Asynchronous Finite Difference Scheme for PDEs

Kumar Saurabh [1] Thomas Camminady[2] Prof. Dr. Martin Frank [2]

[1] Department of Mathematics
IIT Madras

[2] Math CCES
RWTH Aachen

March 22, 2016

Acknowledgement: DAAD, RWTH IT Center

Outline

- Diego A. Donzis and Konduri Aditya. Asynchronous Finite Difference Scheme for Partial Difference Equations *Journal of Computational Physics*. 274(0):370-392,2014

Motivation

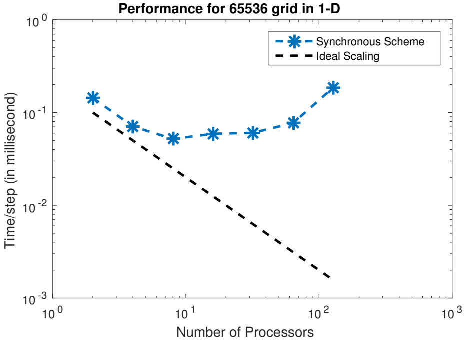
- Many natural and engineering systems can be described with PDEs
  - Fluid mechanics, Electromagnetism, Quantum Mechanics
- Analytical Solution not known. Need to solve these problems numerically.

- Large number of numerical methods: Finite Difference, Finite Volume, Spectral Method, etc.
  - The complexity of systems at realistic conditions typically requires massive computational resources.
  - The problem is decomposed into a large number of Processing Element (PEs).
  - The communication is required between the PEs to solve the PDEs to compute spatial derivatives.
- Computation rates are much faster than communication
  - Exascale: Communication likely to be bottleneck.

Examples

- Direct Numerical Simulation (DNS)
  - Resolve all scales in space and time.
  - Computationally very expensive.
  - Communication or synchronization takes upto 50 - 70 % of the total computation time.[1]

[1] Jagannathan & Donzis (XSEDE 2012), Sankaran et al. (SC 2012), Lee et al. (SC 2013)



Performance for 65536 grid in 1-D
Synchronous Scheme
Ideal Scaling
Time/step (in millisecond)
10 0
10 -1
10 -2
10 -3
10 0
10 1
10 2
10 3
Number of Processors

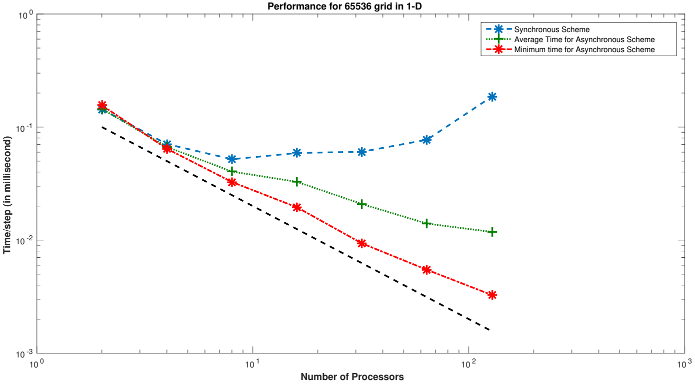
Simulations at exascale

Issues with the current Method:

- Communications.
- Synchronization at various time level.

New approach:

- Can we relax Synchronization?
- Asynchronous numerical Schemes.
- Objective: trade-off accuracy and performance quantitatively and predictably.


Performance for 65536 grid in 1-D
Synchronous Scheme
Average Time for Asynchronous Scheme
Minimum time for Asynchronous Scheme
Time/step (in millisecond)
Number of Processors
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{0}
10^{1}
10^{2}
10^{3}

Performance Improvement

- Computation time
  - Hardware: Faster hardware, Larger Memory Size.
  - Numerical schemes: Fewer operations.
- Communication time
  - Hardware: Network topology, switches, etc.
  - Numerical Scheme:
    - Fewer communications, Larger messages
    - Asynchronous Scheme: Stable and Consistent.

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

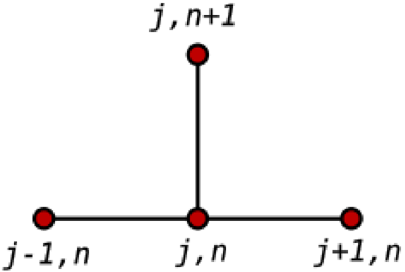


Figure : Finite Difference Stencil for Diffusion Problem.[2]

[2] Adapted From Wikepedia

Stencil needs information not on the PE for boundary nodes.

D.A. Donzis, K. Aditya / Journal of Computational Physics 274 (2014) 370–392

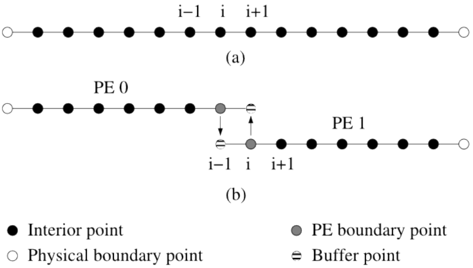


Fig. 1. Discretized one-dimensional domain. (a) Domain in serial codes. (b) Same domain decomposed into two PEs.

Communication between PEs is required.

