Parareal methods and averaged models for solving stiff differential equations

Sever Hirstoaga

ALPINES team, Inria Paris & LJLL, Sorbonne Université, CNRS

Strasbourg, August 2024

Summer school "New trends in computing", Strasbourg

Outline

Outline

Motivation

Challenge: Numerical solving of **nonlinear** Partial Differential Equations in **high** dimensions with **multiple scales**.

AIM: Design efficient and robust solvers for these problems.

- Different methods to speed up their rate of convergence: multigrid methods, domain decomposition methods.
- Parallelization techniques: design algorithms adapted to many-core modern architectures $\rightsquigarrow$ faster simulations.

When communication time completely dominate the overall computing time $\Rightarrow$ use the time direction for parallelization

This talk: focus on parallelism in time, despite the sequential nature of evolution problems.

Motivation

Challenge: Numerical solving of **nonlinear** Partial Differential Equations in **high** dimensions with **multiple scales**.

AIM: Design efficient and robust solvers for these problems.

- Different methods to speed up their rate of convergence: multigrid methods, domain decomposition methods.
- Parallelization techniques: design algorithms adapted to many-core modern architectures $\rightsquigarrow$ faster simulations.

When communication time completely dominate the overall computing time $\Rightarrow$ use the time direction for parallelization

This talk: focus on parallelism in time, despite the sequential nature of evolution problems.

Motivation

Challenge: Numerical solving of **nonlinear** Partial Differential Equations in **high** dimensions with **multiple scales**.

AIM: Design efficient and robust solvers for these problems.

- Different methods to speed up their rate of convergence: multigrid methods, domain decomposition methods.
- Parallelization techniques: design algorithms adapted to many-core modern architectures $\rightsquigarrow$ faster simulations.

When communication time completely dominate the overall computing time $\Rightarrow$ use the time direction for parallelization

This talk: focus on parallelism in time, despite the sequential nature of evolution problems.

Motivation

Challenge: Numerical solving of **nonlinear** Partial Differential Equations in **high** dimensions with **multiple scales**.

AIM: Design efficient and robust solvers for these problems.

- Different methods to speed up their rate of convergence: multigrid methods, domain decomposition methods.
- Parallelization techniques: design algorithms adapted to many-core modern architectures $\rightsquigarrow$ faster simulations.

When communication time completely dominate the overall computing time $\Rightarrow$ use the time direction for parallelization

This talk: focus on parallelism in time, despite the sequential nature of evolution problems.

Motivation

Challenge: Numerical solving of **nonlinear** Partial Differential Equations in **high** dimensions with **multiple scales**.

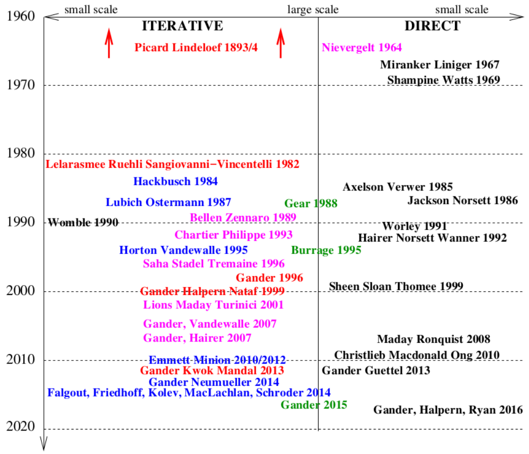
AIM: Design efficient and robust solvers for these problems.

- Different methods to speed up their rate of convergence: multigrid methods, domain decomposition methods.
- Parallelization techniques: design algorithms adapted to many-core modern architectures $\rightsquigarrow$ faster simulations.

When communication time completely dominate the overall computing time $\Rightarrow$ use the time direction for parallelization

This talk: focus on parallelism in time, despite the sequential nature of evolution problems.

Parallelism in time overview



- multiple shooting
- domain decomposition and waveform relaxation
- time-space multigrid
- direct method
- overview papers

from M. J. Gander: *50 Years of Time Parallel Time Integration, 2015*.

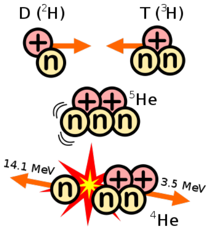
Domain of application

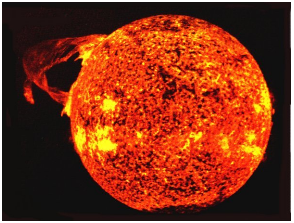
Plasma physics

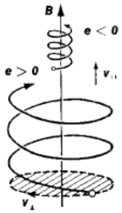
A gas heated at more than 10000 K $\rightsquigarrow$ electrons leave the orbit of their atoms $\rightsquigarrow$ **plasma** = a mixture of ions, neutrals and free electrons.

Plasma is sensitive to electromagnetic fields $\rightsquigarrow$ **complex dynamics**.

The thermonuclear fusion, by magnetic confinement:
strong magnetic field $\Rightarrow$ trapped particles $\Rightarrow$ fusion possibility







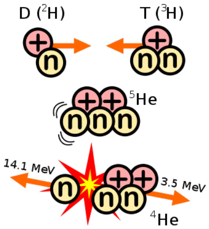
Domain of application

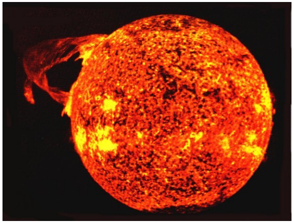
Plasma physics

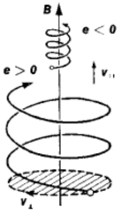
A gas heated at more than 10000 K $\rightsquigarrow$ electrons leave the orbit of their atoms $\rightsquigarrow$ **plasma** = a mixture of ions, neutrals and free electrons.

Plasma is sensitive to electromagnetic fields $\rightsquigarrow$ **complex dynamics**.

The thermonuclear fusion, by magnetic confinement:
strong magnetic field $\Rightarrow$ trapped particles $\Rightarrow$ fusion possibility







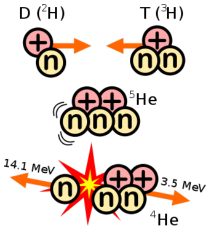
Domain of application

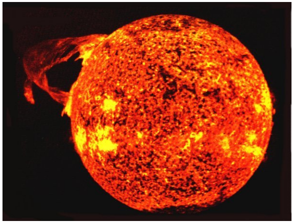
Plasma physics

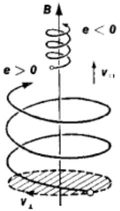
A gas heated at more than 10000 K $\rightsquigarrow$ electrons leave the orbit of their atoms $\rightsquigarrow$ **plasma** = a mixture of ions, neutrals and free electrons.

Plasma is sensitive to electromagnetic fields $\rightsquigarrow$ **complex dynamics**.

The thermonuclear fusion, by magnetic confinement:
strong magnetic field $\Rightarrow$ trapped particles $\Rightarrow$ fusion possibility





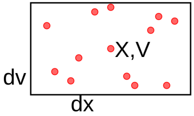


Modelization

1. Microscopic (Newton law)
2. Mesoscopic (kinetic model)
3. Macroscopic (fluid model)

<u>Kinetic approach</u>

- a large number of particles $\longrightarrow$ statistic description $\longrightarrow$ evolution in time of a function f entailing how the particles are placed.
- $f \equiv f(t, \mathbf{x}, \mathbf{v})$. $f d\mathbf{x} d\mathbf{v}$ a statistical mean of the repartition of particles in the box of the phase space



Vlasov equation

- after Anatoly A. Vlasov (1908-1975)
- $f = f_s$ distribution function of particle species s.

$$\underbrace{\frac{\partial f_s}{\partial t}}_{\text{variations in time}} + \underbrace{\mathbf{v}\cdot\frac{\partial f_s}{\partial \mathbf{x}}}_{\text{variations in space}} + \underbrace{\frac{q}{m}(\mathbf{E}+\mathbf{v}\times\mathbf{B})\cdot\frac{\partial f_s}{\partial \mathbf{v}}}_{\text{variations en velocity}} = 0$$

where

- q the species charge (± 1)
- m the particle mass
- $\mathbf{E}$ electric field, $\mathbf{B}$ magnetic field, which are **external**.

