

Reflections on three decades of science and camaraderie

Thomas F. Russell
National Science Foundation

In honor of Jean Roberts and Jérôme Jaffré

MSPM-JRJJ-2014:
Modeling and Simulation in Porous Media

INRIA-Rocquencourt, Le Chesnay, France
December 8-9, 2014

Shared history

- *Projets de restauration*
 - Tom (by Jim Douglas)
 - Jean (by Jim Douglas)
 - Tom's French (by Jérôme)
- *N.B.:* Jérôme also could have chosen to be a *restaurateur gastronomique*, but fortunately for science, he didn't
- A few stories ...
- No new results to report ...

My real job for NSF (1)

Dear Colleague Letter: Integrated NSF Support Promoting Interdisciplinary Research and Education (INSPIRE)

BACKGROUND

The Integrated NSF Support Promoting Interdisciplinary Research and Education (INSPIRE) pilot seeks to support bold interdisciplinary projects in all NSF-supported areas of science, engineering, and education research. INSPIRE has no targeted themes and serves as a funding mechanism for proposals that are required both to be interdisciplinary and to exhibit potentially transformative research (IDR and PTR, respectively). Complementing existing NSF efforts, INSPIRE was created to handle proposals whose:

- Scientific advances lie outside the scope of a single program or discipline, such that substantial funding support from more than one program or discipline is necessary.
- Lines of research promise transformational advances.
- Prospective discoveries reside at the interfaces of disciplinary boundaries that may not be recognized through traditional review or co-review.

To receive funding as an INSPIRE-appropriate project, all three criteria must be met. INSPIRE is not intended to be used for interdisciplinary projects that can be accommodated within other NSF funding mechanisms or that continue well-established practices.

The implementation of the INSPIRE pilot is based on two overarching goals:

Goal 1: To emphasize to the science, mathematics, engineering and education research community that NSF is welcoming to bold, unconventional ideas incorporating creative interdisciplinary approaches. INSPIRE seeks to attract unusually creative high-risk/high-reward "out of the box" interdisciplinary proposals.

Goal 2: To provide NSF Program Officers (POs) with additional tools and support to engage in cross-cutting collaboration and risk-taking in managing their awards portfolios.

INSPIRE supports projects that lie at the intersection of traditional disciplines, and is intended to 1) attract unusually creative high-risk / high-reward interdisciplinary proposals; 2) provide substantial funding, not limited to the exploratory stage of the pursuit of novel ideas (unlike NSF's EARly-concept Grants for Exploratory Research, or EAGER); and 3) be open to all NSF-supported areas of science, mathematics, engineering, and education research. NSF will initiate an external formative assessment to test whether the INSPIRE pilot is achieving program and portfolio-level goals.

My real job for NSF (2) “Stampede”



[Search Awards](#)

[Recent Awards](#)

[Presidential and Honorary Awards](#)

[About Awards](#)

How to Manage Your Award

[Grant Policy Manual](#)

[Grant General Conditions](#)

[Cooperative Agreement Conditions](#)

[Special Conditions](#)

[Federal Demonstration Partnership](#)

[Policy Office Website](#)

Award Abstract #1134872

Enabling, Enhancing, and Extending Petascale Computing for Science and Engineering

NSF Org:	ACI Division of Advanced CyberInfrastructure
Initial Amendment Date:	September 15, 2011
Latest Amendment Date:	September 15, 2014
Award Number:	1134872
Award Instrument:	Cooperative Agreement
Program Manager:	Thomas F. Russell ACI Division of Advanced CyberInfrastructure CSE Directorate for Computer & Information Science & Engineering
Start Date:	September 1, 2011
Expires:	August 31, 2017 (Estimated)
Awarded Amount to Date:	\$51,500,000.00
Investigator(s):	Daniel Stanzione dan@tacc.utexas.edu (Principal Investigator) John Boisseau (Former Principal Investigator) Niall Gaffney (Co-Principal Investigator) William Barth (Co-Principal Investigator) Tommy Minyard (Co-Principal Investigator) Daniel Stanzione (Former Co-Principal Investigator) Karl Schulz (Former Co-Principal Investigator)

Outline

(in defense of operator splitting)

- Operator splitting “decoupling” philosophy for modeling of complex multiphase systems
- Examples
 - 2-phase 2-component flow
 - Choice of multicomponent primary variables
 - Space-time AMR, multidomain / scale / grid / physics
 - Lagrangian transport, Eulerian diffusion, etc.
- Adjoint \leftrightarrow particle; multiphase multicomponent
- Generic approach
- Potential payoffs
- Mixed methods on hexahedra

Modeling of complex multiphase systems

- Model complexity – processes, domains, phases, components
- Include important couplings to maintain fidelity to physics
- Computational complexity – too expensive to keep strong couplings throughout
- Thus: need “smart”, robust decoupling approaches to computation
- Intelligent operator splitting: decouple judiciously (accurately) at intermediate steps – efficient computations on weakly coupled subsystems
- Restore couplings in final result (it’s not just model reduction; the complex whole is not the sum of the parts) – converge nonlinear residual to 0 as fast as possible

2-phase 2-component flow

- Water / oil or water / NAPL
- Phase velocity V_α : strong function of phase saturation S_α
- Total velocity V : comparatively weak function of phase saturation
- Small capillary pressure \rightarrow phase pressure p_α strong function of phase pressure p_β
- Thus: *Model* (V, S_α) is better for decoupling than *Model* (p_α, p_β) [different from (p_α, p_c)]

Another familiar example

- Choice of primary variables in a multiphase multicomponent system (compositional oil reservoir model)
- Lump certain subsets of components into pseudocomponents
- Stronger phase-behavior coupling **within** these subsets than **between** the subsets

Space-time adaptive mesh refinement

(multidomain / scale / grid / physics)

