A domain decomposition approach to exponential methods for time discretization of PDEs

Luca Bonaventura

MOX - Politecnico di Milano

MAURIZIO60 5.12.2014
Outline of the talk
Outline of the talk

- Short review of exponential methods
Outline of the talk

- Short review of exponential methods
- An accuracy and efficiency assessment of the simplest approaches for their application to PDEs
Outline of the talk

- Short review of exponential methods
- An accuracy and efficiency assessment of the simplest approaches for their application to PDEs
- Local Exponential Methods: a domain decomposition approach to exponential methods for time discretization of PDEs
Outline of the talk

- Short review of **exponential methods**
- An **accuracy and efficiency** assessment of the simplest approaches for their application to PDEs
- Local Exponential Methods: a **domain decomposition** approach to exponential methods for time discretization of PDEs
- Some **preliminary** numerical results
Outline of the talk

- Short review of exponential methods
- An accuracy and efficiency assessment of the simplest approaches for their application to PDEs
- Local Exponential Methods: a domain decomposition approach to exponential methods for time discretization of PDEs
- Some preliminary numerical results
- Conclusions and perspectives for future work
Basic idea of exponential methods
Basic idea of exponential methods

- Cauchy problem for nonhomogeneous linear ODE system:

\[
\frac{du}{dt} = Au + g(t) \quad u(0) = u_0
\]
Basic idea of exponential methods

- Cauchy problem for nonhomogeneous linear ODE system:

\[ \frac{du}{dt} = Au + g(t) \quad u(0) = u_0 \]

- Representation formula for the exact solution:

\[ u(t) = \exp(A t) u_0 + \int_0^t \exp(A(t - s)) g(s) \, ds \]
Basic idea of exponential methods

- Cauchy problem for nonhomogeneous linear ODE system:

\[
\frac{du}{dt} = Au + g(t) \quad u(0) = u_0
\]

- Representation formula for the exact solution:

\[
u(t) = \exp(A t)u_0 + \int_0^t \exp(A(t - s))g(s) \, ds
\]

- Exponential methods: turn this into a numerical method with errors and stability independent of \( \Delta t \) for linear problems
Basic idea of exponential methods

- Cauchy problem for nonhomogeneous linear ODE system:

\[
\frac{du}{dt} = Au + g(t) \quad u(0) = u_0
\]

- Representation formula for the exact solution:

\[
u(t) = \exp(At)u_0 + \int_0^t \exp(A(t-s))g(s) \, ds
\]

- Exponential methods: turn this into a numerical method with errors and stability independent of \( \Delta t \) for linear problems

- Various extensions to nonlinear problems are available
A long story...
A long story...

A long story...


A long story...

A long story...

A long story...


Exponential Euler Rosenbrock methods
Exponential Euler Rosenbrock methods

- **Linearize numerically 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** numerically at each timestep

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

- **Freeze** nonlinear terms to obtain

\[
u^{n+1} = u^n + \Delta t \phi(J^n \Delta t) f(u^n) \quad \phi(z) = \frac{\exp(z) - 1}{z}
\]
Exponential Euler Rosenbrock methods

- **Linearize** numerically at each timestep

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

- **Freeze** nonlinear terms to obtain

\[
u^{n+1} = u^n + \Delta t \phi(J^n \Delta t)f(u^n) \quad \phi(z) = \frac{\exp(z) - 1}{z}
\]

- 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** numerically at each timestep

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

- **Freeze** nonlinear terms to obtain

\[
u^{n+1} = u^n + \Delta t \phi(J^n \Delta t) f(u^n) \quad \phi(z) = \frac{\exp(z) - 1}{z}
\]

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

- One step, one stage, second order **stiff** solver with one evaluation of RHS
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
- $\exp(\Delta tA)v$ 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
- $\exp(\Delta t A)v$ 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
- \( \exp(\Delta tA)v \) 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 for the computation of \( \exp(\Delta tA)v \) imply similar costs for large scale problems
Some numerical results
Some numerical results

- **NUMA model** (courtesy of F.X. Giraldo, NPS): Euler equations with gravity, spatial discretization employing **fifth order spectral elements**
Some numerical results

- **NUMA** model (courtesy of F.X. Giraldo, NPS): Euler equations with gravity, spatial discretization employing *fifth order spectral elements*

- 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): Euler equations with gravity, spatial discretization employing **fifth order spectral elements**

- **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 (L.B. and Ringler, MWR 2005): low order mimetic spatial discretization in spherical geometry
Some numerical results

- **ICON** shallow water model (L.B. and Ringler, MWR 2005): low order *mimetic* spatial discretization in spherical geometry
- **Test case 5** $t = 360 \text{ h}$, $\Delta x \approx 80 \text{ km}$, $\Delta t = 1 \text{ h}$, $C \approx 10$
Some numerical results

- **ICON shallow water model** (L.B. and Ringler, MWR 2005): low order **mimetic** spatial discretization in spherical geometry
- **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** (L.B. and Ringler, MWR 2005): low order mimetic spatial discretization in spherical geometry

- **Test case 5** \( t = 360 \text{ h}, \Delta x \approx 80 \text{ km}, \Delta t = 1 \text{ h}, C \approx 10 \)

- **Test case 6 at** \( t = 240 \text{ h}, \Delta x \approx 80 \text{ km}, \Delta t = 0.5 \text{ h} \quad C \approx 10 \)

- **Reference solution computed by explicit Runge Kutta method of order 4 with** \( \Delta t = 180 \text{ s} \)
Some numerical results

- **ICON** shallow water model (L.B. and Ringler, MWR 2005): low order **mimetic** spatial discretization in spherical geometry
- **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

<table>
<thead>
<tr>
<th></th>
<th>( h ) error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LP</td>
</tr>
<tr>
<td><strong>Test 5</strong></td>
<td>1.2e-2</td>
</tr>
<tr>
<td><strong>Test 6</strong></td>
<td>5.9e-2</td>
</tr>
</tbody>
</table>
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 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** $A = (a_{i,j})$ with $|a_{i,j}| \leq \rho$, let $\exp(A) = (e_{i,j})$.