To take into account the **self-consistent electric field** $\longrightarrow$ coupling with the Poisson equation.

$$\begin{cases} \dfrac{\partial f_s}{\partial t} + \mathbf{v}\cdot\dfrac{\partial f_s}{\partial \mathbf{x}} + \dfrac{q}{m}(\mathbf{E}_{\text{sc}}+\mathbf{v}\times\mathbf{B})\cdot\dfrac{\partial f_s}{\partial \mathbf{v}} = 0, \\ \nabla\cdot\mathbf{E}_{\text{sc}} = \rho, \text{ where } \rho(t,\mathbf{x}) = q\displaystyle\int f_s(t,\mathbf{x},\mathbf{v})d\mathbf{v}. \\ f_s(0,\mathbf{x},\mathbf{v}) = f_0(\mathbf{x},\mathbf{v}). \end{cases}$$

Vlasov equation

- after Anatoly A. Vlasov (1908-1975)
- $f = f_s$ distribution function of particle species s.

$$\underbrace{\frac{\partial f_s}{\partial t}}_{\text{variations in \underline{time}}} + \underbrace{\mathbf{v}\cdot\frac{\partial f_s}{\partial \mathbf{x}}}_{\text{variations in \underline{space}}} + \underbrace{\frac{q}{m}(\mathbf{E}+\mathbf{v}\times\mathbf{B})\cdot\frac{\partial f_s}{\partial \mathbf{v}}}_{\text{variations en \underline{velocity}}} = 0$$

where

- q the species charge (± 1)
- m the particle mass
- $\mathbf{E}$ electric field, $\mathbf{B}$ magnetic field, which are **external**.

To take into account the **self-consistent electric field** $\longrightarrow$ coupling with the Poisson equation.

$$\begin{cases} \dfrac{\partial f_s}{\partial t} + \mathbf{v}\cdot\dfrac{\partial f_s}{\partial \mathbf{x}} + \dfrac{q}{m}(\mathbf{E}_{\rm sc}+\mathbf{v}\times\mathbf{B})\cdot\dfrac{\partial f_s}{\partial \mathbf{v}} = 0, \\ \nabla\cdot\mathbf{E}_{\rm sc} = \rho, \text{ where } \rho(t,\mathbf{x}) = q\displaystyle\int f_s(t,\mathbf{x},\mathbf{v})d\mathbf{v}. \\ f_s(0,\mathbf{x},\mathbf{v}) = f_0(\mathbf{x},\mathbf{v}). \end{cases}$$

Vlasov equation

- after Anatoly A. Vlasov (1908-1975)
- $f = f_s$ distribution function of particle species s.

$$\underbrace{\frac{\partial f_s}{\partial t}}_{\text{variations in \underline{time}}} + \underbrace{\mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}}}_{\text{variations in \underline{space}}} + \underbrace{\frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}}}_{\text{variations en \underline{velocity}}} = 0$$

where

- q the species charge (± 1)
- m the particle mass
- $\mathbf{E}$ electric field, $\mathbf{B}$ magnetic field, which are **external**.

To take into account the **self-consistent electric field** $\longrightarrow$ coupling with the Poisson equation.

$$\begin{cases} \frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q}{m}(\mathbf{E}_{\text{sc}} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0, \\ \nabla \cdot \mathbf{E}_{\text{sc}} = \rho, \text{ where } \rho(t, \mathbf{x}) = q \int f_s(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}. \\ f_s(0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v}). \end{cases}$$

Vlasov equation

- after Anatoly A. Vlasov (1908-1975)
- $f = f_s$ distribution function of particle species s.

$$\underbrace{\frac{\partial f_s}{\partial t}}_{\text{variations in \underline{time}}} + \underbrace{\mathbf{v}\cdot\frac{\partial f_s}{\partial \mathbf{x}}}_{\text{variations in \underline{space}}} + \underbrace{\frac{q}{m}(\mathbf{E}+\mathbf{v}\times\mathbf{B})\cdot\frac{\partial f_s}{\partial \mathbf{v}}}_{\text{variations en \underline{velocity}}} = 0$$

where

- q the species charge (± 1)
- m the particle mass
- $\mathbf{E}$ electric field, $\mathbf{B}$ magnetic field, which are **external**.

To take into account the **self-consistent electric field** $\longrightarrow$ coupling with the Poisson equation.

$$\begin{cases} \dfrac{\partial f_s}{\partial t} + \mathbf{v}\cdot\dfrac{\partial f_s}{\partial \mathbf{x}} + \dfrac{q}{m}(\mathbf{E}_{\text{sc}}+\mathbf{v}\times\mathbf{B})\cdot\dfrac{\partial f_s}{\partial \mathbf{v}} = 0, \\ \nabla\cdot\mathbf{E}_{\text{sc}} = \rho, \text{ where } \rho(t,\mathbf{x}) = q\displaystyle\int f_s(t,\mathbf{x},\mathbf{v})d\mathbf{v}. \\ f_s(0,\mathbf{x},\mathbf{v}) = f_0(\mathbf{x},\mathbf{v}). \end{cases}$$

Characteristics

The solution of the Vlasov equation can be expressed by the characteristics, solutions of the ODEs

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{V}, \\ \dfrac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} = \dfrac{q}{m}\big(\mathbf{E} + \mathbf{V} \times \mathbf{B}\big), \\ + \text{i.c.} \end{cases}$$

Then, the solution of the Vlasov equation writes

$$f(t, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{X}(0, \mathbf{x}, \mathbf{v}, t), \mathbf{V}(0, \mathbf{x}, \mathbf{v}, t)).$$

Outline

Parallelism in time

Initial-value problem

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = f(\mathbf{u}) \text{ in } [0, T_{\text{end}}],\ \mathbf{u}(0) = \mathbf{u}_0. \tag{1}$$

Build N time slices $[T_n, T_{n+1}]$ such that
$0 = T_0 < T_1 < \cdots < T_N = T_{\text{end}}$. Denote $\Delta t = T_{n+1} - T_n$.

Then, replace (1) by

$$\frac{\mathrm{d}\mathbf{u}_n}{\mathrm{d}t} = f(\mathbf{u}_n) \text{ in } [T_n, T_{n+1}],\ \mathbf{u}_n(T_n) = U_n, \quad n = 0, 1, \dots, N-1,$$

where the initial values $(U_n)_{n\in\{0,1,\dots,N-1\}}$ are to be found.

Propagator notation: $\mathbf{u}_n(T_{n+1}) = P(U_n)$, where $P(\cdot) = P(\cdot, \Delta t)$.

Parallelism in time

Initial-value problem

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = f(\mathbf{u}) \text{ in } [0, T_{\text{end}}], \ \mathbf{u}(0) = \mathbf{u}_0. \tag{1}$$

Build N time slices $[T_n, T_{n+1}]$ such that
$0 = T_0 < T_1 < \cdots < T_N = T_{\text{end}}$. Denote $\Delta t = T_{n+1} - T_n$.

Then, replace (1) by

$$\frac{\mathrm{d}\mathbf{u}_n}{\mathrm{d}t} = f(\mathbf{u}_n) \text{ in } [T_n, T_{n+1}], \ \mathbf{u}_n(T_n) = U_n, \quad n = 0, 1, \dots, N-1,$$

where the initial values $(U_n)_{n\in\{0,1,\dots,N-1\}}$ are to be found.

Propagator notation: $\mathbf{u}_n(T_{n+1}) = P(U_n)$, where $P(\cdot) = P(\cdot, \Delta t)$.

The problem

$(U_n)_n$ are such that $P(U_0) = U_1$, $P(U_1) = U_2, \ldots, P(U_{N-2}) = U_{N-1}$.

Reformulation: Denoting $U = (U_0, U_1, \ldots, U_{N-1})$ we have to solve

$$\text{find } \ U \ \text{ such that } \ \mathcal{F}(U) = 0,$$

where

$$\mathcal{F}(U) := \begin{pmatrix} U_0 - \mathbf{u}_0 \\ U_1 - P(U_0) \\ \vdots \\ U_{N-1} - P(U_{N-2}) \end{pmatrix}.$$

The problem

$(U_n)_n$ are such that $P(U_0) = U_1,\ P(U_1) = U_2, \dots, P(U_{N-2}) = U_{N-1}$.

