A domain decomposition approach to exponential methods for PDEs

Luca Bonaventura

MOX - Politecnico di Milano

Boulder, 8.04.2014

Outline of the talk

Outline of the talk

- Short review of exponential integrators

Outline of the talk

- Short review of exponential integrators
- An accuracy and efficiency assessment of simple approaches to their application

Outline of the talk

- **Short review of exponential integrators**
- **An accuracy and efficiency assessment of simple approaches to their application**
- **Local Exponential Methods:**
  a domain decomposition approach to exponential methods

Outline of the talk

- Short review of exponential integrators
- An accuracy and efficiency assessment of simple approaches to their application
- Local Exponential Methods:
  a domain decomposition approach to exponential methods
- Some preliminary numerical results

Outline of the talk

- Short review of exponential integrators
- An accuracy and efficiency assessment of simple approaches to their application
- Local Exponential Methods:
  a domain decomposition approach to exponential methods
- Some preliminary numerical results
- Conclusions and perspectives for atmospheric modelling

Basic idea of exponential methods

Basic idea of exponential methods

- **Cauchy problem for nonhomogeneous linear ODE system:**

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{g}(t) \qquad \mathbf{u}(0) = \mathbf{u}_0$$

Basic idea of exponential methods

- **Cauchy problem for nonhomogeneous linear ODE system:**

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{g}(t) \qquad \mathbf{u}(0) = \mathbf{u}_0$$

- **Representation formula for the exact solution:**

$$\mathbf{u}(t) = \exp(\mathbf{A}t)\mathbf{u}_0 + \int_0^t \exp(\mathbf{A}(t-s))\mathbf{g}(s)\ ds$$

Basic idea of exponential methods

- **Cauchy problem for nonhomogeneous linear ODE system:**

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{g}(t) \qquad \mathbf{u}(0) = \mathbf{u}_0$$

- **Representation formula for the exact solution:**

$$\mathbf{u}(t) = \exp\left(\mathbf{A}t\right)\mathbf{u}_0 + \int_0^t \exp\left(\mathbf{A}(t-s)\right)\mathbf{g}(s)\ ds$$

- **Exponential methods: turn this into a numerical method with errors indepentent of Δt for linear problems**

Basic idea of exponential methods

- Cauchy problem for nonhomogeneous linear ODE system:

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{g}(t) \qquad \mathbf{u}(0) = \mathbf{u}_0$$

- Representation formula for the exact solution:

$$\mathbf{u}(t) = \exp(\mathbf{A}t)\mathbf{u}_0 + \int_0^t \exp(\mathbf{A}(t-s))\mathbf{g}(s)\; ds$$

- Exponential methods: turn this into a numerical method with errors indepentent of Δt for linear problems
- Various extensions to nonlinear problems are available

Exponential Euler Rosenbrock methods

Exponential Euler Rosenbrock methods

- **Linearize around initial datum at each timestep**

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{u}^n) + \mathbf{J}^n(\mathbf{u} - \mathbf{u}^n) + \mathbf{R}(\mathbf{u}) \quad t \in [t^n, t^{n+1}]$$

Exponential Euler Rosenbrock methods

- **Linearize** around initial datum at each timestep

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{u}^n) + \mathbf{J}^n(\mathbf{u} - \mathbf{u}^n) + \mathbf{R}(\mathbf{u}) \quad t \in [t^n, t^{n+1}]$$

- **Freezing nonlinear terms yields**

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \phi\big(\mathbf{J}^n \Delta t\big)\mathbf{f}(\mathbf{u}^n) \qquad \phi(z) = \frac{\exp(z) - 1}{z}$$

Exponential Euler Rosenbrock methods

- Linearize around initial datum at each timestep

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{u}^n) + \mathbf{J}^n(\mathbf{u} - \mathbf{u}^n) + \mathbf{R}(\mathbf{u}) \quad t \in [t^n, t^{n+1}]$$

