Report: Application of the Parareal Algorithm to SOLPS

Debasmita Samaddar[1], David Coster[2], Xavier Bonnin[3], Christoph Bergmeister[1], Eva Havlickova[1], Wael R. Elwasif[4], Lee A. Berry[4], Donald B. Batchelor[4]

1 – EURATOM/CCFE, Culham Science Centre, UK

2 - Max-Planck-Institut für Plasmaphysik, Germany

3 – CNRS-LIMHP, Université Paris 13, France

4 – ORNL, USA

December, 2014

"Take Home Message"

•Parareal algorithm parallelizes the time domain - is an innovative technique that may be applied for parallelization to achieve computational speedup.

ALGORITHM SHOWN TO WORK FOR EDGE PHYSICS CODE - SOLPS, FOR TOKAMAK SIMULATIONS.

Outline

- Motivation
- Overview of algorithm
- SOLPS results
- Frameworks and parareal scheme
- Conclusion

Motivation

- Simulations of fusion plasma are numerically very challenging. SOLPS with B2-Eirene is a good example!
- Space parallelization is not enough.
- Is time parallelization an option? Well, parareal algorithm has helped in achieving significant speedup in cases already studied.

Parareal Algorithm -
a quick overview

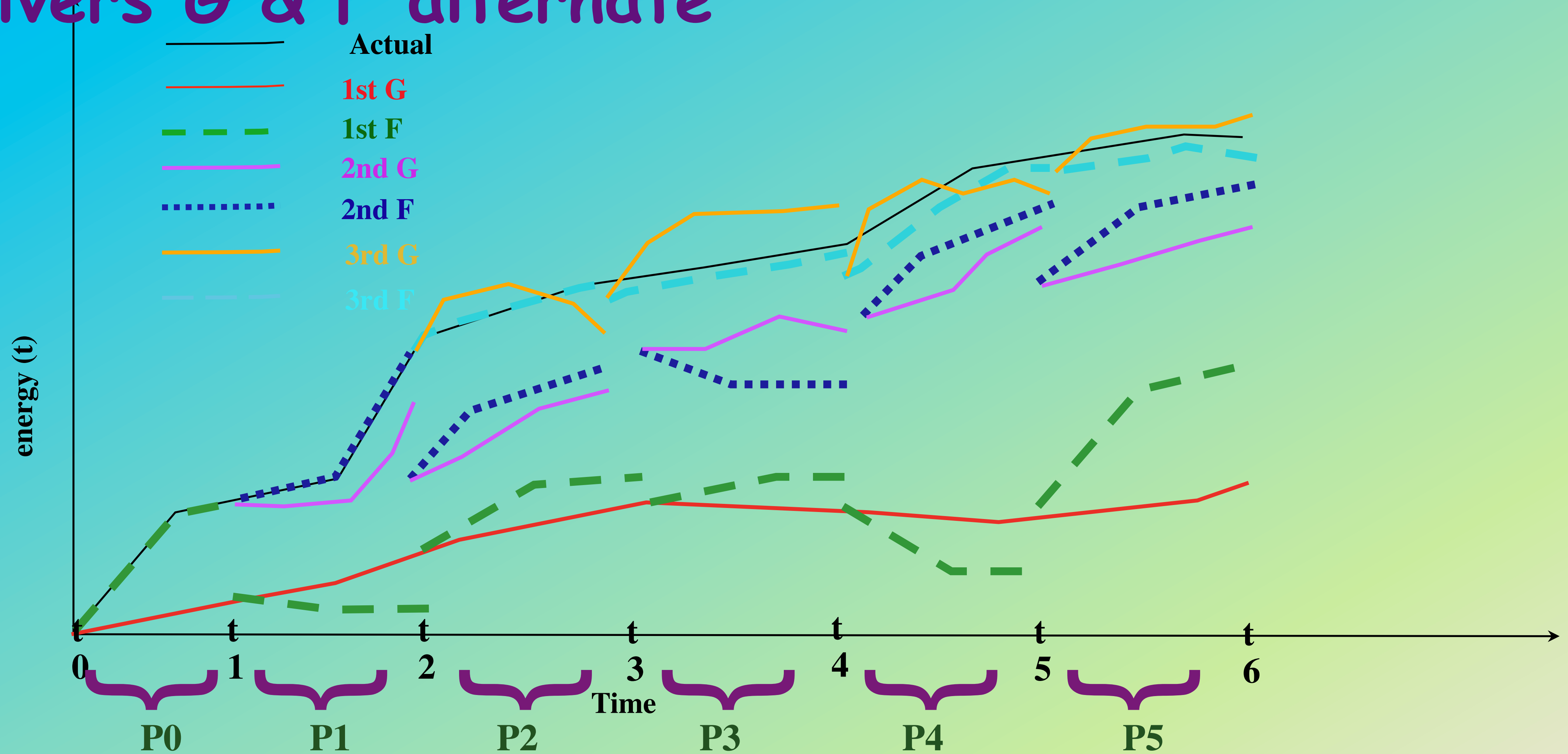
Parareal Algorithm : Distinct in many ways

- Algorithm first proposed by Lions et al. in 2001.
- Parallelizes in time, despite the sequential nature of the time domain.
- Very non-intuitive as this is an initial value problem, and the result of each time step should depend on that of the previous timestep. However, in this case, "timesteps" (chunks) are solved in parallel.
- Uses predictor - corrector approach.

F is a propagator evolving the function (energy(t)) from initial time, t_0, to a later time ...

G - faster but inaccurate propagator

Solvers G & F alternate



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

New G

Old G

Success of Algorithm Depends on Multiple Factors

Algorithm always converges if k=N.

But, success in achieving significant speedup if

- k<<N.

G is much cheaper than F.

- "Good" G: Solutions converge
- "Bad" G: Solutions diverge

- No "fixed recipe" for G !

- Despite solutions being very sensitive to initial conditions for - it is possible to choose G.

Selecting Optimum Coarse Solver is Important

Different approaches can/should be explored to find G. Any one of them, or a combination of them, may work :

- Some of the physics may be ignored when solving with G, to achieve speedup.
- G can be same as F, but may be solved over a coarser k-mesh (or spatial grid).
- G may be same as F, but may be solved with a larger timestep (dt) and less accuracy.
- Use a different G.

Results of Application to SOLPS :

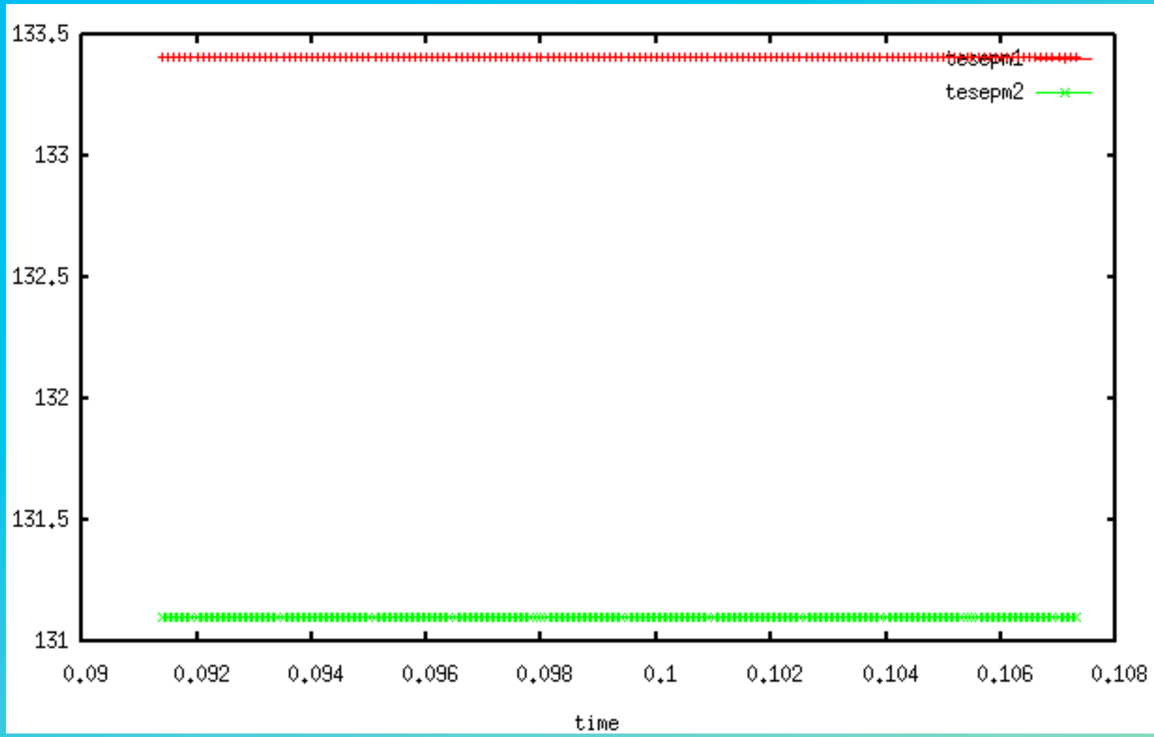
Scrape Off Layer Plasma Simulator

Parareal application: features

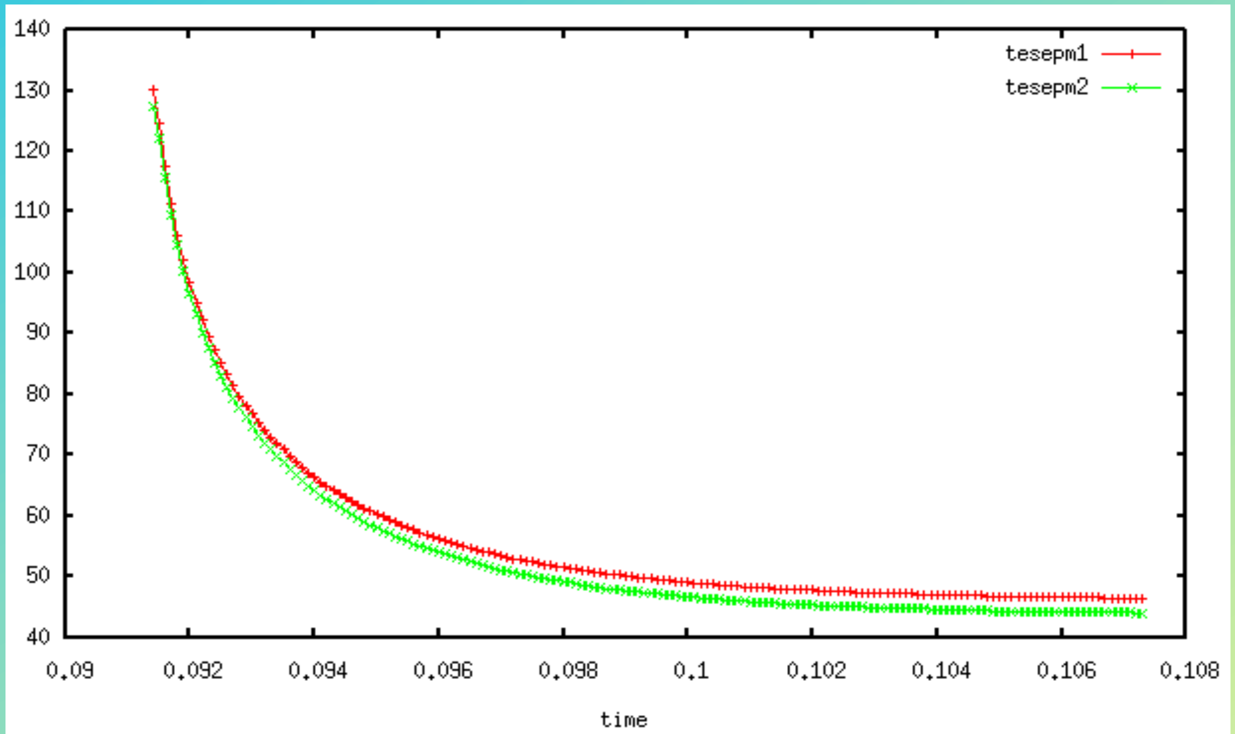
- Parareal convergence based on pwmxip and pwmxap (maximum total power fluxes inboard & outboard divertor, respectively).
- Parareal correction to: na, ne, te, ti, ua and po (the primary variables of the code).
- Eirene uses Monte Carlo treatment of neutral particle transport solving Boltzman equation for distribution functions for neutrals.

Results –
G or coarse solver: Replace Eirene with fluid neutrals model (faster computation):