Consider the explicit form of the heat equation:

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

$$u_i^{n+1} = u_i^n + r_\alpha(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{4}$$

where:

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

Compute:

$$V^{n+1} = [u_1^{n+1} \quad u_2^{n+1} \quad ... \quad u_N^{n+1}]^T \tag{5}$$

according to

$$V^{n+1} = A^n V^n \tag{6}$$

where, assuming periodic boundary conditions:

$$A^n = \begin{pmatrix} 1-2r_\alpha & r_\alpha & 0 & 0 & \dots & 0 & 0 & r_\alpha \\ r_\alpha & 1-2r_\alpha & r_\alpha & 0 & \dots & 0 & 0 & 0 \\ & & \ddots & & & & & \\ r_\alpha & 0 & 0 & \dots & 0 & 0 & r_\alpha & 1-2r_\alpha \end{pmatrix} \tag{7}$$

Stability

- For stability:

$$\frac{\left\|V^{n+1}\right\|}{\left\|V^{0}\right\|} \leq 1 \tag{8}$$

- Since $V^{n+1} = A^n V^n = A^n A^{n-1} V^{n-1} = ... = A^n A^{n-1} ... A^0 V^0$.
- A is independent of n. ($A^n = A^{n-1} = ... = A$).
- $\|A\|_\infty \leq 1$ [3]which is reduced to

$$|r_\alpha| + |(1-2r_\alpha)| + |r_\alpha| \leq 1 \tag{9}$$

which is satisfied by $r_\alpha \leq \frac{1}{2}$

[3] $\|B\|_\infty = max_i \sum_j |b_{ij}|$

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

- For the left boundary

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

- For the right boundary

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

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.

Assume $\tilde{n}$ is n or $n-1$.
Delay occurs only on the right boundary.

$$W^{n+1} = \tilde{C}_n W_n \tag{12}$$

with

$$W^{n+1} = [V^{n+1}, V^n]^T \tag{13}$$

and

$$\tilde{C}^n = \begin{bmatrix} \tilde{A}_0^n & \tilde{A}_1^n \\ I & 0 \end{bmatrix} \tag{14}$$

where

$$\tilde{A}_0^n = \begin{pmatrix} 1-2r_\alpha & (1-\tilde{k_2})r_\alpha & 0 & 0 & \dots & 0 & 0 & r_\alpha \\ r_\alpha & 1-2r_\alpha & (1-\tilde{k_3})r_\alpha & 0 & \dots & 0 & 0 & 0 \\ & & \ddots & & & & & \\ (1-\tilde{k_1})r_\alpha & 0 & 0 & \dots & 0 & 0 & r_\alpha & 1-2r_\alpha \end{pmatrix} \tag{15}$$

$$\tilde{A}_1^n = \begin{pmatrix} 0 & \tilde{k_2}r_\alpha & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \tilde{k_3}r_\alpha & 0 & \dots & 0 & 0 & 0 \\ & & \ddots & & & & & \\ \tilde{k_1}r_\alpha & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix} \tag{16}$$

$\tilde{k_i}$ switch between 0 and 1.

Consider the i^{th} row of $\tilde{C}^n$:

$$\begin{bmatrix} 0 & \dots & r_\alpha & (1-2r_\alpha) & (1-\tilde{k}_i)r_\alpha & 0 & \dots & 0 | 0 & \dots & \tilde{k}_i r_\alpha & \dots & 0 \end{bmatrix} \tag{17}$$

and thus:

$$|r_\alpha| + |(1-2r_\alpha)| + |(1-\tilde{k}_i)r_\alpha| + |\tilde{k}_i r_\alpha| \leq 1 \tag{18}$$

for convergence.
Since $\tilde{k}_i$ can be either 0 or 1, the condition becomes equivalent to:

$$|r_\alpha| + |(1-2r_\alpha)| + |r_\alpha| \leq 1 \tag{19}$$

Similarly, this approach can be extended to any arbitrary value of delay.
$\Rightarrow$ Asynchronous Scheme is Stable if Synchronous Scheme is stable.

Synchronous Scheme:

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \alpha\frac{u''''}{12}\Delta x^2 + \mathscr{O}(\Delta t^2, \Delta x^4) \tag{20}$$

Asynchronous Scheme: Assuming $\tilde{n} = n - k$

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \alpha\frac{u''''}{12}\Delta x^2 - \alpha k\dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k\dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k\dot{u}''}{2}\Delta t + \mathscr{O}(\Delta t^2, \Delta x^3, \Delta x^p\Delta t^q)^4 \tag{21}$$

Assuming a constant Courant Number r_α,

$$E_i^n = -\frac{r_\alpha\ddot{u}}{2\alpha}\Delta x^2 + \alpha\frac{u''''}{12}\Delta x^2 - r_\alpha k\dot{u} + r_\alpha k\dot{u}'\Delta x - \frac{r_\alpha k\dot{u}''}{2}\Delta x^2 + \mathscr{O}(\Delta t^2, \Delta x^3, \Delta x^p\Delta t^q) \tag{22}$$

[4] p and q are bounded from below by -2 and 1 respectively

Truncation Error

- Not homogeneous in space and random.
- Need for statistical description of the truncation error.
- $\langle E \rangle$: Spatial average taken over the entire domain.
- $\overline{E}$: Ensemble average taking into account the stochastic nature of delay.

$$\langle \overline{E} \rangle = \frac{1}{N} \sum_{i=1,N} \overline{E_i^n} \tag{23}$$

Let I_I denotes the point belonging to interior nodes of PE and I_B the boundary nodes.

$$\langle \overline{E} \rangle = \frac{1}{N} \left[\sum_{i \in I_I} \overline{E_i^n} + \sum_{i \in I_B} \overline{\tilde{E}_i^n} |_{\tilde{k}_{i+1}} \right] \tag{24}$$

