B-char: an efficient (and feasible!) approach for mass-conserving characteristic schemes in 2D and 3D

J. Droniou (Monash University)

Monash Workshop on Numerical Differential Equations 2020

Joint work with Hanz M. Cheng (formerly Monash, now Eindhoven University of Technology)

Australian Government
Australian Research Council

Discrete Functional Analysis: bridging pure and numerical mathematics

Plan

1. **The problem: numerical methods with inexact calculations**

2. **B-char method: cheap, and perfectly mass conservative**

3. **Numerical tests**
   - 2D tests
   - 3D tests

Plan

1. **The problem: numerical methods with inexact calculations**
2. B-char method: cheap, and perfectly mass conservative
3. Numerical tests
   - 2D tests
   - 3D tests

Linear advection model

$$\begin{cases} \phi \dfrac{\partial c}{\partial t} + \mathrm{div}(\mathbf{u}c) = 0 & \text{on } Q_T := \Omega \times (0,T), \\ c(\cdot, 0) = c_{\mathrm{ini}} & \text{on } \Omega. \end{cases}$$

- Ω: polygonal/polyhedral domain, with mesh $\mathcal{M}$.
- ϕ: porosity, $0 < \phi_* \le \phi \le \phi^*$, piecewise constant on mesh.
- $\mathbf{u}$: Darcy velocity, $\mathbf{u} \in L^\infty(0,T;L^2(\Omega))$, $\mathrm{div}\mathbf{u} = 0$ and $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$.
- c_{ini}: initial concentration, $c_{\mathrm{ini}} \in L^\infty(\Omega)$.

ELLAM method

Time steps: Time discretisation

$$0 = t^{(0)} < t^{(1)} < \ldots < t^{(N)} = T\,, \quad \text{with } \delta t^{(n+\frac{1}{2})} = t^{(n+1)} - t^{(n)}.$$

Let $\mathbf{u}^{(n+1)} \in L^2(\Omega)^d$ approximate $\mathbf{u}$ on $(t^{(n)}, t^{(n+1)})$, with $\mathrm{div}\mathbf{u}^{(n+1)} = 0$ and $\mathbf{u}^{(n+1)} \cdot \mathbf{n} = 0$ on $\partial\Omega$.

ELLAM method

Time steps: Time discretisation

$$0 = t^{(0)} < t^{(1)} < \ldots < t^{(N)} = T\,, \quad \text{with } \delta t^{(n+\frac{1}{2})} = t^{(n+1)} - t^{(n)}.$$

Let $\mathbf{u}^{(n+1)} \in L^2(\Omega)^d$ approximate $\mathbf{u}$ on $(t^{(n)}, t^{(n+1)})$, with $\mathrm{div}\mathbf{u}^{(n+1)} = 0$ and $\mathbf{u}^{(n+1)} \cdot \mathbf{n} = 0$ on $\partial\Omega$.

Test function: ψ satisfying

$$\phi\frac{\partial \psi}{\partial t} + \mathbf{u}^{(n+1)} \cdot \nabla\psi = 0 \quad \text{on } \Omega \times (t^{(n)}, t^{(n+1)})\,, \quad \psi(\cdot, t^{(n+1)}) \text{ given.}$$

ELLAM method

Time steps: Time discretisation

$$0 = t^{(0)} < t^{(1)} < \ldots < t^{(N)} = T\,, \quad \text{with } \delta t^{(n+\frac{1}{2})} = t^{(n+1)} - t^{(n)}.$$

Let $\mathbf{u}^{(n+1)} \in L^2(\Omega)^d$ approximate $\mathbf{u}$ on $(t^{(n)}, t^{(n+1)})$, with $\mathrm{div}\mathbf{u}^{(n+1)} = 0$ and $\mathbf{u}^{(n+1)} \cdot \mathbf{n} = 0$ on $\partial\Omega$.

