Parallel Scalability Analysis of 1-D and 2-D Lagrangian Interpolation Algorithms

Madeleine Biagioli

CS 554/CSE 512: Parallel Numerical Algorithms

December 18, 2017

Overview and 1-D Serial Implementation

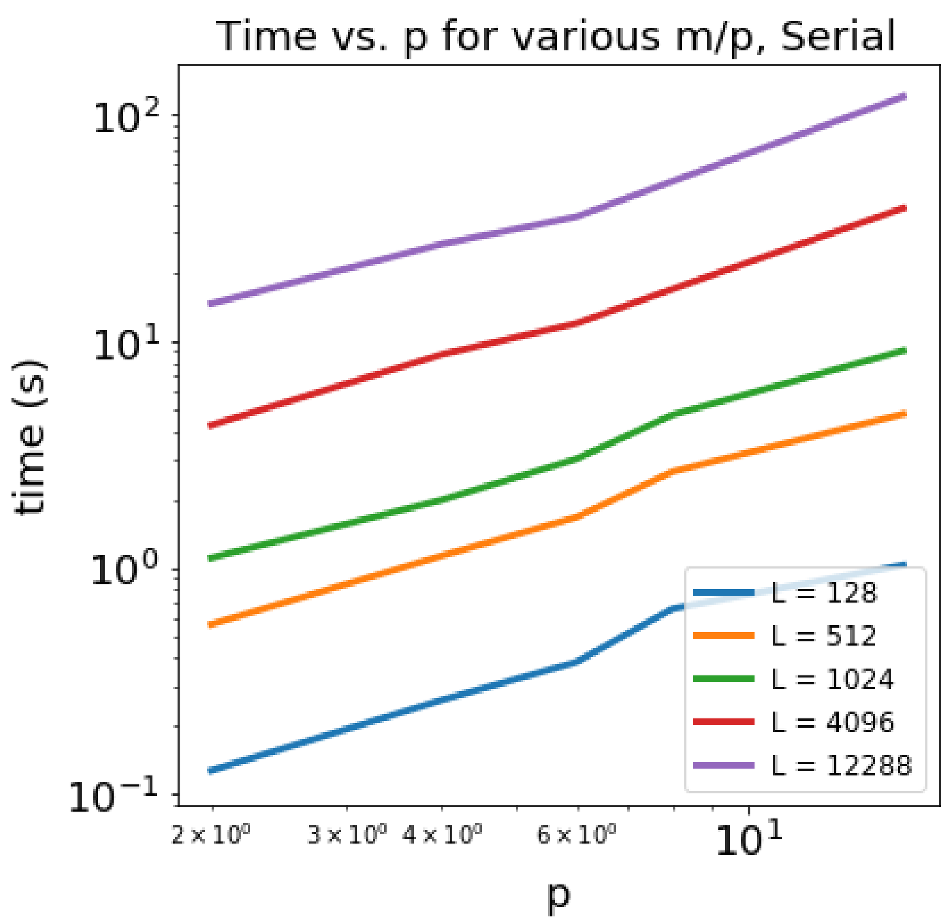
Objectives

1. To develop 1-D and 2-D Lagrangian interpolation algorithms
2. To analyze the theoretical parallel scalability of both algorithms
3. To implement the 1-D algorithm and analyze the experimental scalability

