Software Abstractions for Structure Preservation

Matt Knepley

Computer Science and Engineering
University at Buffalo

Geometric Mechanics Formulations for Continuum Mechanics
Banff, CA
Mar 20, 2025

University at Buffalo
Center for Hybrid Rocket
Exascale Simulation Technology

Never believe anything

until you run it.

Outline

Structure Preservation

Finite Elements

Algebraic Solvers

Timesteppers

Lessons

What is a Structure-Preserving Method?

Conservation

Conserved quantities can arise,

from continuous symmetries.

(Neuenschwander 2017)

What is a Structure-Preserving Method?

Montonicity

We can preserve

What is a Structure-Preserving Method?

Montonicity

We can preserve

- solution positivity (Horvath 2004)
- entropy monotonicity (Kraus and Hirvijoki 2017)
- solution monotonicity (Suresh and Huynh 1997)

What is a Structure-Preserving Method?

Manifolds

We can preserve

What is a Structure-Preserving Method?

Manifolds

We can preserve

symplectic manifolds, (Ranocha and Ketcheson 2020)

chiral manifolds, (Brower 1971)

group manifolds. (Munthe-Kaas 1999)

What is a Structure-Preserving Method?

Algebraic Relations

We can preserve

What is a Structure-Preserving Method?

Algebraic Relations

We can preserve

symplecticity, (Skeel and Cieśliński 2020)

algebraic compatibility, (Bonelle 2014)

null spaces. (Schöberl 1999)

What might I want from Software?

Accuracy

What might I want from Software?

Accuracy

Stability

What might I want from Software?

Accuracy

Stability

Efficiency

What might I want from Software?

Accuracy

Stability

Efficiency

Scalability

What might I want from Software?

Understandibility

What might I want from Software?

Understandibility

Concision and Simplicity

What might I want from Software?

Understandibility

Concision and Simplicity

Fewest Concepts (Occam’s Razor)

What might I want from Software?

Understandibility

Concision and Simplicity

Fewest Concepts (Occam’s Razor)

Ex. UNIX File abstraction

What might I want from Software?

Maintainability

What might I want from Software?

Maintainability

Small maintainer group

What might I want from Software?

Maintainability

Small maintainer group

Often depends on workflows (Configure/Build/Test)

What might I want from Software?

Maintainability

Small maintainer group

Often depends on workflows (Configure/Build/Test)

Porting and optimization

What might I want from Software?

Maintainability

Small maintainer group

Often depends on workflows (Configure/Build/Test)

Porting and optimization

Ex. PETSc (Brown, Knepley, and Smith 2015)

What might I want from Software?

Extensibility

What might I want from Software?

Extensibility

Composability (Brown, Knepley, May, et al. 2012)

What might I want from Software?

Extensibility

Composability (Brown, Knepley, May, et al. 2012)

Can existing objects form a new thing?

What might I want from Software?

Extensibility

Composability (Brown, Knepley, May, et al. 2012)

Can existing objects form a new thing?

Increasing functionality, constant maintenance burden

What might I want from Software?

Extensibility

Composability (Brown, Knepley, May, et al. 2012)

Can existing objects form a new thing?

Increasing functionality, constant maintenance burden

Ex. Reynolds-robust PC (Farrell, Mitchell, et al. 2019)

What might I want from Software?

Humility

What might I want from Software?

Humility

Your interface might not be for everyone

What might I want from Software?

Humility

Your interface might not be for everyone

Expose multiple layers

What might I want from Software?

Humility

Your interface might not be for everyone

Expose multiple layers

Ex. BLAS/LAPACK (Anderson et al. 1995)

Outline

Structure Preservation

Finite Elements

- Primal Basis
- Dual Basis
- Finite Elements

Algebraic Solvers

Timesteppers

Lessons

Outline

- Finite Elements
  - Primal Basis
  - Dual Basis
  - Finite Elements

Basis Representation

How should we represent an element?

Basis Representation

How should we represent an element?

Tabulate basis functions

Assemble residual/Jacobian

L_2 projection

Boundary conditions

Basis Representation

How should we represent an element?

Tabulate basis functions

FIAT

Deal.II

DUNE

FreeFEM++

Basis Representation

What could we do with an explicit basis representation?

Basis Representation

What could we do with an explicit basis representation?

Runtime tabulation for particle methods

Outline

Finite Elements

- Primal Basis
- Dual Basis
- Finite Elements

Dual Basis

How could we tabulate the dual basis?

Dual Basis

How could we tabulate the dual basis?

By Riesz-Markov-Kakutani (Wikipedia 2015a),

dual vectors are quadrature rules.

$$\psi_i \rightarrow \{\mathbf{x}_i, \mathbf{w}_i\}$$

Dual Basis

How could we tabulate the dual basis?