Reformulation: Denoting $U = (U_0, U_1, \dots, U_{N-1})$ we have to solve

$$\text{find } U \text{ such that } \mathcal{F}(U) = 0,$$

where

$$\mathcal{F}(U) := \begin{pmatrix} U_0 - \mathbf{u}_0 \\ U_1 - P(U_0) \\ \vdots \\ U_{N-1} - P(U_{N-2}) \end{pmatrix}.$$

Solving with Newton method

For a given $U^0 \in \mathbb{R}^N$, iterate

$$U^{k+1} = U^k - \left[\mathcal{F}'(U^k)\right]^{-1}\mathcal{F}(U^k), \quad \text{for} \quad k = 0, 1, \ldots,$$

where $\mathcal{F}'$ is the Jacobian of $\mathcal{F}$ and

$$\mathcal{F}'(U) = \begin{pmatrix} I & & & & \\ -P'(U_0) & I & & & \\ & -P'(U_1) & I & & \\ & & \ddots & \ddots & \\ & & & -P'(U_{N-2}) & I \end{pmatrix}.$$

$=>$

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + P'(U_n^k)\big(U_n^{k+1} - U_n^k\big), \quad \text{for } n = 0, 1, \ldots, N-1. \end{cases}$$

Solving with Newton method

For a given $U^0 \in \mathbb{R}^N$, iterate

$$U^{k+1} = U^k - \left[\mathcal{F}'(U^k)\right]^{-1}\mathcal{F}(U^k), \quad \text{for} \quad k = 0, 1, \ldots,$$

where $\mathcal{F}'$ is the Jacobian of $\mathcal{F}$ and

$$\mathcal{F}'(U) = \begin{pmatrix} I & & & & \\ -P'(U_0) & I & & & \\ & -P'(U_1) & I & & \\ & & \ddots & \ddots & \\ & & & -P'(U_{N-2}) & I \end{pmatrix}.$$

=>

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + P'(U_n^k)\big(U_n^{k+1} - U_n^k\big), \quad \text{for } n = 0, 1, \ldots, N-1. \end{cases}$$

Solving with Newton method

For a given $U^0 \in \mathbb{R}^N$, iterate

$$U^{k+1} = U^k - \left[\mathcal{F}'(U^k)\right]^{-1} \mathcal{F}(U^k), \quad \text{for} \quad k = 0, 1, \dots,$$

where $\mathcal{F}'$ is the Jacobian of $\mathcal{F}$ and

$$\mathcal{F}'(U) = \begin{pmatrix} I & & & & \\ -P'(U_0) & I & & & \\ & -P'(U_1) & I & & \\ & & \ddots & \ddots & \\ & & & -P'(U_{N-2}) & I \end{pmatrix}.$$

$=>$

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + P'(U_n^k)\big(U_n^{k+1} - U_n^k\big), \quad \text{for } n = 0, 1, \dots, N-1. \end{cases}$$

Computing the terms $P'(U_n^k)$ can be too expensive.

Approximations:

- Use $P(U_n^{k+1}) - P(U_n^k) \approx P'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$
  $=> \ U_{n+1}^{k+1} = P(U_n^k)$.
- Use $P(U_n^{k+1}) - P(U_n^k) \approx G(U_n^{k+1}) - G(U_n^k)$, where G is a cheap approximation.

$$\Downarrow$$

Parareal iteration:

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + G(U_n^{k+1}) - G(U_n^k), \quad \text{for } n = 0, 1, \ldots, N-1, \end{cases}$$

- **Derivative Parareal alg.** $U_{n+1}^{k+1} = P(U_n^k) + G'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$

Computing the terms $P'(U_n^k)$ can be too expensive.

Approximations:

- Use $P(U_n^{k+1}) - P(U_n^k) \approx P'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$
  $=>\ U_{n+1}^{k+1} = P(U_n^k)$.
- Use $P(U_n^{k+1}) - P(U_n^k) \approx G(U_n^{k+1}) - G(U_n^k)$, where G is a cheap approximation.

$$\Downarrow$$

Parareal iteration:

$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + G(U_n^{k+1}) - G(U_n^k), \quad \text{for } n = 0, 1, \dots, N-1, \end{cases}$$

- **Derivative Parareal alg.** $U_{n+1}^{k+1} = P(U_n^k) + G'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$

Computing the terms $P'(U_n^k)$ can be too expensive.

Approximations:

- Use $P(U_n^{k+1}) - P(U_n^k) \approx P'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$
  $=> \; U_{n+1}^{k+1} = P(U_n^k)$.
- Use $P(U_n^{k+1}) - P(U_n^k) \approx G(U_n^{k+1}) - G(U_n^k)$, where G is a cheap approximation.

  $$\Downarrow$$

  Parareal iteration:

  $$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = P(U_n^k) + G(U_n^{k+1}) - G(U_n^k), \quad \text{for } n = 0, 1, \ldots, N-1, \end{cases}$$

- **Derivative Parareal alg.** $U_{n+1}^{k+1} = P(U_n^k) + G'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$

Computing the terms $P'(U_n^k)$ can be too expensive.

Approximations:

- Use $P(U_n^{k+1}) - P(U_n^k) \approx P'(U_n^k)\big(U_n^{k+1} - U_n^k\big)$
  $=>\ U_{n+1}^{k+1} = P(U_n^k)$.
- Use $P(U_n^{k+1}) - P(U_n^k) \approx G(U_n^{k+1}) - G(U_n^k)$, where G is a cheap approximation.

$$\Downarrow$$

Parareal iteration:

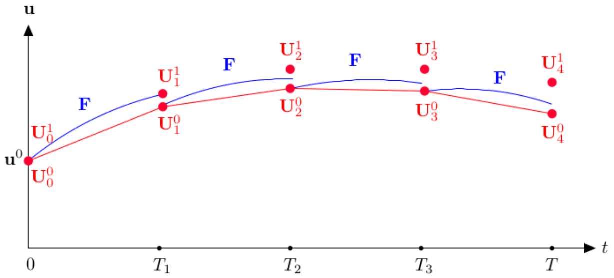
$$\begin{cases} U_0^{k+1} = \mathbf{u}_0, \\ U_{n+1}^{k+1} = F(U_n^k) + G(U_n^{k+1}) - G(U_n^k), \quad \text{for } n = 0, 1, \dots, N-1, \end{cases}$$

where $F(U_n^k)$ is an accurate approximation of $\mathbf{u}_n(T_{n+1})$ and $G(U_n^k)$ is a less accurate but cheaper approximation of $\mathbf{u}_n(T_{n+1})$.

F=the fine solver $\qquad\qquad$ G=the coarse solver.

Parareal algorithm

- First step
  $U^0_{n+1} = G(T_{n+1}, T_n, U^0_n)$ for $n = 0, 1, \dots, N-2$, $U^0_0 = \mathbf{u}_0$.
- Fix $k \in \{0, 1, \dots\}$. Assume $(U^k_n)_{n\in\{0,1,\dots,N-1\}}$ known. $U^{k+1}_0 = \mathbf{u}_0$.
  1. Compute in parallel $F(T_{n+1}, T_n, U^k_n)$.
  2. For $n = 0, 1, \dots, N-1$ do
     $$U^{k+1}_{n+1} = G(T_{n+1}, T_n, U^{k+1}_n) + F(T_{n+1}, T_n, U^k_n) - G(T_{n+1}, T_n, U^k_n).$$



from M. Gander's paper.