- Freezing nonlinear terms yields

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \phi\big(\mathbf{J}^n \Delta t\big)\mathbf{f}(\mathbf{u}^n) \qquad \phi(z) = \frac{\exp(z) - 1}{z}$$

- Essentially exact for linear, constant coefficient problems, unconditionally A-stable, second order for nonlinear problems, higher order variants available (Hochbruck et al 1997)

Exponential Euler Rosenbrock methods

- **Linearize** around initial datum at each timestep

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{u}^n) + \mathbf{J}^n(\mathbf{u} - \mathbf{u}^n) + \mathbf{R}(\mathbf{u}) \quad t \in [t^n, t^{n+1}]$$

- Freezing nonlinear terms yields

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \phi\big(\mathbf{J}^n \Delta t\big)\mathbf{f}(\mathbf{u}^n) \qquad \phi(z) = \frac{\exp(z) - 1}{z}$$

- Essentially **exact** for linear, constant coefficient problems, **unconditionally A-stable**, **second order** for nonlinear problems, higher order variants available (Hochbruck et al 1997)
- Stiff one step, one stage second order solver with **one** evaluation of RHS: think of the **physics...**

Main computational problems and solutions

Main computational problems and solutions

- **Exponential matrix cannot be stored for realistic PDE problems**

Main computational problems and solutions

- **Exponential matrix cannot be stored for realistic PDE problems**
- $\exp(\Delta t \mathbf{A})\mathbf{v}$ **can be approximated by the same Krylov space techniques employed in GMRES (Saad 1992)**

Main computational problems and solutions

- **Exponential matrix cannot be stored for realistic PDE problems**
- $\exp(\Delta t \mathbf{A})\mathbf{v}$ **can be approximated by the same Krylov space techniques employed in GMRES (Saad 1992)**
- **Krylov space dimension (and cost of time step) depend on the Courant number**

Main computational problems and solutions

- **Exponential matrix cannot be stored for realistic PDE problems**
- $\exp(\Delta t \mathbf{A})\mathbf{v}$ **can be approximated by the same Krylov space techniques employed in GMRES (Saad 1992)**
- **Krylov space dimension (and cost of time step) depend on the Courant number**
- **Alternative techniques imply similar costs for large scale problems**

Some numerical results

Some numerical results

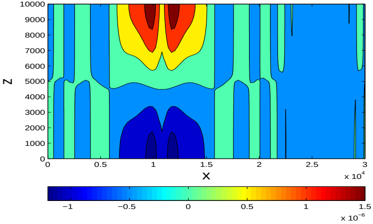
- **NUMA** model (courtesy of F.X.Giraldo, NPS), spatial discretization employing CG with fifth order polynomials

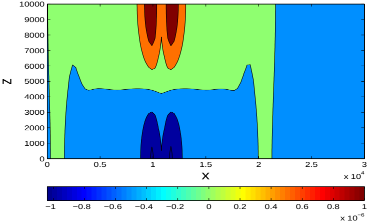
Some numerical results

- **NUMA model (courtesy of F.X.Giraldo, NPS), spatial discretization employing CG with fifth order polynomials**
- **Klemp-Skamarock test, Courant number approx.** 23, **density fields computed by second order exponential method and BDF2 at** $t = 400\ s$.

Some numerical results

- **NUMA model (courtesy of F.X.Giraldo, NPS), spatial discretization employing CG with fifth order polynomials**
- **Klemp-Skamarock test, Courant number approx.** 23, **density fields computed by second order exponential method and BDF2 at** $t = 400\ s$.