Interpolants

1-D: $$p(x) = \sum_{i=1}^{n} f(x_i)\mathcal{L}_i(x)$$ $Q_1 = O(mn)$

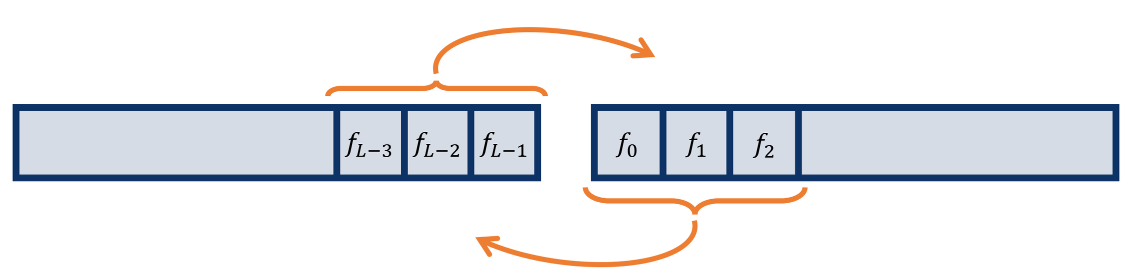
2-D: $$p(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} f(x_i, y_j)\mathcal{L}_i(x)\mathcal{L}_j(y)$$ $Q_1 = O(m^2n^2)$



1-D Algorithm

Features

- Continuous first derivative enforced between elements using $O(h^4)$ accurate finite difference formula
- Error norm of solution vector $\sim 10^{-14}$ for serial and parallel implementations



Cost and Scalability Analysis

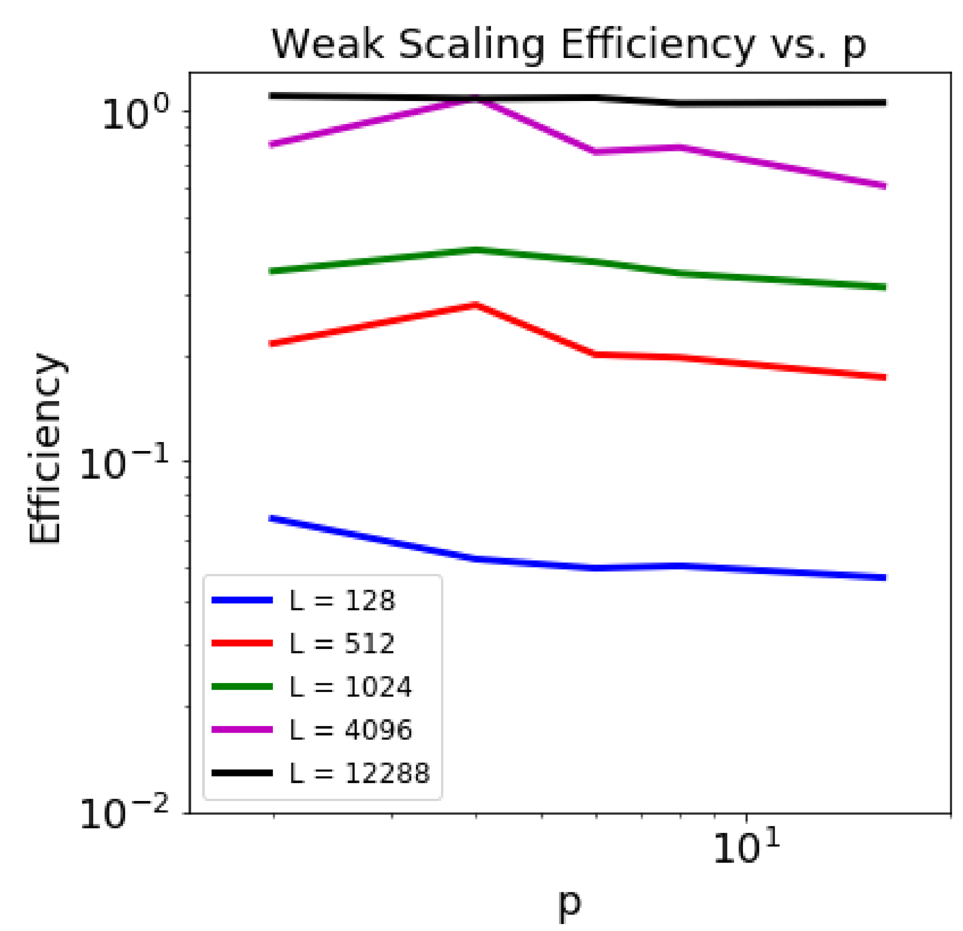
Execution Time: $$T_p = \Theta\left(\alpha + 6\beta + \frac{\gamma mn}{p}\right)$$