Synchronous part:

$$\begin{aligned} \sum_{i \in I_I} E_i^n \approx & \Delta x^2 \sum_{i \in I_I} K_S \\ \approx & N_I \langle K_S \rangle_I \Delta x^2 \end{aligned} \tag{25}$$

where

$$\langle K_S \rangle = -\frac{r_\alpha \ddot{u}}{2\alpha} \Delta x^2 + \alpha \frac{u''''}{12} \Delta x^2 \tag{26}$$

Asynchronous part:

$$\sum_{i \in I_B} \overline{\tilde{E}_i^n}|_{\tilde{k}_{i+1}} \approx \sum_{k=0}^{L-1} p_{k[i+1]} \tilde{E}_i^n|_{\tilde{k}_{i+1}=k}$$

$$\approx \left(-\frac{r_\alpha \ddot{u}}{2\alpha}\Delta x^2 + \alpha \frac{u''''}{12}\Delta x^2\right) + \left(-r_\alpha \dot{u} + r_\alpha \dot{u}' \Delta x - \frac{r_\alpha \dot{u}''}{2}\Delta x^2\right)\bar{\tilde{k}}^5 \quad (27)$$

$$\approx N_B \langle K_S \rangle_B \Delta x^2 + N_B \left(-r_\alpha \langle \dot{u} \rangle_B + r_\alpha \langle \dot{u}' \rangle_B \Delta x - \frac{r_\alpha \langle \dot{u}'' \rangle_B}{2}\Delta x^2\right)\bar{\tilde{k}}$$

Overall Error:

$$\langle E \rangle \approx \langle K_S \rangle \Delta x^2 + \frac{N_B}{N}\left(-r_\alpha \langle \dot{u} \rangle_B + r_\alpha \langle \dot{u}' \rangle_B \Delta x - \frac{r_\alpha \langle \dot{u}'' \rangle_B}{2}\Delta x^2\right)\bar{\tilde{k}} \quad (28)$$

[5] Assumption: k_i is independent of i

- For zero delays: Second Order Accurate Scheme.
- Even a very small amount of asynchrony decreases the order of convergence significantly.

Asymptotically:

$$\frac{N_B}{N} \propto \frac{P}{N} \tag{29}$$

which results:

$$\begin{aligned} \langle \overline{E} \rangle &\approx -\frac{N_B}{N} \overline{\tilde{k}} r_\alpha \langle \dot{u} \rangle_B \\ &\propto \frac{P}{N} \propto \Delta x \end{aligned} \tag{30}$$

Recap

- Synchronous Scheme: Stable $\Rightarrow$ Asynchronous Scheme: Stable.
- Second order Convergence without delays.
- First Order Convergence in presence of delays.

Numerical Results

Problem

Advection - Diffusion Equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{31}$$

Initial Condition:

$$u(x,0) = \sum_{\kappa} A(\kappa) sin(\kappa x) \tag{32}$$

Boundary Conditions: Periodic

Analytical result:

$$u_a(x,t) = \sum_{\kappa} \exp^{(-\alpha\kappa^2 t)} A(\kappa) sin(\kappa(x - ct)) \tag{33}$$

Parameters for Simulation

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

where

$$r_\alpha = \frac{\alpha \triangle t}{\triangle x^2} \tag{34}$$

Discretization

Interior nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{35}$$

Left boundary nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^{\tilde{n}}) = \frac{\alpha}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^{\tilde{n}}) \tag{36}$$

Right boundary nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^{\tilde{n}} - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n) \tag{37}$$

Results

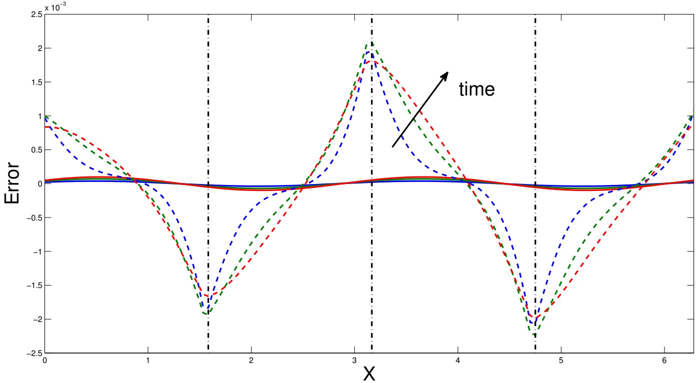


Figure : Comparison of Error in case of Synchronous and Asynchronous Scheme. Dashed line represent Asynchronous Case and Solid Lines Synchronous Case. Vertical Lines represent PE boundaries

