation

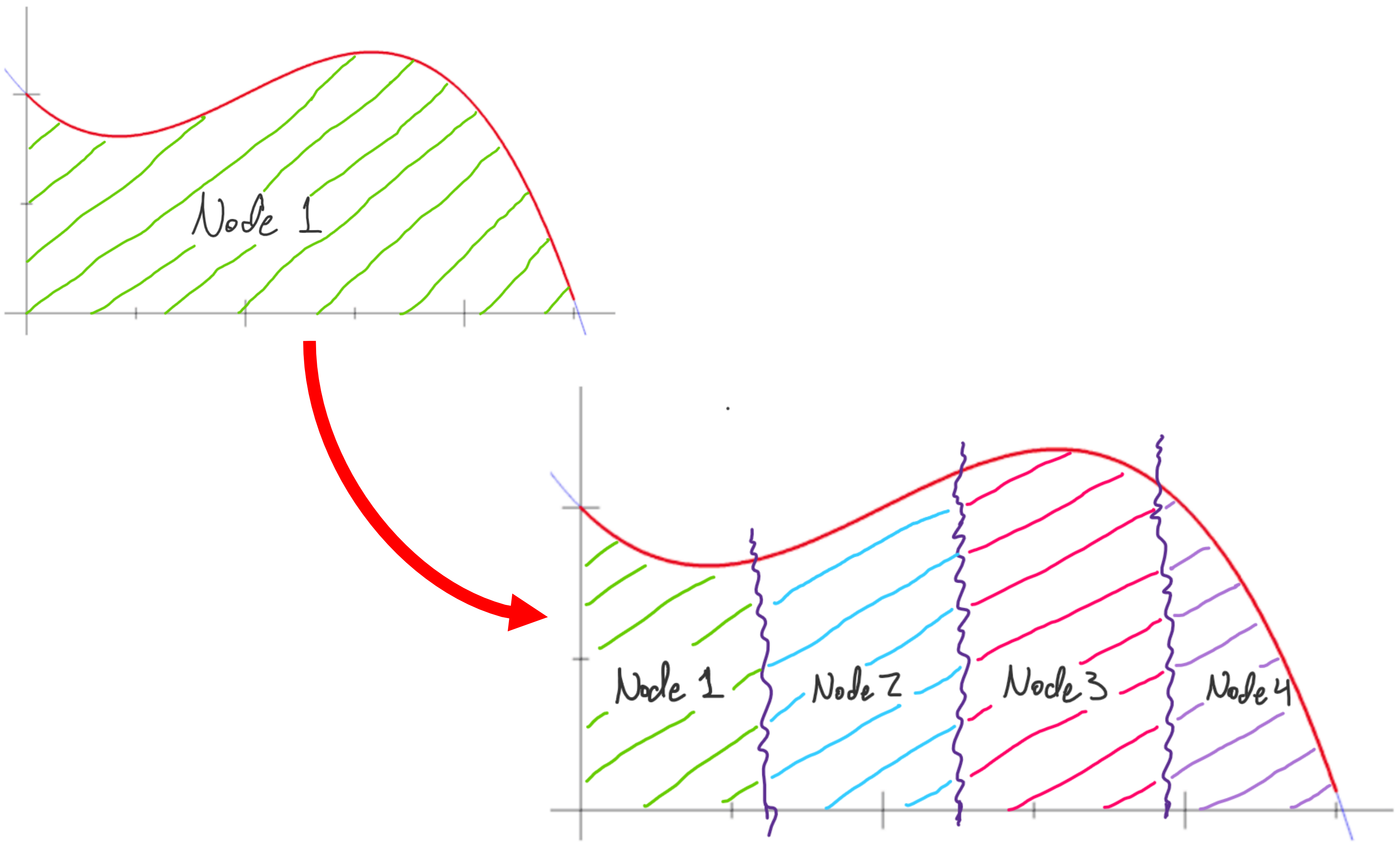
$$\int_a^b f(x)dx = \lim_{N \to \infty} \sum_{i=1}^{N} f(x_i)\Delta x_i$$

Simple Integration



Images by Lucas Vieira - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=19609168