We also have a geometric decomposition of the dual space

$$\text{mesh point} \to \{\psi_i\}$$

Dual Basis

Exposing the dual basis allows cheap, custom interpolation.

Dual Basis

Exposing the dual basis allows cheap, custom interpolation.

The geometric decomposition makes interpolation on embedded manifolds easy.

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

(Isaac 2022)

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

a single isomorphism to trace-free subspaces

$$\mathring{\star}_T : \mathcal{P}_r\Lambda^k(T) \to \mathring{\mathcal{P}}^-_{r+k+1}\Lambda^{n-k}(T)$$

$$\mathring{\star}_T : \mathcal{P}^-_r\Lambda^k(T) \to \mathring{\mathcal{P}}_{r+k}\Lambda^{n-k}(T)$$

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

a single isomorphism to trace-free subspaces

$$\mathring{\star}_T : \Lambda^k(\overline{T}) \to \mathring{\Lambda}^{n-k}(\overline{T})$$

that acts pointwise.

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

a single extension operator

$$E_{f,g} : \mathring{\Lambda}^k(\overline{f}) \to \Lambda^k(\overline{g})$$

that acts pointwise.

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

a single extension operator that decomposes

$$\mathcal{P}_r\Lambda^k(\overline{g}) = \bigoplus_{f\in\Delta(g)} E_{f,g}\left(\mathring{\mathcal{P}}_r\Lambda^k(\overline{f})\right)$$

$$\mathcal{P}_r^-\Lambda^k(\overline{g}) = \bigoplus_{f\in\Delta(g)} E_{f,g}\left(\mathring{\mathcal{P}}_r^-\Lambda^k(\overline{f})\right)$$

Dual Basis

Geometric decomposition +

Discrete Hodge Star =

a single extension operator that decomposes

$$\Lambda^k(\overline{g}) = \bigoplus_{f \in \Delta(g)} E_{f,g}\left(\mathring{\Lambda}^k(\overline{f})\right)$$

Outline

Finite Elements

- Primal Basis
- Dual Basis
- Finite Elements

FE Representation

Instead of names,

we can refer to elements by structure

FE Representation

Instead of names,

we can refer to elements by structure

Primal Space

polynomial

trimmed polynomial

direct product

direct sum

FE Representation

Instead of names,

we can refer to elements by structure

Dual Space

- Lagrange
- direct sum

Raviart-Thomas on a Simplex

```
-petscspace_degree <k>
-petscspace_type ptrimmed
```

Raviart-Thomas on a Simplex

```
-petscspace_degree <k>
-petscspace_type ptrimmed

-petscdualspace_form_degree -1
-petscdualspace_order <k>
-petscdualspace_lagrange_trimmed true
```

Raviart-Thomas on a Quadrilateral

```
-petscspace_type sum
-petscspace_variables 2
-petscspace_components 2
-petscspace_sum_spaces 2
-petscspace_sum_concatenate true
```

Raviart-Thomas on a Quadrilateral

```
-petscspace_type sum
-petscspace_variables 2
-petscspace_components 2
-petscspace_sum_spaces 2
-petscspace_sum_concatenate true

-sumcomp_0_petscspace_variables 2
-sumcomp_0_petscspace_type tensor
-sumcomp_0_petscspace_tensor_spaces 2
-sumcomp_0_petscspace_tensor_uniform false
-sumcomp_0_tensorcomp_0_petscspace_degree <k>
-sumcomp_0_tensorcomp_1_petscspace_degree <k-1>
```

Raviart-Thomas on a Quadrilateral

```
-sumcomp_1_petscspace_variables 2
-sumcomp_1_petscspace_type tensor
-sumcomp_1_petscspace_tensor_spaces 2
-sumcomp_1_petscspace_tensor_uniform false
-sumcomp_1_tensorcomp_0_petscspace_degree <k-1>
-sumcomp_1_tensorcomp_1_petscspace_degree <k>
```

Raviart-Thomas on a Quadrilateral

```
-sumcomp_1_petscspace_variables 2
-sumcomp_1_petscspace_type tensor
-sumcomp_1_petscspace_tensor_spaces 2
-sumcomp_1_petscspace_tensor_uniform false
-sumcomp_1_tensorcomp_0_petscspace_degree <k-1>
-sumcomp_1_tensorcomp_1_petscspace_degree <k>

-petscdualspace_form_degree -1
-petscdualspace_order <k>
-petscdualspace_lagrange_trimmed true
```

Brezzi-Douglas-Marini on a Simplex

```
-petscspace_degree <k>
```

Brezzi-Douglas-Marini on a Simplex

```
-petscspace_degree <k>

-petscdualspace_form_degree -1
-petscdualspace_order <k>
-petscdualspace_lagrange_trimmed false
```

