Parallel Exponential Time Differencing Methods for Ocean Dynamics

Lili Ju, Zhu Wang, Rihui Lan, *University of South Carolina*
Wei Leng, *Chinese Academy of Sciences*
Max Gunzburger, *Florida State University*

2020 ESMD-E3SM PI Meeting
Oct. 26 – Oct. 29, 2020

Outline

1. Parallel ETD schemes for rotating SWEs
   - Rotating SWEs and TRiSK scheme
   - Exponential time differencing Runge-Kutta method
   - Parallel global ETD method
2. Parallel ETD schemes for primitive equations
   - Primitive equations
   - Barotropic-baroclinic splitting
   - Parallel ETD for the barotropic solve
3. Ongoing work

Project information

DOE Award

- "Efficient and Scalable Time-Stepping Algorithms and Reduced-Order Modeling for Ocean System Simulations", Department of Energy Office of Science, 09/01/2019-08/31/2022.

Members

- FSU: Max Gunzburger (Project PI and Institution Lead PI)
- UofSC: Lili Ju (Institutional Lead PI), Zhu Wang, Rihui Lan
- LANL: Philip Jones (Institution Lead PI), Sara Calandrini

Rotating shallow water equations - Single-layer case

Single-layer rotating SWEs in vector-invariant form

(1) $$\frac{\partial h}{\partial t} + \nabla \cdot (h\boldsymbol{u}) = 0,$$

(2) $$\frac{\partial \boldsymbol{u}}{\partial t} + q(h\boldsymbol{u}^{\perp}) + g\nabla(h+b) + \nabla K = \boldsymbol{G}(h, \boldsymbol{u}),$$

- h: the fluid thickness, $\boldsymbol{u}$: the fluid velocity,
- $\boldsymbol{k}$: the unit vector pointing in the local vertical direction,
- $\boldsymbol{u}^{\perp} = \boldsymbol{k} \times \boldsymbol{u}$: the velocity rotated through a right angle,
- $\eta = \boldsymbol{k} \cdot \nabla \times \boldsymbol{u} + f$: the absolute vorticity and $q = \dfrac{\eta}{h}$: the fluid potential vorticity,
- $K = |\boldsymbol{u}|^2/2$: the kinetic energy,
- g: gravity, f: Coriolis parameter and b: bottom topography,
- $\boldsymbol{G}$: additional stress or diffusion terms.

Rotating shallow water equations – Multi-layer case

Assume that there are totally L layers of fluids.

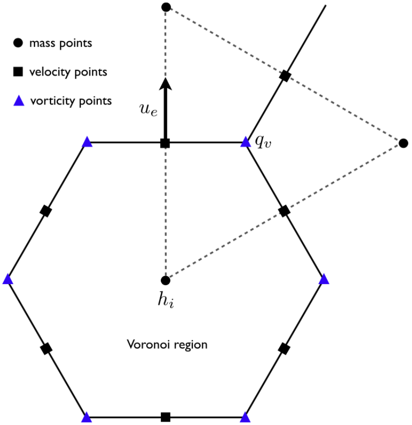
Multi-layer rotating SWEs for the l-th layer

$$\frac{\partial h_l}{\partial t} + \nabla \cdot (h_l \boldsymbol{u}_l) = 0, \tag{3}$$

$$\frac{\partial \boldsymbol{u}_l}{\partial t} + q(h_l, \boldsymbol{u}_l)(h_l \boldsymbol{u}_l^{\perp}) + \nabla(K_l + g p_l(\boldsymbol{h})/\rho_l) = \boldsymbol{G}_l(\boldsymbol{h}, \boldsymbol{u}), \tag{4}$$

- The subscript l specifies the related layer with $1 \le l \le L$,
- ρ_l: the fluid density of layer l, and $\rho_l < \rho_{l+1}$, for $l = 1, \dots, L-1$,
- $\boldsymbol{h} = (h_1, h_2, \dots, h_L)^T$, $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_L)^T$,
- Coupling through $\xi_l(\boldsymbol{h}) = b + \sum_{k=l}^{L} h_k$: the layer coordinates and $p_l(\boldsymbol{h}) = \rho_l \xi_l(\boldsymbol{h}) + \sum_{k=1}^{l-1} \rho_k h_k$: the dynamical pressure,
- $\boldsymbol{G}_l$: additional stress or diffusion terms, e.g., wind stress or bottom friction.