Results: Electron temperature at separatrix

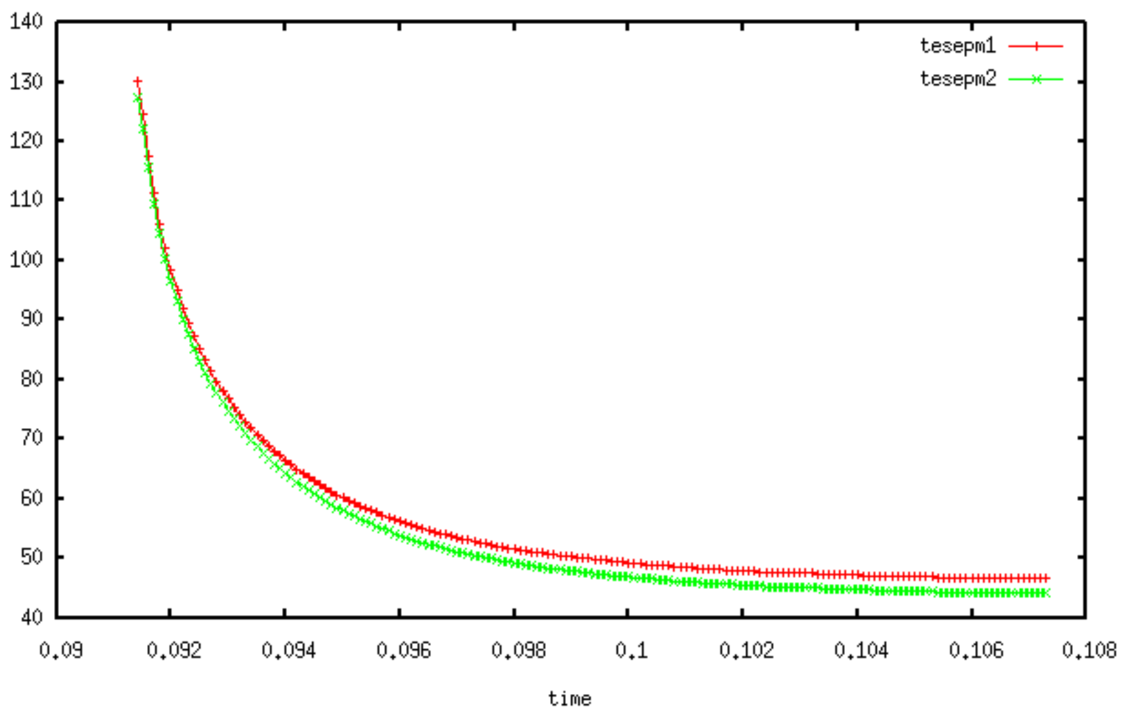


Coarse estimate



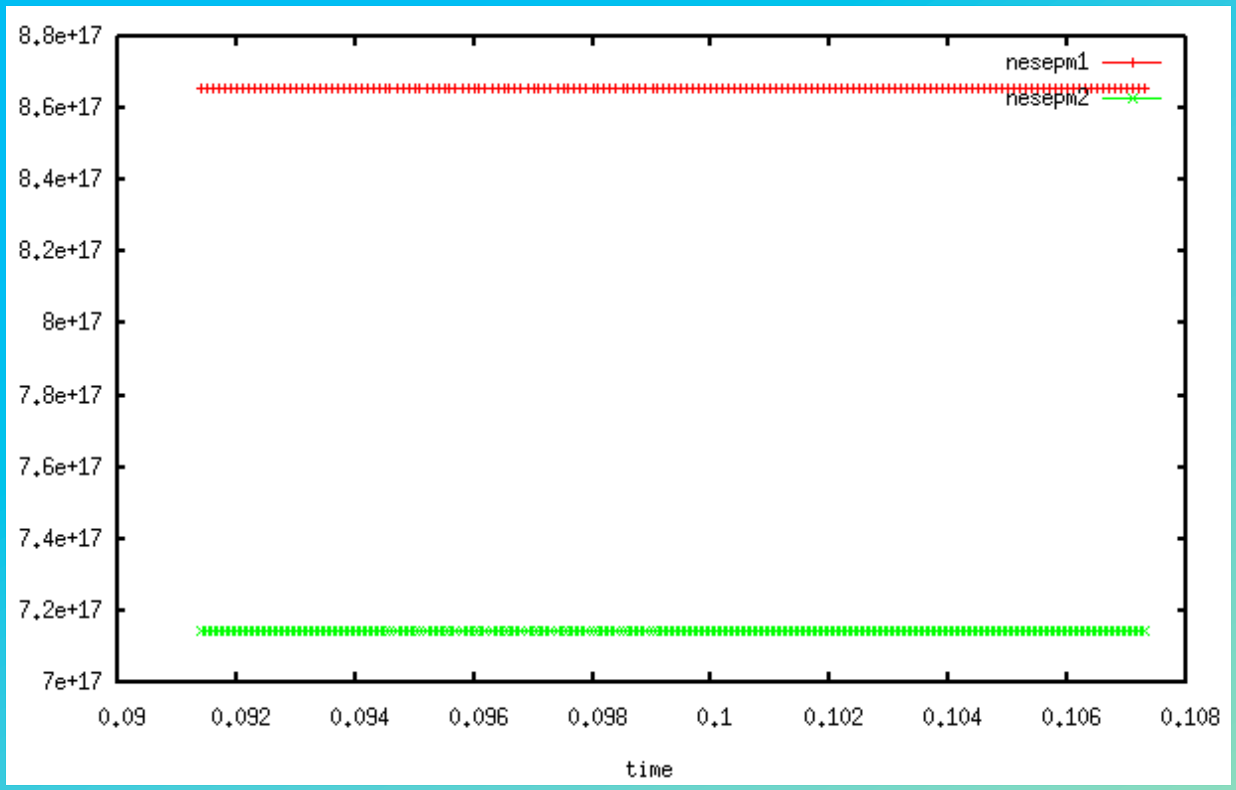
Fine (serial) solution

Parareal solution

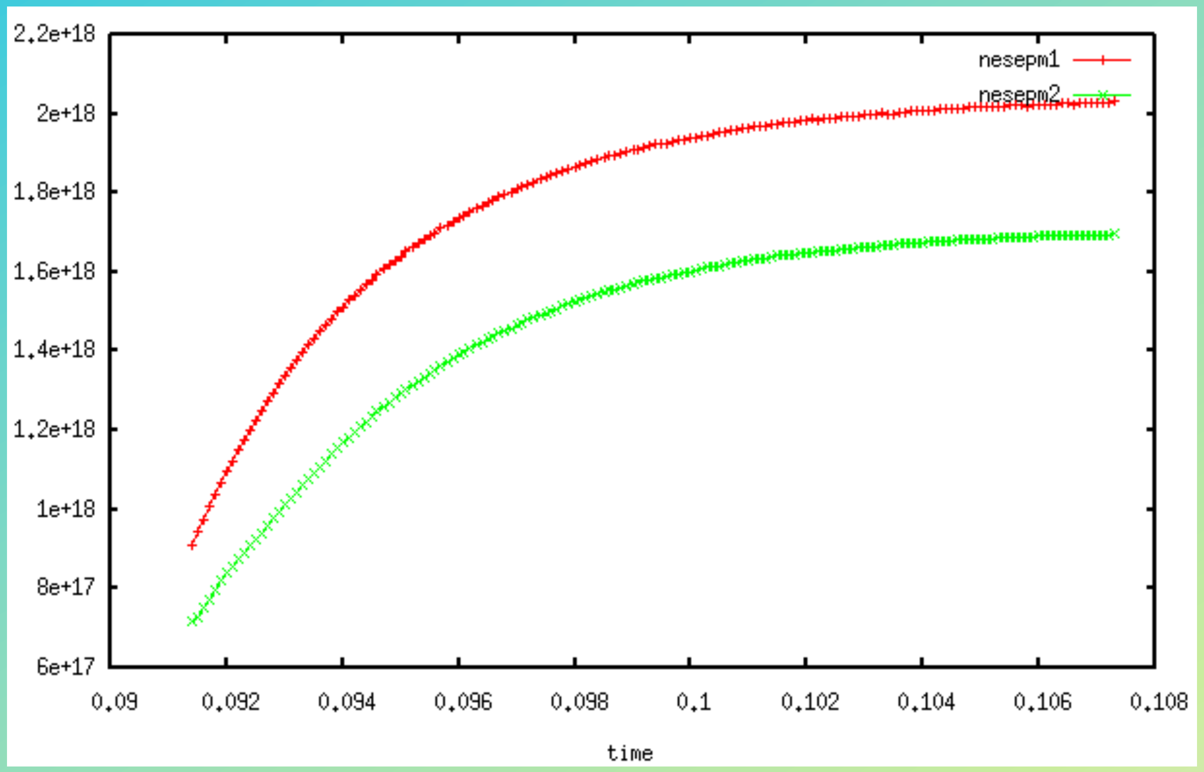


Conclusion: The parareal solution matches the serial solution.

Results: Electron density at separatrix

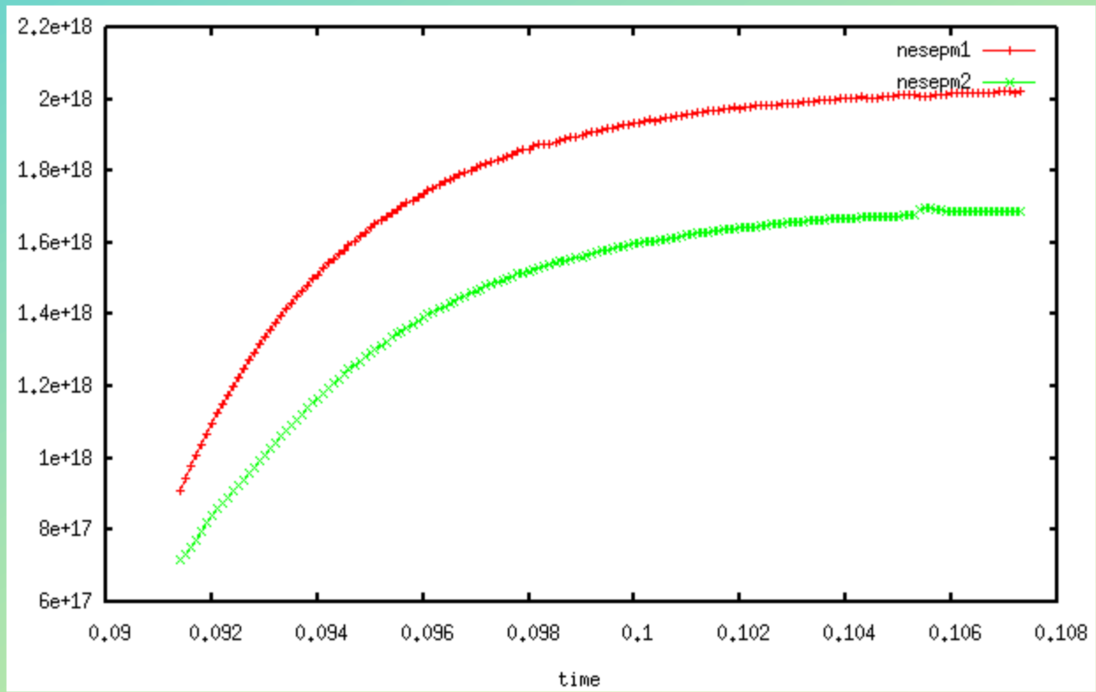


Coarse estimate



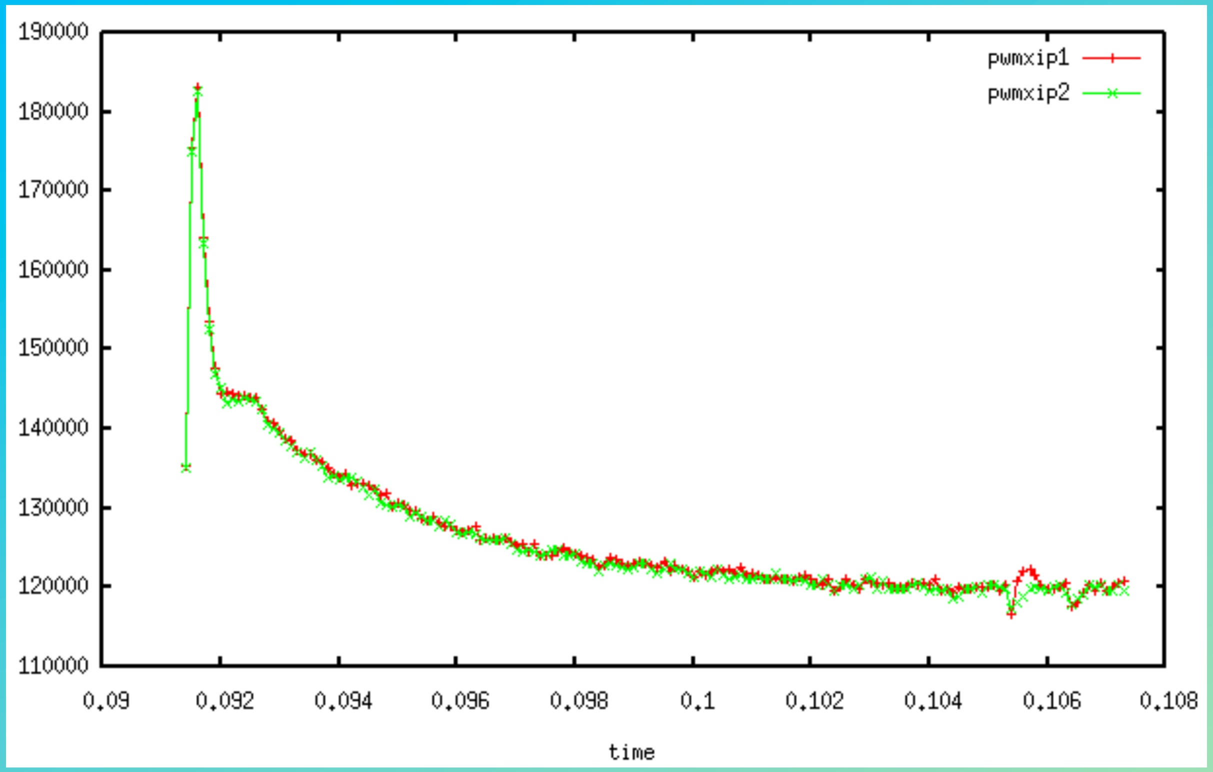
Fine (serial) solution

Parareal solution

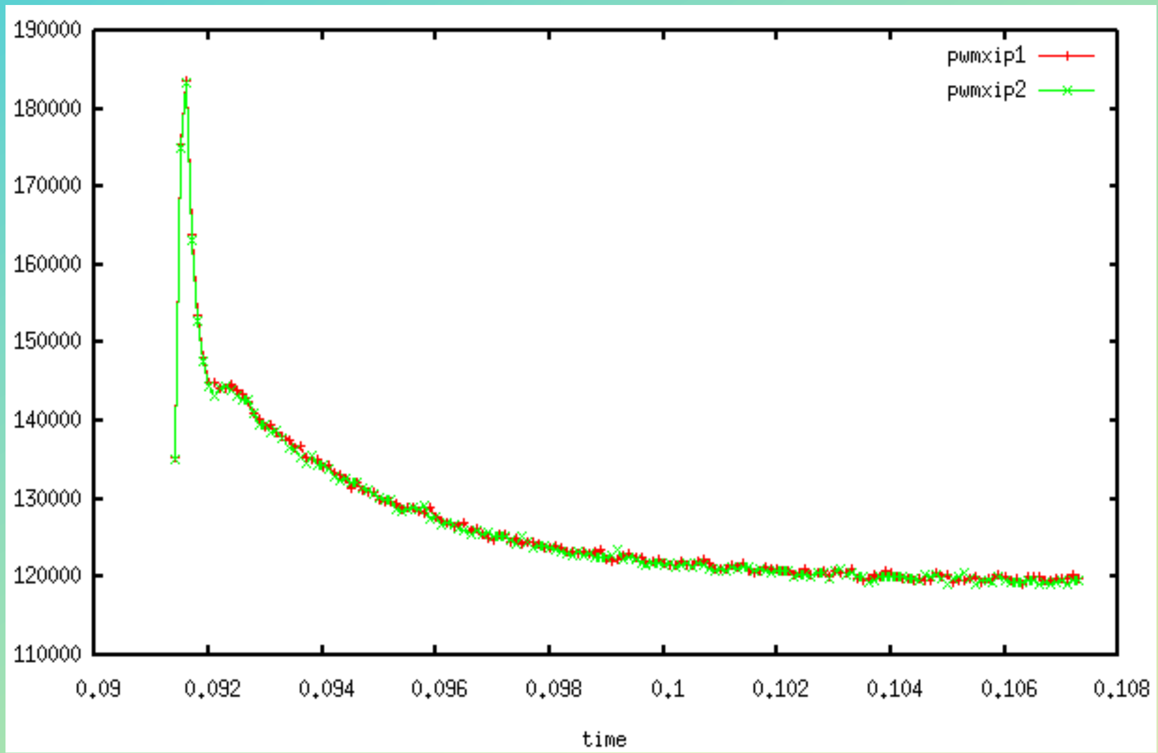


Conclusion: The parareal solution matches the serial solution.

Results: Maximum flux of the total power inboard the divertor :



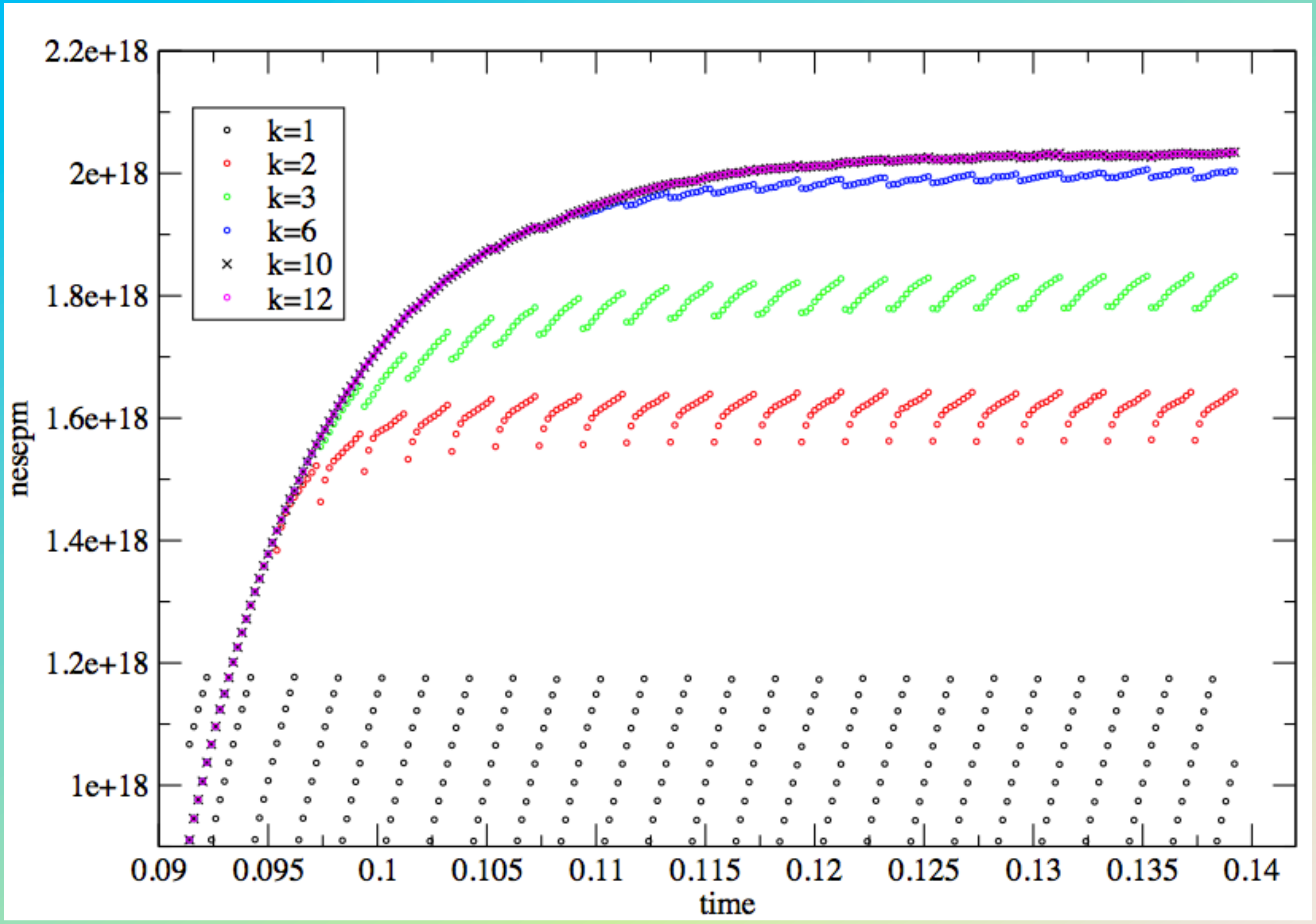
Fine (serial) solution



Parareal solution

Computational gain = 12.58 with 240 processors (may increase with processors!)