Some numerical results

Some numerical results

- **ICON shallow water model, low order mimetic spatial discretization**

Some numerical results

- **ICON shallow water model, low order mimetic spatial discretization**
- **Test case 5** $t = 360$ h, $\Delta x \approx 80$ km, $\Delta t = 1$ h, $C \approx 10$

Some numerical results

- **ICON shallow water model, low order mimetic spatial discretization**
- **Test case 5** $t = 360$ h, $\Delta x \approx 80$ km, $\Delta t = 1$ h, $C \approx 10$
- **Test case 6 at** $t = 240$ h, $\Delta x \approx 80$ km, $\Delta t = 0.5$ h $C \approx 10$

Some numerical results

- **ICON shallow water model, low order mimetic spatial discretization**
- **Test case 5** $t = 360\ \mathrm{h},\ \Delta x \approx 80\ \mathrm{km},\ \Delta t = 1\ \mathrm{h},\ C \approx 10$
- **Test case 6 at** $t = 240\ \mathrm{h},\ \Delta x \approx 80\ \mathrm{km},\ \Delta t = 0.5\ \mathrm{h}\ \ C \approx 10$
- **Reference solution computed by explicit Runge Kutta method of order 4 with** $\Delta t = 180\ \mathrm{s}$

Some numerical results

- **ICON shallow water model, low order mimetic spatial discretization**
- **Test case 5** $t = 360$ h, $\Delta x \approx 80$ km, $\Delta t = 1$ h, $C \approx 10$
- **Test case 6 at** $t = 240$ h, $\Delta x \approx 80$ km, $\Delta t = 0.5$ h $C \approx 10$
- **Reference solution computed by explicit Runge Kutta method of order 4 with** $\Delta t = 180$ s

	h error			
	LP	CN	EX2	EX3
Test 5	1.2e-2	9.1e-3	1.2e-3	1.1e-3
Test 6	5.9e-2	1.7e-2	3.8e-4	4.0e-4

A cost benefit analysis

A cost benefit analysis

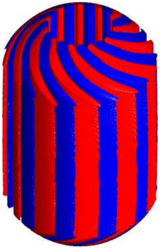
- Exponential vs high order IMEX methods

A cost benefit analysis

- **Exponential vs high order IMEX methods**
- **Spectral discretization of incompressible NS - Boussinesq in spherical geometry (Ferran, B., et al, JCP 2014)**

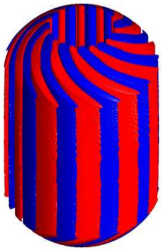
A cost benefit analysis

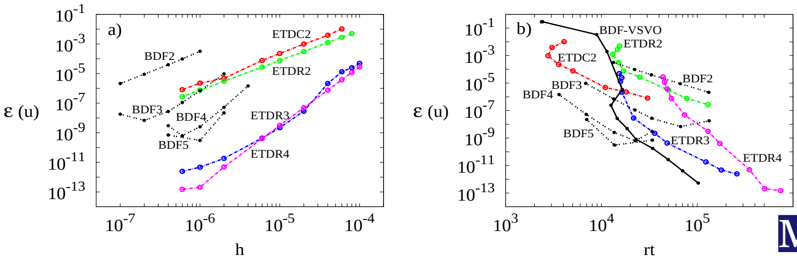
- Exponential vs high order IMEX methods
- Spectral discretization of incompressible NS - Boussinesq in spherical geometry (Ferran, B., et al, JCP 2014)



A cost benefit analysis

- Exponential vs high order IMEX methods
- Spectral discretization of incompressible NS - Boussinesq in spherical geometry (Ferran, B., et al, JCP 2014)





A more local approach

A more local approach

- **PDEs of interest are local in space: physical and numerical domain of dependence are finite**

A more local approach

- PDEs of interest are local in space: physical and numerical domain of dependence are finite
- Local problems discretized by FD, FV, FE methods yield sparse matrices

A more local approach

- **PDEs of interest are local in space: physical and numerical domain of dependence are finite**
- **Local problems discretized by FD, FV, FE methods yield sparse matrices**
- **Exponential of a sparse matrix is almost sparse (Iserles 2001)**