- Space-time refined patches, non-matching, hierarchical, etc. (adaptive → nonlinear)
- Level of resolution such that couplings (exchange terms) with larger system are nearly static or linear at that scale – not really **weak** coupling, but **tractable** coupling
- Can be preconditioned by linearized terms
- Thus: rapid nonlinear convergence

Mass transfer / reactions and transport

- Lagrangian frame of reference
- Follow moving fluid mixtures in each phase
- Expect weaker coupling of transport to phase behavior and reactions than in Eulerian frame
- *Basic idea of Eulerian-Lagrangian methods:* honor physical constraints on efficiency, avoid artificial mathematical constraints (e.g., CFL)

Multiphase multicomponent system

- **Time step:** compute nonlinear residual⁽⁰⁾ and Jacobian, then for Newton iteration⁽¹⁾ to convergence do
 - Solve pressure (flow) equation for phase velocities V_α (non-Newt. OK)
 - Use explicit Lagrangian algorithm to advect each component $i = 1, \dots, n_i$ in each phase $\alpha = 1, \dots, n_\alpha$, putting resulting masses on right-hand sides of appropriate cell component equations (almost like source/sink)
 - “Flash” each cell to re-equilibrate phases (or apply kinetics over the time step), changing the RHSs (like source/sink); similarly for other reactions
 - Solve for phase saturations, component concentrations, etc., accounting for remaining physical processes, if any (diffusion, dispersion, thermal conduction, sorption, ...)
 - Compute nonlinear residual^(k) and Jacobian
- **Remark:** most of this is done in existing non-simultaneous-implicit codes; main change is substitution of advection algorithm
- **Remark:** mass is conserved; volume might not be conserved at intermediate iterations; reminiscent of Watts (c. 1983) self-correcting volume errors → full volume convergence might not be necessary
- **Our focus:** inside iteration^(k), advect component i in phase α , known velocity V_α , no mass transfer; thus all components in phase α advect together at the same velocity

Modified method of characteristics (MMOC)

“Eulerian-Lagrangian”, “Semi-Lagrangian”

MMOC

EULERIAN-LAGRANGIAN METHOD (ELM)

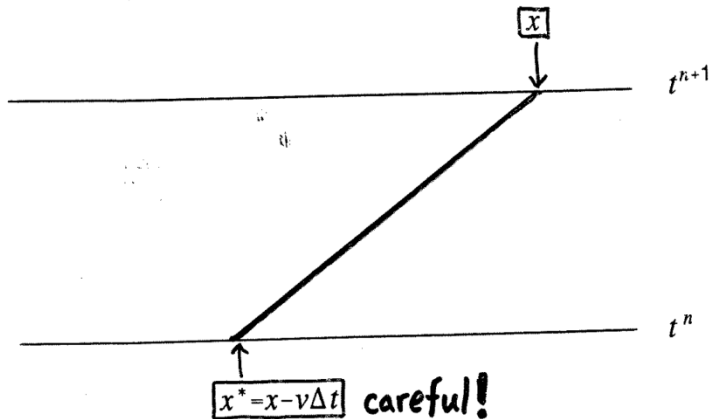
$$\frac{Du}{Dt} = u_t + vu_x \quad \left\{ \begin{array}{l} \text{Douglas, Russell,} \\ \text{Ewing, Wheeler, ...} \end{array} \right.$$

$$\frac{Du}{Dt} - Du_{xx} = 0 \quad \left\{ \begin{array}{l} \text{Pironneau} \\ \text{Morton, Süli, Childs,} \\ \text{Priestley, Sweby, ...} \end{array} \right.$$

Neuman
Baptista

Time-step in characteristic direction

$$\frac{Du}{Dt}(x, t^{n+1}) \approx \frac{u(x, t^{n+1}) - u(x^*, t^n)}{\Delta t}$$



Problem behaves like heat equation

Advantages

- Self-adjoint (better for iterative solvers)
- Parallelizable
- Straightforward in 2D, 3D

Disadvantages

- Non-conservative
- *Ad hoc* treatment of boundary conditions

Results

- Theoretical convergence proofs (optimal)
- Excellent nonoscillatory approximations if at least 3 to 4 intervals across steep fronts
- Large, accurate time steps

Eulerian-Lagrangian localized adjoint method (ELLAM)

LOCALIZED ADJOINT METHODS

Celia, Herrera
Others since 1950's

Adjoint method for $Lu = f$:

$$\int_{\Omega} f w \, dx = \int_{\Omega} Lu w \, dx = \int_{\Omega} u L^* w \, dx + \int_{\partial\Omega} (\text{bdry terms}) \, dS$$

Choose test function w such that $L^* w = 0$

Not practical in general (like global Green's function)

Localize: $L^* w = 0$ on each element, w continuous

Example:

$$Lu = u_t + Vu_x - Du_{xx}$$

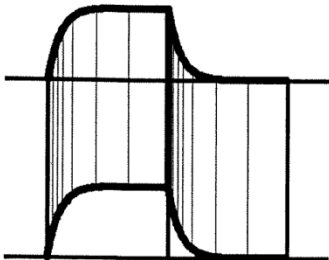
$$L^* w = -w_t - Vw_x - Dw_{xx}$$

Define w so that locally $\begin{cases} w_t = 0 \\ Vw_x + Dw_{xx} = 0 \end{cases}$

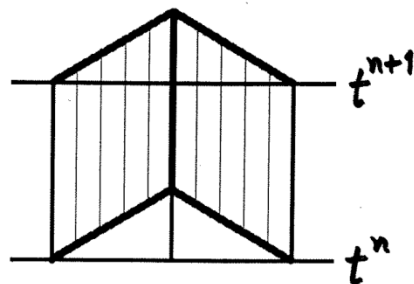
Cases:

$pe \gg 2$ (advection-dominated) | $pe \ll 2$ (dispersion-dominated)

$$w(x) = \frac{1 - e^{-pe(x/\Delta x)}}{1 - e^{-pe}}$$



$$w(x) = \frac{x}{\Delta x}$$

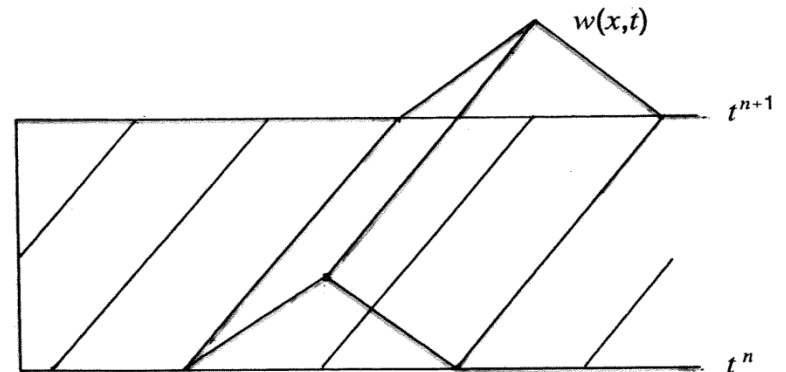


Same disadvantages as Eulerian methods

Celia, Russell, Herrera, Ewing
AWR (1990)

WEAK ELLAM

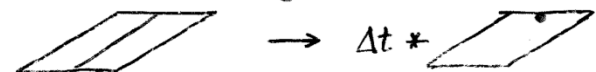
- Space-time finite elements and test functions
- Oriented along characteristics



$$\int_{t^n}^{t^{n+1}} \int_x (u_t + vu_x - Du_{xx}) w \, dx \, dt$$

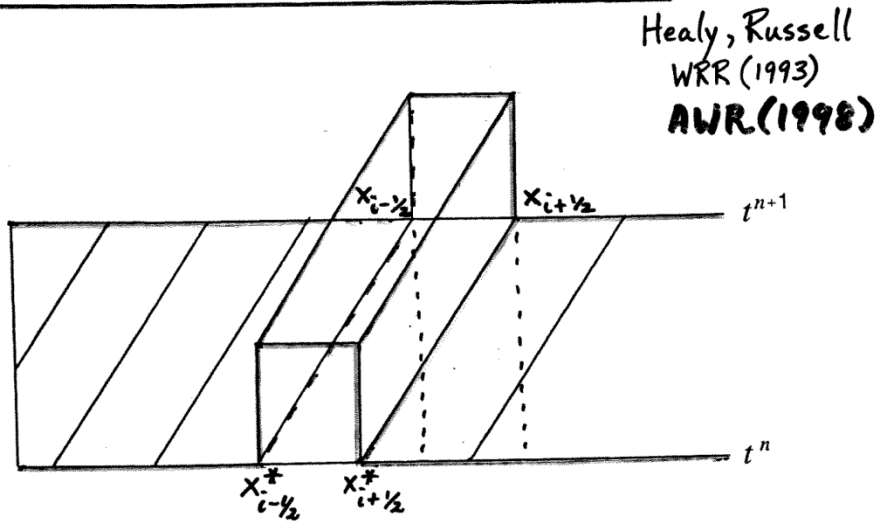
Integrate by parts in space

Backward Euler time integration \leftrightarrow MMOC



Finite volume ELLAM; properties

CONTROL-VOLUME ELLAM / FVE



Healy, Russell
WRR (1993)
AWR (1998)

- Piecewise-constant test function
- Local conservation
- Finite-element representation of solution
- Finite-volume element (FVE) method on Lagrangian elements (FVE well-designed for AFAC) *Baliga, Patankar, McCormick, ...*
- Similar numerical results *Celia (exact RHS integration)*
- Implemented in 3-D solute transport code (MOC3D-ELLAM)

- Extension of MMOC
- Conservative, boundary conditions systematic
- Physical integral terms
- 3D, parallelizability, ...
- Framework for space-time local refinement
- Convergence analysis
- No maximum principle; nonphysical oscillations on too-coarse grids

Immiscible displacement

- Phase = component, $c_{i\alpha} = \delta_{i\alpha}$
- Constant density $\rho_\alpha \equiv 1$, porosity $\Phi \equiv 1$, no reactions / sources ($R_i = 0$)
- Reduces to

$$(S_\alpha)_t + \operatorname{div} V_\alpha = 0, \quad V_\alpha = V f(S_\alpha)$$

- Shows how to advect component α in phase α ; simple generalization to mixture of components i in phase α

Adjoint \leftrightarrow Particle / mass / volume propagation

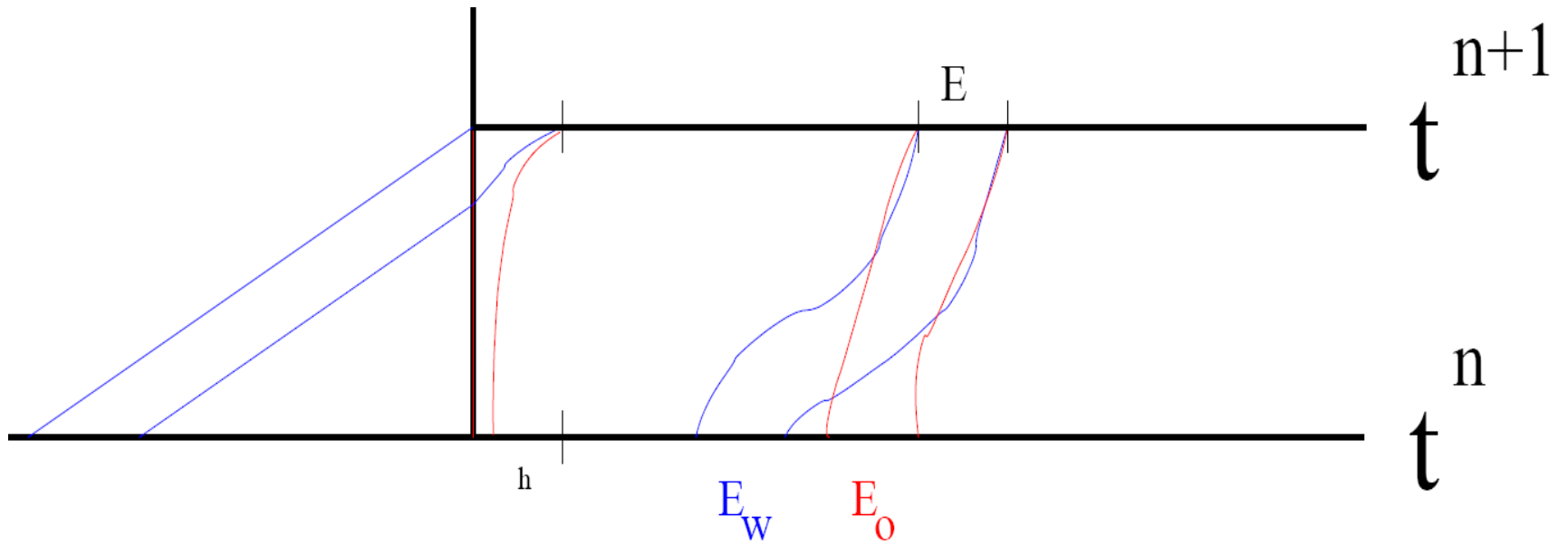
- Space-time test function $w_{\alpha,E}(x,t)$:
 - Characteristic (indicator) function of element E at time t^1
 - Satisfies adjoint condition $A_u^* w_{\alpha,E} = 0 \leftrightarrow$ **constant along particle trajectories**; E traces back to E_α