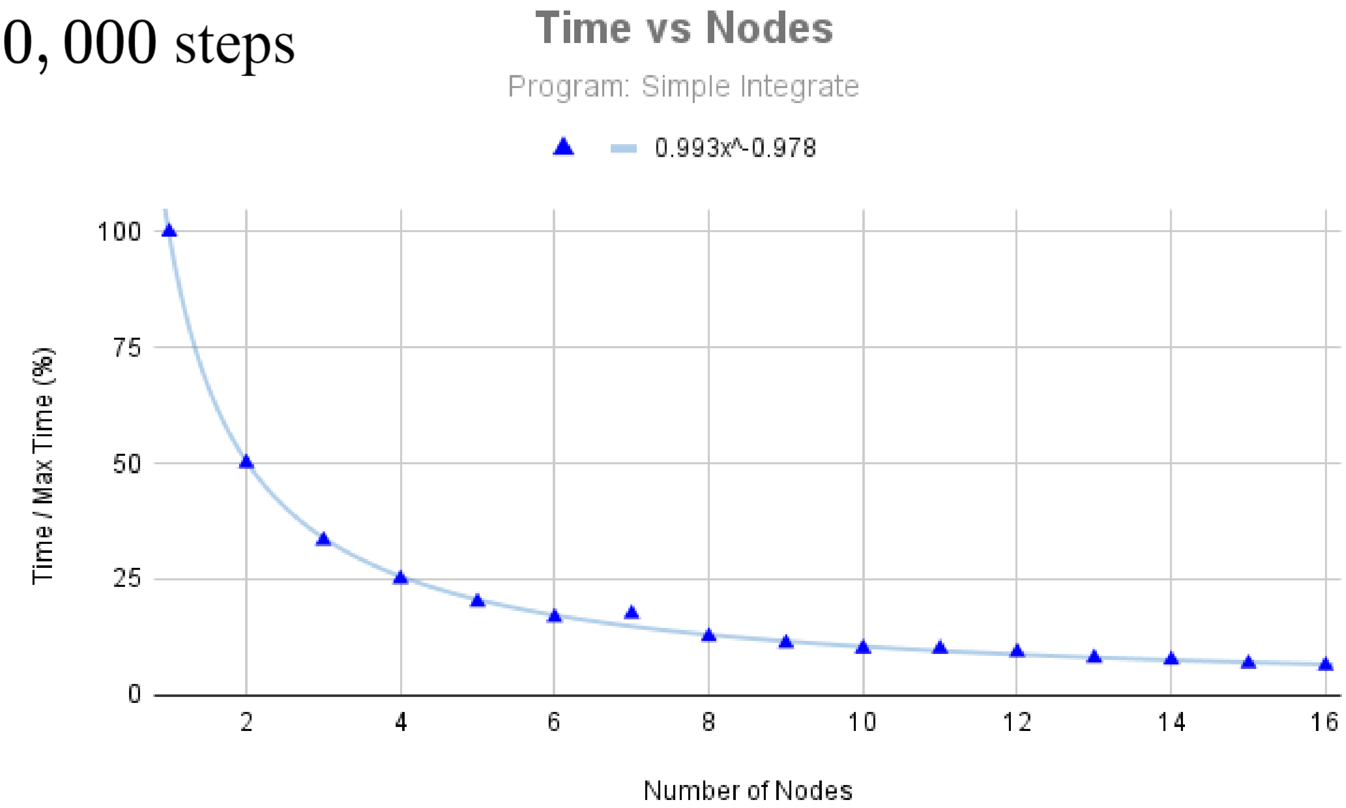
Simple Integration - Parallelized



Simple Integration - Results

$$\int_{-0.1}^{0.1} \ln(x^2 + 0.000000001)dx$$