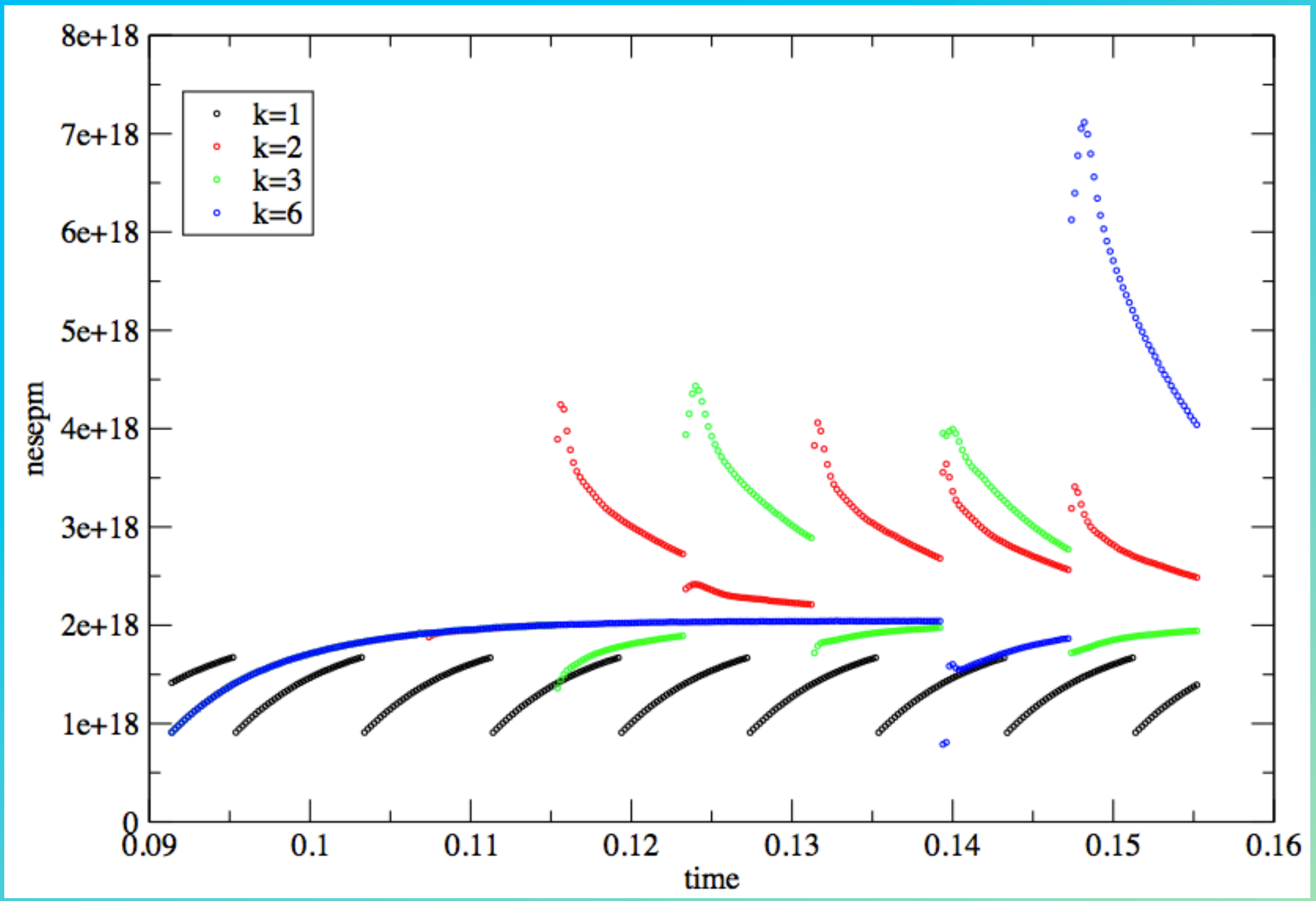
Parareal works perfectly with timeslice per processors =10



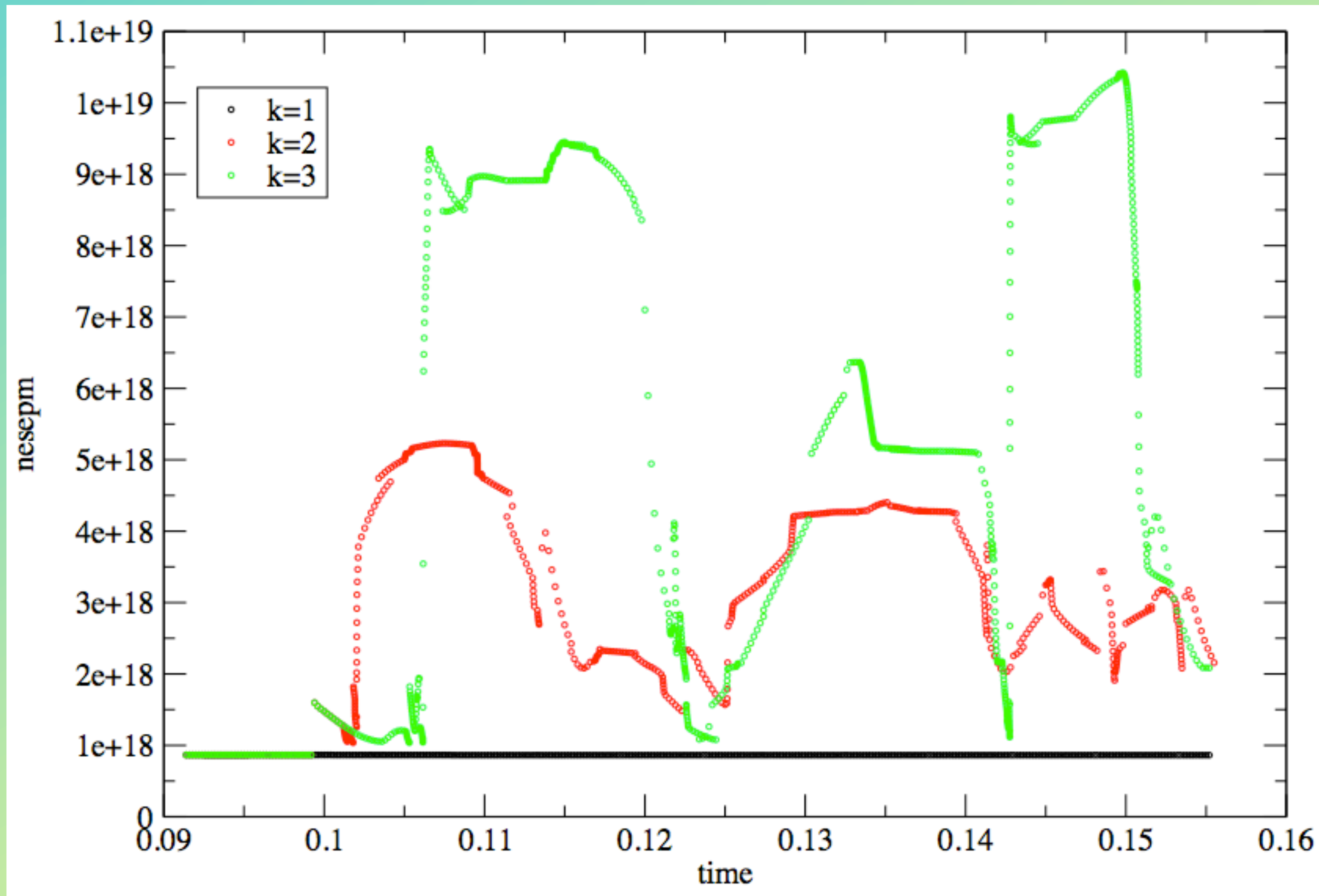
Solutions converge with increasing k.

UNIVERSITÉ PARIS 13 NORD
EFDA
CCFE CULHAM CENTRE FOR FUSION ENERGY
IPP
Fusion Energy Division
OAK RIDGE NATIONAL LABORATORY
Managed by UT-Battelle for the Department of Energy

Parareal fails with timeslice per processors >10



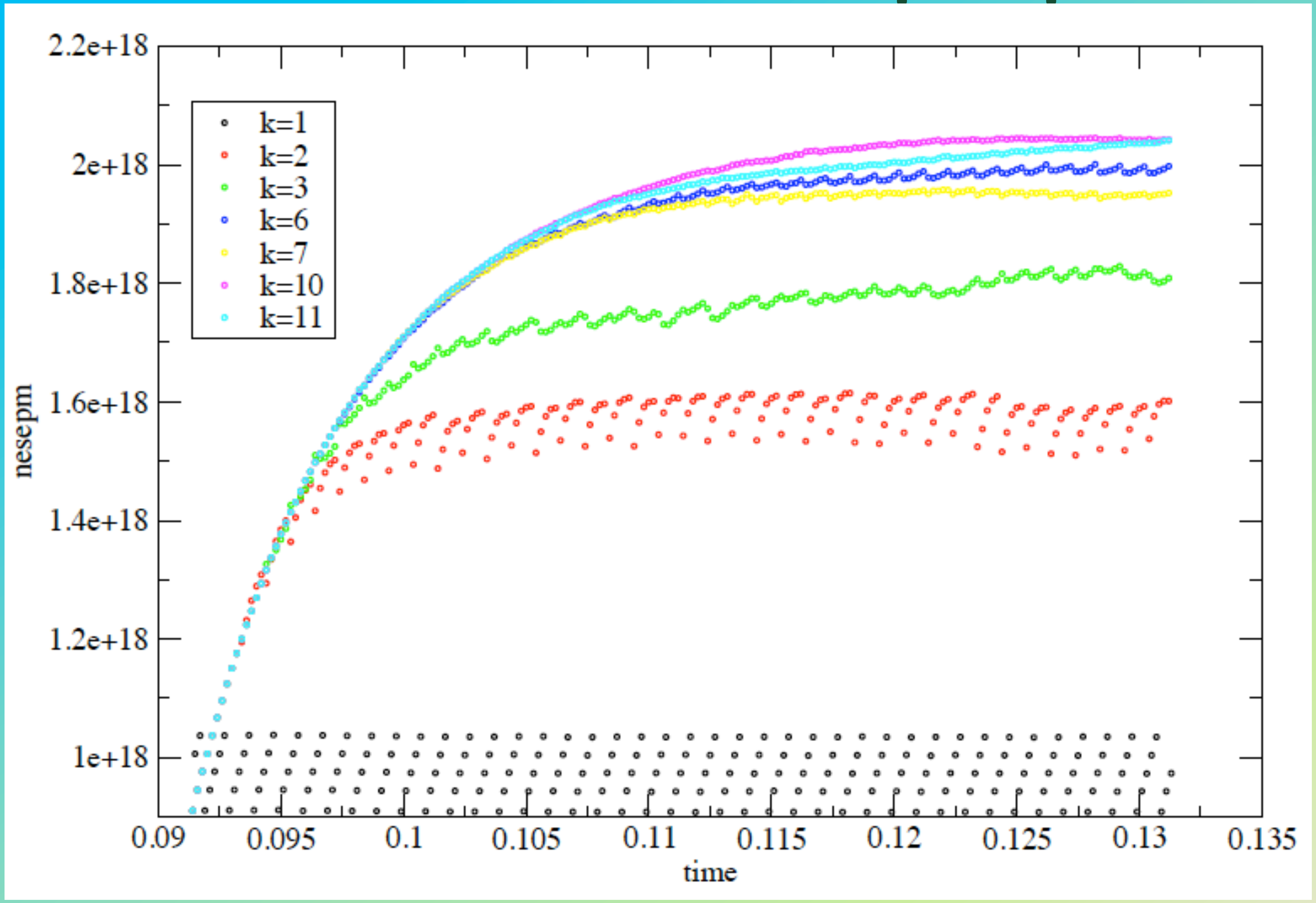
Fine solution for different k, with timeslice=40



Coarse estimate

Coarse estimate deviates too far from fine solution?

Parareal fails with timeslice per processors <10



Fine solution for different k, with timeslice=5

Fine solution not allowed to evolve enough?

Results –

G or coarse solver: Reduced grid (2 studies: MAST & DIIID):

Can experience gathered with previous cases be helpful now?

Parareal converges with varying coarse grid sizes

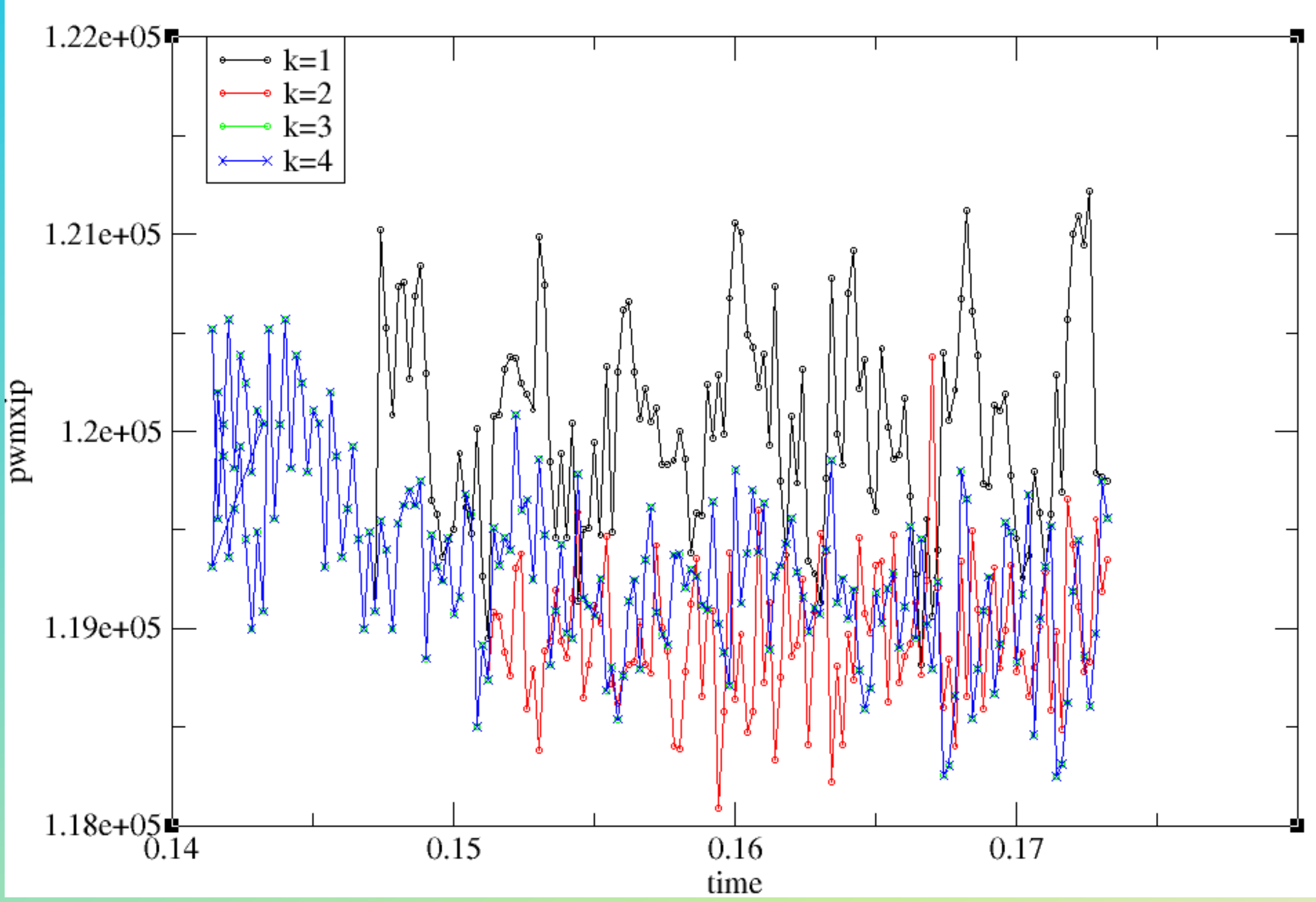
a)Fine grid 150X36: MAST.

Coarse grids: 150X18, 76X36, 76X18

b)Fine grid 96X36: DIIID Coarse grids: 48X36, 32X36

CFL condition allows bigger dt with reduced grid sizes.