- Resulting primal equation:

$$\int_E S_\alpha(t^1) + \text{spatial boundary term} = \int_{E_\alpha} S_\alpha(t^0)$$

- Thus, advection algorithm maps E_α at t^0 to E at t^1 , moves volumes through particle-velocity field
- Particle velocity **always** appears: consider case of shrinking E to a point
- **Hard part of advection algorithm**: computing correspondence $E \leftrightarrow E_\alpha$; once done, simply put E_α integral on right-hand side of primal cell E equation

Multiphase tracebacks



Remarks on adjoint \leftrightarrow particle

- Adjoint relation is linear with respect to w ; characteristics don't cross, no shocks form
- Matches physical interpretation of mass propagation – trailing particles don't overtake leading ones, particles don't disappear into a shock; method mimics multiphase flow physics
- Dual space (e.g., ELLAM) is natural framework for particle-propagation scheme; primal methods (e.g., MMOC) see wave velocities, could encounter greater difficulties with shocks, etc.

Primal / dual comparison

NONLINEAR FLUX, SHOCKS, SYSTEMS

Primal direct equation

Solution u : element of primal space

$$Au \equiv u_t + (f(u))_x = 0$$

$$\text{Weak form } \langle Au, w \rangle \equiv \int_t \int_x Au w = 0$$

Wave velocity $c = f'(u)$
propagates solution values

Shocks (waves break)

Nonlinear

Wave methods (Riemann, etc.)

w sat. adjoint eq. $\Rightarrow u$ sat. $\int_x u(t_1)w(t_1) = \int_x u(t_0)w(t_0)$ (advection)

Dual adjoint equation

Test fct w : linear fnal in dual space
e.g., evaluate mass in a subdomain

$$A_u^* w \equiv u w_t + f(u) w_x = 0$$

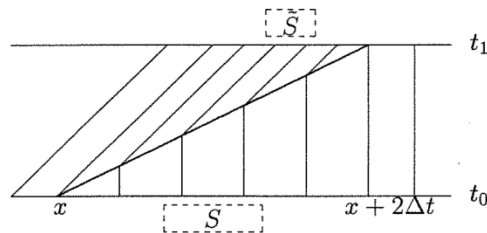
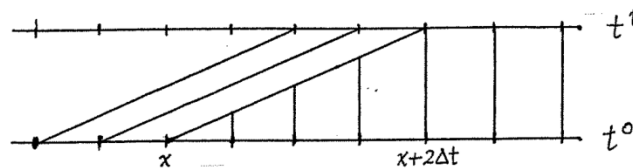
$$\langle A_u^* w, u \rangle = \int_t \int_x A_u^* w u + \text{boundary terms}$$

Particle velocity $v = f(u)/u$
propagates mass

No shocks (particles don't disappear)

Linear

Mass-carrying particle methods



Multiphase multicomponent flow: system of nonlinear PDEs

- Mass particles move in phases
- Test functions w_α , one for each phase
- Conservation eq. for each component, summed over its phases

Multiphase multicomponent system

MULTIPHASE MULTICOMPONENT FLOW

Assume phase Darcy velocities \vec{V}_α available from flow solver (previous time step or iteration)

Conservation of mass equation for component i :

$$\left(\sum_\alpha \phi S_\alpha \rho_\alpha c_{i\alpha} \right)_t + \nabla \cdot \left(\sum_\alpha \rho_\alpha \vec{V}_\alpha c_{i\alpha} \right) = R_i, \quad i = 1, \dots, n_i,$$

where $\alpha = 1, \dots, n_\alpha$.

Weak form: Space-time test function $w_\alpha(\vec{x}, t)$ for each phase. Decompose component i conservation equation into n_α sub-equations.

Weak form *for component i in phase α* is

$$\begin{aligned} & \int_\Omega (\phi S_\alpha \rho_\alpha c_{i\alpha} w_\alpha)^{n+1} + \int_{t^n}^{t^{n+1}} \int_{\partial\Omega} \rho_\alpha \vec{V}_\alpha \cdot \vec{n} c_{i\alpha} w_\alpha \\ & = \int_{t^n}^{t^{n+1}} \int_\Omega R_{i\alpha} w_\alpha + \int_\Omega (\phi S_\alpha \rho_\alpha c_{i\alpha} w_\alpha)^n \end{aligned}$$

subject to the adjoint particle-velocity equation

$$\phi S_\alpha (w_\alpha)_t + \vec{V}_\alpha \cdot \nabla w_\alpha = 0, \quad \alpha = 1, \dots, n_\alpha.$$

Reduction from $n_i n_\alpha$ equations to n_i equations in terms of global concentrations c_i or global mole numbers comes from phase equilibria. Advection is done (explicitly, via Lagrangian tracking) for each component in each phase as above. Assuming that phase equilibria can be defined in terms of global concentrations, re-sum the above weak form over the phases after the advection step.