A more local approach

- **PDEs of interest are local in space: physical and numerical domain of dependence are finite**
- **Local problems discretized by FD, FV, FE methods yield sparse matrices**
- **Exponential of a sparse matrix is almost sparse (Iserles 2001)**
- **For $s-$banded $\mathbf{A} = (a_{i,j})$ with $|a_{i,j}| \leq \rho$, let $\exp(\mathbf{A}) = (e_{i,j})$.**

$$
\begin{aligned}
|e_{i,j}| &\leq \left(\frac{\rho s}{|i-j|}\right)^{\frac{|i-j|}{s}} \left[\mathrm{e}^{\frac{|i-j|}{s}} - \sum_{k=0}^{|i-j|-1} \frac{(|i-j/s|)^k}{k!}\right] \\
&\approx \left(\frac{\rho s}{|i-j|}\right)^{\frac{|i-j|}{s}} \frac{(|i-j|/s)^{|i-j|}}{|i-j|!}
\end{aligned}
$$

Application to PDE problems

Application to PDE problems

- **Advection diffusion problem: entries of matrix $\Delta t\mathbf{A}$ scale as**

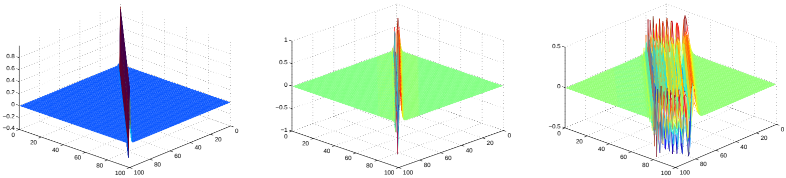
$$\frac{u\Delta t}{\Delta x} + \frac{\mu\Delta t}{\Delta x^2}$$

Application to PDE problems

- **Advection diffusion problem: entries of matrix $\Delta t\mathbf{A}$ scale as**

$$\frac{u\Delta t}{\Delta x} + \frac{\mu\Delta t}{\Delta x^2}$$

- **Example: $\exp(\Delta t\mathbf{A})$ for 1D centered finite difference advection at Courant numbers 0.5, 5, 20**

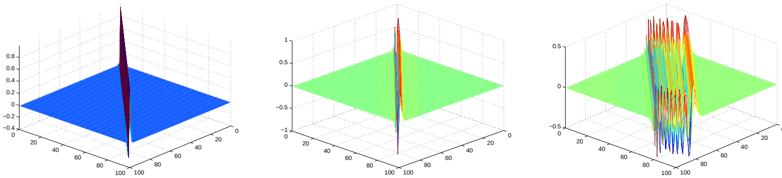


Application to PDE problems

- **Advection diffusion problem: entries of matrix $\Delta t \mathbf{A}$ scale as**

$$\frac{u\Delta t}{\Delta x} + \frac{\mu \Delta t}{\Delta x^2}$$

- **Example: $\exp(\Delta t \mathbf{A})$ for 1D centered finite difference advection at Courant numbers 0.5, 5, 20**



- **There is no real need to compute a global exponential matrix: Local Exponential Methods (LEM)**

LEM: a domain decomposition approach

LEM: a domain decomposition approach

- **Decompose mesh in overlapping regions**

$$\mathcal{M} = \bigcup_{i=1}^{N} \mathcal{M}_i \quad \mathcal{M}_i = \mathcal{D}_i \cup \mathcal{B}_i$$

where $\mathcal{D}_i$ non overlapping, $\mathcal{B}_i$ boundary buffer zones whose size depends on the Courant number

LEM: a domain decomposition approach

- Decompose mesh in overlapping regions

$$\mathcal{M} = \bigcup_{i=1}^{N} \mathcal{M}_i \quad \mathcal{M}_i = \mathcal{D}_i \cup \mathcal{B}_i$$

  where $\mathcal{D}_i$ non overlapping, $\mathcal{B}_i$ boundary buffer zones whose size depends on the Courant number
- For $i = 1, \dots, N$, solve local problem restricted to $\mathcal{M}_i$ by a local exponential method

$$\mathbf{u}_{\mathcal{M}_i}^{n+1} = \mathbf{u}_{\mathcal{M}_i}^{n} + \Delta t \phi\big(\mathbf{J}_{\mathcal{M}_i}^{n} \Delta t\big) \mathbf{f}(\mathbf{u}_{\mathcal{M}_i}^{n})_{\mathcal{M}_i}$$