Bibliography

Parareal algorithm:

- Original paper: J.-L. Lions, Y. Maday, G. Turinici: Comptes Rendus de l'Acad. des Sciences Paris Sér. I Math., vol. 332, 2001
- Interpretation in terms of approximate Newton's iterations and convergence result for a linear ODE: M. J. Gander, S. Vandewalle: SISC, vol. 29, No. 2, 2007
- Convergence result for Hamiltonian differential equations M. J. Gander, E. Hairer : Journal of Computational and Applied Mathematics, Vol. 259, 2014

Efficiency

Finite step convergence

The outcome of the parareal algorithm verifies

$$U_n^k = F(T_n, 0, \mathbf{u}_0) \text{ when } k \geq n.$$

However

Goal: Speed up the simulation if

- Cost$(G) \ll$ Cost(F)
- achieve convergence for

$$K \ll N.$$

Efficiency

Finite step convergence

The outcome of the parareal algorithm verifies

$$U_n^k = F(T_n, 0, \mathbf{u}_0) \text{ when } k \geq n.$$

However

Goal: Speed up the simulation if

- $\text{Cost}(G) \ll \text{Cost}(F)$
- achieve convergence for
$$K \ll N.$$

Theoretic Speedup

Computing $(F(T_{n+1}, T_n, U_n^k))_{n=0,\dots,N-1}$ in parallel over N processors.

The total time of the parareal run is

$$T_{\mathrm{par}} = T_{\mathrm{init}} + K\Big(\frac{T_{\mathrm{fine}}}{N} + T_{\mathrm{coarse}}\Big),$$

where K is the number of parareal iterations leading to the target error.

Thus $$\mathbf{S}(N) = \frac{T_{\mathrm{fine}}}{T_{\mathrm{par}}} = \frac{1}{\Big(1+K\Big)\frac{T_c}{T_f} + \frac{K}{N}}$$

where $T_{\mathrm{fine}} = NT_f$, $T_{\mathrm{coarse}} = NT_c$.

T_f is the computation time of the fine solver.

AIM: $KT_c/T_f \ll 1$ and $K \ll N$.

Theoretic Speedup

Computing $(F(T_{n+1}, T_n, U_n^k))_{n=0,\dots,N-1}$ in parallel over N processors.

The total time of the parareal run is

$$T_{\text{par}} = T_{\text{init}} + K\Big(\frac{T_{\text{fine}}}{N} + T_{\text{coarse}}\Big),$$

where K is the number of parareal iterations leading to the target error.

Thus $$\mathbf{S}(N) = \frac{T_{\text{fine}}}{T_{\text{par}}} = \frac{1}{\Big(1+K\Big)\frac{T_c}{T_f} + \frac{K}{N}}$$

where $T_{\text{fine}} = NT_f$, $T_{\text{coarse}} = NT_c$.

T_f is the computation time of the fine solver.

AIM: $KT_c/T_f \ll 1$ and $K \ll N$.

The strategy

Question: What choice for the coarse solver G?

Standard choices:

- G = approximation scheme of F solver but with a larger time step
- G = different approximation scheme than F's, with lower accuracy

Nice examples in M. J. Gander and E. Hairer. *Nonlinear convergence analysis for the parareal algorithm*. Lecture Notes in Computational Science and Engineering, 2008.

- Brusselator eq. (speedup of 8 for $N = 32$)
- Arenstorf orbit (speedup of 62 for $N = 250$)
- Lorenz eq. (speedup of 18 for $N = 180$)

Choice: 4th order Runge-Kutta method for both solvers, with $\Delta t \gg \delta t$.

The strategy

Question: What choice for the coarse solver G?

Standard choices:

- $G =$ approximation scheme of F solver but with a larger time step
- $G =$ different approximation scheme than F's, with lower accuracy

Nice examples in M. J. Gander and E. Hairer. *Nonlinear convergence analysis for the parareal algorithm*. Lecture Notes in Computational Science and Engineering, 2008.

- Brusselator eq. (speedup of 8 for $N = 32$)
- Arenstorf orbit (speedup of 62 for $N = 250$)
- Lorenz eq. (speedup of 18 for $N = 180$)

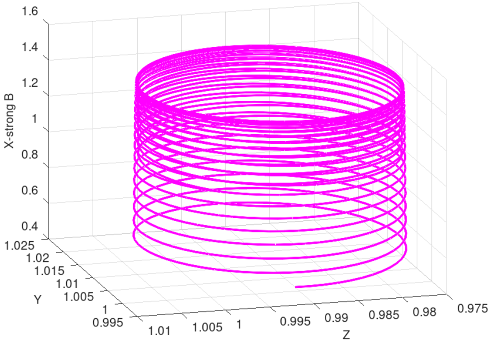
Choice: 4th order Runge-Kutta method for both solvers, with $\Delta t \gg \delta t$.

Charged particle example

For $\varepsilon = 0.01$ solve

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}, & \mathbf{x}(0) = \mathbf{x}_0, \\ \dfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \dfrac{1}{\varepsilon}\mathbf{v} \times \mathbf{e}_1 + \mathbf{E}(\mathbf{x}), & \mathbf{v}(0) = \mathbf{v}_0, \end{cases} \tag{2}$$

where $\mathbf{v} \times \mathbf{e}_1 = (0, v_3, -v_2)^T$. Take $\mathbf{E}(\mathbf{x}) = (-x_1, 0, 0)^T$.
Initial condition $\mathbf{x}_0 = \mathbf{v}_0 = (1, 1, 1)^T$ and $T_{\text{end}} = 2$.



Using standard Parareal

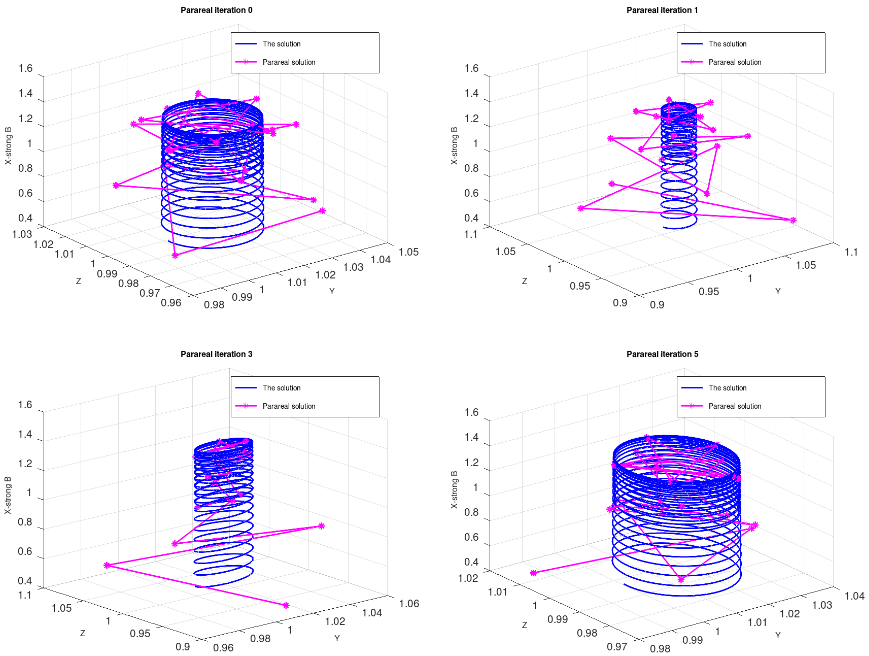
Take $N = 20$ time windows for $[0, T_{\text{end}}]$.

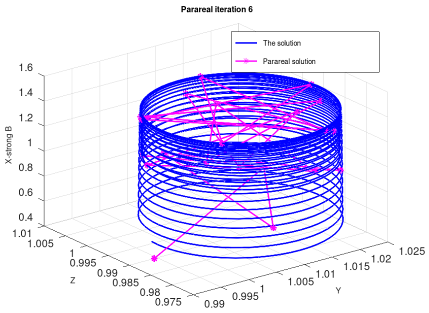
Fine solver F is 2nd order Runge-Kutta with $\delta t = T_{\text{end}}/1800$.
Coarse solver G is 2nd order Runge-Kutta with $\Delta t = T_{\text{end}}/600$.

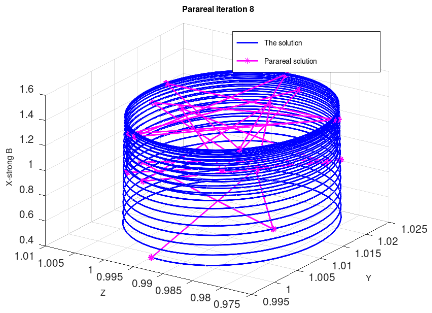
$=> T_c/T_f = 1/3$.