Remark: $\vec{v}_\alpha = \vec{V}_\alpha / \phi S_\alpha$ is the particle velocity (of all components) in phase α .

Remark: This reduces to familiar special cases under simplifying assumptions; e.g., two-component incompressible oil-water with no mass transfer, single-phase multicomponent.

Remark: In \vec{V}_α , could encounter discontinuities or rapid changes that propagate at wave speed (not particle speed). In a realistic problem, these are not likely to be as severe as those in the 1D examples.

Summary and conclusions (ELM)

- Lagrangian advection more favorable than Eulerian for decoupling during nonlinear iterations, and for solution accuracy
- Adjoint \leftrightarrow mass / volume; primal \leftrightarrow waves
- Adjoint characteristics are more tractable; track mixture of components in each phase; advection similar to source/sink in full system
- Core advection algorithm of tracking masses / volumes in phases is the hard part
- Conceptually, extension to general complex fluid systems appears to be worked out

Generic Newton-like time-stepping algorithm

- Nonlinear residual includes all couplings
- Each iteration decouples subsystems
 - Neglect weak couplings, **or**
 - Approximate couplings with static or linear (i.e., tractable) terms
- **Objective:**
 - Uncover unifying principles that could enable a broad range of systems to be analyzed in this “smart” way
- Decouple (nonlinear), then linearize:
preferable to linearize, then decouple (linear)

Potential payoffs

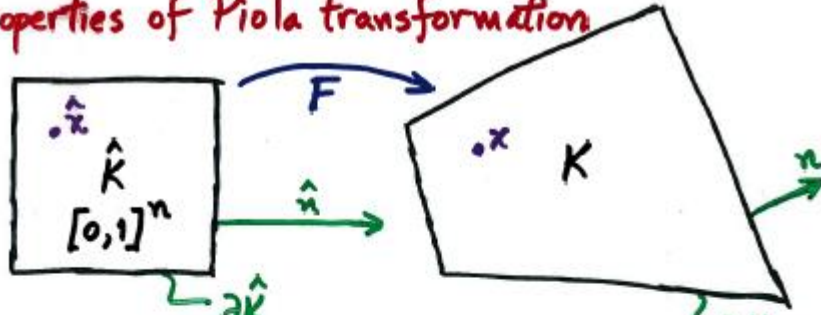
- Physical understanding of decomposition
- Smaller systems, few nonlinear iterations
- Parallelism: divide and conquer (but watch out for multicore architectures)
- Uncertainty analysis:
 - Uncertainty of whole is close to aggregated uncertainties of weakly coupled subsystems
 - Perform expensive uncertainty analysis on subsystems
- Similarly for sensitivity analysis

Summary and conclusions (operator splitting)

- Conceptual framework that potentially applies to a wide variety of coupled systems, operators
- Look for weak or tractable couplings at nonlinear level
- Avoid near-redundant tight couplings
- Real (e.g., phase) transition points are hard
- May all your bottlenecks be physical

Mixed methods, hexahedra, RT_0 , Piola transform, etc.

Properties of Piola transformation



$$\hat{\mathbf{v}}: \hat{K} \rightarrow \mathbb{R}^n$$

$$\mathbf{v}: K \rightarrow \mathbb{R}^n$$

$$\mathbf{v}(x) = \frac{DF(\hat{x})}{J(\hat{x})} \hat{\mathbf{v}}(\hat{x})$$

Preserves normal fluxes across surfaces:

$$\int_{\partial K} \mathbf{v} \cdot \mathbf{n} \, ds = \int_{\partial \hat{K}} \hat{\mathbf{v}} \cdot \hat{\mathbf{n}} \, ds$$

$\hat{\mathbf{v}} = RT_0$ basis function for face $\hat{x} = 1$

$$\Rightarrow \hat{\mathbf{v}}(\hat{x}) = (\hat{x}, 0) \text{ in 2-D, } (\hat{x}, 0, 0) \text{ in 3-D}$$

for \mathbf{v} {
 \Rightarrow flux across $\hat{x} = c$ is c (normal component = c /surface Jacobian, pointwise)
 flux across $\hat{y} = c$ or $\hat{z} = c$ is 0, pointwise

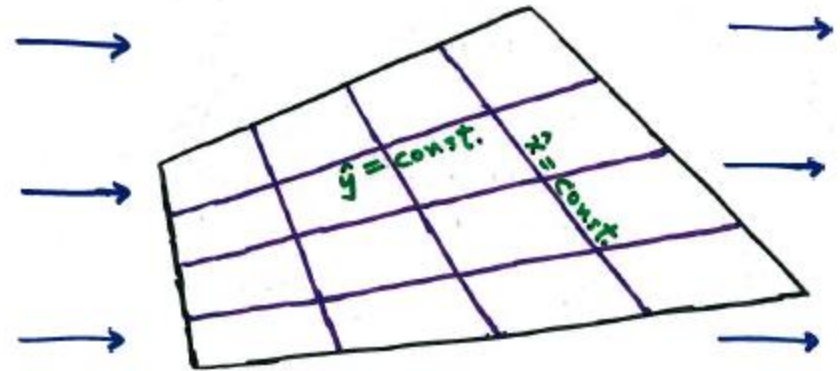
Trial space $Piola(RT_0)$ characterized by:

Normal flux across $\hat{x} = \text{const.}$ (resp. $\hat{y} = \text{const.}, \hat{z} = \text{const.}$) is a linear function of \hat{x} (resp. \hat{y}, \hat{z})