Test function: ψ satisfying

$$\phi \frac{\partial \psi}{\partial t} + \mathbf{u}^{(n+1)} \cdot \nabla \psi = 0 \quad \text{on } \Omega \times (t^{(n)}, t^{(n+1)})\,, \quad \psi(\cdot, t^{(n+1)}) \text{ given.}$$

- Set $F_t(x)$ flow of $\mathbf{u}^{(n+1)}/\phi$, that is

$$\frac{dF_t(x)}{dt} = \frac{\mathbf{u}^{(n+1)}(F_t(x))}{\phi(F_t(x))}\,, \quad F_0(x) = x.$$

Then

$$\psi(x, t^{(n)}) = \psi(F_{\delta t^{(n+\frac{1}{2})}}(x), t^{(n+1)}).$$

ELLAM method

Test function: ψ satisfying

$$\phi\frac{\partial\psi}{\partial t}+\mathbf{u}^{(n+1)}\cdot\nabla\psi=0 \quad \text{on } \Omega\times(t^{(n)},t^{(n+1)}), \quad \psi(\cdot,t^{(n+1)}) \text{ given.}$$

- Set $F_t(x)$ flow of $\mathbf{u}^{(n+1)}/\phi$, that is

$$\frac{dF_t(x)}{dt}=\frac{\mathbf{u}^{(n+1)}(F_t(x))}{\phi(F_t(x))}, \quad F_0(x)=x.$$

Then

$$\psi(x,t^{(n)})=\psi(F_{\delta t^{(n+\frac{1}{2})}}(x),t^{(n+1)}).$$

Time stepping in ELLAM (=Eulerian Lagrangian Localised Adjoint Method):

$$\int_\Omega \phi(x)(c\psi)(x,t^{(n+1)})\,dx=\int_\Omega \phi(x)(c\psi)(x,t^{(n)})\,dx$$

ELLAM method: global and local mass conservation

Global mass conservation: make $\psi(x, t^{(n+1)}) \equiv 1$:

$$\int_\Omega \phi(x) c(x, t^{(n+1)})\, dx = \int_\Omega \phi(x) c(x, t^{(n)})\, dx.$$

Local mass conservation: since $\mathrm{div}\mathbf{u} = 0$,

$$\text{If } c(\cdot, t^{(n)}) = 1 \text{ then } c(\cdot, t^{(n+1)}) = 1.$$

ELLAM for piecewise constant approximations

- At each time, we are looking for $c_h(\cdot, t^{(n)}) = (c_M^{(n)})_{M\in\mathcal{M}}$ piecewise constant approximation of c on $\mathcal{M}$.
- Notation: the porous volume in a set A is

$$|A|_\phi = \int_A \phi.$$

ELLAM formulation: take $\psi(\cdot, t^{(n+1)}) = \mathbf{1}_K$ for a cell $K \in \mathcal{M}$:

$$|K|_\phi c_K^{(n+1)} = \sum_{M\in\mathcal{M}} |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi c_M^{(n)}.$$

Global and local mass conservation

$$|K|_\phi c_K^{(n+1)} = \sum_{M \in \mathcal{M}} |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi c_M^{(n)}.$$

Global and local mass conservation

$$|K|_\phi c_K^{(n+1)} = \sum_{M\in\mathcal{M}} |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi c_M^{(n)}.$$

Global mass conservation: OK by summing over K and using

$$\sum_{K\in\mathcal{M}} |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |M|_\phi.$$

Global and local mass conservation

$$|K|_\phi c_K^{(n+1)} = \sum_{M\in\mathcal{M}} |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi c_M^{(n)}.$$

Global mass conservation: OK by summing over K and using

$$\sum_{K\in\mathcal{M}} |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |M|_\phi.$$

Local mass conservation: OK because

$$\sum_{M\in\mathcal{M}} |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi.$$

ELLAM in practice: what needs to be computed

Transport of cells: K polygonal/polyhedral cell, but $F_{-\delta t^{(n+\frac{1}{2})}}(K)$ is a generic potato, that needs to be approximated...

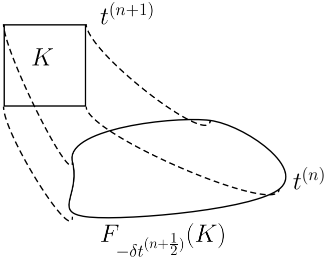


Figure: Exact (left) and approximated (right) trace-back of K.

ELLAM in practice: what needs to be computed

Intersection of regions: need to compute (porous volume of) $M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)$.

▶ Algorithms for areas of intersections of polygons (2D) are ok, but expensive.

▶ Algorithms for volume of intersections of polyhedras (3D) are terrible!