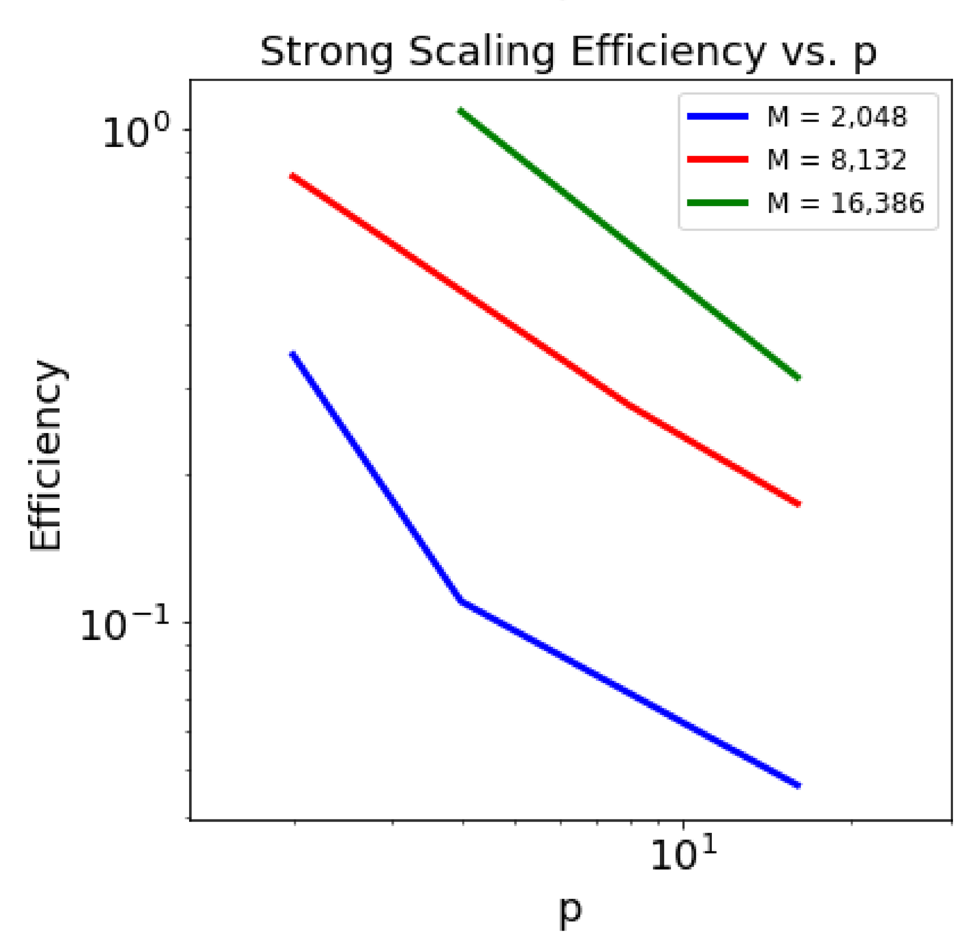
Parallel Efficiency: $$E_p = \frac{1}{\left(\frac{p}{mn}\right)\left[\left(\frac{\alpha}{\gamma}\right) + 6\left(\frac{\beta}{\gamma}\right)\right] + 1}$$

↓

Strong Scaling: $$p_s = \Theta\left(\frac{\gamma mn}{\alpha + 6\beta}\right)$$

Weak Scaling: unconditional





2-D Algorithm

Cost and Scalability Analysis

Execution Time: $$T_p = \Theta\left(4\alpha + 24\beta\frac{m}{\sqrt{p}} + \frac{\gamma m^2n^2}{p}\right)$$

Parallel Efficiency: $$E_p = \frac{1}{4\left(\frac{\alpha}{\beta}\right)\left(\frac{p}{m^2n^2}\right) + 24\left(\frac{\beta}{\gamma}\right)\left(\frac{\sqrt{p}}{mn^2}\right) + 1}$$

➡

Strong Scaling: $$p_s = \Theta\left(\min\left[\left(\frac{m^2n^2}{4}\right)\left(\frac{\gamma}{\alpha}\right), \left(\frac{m^2n^4}{576}\right)\left(\frac{\gamma}{\beta}\right)^2\right]\right)$$

Weak Scaling: unconditional

- Maximum of 4 messages sent in the 5-point stencil. Each pair of simultaneous messages contains $6m/\sqrt{p}$ words.
- Weak and strong scaling properties carry over from 1-D to 2-D algorithm.

