Computational Multiphysics in a Categorical Framework

Luke L. Morris — luke.morris@ufl.edu

George R. Rauta — grauta@ufl.edu

James P. Fairbanks — fairbanksj@ufl.edu

University of Florida, Gainesville, Florida, USA

Decapodes Decapodes.jl is a framework for encoding multiphysics equations, managing the composition of complex multiphysics systems, and automatically generating performant simulation code. A Decapode diagram is a combinatorial data structure in which nodes define physical quantities, and directed edges define the computational relationship between these quantities. A prior talk [4] focused on the theoretical aspects of encoding models. Here, we present the computational aspects from our Julia implementation: Decapodes.jl. Our goals are achieved by employing attributed C-Sets (ACSets) from Catlab.jl, specifying operadic composition patterns via RelationalDiagrams from Catlab.jl, and using differential operators from the Discrete Exterior Calculus (DEC). This talk builds off work in a manuscript under review in the *Journal of Computational Sciences*.

Motivation In 2013, a diverse group of 45 researchers assessed the state of simulation tools for coupled physical processes [5]. Scientists interested in implementing novel multiphysics simulations are faced with a dilemma of either implementing novel multiphysics solvers tailored to their problem or adapting and extending an existing multiphysics solver. The first of these options is, in general, expensive and time-consuming, and there is a possibility of bugs being introduced due to the communication gap between scientist and programmer. The second option, when possible and affordable, can still be difficult to integrate into established workflows. Decapodes.jl is an attempt at answering both legs of this dilemma by automating some roles of the programmer, by exploiting explicit rather than implicit model representation in code.

Discrete Exterior Calculus We provide here a short motivation for the DEC. Issues arise in numerical partial differential equations (PDEs) when discretizing continuous mathematics with informal knowledge about compatible discretizations. In the exterior calculus, the ith exterior derivative d_i takes i-forms to $i+1$-forms. This discrete differential operator satisfies $d_i d_{i+1} = 0$, preserving for example that the divergence of a gradient is 0. However, due to discreteness, the discrete Hodge star operator, $\star$ is not an isomorphism between primal and dual forms. The DEC makes this process rigorous via the de Rahm complex. A categorical presentation of equations over the de Rahm complex is given in prior work [8].

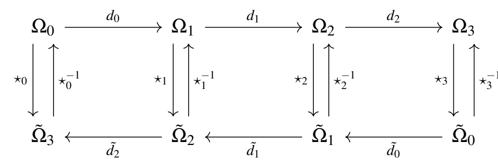
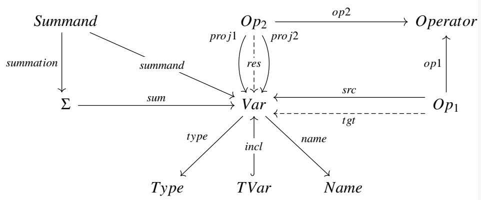


Figure 1: The de Rahm complex in 3D.

The Decapode ACSet A Decapode diagram is a combinatorial data structure in which nodes define physical quantities, and directed edges define the computational relationship between these quantities. Using the foundations of categorical databases [10] and the technique of representation of mathematical objects as instances of categorical databases [9], we encode a Decapode as an "ACSet". An ACSet is a functor from a schema category C to Set, with attributes. In essence, a Decapode diagram is an in-memory database, whose schema is that shown in Figure 2. An example instance of such a database is shown tabularly in Figure 2, corresponding to the diagram shown in Figure 2. This approach builds off of the HypΣ approach to typed hypergraphs introduced by Bonchi et al. [2].

Fluid Dynamics To demonstrate the expressiveness of Decapodes.jl, we offer an implementation of the DEC formulation of the incompressible Navier-Stokes equations as given by Mohamed et al. [7]. A diagram of this formulation is included in Figure 3. Code which can be parsed to encode this Decapode is shown in Figure 3.

Composition & Compilation of Physics We observe that large multiphysics systems are typically more complex in terms of dependencies between computations of physical quantities. Without a formal description of composition, extending a system is time consuming and error-prone. To handle the complexity introduced by extending multiphysics systems, we employ composition patterns



Var	type	name
1	Form0	C
2	Form1	V
3	Form1	ϕ

Op2	proj1	proj2	res	op2
1	1	2	3	∧₀₁

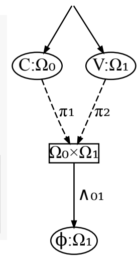
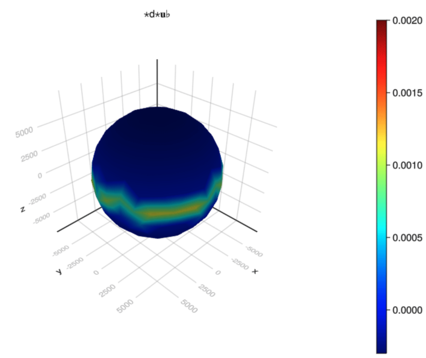


