‘Streamline’ Flux-form Semi-Lagrangian Methods for Scalar Transport

Darren Engwirda [1,2] {darren.engwirda@columbia.edu}

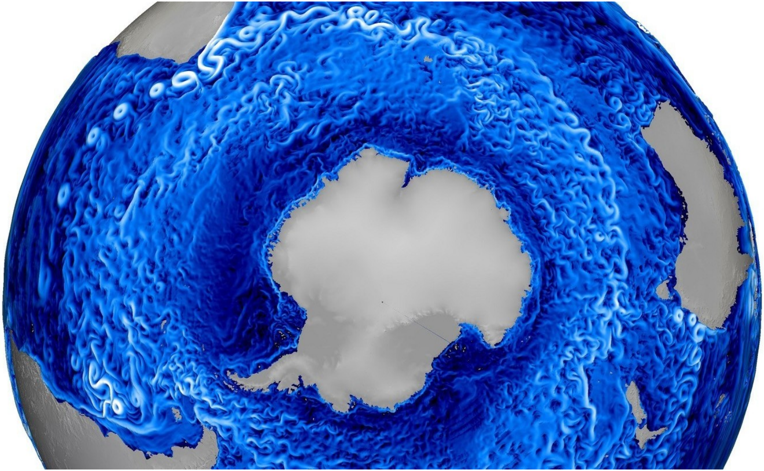
Mike Herzfeld + {CO}MPAS team [3]

[1] Center for Climate Systems Research, Columbia University, [2] NASA Goddard Institute for Space Studies, [3] Oceans and Atmospheres, CSIRO

SIAM Mathematical & Computational Issues in the Geosciences
Houston, March 2019

Motivation: unstructured schemes for geophysical flows

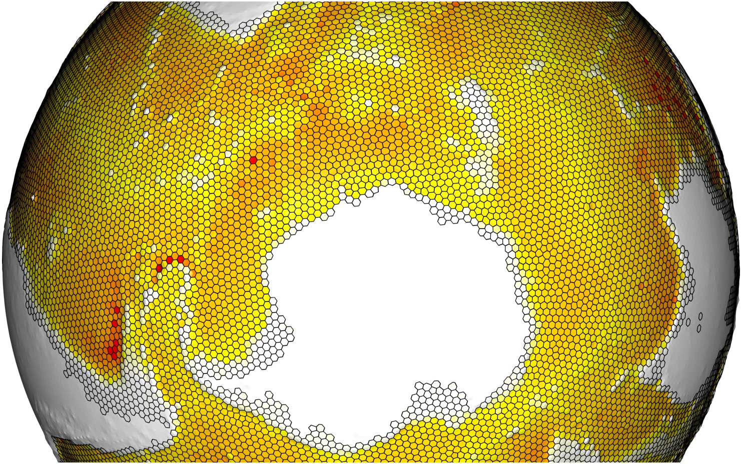
A current focus is the development and use of flexible, unstructured schemes for geophysical flows — **climate dynamics**, **ocean modelling**, etc...



**Eddy-permitting global ocean, showing dynamics in the ACC (MPAS-O team, LANL).

Motivation: unstructured schemes for geophysical flows

A current focus is the development and use of flexible, unstructured schemes for geophysical flows — **unstructured meshes**, **orthogonal polygons**, etc...

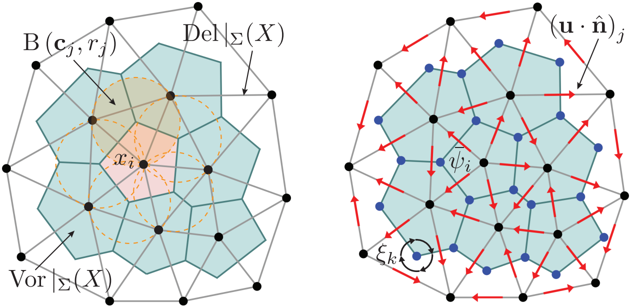


**Spherical Voronoi-type grid for global ocean simulation (MPAS-O team, LANL).

Formulation: staggered unstructured finite-volumes

An unstructured generalisation of Arwkawa **C-grid** scheme:

- Staggered finite-difference/finite-volume mimetic schemes.
- Vector components staggered on primal cell edges.
- Conserved quantities centred in dual control volumes.