Normal component across \hat{x} or \hat{y} or $\hat{z} = \text{const.}$ is proportional to $1/\text{surface Jacobian}$

In 2-D, constant vector field $\in Piola(RT_0)$

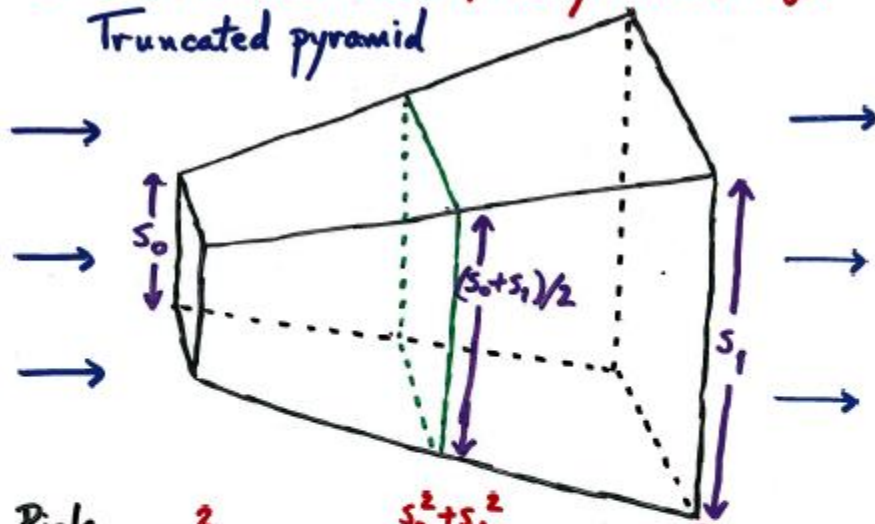
Note that surface (edge) Jacobian is constant



Constant vector field (in any direction) satisfies properties above

Difficulties in 3-D

In 3-D, constant vector field \notin Piola(RT_0)
Truncated pyramid



Piola flux s_0^2

$$\frac{s_0^2 + s_1^2}{2}$$

s_1^2

Physical flux s_0^2

$$\left(\frac{s_0 + s_1}{2}\right)^2$$

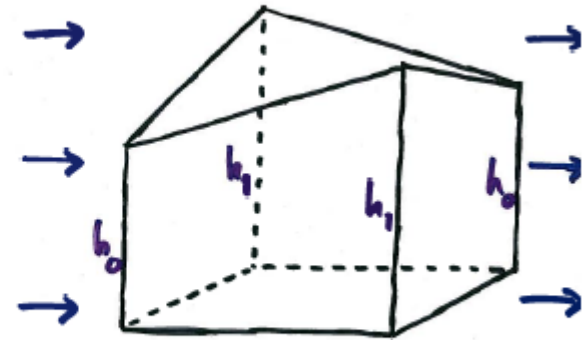
s_1^2

Unlike 2-D, cross-sectional area (length for 2-D) is not a linear function of \hat{x}

A further thought experiment...

Top face is $\hat{z}=1$
(curvilinear)

Net flux across top face = 0
(not pointwise)



Basis vector fields for other 5 faces have normal component 0 on top face, pointwise

Coefficient of basis vector field for top face is 0

\Rightarrow Trial function attempting to represent constant vector field has 0 normal component on top face, pointwise

Conclusion: A trial space with

(1) degrees of freedom \leftrightarrow face fluxes

(2) pointwise preservation of 0 fluxes across surfaces

cannot (in general) contain the constant vector fields.

Control-volume RT_0 mixed method

CONTROL-VOLUME MIXED METHOD ON RECTANGLES

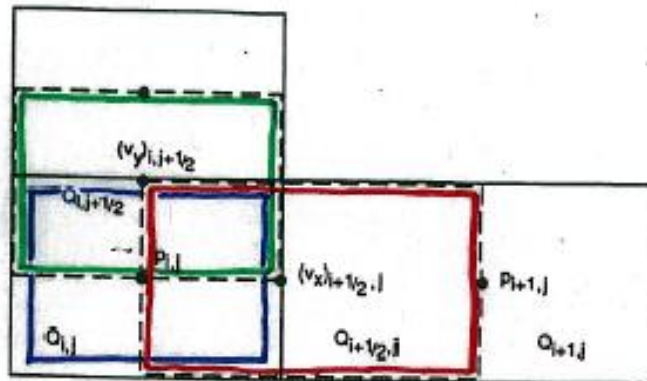
Shape and weighting functions are different

- Shape functions as in standard mixed method
- Velocity weighting functions constant on halves of cells
- Pressure weighting functions as in standard mixed method

$$Q_{i,j} = (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2}),$$

$$Q_{i+1/2,j} = (x_i, x_{i+1}) \times (y_{j-1/2}, y_{j+1/2}),$$

$$Q_{i,j+1/2} = (x_{i-1/2}, x_{i+1/2}) \times (y_j, y_{j+1}),$$



CONTROL-VOLUME MIXED METHOD ON QUADRILATERALS

Reference mapping

$$x(\hat{x}, \hat{y}) = x_{00} + (x_{10} - x_{00})\hat{x} + (x_{01} - x_{00})\hat{y} \\ + (x_{11} - x_{10} - x_{01} + x_{00})\hat{x}\hat{y},$$

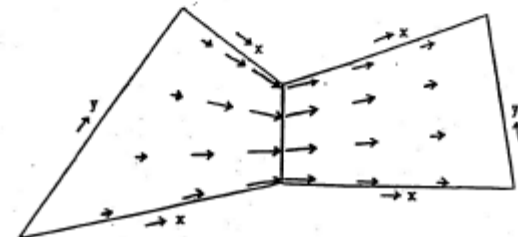
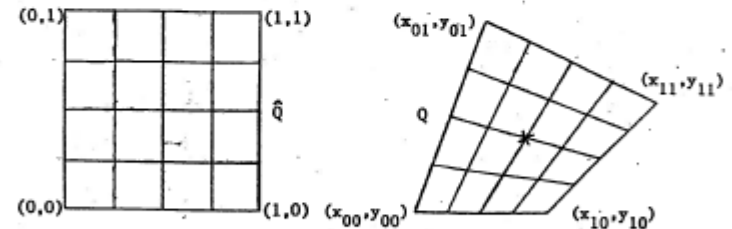
$$y(\hat{x}, \hat{y}) = y_{00} + (y_{10} - y_{00})\hat{x} + (y_{01} - y_{00})\hat{y} \\ + (y_{11} - y_{10} - y_{01} + y_{00})\hat{x}\hat{y}.$$