Fine solution for different k, with fine timeslice=20:

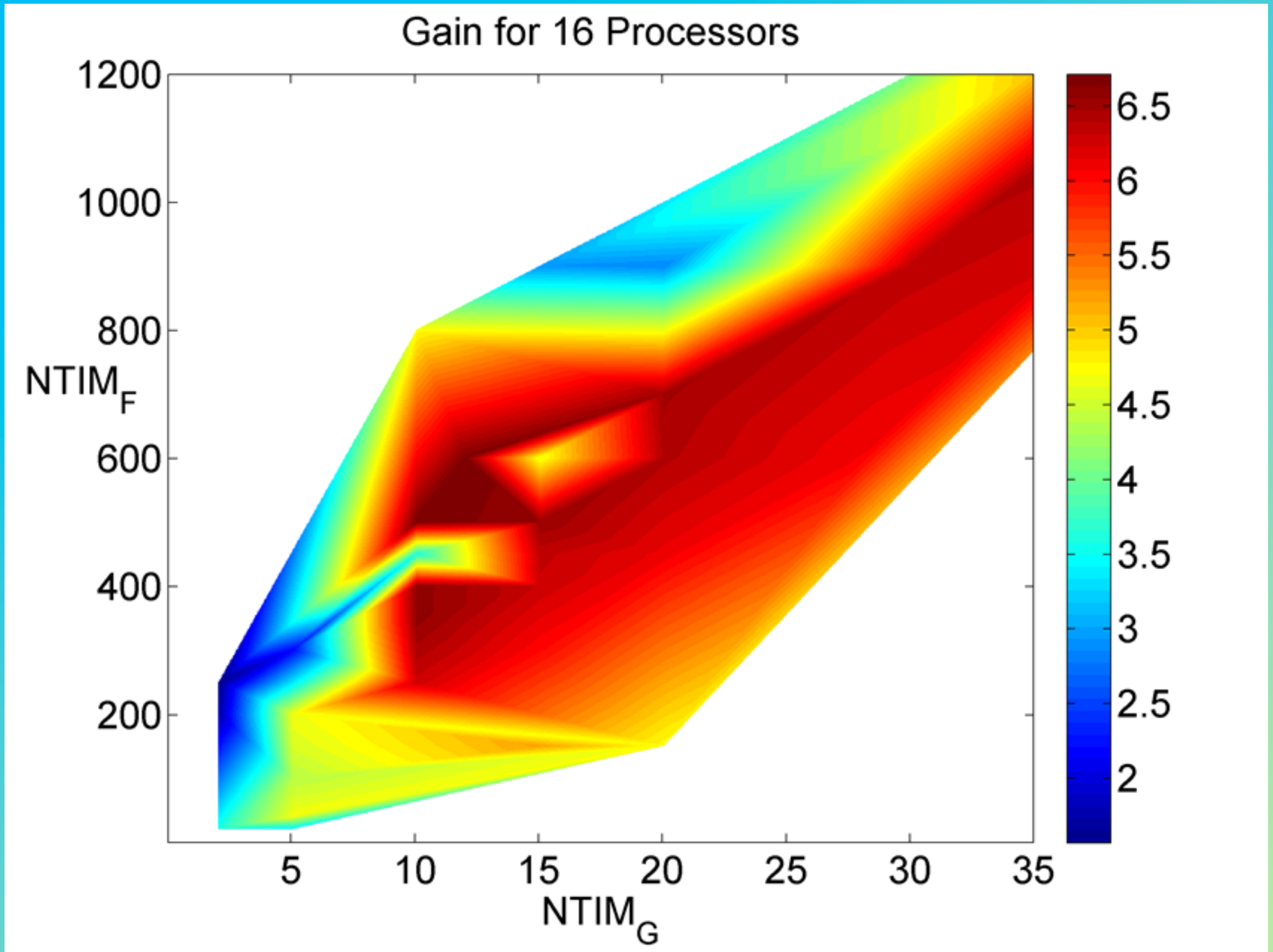


Reduced grid: 150X18 Gain=4.9 with 32 processors.

Convergence at k = 4

dt_g = 10dt_f

Parareal convergence & gain depend on size of time slice per processor - DIIID



DIIID case :

$NTIM_F$ = # of Fine timesteps.

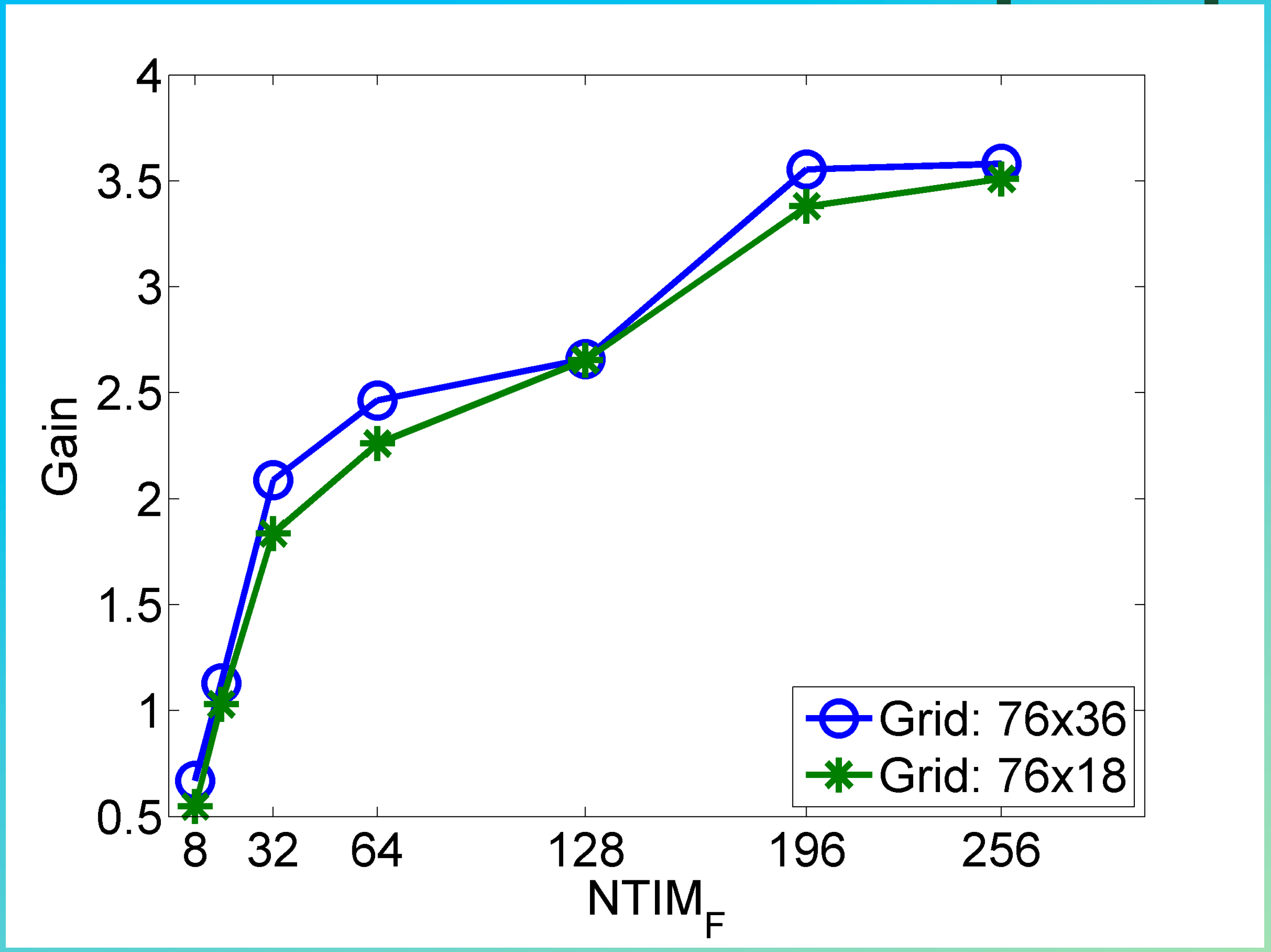
$NTIM_G$ = # of Coarse timesteps.

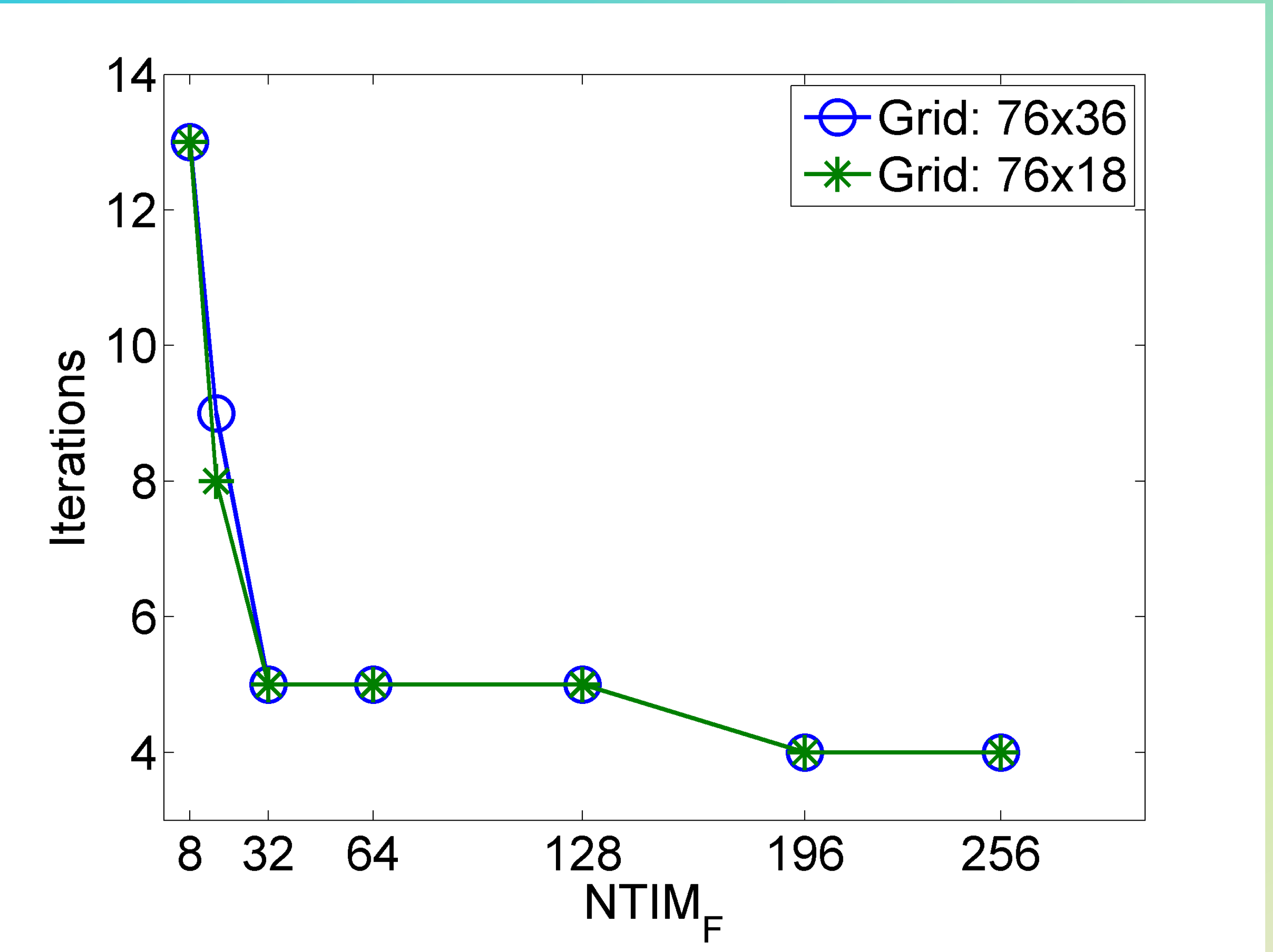
Gain improves with increasing $NTIM_F$ for same $NTIM_G$

Fine grid: 96X36, Reduced grid: 48X36, dtG=30dtF

Gain=21.8 with 96 processors.

Parareal convergence & gain depend on size of time slice per processor - MAST





MAST case :

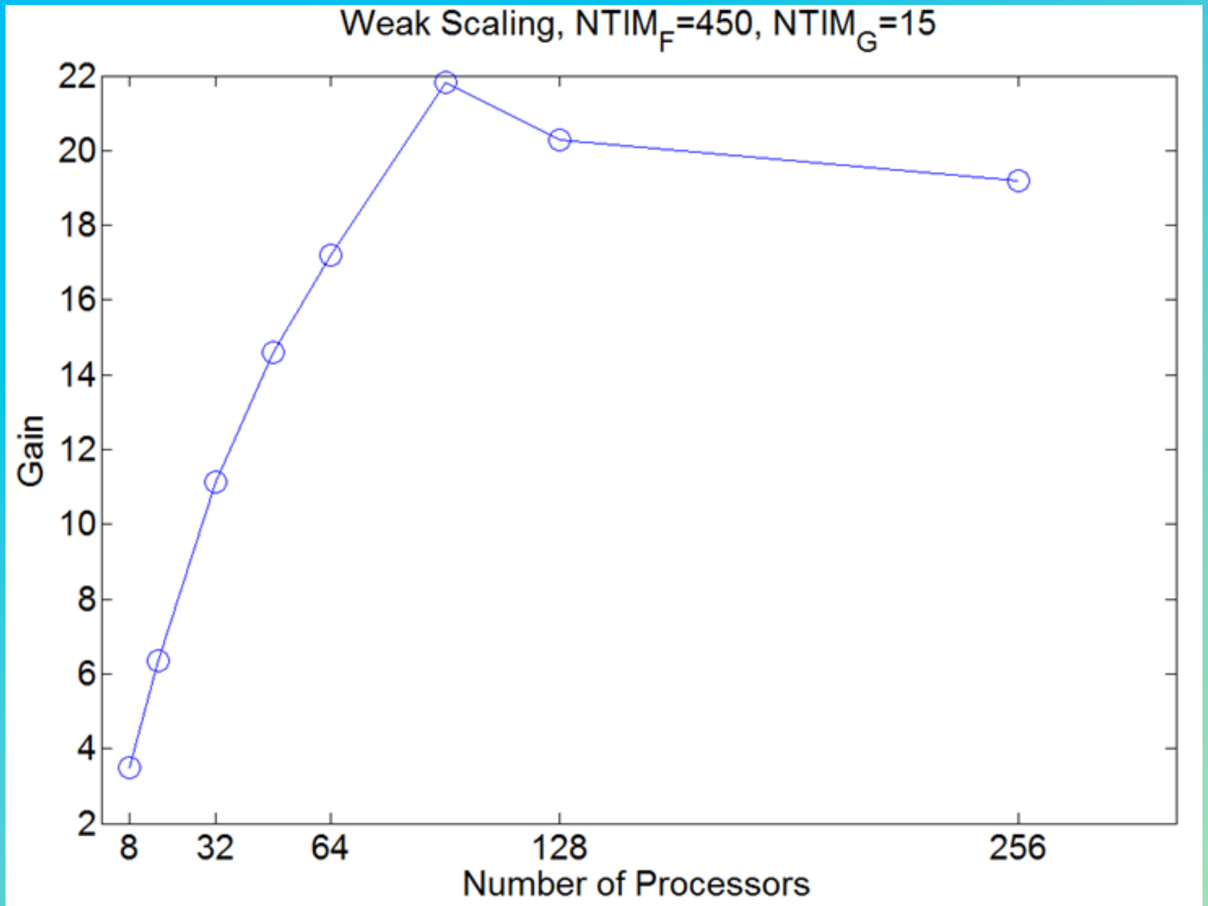
Gain improves with increasing NTIM$_F$ for same NTIM$_G$

Fine grid: 150X36, Reduced grid: 150X18, dtG=32dtF, dtF=256

Gain=15.9 with 64 processors.

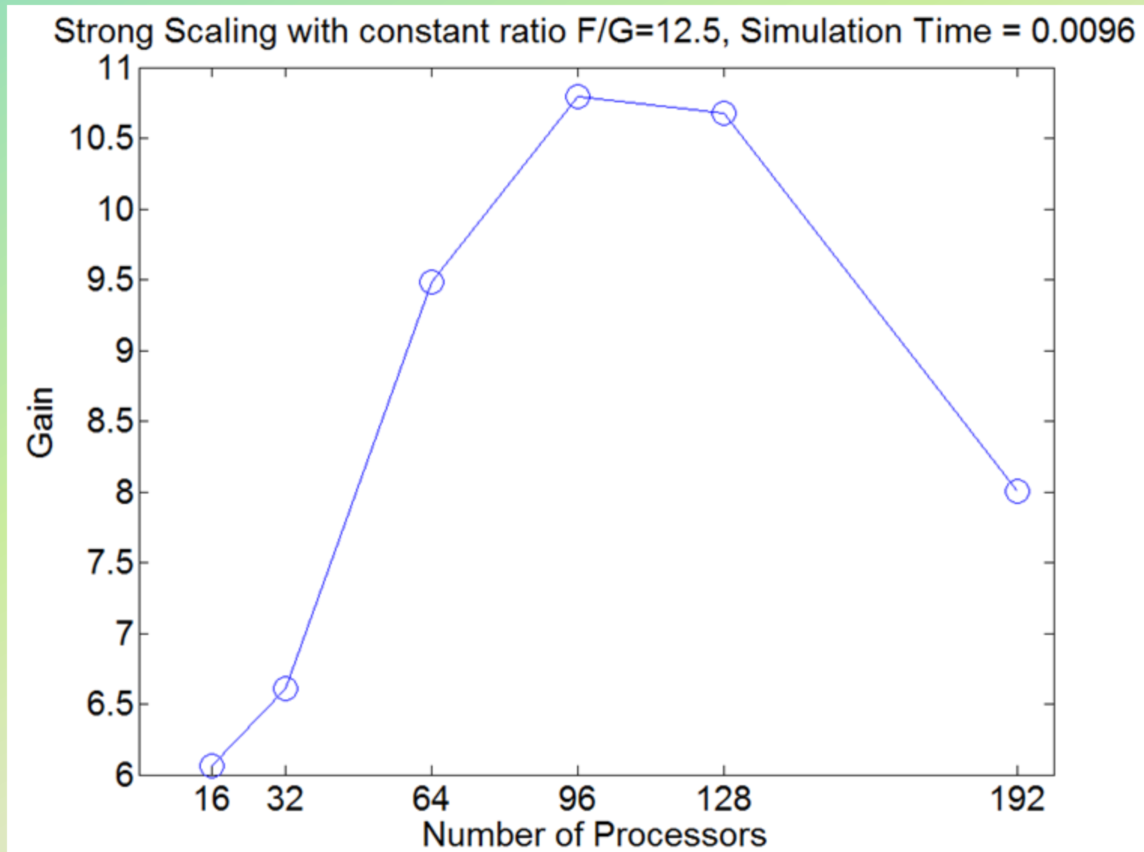
Computational gain may be optimized by scaling studies

DIIID :



Weak scaling: Gain may be maximized by optimising $NTIM_F$ / $NTIM_G$

Strong scaling: Gain will reduce for high processor number as $NTIM_F$ & $NTIM_G$ reduce significantly.