This strategy needs $K = 8$ parareal iterations for a sufficiently small error.

Speedup is $\mathbf{S} \sim 0.3$.


Parareal iteration 0
Parareal iteration 1
Parareal iteration 3
Parareal iteration 5
The solution
Parareal solution
X-strong B
Z
Y



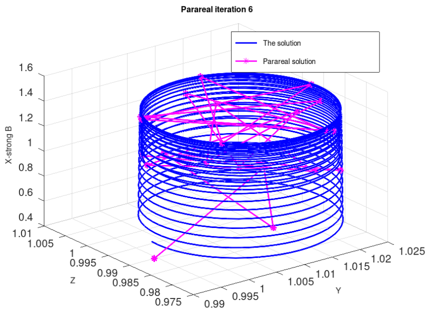


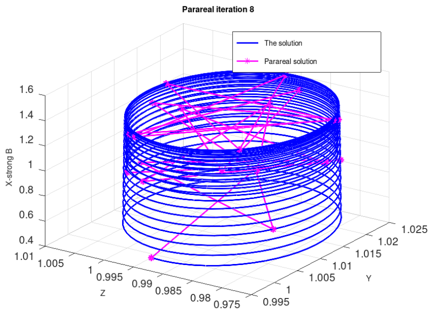
If we lower the accuracy of G to $\Delta t = T_{\text{end}}/400$

$$=> T_f/T_c = 4.5$$

we need more parareal iterations for the similar target error, $K = 15$.

Speedup is then $\mathbf{S} = 0.23$.





If we lower the accuracy of G to $\Delta t = T_{\text{end}}/400$

$$=> T_f/T_c = 4.5$$

we need more parareal iterations for the similar target error, $K = 15$.

Speedup is then $\mathbf{S} = 0.23$.

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\ &= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\ &= (F-G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}\end{aligned}$$

Assumptions:

- truncation error of G : $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G : $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\ &= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\ &= (F-G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}\end{aligned}$$

Assumptions:

- truncation error of G: $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G: $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}
\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\
&= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\
&= (F - G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}
\end{aligned}$$

Assumptions:

- truncation error of G : $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G : $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}
\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\
&= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\
&= (F - G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}
\end{aligned}$$

Assumptions:

- truncation error of G : $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G : $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\ &= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\ &= (F-G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}\end{aligned}$$

Assumptions:

- truncation error of G : $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G : $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Recall $\mathbf{u}$ is the solution to (1). Assume F is exact.

$$\begin{aligned}\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1} &= F\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k \\ &= F\mathbf{u}(T_n) - G\mathbf{u}(T_n) - FU_n^k - GU_n^{k+1} + GU_n^k + G\mathbf{u}(T_n) \\ &= (F-G)(\mathbf{u}(T_n) - U_n^k) + G\mathbf{u}(T_n) - GU_n^{k+1}\end{aligned}$$

Assumptions:

- truncation error of G : $\|(F-G)x\| \leq C_1(\Delta t)^{p+1}\|x\|$
- Lipschitz property for G : $\|Gx - Gy\| \leq (1 + C_2\Delta t)\|x - y\|$

Then

$$\|\mathbf{u}(T_{n+1}) - U_{n+1}^{k+1}\| \leq C_1(\Delta t)^{p+1}\|\mathbf{u}(T_n) - U_n^k\| + (1 + C_2\Delta t)\|\mathbf{u}(T_n) - U_n^{k+1}\|.$$

Error estimation

Theorem

Under these assumptions

$$\|\mathbf{u}(T_n) - U_n^k\| \leq \frac{(C_1(\Delta t)^{p+1})^{k+1}}{(k+1)!}(1 + C_2\Delta t)^{n-k-1}\prod_{j=0}^{k}(n-j)$$
$$\leq \frac{(C_1 T_n)^{k+1}}{(k+1)!} e^{C_2(T_n - T_{k+1})}(\Delta t)^{p(k+1)}.$$

Details in M. J. Gander and E. Hairer. *Nonlinear convergence analysis for the parareal algorithm*. Domain Decomposition Methods in Science and Engineering XVII, vol. 60 of Lecture Notes in Computational Science and Engineering, pag. 45–56, 2008.

Error estimation

Theorem

Under these assumptions

$$\|\mathbf{u}(T_n) - U_n^k\| \leq \frac{(C_1(\Delta t)^{p+1})^{k+1}}{(k+1)!}(1 + C_2\Delta t)^{n-k-1}\prod_{j=0}^{k}(n-j)$$
$$\leq \frac{(C_1 T_n)^{k+1}}{(k+1)!} e^{C_2(T_n - T_{k+1})}(\Delta t)^{p(k+1)}.$$

Details in M. J. Gander and E. Hairer. *Nonlinear convergence analysis for the parareal algorithm*. Domain Decomposition Methods in Science and Engineering XVII, vol. 60 of Lecture Notes in Computational Science and Engineering, pag. 45–56, 2008.

Outlook

" ... also the coarse integrator should have a certain accuracy. Otherwise the convergence of the parareal iterations would be too slow, and the time window, where the algorithm can be applied, would be rather small preventing an efficient integration. "

from M. J. Gander, E. Hairer : Journal of Computational and Applied Mathematics, Vol. 259, 2014.

Outline

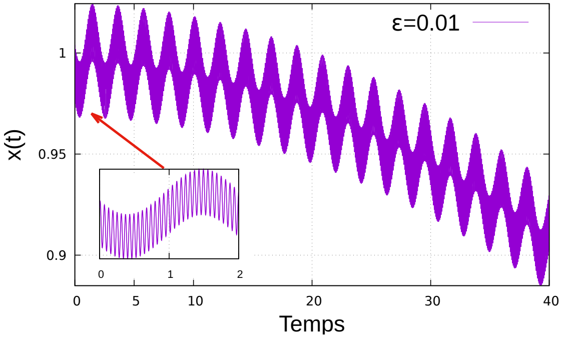
General problem

Solve

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = f(t,\mathbf{u}) \ \text{ in } [0,T_{\text{end}}],\ \mathbf{u}(0)=\mathbf{u}_0,$$

where the unknown is $\mathbf{u}\equiv\mathbf{u}(t)$, $\mathbf{u}:[0,T_{\text{end}}]\to\mathbb{R}^n$, where $n\in\mathbb{N}, n\geq 2$.
f is given and is Lipschitz continuous in $\mathbf{u}$ etc.

Stiff equation: highly oscillatory case. The solution evolves at several (different) time scales.



General problem – Motivation

no analytic solution $\Rightarrow$ numerical solving of the ODE.

- explicit methods lack stability $\rightsquigarrow$ **tiny** time steps.
- implicit methods: don't need small time steps but still not accurate.

Goals:

1. High accuracy $\longleftrightarrow$ the method needs to resolve **all** the oscillations in the solution
2. Long simulations (millions of time steps) $\rightsquigarrow$ high computational cost $\rightsquigarrow$ numerical inefficiency.

AIM: time schemes for solving accurately and efficiently stiff ODEs.

General problem – Motivation

no analytic solution $\Rightarrow$ numerical solving of the ODE.

- explicit methods lack stability $\rightsquigarrow$ **tiny** time steps.
- implicit methods: don't need small time steps but still not accurate.

Goals:

1. High accuracy $\hookleftarrow$ the method needs to resolve **all** the oscillations in the solution
2. Long simulations (millions of time steps) $\rightsquigarrow$ high computational cost $\rightsquigarrow$ numerical inefficiency.

AIM: time schemes for solving <u>accurately</u> and <u>efficiently</u> stiff ODEs.

General problem – Motivation

no analytic solution $\Rightarrow$ numerical solving of the ODE.

- explicit methods lack stability $\rightsquigarrow$ **tiny** time steps.
- implicit methods: don't need small time steps but still not accurate.

Goals:

1. High accuracy $\hookleftarrow$ the method needs to resolve **all** the oscillations in the solution
2. Long simulations (millions of time steps) $\rightsquigarrow$ high computational cost $\rightsquigarrow$ numerical inefficiency.

AIM: time schemes for solving $\underline{\text{accurately}}$ and $\underline{\text{efficiently}}$ stiff ODEs.