**Staggered, mimetic finite-volume scheme used in the Model for Prediction Across Scales (MPAS-O), Ringler et al., 2013, and the Coastal Model for Prediction Across Scales {CO}MPAS, Herzfeld, Engwirda, Rizwi, 2019.

Main Aim: Try to design 'good' schemes for scalar transport:

- Applicable to general unstructured meshes (on the sphere).
- Efficient and accurate treatment for multiple tracers.
- Exactly conservative and shape-preserving (monotone).

Can we do better than 'standard' multi-stage Runge-Kutta schemes?

Semi-discrete scheme: finite-volume integration

The evolution of a conserved quantity $Q(\mathbf{x},t)$ in the presence of an advecting flow $\mathbf{u}(\mathbf{x},t)$ can be written

$$\partial_t Q(\mathbf{x},t) + \nabla\cdot\left(\mathbf{u}(\mathbf{x},t)\,Q(\mathbf{x},t)\right) = \mathrm{RHS}(Q(\mathbf{x},t),t) \tag{1}$$

The RHS contains various mixing/viscous effects, sources terms, etc.

Standard finite-volume discretisation obtained through integration over a discrete cell Ω

$$\partial_t \int_\Omega Q\,\mathrm{d}A + \oint_{\partial\Omega} (\hat{\mathbf{n}}\cdot\mathbf{u}\,Q)\,\mathrm{d}C = 0\,, \tag{2}$$

$$\partial_t \bar{Q} + A_\Omega^{-1} \sum_{\text{edges}} \int_e (\hat{\mathbf{n}}_e\cdot\mathbf{u}\,Q)\,\mathrm{d}l = 0\,. \tag{3}$$

Note that (3) is *exact* if the integration is done exactly.

Semi-discrete scheme: finite-volume integration

Various options exist to discretise in time — a popular '*standard*' approach is to apply Strong Stability Preserving Runge-Kutta (SSP-RK) methods to solve (3) as a set of ODE's. **More on this later!**

A different approach is to integrate in space *and* time directly

$$\bar{Q}^{n+1} = \bar{Q}^{n} - A_{\Omega}^{-1} \sum_{\text{edges}} \underbrace{\int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t}_{\text{'time-integrated' flux over edge}} \quad . \tag{4}$$

The 'time-integrated' flux represents the material advected across a cell boundary in a given time-step — a **Lagrangian** perspective.

→ Leads to **a class of 'flux-form' semi-Lagrangian methods (FFSL).**

Semi-discrete schemes: SSP-RK vs FFSL

2-stage SSP-RK approach:

$$F = A_\Omega^{-1} \sum_{\text{edges}} \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l$$

$$\bar{Q}_0 = \bar{Q}^n$$

$$\bar{Q}_1 = \bar{Q}_0 - \Delta t\, F(Q_0(\mathbf{x}))$$

$$\bar{Q}_2 = \frac{1}{2}\bar{Q}_0 + \frac{1}{2}\bar{Q}_1 - \frac{1}{2}\Delta t\, F(\bar{Q}_1)$$

$$\bar{Q}^{n+1} = \bar{Q}_2$$

FFSL approach:

$$F = A_\Omega^{-1} \sum_{\text{edges}} \int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t$$

$$\bar{Q}^{n+1} = \bar{Q}^n - F$$

Semi-discrete schemes: SSP-RK vs FFSL

3-stage SSP-RK approach:

$$F = A_{\Omega}^{-1} \sum_{\text{edges}} \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l$$