TRiSK scheme in MPAS-Ocean



- C-grid staggering in space
- **Primal mesh**: a Voronoi tessellation
- **Dual mesh**: its associated Delaunay triangulation
- Duality and orthogonality
- h_i: the mean thickness over primal cell P_i
- u_e: the component of the velocity vector in the direction normal to primal edges
- q_v: the mean vorticity over dual cell D_v
- **Finite volume discretization**

[Thuburn, Ringler, Skamarock and Klemp, *JCP*, 2009; Ringler, Thuburn, Klemp and Skamarock, *JCP*, 2010]

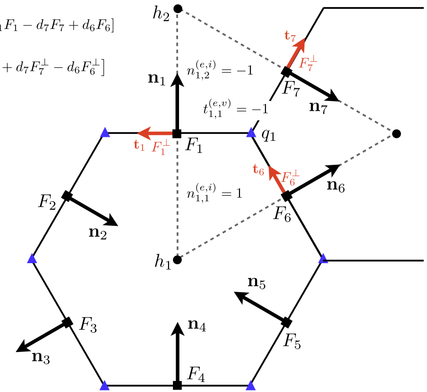
Discrete div, grad and curl operators

$$[\nabla \cdot F\mathbf{n}]_1^{(i)} = \frac{1}{A^{(i)}}[l_1F_1 - l_2F_2 + l_3F_3 - l_4F_4 - l_5F_5 + l_6F_6]$$

$$[\mathbf{k} \cdot \nabla\times\ F\mathbf{n}]_1^{(v)} = \frac{1}{A_1^{(v)}}[-d_1F_1 - d_7F_7 + d_6F_6]$$

$$[\nabla \cdot F^{\perp}\mathbf{t}]_1^{(v)} = \frac{1}{A_1^{(v)}}[d_1F_1^{\perp} + d_7F_7^{\perp} - d_6F_6^{\perp}]$$

$$[\nabla h]_1^{(e)} = \frac{1}{d_1}[h_2 - h_1]$$



Divergence Th.: $\iint_S \nabla \cdot \mathbf{F}\, dS = \oint_\Gamma \mathbf{F} \cdot \mathbf{n}\, d\Gamma$ **Stokes' Th.:** $\iint_S \nabla \times \mathbf{F} \cdot d\mathbf{S} = \oint_\Gamma \mathbf{F} \cdot d\Gamma$

Exponential time differencing

- System of ODEs resulting from spatial discretization:

$$\partial_t \boldsymbol{W} = \boldsymbol{F}(\boldsymbol{W}). \tag{5}$$

- Exponential time differencing (ETD) at each time step interval $[t_n, t_{n+1}]$:
  - Split the forcing term into a linear part and a remainder part:

$$\partial_t \boldsymbol{W} = \boldsymbol{J}_n \boldsymbol{W}(t) + \boldsymbol{R}_n\left(\boldsymbol{W}(t)\right), \tag{6}$$

    where $\boldsymbol{J}_n = \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{W}}(\boldsymbol{W}_n)$ is the Jacobian matrix evaluated at $\boldsymbol{W}_n$ and $\boldsymbol{R}_n = \boldsymbol{F}(\boldsymbol{W}) - \boldsymbol{J}_n\boldsymbol{W}$ is the remainder.
  - Use the variation of constants formula:

$$\boldsymbol{W}_{n+1} = e^{\Delta t \boldsymbol{J}_n}\boldsymbol{W}_n + e^{\Delta t \boldsymbol{J}_n}\int_0^{\Delta t} e^{(\Delta t - \tau)\boldsymbol{J}_n}\boldsymbol{R}_n(\boldsymbol{W}(t_n+\tau))\, d\tau, \tag{7}$$

    where the time step size $\Delta t = t_{n+1} - t_n$.