Specific problem

Solve for $0 < \varepsilon \ll 1$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{1}{\varepsilon} L\mathbf{u} + N(\mathbf{u}), \qquad \mathbf{u}(0) = \mathbf{u}_0,$$

where L is a skew-Hermitian matrix with imaginary eigenvalues of large modulus and N is a nonlinear operator.

AIM: Solve the equation with a method which is not constraint by ε.

Possible strategies:

1. Infer a limit model when $\varepsilon \to 0$, that can accurately be solved with large time steps.
2. Use a parallel in time method.

Specific problem

Solve for $0 < \varepsilon \ll 1$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{1}{\varepsilon} L\mathbf{u} + N(\mathbf{u}), \qquad \mathbf{u}(0) = \mathbf{u}_0,$$

where L is a skew-Hermitian matrix with imaginary eigenvalues of large modulus and N is a nonlinear operator.

AIM: Solve the equation with a method <u>which is not constraint</u> by ε.

Possible strategies:

1. Infer a limit model when $\varepsilon \to 0$, that can accurately be solved with large time steps.
2. Use a parallel in time method.

Specific problem

Solve for $0 < \varepsilon \ll 1$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{1}{\varepsilon} L\mathbf{u} + N\Big(\frac{t}{\varepsilon}, \mathbf{u}\Big), \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{3}$$

where L is a skew-Hermitian matrix with imaginary eigenvalues of large modulus and N is a nonlinear operator.

AIM: Solve the equation with a method which is not constraint by ε.

Possible strategies:

1. Infer a limit model when $\varepsilon \to 0$, that can accurately be solved with large time steps.
2. Use parallel in time method.

Application - Newton equations

Study the dynamics of charged particles (ions and free electrons) in electromagnetic fields.

Consider in (3) $\mathbf{u} = (\mathbf{x}, \mathbf{v})$.

Equations of motion for $\mathbf{x}(t)$=position, $\mathbf{v}(t)$=velocity

Solve for $0 < \varepsilon \ll 1$

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}, & \mathbf{x}(0) = \mathbf{x}_0, \\ \dfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \dfrac{1}{\varepsilon}\mathbf{v} \times \mathbf{B} + \mathbf{E}\Big(\dfrac{t}{\varepsilon}, \mathbf{x}\Big), & \mathbf{v}(0) = \mathbf{v}_0, \end{cases}$$

where

- $(\mathbf{x}_0, \mathbf{v}_0) \in \mathbb{R}^6$ is an initial condition at the initial time $t = 0$.
- $\mathbf{B} \in \mathbb{R}^3$ is a given constant magnetic field, $\mathbf{B} = \mathbf{e}_1 = (1, 0, 0)^T$.
- $\mathbf{E} : \mathbb{R}^+ \times \mathbb{R}^3 \to \mathbb{R}^3$ is the electric field, 2π-periodic in τ.

Application - Newton equations

Study the dynamics of charged particles (ions and free electrons) in electromagnetic fields.

Consider in (3) $\mathbf{u} = (\mathbf{x}, \mathbf{v})$.

Equations of motion for $\mathbf{x}(t)$=position, $\mathbf{v}(t)$=velocity

Solve for $0 < \varepsilon \ll 1$

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}, & \mathbf{x}(0) = \mathbf{x}_0, \\ \dfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \dfrac{1}{\varepsilon}\mathbf{v} \times \mathbf{B} + \mathbf{E}\Big(\dfrac{t}{\varepsilon}, \mathbf{x}\Big), & \mathbf{v}(0) = \mathbf{v}_0, \end{cases}$$

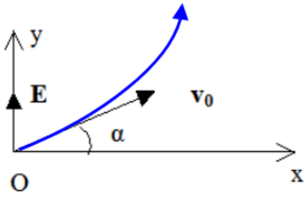
where

- $(\mathbf{x}_0, \mathbf{v}_0) \in \mathbb{R}^6$ is an initial condition at the initial time $t = 0$.
- $\mathbf{B} \in \mathbb{R}^3$ is a given constant magnetic field, $\mathbf{B} = \mathbf{e}_1 = (1, 0, 0)^T$.
- $\mathbf{E} : \mathbb{R}^+ \times \mathbb{R}^3 \to \mathbb{R}^3$ is the electric field, 2π-periodic in τ.

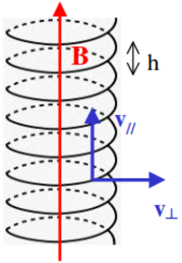
Dynamics of a particle

A charged particle ($q = \pm 1$) with velocity $\mathbf{v}$
in an electric field $\mathbf{E}$ and a magnetic field $\mathbf{B}$ undergoes the Lorentz force

$$\boxed{\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}).}$$



The electric force **accelerates** (or **slows down**) the charge.



The magnetic force **deflects** the charge.

Application - II

Previous system is of the form (3), $\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \frac{1}{\varepsilon} L \begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} + N\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix}$

by taking

$$L = \begin{pmatrix} O_3 & O_3 \\ O_3 & l \end{pmatrix} \quad \text{where } \; l = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

and

$$N(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{E}(\mathbf{x}) \end{pmatrix}.$$

Outline

Toward averaged model I

Consider **only the rapid term** equation

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{1}{\varepsilon} L\mathbf{u} \ \text{ in } [0, T_{\text{end}}], \qquad \mathbf{u}(0) = \mathbf{u}_0,$$

to which the solution is

$$\mathbf{u}(t) = e^{(t/\varepsilon)L}\mathbf{u}_0,$$

where for any matrix A, we have $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$.

Remark: In the example above, we can compute

$$e^{tL} = \begin{pmatrix} I_3 & O_3 \\ O_3 & R(t) \end{pmatrix} \quad \text{where} \quad R(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix}.$$

Toward averaged model I

Consider **only the rapid term** equation

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{1}{\varepsilon} L\mathbf{u} \ \text{ in } [0, T_{\mathrm{end}}], \qquad \mathbf{u}(0) = \mathbf{u}_0,$$

to which the solution is

$$\mathbf{u}(t) = e^{(t/\varepsilon)L}\mathbf{u}_0,$$

where for any matrix A, we have $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$.

Remark: In the example above, we can compute

$$e^{tL} = \begin{pmatrix} I_3 & O_3 \\ O_3 & R(t) \end{pmatrix} \quad \text{where} \quad R(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix}.$$

Toward averaged model II

More generally, assuming that $\underline{e^{tL} \text{ is easy to compute}}$, we use the change of variable

$$\mathbf{w}(t) := e^{-\frac{t}{\varepsilon}L}\mathbf{u}(t)$$

and thus, $\mathbf{w}$ is the solution to

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = e^{-\frac{t}{\varepsilon}L}N\big(e^{\frac{t}{\varepsilon}L}\mathbf{w}\big), \qquad \mathbf{w}(0) = \mathbf{u}_0.$$

Remark: $\underline{\text{no singular term}}$ but time oscillations are still present.

Toward averaged model II

More generally, assuming that $\underline{e^{tL}\text{ is easy to compute}}$, we use the change of variable

$$\mathbf{w}(t) := e^{-\frac{t}{\varepsilon}L}\mathbf{u}(t)$$

and thus, $\mathbf{w}$ is the solution to

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = e^{-\frac{t}{\varepsilon}L}N\big(e^{\frac{t}{\varepsilon}L}\mathbf{w}\big), \qquad \mathbf{w}(0) = \mathbf{u}_0.$$

Remark: $\underline{\text{no singular term}}$ but time oscillations are still present.

Toward averaged model III

The equation is of the form $$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = \mathcal{N}\Big(\frac{t}{\varepsilon}, \mathbf{w}\Big).$$

Theorem (Sanders-Verhulst, 1985 – periodic case)

If $\mathcal{N} : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ satisfies

- λ-Lipschitz continuous in $\mathbf{w}$,
- continuous over $[0, T_{\mathrm{end}}] \times D$ where $D \subset \mathbb{R}^n$ bounded,
- η-periodic in t,
- $\sup_{\mathbf{w}\in D} \sup_{t\in[0,1]} |\mathcal{N}(t, \mathbf{w})| < \infty$ is ε-independent.