ELLAM in practice: revisiting mass conservation

- Global and local mass conservation are based on

$$\sum_{K\in\mathcal{M}} |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |M|_\phi \qquad \text{(global)},$$

$$\sum_{M\in\mathcal{M}} |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi \qquad \text{(local)}.$$

- Issue: we only compute $\widehat{K}$, and

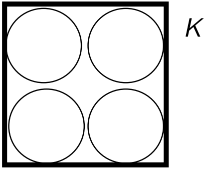
$$|M\cap \widehat{K}|_\phi \approx |M\cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi.$$

Not a problem for global mass conservation (as $(\widehat{K})_{K\in\mathcal{M}}$ *forms a partition of the domain), but* **breaks down local mass conservation**...

Plan

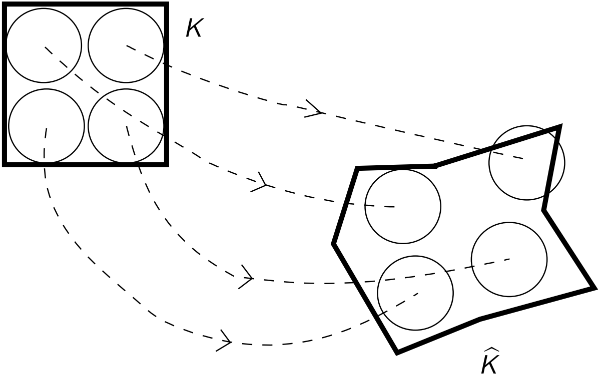
An original idea...

Approximate polygons/polyhedras by balls,



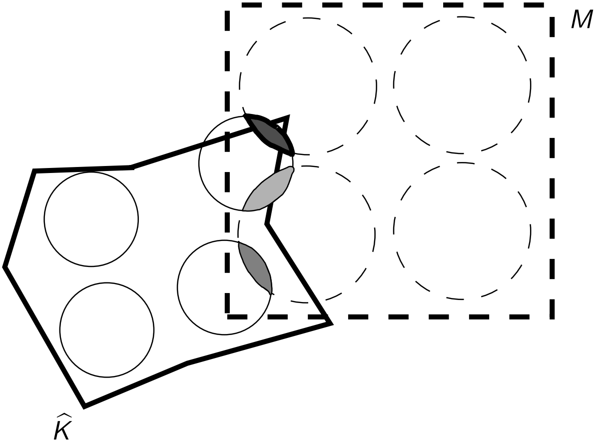
An original idea...

Approximate polygons/polyhedras by balls, track balls (keeping them as balls),



An original idea...

Approximate polygons/polyhedras by balls, track balls (keeping them as balls), intersect balls.



… that needs to be enhanced!

- Loss of volume in K when approximating by balls (gaps), and loss of volume when intersecting balls.

- Very inaccurate approximation of $\widehat{K}$ (and thus of $F_{-\delta t^{(n+\frac{1}{2})}}(K)$) by tracked balls.

$\rightsquigarrow$ bad solutions, clearly not conserving mass.

Initial adjustments

- Cell K with balls $(B_{K,s})_{s=1,\ldots,n_K}$.

Initial adjustments

- Cell K with balls $(B_{K,s})_{s=1,\ldots,n_K}$.

Distribution of porous volume: introduce *porous density* ρ_K, constant during evolution, such that

$$\rho_K \sum_{s=1}^{n_K} |B_{K,s}|_\phi = |K|_\phi.$$

- $\rho_K|B_{K,s}|_\phi$ *equivalent* porous volume inside ball.

Initial adjustments

- Cell K with balls $(B_{K,s})_{s=1,\ldots,n_K}$.

Distribution of porous volume: introduce *porous density* ρ_K, constant during evolution, such that

$$\rho_K \sum_{s=1}^{n_K} |B_{K,s}|_\phi = |K|_\phi.$$

- $\rho_K |B_{K,s}|_\phi$ *equivalent* porous volume inside ball.

Tracking of balls: assuming ϕ constant, the volume (and radius) of $B_{K,s}$ remains constant during tracking *(generalised Liouville theorem)*.