ETD-RK schemes

- The exponential Rosenbrock-Euler (ETD-Rosenbrock):

$$\boldsymbol{w}_{n,1} = \boldsymbol{W}_n + \Delta t \varphi_1(\Delta t \boldsymbol{J}_n)\boldsymbol{F}(\boldsymbol{W}_n)$$

- A two-stage second-order exponential Heun method:

$$\begin{cases} \boldsymbol{w}_{n,1} = \boldsymbol{W}_n, \\ \boldsymbol{w}_{n,2} = \boldsymbol{W}_n + \Delta t \varphi_1(\Delta t \boldsymbol{J}_n)\boldsymbol{F}(\boldsymbol{W}_n), \\ \boldsymbol{W}_{n+1} = \boldsymbol{W}_n + \Delta t \varphi_1(\Delta t \boldsymbol{J}_n)\boldsymbol{F}(\boldsymbol{W}_n) + \Delta t \varphi_2(\Delta t \boldsymbol{J}_n)\Big(\boldsymbol{R}_n(\boldsymbol{w}_{n,2}) - \boldsymbol{R}_n(\boldsymbol{W}_n)\Big) \end{cases}$$

  where the φ-functions are $\varphi_1(z) = \frac{e^z-1}{z}$ and $\varphi_2(z) = \frac{e^z-1-z}{z^2}$.

- Allow for stable large time stepping with better accuracy than classic implicit schemes.
- Use **Krylov subspace method** to compute the products of matrix exponential and vector.
- The adaptive Krylov subspace method + Incomplete orthogonalization method (IOM), e.g., phipm/IOM2 [*Gaudreault and Pudykiewicz, JCP, 2016*].

Algebraic parallelization of ETDs

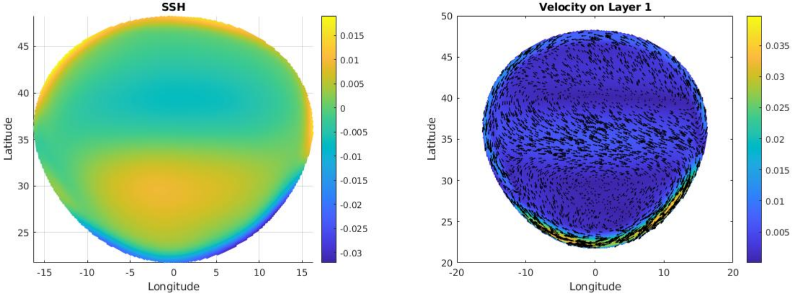
- The standard data-parallel is taken: each vector is split across all the processors/cores with corresponding subdomains, and the MPI environment is used for communications and performing the matrix exponential-vector product operations.
- Three types of ETD methods with the same second order accuracy are considered, for time stepping in the rotating shallow water equations: the ETD2-wave, the B-ETD2wave [*Pieper, Sockwell and Gunzburger, JCP, 2020*], and the ETD-Rosenbrock.
  - The first two (use skew-Lanczos iteration in Krylov subspace method) rely on a Hamiltonian form of the equations and the assumption of zero reference state of the SWEs during simulations;
  - The third (use Arnoldi iteration in Krylov subspace method) is numerically much more stable for general cases without these assumptions.
- We use the "**Trilinos**" Epetra package as the base for our parallel implementation within the MPAS framework.

A technique for ETD-Rosenbrock

- In ETD-Rosenbrock, we split the Jacobian matrix into two parts for the multi-layer SWE model.
- The first part is layer-independent without the pressure term, so that we can compute the sub-Jacobian matrix on each layer independently; during the Arnoldi-process, we combine the sub-Jacobian matrix with its own layer data.
- The remaining Jacobian matrix only has the pressure part, which gathers all layers' thickness.
- In order to reduce its communication among all the calling processors/cores during the Arnoldi process, we first take the gradient operation locally on each layer, then combine all the resulting vectors.
- Update the Jacobian every 20 time steps.

The three-layer SOMA test: zero initial velocity

- 3 SCVT meshes with different resolutions
  - 16 km: 22,007 cells, 66,560 edges and 44,554 vertices;
  - 8 km: 88,056 cells, 265,245 edges and 177,190 vertices;
  - 4 km: 352,256 cells, 1,058,922 edges and 706,667 vertices.
- To measure the parallel efficiency, we define $E_p = \frac{r \cdot T_r}{p \cdot T_p}$, where T_r is the CPU time when the referential r processors are used, T_p is the running time for p processors.



Sea surface height (left) and velocity on the top layer (right) at T=1 day.

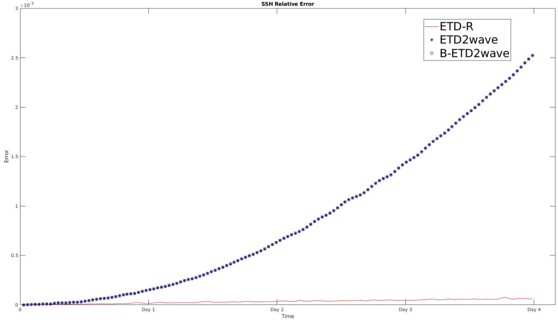
Parallel performance

- Run a 1-day-long simulation on the NERSC Cori system: the time step size $\Delta t = 107s$ and the maximum number of Krylov vectors $M = 45$.