$$N = 1,600,000 \text{ steps}$$



Adaptive Integration

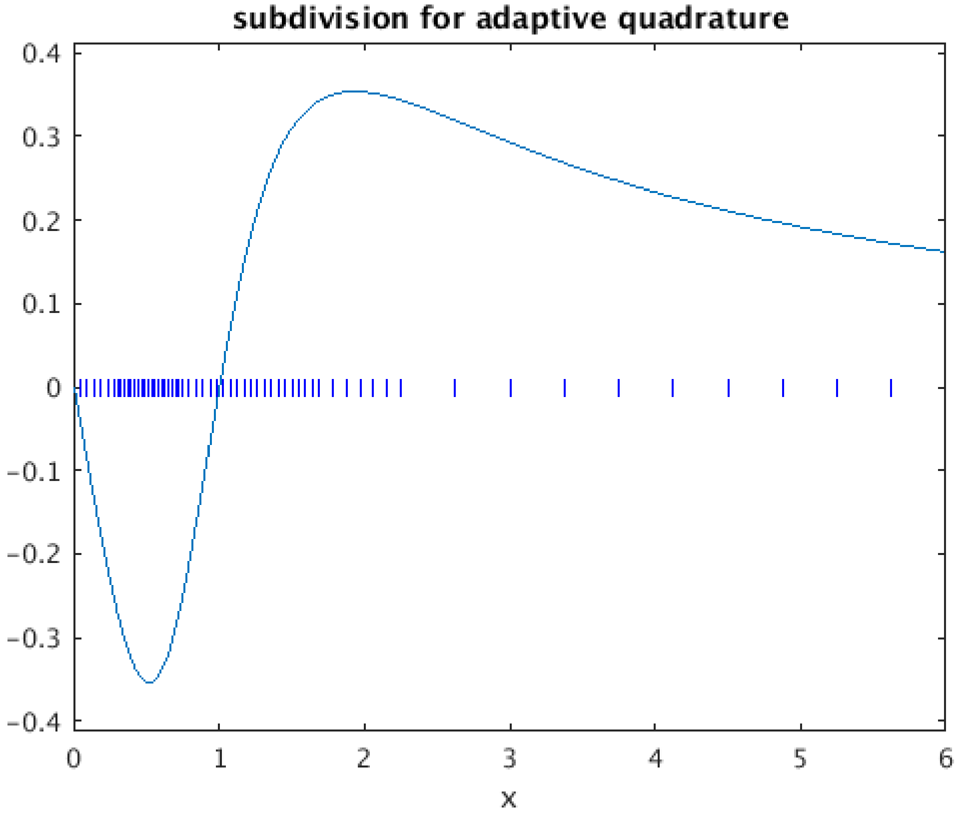
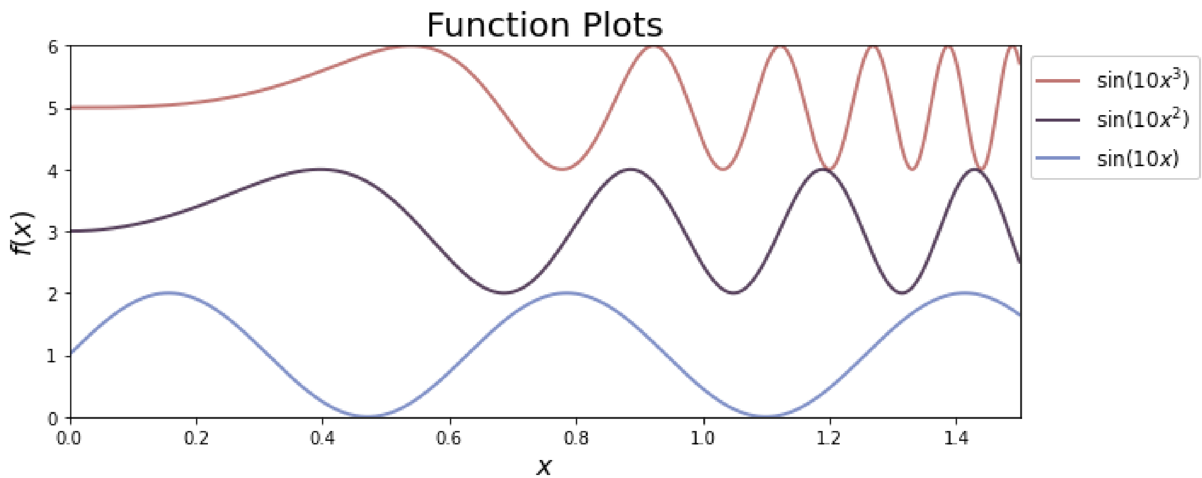
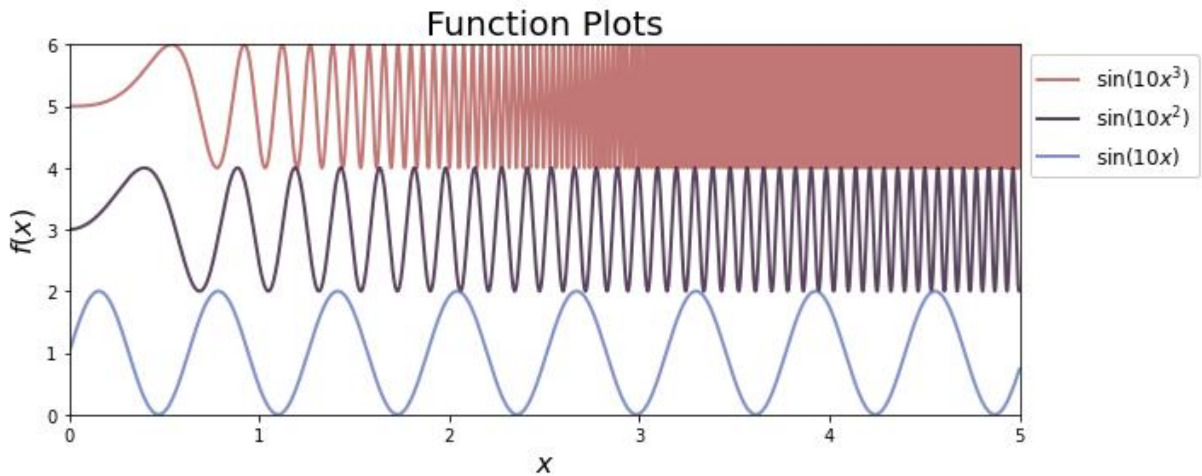