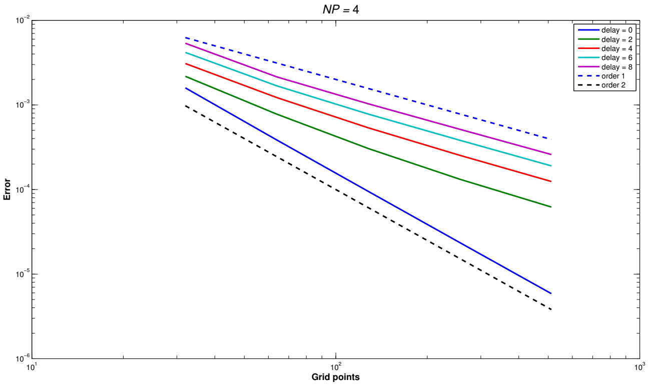


Figure : Number of PEs = 4

Results

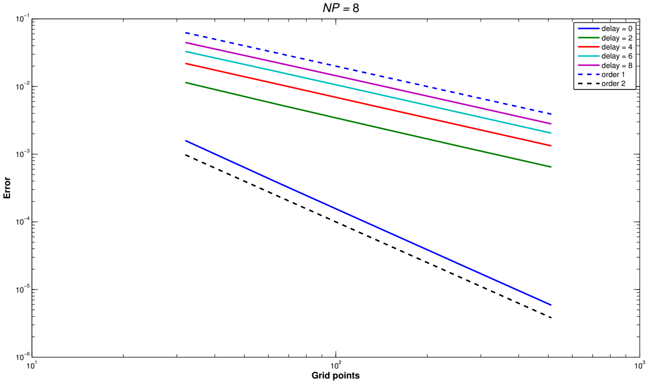
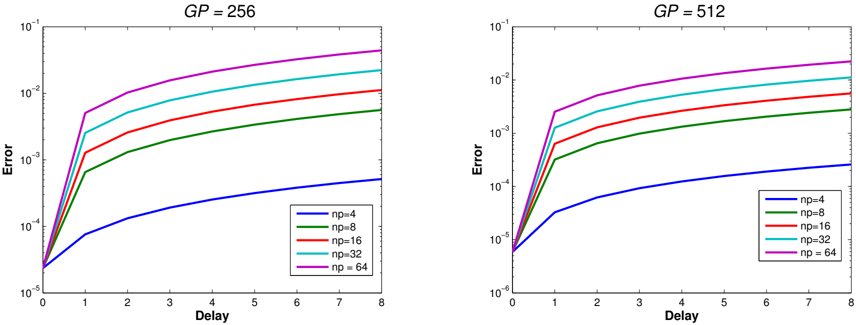


Figure : Number of PEs = 8

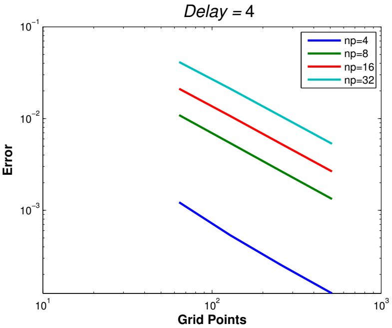
Order

Delay	Async
0(sync)	-2.0195
1	-1.0764
2	-1.0371
4	-1.0117
6	-1.0033
8	-0.9995

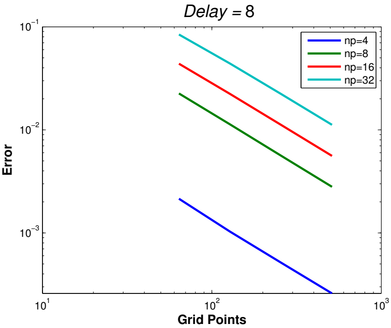


(a) Number of Grid Point = 256 (b) Number of Grid Point = 512

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

Observations

- Maximum error is observed at the boundaries in case of asynchronous Scheme.
- Second Order Convergence in space without delays.
- First Order Convergence in space with delays.
- Larger delays means larger error.
- More number of Processor $\Rightarrow$ More Boundary Points $\Rightarrow$ More Error.

- Order of Convergence falls to first order in case of Asynchronous Scheme.
- Need to recover the order.
- Can be achieved by analysis of Truncation Error and eliminating terms that effect accuracy.
  - Changing the time step size Δt.
  - Newer Scheme.

Order Recovery

Approximating Δt as Δx^3 and substituting in the truncation error of the following scheme:

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \alpha\frac{u''''}{12}\Delta x^2 - \alpha k\dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k\dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k\dot{u}''}{2}\Delta t + \mathscr{O}(\Delta t^2, \Delta x^3, \Delta x^p\Delta t^q) \tag{38}$$

gives a scheme of $\mathscr{O}(\Delta x)$ and overall order of the Scheme will be $\mathscr{O}(\Delta x^2)$

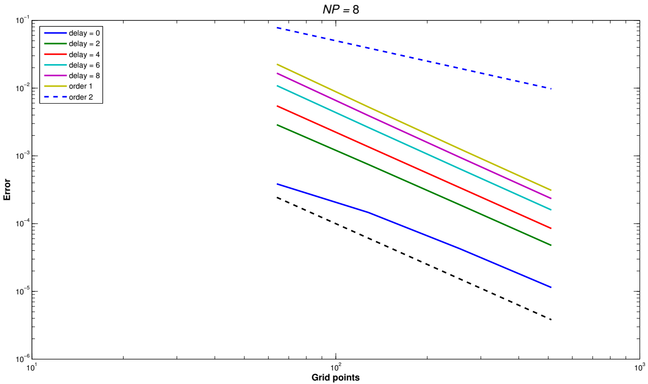


Figure : Number of PEs = 4

Delay	Async	Changing Δt
0(sync)	-2.0195	-1.9008
1	-1.0764	-1.9712
2	-1.0371	-2.0048
4	-1.0117	-2.0313
6	-1.0033	-2.0488
8	-0.9995	-2.0609

Order Recovery Scheme

Aim: To obtain a scheme of $\mathscr{O}(\Delta x)$.
Let us consider the modified Stencil:

$$\frac{\partial^2 u}{\partial x^2} = \frac{b_{-2} u_{i-2}^{n} + b_{-1} u_{i-1}^{n} + b_0 u_i^n + b_1 u_{i+1}^{\tilde{n}} + b_2 u_{i+2}^{\tilde{n}}}{\Delta x^2} \tag{39}$$