Pseudocode

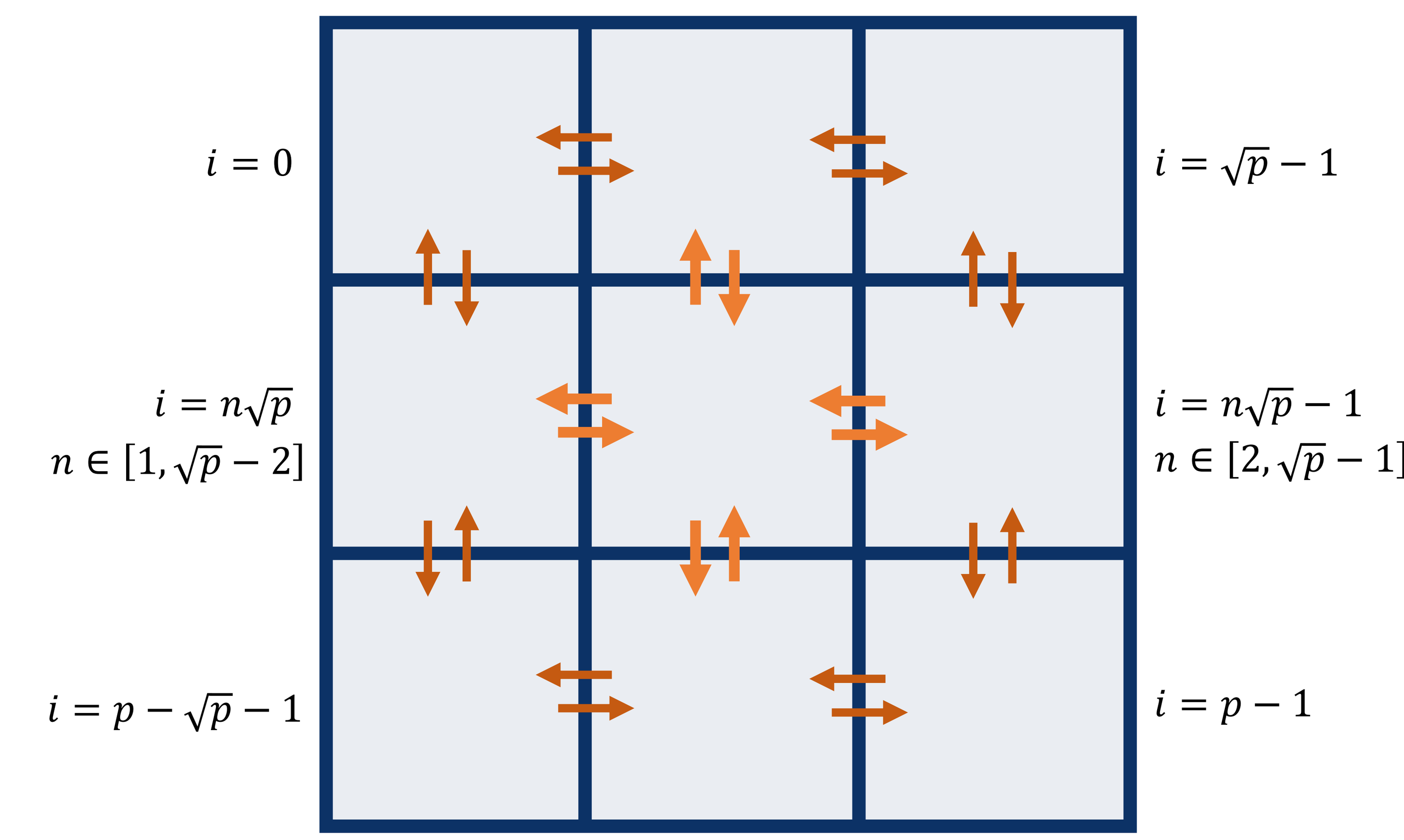
Initialize empty matrix F of element size $L \times L$
Determine basis points $x_0, \dots, x_{n-1}$ and $y_0, \dots, y_{n-1}$
Determine evaluation points $X_0, \dots, X_{L-1}$ $Y_0, \dots, Y_{L-1}$ and near-boundary step size