Cores	ETD-Rosenbrock		ETD2wave		B-ETD2wave	
	Time	Efficiency	Time	Efficiency	Time	Efficiency
16 km						
8	194.64	-	76.64	-	27.99	-
16	100.14	92%	43.11	89%	17.87	78%
32	84.82	58%	27.22	70%	12.56	56%
64	35.71	68%	16.10	70%	8.17	43%
8 km						
8	855.73	-	470.61	-	140.25	-
16	482.32	89%	281.73	84%	78.26	90%
32	343.71	62%	216.39	54%	51.67	68%
64	173.84	62%	93.15	63%	26.47	66%
128	92.2	58%	33.67	87%	14.5	60%
4 km						
16	2316.33	-	1414.57	-	528.83	-
32	1657.91	70%	1069.01	66%	364.62	73%
64	737.64	79%	478.49	74%	157.13	84%
128	369.09	78%	221.02	80%	53.99	122%

The three-layer SOMA test: nonzero initial velocity

- ETD-Rosenberg provides the smallest approximation error, while the other two methods have approximation errors growing with time.
- Both ETD2wave and B-ETD2wave assume a zero reference velocity during the simulation in order to simplify the Jacobian matrix in the Hamiltonian fashion and utilize its skew-symmetry, which does not hold in this test case.



Time evolution of relative errors in fluid height in 4 days

Primitive equations

The equations for momentum, thickness, tracer, and state.

Primitive equations for z-level ocean motion

$$(8)\quad \begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + \eta \boldsymbol{k} \times \boldsymbol{u} + \omega \dfrac{\partial \boldsymbol{u}}{\partial z} = -\dfrac{1}{\rho_0}\nabla p - \dfrac{\rho g}{\rho_0}\nabla z^{\text{mid}} - \nabla K + \boldsymbol{D}_h^u + \boldsymbol{D}_v^u + \mathcal{F}^u, \\ \dfrac{\partial h}{\partial t} + \nabla \cdot (h\overline{\boldsymbol{u}}^z) = 0, \\ \dfrac{\partial}{\partial t} h\overline{\varphi}^z + \nabla \cdot (h\overline{\varphi \boldsymbol{u}}^z) = \boldsymbol{D}_h^\varphi + \boldsymbol{D}_v^\varphi + \mathcal{F}^\varphi, \\ \rho = f_{\text{eos}}(\Theta, S, p), \end{cases}$$

Due to the well-posedness, (8) also needs the hydrostatic condition:

$$(9)\quad p(x,y,z) = p^s(x,y) + \int_z^{z^s} \rho g dz'.$$

Primitive equations

Variables definitions

- $\boldsymbol{u}$: horizontal velocity; h: layer thickness; Θ: potential temperature; S: salinity;
- φ: generic tracer, it can be Θ or S;
- p: pressure; p^s: surface pressure;
- z^{mid}: z-location of middle of layer; z^s: z-location of sea surface;
- $\boldsymbol{D}_h^u$, $\boldsymbol{D}_v^u$: momentum diffusion terms for horizontal and vertical directions;
- $\boldsymbol{D}_h^\varphi$, $\boldsymbol{D}_v^\varphi$: tracer diffusion terms for horizontal and vertical directions;
- Operator $\overline{(\cdot)}^z$: vertical average over the layer;
- ω: relative vorticity, $\omega = \boldsymbol{k} \cdot (\nabla \times \boldsymbol{u})$;
- η: absolution vorticity, $\eta = \omega + f$, where f is the Coriolis parameter;
- $\mathcal{F}^u, \mathcal{F}^\varphi$: momentum/tracer forcing.