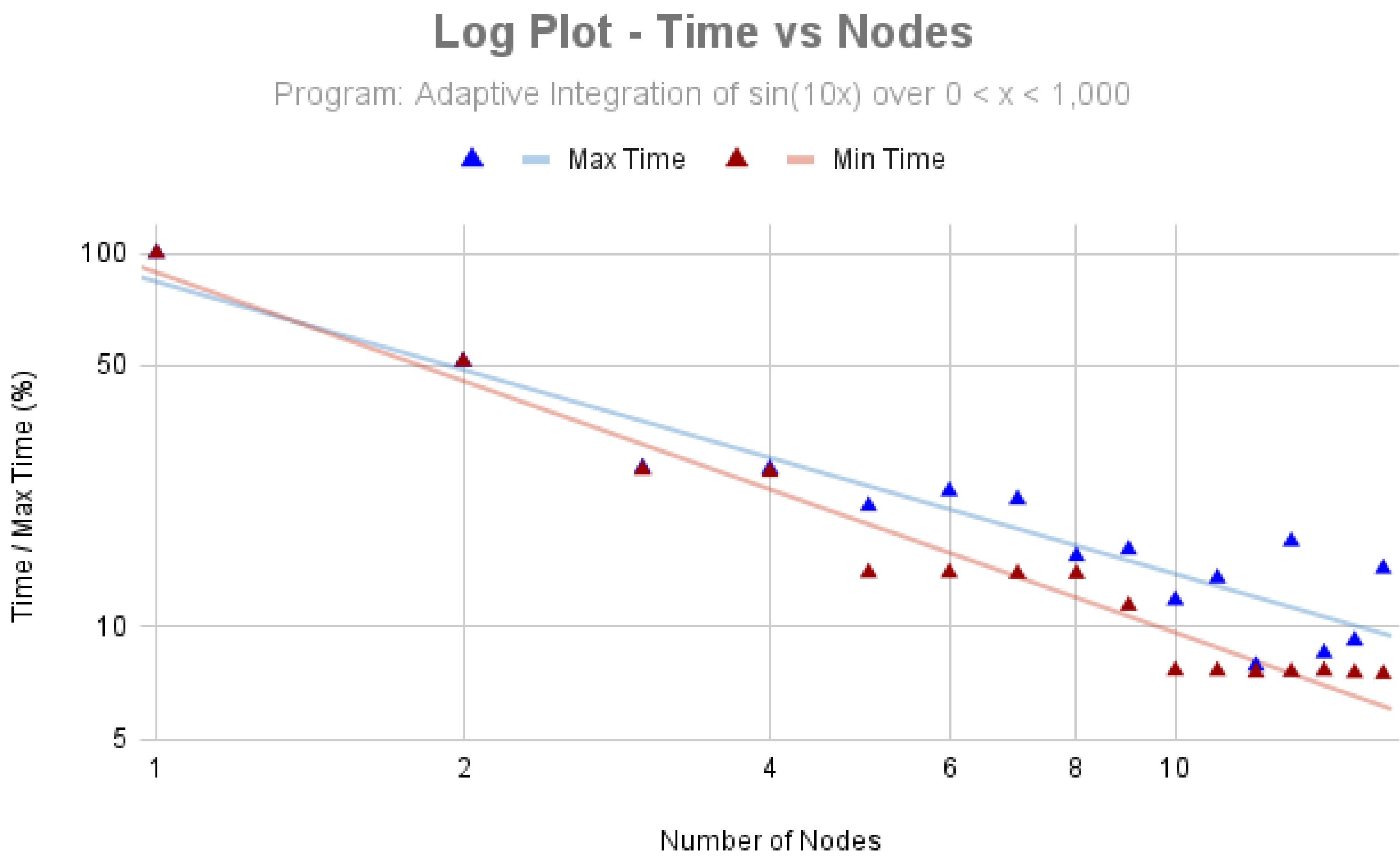
Image courtesy of https://www.math.umd.edu/~petersd/460/html/adapt_test.html