$$\bar{Q}_0 = \bar{Q}^n$$

$$\bar{Q}_1 = \bar{Q}_0 - \Delta t\, F(\bar{Q}_0)$$

$$\bar{Q}_2 = \frac{3}{4}\bar{Q}_0 + \frac{1}{4}\bar{Q}_1 - \frac{1}{4}\Delta t\, F(\bar{Q}_1)$$

$$\bar{Q}_3 = \frac{1}{3}\bar{Q}_0 + \frac{2}{3}\bar{Q}_1 - \frac{2}{3}\Delta t\, F(\bar{Q}_2)$$

$$\bar{Q}^{n+1} = \bar{Q}_3$$

More evaluations

FFSL approach:

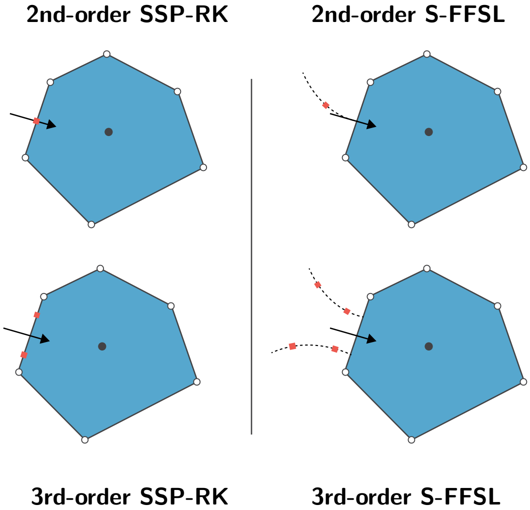
$$F = A_{\Omega}^{-1} \sum_{\text{edges}} \int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t$$

$$\bar{Q}^{n+1} = \bar{Q}^n - F$$

More complex integral

How to actually ‘do’ the integrals?

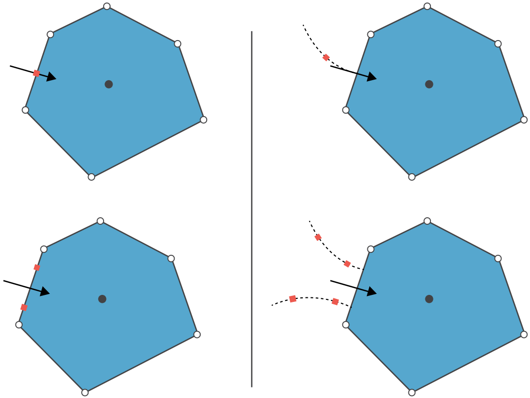
Approximate via quadrature — either in space-only (**SSP-RK**), or in space and time (**FFSL**).



How to actually ‘do’ the integrals?

The **‘S’** in **S-FFSL** denotes a ‘streamline’ method — the space-time integral is approximated by tracing edge quadrature point along streamlines.

2nd-order SSP-RK **2nd-order S-FFSL**



3rd-order SSP-RK **3rd-order S-FFSL**

How to actually 'do' the integrals?

The space-time integrals in the **S-FFSL** method can be approximated by 'nested' quadrature

$$F = A_\Omega^{-1} \sum_{\text{edges}} \int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t\,,$$

is discretised via

$$\int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t \to (\hat{\mathbf{n}}_e \cdot \mathbf{u}) \overbrace{\left(\sum_e w_e \underbrace{\sum_t w_t\, Q(\mathbf{x}_{e,t})}_{\text{integrate along streamline}} \right)}^{\text{integrate over area 'swept' by edge}} . \qquad (5)$$

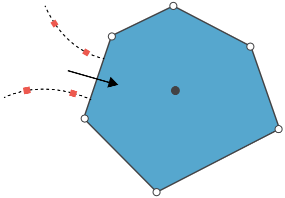
Here, w_e, w_t are quadrature weights, and the streamline position $\mathbf{x}_{e,t}$ is a solution to the standard Lagrangian 'back-trajectory' problem

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\mathbf{u}(\mathbf{x}, t)\,. \qquad (6)$$

How to actually ‘do’ the integrals?