The resulting Stable Scheme [6]obtained after truncation error analysis is:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) = \frac{\alpha}{2\triangle x^2}(u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_i^n + u_{i-1}^n) + \mathscr{O}(k\Delta t, \Delta x, k\Delta t/\Delta x) \tag{40}$$

[6]Stable for $r_\alpha \leq 0.5$ and $r_c \leq \sqrt{2r_\alpha} - r_\alpha (r_c = \frac{c\Delta t}{\Delta x})$

Scheme

Interior nodes:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (41)$$

Right Boundary node:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^{\tilde{n}} - u_{i-1}^n) = \frac{\alpha}{2\triangle x^2}(u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_i^n + u_{i-1}^n) \quad (42)$$

Left Boundary Node

Scheme 1

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^{\tilde{n}}) = \frac{\alpha}{2\triangle x^2}(u_{i+2}^n - u_{i+1}^n - u_i^n + u_{i-1}^{\tilde{n}}) \tag{43}$$

Scheme 2:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^n - u_{i-1}^{\tilde{n}}) = \frac{\alpha}{2\triangle x^2}(u_{i+1}^n - u_i^n - u_{i-1}^{\tilde{n}} + u_{i-2}^{\tilde{n}}) \tag{44}$$

Scheme 1

- Left Boundary Node:

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) = \frac{\alpha}{2\triangle x^2}(u_{i+2}^n - u_{i+1}^n - u_i^n + u_{i-1}^{\tilde{n}}) + \mathscr{O}(k\Delta t, \Delta x, k\Delta t/\Delta x^2) \quad (45)$$

At constant r_α the scheme again gets reduced to the first order Scheme.

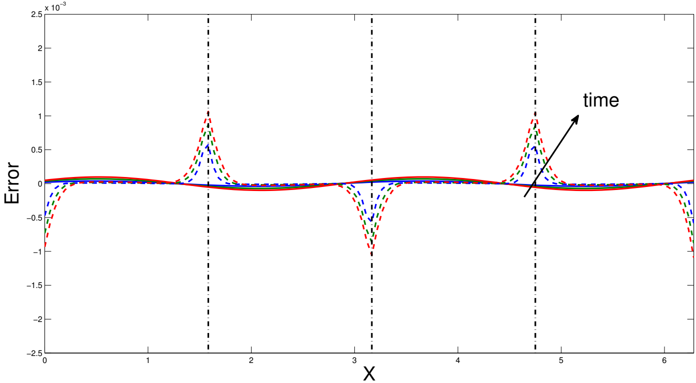


Figure : Comparison of Error in case of Synchronous and Asynchronous Scheme. Dashed line represent Asynchronous Case and Solid Lines Synchronous Case. Vertical Lines represent PE boundaries

Scheme 1

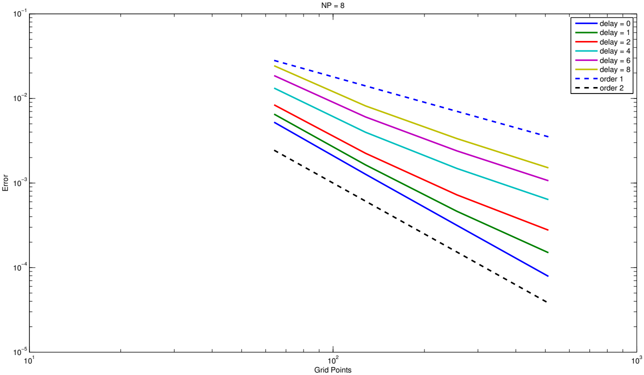


Figure : Number of PEs = 4

Delay	Async	Changing Δt	Scheme 1
0(sync)	-2.0195	-1.9008	-2.0160
1	-1.0764	-1.9712	-1.8141
2	-1.0371	-2.0048	-1.6396
4	-1.0117	-2.0313	-1.4558
6	-1.0033	-2.0488	-1.3719
8	-0.9995	-2.0609	-1.3311

Scheme 2

- Right Boundary node

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) = \frac{\alpha}{2\triangle x^2}(u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_i^n + u_{i-1}^n) + \mathscr{O}(k\Delta t, \Delta x, k\Delta t/\Delta x) \quad (46)$$

- Left Boundary Node:

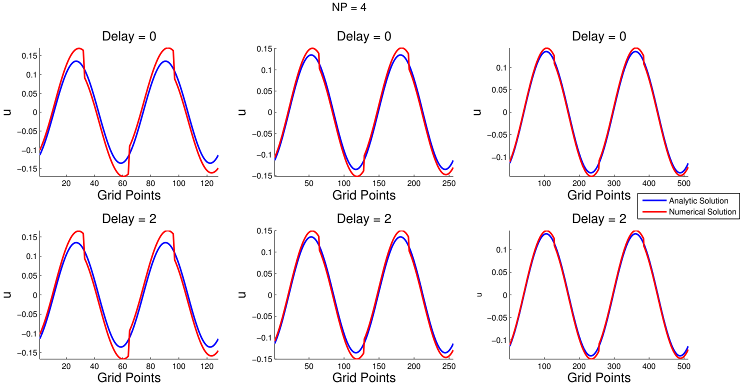
$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) = \frac{\alpha}{2\triangle x^2}(u_{i+1}^n - u_i^n - u_{i-1}^{\tilde{n}} + u_{i-2}^{\tilde{n}}) + \mathscr{O}(k\Delta t, \Delta x, k\Delta t/\Delta x) \quad (47)$$

- Right Boundary Point on a processor: ne.
- Left Boundary Point on the next processor: ne + 1.
- Considering Synchronous case: Spatial Approximation of derivative for both the nodes is same.