TRiSK scheme

Layered system

Decompose the space vertically into L layers, and on Layer l:

$$\frac{\partial h_l}{\partial t} + \nabla\cdot(\widehat{h}_{l,e}u_l) + \frac{\partial}{\partial z}(h_l\omega_l) = 0, \tag{10}$$

$$\frac{\partial \boldsymbol{u}_l}{\partial t} + \frac{1}{2}\nabla|\boldsymbol{u}_l|^2 + (\boldsymbol{k}\cdot\nabla\times\boldsymbol{u}_l)\boldsymbol{u}_l^\perp + f\boldsymbol{u}_l^\perp + \omega_{l,e}\frac{\partial \boldsymbol{u}_l}{\partial z} = -\frac{1}{\rho_0}\nabla p_l + \nu_h\nabla^2\boldsymbol{u}_l + \frac{\partial}{\partial z}(\nu_v\frac{\partial \boldsymbol{u}_l}{\partial z}), \tag{11}$$

$$\frac{\partial h_l\varphi_l}{\partial t} + \nabla\cdot(\widehat{h}_{l,e}\varphi_{l,e}\boldsymbol{u}_l) + \frac{\partial}{\partial z}(h_l\varphi_l\omega_l) = \nabla\cdot(\widehat{h}_{l,e}\kappa_h\nabla\varphi_l) + h_l\frac{\partial}{\partial z}(\kappa_v\frac{\partial\varphi_l}{\partial z}). \tag{12}$$

Barotropic mode

- Barotropic mode is to model the rapid external gravity waves;
- Barotropic velocity $\overline{\boldsymbol{u}}$ is defined as the mass-weighted vertical average:

$$\overline{\boldsymbol{u}} = \sum_{k=1}^{L} \widehat{h}_{k,e} \boldsymbol{u}_k \Big/ \sum_{k=1}^{L} \widehat{h}_{k,e}$$

- The perturbation of the sea surface height (SSH) $\zeta = h_1 - \Delta z_1$, where Δz_1 is the referential top layer thickness;
- Averaging (10)-(11) yields the barotropic thickness and momentum equations

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} \sum_{k=1}^{L} \widehat{h}_{k,e}) = 0, \tag{13}$$

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + f \overline{\boldsymbol{u}}^{\perp} = -g \nabla \zeta + \overline{\boldsymbol{G}}, \tag{14}$$

where the barotropic force $\overline{\boldsymbol{G}}$ includes all the other terms in the barotropic equation.

Baroclinic mode

- Baroclinic mode is the remaining motions including the advective motions and internal waves;
- The baroclinic velocity $\boldsymbol{u}_l'$ is defined as

$$\boldsymbol{u}_l' = u_l - \overline{u},\ l = 1, \ldots, L.$$

- Subtracting (14) from (11) yields the baroclinic momentum equation

$$\begin{aligned}\frac{\partial \boldsymbol{u}_l'}{\partial t} + \frac{1}{2}\nabla|\boldsymbol{u}_l|^2 + (\boldsymbol{k}\cdot\nabla\times\boldsymbol{u}_l)\boldsymbol{u}_l^{\perp} + f\boldsymbol{u}_l'^{\perp} + \omega_l\frac{\partial \boldsymbol{u}_l}{\partial z} \\ = g\nabla\zeta - \frac{1}{\rho_0}\nabla p_l + \nu_h\nabla^2\boldsymbol{u}_l + \frac{\partial}{\partial z}\left(\nu_v\frac{\partial \boldsymbol{u}_l}{\partial z}\right) - \overline{\boldsymbol{G}}.\end{aligned} \tag{15}$$

Two-level approach with ETD for the barotropic solve

Two-level coupling approach

Solve the primitive equations with *large globally uniform time steps* based on the two-level coupling framework, which contains three stages at each iterative step:

- **Stage 1: Advance the baroclinic velocity explicitly;**
- **Stage 2: Compute the barotropic velocity by ETD method;**
- **Stage 3: Update thickness, tracers, density and pressure explicitly.**

Stage 1 – Solve (15) for $\boldsymbol{u}'_{l,n+1}$:

- Firstly, ignore $\overline{\boldsymbol{G}}$ and predict the baroclinic velocity by forword-Euler scheme:

(16) $$\tilde{\boldsymbol{u}}'_{l,n+1} = \boldsymbol{u}'_{l,n} + \Delta t\left(-f\boldsymbol{u}^{\perp}_{l,n} + \boldsymbol{T}^{u} + g\nabla\zeta_n\right).$$

- Secondly, compute $\overline{\boldsymbol{G}}$:

(17) $$\overline{\boldsymbol{G}} = \frac{1}{\Delta t}\sum_{k=1}^{L}\widehat{h}_{k,e}\tilde{\boldsymbol{u}}'_{k,n+1}/\Sigma_{k=1}^{L}\widehat{h}_{k,e}.$$

- Lastly, correct the baroclinic velocity $\boldsymbol{u}'_{l,n+1} = \tilde{\boldsymbol{u}}'_{l,n+1} - \Delta t\overline{\boldsymbol{G}}$.