First Kind Nedelec on a Simplex

```
-petscspace_degree <k>
-petscspace_type ptrimmed
```

First Kind Nedelec on a Simplex

```
-petscspace_degree <k>
-petscspace_type ptrimmed

-petscdualspace_form_degree 1
-petscdualspace_order <k>
-petscdualspace_lagrange_trimmed true
```

Second Kind Nedelec on a Simplex

```
-petscspace_degree <k>
```

Second Kind Nedelec on a Simplex

```
-petscspace_degree <k>

-petscdualspace_form_degree 1
-petscdualspace_order <k>
-petscdualspace_lagrange_trimmed false
```

Exterior Calculus

Could we build an explicit basis for $\mathrm{Alt}^k V$?

src/dm/dt/interface/dtaltv.c

Exterior Calculus

We can check that the differential commutes with discretization:

$$\mathrm{d}\Pi(\omega) = \Pi(\mathrm{d}\omega)$$

src/dm/dt/tests/ex14.c

Exterior Calculus

We can check that the differential commutes with discretization:

$$\mathrm{d}\Pi(\omega) = \Pi(\mathrm{d}\omega)$$

src/dm/dt/tests/ex14.c

Produce a constructive proof in Lean?

Outline

Structure Preservation

Finite Elements

Algebraic Solvers

Timesteppers

Lessons

Parameter-Robust Smoothers

Smoothers for

$$L + \alpha K$$

can suffer as $\alpha \to \infty$ if

$$\mathcal{N}(K) \neq \emptyset.$$

Parameter-Robust Smoothers

Smoothers for

$$-\nabla \cdot 2\nu\epsilon(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} - \alpha\nabla(\nabla \cdot \mathbf{u})$$

can suffer as $\alpha \to \infty$ if

$$\mathcal{N}(\nabla(\nabla \cdot \mathbf{u})) \neq \emptyset.$$

Parameter-Robust Smoothers

The Schur complement is almost

$$S^{-1} \approx -(\nu + \alpha) M_p^{-1}$$

but the velocity smoother is hard.

Parameter-Robust Smoothers

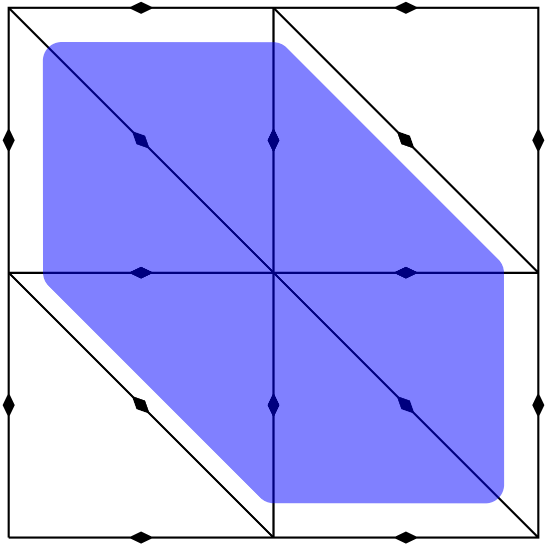
Patch smoothers satisfying

$$\mathcal{N}(K) = \sum_{i} V_i \bigcap \mathcal{N}(K)$$

are robust.

(Schöberl 1999)