Initial adjustments

Intersections of balls without loss of mass: straight intersection of balls in $\widehat{K}$ and M leads to

$$|\widehat{K} \cap M|_\phi \approx \sum_s \sum_m \rho_M \phi_M |\widehat{B}_{K,s} \cap B_{M,m}|.$$

Initial adjustments

Intersections of balls without loss of mass: straight intersection of balls in $\widehat{K}$ and M leads to

$$|\widehat{K}\cap M|_\phi \approx \sum_s \sum_m \rho_M \phi_M |\widehat{B}_{K,s}\cap B_{M,m}|.$$

▶ But loss of mass through intersection of balls. So we compute the fraction of mass of $\widehat{B}_{K,s}$ that comes from $B_{M,m}$:

$$f_{K,s,M,m} = \frac{\rho_M \phi_M |\widehat{B}_{K,s}\cap B_{M,m}|}{\sum_{L\in\mathcal{M}} \sum_{\ell=1}^{n_L} \rho_L \phi_L |\widehat{B}_{K,s}\cap B_{L,\ell}|}$$

and we set

$$|M\cap \widehat{K}|_\phi \approx V_{\widehat{K},M} = \sum_{s=1}^{n_K} \rho_K \widehat{\phi}_{K,s} |\widehat{B}_{K,s}| \sum_{m=1}^{n_M} f_{K,s,M,m}.$$

Mass conservations?

Local mass conservation: came from

$$\sum_{M} |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi.$$

Mass conservations?

Local mass conservation: came from

$$\sum_M |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi.$$

We therefore need

$$\sum_M V_{\widehat{K},M} = |K|_\phi.$$

OK because $\sum_M \sum_m f_{K,s,M,m} = 1$.

Mass conservations?

Local mass conservation: came from

$$\sum_M |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi.$$

We therefore need

$$\sum_M V_{\widehat{K},M} = |K|_\phi.$$

OK because $\sum_M \sum_m f_{K,s,M,m} = 1$.

Global mass conservation: came from

$$\sum_K |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |M|_\phi.$$

Mass conservations?

Local mass conservation: came from

$$\sum_M |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |K|_\phi.$$

We therefore need

$$\sum_M V_{\widehat{K},M} = |K|_\phi.$$

OK because $\sum_M \sum_m f_{K,s,M,m} = 1$.

Global mass conservation: came from

$$\sum_K |M \cap F_{-\delta t^{(n+\frac{1}{2})}}(K)|_\phi = |M|_\phi.$$

We therefore need

$$\sum_K V_{\widehat{K},M} = |M|_\phi.$$

KO!

Second adjustment: redistributions

Global: $\displaystyle\sum_{K} V_{\widehat{K},M} = |M|_\phi.$ Local: $\displaystyle\sum_{M} V_{\widehat{K},M} = |K|_\phi.$

Second adjustment: redistributions

Global: $\displaystyle\sum_K V_{\widehat{K},M} = |M|_\phi.$ Local: $\displaystyle\sum_M V_{\widehat{K},M} = |K|_\phi.$

- Step 0: set $V^{(0)}_{\widehat{K},M} = V_{\widehat{K},M}$.

Second adjustment: redistributions

Global: $\displaystyle\sum_K V_{\widehat{K},M} = |M|_\phi.$ Local: $\displaystyle\sum_M V_{\widehat{K},M} = |K|_\phi.$

For $n = 0, \ldots, N$, iterate:

- Step 1: redistribute to get global mass conservation

$$V_{\widehat{K},M}^{(n+\frac{1}{2})} = \frac{|M|_\phi}{\sum_R V_{\widehat{R},M}^{(n)}} V_{\widehat{K},M}^{(n)}.$$

Second adjustment: redistributions

Global: $\displaystyle\sum_K V_{\widehat{K},M} = |M|_\phi.$ Local: $\displaystyle\sum_M V_{\widehat{K},M} = |K|_\phi.$

For $n = 0, \ldots, N$, iterate:

- Step 1: redistribute to get global mass conservation

$$V_{\widehat{K},M}^{(n+\frac{1}{2})} = \frac{|M|_\phi}{\sum_R V_{\widehat{R},M}^{(n)}} V_{\widehat{K},M}^{(n)}.$$

- Step 2: redistribute to get local mass conservation

$$V_{\widehat{K},M}^{(n+1)} = \frac{|K|_\phi}{\sum_L V_{\widehat{K},L}^{(n+\frac{1}{2})}} V_{\widehat{K},M}^{(n+\frac{1}{2})}.$$

Second adjustment: redistributions

- Error in global/local mass tends to reduce at each iteration...
but very slowly after the first few steps.

Second adjustment: redistributions

- Error in global/local mass tends to reduce at each iteration... but very slowly after the first few steps.

Achieving exact conservation: after $n \sim 10$, stop iterations and find, in the vicinity of the current $(V^{(n)}_{\widehat{K},M})_{K,M}$, one solution to the global and local mass conservation equations.