Inverse mapping exists for convex quadrilaterals

Pressure shape functions: constant on cells $Q_{i,j}$

Velocity shape functions:

- Correspond to unit flux across each edge
- Constant normal component on edge, continuous



Linear and quadratic fluxes

FLUX APPROXIMATIONS

Exact flux

$$F_x(\hat{x}) = \int_0^1 \int_0^1 V(\hat{x}, \hat{y}, \hat{z}) \cdot (Y \times Z)(\hat{x}, \hat{y}, \hat{z}) d\hat{y} d\hat{z}$$

Surface Jacobian

$$Y \times Z = (1-\hat{x}) Y_0 \times Z_0 + \hat{x} Y_1 \times Z_1 - \hat{x}(1-\hat{x}) \underbrace{(Y_1 - Y_0)}_{\delta Y} \times \underbrace{(Z_1 - Z_0)}_{\delta Z}$$

Piola approximate flux (linear interpolation)

$$LF_x(\hat{x}) = (1-\hat{x}) F_x(0) + \hat{x} F_x(1)$$

Quadratic approximate flux

$$QF_x(\hat{x}) = (1-\hat{x}) F_x(0) + \hat{x} F_x(1) - \hat{x}(1-\hat{x}) F_{xi}$$

$$F_{xi} = \int_0^1 \int_0^1 V\left(\frac{1}{2}, \hat{y}, \hat{z}\right) \cdot (\delta Y \times \delta Z) d\hat{y} d\hat{z}$$

Uniform flow (V constant) $\Rightarrow F_x \equiv QF_x$

Not necessarily exact pointwise velocities

FLUX TRUNCATION ERRORS

$$F_x - QF_x = \int_0^{\Delta x} \overbrace{G(x, \xi)}^{O(\Delta x)} \int_0^1 \int_0^1 \left\{ O(1) + \frac{2}{\Delta x} \frac{\partial V}{\partial x} \cdot \underbrace{(Y_1 \times Z_1 - Y_0 \times Z_0)}_{O(\Delta x) \text{ if regular refinement}} \right\} d\hat{y} d\hat{z} d\xi$$

$O(1)$ if random refinement

$$+ \left[\frac{2}{\Delta x^2} \left(V\left(\frac{1}{2}\right) - \underbrace{V\left(\frac{\Delta x}{2}\right)}_{LF_x} \right) + O\left(\frac{1}{\Delta x}\right) \right] \cdot \underbrace{[\delta Y \times \delta Z]}_{O(\Delta x^2) \text{ if regular}} \int_0^1 \int_0^1 d\hat{y} d\hat{z} d\xi$$

$O(1/\Delta x)$ if QF_x
 $O(1/\Delta x^2)$ if LF_x
 $O(\Delta x^2)$ if regular
 $O(1)$ if random

	$F_x - LF_x$	$F_x - QF_x$
Regular refinement	$O(\Delta x^2)$	$O(\Delta x^2)$
Random refinement	$O(1)$	$O(\Delta x)$
Uniform flow	not exact	exact

Quadratic flux and velocity interpolation

DETERMINATION OF QUADRATIC CORRECTION

Algorithm gives primary face fluxes

$$F_x(0), F_x(1), F_y(0), F_y(1), F_z(0), F_z(1)$$

(RT₀ degrees of freedom)

F_{xi} approximation, $O(\Delta x^2(\Delta x^2 + \Delta y^2 + \Delta z^2))$:

$$F_{xi} \approx \frac{1}{8} \left(\sum_{8 \text{ corners}} V(\text{corner}) \right) \cdot \int_0^1 \int_0^1 \delta Y \times \delta Z \, d\hat{y} d\hat{z}$$

($V(\text{corner})$ determined from primary face fluxes)

FROM FLUXES TO VELOCITIES (trial functions)

Normal component magnitude

$$U_x = (1 - \hat{x}) \underbrace{\frac{F_x(0)}{A_x(0)}}_{\text{cross-sectional area}} \frac{|Y_0 \times Z_0|}{|Y \times Z|} + \hat{x} \underbrace{\frac{F_x(1)}{A_x(1)}}_{\text{cross-sectional area}} \frac{|Y_1 \times Z_1|}{|Y \times Z|} - \hat{x}(1 - \hat{x}) \underbrace{\frac{F_{xi}}{A_{xi}}}_{\int_0^1 \int_0^1 |\delta Y \times \delta Z|(\hat{y}, \hat{z}) d\hat{y} d\hat{z}} \frac{|\delta Y \times \delta Z|}{|Y \times Z|}$$

Quadratic flux

$$\int_0^1 \int_0^1 U_x(\hat{x}, \hat{y}, \hat{z}) |Y \times Z| d\hat{y} d\hat{z} = Q F_x(\hat{x})$$