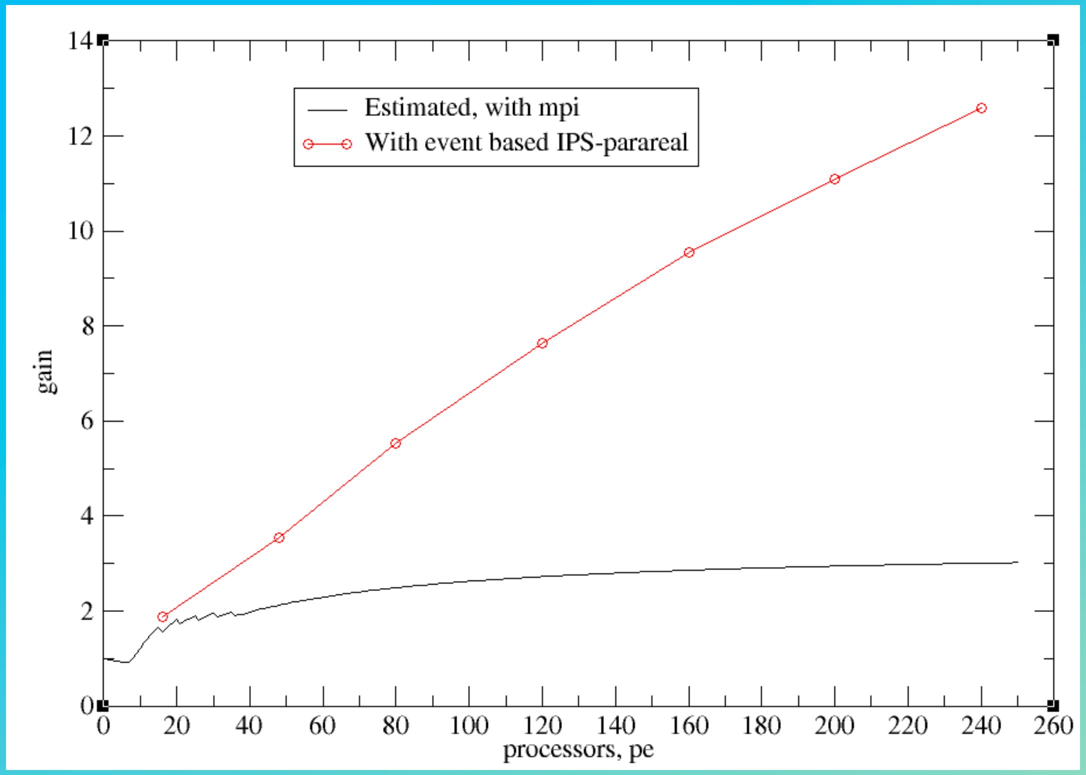


Parareal Algorithm Using the IPS Framework

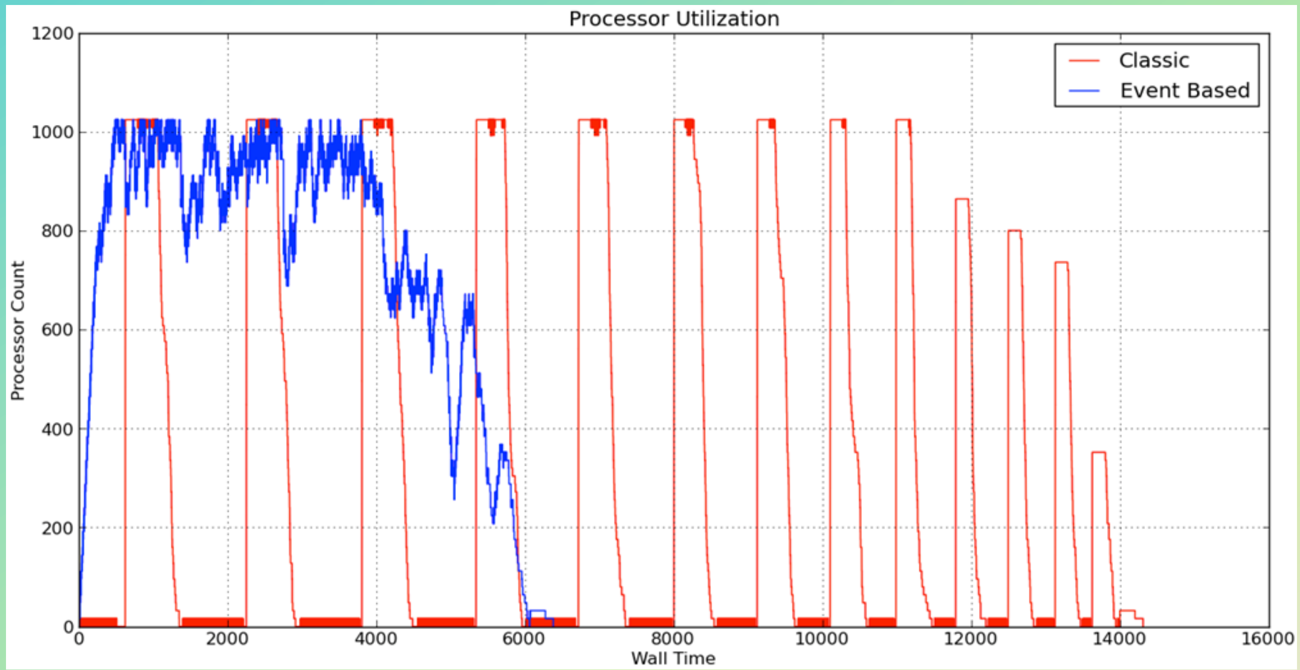
Advantages of using the IPS Framework

- portable parareal framework (L.Berry, W. Elwasif, ORNL) –written in python.
- exploring multiple cases with relative ease.
- hybrid parallelization (space + time).
- Less focus on numerics of parareal scheme.
- Prime focus on coarse solver.
- Reuse of processors already having attained convergence.

Event based implementation greatly enhances performance



Case: “G with no Eirene”.



Event based parareal implementation using the IPS framework greatly improves resource utilization as well as gain.

Using IPS-parareal is way better than traditional MPI implementation!

Conclusions

- Parareal algorithm may be successfully applied to edge physics simulations, hence studies of the scrape off layer may become more tractable.

- For case with "no Eirene in G", a gain of 12.58 was observed with 240 processors.

- Another coarse solver, G is explored where the grid size and bigger dt are reduced – for MAST & DIIID simulations.

- DIIID: Gain=21.8 with 96 processors & MAST: Gain=15.9 with 64 processors were observed.

- For both coarse solvers, convergence is sensitive to the size of time slices per processor.

- Time parallelization may be coupled with space parallelization to yield maximum gain and efficiency.

- IPS framework (from ORNL) greatly simplifies the use of the scheme and enhances performance.

Acknowledgements:

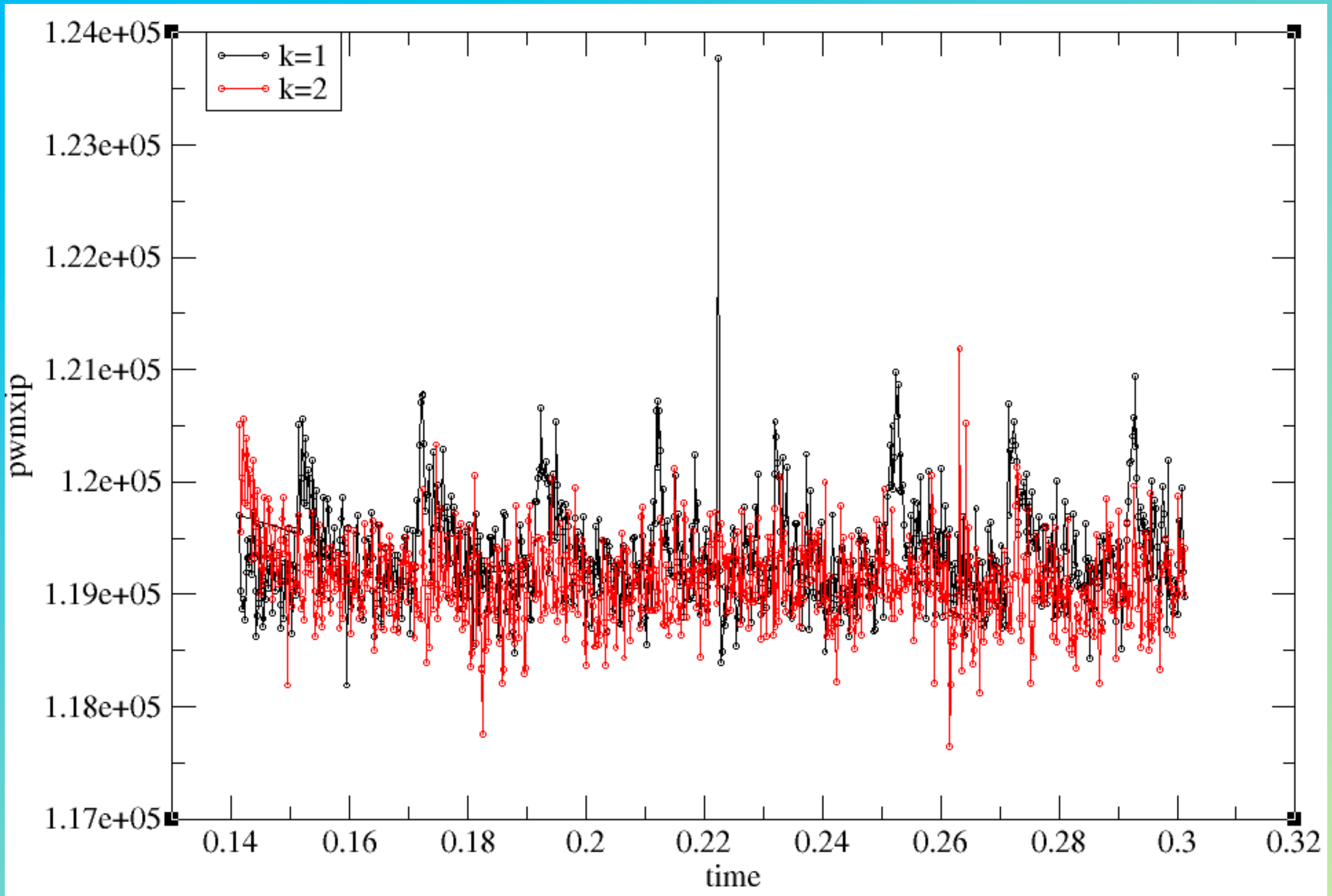
- Consultants at ITM-Gateway.
- WP13-SOL-01-01/CCFE/PS funding from EFDA.

References

- Lions J. L et al, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp. 661668
- Samaddar D. et al, J. Comp Phys,18 (229) (2010)
- Schneider, R. et al, Contrib. Plasma Phys. 46, No. 1-2,3-191 (2006)
- Berry, L. A. et. al, Journal of Computational Physics 231(2012) 59455954
- Elwasif W.R et al, 4th IEEE Workshop on Many-Task Computing on Grids and Supercomputers, MTAGS 2011, (2011)

Thank you

Parareal converges with fine timeslice per processors =100



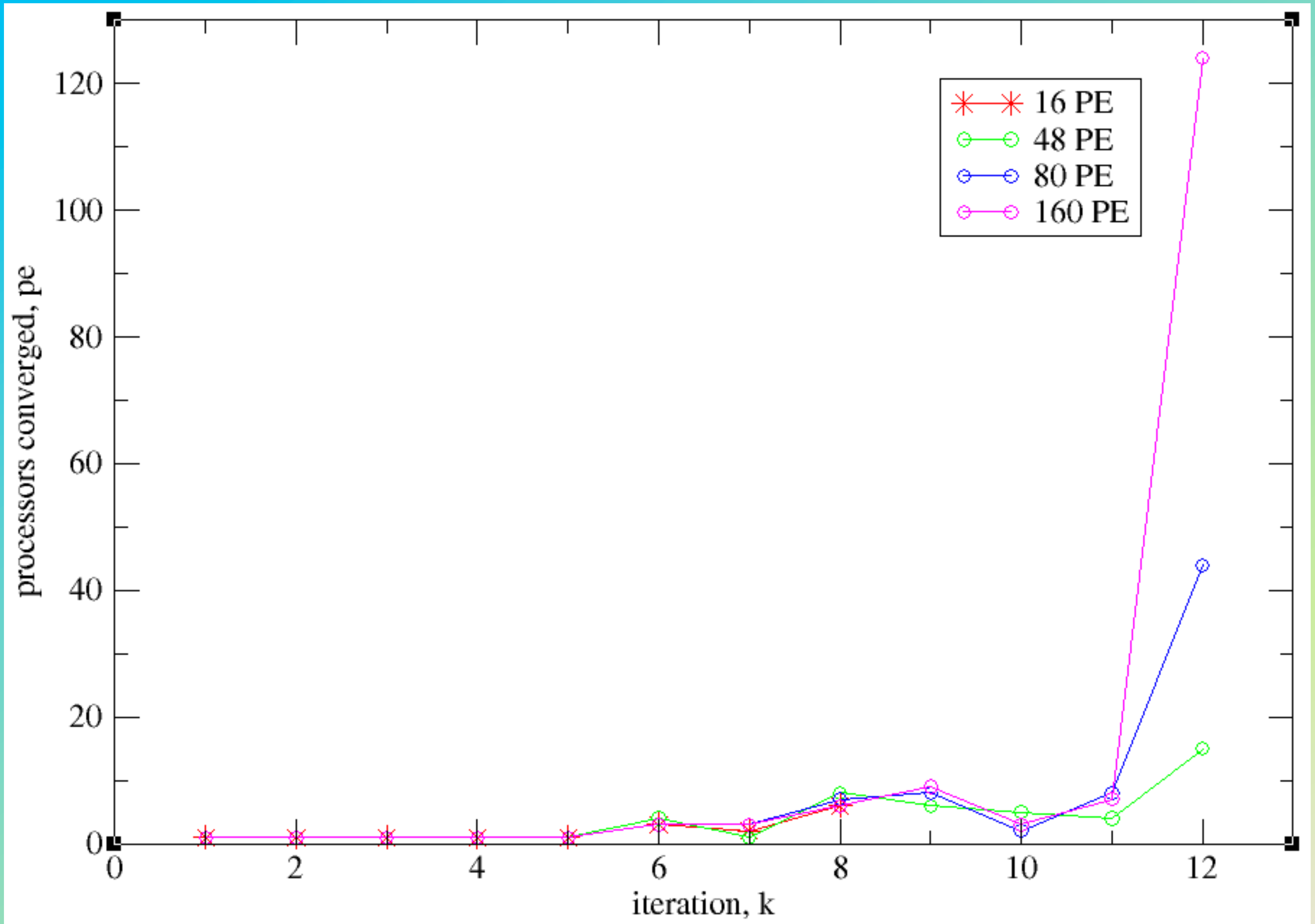
Reduced grid: 150X18
Gain=12.536 with 32 processors!!

Fine solution for different k, with fine timeslice=100.

Convergence at k = 2

dt_g = 50dt_f

Weak scaling (no Eirene)



Convergence in 12 iterations, irrespective of processor numbers.

SOLPS – code used for edge physics studies

•Package consists of 2 codes: B2(plasma fluid transport) and Eirene(neutral particle transport).

•Parallel and perpendicular transport described in 2D system.

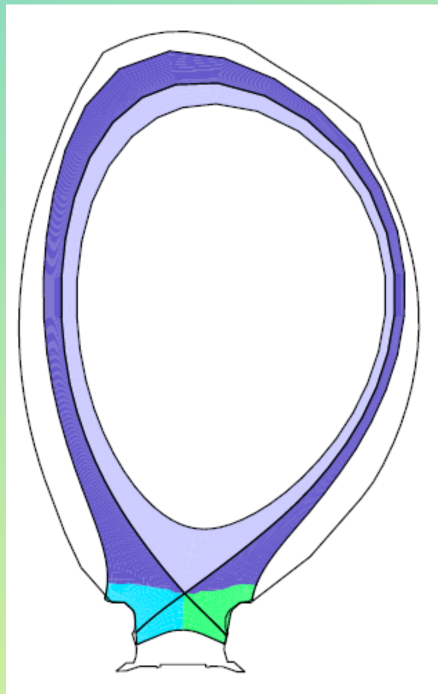
•SOL – characterized by open field lines at surfaces of device and atomic processes are important.

•SOLPS – widely used to understand physics of SOL.

