Asynchronous Finite Difference Scheme for PDEs

MA14M004
Kumar Saurabh

RWTH Aachen University
Aachen

December 4, 2015

FD Schemes

Finite Difference Schemes

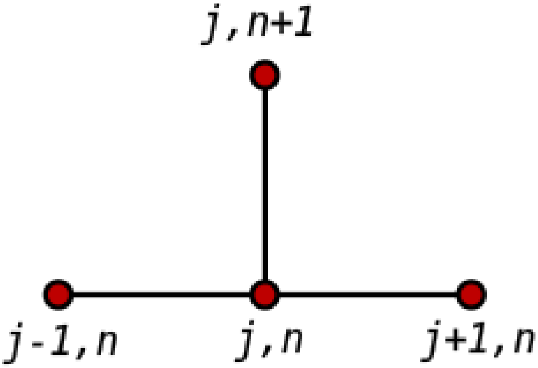
FD method can be used to discretize and solve PDEs.
Consider the heat equation:

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} \tag{1}$$

which can be discretized as:

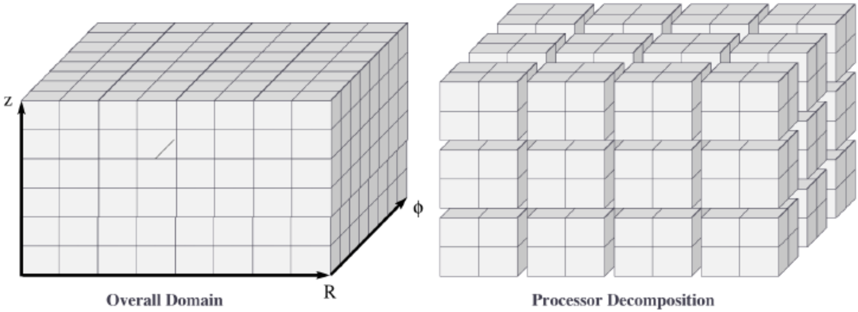
$$\frac{u_i^{n+1} - u_i^n}{\triangle t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\triangle x^2} + \mathscr{O}(\triangle x^2, \triangle t) \tag{2}$$

FD Stencil



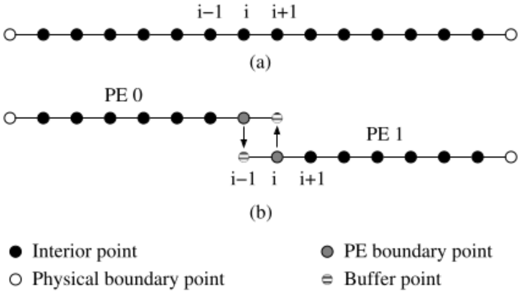
Parallel Computing

Divide Workload among different processor.



Synchronization

Stencil needs information not on the PE for boundary nodes.



Communication between PEs is required.

Algorithm

```
for(t = 0; t < final_T; t+=delta_t){
        for(i = 0; i < ne; i++){
        //Calculate according to the scheme
        //ne: number of elements on each
   processor
        .......
        }
        MPI_Send(boundary nodes);
        MPI_Recv(boundary nodes);
        MPI_Barrier(MPI_COMM_WORLD)

}
```

- Can we somehow avoid synchronization and reduce communication overhead?

Consider the explicit form of the heat equation:

$$u_i^{n+1} = u_i^n + \frac{\alpha \triangle t}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{3}$$

Problem: To compute u_i^{n+1} we need values of u that are possibly not on the PE

Solution:

- For the left boundary

$$u_i^{n+1} = u_i^n + \frac{\alpha \triangle t}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^{\tilde{n}}) \tag{4}$$

- For the right boundary

$$u_i^{n+1} = u_i^n + \frac{\alpha \triangle t}{\triangle x^2}(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n) \tag{5}$$

where $\tilde{n}$ is the last available value for a particular node.

Regarding $\tilde{n}$

- Synchronous when $\tilde{n} = n$
- $\tilde{n}$ can be $n, n-1, n-2, \ldots$
- Concrete value of $\tilde{n}$ depends on hardware, network, traffic, (possible) unpredictable factors,...
- $\tilde{n}$ is in fact a principle random variable.

Results from Theoretical Analysis

- Stability is independent of statistics of delay.
- Stability is independent of the number of PEs.
- However, accuracy depends on both delay and number of PEs.