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{ne+2}^n - u_{ne+1}^n - u_{ne}^n + u_{ne-1}^n}{2\Delta x^2} \quad (48)$$

- Inconsistent Formulation.

Scheme 2


NP = 4
Delay = 0
Delay = 0
Delay = 0
Delay = 2
Delay = 2
Delay = 2
u
Grid Points
Analytic Solution
Numerical Solution

Scheme 3

- Assumption: Only one of the neighbouring processor faces delay. (No communication delay)
- For the processor that faces delay, compute according to the new asynchronous scheme.
- For the processor that does not face delay, compute according to the standard scheme.

Scheme 3

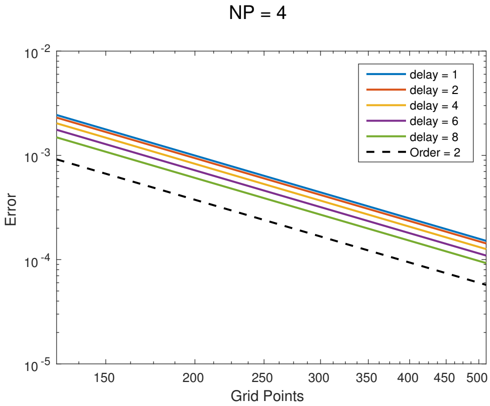


Figure : Number of PEs = 4

Delay	Async	Changing Δt	Scheme 1	Scheme 3
0(sync)	-2.0195	-1.9008	-2.0160	-2.0195
1	-1.0764	-1.9712	-1.8141	-2.0050
2	-1.0371	-2.0048	-1.6396	-2.0049
4	-1.0117	-2.0313	-1.4558	-2.0044
6	-1.0033	-2.0488	-1.3719	-2.0039
8	-0.9995	-2.0609	-1.3311	-2.0029

Can be implemented as:

- Deterministic Asynchronous Scheme
- Stochastic Asynchronous Scheme

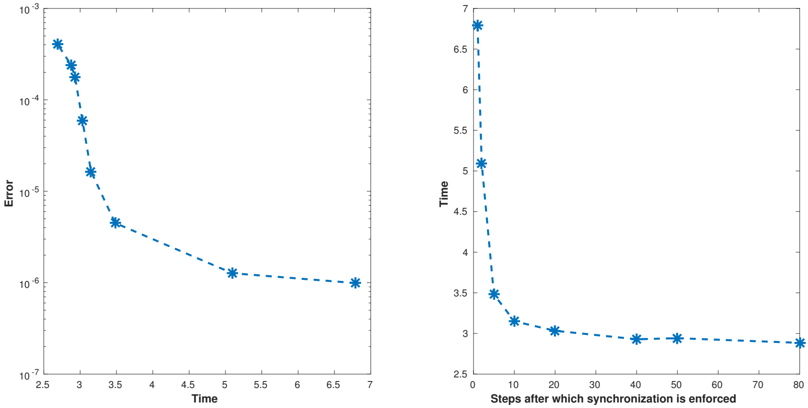
Deterministic Asynchronous Scheme

- Error: Deterministic.
- Exchange the information after a certain amount of steps. (SYNC_STEP)
- Trade-off between time and error.
- Larger the value of SYNC_STEP, lesser will be time, but more will be the error.

Test Case

- Number of Grid Points = 4096.
- Number of Processors = 8
- Courant Number = 0.1
- Final t = 0.08*len/c
- Number of Steps = 2135099

4096 grid on 8 processors



Observations

- Trade-off between accuracy and Error.
- In the starting the time falls rapidly and becomes approximately constant after a certain stage.

Test Case

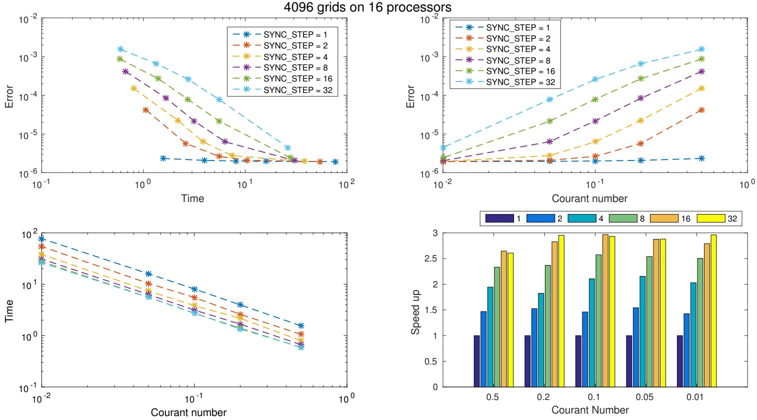
- Number of Grid Points = 4096 and 16384
- Final t = 0.08*len/c
- Study the effect of Courant Number in case of Synchronous and Asynchronous Scheme.
- Measure the Speedup in case of relaxed Synchronization.