LEM: a domain decomposition approach

- Decompose mesh in overlapping regions

$$\mathcal{M} = \bigcup_{i=1}^{N} \mathcal{M}_i \quad \mathcal{M}_i = \mathcal{D}_i \cup \mathcal{B}_i$$

  where $\mathcal{D}_i$ non overlapping, $\mathcal{B}_i$ boundary buffer zones whose size depends on the Courant number

- For $i = 1, \ldots, N$, solve local problem restricted to $\mathcal{M}_i$ by a local exponential method

$$\mathbf{u}^{n+1}_{\mathcal{M}_i} = \mathbf{u}^n_{\mathcal{M}_i} + \Delta t \phi\big(\mathbf{J}^n_{\mathcal{M}_i} \Delta t\big) \mathbf{f}(\mathbf{u}^n_{\mathcal{M}_i})_{\mathcal{M}_i}$$

- Overwrite degrees of freedom belonging to $\mathcal{B}_i$

LEM: cons and pros

LEM: cons and pros

- **Overhead increases with Courant number, both for computation and communication...**

LEM: cons and pros

- Overhead increases with Courant number, both for computation and communication...
- ...but should not too bad for high order methods, anisotropic meshes and heavy physics

LEM: cons and pros

- Overhead increases with Courant number, both for computation and communication...
- ...but should not too bad for high order methods, anisotropic meshes and heavy physics
- No global matrix to be computed, local problems can be parallelized trivially

LEM: cons and pros

- **Overhead increases** with Courant number, both for computation and communication...
- ...but should not too bad for **high order** methods, **anisotropic** meshes and heavy **physics**
- No **global** matrix to be computed, **local** problems can be **parallelized** trivially
- For small enough $\mathcal{D}_i$ local matrices **can be stored**: computational gain if Jacobian is frozen every few time steps and in the limit of **large number** of advected species

A 1D numerical example

A 1D numerical example

- **Viscous Burgers equation with periodic boundary conditions, exact solution via Cole-Hopf transformation**

A 1D numerical example

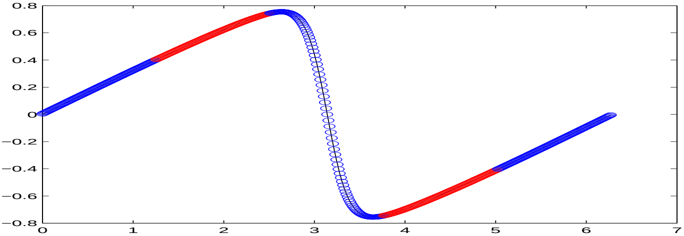
- **Viscous Burgers equation with periodic boundary conditions, exact solution via Cole-Hopf transformation**
- **Fourth order finite differences for advection, second order finite differences for diffusion, Courant number 15**

A 1D numerical example

- Viscous Burgers equation with periodic boundary conditions, exact solution via Cole-Hopf transformation
- Fourth order finite differences for advection, second order finite differences for diffusion, Courant number 15
- Second order exponential Rosenbrock method, stored local matrices computed without Krylov spaces

A 1D numerical example

- Viscous Burgers equation with periodic boundary conditions, exact solution via Cole-Hopf transformation
- Fourth order finite differences for advection, second order finite differences for diffusion, Courant number 15
- Second order exponential Rosenbrock method, stored local matrices computed without Krylov spaces