Parameter-Robust Smoothers



Parameter-Robust Smoothers

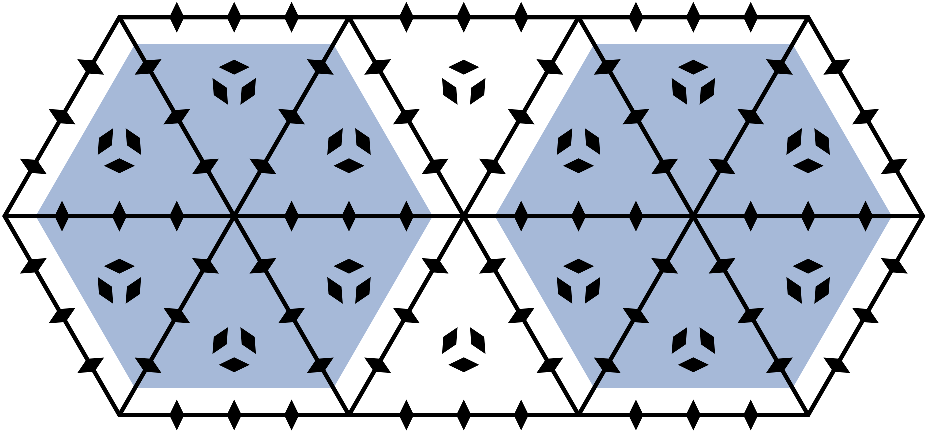
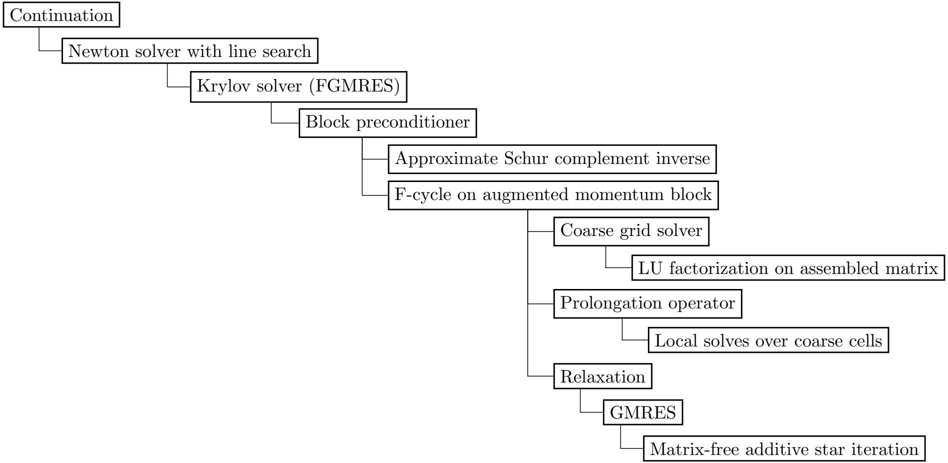


Fig. 3.1. *Star patch for $\mathbb{BDM}_2$-elements.*

Parameter-Robust Smoothers

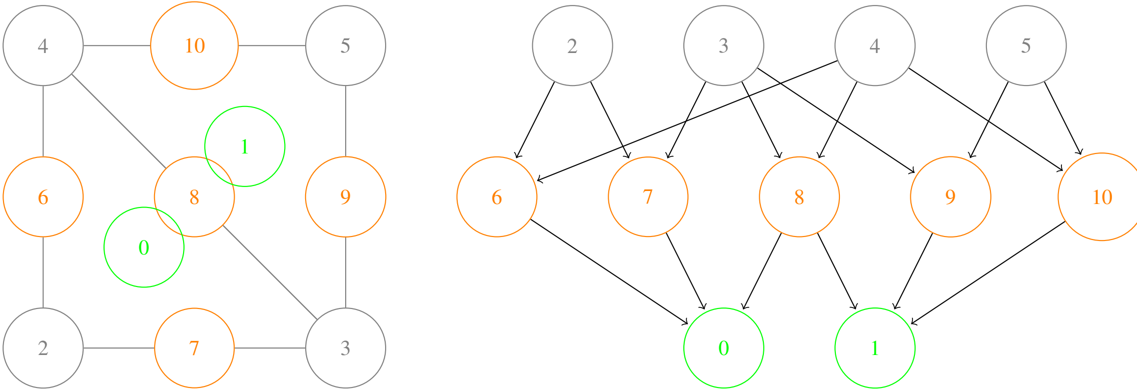
Incompressible Navier-Stokes



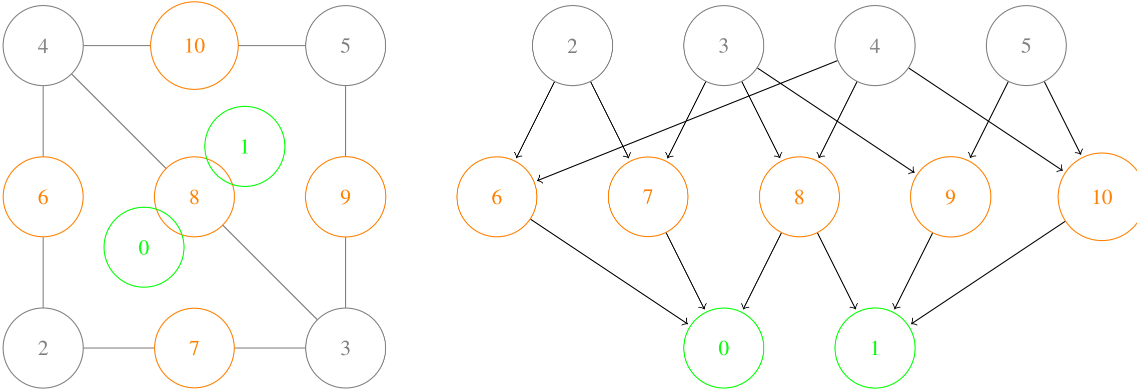
- Continuation
  - Newton solver with line search
    - Krylov solver (FGMRES)
      - Block preconditioner
        - Approximate Schur complement inverse
        - F-cycle on augmented momentum block
          - Coarse grid solver
            - LU factorization on assembled matrix
          - Prolongation operator
            - Local solves over coarse cells
          - Relaxation
            - GMRES
              - Matrix-free additive star iteration