Speedup

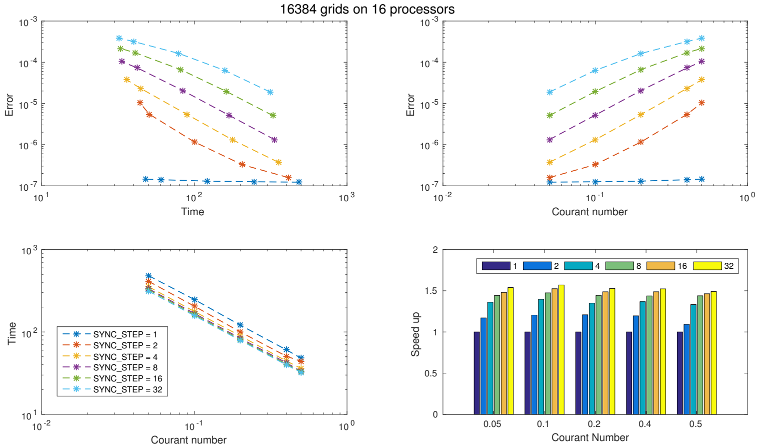
Speedup can be defined as the ratio of time taken by synchronous scheme to the time taken by the time taken under relaxed synchronization.

$$Speedup = \frac{\text{Time taken by Synchronous Scheme}}{\text{Time taken by scheme under relaxed synchronization}} \tag{49}$$

4096 grid on 16 processor



16384 grid on 16 processor



- Computation Cost: X
- Synchronization cost/ Synchronization : Y
- Number of Time Steps: N_T
- Total Number of Synchronization: $\frac{N_T}{SYNC\_STEP}$
- Speedup = $\frac{X+N_T Y}{X+\frac{N_T}{SYNC\_STEP} Y}$

Observations

- Synchronous Scheme is bounded by $\mathscr{O}(\Delta x^2)$.
- Higher Value of SYNC_STEP $\Rightarrow$ More Speedup
- More load on the processor $\Rightarrow$ Less Speedup

- Do not wait for the communication to complete.
- Use the latest time values.
- Two Sided Non - blocking
    - Achieved using MPI_Isend / MPI_Irecv / MPI_Test
- One Sided RMA
    - Achieved using MPI_Put / MPI_Lock / MPI_Unlock

Delay statistics

The statistics of delay is measured in terms of k^* which is defined as:

$$k^* = \sum_{i=0}^{i=\infty} i * k_i \tag{50}$$

where k_i is defined as the ratio of the number of Time Steps that faced Delay = i and total number of Time Steps.

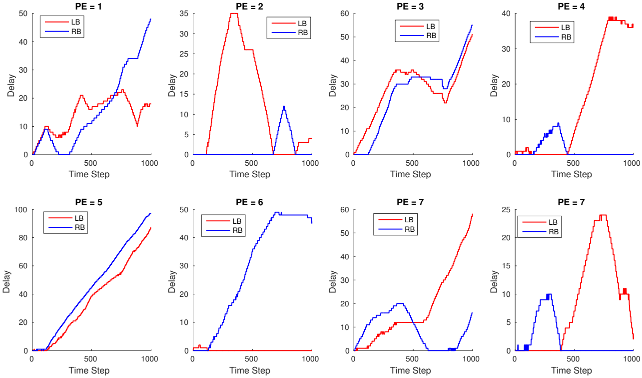


Figure : Evolution of Delay for 1024^3 grid for 1000 time steps. Each process has two boundary point. LB denotes the delay faced by left boundary point and RB the right boundary point

Two Sided Non - blocking: PE = 8

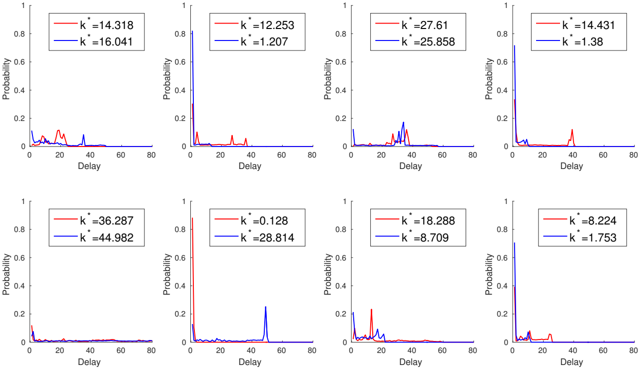


Figure : Delay Distribution for 1024^3 grid for 1000 time steps

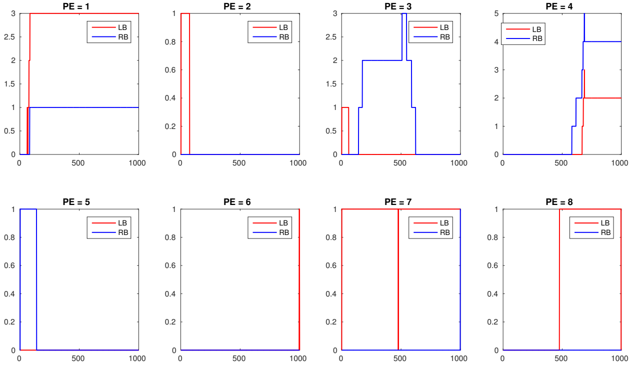


Figure : Evolution of Delay for 1024^3 grid for 1000 time steps. Each process has two boundary point. LB denotes the delay faced by left boundary point and RB the right boundary point

