Implementation of Parareal algorithm

Group 25
Gursimran Singh Saluja
Abdullah Mujahid
Basanagouda Somanakatti

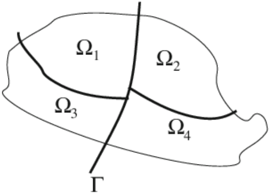
Dr. Karlheinz Lehner
Chair of Computing in Engineering

Computational Engineering
Ruhr Universität Bochum

Parallelization in PDE solvers

Methods

- Spatial domain decomposition
- Time domain decomposition



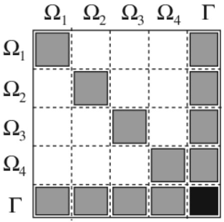


Figure: Spatial Domain Decompostion Illustration [1]

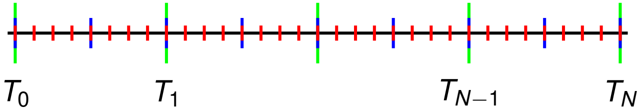
Parallelization in time - Motivation

$$\frac{\partial y(t)}{\partial t} + \mathcal{A}y(t) = 0, \;\; t \in [T_0, T_N]$$
$$y(T_0) = y_0$$

Run-time Consideration

- c : Time for computing one time step in numerical integration
- $T = M\,c$, total time for full simulation, M : Number of fine steps
- For $c = 0.04s$, $M = 1000,000$, $T = 40,000s \approx 11.1hrs$

Parallelization in time - Idea



Idea

- Split the time domain into a set of subdomains
- Solve the IVP on a coarse grid to get an approx. sol at $T_1, T_2, \ldots, T_{N-1}, T_N$
- This can be used as initial value for the IVP on each sub-domain
- Initial value is only serves as guess (coarse sol)
- Iteratively better solution with better initial guess for initial value in each iteration

Parareal Algorithm

Convention

- δt_c : coarse grid time step
- δt_f : fine grid time step
- $\mathcal{G}$: coarse grid integrator
- $\mathcal{F}$: fine grid integrator
- $Y_0, Y_1, \ldots Y_N$: numerical solution
- Yf_n^k, Yc_n^k : Fine and Coarse Integrator solutions

Construction

$$Y_n^k = \mathcal{F}(Yc_{n-1}^{k-1}) + \mathcal{G}(Y_{n-1}^k) - \mathcal{G}(Y_{n-1}^{k-1})$$

where

- k : iteration
- n : grid index

Parareal Algorithm

Algorithm 1 Parareal Algorithm

Iteration 0: $Y_0^0 = y_0$, given initial value
for $n = 1$ to N **do**
    $Y_n^0 = \mathcal{G}(Y_{n-1}^0)$
end
for $k = 1$ to N **do**
    parallel for $n = 1$ to N **do**
        $Yf_n^{k-1} = \mathcal{F}(Yc_{n-1}^{k-1})$
    end
    for $n = 1$ to N **do**
        $Yc_n^k = \mathcal{G}(Y_{n-1}^{k-1})$
    end
    $Y_n^k = Yc_n^k + Yf_n^{k-1} - Yc_n^{k-1}$
    if $|Y^k - Y^{k-1}| < \epsilon$ **then**
        break
    end if
end

Test problem

Problem 1

$$\frac{dy}{dt} = A(t)\, y(t)$$
$$y(0) = y_0, \quad t \in [0, 10]$$

where $A = -0.2$, $y_0 = 100$

Problem 2

$$\frac{dy}{dt} = \sin(t)\, y(t) + t$$
$$y(0) = 1, \quad t \in [0, 14]$$

Numerical Method

Classical Runge-Kutta scheme of 2^{nd} order is used for 1D ODE.

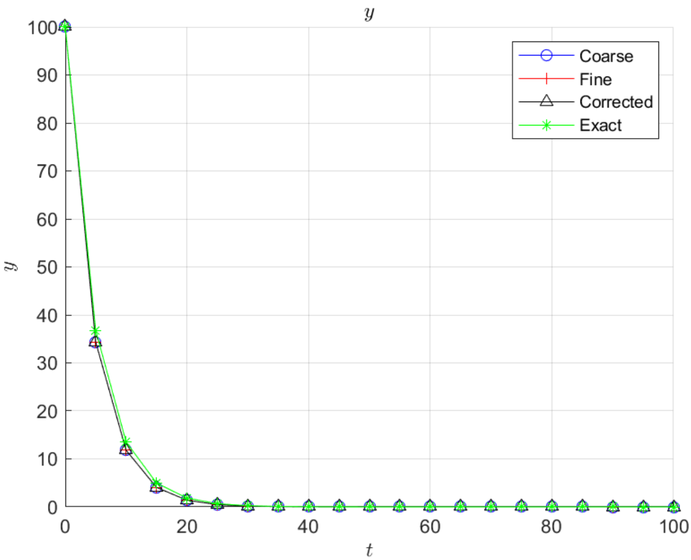
$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1) \\
k &= \frac{k_1 + k_2}{2} \\
y_{n+1} &= y_n + h\, k
\end{aligned}
$$

OpenMP implementation

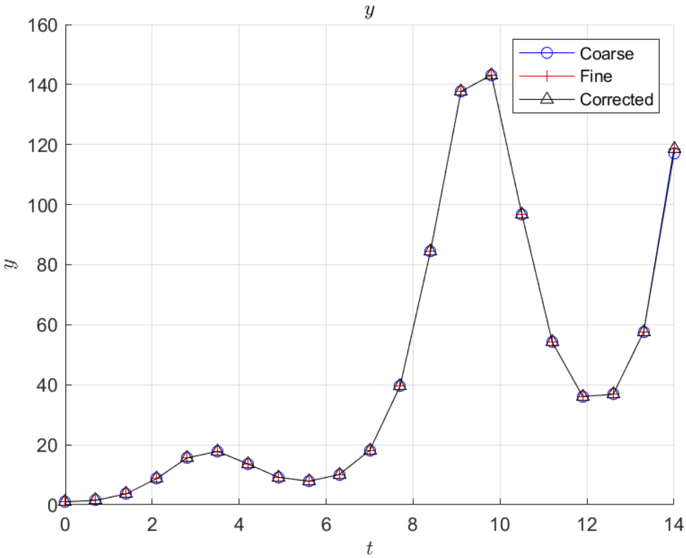
parareal.cpp > ...

```cpp
#include <stdio.h>
#include <omp.h>
#include <math.h>
// global variables
int main()
{
    // coarse grid integrator
    // ...
    while (err > tol and iter < max_iter)
    {
    #pragma omp parallel num_threads(NUM_THREADS) shared(t, y) private(i)
        {
        #pragma omp for schedule(dynamic, chunk)
            for (int i = 0; i < n_sub; i++)
            {
                //fine integrator
                //...
            }
        }
        // predictor: coarse integrator
        // corrector
        // error check
    }
    return 0;
}
```

Simulation : Problem 1



Simulation : Problem 2



Speedup

CPU	Cores	Multithreading	Num of threads	time serial	time parallel
i5 9500	6 cores	6 way	100	201.222	117.047
i5 8250u	4 cores	8 way	100	201.380	117.070
i3 5010u	2 cores	4 way	100	201.251	117.054

Speedup
1.72
1.72
1.72

Grasedyck, Lars and Kriemann, Ronald and Le Borne, Sabine
Domain decomposition based $\mathcal{H}$-LU preconditioning,
Volume 112, Numerische Mathematik, May 2009.

Allan S. Nielsen
Feasibility study of the parareal algorithm
IMM-MSc-2012-nr.134, Technical University of Denmark