Two-level approach with ETD for the barotropic mode (Contd.)

Stage 2 – Solve (13)-(14) for $\overline{\boldsymbol{u}}$:

- Rewrite (13)-(14) as a system,

$$\frac{\partial \boldsymbol{v}}{\partial t} = -\boldsymbol{F}(\boldsymbol{v}) + \boldsymbol{b}, \tag{18}$$

where $\boldsymbol{v} = (\zeta, \overline{\boldsymbol{u}})^T$, $F(\boldsymbol{v}) = \left(\nabla \cdot (\overline{\boldsymbol{u}}\Sigma_{k=1}^{L}\widehat{h}_{k,e}), f\overline{\boldsymbol{u}}^{\perp} + g\nabla\zeta\right)^T$, and $\boldsymbol{b} = \left(0, \overline{\boldsymbol{G}}\right)^T$.

- The solution to Eq. (18) is

$$\boldsymbol{v}(t_{n+1}) = e^{\Delta t J_n}\boldsymbol{v}(t_n) + \int_0^{\Delta t} e^{(\Delta t-\tau)J_n}\boldsymbol{b}d\tau, \tag{19}$$

where $J_n = -DF(\boldsymbol{v}_n) = \begin{bmatrix} -\nabla \cdot (\overline{\boldsymbol{u}}\bullet) & -\nabla \cdot \left(\bullet\Sigma_{k=1}^{L}\widehat{h}_{k,e}\right) \\ -g\nabla\bullet & -f\boldsymbol{k} \times \bullet \end{bmatrix}$.

- We adopt the ETD-Rosenbrock to solve the barotropic equations.

Stage 3 – Update thickness, tracers, density and pressure.

The baroclinic channel test case

- SP1: barotropic mode advances to Δt –> average with data of t_n;
- SP2: barotropic mode advances to $2\Delta t$ –> average with data of t_n.

Comparison schemes:

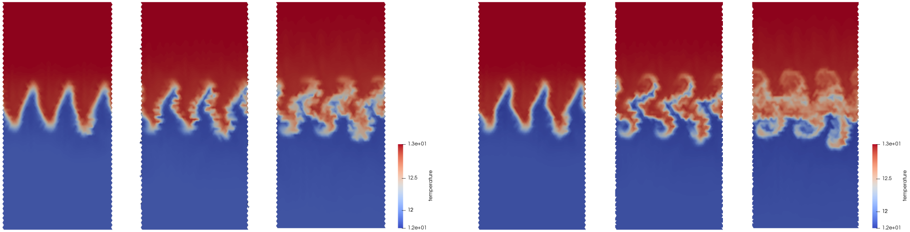
- ETD-SP1 and ETD-SP2 (one ETD stepping with $2\Delta t$) use uniform time stepping with the barotropic-baroclinic splitting;
- Currently in MPAS-Ocean: SP1 and SP2 (two different time steps with the barotropic-baroclinic splitting), RK4 (uniform time stepping without the barotropic-baroclinic splitting)

Test settings:

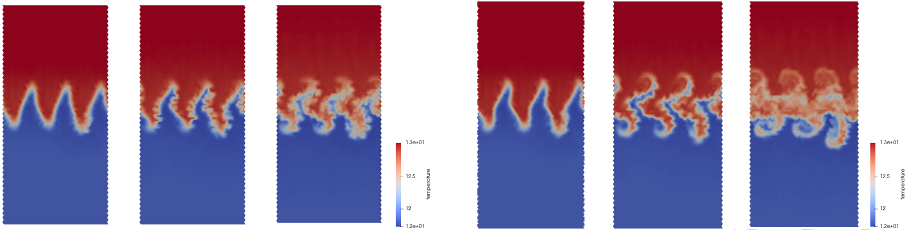
- A planar channel with 20 layers, 160 km longitudinal extent and 500 km latitudinal extent;
- A SCVT mesh with a 10 km solution containing 3920 cells, 11840 edges and 7920 vertices;
- 15-day-long simulation by parallel computing with 8 cores.