One - Sided RMA: PE = 8

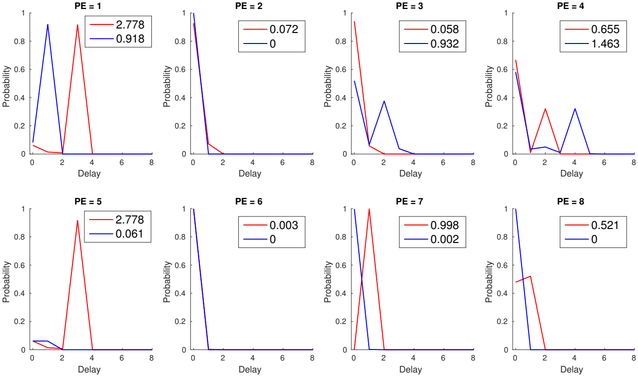


Figure : Delay Distribution for 1024^3 grid for 1000 time steps

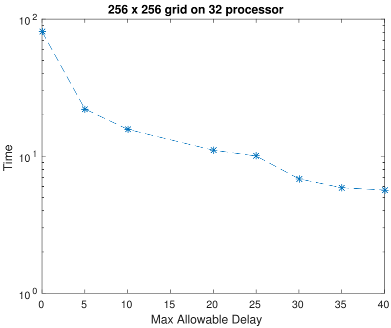
Stochastic Asynchronous Scheme

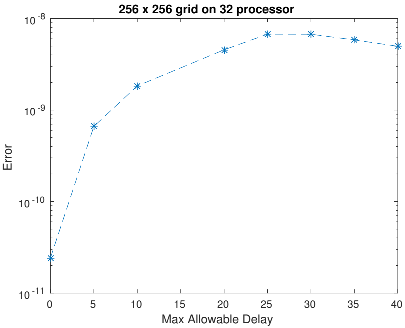
- Need an error control knob.
- $\langle \overline{E} \rangle \propto \overline{\tilde{k}}$
- Enforce partial/total synchronization when k = L [7]

[7] L represents the maximum allowable delay.

Test Case

- Number of Grid Points = 65536.
- Number of Processors = 32
- Courant Number = 0.1
- Final t = 0.08*len/c
- Study effect of Maximum Allowable Delay on Error and Time


256 x 256 grid on 32 processor
Time
10^2
10^1
10^0
0
5
10
15
20
25
30
35
40
Max Allowable Delay


256 x 256 grid on 32 processor
Error
10^{-8}
10^{-9}
10^{-10}
10^{-11}
0
5
10
15
20
25
30
35
40
Max Allowable Delay

Observations

- Trade-off between Error and Accuracy.
- More the Maximum Allowable Delay $\Rightarrow$ Less Time $\Rightarrow$ More Error.

- Nodes
  - Called "host", "computer", "machine"
  - A certain amount of memory (RAM) is physically allocated on each node
  - Each node contains multiple sockets.
- Sockets
  - Collection of cores with a direct pipe to memory
  - Contains multiple cores.
- Cores
  - Single processing unit capable of performing computations.
  - Smallest unit of resource allocation.

Test Case

- Number of Grid Points = 1024.
- Number of Processors = 8
- Courant Number = 0.1
- Number of Time Steps = 150000
- Communication: RMA
- Study effect of Process Distribution on Various Nodes on Time and Delay Statistics

Processor per node	Minimum Time	Maximum Time	Average Time	Maximum Delay	Average Delay
8	1.632993	1.898601	1.747674	26729	1152.118219
4	1.306848	2.515238	1.935257	63719	595.315498
2	2.310587	2.394653	2.357168	5074	23.086279
1	2.575294	2.636949	2.604869	1912	49.434085

Test Case

- Number of Grid Points = 268435456.
- Number of Processors = 8
- Courant Number = 0.1
- Number of Time Steps = 500
- Communication: RMA and Two Sided Non-blocking
- Study effect of Process Distribution on Various Nodes on Time and Delay Statistics

RMA operation

Processor per node	Minimum Time	Maximum Time	Average Time	Maximum Delay	Average Delay
8	196.022283	210.897438	204.288266	38	0.152375
4	157.093253	168.151056	162.511951	42	0.013750
2	112.703603	126.833151	122.102558	44	0.075125
1	91.371902	96.440170	93.667287	20	0.132000

Two Sided Non blocking Send-Receive

Processor per node	Minimum Time	Maximum Time	Average Time	Maximum Delay	Average Delay
8	143.154700	158.406454	151.922679	65	10.797625
4	95.791654	137.700051	112.257639	123	20.124625
2	85.375067	141.165011	111.381462	137	17.850750
1	68.627403	105.315875	84.915095	110	13.006375

Observations

- Distribution of Processor on different nodes $\Rightarrow$ Dependent on Memory Requirement.[8]
- Delay Statistics: Random
- Increasing load $\Rightarrow$ Lower value of Average Delays.
- RMA operation $\Rightarrow$ more favourable for Asynchronous Operation.

[8] MPI for Big Data: Dominique LaSalle, Parallel Computing, 2014

Conclusion

- Original Scheme:
  - Second Order Convergence without delays.
  - First Order Convergence in presence of delays.
- Second Order Convergence in presence of delays.
  - Reducing Δt
  - Newer Scheme
- Fewer Communication calls $\Rightarrow$ More speedup.
- Higher Load on processor $\Rightarrow$ Communication cost less significant.
- Study of Delay Statistics.
  - MPI_RMA: better for asynchronous case in terms of average delay statistics.
  - Dependence of load on the processor. More frequent communication $\Rightarrow$ Larger Delays.
  - Impact of Computer Architecture.

Future Work

- Development of Consistent Scheme:
  - Independent of Delay Statistics.
  - Second Order Convergence in space.
- Extension in multi-dimensional case.

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.

Thank You for your attention!!!