Results from Theoretical Analysis

- Second order convergence in space without delays
- First order convergence in space with delay

Numerical Results

Problem

Advection - Diffusion Equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2} \tag{6}$$

Initial Condition:

$$u(x,0) = \sum_{\kappa} A(\kappa) sin(\kappa x) \tag{7}$$

which has the analytical result:

$$u_a(x,t) = \sum_{\kappa} \exp^{(-\alpha\kappa^2 t)} A(\kappa) sin(\kappa(x - ct)) \tag{8}$$

Discretization

Interior nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{9}$$

Left boundary nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^{\tilde{n}}) = \frac{\alpha}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^{\tilde{n}}) \tag{10}$$

Right boundary nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^{\tilde{n}} - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n) \tag{11}$$

Parameters for Simulation

```
len = 2*pi; % length of domain
A = 1;
k = 2;
alpha = 1;
r_alpha = 0.1; % CFL
c = 1;
final_t = 0.08*len/c
```

where

$$r_\alpha = \frac{\alpha \triangle t}{\triangle x^2} \tag{12}$$

Results from Matlab Implementation

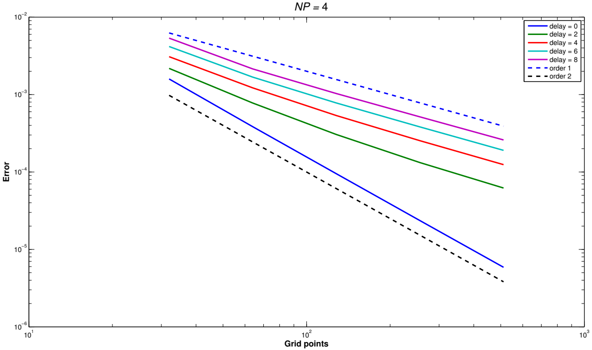


Figure : Number of PEs = 4

Results from Matlab Implementation

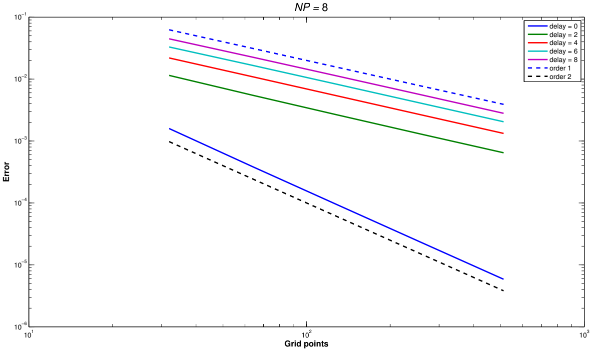
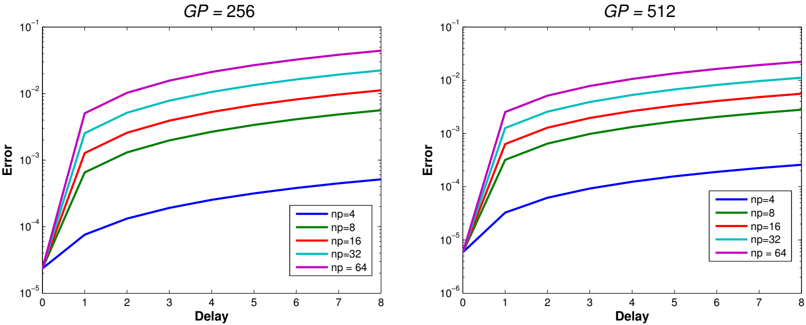
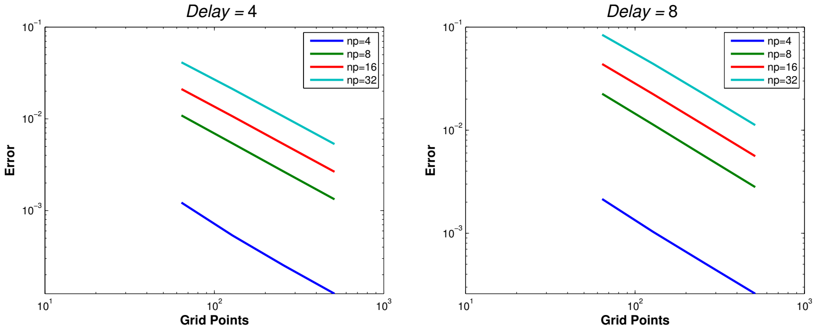


Figure : Number of PEs = 8



(a) Number of Grid Point = 256 (b) Number of Grid Point = 512

Figure : Influence of Delay on Grid Point



(a) Delay = 4 (b) Delay = 8

Figure : Influence of number of Grid Point for different number of PEs

C++ implementation

- Master Work Paradigm using MPI.
- OpenMP shared Memory
- MPI approach

Master Work Paradigm using MPI

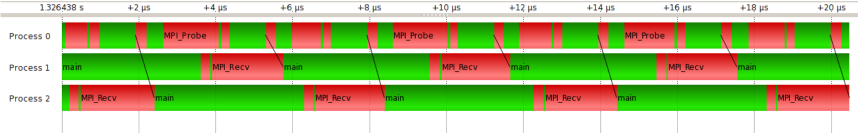


Figure : Vampir Trace with Grid Point 512.