With $\mathrm{CFL} \leq 1$, the space-time discretisation leads to a relatively simple scheme in practice

$$\int_t \int_e (\hat{\mathbf{n}}_e \cdot \mathbf{u}\, Q)\, \mathrm{d}l\, \mathrm{d}t \rightarrow (\hat{\mathbf{n}}_e \cdot \mathbf{u}) \overbrace{\left(\sum_e w_e \underbrace{\sum_t w_t\, Q(\mathbf{x}_{e,t})}_{\text{integrate along streamline}} \right)}^{\text{integrate over area ‘swept’ by edge}} .$$



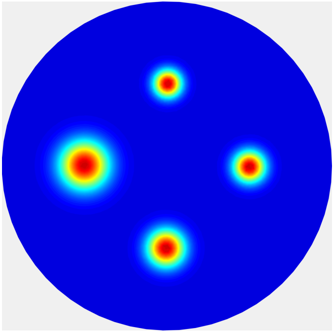
- Trace streamlines: $\mathrm{d}_t \mathbf{x}_{e,t} = -\mathbf{u}(\mathbf{x}, t)$.
- Eval. $Q(\mathbf{x}_{e,t})$ at quadrature points.
- Accumulate quadrature contributions.

With $\mathrm{CFL} \leq 1$, all operations are *localised* within the ‘upwind’ cell $\rightarrow$ **simple, efficient** discrete scheme.

Results: solid-body rotation

Initial experiments using standard **solid-body rotation** tests → assess deformation of (smooth) initial distribution after complete revolutions.

- Try **S-FFSL** + **SSP-RK** schemes.
- Try 2-stage + 3-stage **SSP-RK**.
- Piecewise linear reconstruction (PLM) → **2nd-order**, **monotone**.
- Barth-Jespersen type slope-limiter.
- Set Δt such that $\mathrm{CFL} \simeq 1$.
- Quasi-uniform, unstructured Centroidal Voronoi Tessellation.

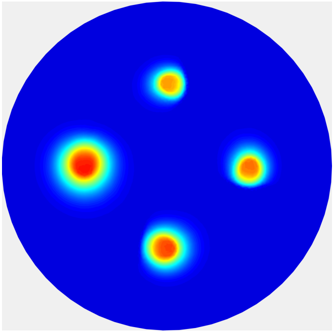


$$u_x = +y$$
$$u_y = -x$$

Results: solid-body rotation

Initial experiments using standard **solid-body rotation** tests → assess deformation of (smooth) initial distribution after complete revolutions.

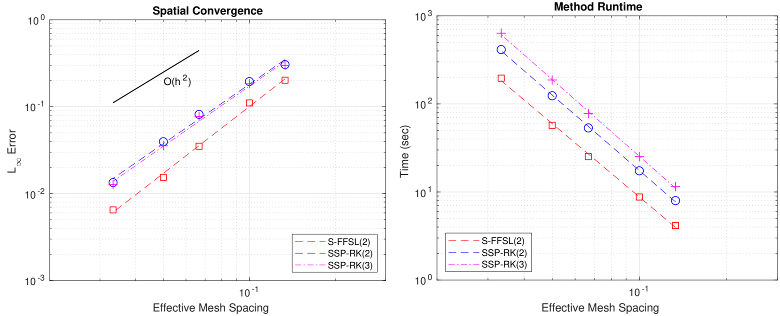
- Try **S-FFSL** + **SSP-RK** schemes.
- Try 2-stage + 3-stage **SSP-RK**.
- Piecewise linear reconstruction (PLM) → **2nd-order**, **monotone**.
- Barth-Jespersen type slope-limiter.
- Set Δt such that $\mathrm{CFL} \simeq 1$.
- Quasi-uniform, unstructured Centroidal Voronoi Tessellation.



