Localized Exponential Time Differencing Methods Based on Domain Decomposition

Zhu Wang
Department of Mathematics
University of South Carolina

Collaborators:
T.-T.-P. Hoang (USC/Auburn), L. Ju and X. Meng (USC)

E3SM Concall Meeting
January 17, 2019

Work overview

Idea

Applying parallel Schwarz algorithms with overlapping domain decomposition to time evolution problems discretized in time by the exponential time differencing methods.

Advantages

- Using exponential integrator allows large time step sizes.
- Solving subdomain problems of smaller sizes in parallel, possibly with different time steps in different subdomains.
- Reducing computational cost without affecting the accuracy of the approximate solution.

Global numerical solution

- PDE models: parabolic or hyperbolic types such as shallow water equations

  Spatial discretization: $\boldsymbol{u}'(t) = \boldsymbol{L}\boldsymbol{u}(t) + \boldsymbol{R}(t, \boldsymbol{u}(t), \psi(t)), \quad 0 < t < T, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0.$

  Time integration: exponential time differencing

  - Given solution $\boldsymbol{u}_m$ at t_m and a time step $\Delta t = t_{m+1} - t_m$.

$$\boldsymbol{u}_{m+1} = \mathrm{e}^{\Delta t\,\boldsymbol{L}}\boldsymbol{u}_m + \int_0^{\Delta t} \mathrm{e}^{(\Delta t - s)\,\boldsymbol{L}} \left[\frac{\boldsymbol{R}(t_{m+1}) - \boldsymbol{R}(t_m)}{\Delta t} s + \boldsymbol{R}(t_m) \right] ds \qquad (1)$$

$$= \mathrm{e}^{\Delta t\,\boldsymbol{L}}\boldsymbol{u}_m + \Delta t \varphi_1(\Delta t\,\boldsymbol{L})\,\boldsymbol{R}(t_m) + \Delta t \varphi_2(\Delta t\,\boldsymbol{L})\,[\boldsymbol{R}(t_{m+1}) - \boldsymbol{R}(t_m)] \qquad (2).$$

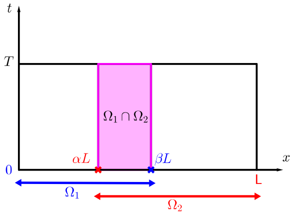
  - Denote $\boldsymbol{R}(t) \equiv \boldsymbol{R}(t, \boldsymbol{u}(t), \psi(t))$; define $\varphi_1(z) = \frac{e^z - 1}{z}$ and $\varphi_2(z) = \frac{\varphi_1(z) - 1}{z}$.

  - Second-order accuracy in time, named **ETD2**.

    It can be formulated as a two stage approach (see Konstantin's slides).

- High performance computing ⇒ localized ETD based on domain decomposition

Multidomain formulation



- Partition Ω into overlapping subdomains Ω_1 and Ω_2.

  Partition $\boldsymbol{u}$ into overlapping subsets $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$.

  Solve subdomain problems separately.

  Transmission conditions on the interfaces:

  $$\boldsymbol{u}_1(N_\beta, t) = \boldsymbol{u}_2(N_{\beta,\alpha}, t) \;\text{ and }\; \boldsymbol{u}_2(1, t) = \boldsymbol{u}_1(N_\alpha, t),$$

  where $N_\alpha h = \alpha L$, $N_\beta h = \beta L$, $N_{\beta,\alpha} = N_\beta - N_\alpha + 1$.

- Recall a two-stage ETD2:

$$\widetilde{\boldsymbol{u}}_{m+1} = e^{\Delta t \boldsymbol{L}} \boldsymbol{u}_m + \Delta t \varphi_1(\Delta t \boldsymbol{L}) \boldsymbol{R}(t_m, \boldsymbol{u}_m, \psi_1, \psi_2);$$

$$\boldsymbol{u}_{m+1} = \widetilde{\boldsymbol{u}}_{m+1} + \Delta t \varphi_2(\Delta t \boldsymbol{L}) \left[\boldsymbol{R}(t_{m+1}, \widetilde{\boldsymbol{u}}_{m+1}, \psi_1, \psi_2) - \boldsymbol{R}(t_m, \boldsymbol{u}_m, \psi_1, \psi_2)\right].$$

- Assume that subdomain solutions at time t_m, $\boldsymbol{u}_{1,m}$ and $\boldsymbol{u}_{2,m}$, are obtained.