Second adjustment: redistributions

Achieving exact conservation: after $n \sim 10$:

Find $\mathbf{x} = (x_{\widehat{K},M})_{K,M}$ such that:

- $((1 + x_{\widehat{K},M})V^{(n)}_{\widehat{K},M})_{K,M}$ exactly satisfies the global and local mass balance equations,
- $0 \le 1 + x_{\widehat{K},M} \le 2$,
- $|\mathbf{x}|^2$ is minimal.

Then, use $V_{\widehat{K},M} = (1 + x_{\widehat{K},M})V^{(n)}_{\widehat{K},M}$ as porous volumes of cell intersections.

Second adjustment: redistributions

Achieving exact conservation: after $n \sim 10$:

Find $\mathbf{x} = (x_{\widehat{K},M})_{K,M}$ such that:

- $((1 + x_{\widehat{K},M})V^{(n)}_{\widehat{K},M})_{K,M}$ exactly satisfies the global and local mass balance equations,
- $0 \leq 1 + x_{\widehat{K},M} \leq 2$,
- $|\mathbf{x}|^2$ is minimal.

Then, use $V_{\widehat{K},M} = (1 + x_{\widehat{K},M})V^{(n)}_{\widehat{K},M}$ as porous volumes of cell intersections.

- $(x_{\widehat{K},M})_{K,M}$ are ♯cells × ♯cells unknowns, but the actual minimisation problem is much smaller (only a few $V^{(n)}_{\widehat{K},M}$ are non-zero).

Plan

1. The problem: numerical methods with inexact calculations
2. B-char method: cheap, and perfectly mass conservative
3. **Numerical tests**
   - 2D tests
   - 3D tests

Plan

1. The problem: numerical methods with inexact calculations
2. B-char method: cheap, and perfectly mass conservative
3. Numerical tests
   - 2D tests
   - 3D tests

Comparison with "polygonal" ELLAM: translation

- "Polygonal" ELLAM: classical approach, computing $\widehat{K}$ and intersection $M \cap \widehat{K}$.
- B-char: 4 balls in each cell.

Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on $(\frac{1}{16}, \frac{5}{16}) \times (\frac{1}{16}, \frac{5}{16})$, velocity $\mathbf{u} = (\frac{1}{16}, 0)$, final time $T = 8$.

Comparison with "polygonal" ELLAM: translation

- "Polygonal" ELLAM: classical approach, computing $\widehat{K}$ and intersection $M \cap \widehat{K}$.
- B-char: 4 balls in each cell.

Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on $(\frac{1}{16}, \frac{5}{16}) \times (\frac{1}{16}, \frac{5}{16})$, velocity $\mathbf{u} = (\frac{1}{16}, 0)$, final time $T = 8$.

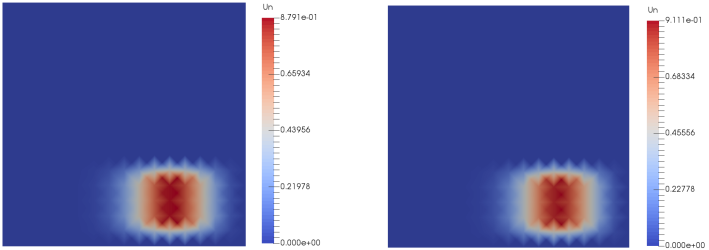


Figure: $16{\times}16$ grid, $\delta t = 0.8$ (left: polygonal; right: B-char).

Comparison with "polygonal" ELLAM: translation

- "Polygonal" ELLAM: classical approach, computing $\widehat{K}$ and intersection $M \cap \widehat{K}$.
- B-char: 4 balls in each cell.

Test case: $\Omega = (0,1)^2$, $c_{\text{ini}} = 1$ on $(\frac{1}{16}, \frac{5}{16}) \times (\frac{1}{16}, \frac{5}{16})$, velocity $\mathbf{u} = (\frac{1}{16}, 0)$, final time $T = 8$.

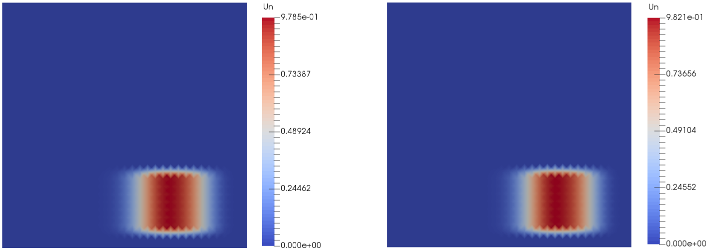


Figure: 32×32 grid, $\delta t = 0.4$ (left: polygonal; right: B-char).