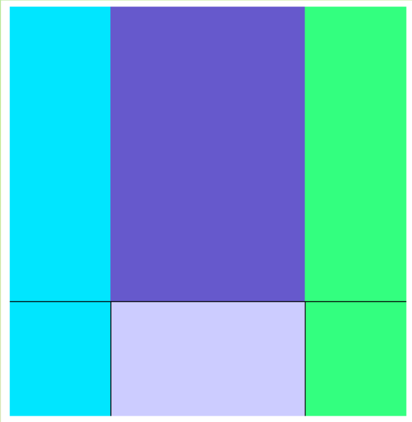
•SOLPS – extremely computationally intensive.

geometry

- toroidal symmetry
- equations written in curvlinear coordinates coinciding with magnetic geometry



physical plane



computational plane

Equations in SOLPS

transport equations

$$V_x = b_z V_\perp + b_x V_\parallel$$

c u r v l i n e a r
coordinates

metric coefficients

geometry

continuity ions

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} n \left(b_x V_\parallel + b_z V_\perp^{(0)}\right)\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} n V_y^{(0)}\right) = S^n$$

$$V_\perp^{(0)} = V_\perp^{(a)} + V_\perp^{(in)} + V_\perp^{(vis)} + V_\perp^{(s)} + \tilde{V}_\perp^{(dia)}$$

$$V_y^{(0)} = V_y^{(a)} + V_y^{(in)} + V_y^{(vis)} + V_y^{(s)} + \tilde{V}_y^{(dia)}$$

ExB & diffusive (ambipolar)
inertial, viscous
ion-neut. friction
diamagnetic

flow

parallel momentum ions

$$m_i \left[\frac{\partial n V_\parallel}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} n \left(V_\perp^{(0)} b_z + V_\parallel b_x\right) V_\parallel\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} n V_y^{(0)} V_\parallel\right)\right] =$$

$$- \frac{b_x}{h_x}\frac{\partial n T_i}{\partial x} - b_x \frac{en}{h_x}\frac{\partial \Phi}{\partial x} + F_k + \frac{4}{3} b_x B^{3/2} \frac{\partial}{h_x \partial x}\left(\frac{\eta_0 b_x}{B^2}\frac{\partial\left(\sqrt{B}\left(V_\parallel + b_x\left(V_\perp^{(dia)} + V_\perp^{(E)}\right)\right)\right)}{h_x \partial x}\right)$$

$$+ B^{3/2} b_x \frac{\partial}{h_x \partial x}\left(\frac{b_x}{\nu_{ii} B^2}\frac{\partial\left(\sqrt{B} q_{i\parallel}^{(0)}\right)}{h_x \partial x}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y^2}\eta_2 \frac{\partial V_\parallel}{\partial y}\right)$$

$$+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x^2}\eta_2 \frac{\partial V_\parallel}{\partial x}\right) + S_{i\parallel}^m + R_{ie\parallel}$$

… Equations in SOLPS continued

current continuity

$$j_{\parallel} = \sigma_{\parallel} \left(\frac{b_x}{e} \frac{1}{h_x} \left(\frac{\partial n T_e}{n \partial x} + 0.71 \frac{\partial T_e}{\partial x} \right) - \frac{b_x}{h_x} \frac{\partial \Phi}{\partial x} \right)$$

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{j}_x \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{j}_y \right) = 0$$

energy conservation ions and electrons

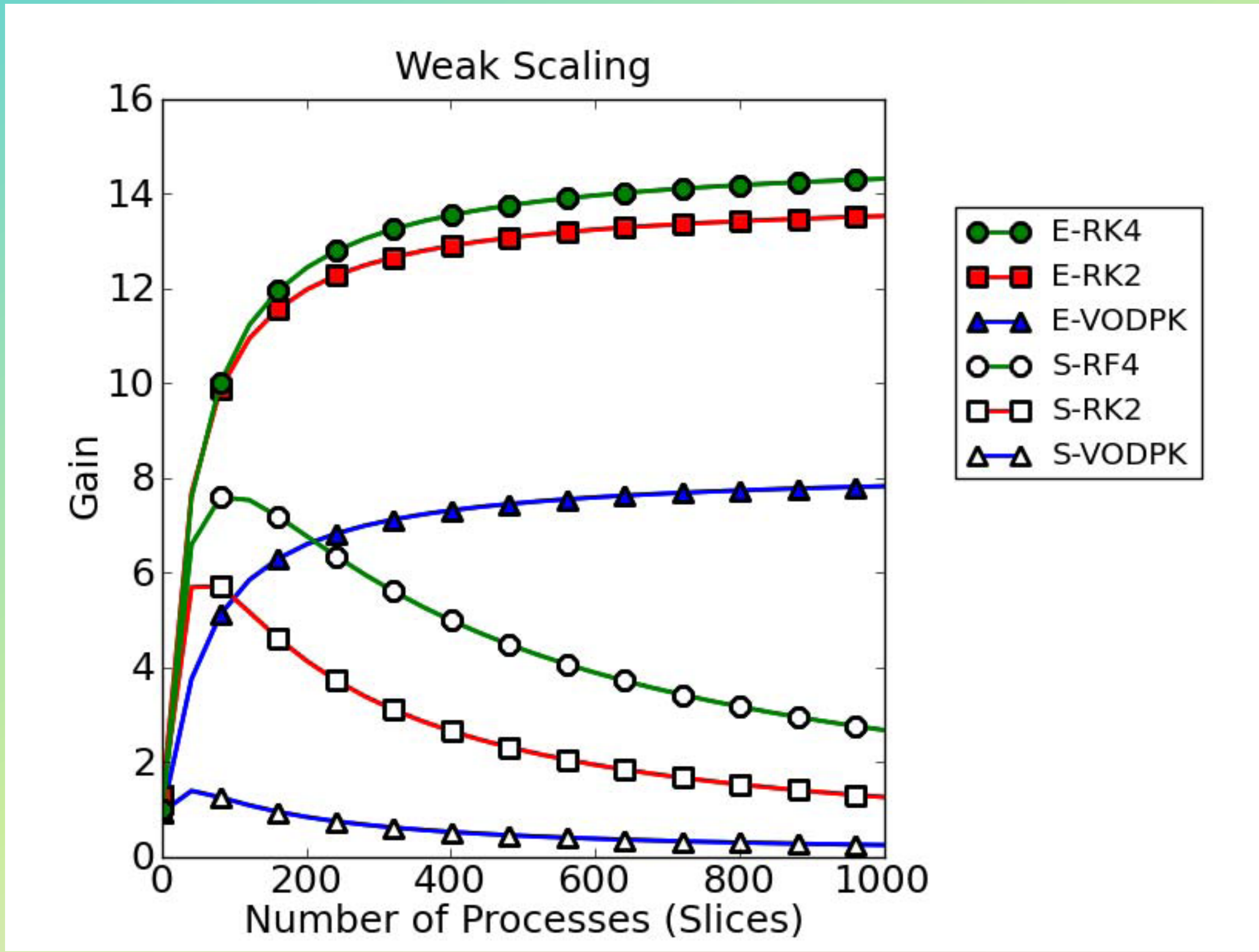
$$\begin{aligned} &\frac{3}{2} \frac{\partial n T_e}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ex} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{ey} \right) + \frac{n T_e}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \left(V_{\parallel} - j_{\parallel}/en \right) b_x \right) \\ &\quad = Q_e + n T_e B \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) \end{aligned}$$

$$\begin{aligned} &\frac{3}{2} \frac{\partial n T_i}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} \tilde{q}_{ix} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\frac{\sqrt{g}}{h_y} \tilde{q}_{iy} \right) + \frac{n T_i}{\sqrt{g}} \frac{\partial}{\partial x} \left(\frac{\sqrt{g}}{h_x} V_{\parallel} b_x \right) \\ &\quad = Q_{\Delta} + Q_{AN} + \frac{\eta_0}{3} \left(2 b_x \frac{\partial V_{\parallel}}{h_x \partial x} \right)^2 + n T_i B \frac{1}{h_x h_y} \left(\frac{\partial \Phi}{\partial y} \frac{\partial}{\partial x} \left(\frac{1}{B^2} \right) - \frac{\partial \Phi}{\partial x} \frac{\partial}{\partial y} \left(\frac{1}{B^2} \right) \right) \end{aligned}$$

Advantages of using a parareal framework (contd …)

- Opportunity to use event based parareal scheme, leading to multilevel concurrency and more flexibility with G.

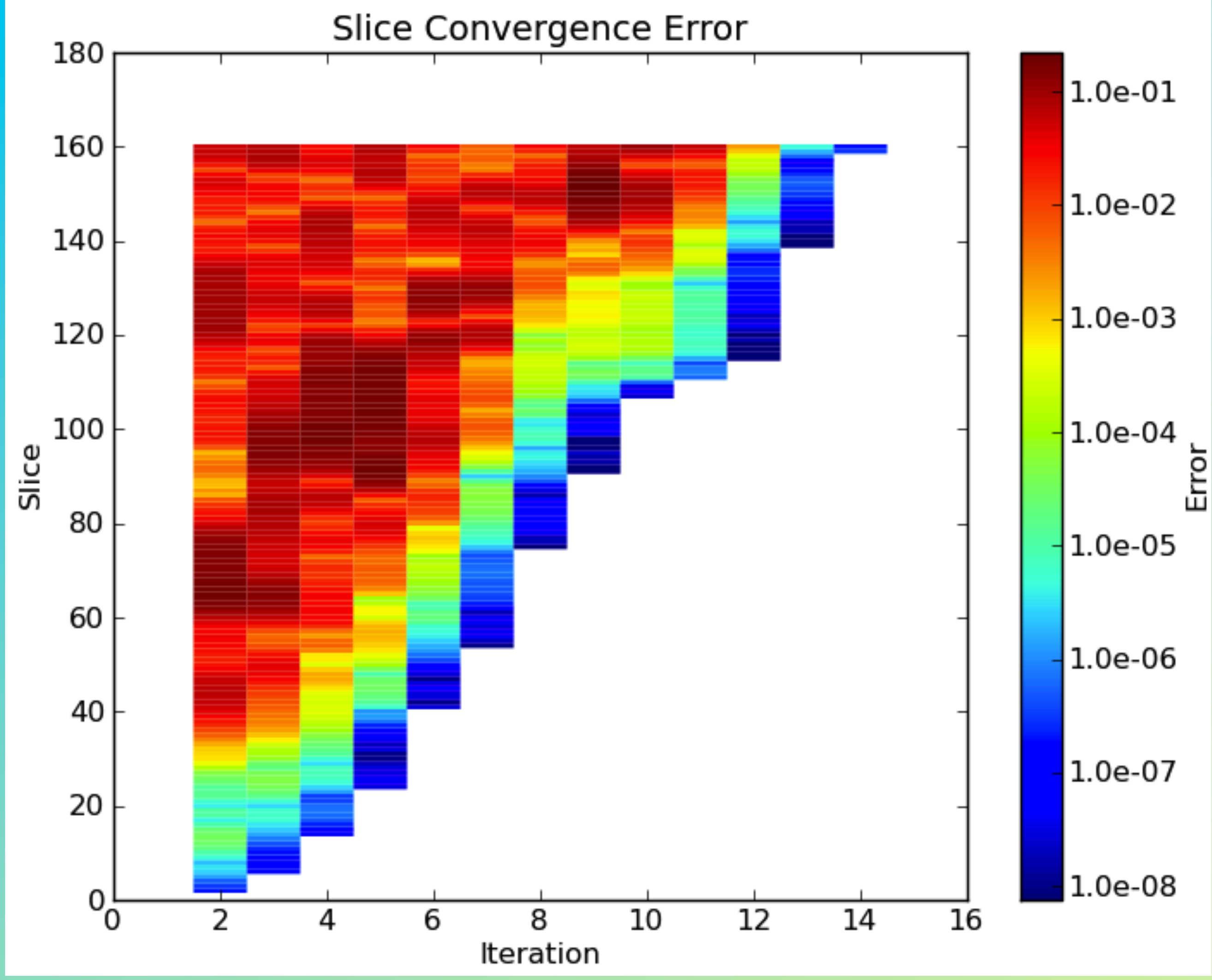
Event based implementation gives much better gain.



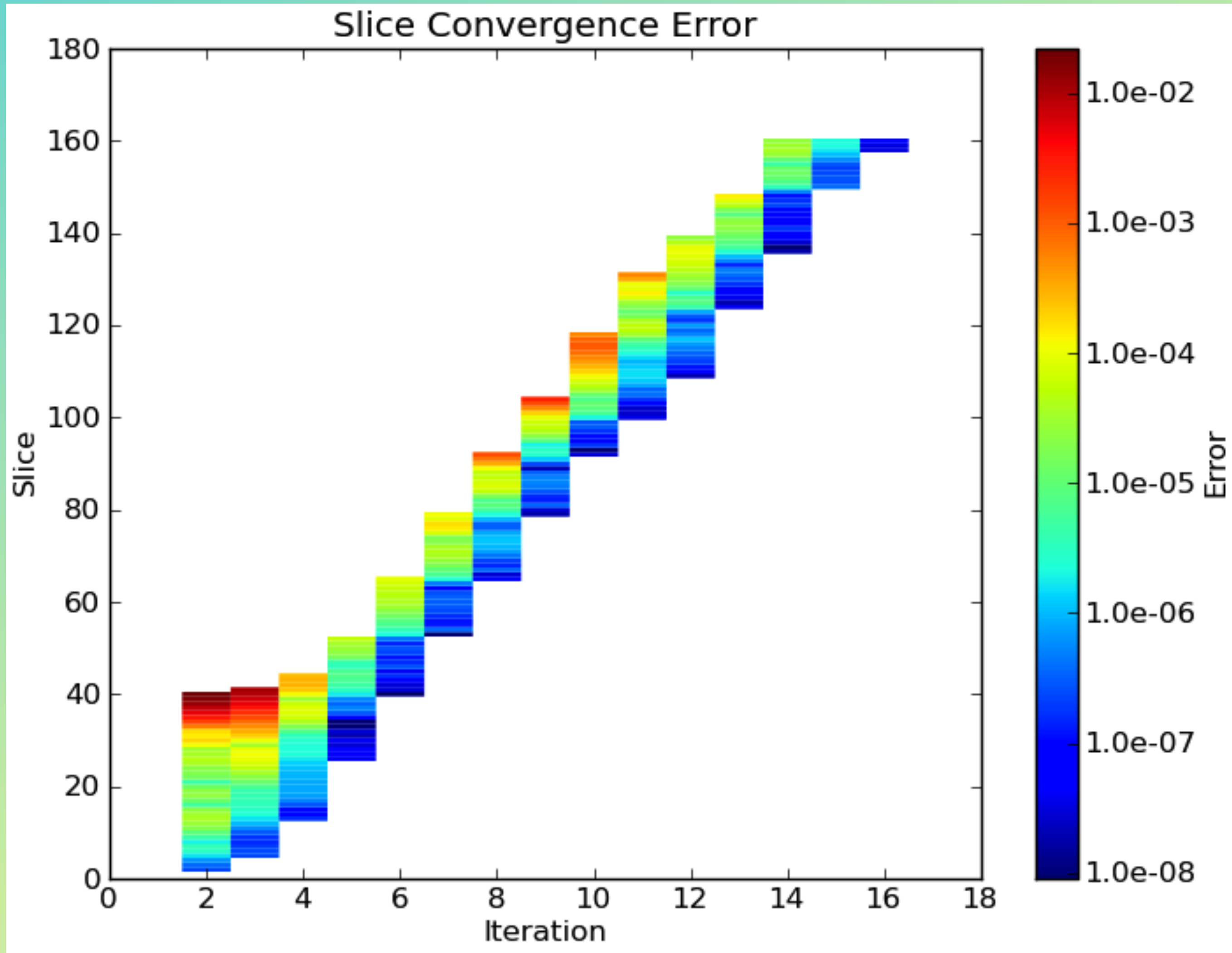
Ref: L. A. Berry et al. (2011)

Advantages of using a parareal framework (contd …)

- "Moving Window" parareal scheme allows even better resource utilization (Elwasif et al. 2011).