Second-order localized ETD (LETD) algorithm

- First compute subdomain solutions $\widetilde{\boldsymbol{u}}_{1,m+1}$ and $\widetilde{\boldsymbol{u}}_{2,m+1}$.

  For instance, in Ω_1,

$$\widetilde{\boldsymbol{u}}_{1,m+1} = e^{\Delta t \boldsymbol{L}_1} \boldsymbol{u}_{1,m} + \Delta t \varphi_1(\Delta t \boldsymbol{L}_1) \boldsymbol{R}_1 \left(t_m, \boldsymbol{u}_{1,m}, \psi_1(t_m), \boldsymbol{u}_{2,m}(N_{\beta,\alpha})\right).$$

- Set $\boldsymbol{u}^{(0)}_{1,m+1}(N_\alpha) = \widetilde{\boldsymbol{u}}_{1,m+1}(N_\alpha)$ and $\boldsymbol{u}^{(0)}_{2,m+1}(N_{\beta,\alpha}) = \widetilde{\boldsymbol{u}}_{2,m+1}(N_{\beta,\alpha})$.

- Start the iteration: for $k = 0, 1, \cdots$, compute $\boldsymbol{u}^{(k+1)}_{1,m+1}$ and $\boldsymbol{u}^{(k+1)}_{2,m+1}$.

  For instance, in Ω_1,

$$\begin{aligned}\boldsymbol{u}^{(k+1)}_{1,m+1} = \widetilde{\boldsymbol{u}}_{1,m+1} + \Delta t \varphi_2(\Delta t \boldsymbol{L}_1) \cdot \Big[& \boldsymbol{R}_1 \left(t_{m+1}, \widetilde{\boldsymbol{u}}_{1,m+1}, \psi_1(t_{m+1}), \boldsymbol{u}^{(k)}_{2,m+1}(N_{\beta,\alpha})\right) \\ & - \boldsymbol{R}_1 \left(t_m, \boldsymbol{u}_{1,m}, \psi_1(t_m), \boldsymbol{u}_{2,m}(N_{\beta,\alpha})\right) \Big].\end{aligned}$$

- Stop if interface values from subdomain solutions are close enough.

Model problem

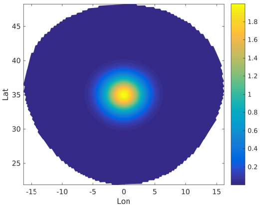
Rotating Shallow water equation (SWE)

$$\begin{cases} \partial_t h + \nabla \cdot (h\boldsymbol{u}) = 0, \text{ in } \Omega \times (0,T), \\ \partial_t \boldsymbol{u} + (f+\omega)\boldsymbol{k} \times \boldsymbol{u} + \nabla \left(\frac{|\boldsymbol{u}|^2}{2} + g(h+b) \right) = \boldsymbol{0}, \text{ in } \Omega \times (0,T), \end{cases}$$

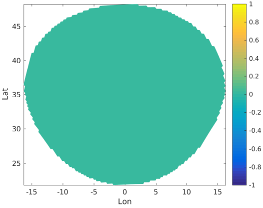
- h the fluid thickness, $\boldsymbol{u}$ the velocity field, $\omega = \boldsymbol{k} \cdot (\nabla \times \boldsymbol{u})$ the relative vorticity, $\boldsymbol{k}$ is the surface normal vector, g the acceleration of gravity, b the bottom topography and f the Coriolis parameter.
- Application of TRiSK scheme leads: $\boldsymbol{U}' = \boldsymbol{F}(\boldsymbol{U}, \psi)$ (see Lili's slides).
  - Approach I: $\boldsymbol{F} = \boldsymbol{J}_m \boldsymbol{U} + \boldsymbol{R}_m$,
    where $\boldsymbol{J}_m$ the Jacobian of $\boldsymbol{F}$ at $\boldsymbol{U}(t_m)$ and $\boldsymbol{R}_m = \boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{J}_m \boldsymbol{U}$ the remainder.
  - Approach II: $\boldsymbol{F} = \boldsymbol{A}_{ref} \boldsymbol{U} + \boldsymbol{R}_{ref}$, using Hamiltonian view (see Konstantin's slides).
- Application of LETD2 (Approach I $\rightarrow$ **LETD2**; Approach II $\rightarrow$ **LETD2-wave**).