F = lagrange(f, x, y, X, Y) # call interpolation function
if $i < p - \sqrt{p} - 1$: send $F(\mathbf{X}, Y_{L-3}), F(\mathbf{X}, Y_{L-2}), F(\mathbf{X}, Y_{L-1})$ to $i + \sqrt{p}$
if $i > \sqrt{p} - 1$: send $F(\mathbf{X}, Y_0), F(\mathbf{X}, Y_1), F(\mathbf{X}, Y_2)$ to $i - \sqrt{p}$
if $i \neq n\sqrt{p} - 1$ for $n = 1, \dots \sqrt{p}$: send $F(X_{L-3}, \mathbf{Y}), F(X_{L-2}, \mathbf{Y}), F(X_{L-1}, \mathbf{Y})$ to $i + 1$
if $i \neq n\sqrt{p}$ for $n = 0, \dots, \sqrt{p} - 1$: send $F(X_0, \mathbf{Y}), F(X_1, \mathbf{Y}), F(X_2, \mathbf{Y})$ to $i - 1$

if $i > \sqrt{p} - 1$: recv $F(\mathbf{X}, Y_L), F(\mathbf{X}, Y_{L+1}), F(\mathbf{X}, Y_{L+2})$ from $i - \sqrt{p}$
if $i < p - \sqrt{p} - 1$: recv $F(\mathbf{X}, Y_{-3}), F(\mathbf{X}, Y_{-2}), F(\mathbf{X}, Y_{-1})$ from $i + \sqrt{p}$
if $i \neq n\sqrt{p}$ for $n = 0, \dots, \sqrt{p} - 1$: recv $F(X_{-3}, \mathbf{Y}), F(X_{-2}, \mathbf{Y}), F(X_{-1}, \mathbf{Y})$ from $i - 1$
if $i \neq n\sqrt{p} - 1$ for $n = 1, \dots \sqrt{p}$: recv $F(X_L, \mathbf{Y}), F(X_{L+1}, \mathbf{Y}), F(X_{L+2}, \mathbf{Y})$ from $i + 1$

if $i < p - \sqrt{p} - 1$:
    enforce continuous first derivative condition and update $F(\mathbf{X}, Y_{L-1})$
if $i > \sqrt{p} - 1$:
    enforce continuous first derivative condition and update $F(\mathbf{X}, Y_0)$
if $i \neq n\sqrt{p} - 1$ for $n = 1, \dots \sqrt{p}$:
    enforce continuous first derivative condition and update $F(X_0, \mathbf{Y})$
if $i \neq n\sqrt{p}$ for $n = 0, \dots, \sqrt{p} - 1$:
    enforce continuous first derivative condition and update $F(X_{L-1}, \mathbf{Y})$

F_{total} = gather(F) # compile local results



Conclusions

Objectives

1. Developed 1-D and 2-D Lagrangian interpolation algorithms.
   Central difference formula was used to ensure a continuous first derivative between elements.
   Elements of size m or $m \times m$ were interpolated from n^{th} order polynomials.
2. Analyzed the theoretical parallel scalability of both algorithms.
   Both algorithms are strongly scalable to a finite number of processors.
   Both algorithms are unconditionally weakly scalable.
3. Implemented the 1-D algorithm and analyze the experimental scalability.
   Unconditional weak scalability was verified experimentally.
   Finite strong scalability was verified experimentally.
   Error for serial and parallel implementation was consistent.