A 2D numerical example

A 2D numerical example

- **Advection-diffusion equation with rotational velocity field**

A 2D numerical example

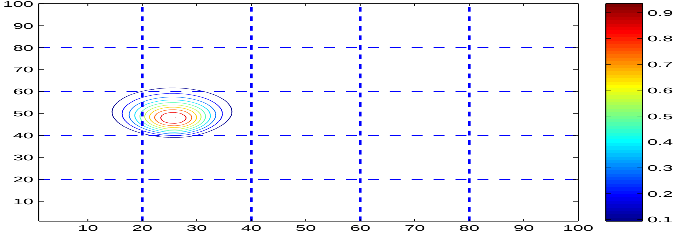
- **Advection-diffusion** equation with rotational velocity field
- Monotonic finite volume method for advection, second order finite volume method for diffusion, **Courant number 4**

A 2D numerical example

- Advection-diffusion equation with rotational velocity field
- Monotonic finite volume method for advection, second order finite volume method for diffusion, Courant number 4
- Second order exponential Rosenbrock method with local matrices computed by Krylov space techniques

A 2D numerical example

- **Advection-diffusion** equation with rotational velocity field
- Monotonic finite volume method for advection, second order finite volume method for diffusion, **Courant number 4**
- Second order exponential Rosenbrock method with **local matrices** computed by Krylov space techniques



A 2D nonlinear example

A 2D nonlinear example

- **Viscous Burgers equation**

A 2D nonlinear example

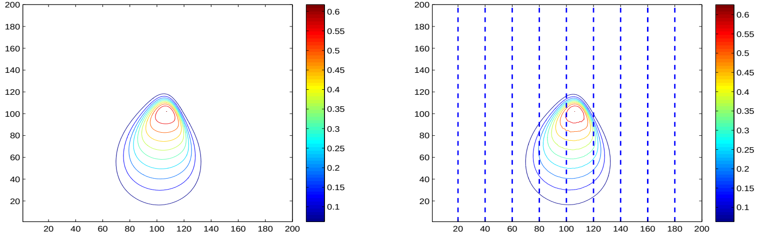
- **Viscous Burgers equation**
- **Centered finite volume method for advection, second order finite volume method for diffusion, anisotropic mesh with Courant number 6 in the vertical**

A 2D nonlinear example

- Viscous Burgers equation
- Centered finite volume method for advection, second order finite volume method for diffusion, anisotropic mesh with Courant number 6 in the vertical
- Second order exponential Rosenbrock method with local matrices computed by Krylov space techniques

A 2D nonlinear example

- Viscous **Burgers** equation
- Centered finite volume method for advection, second order finite volume method for diffusion, anisotropic mesh with **Courant number 6** in the vertical
- Second order exponential Rosenbrock method with **local matrices** computed by Krylov space techniques



Conclusions and perspectives

Conclusions and perspectives

- Straightforward implementation of exponential methods leads to very accurate but very costly solutions

Conclusions and perspectives

- **Straightforward implementation of exponential methods leads to very accurate but very costly solutions**
- **For standard PDE problems, a local approximation of** $\exp(\Delta t \mathbf{A})\mathbf{v}$ **is feasible**

Conclusions and perspectives

- **Straightforward implementation of exponential methods leads to very accurate but very costly solutions**
- **For standard PDE problems, a local approximation of $\exp(\Delta t \mathbf{A})\mathbf{v}$ is feasible**
- **Computation of exponential matrix becomes trivially parallel**

Conclusions and perspectives

- Straightforward implementation of exponential methods leads to very accurate but very costly solutions
- For standard PDE problems, a local approximation of $\exp(\Delta t \mathbf{A})\mathbf{v}$ is feasible
- Computation of exponential matrix becomes trivially parallel
- Computational overhead due to boundary buffer regions is limited in the case of anisotropic meshes and high order finite elements

Conclusions and perspectives

- Straightforward implementation of exponential methods leads to very accurate but very costly solutions
- For standard PDE problems, a local approximation of $\exp(\Delta t\mathbf{A})\mathbf{v}$ is feasible
- Computation of exponential matrix becomes trivially parallel
- Computational overhead due to boundary buffer regions is limited in the case of anisotropic meshes and high order finite elements
- Next on the to do list: use Local Exponential Methods in a high order FE framework and with complex forcing terms (multiple ARD with chemistry)