Comparison with "polygonal" ELLAM: translation

- "Polygonal" ELLAM: classical approach, computing $\widehat{K}$ and intersection $M \cap \widehat{K}$.
- B-char: 4 balls in each cell.

Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on $(\frac{1}{16}, \frac{5}{16}) \times (\frac{1}{16}, \frac{5}{16})$, velocity $\mathbf{u} = (\frac{1}{16}, 0)$, final time $T = 8$.

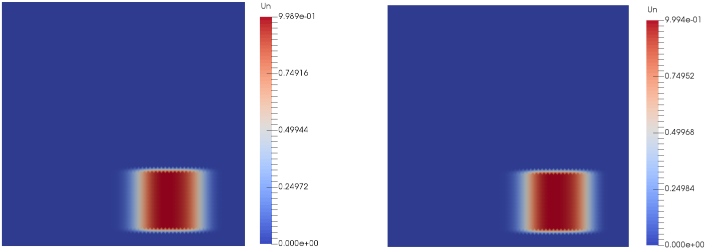


Figure: 64×64 grid, $\delta t = 0.2$ (left: polygonal; right: B-char).

Comparison with “polygonal” ELLAM: translation

- “Polygonal” ELLAM: classical approach, computing $\widehat{K}$ and intersection $M \cap \widehat{K}$.
- B-char: 4 balls in each cell.

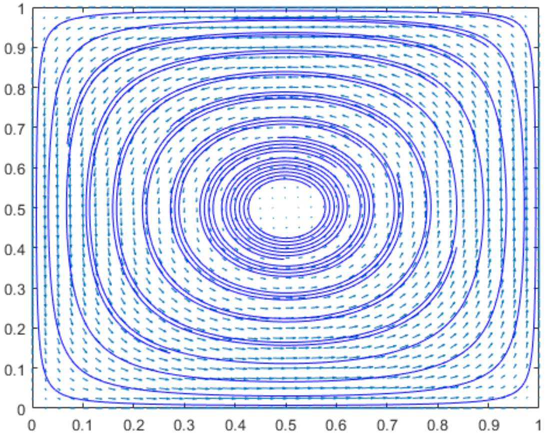
Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on $(\frac{1}{16}, \frac{5}{16}) \times (\frac{1}{16}, \frac{5}{16})$, velocity $\mathbf{u} = (\frac{1}{16}, 0)$, final time $T = 8$.

		Polygonal		B-char	
Mesh	δt	CPU (1 step)	L^2 error	CPU (1 step)	L^2 error
16×16	0.8	0.5s	3.7e-01	0.1s	3.8e-01
32×32	0.4	6.5s	3.2e-01	0.4s	3.3e-01
64×64	0.2	97.4s	2.7e-01	3.5s	2.9e-01

Table: CPU runtime and errors

Comparison with “polygonal” ELLAM: rotation

Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on disc of center $(\frac{1}{4}, \frac{3}{4})$ and radius $\frac{1}{8}$, final time $T = 8$. Streamlines of velocity:



Comparison with "polygonal" ELLAM: rotation

Test case: $\Omega = (0,1)^2$, $c_{\rm ini} = 1$ on disc of center $(\frac{1}{4}, \frac{3}{4})$ and radius $\frac{1}{8}$, final time $T = 8$.

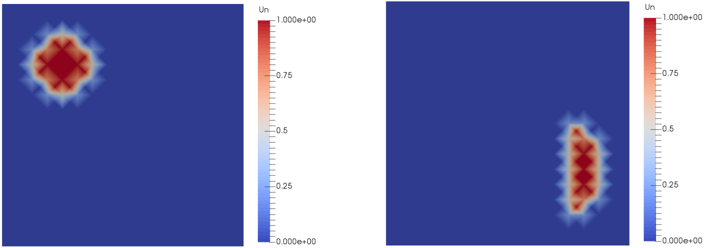


Figure: Initial condition (left), final solution (right).