MPI approach

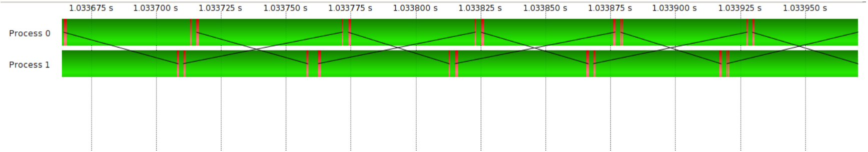
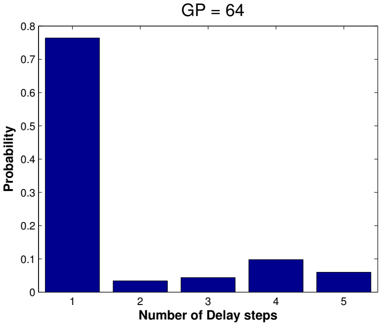
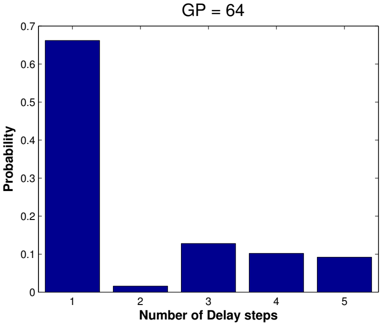


Figure : Vampir Trace with Grid Point 512.

Statistics of Delay. Maximum Allowable = 5 time steps



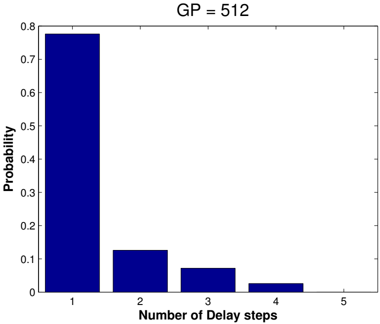
(a) Processor 1



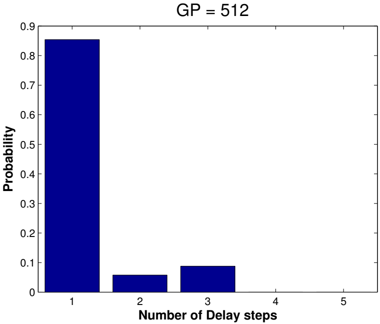
(b) Processor 2

Figure : Grid Points = 64

Statistics of Delay. Maximum Allowable = 5 time steps



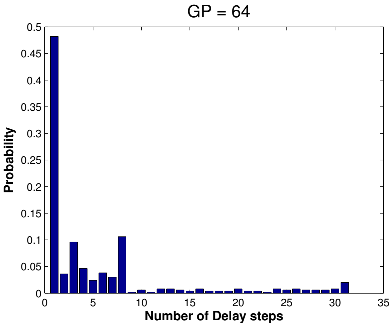
(a) Processor 1



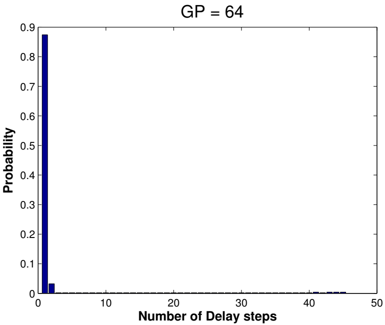
(b) Processor 2

Figure : Grid Points = 512

Statistics of Delay. No Waiting Time



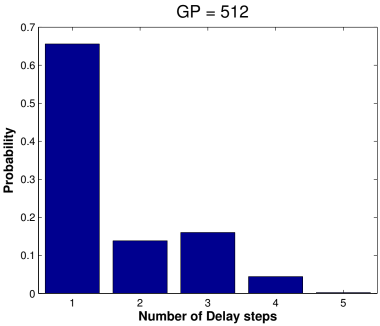
(a) Processor 1



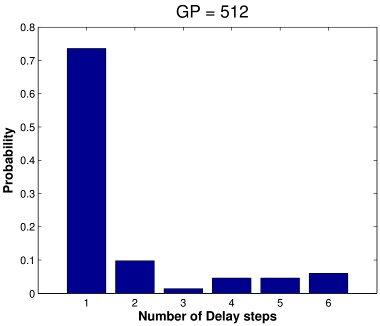
(b) Processor 2

Figure : Grid Points = 64

Statistics of Delay. No Waiting Time



(a) Processor 1



(b) Processor 2

Figure : Grid Points = 512

Remarks

- The statistics of delay is highly stochastic.
- The above results are obtained by 500 time step runs.
- The buffer size must be sufficiently large to hold values at all time steps.

Summary

- Avoid synchronization by asynchronous schemes.
- The statistics of delay is highly stochastic.

Future Work

- Extend the approach to multiple processor.
- Formally verify the difference in all the approaches.
- Order Recovery.
- Performance Analysis

Refrences

- Diego A. Donzis and Konduri Aditya. Asynchronous Finite Difference Scheme for Partial Difference Equations *Journal of Computational Physics*. 274(0):370-392,2014
- Thomas Camminady. CES Seminar Paper on Asynchronous Finite Difference Scheme for Partial Difference Equation. January 9,2015
- MPICH , `http://www.mpich.org/`, 4 12 2015.