(Farrell, Mitchell, et al. 2019)

Hasse Diagram (Wikipedia 2015b)



DMPlex (Lange et al. 2016)



Parameter-Robust Smoothers

Solver for the $\mathcal{H}(\text{div})$ Riesz map

```
-ksp_type cg
-pc_type mg
-mg_levels_ksp_type richardson
-mg_levels_ksp_richardson_scale 0.333333
-mg_levels_pc_type patch
-mg_levels_patch_pc_patch_local_type additive
-mg_levels_patch_pc_patch_construct_type star
-mg_levels_patch_pc_patch_construct_dim 0
```

(Farrell, Knepley, et al. 2021)

Parameter-Robust Smoothers

Many papers followed

(Adler, Benson, et al. 2021)

(Adler, He, et al. 2022)

(Laakmann, Farrell, et al. 2022)

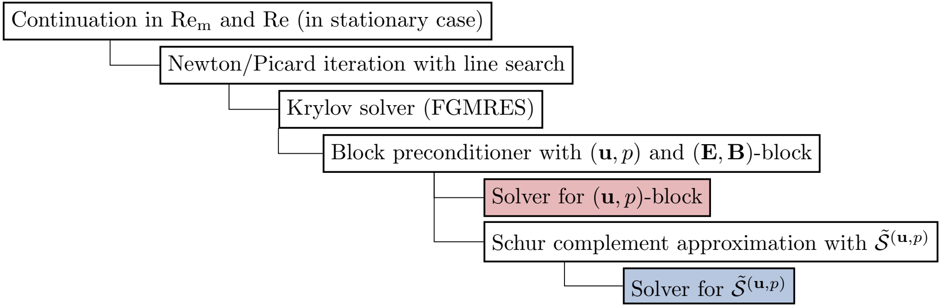
(Abu-Labdeh et al. 2023)

(Laakmann, Hu, et al. 2023)

on different problems.

Composable Solvers

Incompressible Viscoresistive MHD



Continuation in Re_m and Re (in stationary case)

Newton/Picard iteration with line search

Krylov solver (FGMRES)

Block preconditioner with $(\mathbf{u}, p)$ and $(\mathbf{E}, \mathbf{B})$-block

Solver for $(\mathbf{u}, p)$-block

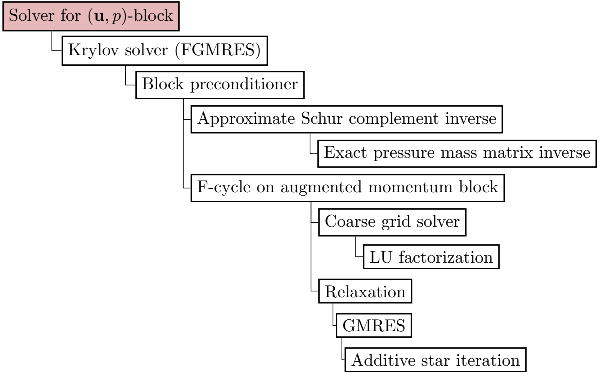
Schur complement approximation with $\tilde{\mathcal{S}}^{(\mathbf{u},p)}$

Solver for $\tilde{\mathcal{S}}^{(\mathbf{u},p)}$

(Laakmann, Farrell, et al. 2022)

Composable Solvers

Incompressible Viscoresistive MHD

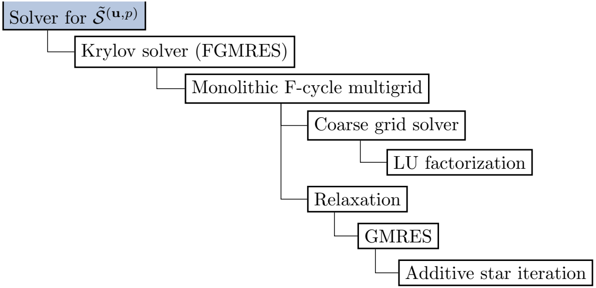


- Solver for $(\mathbf{u}, p)$-block
  - Krylov solver (FGMRES)
    - Block preconditioner
      - Approximate Schur complement inverse
        - Exact pressure mass matrix inverse
      - F-cycle on augmented momentum block
        - Coarse grid solver
          - LU factorization
        - Relaxation
          - GMRES
            - Additive star iteration

(Laakmann, Farrell, et al. 2022)

Composable Solvers

Incompressible Viscoresistive MHD



Solver for $\tilde{\mathcal{S}}^{(\mathbf{u},p)}$

Krylov solver (FGMRES)