$$u_x = +y$$
$$u_y = -x$$

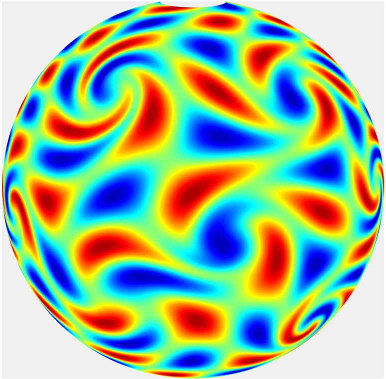
Results: solid-body rotation

All methods exhibit (expected) 2nd-order behaviour, predictable runtime scaling

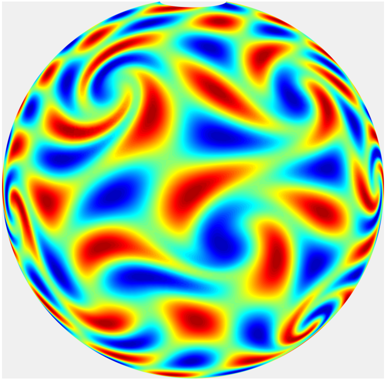


Overall, **S-FFSL(2)** is superior on this problem → **lower error, faster!**

The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



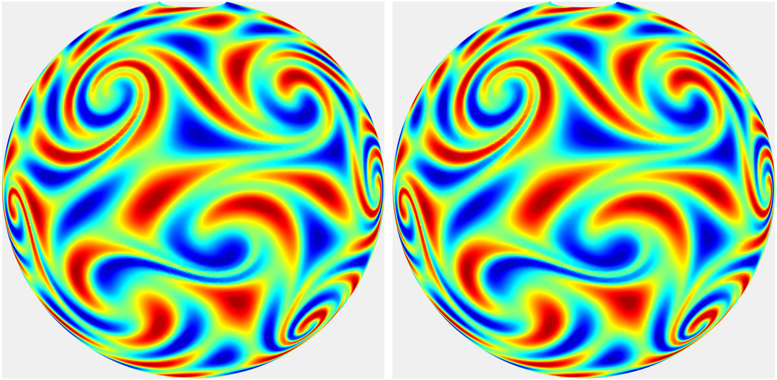
SSP-RK(3)



S-FFSL(2)

Results: swirling flow on a sphere

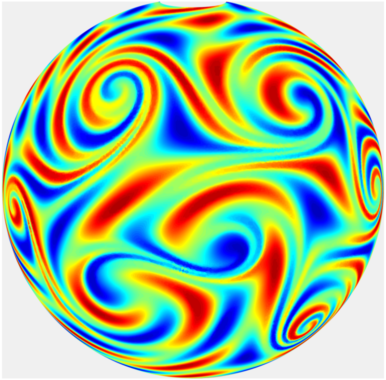
The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



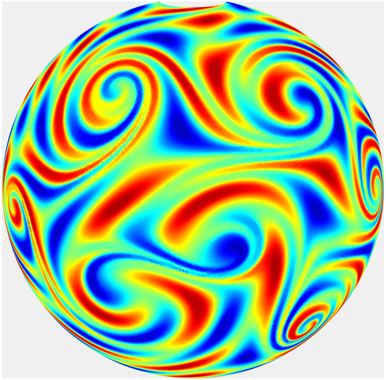
SSP-RK(3) **S-FFSL(2)**

Results: swirling flow on a sphere

The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



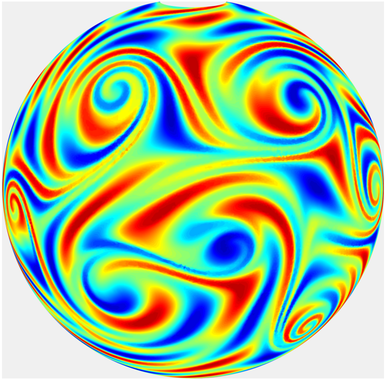
SSP-RK(3)