Gaussian pulse test case

- SOMA test case inspired geometry (Ocean basin) with no forcing or smoothing.
- Primal SCVT mesh consists of 8521 cells, 25898 edges, and 17378 vertices.
- Gaussian initial condition:



Sea surface height



Velocity field

- No normal flow boundary condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$

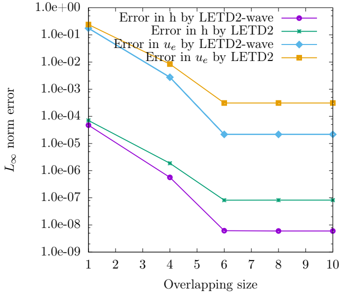
Performance of LETD2

- 10 subdomains with nearly equal parts generated by METIS.
  Overlapping 6 cells, and $\Delta t = 200$ s.
- Relative L_∞ error in h, using RK4 with $\Delta t = 1$ s as benchmark.
  Average CPU time per step (CPU time per processor is shown for localized algorithms).

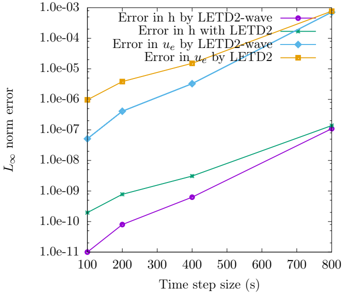
Methods	# Krylov vectors=20		# Krylov vectors=30	
	error	time	error	time
ETD2	8.2e-8	$2.39e-01$ s	8.2e-8 [h]	$3.15e-01$ s
LETD2	8.2e-8	$5.12e-02$ s	8.2e-8 [h]	$7.01e-02$ s
ETD2-wave	6.0e-9	$5.84e-02$ s	6.0e-9 [h]	$8.00e-02$ s
LETD2-wave	6.0e-9	$1.73e-02$ s	6.0e-9 [h]	$2.10e-02$ s

- No iteration needed for LETD2 or LETD2-wave.
  Localized schemes achieve the same accuracy as the associated global schemes, while accelerating the simulations; **ETD-wave** models are computationally more efficient than ETD models.

- 10 subdomains, 30 Krylov vectors.
- Relative L_∞ error in h and $\boldsymbol{u}$, using RK4 with $\Delta t = 1$ s as benchmark.



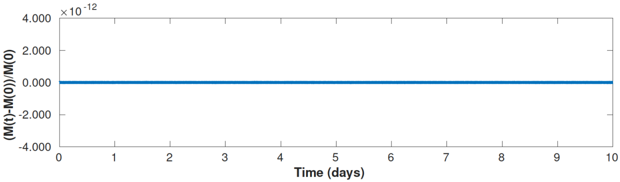
Error vs overlapping size when $\Delta t = 200$ s.



Error vs time step when overlapping 8 cells.

- 10-day simulation using LETD2-wave

  10 subdomains, 30 Krylov vectors, and $\Delta t = 200$ s when overlapping 8 cells.



- Mass conservation up to machine precision.

Conclusion

Summary

- Localized ETD algorithms with overlapping subdomains.

  Reach the same accuracy as global schemes.

  Speed up simulations through parallel performance.

Next steps

- Convergence analysis for Localized ETD applied to SWEs.

  Extensions to multi-layer SWEs, and more complicated systems.