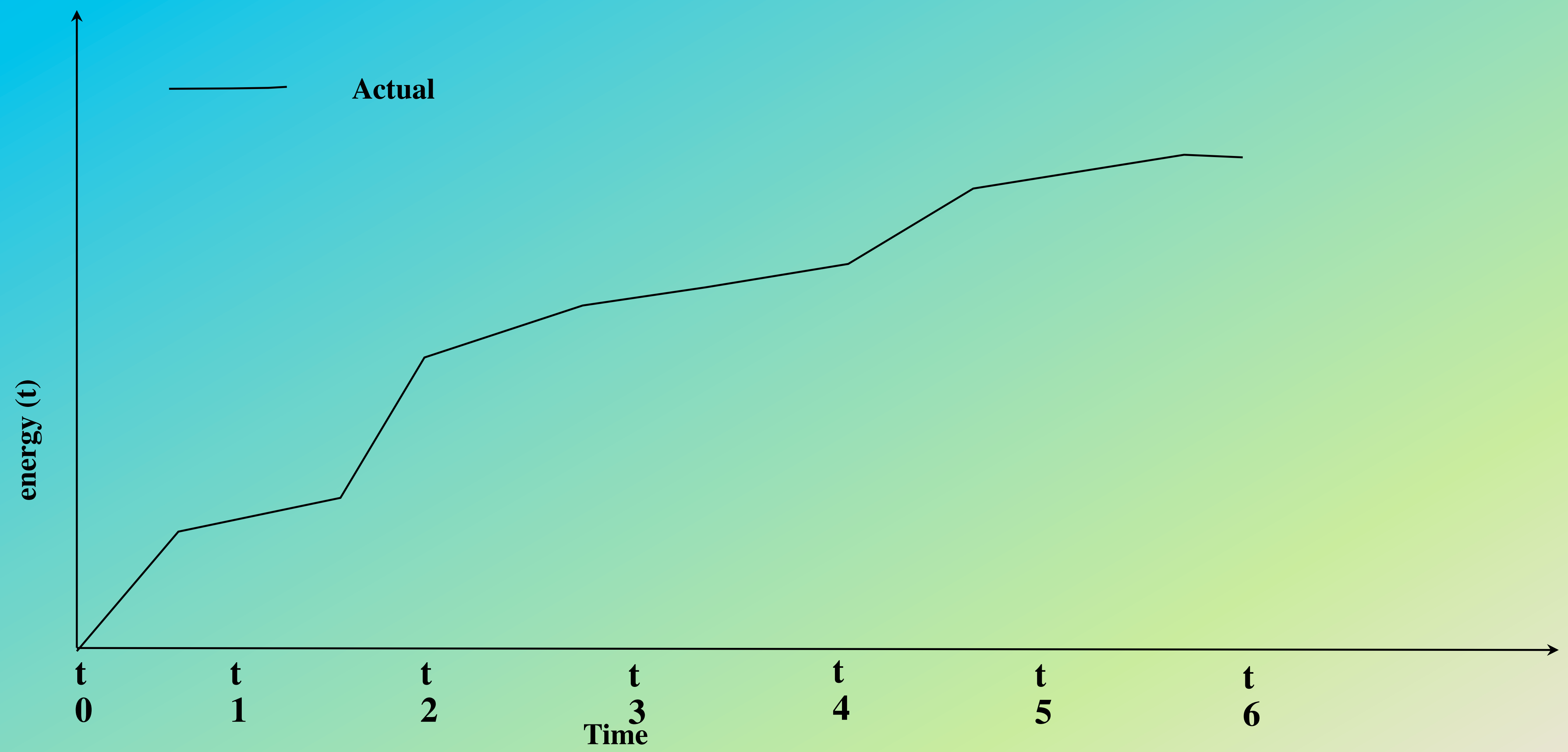


Regular, event-based parareal



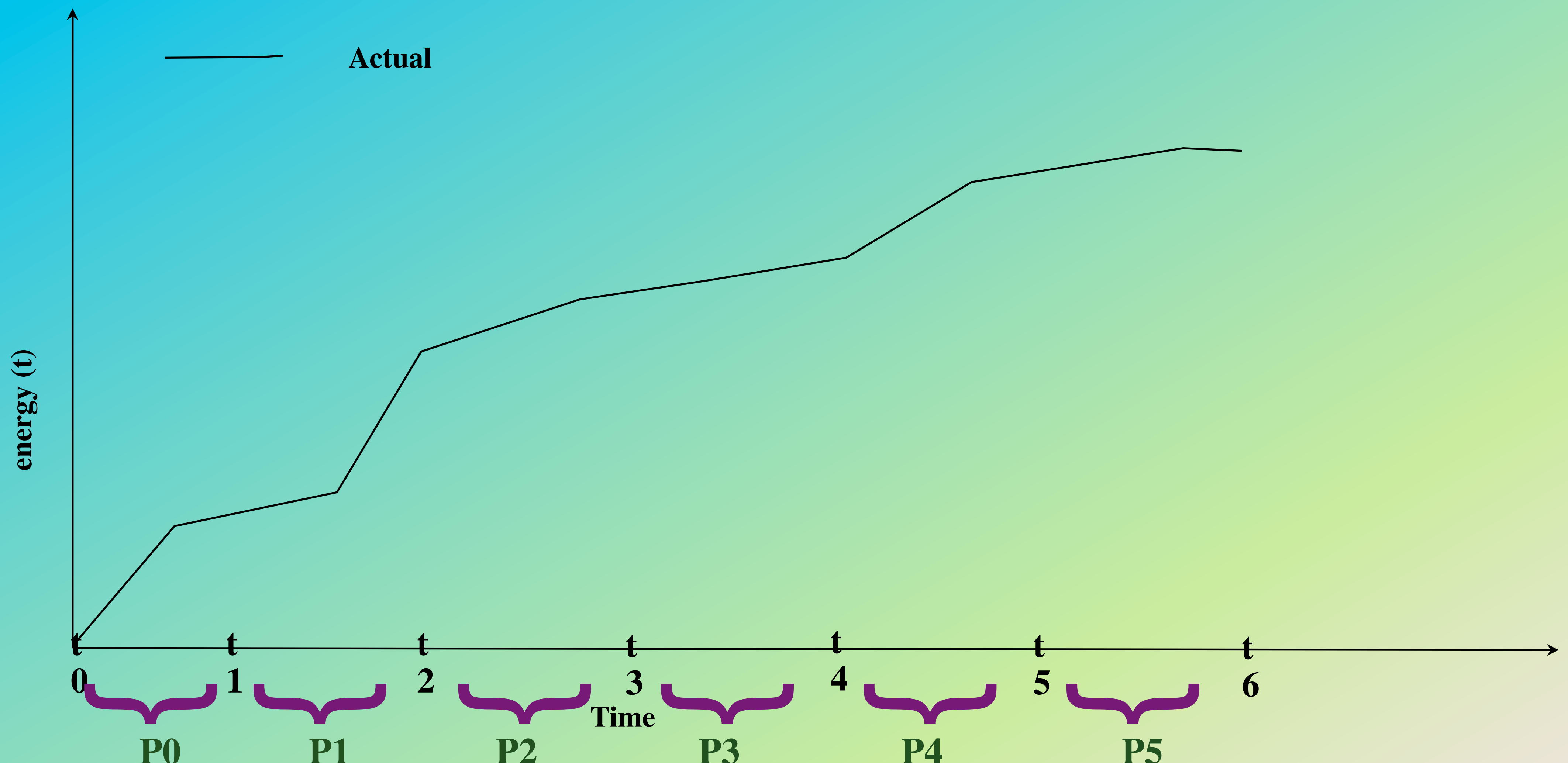
"Moving window" scheme

F is a propagator evolving the state, λ. The function, energy(λ,t) thus changes from initial time, t_0,to a later time ...



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

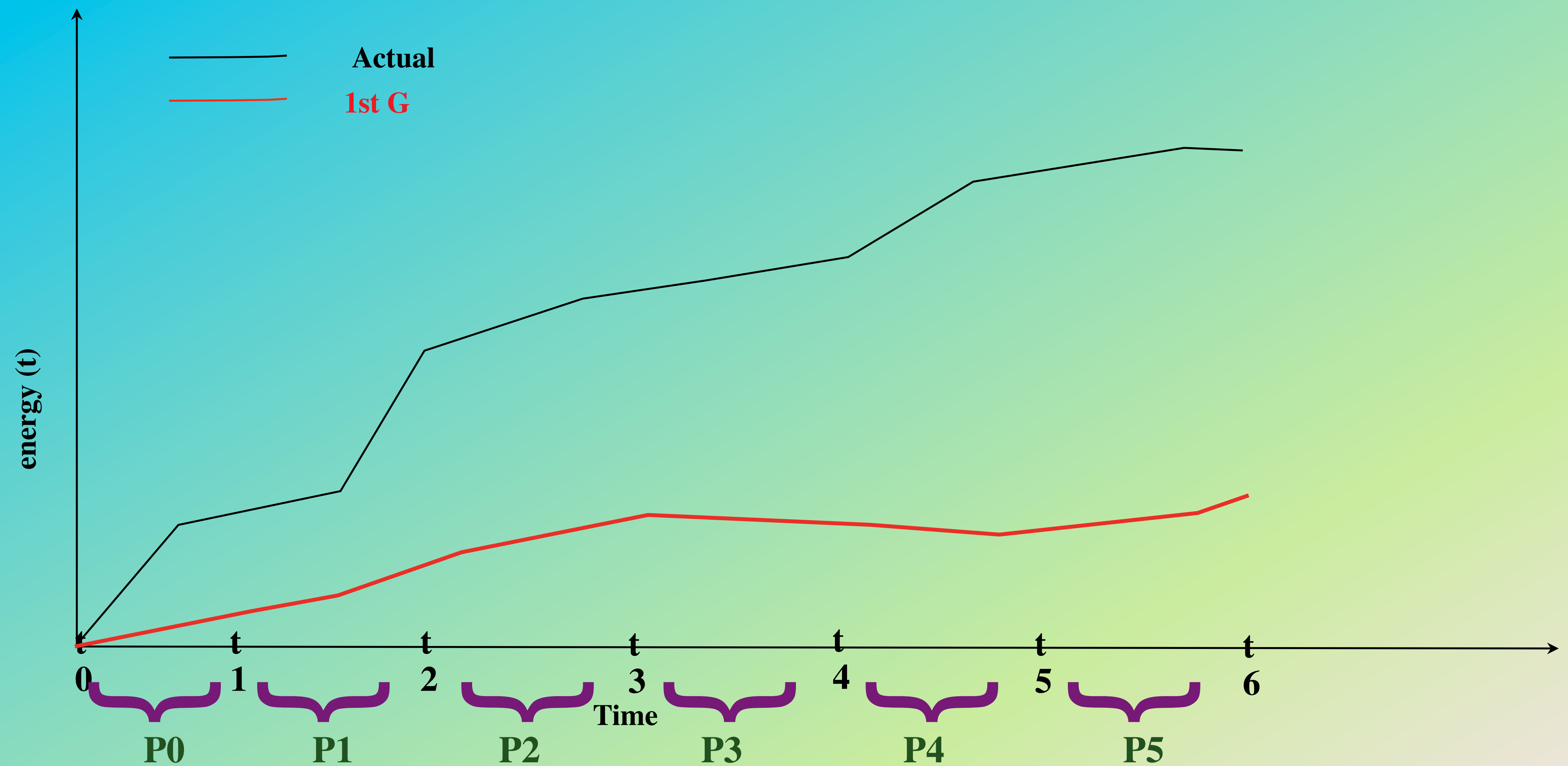
F is a propagator evolving the state, λ. The function, energy(λ,t) thus changes from initial time, t_0, to a later time ...



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

F is a propagator evolving the state, λ. The function, energy(λ,t) thus changes from initial time, t_0,to a later time ...

G - faster but inaccurate propagator

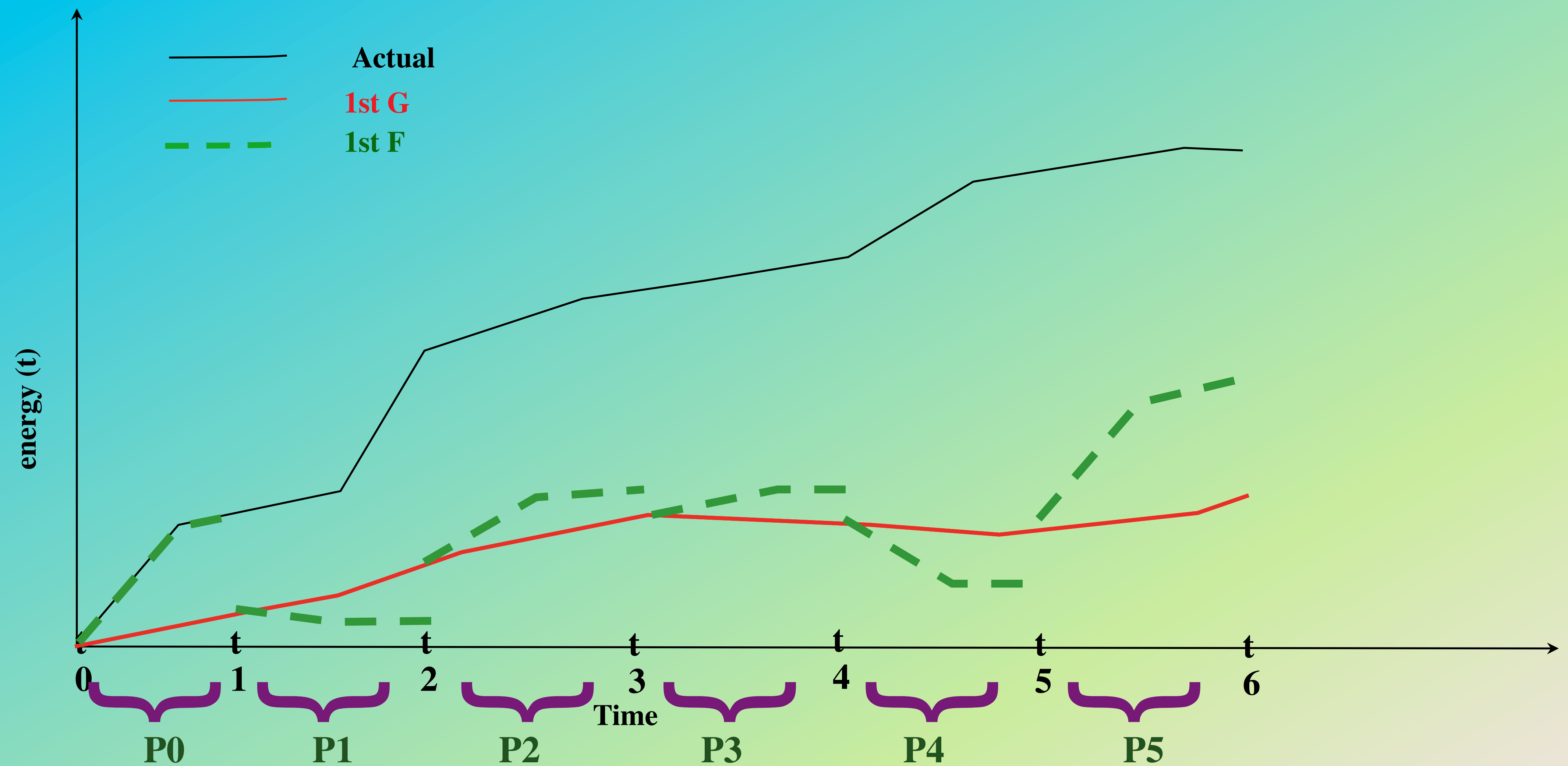


$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

cnrs LSPM UNIVERSITÉ PARIS 13 NORD EFDA CCFE CULHAM CENTRE FOR FUSION ENERGY IPP Fusion Energy Division OAK RIDGE NATIONAL LABORATORY Managed by UT-Battelle for the Department of Energy

F is a propagator evolving the state, λ. The function, energy(λ,t) thus changes from initial time, t_0,to a later time ...

G - faster but inaccurate propagator

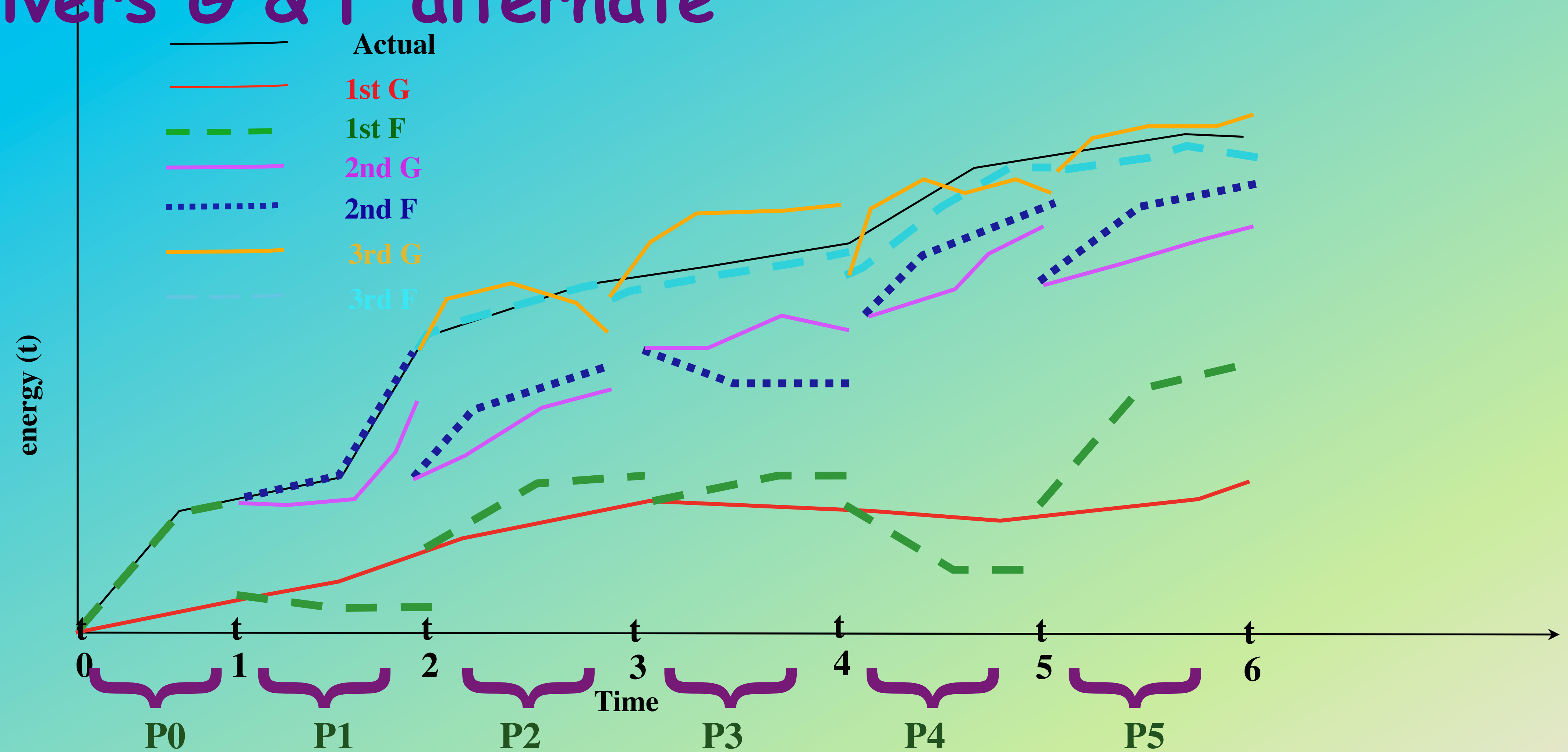


$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

F is a propagator evolving the function (energy(t)) from initial time, t_0, to a later time ...