Constant normal component on primary cell faces

Velocity vector field

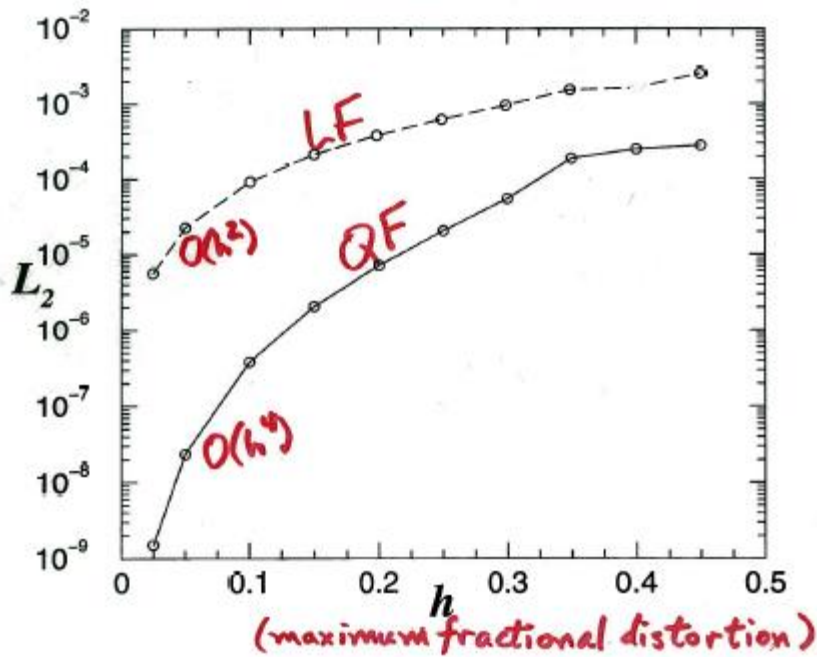
$$V = V_x + V_y + V_z$$

$$V_x = U_x \frac{|Y \times Z|}{J} \chi$$

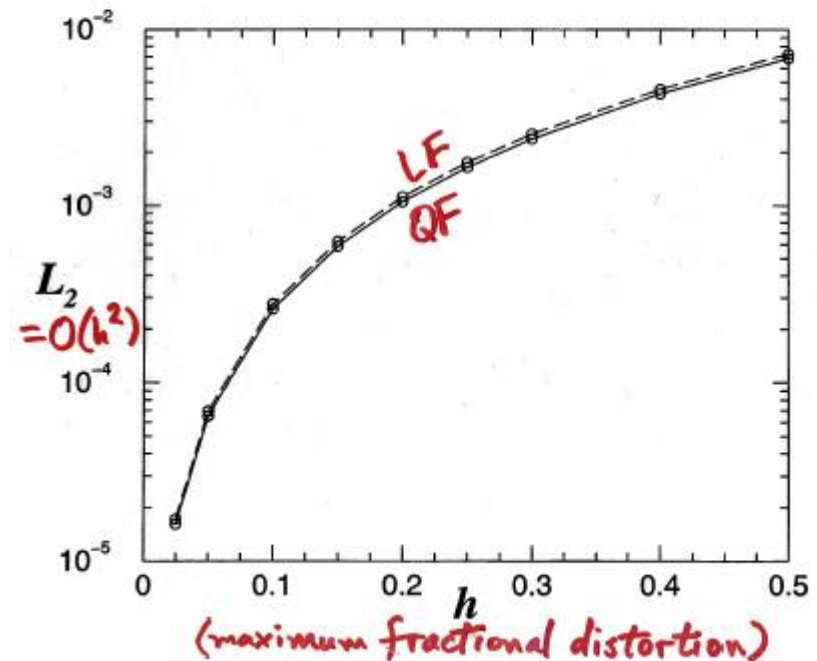
Good for truncated pyramid (planar faces),
not for tent with curved roof (non-planar)

Linear and quadratic flux results

Uniform flow
Planar primary interior faces
All runs $8 \times 8 \times 8$



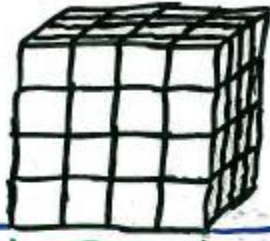
Uniform flow
Random (non-planar) primary interior faces
All runs $8 \times 8 \times 8$



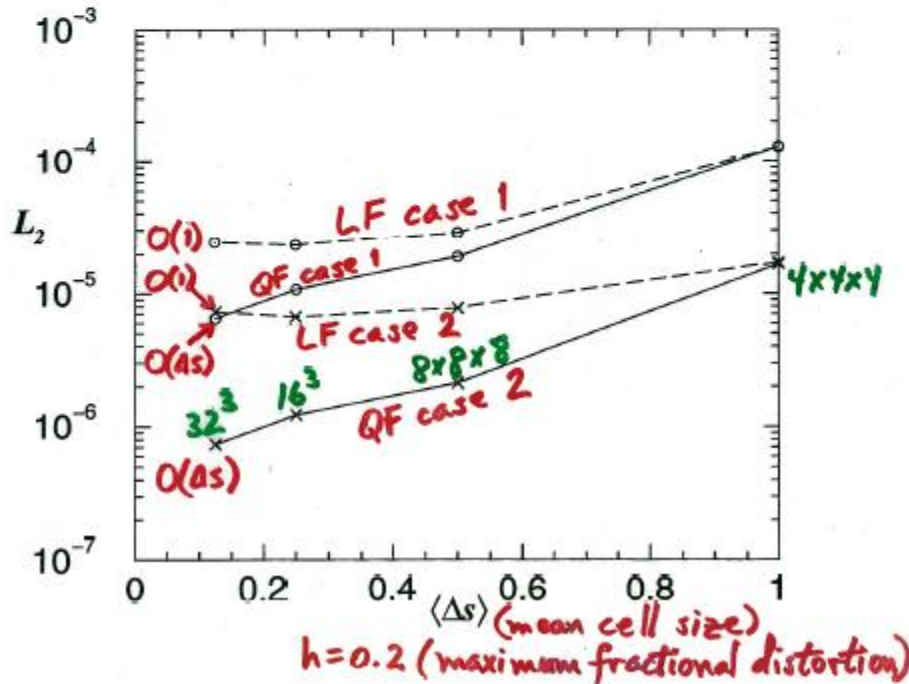
The difficulties aren't just theoretical

Non-uniform flow

Homogeneous
Planar primary
interior faces



Random
refinement



Happy

"Retirement"!