Monolithic F-cycle multigrid

Coarse grid solver

LU factorization

Relaxation

GMRES

Additive star iteration

(Laakmann, Farrell, et al. 2022)

Outline

Structure Preservation

Finite Elements

Algebraic Solvers

Timesteppers

Lessons

High-Level Interface

Separate implicit and explicit parts

$$F(u, \dot{u}, x, t) = G(u, x, t)$$

High-Level Interface

Separate implicit and explicit parts

$$J_\alpha = \alpha \boldsymbol{F}_{\dot{u}} + \boldsymbol{F}_u$$

Im/Ex Split

Explicit Methods:

Define only G,

F is assumed to be $\dot{u}$

Im/Ex Split

Implicit Methods:

Define only F,

G is empty

IMEX Methods:

Define both F and G,

but splitting is fixed

Im/Ex Split

- (Implicit) Runge-Kutta
  - Diagonally implicit Runge-Kutta
  - ARKIMEX
  - Strong Stability Preserving (Ketcheson 2008)
  - Relaxation Runge-Kutta (Ketcheson 2019)
- θ method
- Backward Differentiation Formula
- General Linear (Butcher et al. 2007)
- α method (Jansen et al. 2000)
- Extrapolated IMEX (Constantinescu and Sandu 2010)
- Rosenbrock-W (Shampine 1982)

Implicit Runge-Kutta (IRK) is symplectic (Benettin and Giorgilli 1994)

Hamiltonian Systems

Implicit Runge-Kutta (IRK) is

symplectic (Benettin and Giorgilli 1994)

could also use DIRK

Hamiltonian Systems

Implicit Runge-Kutta (IRK) is

- symplectic (Benettin and Giorgilli 1994)
- could also use DIRK

Relaxation Runge-Kutta (RRK) is

- conservative / monotonic

Hamiltonian Systems

Implicit Runge-Kutta (IRK) is

symplectic (Benettin and Giorgilli 1994)

could also use DIRK

Relaxation Runge-Kutta (RRK) is

conservative / monotonic

Needs a projection $\mathcal{P}$

Hamiltonian Systems

Explicit methods,

Hamiltonian Systems

Explicit methods,

basic symplectic (Hairer et al. 2002)

Hamiltonian Systems

Explicit methods,

basic symplectic (Hairer et al. 2002)

Boris (volume-preserving) (Qin et al. 2013)

Hamiltonian Systems

Explicit methods,

- basic symplectic (Hairer et al. 2002)
- Boris (volume-preserving) (Qin et al. 2013)

need *field splitting*!

High-Level Interface

Separate implicit and explicit parts,

and fields,

$$F_i(u, \dot{u}, x, t) = G_i(u, x, t)$$

High-Level Interface

Separate implicit and explicit parts,

and fields,

$$J_{i,\alpha} = \alpha \boldsymbol{F}_{i,\dot{u}} + \boldsymbol{F}_{i,u}$$

Discrete Gradients

$$u_t = \mathcal{S}(u)\nabla F(u)$$

Discrete Gradients

$$u_{n+1} - u_n = \Delta t \, \overline{\mathcal{S}}(u_n, u_{n+1}) \overline{\nabla F}(u_n, u_{n+1})$$

Discrete Gradients

$$(u_{n+1} - u_n) \cdot \overline{\nabla F}(u_n, u_{n+1}) = F(u_{n+1}) - F(u_n)$$

$$\overline{\nabla F}(u_n, u_n) = \nabla F(u_n)$$

Discrete Gradients

$$\overline{\nabla F}(u_n, u_{n+1}) = \nabla F(u_{n+1/2}) + (u_{n+1} - u_n)\cdot \frac{F(u_{n+1}) - F(u_n) - (u_{n+1} - u_n) \cdot \nabla F(u_{n+1/2})}{||u_{n+1} - u_n||^2}$$

(Gonzalez 1996)

Discrete Gradients

$$\overline{\nabla F}(u_n, u_{n+1}) = \int_0^1 d\xi \, \nabla F \left((1-\xi)u_n + \xi u_{n+1}\right)$$

(Harten et al. 1983)
(Finn et al. 2025)

Discrete Gradients

$$\begin{aligned} S(u_{n+1}) - S(u_n) &= (E(u_{n+1}) - F(u_{n+1})) - (E(u_n) - F(u_n)) \\ &= -\,(F(u_{n+1}) - F(u_n)) \\ &= -\,\Big((u_{n+1} - u_n) \cdot \overline{\nabla F}(u_n, u_{n+1})\Big) \\ &= -\Delta t\, \overline{\nabla F}(u_n, u_{n+1}) \overline{\mathcal{S}}^T \overline{\nabla F}(u_n, u_{n+1}) \end{aligned}$$

For metric systems, $\mathcal{S}$ is symmetric negative definite.