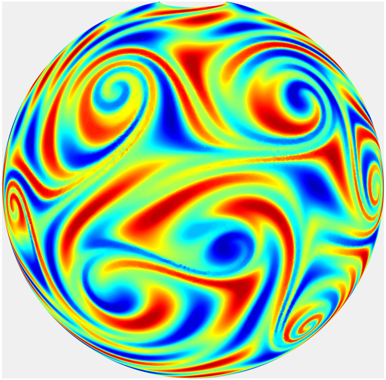
S-FFSL(2)

Results: swirling flow on a sphere

The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



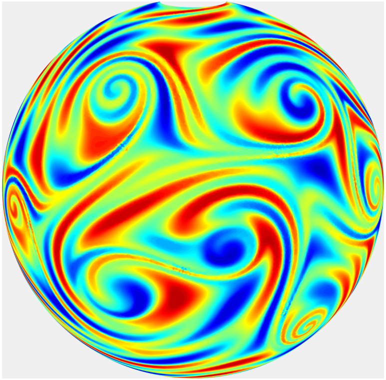
SSP-RK(3)



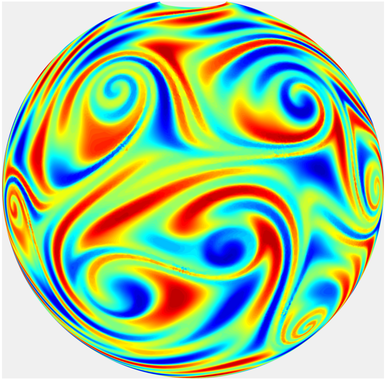
S-FFSL(2)

Results: swirling flow on a sphere

The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



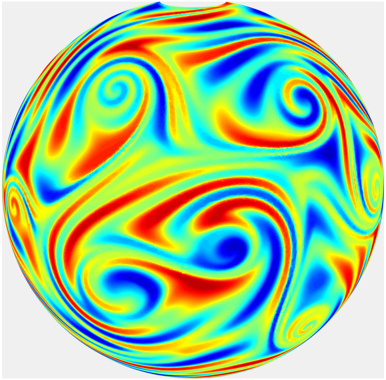
SSP-RK(3)



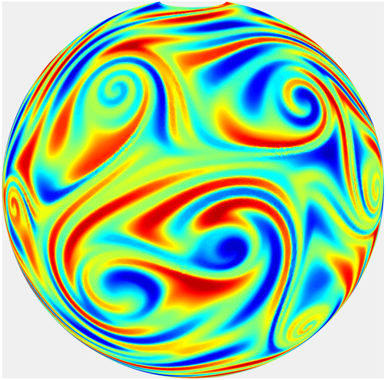
S-FFSL(2)

Results: swirling flow on a sphere

The (2nd-order) **S-FFSL** method gives near-identical results to a 3-stage **SSP-RK** formulation.



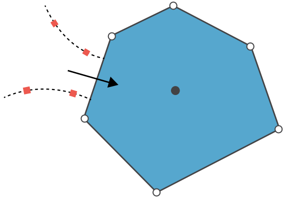
SSP-RK(3)



S-FFSL(2)

Conclusions: 'Streamline' FFSL methods

Presented a new 'Streamline' Flux-Form Semi-Lagrangian (**S-FFSL**) method for scalar transport



- Eval. space-time fluxes at cell edges.
- More accurate + faster than standard Eulerian SSP-RK schemes.
- Space-time approach better captures multi-dimensional effects.
- Leads to 'one-stage' time-stepping methods → reduce no. of (expensive) polynomial reconstructions.

Future work: higher-order spatial reconstruction (i.e. PPM, WENO, etc), parallel GCM implementation, 'long time-step' variants?