Adaptive Integration

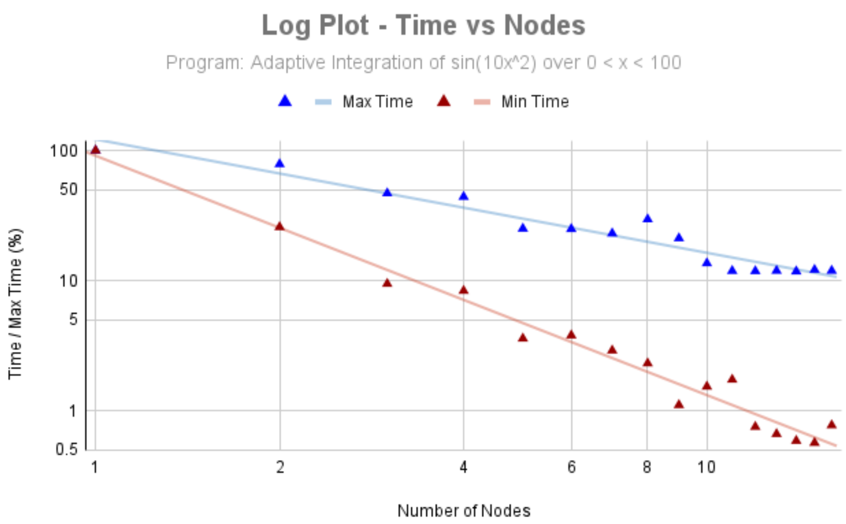


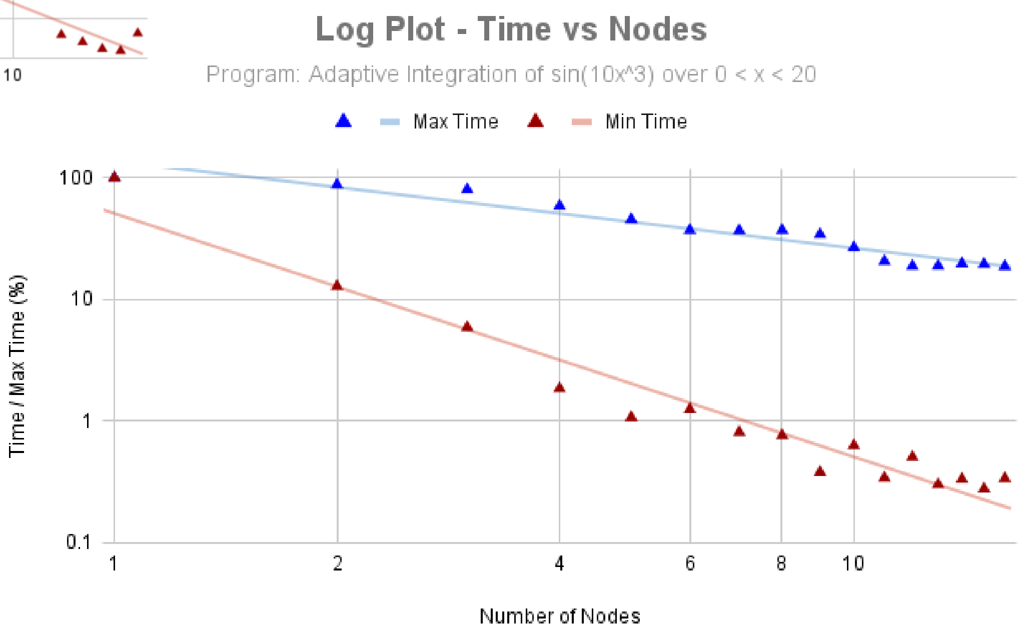


Adaptive Integration



Adaptive Integration





Parallelization - Conclusions

- Parallelization is an extremely powerful computational tool
- Requires careful consideration of the underlying tasks and ways to ensure work is distributed evenly across all nodes
  - Naïvely subdividing tasks evenly only works if the tasks are equally computationally intensive

Thank you for your attention!