G - faster but inaccurate propagator

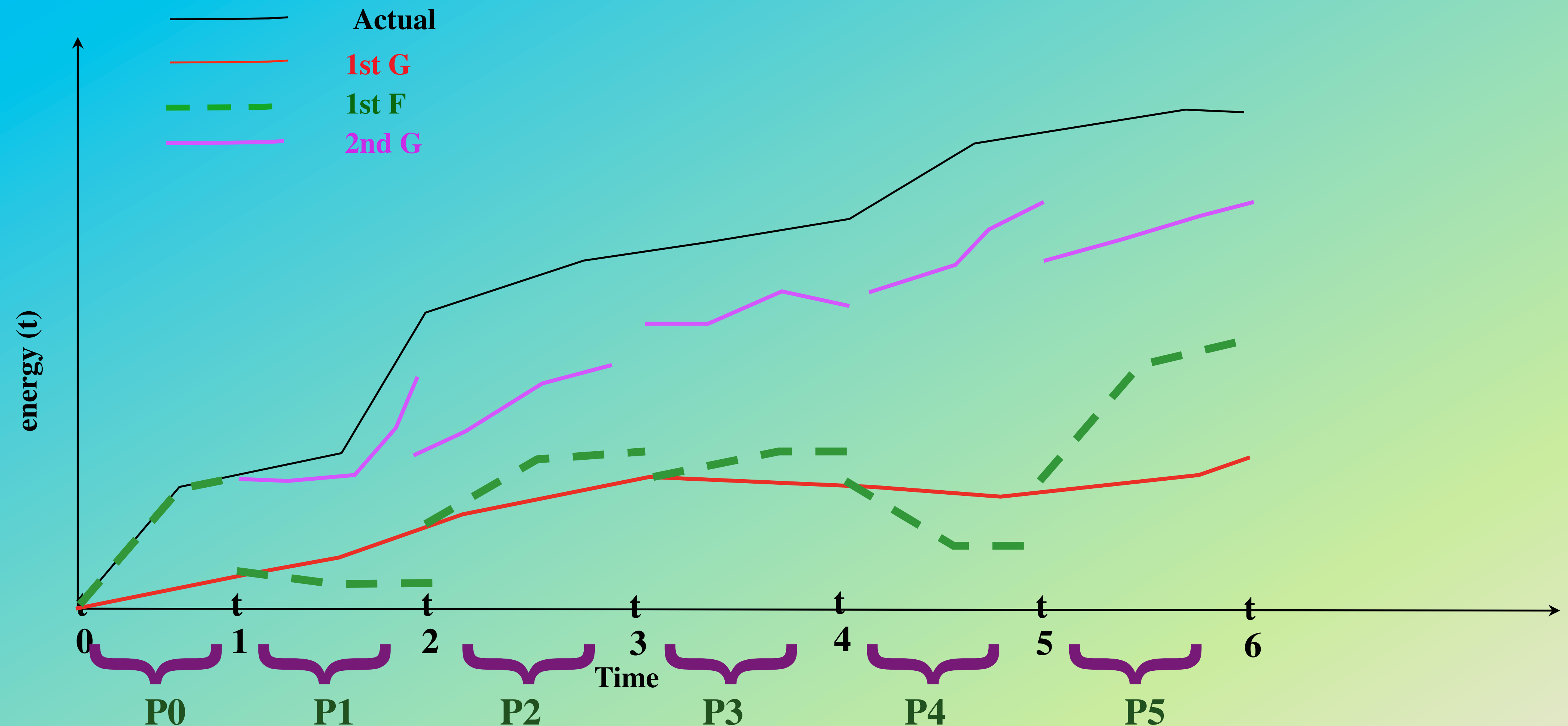
Solvers G & F alternate



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

New G　　　　Old G

F is a propagator evolving the state, λ. The function, energy(λ,t) thus changes from initial time, t_0,to a later time ... G - faster but inaccurate propagator. Solvers G & F alternate.



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

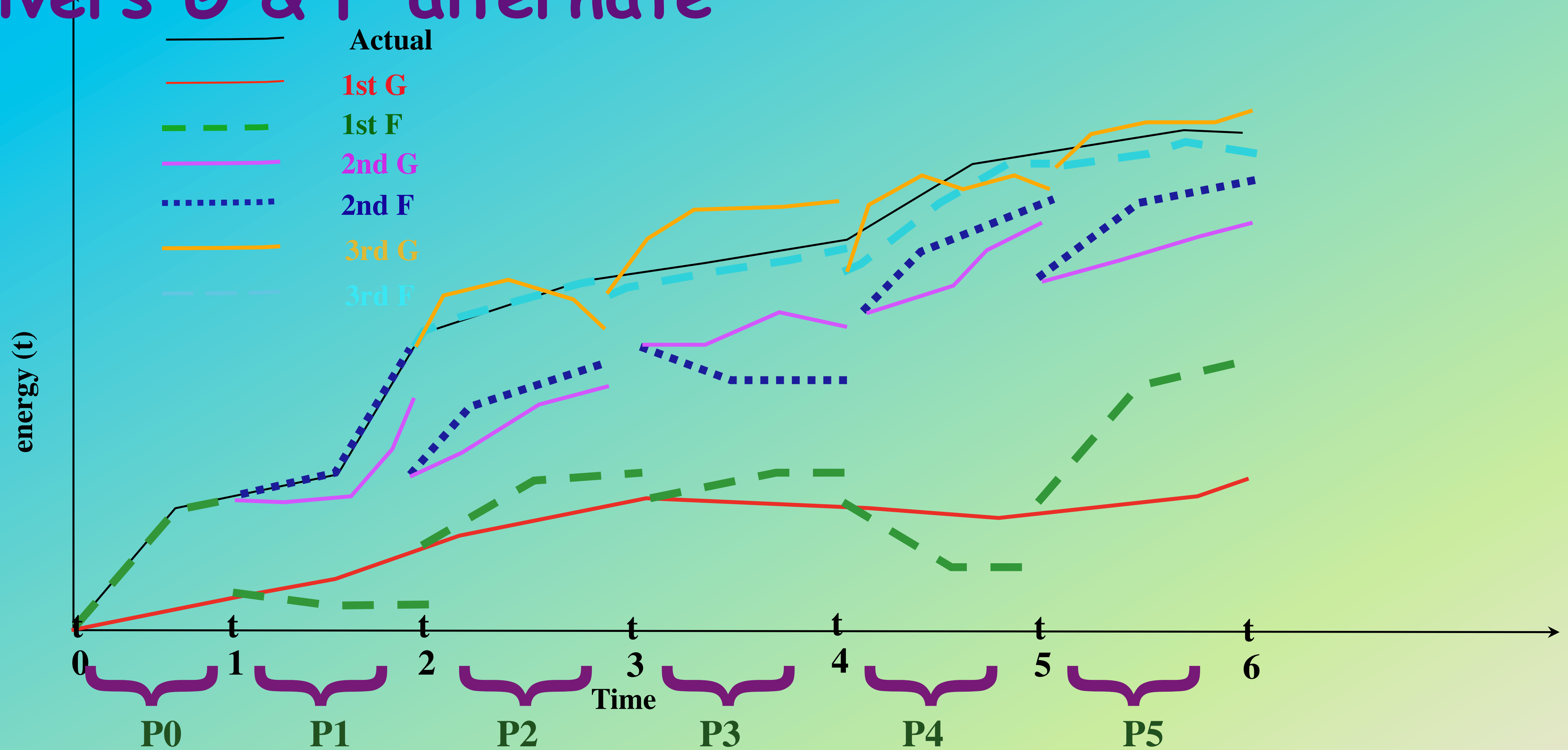
New G

Old G

F is a propagator evolving the function (energy(t)) from initial time, t_0, to a later time ...

G - faster but inaccurate propagator

Solvers G & F alternate



$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

New G

Old G

Basic Algorithm

k = 0 to (N-1),

$\lambda_0^0 = y_0$

i = 0 to (N-1)

$$\lambda_{i+1}^k = G_{\Delta t}\left(\lambda_i^k\right)$$

$F_{\Delta t}\left(\lambda_0^k\right)$ $F_{\Delta t}\left(\lambda_1^k\right)$ $F_{\Delta t}\left(\lambda_2^k\right)$ . . . $F_{\Delta t}\left(\lambda_{(N\text{-}1)}^k\right)$

i = 0 to (N-1)

$$\lambda_{i+1}^{k+1} = G_{\Delta t}\left(\lambda_{(i)}^{k+1}\right) + F_{\Delta t}\left(\lambda_{(i)}^{k}\right) - G_{\Delta t}\left(\lambda_{(i)}^{k}\right)$$

Check Convergence

k = k + 1

Metric for convergence:

$$error = \frac{\sum_i abs(PR_k - PR_{k-1})}{(PR_{k-1}) \times (n_c)}$$

SOLPS with no drifts and $j_\parallel = 0$

$$\frac{\partial n}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} n u_x\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} n u_y\right) = S^{(n)}$$

$$u_x = b_x u_\parallel + b_z u_\perp$$

$$u_y = -(D^n/nh_y)\partial n/\partial y,\ b_z u_\perp = -(D^n/nh_x)\partial n/\partial x$$

$$m_\mathrm{i}\left[\frac{\partial n u_\parallel}{\partial t} + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{h_z\sqrt{g}}{h_x} n u_x u_\parallel\right) + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{h_z\sqrt{g}}{h_y} n u_y u_\parallel\right)\right]$$

$$-\frac{4}{3} b_x B^{\frac{3}{2}} \frac{\partial}{h_x \partial x}\left(\frac{\eta_0 b_x}{B^2}\frac{\partial B^{\frac{1}{2}} u_\parallel}{h_x \partial x}\right) - B^{\frac{3}{2}} b_x \frac{\partial}{h_x \partial x}\left(\frac{b_x}{\nu_\mathrm{ii} B^2}\frac{\partial B^{\frac{1}{2}} q_{\parallel,\mathrm{i}}}{h_x \partial x}\right) =$$

$$-\frac{b_x}{h_x}\frac{\partial p_\mathrm{i}}{\partial x} - b_x \frac{en}{h_x}\frac{\partial \phi}{\partial x} + R_{\parallel,\mathrm{i}} + S_\parallel^{(u)}$$

additional parallel viscosity driven by ion heat flux

3. BENCHMARKING SOLF1D WITH SOLPS

SOLPS with no drifts and $j_\parallel = 0$

$$\frac{3}{2}\frac{\partial nkT_\mathrm{e}}{\partial t}+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}q_{x,\mathrm{e}}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}q_{y,\mathrm{e}}\right)+\frac{nkT_\mathrm{e}}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x u_\parallel\right)=Q_\mathrm{e}+S_\mathrm{e}^{(E)}$$

$$q_{x,\mathrm{e}}=\frac{3}{2}nkT_\mathrm{e}b_x u_\parallel-\kappa_{\parallel,\mathrm{e}}\frac{b_x^2}{h_x}\frac{\partial kT_\mathrm{e}}{\partial x}-\frac{5}{2}nkT_\mathrm{e}b_z D_\mathrm{an}\frac{1}{h_x n}\frac{\partial n}{\partial x}$$

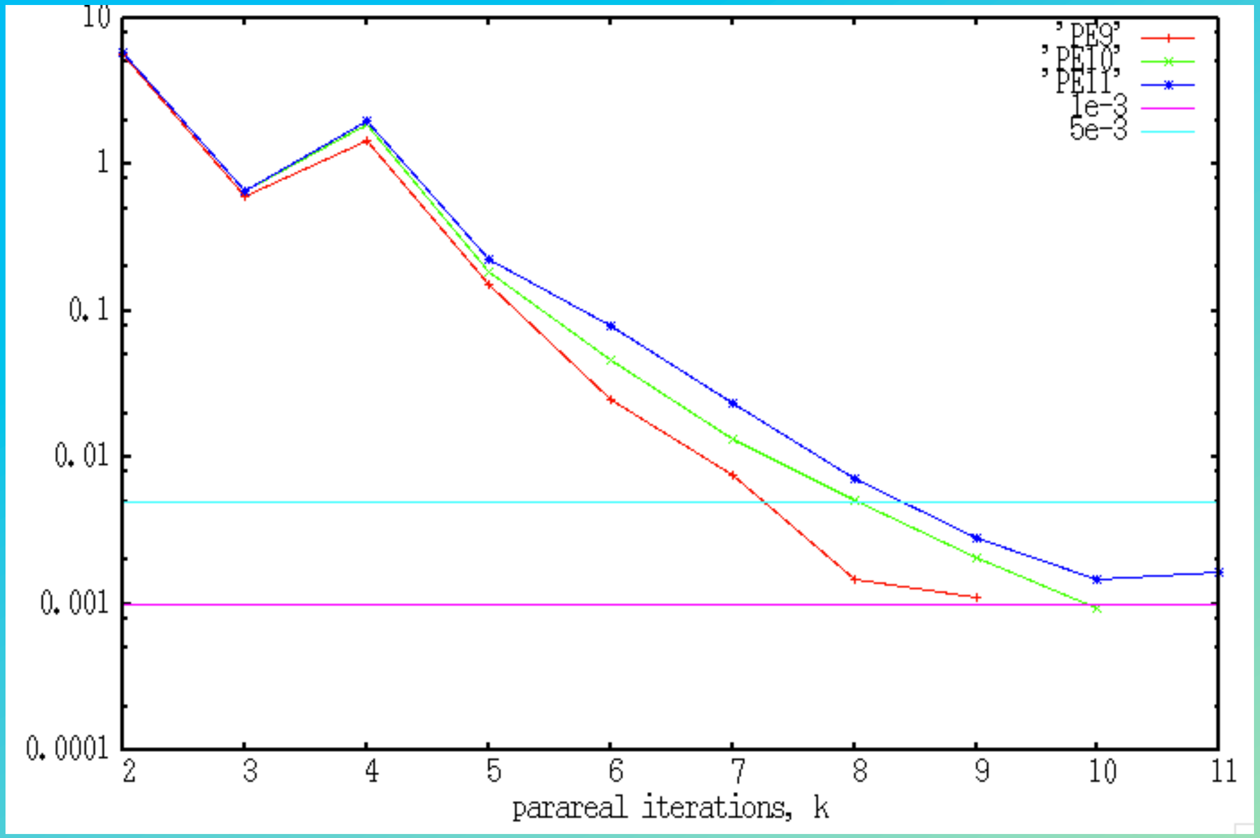
$$q_{y,\mathrm{e}}=-\frac{5}{2}nkT_\mathrm{e}D_\mathrm{an}\frac{1}{h_y n}\frac{\partial n}{\partial y}$$

$$\frac{3}{2}\frac{\partial nkT_\mathrm{i}}{\partial t}+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}q_{x,\mathrm{i}}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}q_{y,\mathrm{i}}\right)+\frac{nkT_\mathrm{i}}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x u_\parallel\right)=Q_\mathrm{vis}+Q_\mathrm{i}+S_\mathrm{i}^{(E)}$$

$$q_{x,\mathrm{i}}=\frac{3}{2}nkT_\mathrm{i}b_x u_\parallel-\kappa_{\parallel,\mathrm{i}}\frac{b_x^2}{h_x}\frac{\partial kT_\mathrm{i}}{\partial x}-\frac{5}{2}nkT_\mathrm{i}b_z D_\mathrm{an}\frac{1}{h_x n}\frac{\partial n}{\partial x}$$

$$q_{y,\mathrm{i}}=-\frac{5}{2}nkT_\mathrm{i}D_\mathrm{an}\frac{1}{h_y n}\frac{\partial n}{\partial y}$$

$$Q_\mathrm{vis}=\frac{\eta_0}{3}\left(\frac{2b_x}{\sqrt{B}}\frac{\partial u_\parallel\sqrt{B}}{h_x\partial x}\right)^2$$



The error starts to oscillate for values lower than ~5E-3

Note: The simulations were done on the ITM gateway with 16 cores per node. But when all cores per node were used simultaneously, the resulting restriction on the memory available per core slowed the simulation. It was observed that with each processor solving a timeslice of 10 (i.e, b2mndr_ntim=10), using 8 cores per node was optimum for the cases investigated.