(Kraus and Hirvijoki 2017)

(Öttinger 2018)

Discrete Gradients

$$u_t = \mathcal{S}(u) \nabla F(u)$$

Interface Extensions

Field split (BSI)

Domain split (PRK)

Term split (IMEX)

Outline

Structure Preservation

Finite Elements

Algebraic Solvers

Timesteppers

Lessons

Software Lessons

Expose better abstractions

Software Lessons

Expose better abstractions

at runtime

Software Lessons

Expose better abstractions

at runtime

that compose together.

Software Lessons

Expose better abstractions

at runtime

that compose together.

(Brown, Knepley, and Smith 2015)

Build in Layers

Software Lessons

Build in Layers

to allow targeted APIs

Software Lessons

Build in Layers
to allow targeted APIs
that preserve understandability.

Software Lessons

Build in Layers
to allow targeted APIs
that preserve understandability.

(Smith and Gropp 1996)

References I

Neuenschwander, Dwight E. (2017). Emmy Noether's wonderful theorem. JHU Press.

Horvath, Zoltan (2004). "On the positivity of matrix-vector products". In: Linear algebra and its applications 393, pp. 253–258.

Kraus, Michael and Eero Hirvijoki (2017). "Metriplectic integrators for the Landau collision operator". In: Physics of Plasmas 24.10.

Suresh, A. and H.T. Huynh (1997). "Accurate Monotonicity-Preserving Schemes with Runge–Kutta Time Stepping". In: Journal of Computational Physics 136.1, pp. 83–99. ISSN: 0021-9991. DOI: 10.1006/jcph.1997.5745.

Ranocha, Hendrik and David I Ketcheson (2020). "Relaxation Runge–Kutta methods for Hamiltonian problems". In: Journal of Scientific Computing 84.1, p. 17.

Brower, Richard C (1971). "A chiral invariant dual model". In: Physics Letters B 34.2, pp. 143–146.

Munthe-Kaas, Hans (1999). "High order Runge–Kutta methods on manifolds". In: Applied Numerical Mathematics 29.1, pp. 115–127.

Skeel, Robert D. and Jan L. Cieśliński (2020). On the famous unpublished preprint "Methods of integration which preserve the contact transformation property of the Hamilton equations" by René De Vogelaere. arXiv: 2003.12268 [math.NA]. URL: https://arxiv.org/abs/2003.12268.

Bonelle, Jérôme (2014). "Compatible Discrete Operator schemes on polyhedral meshes for elliptic and Stokes equations". PhD thesis. Université Paris-Est.

Schöberl, Joachim (1999). "Multigrid methods for a parameter dependent problem in primal variables". In: Numerische Mathematik 84.1, pp. 97–119.

Brown, Jed, Matthew G. Knepley, and Barry Smith (Jan. 2015). "Run-time extensibility and librarization of simulation software". In: IEEE Computing in Science and Engineering 17.1, pp. 38–45. DOI: 10.1109/MCSE.2014.95.

References II

Brown, Jed, Matthew Knepley, David A. May, Lois C. McInnes, and Barry F. Smith (2012). “Composable linear solvers for multiphysics”. In: Proceeedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012). IEEE Computer Society, pp. 55–62. DOI: 10.1109/ISPDC.2012.16.

Farrell, Patrick E, Lawrence Mitchell, and Florian Wechsung (2019). “An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier-Stokes equations at high Reynolds number”. In: SIAM Journal on Scientific Computing 41.5, A3073–A3096. eprint: 1810.03315.

Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Cros, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen (1995). LAPACK User’s Guide. Second. Philadelphia, Pennsylvania: SIAM.

Wikipedia (2015a). Riesz-Markov-Kakutani Representation Theorem. http://en.wikipedia.org/wiki/Riesz-Markov-Kakutani_representation_theorem. URL: http://en.wikipedia.org/wiki/Riesz-Markov-Kakutani_representation_theorem.

Isaac, Toby (2022). Unifying the geometric decompositions of full and trimmed polynomial spaces in finite element exterior calculus. arXiv: 2112.02174 [math.NA]. URL: https://arxiv.org/abs/2112.02174.

Wikipedia (2015b). Hasse Diagram. http://en.wikipedia.org/wiki/Hasse_diagram. URL: http://en.wikipedia.org/wiki/Hasse_diagram.

Lange, Michael, Lawrence Mitchell, Matthew G. Knepley, and Gerard J. Gorman (2016). “Efficient mesh management in Firedrake using PETSc-DMPlex”. In: SIAM Journal on Scientific Computing 38.5, S143–S155. DOI: 10.1137/15M1026092. eprint: http://arxiv.org/abs/1506.07749.