Then we consider the averaged model

$$\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} = \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds, \qquad \overline{\mathbf{w}}(0) = \mathbf{u}_0.$$

which solution is assumed to be bounded over $[0, 1]$.

Then, we have for some constant $K > 0$

$$|\mathbf{w}(t) - \overline{\mathbf{w}}(t)| < K\varepsilon\eta\, e^{\lambda t} \quad \forall t \in [0, 1].$$

Toward averaged model III

The equation is of the form $\dfrac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = \mathcal{N}\Big(\dfrac{t}{\varepsilon}, \mathbf{w}\Big).$

Theorem (Sanders-Verhulst, 1985 – periodic case)

If $\mathcal{N} : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ satisfies

- λ-Lipschitz continuous in $\mathbf{w}$,
- continuous over $[0, T_{\text{end}}] \times D$ where $D \subset \mathbb{R}^n$ bounded,
- η-periodic in t,
- $\sup_{\mathbf{w}\in D} \sup_{t\in[0,1]} |\mathcal{N}(t, \mathbf{w})| < \infty$ is ε-independent.

Then we consider the <u>averaged</u> model

$$\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} = \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds, \qquad \overline{\mathbf{w}}(0) = \mathbf{u}_0.$$

which solution is assumed to be bounded over $[0, 1]$.

Then, we have for some constant $K > 0$

$$|\mathbf{w}(t) - \overline{\mathbf{w}}(t)| < K\varepsilon\eta\ e^{\lambda t} \quad \forall t \in [0, 1].$$

The non-periodic case

Theorem

If $\mathcal{N} : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ satisfies

- λ-Lipschitz continuous in $\mathbf{w}$,
- continuous over $[0, T_{\text{end}}] \times D$ where $D \subset \mathbb{R}^n$ bounded,
- $\sup\limits_{\mathbf{w} \in D} \sup\limits_{t \in [0,1]} |\mathcal{N}(t, \mathbf{w})| < \infty$ is ε-independent.

Then we consider the local averaged model

$$\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} = \frac{1}{\eta} \int_0^\eta \mathcal{N}\Big(\frac{t}{\varepsilon} + s, \overline{\mathbf{w}}\Big) ds, \qquad \overline{\mathbf{w}}(0) = \mathbf{u}_0.$$

which solution is assumed to be bounded over $[0, 1]$.

Then, we have for some constant $K > 0$

$$|\mathbf{w}(t) - \overline{\mathbf{w}}(t)| < K\varepsilon\eta \; e^{\lambda t} \quad \forall t \in [0, 1].$$

Charged particles - Example 1

In the canonical frame of $\mathbb{R}^3$ denote $\mathbf{x} = (x_1, x_2, x_3)^T$.

For $\varepsilon = 0.05$ solve

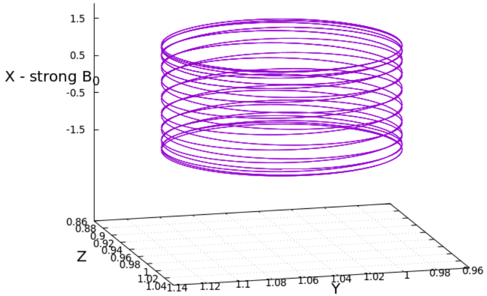
$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}, & \mathbf{x}(0) = \mathbf{x}_0, \\[2ex] \dfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \dfrac{1}{\varepsilon}\mathbf{v} \times \mathbf{e}_1 + \mathbf{E}(\mathbf{x}), & \mathbf{v}(0) = \mathbf{v}_0, \end{cases} \tag{4}$$

where $\mathbf{v} \times \mathbf{e}_1 = (0, v_3, -v_2)^T$.

$\boxed{\text{Take } \mathbf{E}(\mathbf{x}) = (-x_1, 0, 0)^T}$.

Then (4) has an explicit solution.

$\mathbf{x}_0 = \mathbf{v}_0 = (1, 1, 1)^T$ and $T_{\text{end}} = 20$.

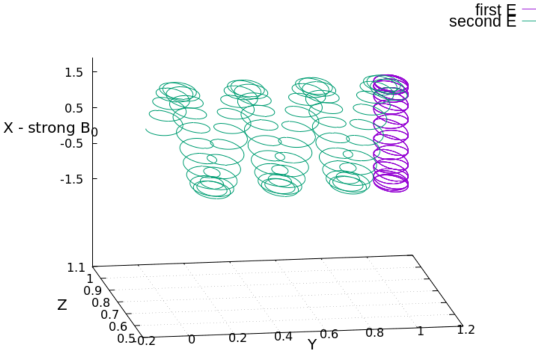


Charged particles - Example 2

$$\boxed{\text{Take } \mathbf{E}(\mathbf{x}) = (-x_1, 0, -x_3)^T}$$

Then (4) has an explicit solution.

$\mathbf{x}_0 = \mathbf{v}_0 = (1, 1, 1)^T$ and $T_{\text{end}} = 20$.



Charged particles - Example 3

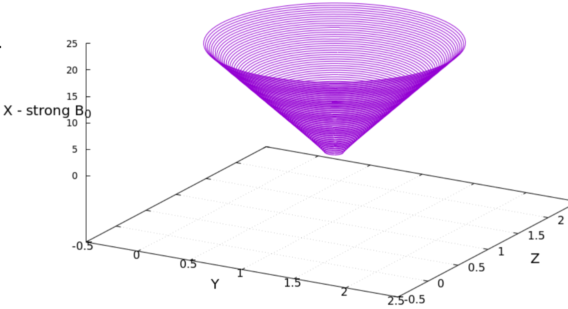
For $\varepsilon = 0.05$ solve

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}, & \mathbf{x}(0) = \mathbf{x}_0, \\ \dfrac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \dfrac{1}{\varepsilon}\mathbf{v} \times \mathbf{e}_1 + \mathbf{E}\Big(\dfrac{t}{\varepsilon}\Big), & \mathbf{v}(0) = \mathbf{v}_0. \end{cases} \tag{5}$$

$\boxed{\text{Take } \mathbf{E}(\tau) = (0, \sin\tau, \cos\tau)^T}$.

Then (5) has an explicit solution.

$\mathbf{x}_0 = \mathbf{v}_0 = (1, 1, 1)^T$ and $T_{\text{end}} = 20$.



Examples - Averaged model

Denoting $\overline{\mathbf{w}} = (\mathbf{X}^0, \mathbf{V}^0)$, recall that for the i.c. $\overline{\mathbf{w}}(0) = \mathbf{u}_0$

$$\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} = \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds$$
$$= \frac{1}{\eta}\int_0^\eta e^{-sL}N(e^{sL}\overline{\mathbf{w}})ds$$
$$= \dots$$

Specifying, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{X}^0\\ \mathbf{V}^0\end{pmatrix} = \frac{1}{2\pi}\int_0^{2\pi}\begin{pmatrix}R(s)\mathbf{V}^0\\ R(-s)\mathbf{E}(s, \mathbf{X}^0)\end{pmatrix}ds$$

$$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T\\ (\mathbf{E}_1(\mathbf{X}^0), 0, 0)^T\end{pmatrix} \qquad \text{for examples 1 and 2}$$

$$\text{or} \quad = \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T\\ (0, 0, 1)^T\end{pmatrix} \qquad \text{for example 3}$$

Examples - Averaged model

Denoting $\overline{\mathbf{w}} = (\mathbf{X}^0, \mathbf{V}^0)$, recall that for the i.c. $\overline{\mathbf{w}}(0) = \mathbf{u}_0$

$$\begin{aligned}\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} &= \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds \\ &= \frac{1}{\eta}\int_0^\eta e^{-sL} N(e^{sL}\overline{\mathbf{w}})ds \\ &= \ldots\end{aligned}$$

Specifying, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{X}^0 \\ \mathbf{V}^0\end{pmatrix} = \frac{1}{2\pi}\int_0^{2\pi}\begin{pmatrix}R(s)\mathbf{V}^0 \\ R(-s)\mathbf{E}(s, \mathbf{X}^0)\end{pmatrix} ds$$

$$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T \\ (\mathbf{E}_1(\mathbf{X}^0), 0, 0)^T\end{pmatrix}$$ for examples 1 and 2

or $$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T \\ (0, 0, 1)^T\end{pmatrix}$$ for example 3