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

\[
\approx \left( \frac{\rho s}{|i - j|} \right)^{\left| \frac{i-j}{s} \right|} \left( \frac{|i-j|/s}{|i-j|} \right)^{|i-j|}
\]
Application to PDE problems
Application to PDE problems

- Advection diffusion problem: entries of matrix $\Delta t A$ 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 $\Delta t A$ scale as

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

- Example: $\exp(\Delta t A)$ for 1D centered finite difference advection at Courant numbers 0.5, 5, 20
Application to PDE problems

- Advection diffusion problem: entries of matrix $\Delta tA$ scale as
  \[
  \frac{u\Delta t}{\Delta x} + \frac{\mu\Delta t}{\Delta x^2}
  \]

- Example: $\exp(\Delta tA)$ 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

\[ M = \bigcup_{i=1}^{N} M_i \quad M_i = D_i \cup B_i \]

where \( D_i \) non overlapping, \( B_i \) 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 \( i = 1, \ldots, N \), solve **local problem** restricted to \( \mathcal{M}_i \) by a local exponential method

  \[ u_{\mathcal{M}_i}^{n+1} = u_{\mathcal{M}_i}^n + \Delta t \phi(J_{\mathcal{M}_i}^n \Delta t) f(u_{\mathcal{M}_i}^n) \mathcal{M}_i \]
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

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

- 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 and for strongly anisotropic meshes
Overhead increases with Courant number, both for computation and communication...

...but should not too bad for high order methods and for strongly anisotropic meshes

No global matrix computation: 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 and for strongly **anisotropic** meshes
- **No global** matrix computation: **local** problems can be **parallelized trivially**
- For small enough $\mathcal{D}_i$ local matrices can be stored...
LEM: cons and pros

- **Overhead increases** with Courant number, both for computation and communication...
- ...but should not too bad for **high order** methods and for strongly **anisotropic** meshes
- **No global** matrix computation: **local** problems can be **parallelized** trivially
- **For small enough** $D_i$ local matrices **can be stored**...
- ... implying a **major cost reduction** if Jacobian is only recomputed every few time steps
A 1D numerical example
A 1D numerical example

- Viscous Burgers equation with Gaussian initial datum
A 1D numerical example

- Viscous Burgers equation with Gaussian initial datum
- Finite volume discretization with flux limiter monotonization, Courant number 4
A 1D numerical example

- Viscous Burgers equation with Gaussian initial datum
- Finite volume discretization with flux limiter monotonization, Courant number 4
- Third order exponential Rosenbrock method, stored local matrices computed without Krylov spaces
A 1D numerical example

- Viscous Burgers equation with Gaussian initial datum
- Finite volume discretization with flux limiter monotonization, Courant number 4
- Third 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, \textit{Courant number 4}
- Second order exponential Rosenbrock method with \textit{local} matrices computed by Krylov space techniques
Computational cost reduction
Computational cost reduction

- Burgers equation, CPU times (in seconds) for scalar LEM runs
Computational cost reduction

- Burgers equation, CPU times (in seconds) for scalar LEM runs
- Results as a function of the time step, the number $D$ of subdomains employed and the number of grid points $B$ in the buffer regions.
## Computational cost reduction

- **Burgers equation**, CPU times (in seconds) for scalar LEM runs
- Results as a function of the time step, the number $D$ of subdomains employed and the number of grid points $B$ in the buffer regions.

<table>
<thead>
<tr>
<th></th>
<th>$D = 1$</th>
<th>$D = 2$</th>
<th>$D = 4$</th>
<th>$D = 5$</th>
<th>$D = 8$</th>
<th>$D = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = 0.4, B = 5$</td>
<td>25.89</td>
<td>13.46</td>
<td>10.63</td>
<td>10.30</td>
<td>10.09</td>
<td>10.42</td>
</tr>
<tr>
<td>$C = 1, B = 15$</td>
<td>30.07</td>
<td>9.02</td>
<td>4.70</td>
<td>4.72</td>
<td>4.78</td>
<td>5.49</td>
</tr>
<tr>
<td>$C = 2, B = 20$</td>
<td>26.77</td>
<td>6.57</td>
<td>2.87</td>
<td>3.24</td>
<td>3.05</td>
<td>3.13</td>
</tr>
</tbody>
</table>
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
- A more **local** approximation of $\exp(\Delta tA)v$ is feasible: LEM
Conclusions and perspectives

- Straightforward implementation of exponential methods leads to very accurate but very costly solutions
- A more local approximation of $\exp(\Delta tA)v$ is feasible: LEM
- Computation of exponential matrix becomes trivially parallel, although an overhead is introduced due to overlapping buffers
Conclusions and perspectives

- Straightforward implementation of exponential methods leads to very accurate but very costly solutions
- A more local approximation of $\exp(\Delta t A)v$ is feasible: LEM
- Computation of exponential matrix becomes trivially parallel, although an overhead is introduced due to overlapping buffers
- Preliminary numerical results show significant reduction of the computational cost
Conclusions and perspectives

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

- A more local approximation of $\exp(\Delta t A)v$ is feasible: LEM

- Computation of exponential matrix becomes trivially parallel, although an overhead is introduced due to overlapping buffers

- Preliminary numerical results show significant reduction of the computational cost

- Ongoing work: application of LEM to high order DG discretizations of various highly oscillating problems: atmospheric dynamics, Schrödinger equation