Comparison in temperature: Days 5, 10, and 15

ETD-SP1 (left) and ETD-SP2 (right) with $\Delta t = 60s$

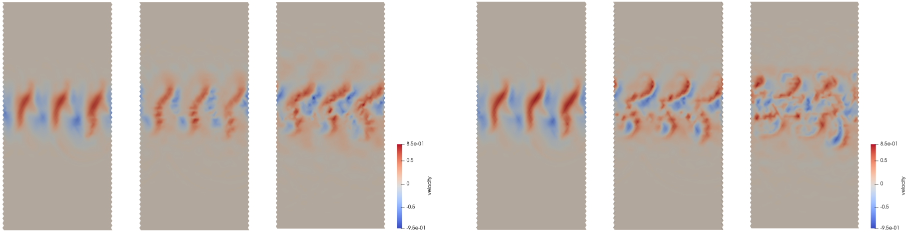


MPAS-SP1 (left) and MPAS-SP2 (right) with $\Delta t = 60s$ and $\Delta t_{\text{btr}} = 4s$

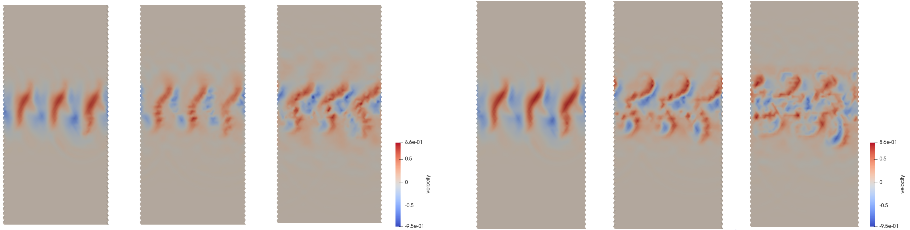


Comparison in velocity: Days 5, 10, and 15

ETD-SP1 (left) and ETD-SP2 (right) with $\Delta t = 60s$

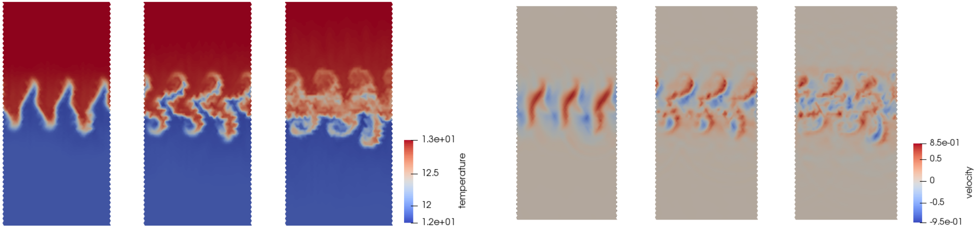


MPAS-SP1 (left) and MPAS-SP2 (right) with $\Delta t = 60s$ and $\Delta t_{\text{btr}} = 4s$



Benchmark results by RK4

RK4 with $\Delta t = 15s$



- **Observation**: ETD-SP1 subsamples the high frequency barotropic motions just like MPAS-SP1.

Quantitative comparisons

- We run 1 hour simulations with the following time step pairs: (60s, 16s), (30s, 8s), (15s, 4s), (8s,2s), and (4s, 1s).
- "a vs. b" means the numbers is calculated by $\frac{\|a-b\|_\infty}{\|b\|_\infty}$.

	ETD-SP1 vs. MPAS-SP1	ETD-SP1 vs. RK4	MPAS-SP1 vs. RK4
60s	4.27E-05	0.0244	0.0245
30s	1.95E-05	0.0245	0.0245
15s	1.20E-05	0.0242	0.0242
8s	6.41E-06	0.0247	0.0247
4s	2.64E-06	0.0251	0.0251
	ETD-SP2 vs. MPAS-SP2	ETD-SP2 vs. RK4	MPAS-SP2 vs. RK4
60s	0.0010	0.0190	0.0184
30s	0.0012	0.0171	0.0162
15s	0.0011	0.0137	0.0128
8s	0.0010	0.0106	0.0097
4s	0.0008	0.0074	0.0066

- Question: are ETD-SP2 and MPAS-SP2 second-order accurate in time theoretically?

Ongoing work

- Design and implement higher-order accurate parallel ETD coupled schemes for ocean dynamics;
- Implement the parallel ETD approach for tracer equations within the MPAS framework;
- Test high-resolution MPAS-Ocean cases;
- Design higher-order accurate multi-rate explicit time-stepping schemes with theoretical guarantees, which only need small modifications on the current schemes in MPAS;
- Integration of codes into MPAS.