Examples - Averaged model

Denoting $\overline{\mathbf{w}} = (\mathbf{X}^0, \mathbf{V}^0)$, recall that for the i.c. $\overline{\mathbf{w}}(0) = \mathbf{u}_0$

$$\begin{aligned}\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} &= \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds \\ &= \frac{1}{\eta}\int_0^\eta e^{-sL} N(e^{sL}\overline{\mathbf{w}})ds \\ &= \ldots\end{aligned}$$

Specifying, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{X}^0\\ \mathbf{V}^0\end{pmatrix} = \frac{1}{2\pi}\int_0^{2\pi}\begin{pmatrix}R(s)\mathbf{V}^0\\ R(-s)\mathbf{E}(s, \mathbf{X}^0)\end{pmatrix} ds$$

$$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T\\ (\mathbf{E}_1(\mathbf{X}^0), 0, 0)^T\end{pmatrix}$$ for examples 1 and 2

or $$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T\\ (0, 0, 1)^T\end{pmatrix}$$ for example 3

Examples - Averaged model

Denoting $\overline{\mathbf{w}} = (\mathbf{X}^0, \mathbf{V}^0)$, recall that for the i.c. $\overline{\mathbf{w}}(0) = \mathbf{u}_0$

$$\begin{aligned}\frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} &= \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds \\ &= \frac{1}{\eta}\int_0^\eta e^{-sL}N(e^{sL}\overline{\mathbf{w}})ds \\ &= \ldots\end{aligned}$$

Specifying, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{X}^0 \\ \mathbf{V}^0\end{pmatrix} = \frac{1}{2\pi}\int_0^{2\pi}\begin{pmatrix}R(s)\mathbf{V}^0 \\ R(-s)\mathbf{E}(s, \mathbf{X}^0)\end{pmatrix}ds$$

$$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T \\ (\mathbf{E}_1(\mathbf{X}^0), 0, 0)^T\end{pmatrix}$$ for examples 1 and 2

or $$= \begin{pmatrix}(\mathbf{V}_1^0, 0, 0)^T \\ (0, 0, 1)^T\end{pmatrix}$$ for example 3

Examples - Averaged model

Denoting $\overline{\mathbf{w}} = (\mathbf{X}^0, \mathbf{V}^0)$, recall that for the i.c. $\overline{\mathbf{w}}(0) = \mathbf{u}_0$

$$\begin{aligned} \frac{\mathrm{d}\overline{\mathbf{w}}}{\mathrm{d}t} &= \frac{1}{\eta}\int_0^\eta \mathcal{N}(s, \overline{\mathbf{w}})ds \\ &= \frac{1}{\eta}\int_0^\eta e^{-sL} N(e^{sL}\overline{\mathbf{w}})ds \\ &= \dots \end{aligned}$$

Specifying, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{X}^0 \\ \mathbf{V}^0\end{pmatrix} = \frac{1}{2\pi}\int_0^{2\pi}\begin{pmatrix} R(s)\mathbf{V}^0 \\ R(-s)\mathbf{E}(s, \mathbf{X}^0)\end{pmatrix} ds$$

$$= \begin{pmatrix} (\mathbf{V}_1^0, 0, 0)^T \\ (\mathbf{E}_1(\mathbf{X}^0), 0, 0)^T \end{pmatrix} \qquad \text{for examples 1 and 2}$$

$$\text{or} \quad = \begin{pmatrix} (\mathbf{V}_1^0, 0, 0)^T \\ (0, 0, 1)^T \end{pmatrix} \qquad \text{for example 3}$$

The theorem can be applied

Result: The Lipschitz condition is satisfied for the 3 examples.

We have the error estimation

$$|\mathbf{w}(t) - \overline{\mathbf{w}}(t)| < K\varepsilon\eta\, e^{\lambda t} \quad \forall t \in [0,1].$$

However, the averaged model is not accurate with respect to the stiff ODE. (It captures only the motion along the e_1 axis.)

The theorem can be applied

Result: The Lipschitz condition is satisfied for the 3 examples.

We have the error estimation

$$|\mathbf{w}(t) - \overline{\mathbf{w}}(t)| < K\varepsilon\eta\, e^{\lambda t} \quad \forall t \in [0,1].$$

However, the averaged model is not accurate with respect to the stiff ODE. (It captures only the motion along the $\mathbf{e}_1$ axis.)

Conclusion

- In long times $\left(\sim \frac{t}{\varepsilon}\right)$ averaged models are not accurate and fail to approximate the original equation.
- The Parareal strategy allows to correct this error efficiently (in a few iterations).

Charged particle - Example 1

Take $N = 20$ time windows for $[0, T_{\mathrm{end}}]$ and $T_{\mathrm{end}} = 2$.

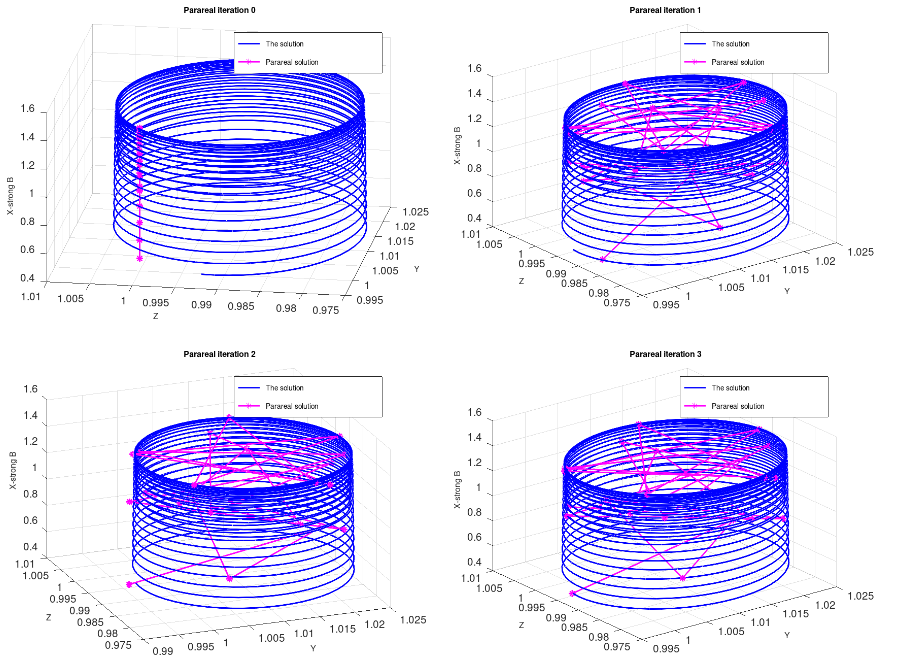
Fine solver F is 2nd order Runge-Kutta with $\delta t = T_{\mathrm{end}}/1800$.

Coarse solver G is 2nd order Runge-Kutta scheme
for **the averaged model**
with $\Delta t = T_{\mathrm{end}}/N$.

$=> T_f/T_c = 90$.

$K = 3$ parareal iterations are enough for the target accuracy.

Speedup is $\mathbf{S} \sim 5$.


Parareal iteration 0
Parareal iteration 1
Parareal iteration 2
Parareal iteration 3
The solution
Parareal solution
X-strong B
Y
Z
0.4
0.6
0.8
1
1.2
1.4
1.6
1.01
1.005
1
0.995
0.99
0.985
0.98
0.975
1.025
1.02
1.015
1.01
1.005

Outlook

- Parareal method with the strategy of using averaged model for the coarse solver is efficient in computational cost.

- need for finding the reduced model (not always an easy task).
- derive estimate for the error of the reduced model.
- averaged model is not always sufficiently accurate. First-order averaged models are to be derived.

Outlook

- Parareal method with the strategy of using averaged model for the coarse solver is efficient in computational cost.

- need for finding the reduced model (not always an easy task).
- derive estimate for the error of the reduced model.
- averaged model is not always sufficiently accurate. First-order averaged models are to be derived.

END part I

Thank you!