Comparison with “polygonal” ELLAM: rotation

Results:

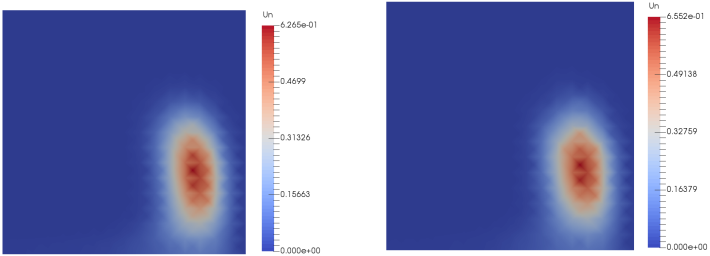


Figure: 16×16 grid, $\delta t = 0.8$ (left: polygonal; right: B-char).

Comparison with “polygonal” ELLAM: rotation

Results:

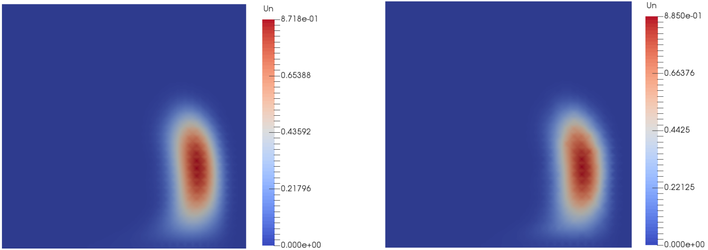


Figure: 32×32 grid, $\delta t = 0.4$ (left: polygonal; right: B-char).

Comparison with “polygonal” ELLAM: rotation

Results:

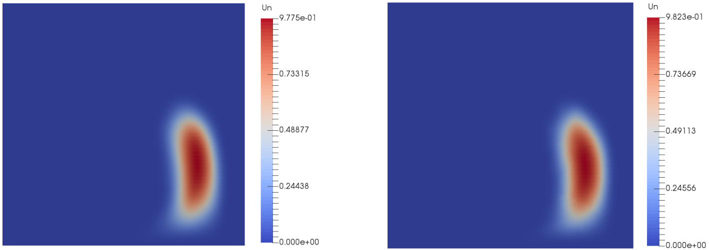


Figure: 64×64 grid, $\delta t = 0.2$ (left: polygonal; right: B-char).

Comparison with “polygonal” ELLAM: rotation

Results:

		Polygonal		B-char	
Mesh	δt	CPU (1 step)	L^2 error	CPU (1 step)	L^2 error
16×16	0.8	2.7s	5.1e-01	0.2s	5.1e-01
32×32	0.4	43s	4.2e-01	1.3s	4.1e-01
64×64	0.2	701s	3.6e-01	14.5s	3.6e-01

Table: CPU runtime and errors

Solid body rotation

Velocity: simple rotation around the center of $\Omega = (0,1)^2$.

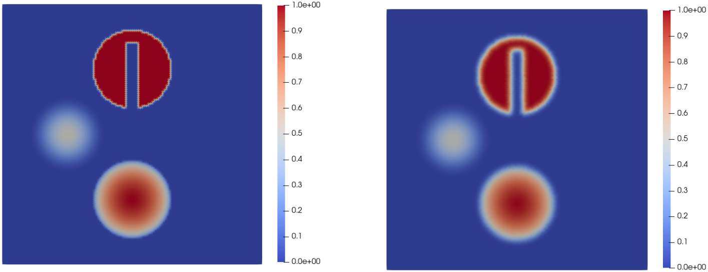


Figure: Solid body rotation on a 128×128 mesh (left: initial condition; right: numerical solution at $T = 2\pi$).

- Underlying ELLAM discretisation allows for larger time steps $\delta t = \frac{2\pi}{10}$ (in literature, usually, $\delta t \leq \frac{2\pi}{810}$).

Deformational flow

Velocity: velocity reverses at half-time $T/2$:

$$\mathbf{u} = (\sin^2(\pi x)\sin(2\pi y)\cos(\pi t/T), -\sin^2(\pi y)\sin(2\pi x)\cos(\pi t/T)).$$

Deformational flow

Results:

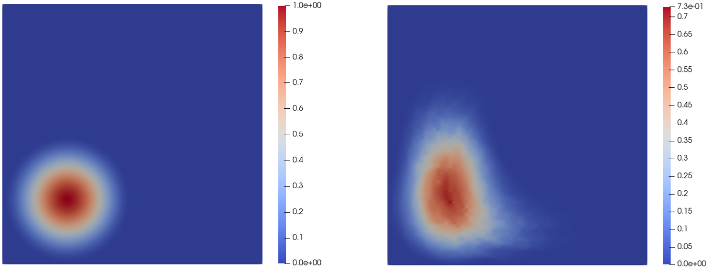


Figure: 64×64 mesh, $\delta t = 0.5$ (left: initial condition; right: numerical solution at $T = 5$).