Any questions?

Backup Slides

Gaussian Quadrature

- Gauss-Legendre integration is based on the roots of Legendre Polynomials.

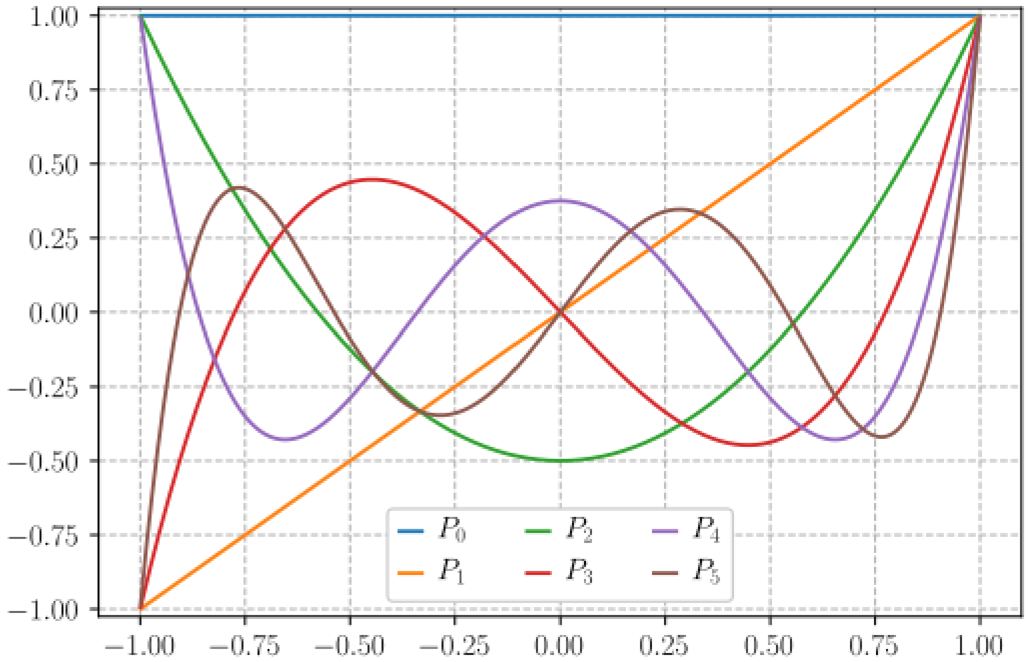


Image courtesy of https://en.wikipedia.org/wiki/Gaussian_quadrature

Gaussian Quadrature

- We approximate a definite integral with a sum of function values over a series of weights, which are computed from the roots
- These roots are computed using a bisection root-finding algorithm

$$\int_{-1}^{1} f(x)dx \simeq \sum_{i=1}^{n} w_i f(x_i)$$

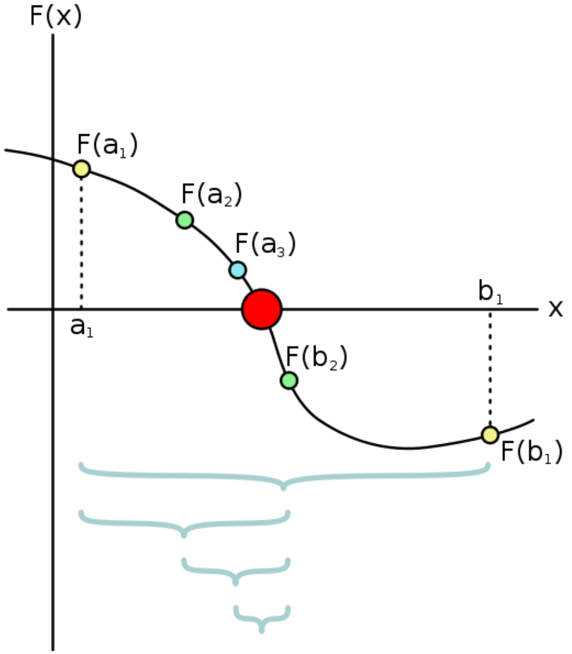


Image courtesy of https://commons.wikimedia.org/wiki/File:Bisection_method.svg

Gaussian Quadrature

- For integrals on an interval other than [-1,1], we can do a change of interval using a standard prescription,

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)\left(\frac{b-a}{2}\right)dx$$

$$\simeq \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right)$$