Farrell, Patrick E, Matthew G Knepley, Lawrence Mitchell, and Florian Wechsung (2021). “PCPATCH: software for the topological construction of multigrid relaxation methods”. In: ACM Transaction on Mathematical Software 47.3, pp. 1–22. ISSN: 0098-3500. DOI: 10.1145/3445791. eprint: http://arxiv.org/abs/1912.08516.

Adler, James H, Thomas R Benson, Eric C Cyr, Patrick E Farrell, Scott P MacLachlan, and Ray S Tuminaro (2021). “Monolithic Multigrid Methods for Magnetohydrodynamics”. In: SIAM Journal on Scientific Computing 0, S70–S91.

References III

Adler, James H, Yunhui He, Xiaozhe Hu, Scott MacLachlan, and Peter Ohm (2022). “Monolithic multigrid for a reduced-quadrature discretization of poroelasticity”. In: SIAM Journal on Scientific Computing 45.3, S54–S81.

Laakmann, Fabian, Patrick E Farrell, and Lawrence Mitchell (2022). “An augmented Lagrangian preconditioner for the magnetohydrodynamics equations at high Reynolds and coupling numbers”. In: SIAM Journal on Scientific Computing 44.4, B1018–B1044.

Abu-Labdeh, Razan, Scott MacLachlan, and Patrick E Farrell (2023). “Monolithic multigrid for implicit Runge–Kutta discretizations of incompressible fluid flow”. In: Journal of Computational Physics 478, p. 111961.

Laakmann, Fabian, Kaibo Hu, and Patrick E Farrell (2023). “Structure-preserving and helicity-conserving finite element approximations and preconditioning for the Hall MHD equations”. In: Journal of Computational Physics 492, p. 112410.

Ketcheson, David I (2008). “Highly Efficient Strong Stability Preserving Runge-Kutta Methods with Low-Storage Implementations”. In: SIAM Journal on Scientific Computing 30, pp. 2113–2136.

— (2019). “Relaxation Runge–Kutta methods: Conservation and stability for inner-product norms”. In: SIAM Journal on Numerical Analysis 57.6, pp. 2850–2870.

Butcher, J.C., Z. Jackiewicz, and W.M. Wright (2007). “Error propagation of general linear methods for ordinary differential equations”. In: Journal of Complexity 23.4-6, pp. 560–580. ISSN: 0885-064X. DOI: 10.1016/j.jco.2007.01.009.

Jansen, Kenneth E, Christian H Whiting, and Gregory M Hulbert (2000). “A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method”. In: Computer methods in applied mechanics and engineering 190.3-4, pp. 305–319.

Constantinescu, E.M. and A. Sandu (2010). “Extrapolated implicit-explicit time stepping”. In: SIAM Journal on Scientific Computing 31.6, pp. 4452–4477. DOI: 10.1137/080732833.

Shampine, Lawrence F (1982). “Implementation of Rosenbrock methods”. In: ACM Transactions on Mathematical Software (TOMS) 8.2, pp. 93–113.

References IV

Benettin, Giancarlo and Antonio Giorgilli (1994). “On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms”. In: Journal of Statistical Physics 74, pp. 1117–1143.

Hairer, E., Ch. Lubich, and G. Wanner (2002). Geometric Numerical Integration. Berlin Heidelberg: Springer-Verlag.

Qin, Hong, Shuangxi Zhang, Jianyuan Xiao, Jian Liu, Yajuan Sun, and William M Tang (2013). “Why is Boris algorithm so good?” In: Physics of Plasmas 20.8.

Gonzalez, Oscar (1996). “Time integration and discrete Hamiltonian systems”. In: Journal of Nonlinear Science 6, pp. 449–467.

Harten, Amiram, Peter D Lax, and Bram van Leer (1983). “On upstream differencing and Godunov-type schemes for hyperbolic conservation laws”. In: SIAM review 25.1, pp. 35–61.

Finn, Daniel S., Joseph V. Pusztay, Matthew G. Knepley, and Mark F. Adams (2025). “Entropy Monotonicity using Discrete Gradients in the Vlasov-Poisson-Landau System”. In: Journal of Computational Physics. Submitted.

Öttinger, Hans Christian (Apr. 2018). “GENERIC Integrators: Structure Preserving Time Integration for Thermodynamic Systems”. In: Journal of Non-Equilibrium Thermodynamics 43 (2), pp. 89–100. ISSN: 14374358. DOI: 10.1515/jnet-2017-0034.

Smith, Barry F. and William D. Gropp (1996). “The Design of Data-structure-neutral Libraries for the Iterative Solution of Sparse Linear Systems”. In: Scientific Programming 5, pp. 329–336.