Deformational flow

Results:

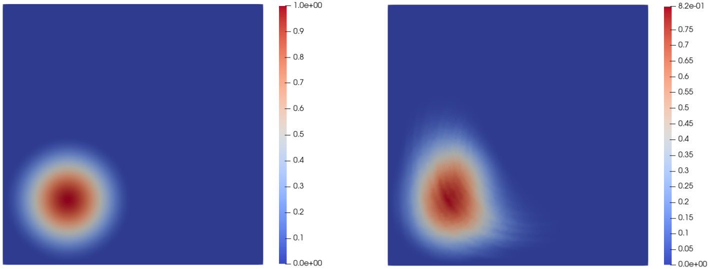
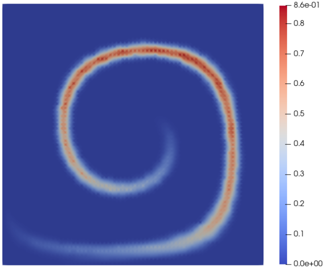


Figure: 128×128 mesh, $\delta t = 0.25$ (left: initial condition; right: numerical solution at $T = 5$).

Deformational flow

Results:



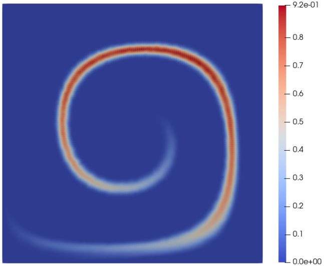


Figure: At halftime $T = 2.5$ (left: 64×64 cells; right: 128×128 cells).

Plan

1. The problem: numerical methods with inexact calculations

2. B-char method: cheap, and perfectly mass conservative

3. **Numerical tests**
   - 2D tests
   - 3D tests

Setting

- $\Omega = (0,1)^3$, $T = 8$.
- B-char with 8 balls per cell, 16^3 mesh, $\delta t = 0.8$.
- 3 test cases:
  1. Piecewise constant $c_{\rm ini}$ in cube, velocity: translation in x.
  2. Piecewise constant $c_{\rm ini}$ in cylinder, velocity: rotation & stretching in (x,y), translation in z.
  3. Continuous bump $c_{\rm init}$, same velocity as in 2.

Results

Test case	δt	CPU time (one time step)	L^1 error	L^2 error
1	0.8	37.2s	4.8e-01	4.1e-01
2	0.8	63.5s	9.6e-01	6.2e-01
3	0.8	63.2s	2.4e-01	2.4e-01

Table: CPU runtime and errors in 3D.

Bibliography

Main paper:

- H. M. Cheng and J. Droniou, "An efficient implementation of mass conserving characteristic-based schemes in 2D and 3D". To appear in SIAM J. Sci. Comput.

Other references:

- M. A. Celia, T. F. Russell, I. Herrera, and R. E. Ewing. "An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation". Adv. Water Resources, 13(4):187–206, 1990.
- T. Arbogast and C. Huang. "A fully mass and volume conserving implementation of a characteristic method for transport problems". SIAM J. on Sci. Comput., 28(6):2001–2022, 2006.
- T. Arbogast and C.-S. Huang. "A fully conservative Eulerian-Lagrangian method for a convection-diffusion problem in a solenoidal field". J. Comput. Phys., 229(9):3415–3427, 2010.
- H. M. Cheng, J. Droniou, and K.-N. Le. "Convergence analysis of a family of ELLAM schemes for a fully coupled model of miscible displacement in porous media". Numer. Math., 141(2):353–397, 2019.
- H. M. Cheng, J. Droniou, and K.-N. Le. "A combined GDM–ELLAM–MMOC scheme for advection dominated PDEs". https://arxiv.org/abs/1805.05585, 35p, 2018.
- M. D'Elia, D. Ridzal, K. J. Peterson, P. Bochev, and M. Shashkov. "Optimization-based mesh correction with volume and convexity constraints". J. Comput. Phys., 313:455–477, 2016.

Thanks.