Figure 2: The schema for representing Decapodes. An instance is a relational database, and can be visualized as a string diagram. (Left-to-right)

```
Equation10ForN2 = @decapode begin
  U::Form1
  (P, pd)::Form0
  μ::Constant

  pd == P + 0.5 * i(u,u)

  dt(u) == μ * ∘(d, ⋆, d, ⋆)(u) -
    ⋆(∧(u, ⋆(d(u)))) + d(pd)
end
```



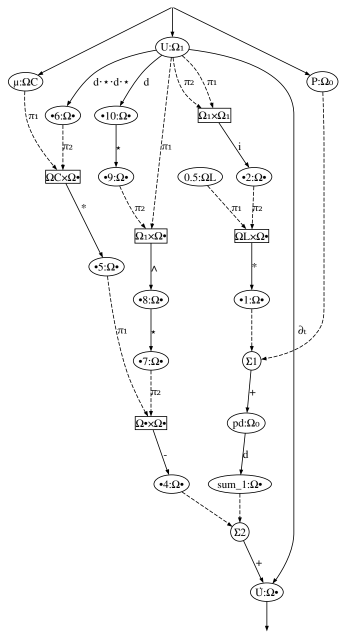


Figure 3: The specification of a conservative Incompressible Navier Stokes on a generic manifold along with a solution on a discretized sphere. Part of the equations are given in a domain specific language, the Decapode representation of the those parts, and the solution visualization (clockwise from top-left). A Decapode can be interpreted as a string diagram and then executed like a functional program that computes the time-step update for the dynamical system. Arrows pointing to a node from nothing are a visual aid to denote that such nodes (U, P, μ) need no prior computation, and nodes ($\dot{U}$) which point to nothing denote tangent variables.

as undirected wiring diagrams [6]. This approach builds off of the work of structured cospans [1, 3]. Physical models compose by sharing variables. Because our Decapodes representation uses copresheaves, the structured cospan construction provides the necessary hypergraph structure. A Decapode is compiled to simulation code via an algorithm similar to topological sort. State variables are inferred, corresponding to those quantities which require initial conditions. Iteratively, the outneighbors of these variables are computed according to the operator stored on the corresponding directed edge. (Binary operators are handled in a similar manner, with the code for computing the arguments emitted prior to that for the argument.)

Summary Decapodes.jl combines many aspects of applied category theory including symmetric monoidal categories, structured cospans, and synthetic approaches to differential geometry to build a practical and useful multiphysics solver framework. A high-level workflow is enabled in which computational problems are set up via the well-defined manipulation of explicitly-represented models formalized via applied category theory.

References

[1] John C. Baez & Kenny Courser (2020): *Structured Cospans*. *Theory and Applications of Categories* 35(48), pp. 1771–1822, doi:10.48550/ARXIV.1911.04630. Available at `http://www.tac.mta.ca/tac/volumes/35/48/35-48abs.html`. Publisher: arXiv Version Number: 3.

[2] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski & Fabio Zanasi (2022): *String Diagram Rewrite Theory I: Rewriting with Frobenius Structure*. *Journal of the ACM* 69(2), pp. 1–58, doi:10.1145/3502719. Available at `https://dl.acm.org/doi/10.1145/3502719`.

[3] José Luiz Fiadeiro & Vincent Schmitt (2007): *Structured Co-spans: An Algebra of Interaction Protocols*. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Till Mossakowski, Ugo Montanari & Magne

Haveraaen, editors: *Algebra and Coalgebra in Computer Science*, 4624, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 194–208, doi:10.1007/978-3-540-73859-6_14. Available at http://link.springer.com/10.1007/978-3-540-73859-6_14. Series Title: Lecture Notes in Computer Science.

[4] Timothy Hosgood (2022): *Diagrammatic differential equations: Formal categorical framework and applications to multiphysics simulation*.

[5] David E Keyes, Lois C McInnes, Carol Woodward, William Gropp, Eric Myra, Michael Pernice, John Bell, Jed Brown, Alain Clo, Jeffrey Connors, Emil Constantinescu, Don Estep, Kate Evans, Charbel Farhat, Ammar Hakim, Glenn Hammond, Glen Hansen, Judith Hill, Tobin Isaac, Xiangmin Jiao, Kirk Jordan, Dinesh Kaushik, Efthimios Kaxiras, Alice Koniges, Kihwan Lee, Aaron Lott, Qiming Lu, John Magerlein, Reed Maxwell, Michael McCourt, Miriam Mehl, Roger Pawlowski, Amanda P Randles, Daniel Reynolds, Beatrice Rivière, Ulrich Rüde, Tim Scheibe, John Shadid, Brendan Sheehan, Mark Shephard, Andrew Siegel, Barry Smith, Xianzhu Tang, Cian Wilson & Barbara Wohlmuth (2013): *Multiphysics simulations: Challenges and opportunities*. The International Journal of High Performance Computing Applications 27(1), pp. 4–83, doi:10.1177/1094342012468181. Available at http://journals.sagepub.com/doi/10.1177/1094342012468181.

[6] Sophie Libkind, Andrew Baas, Evan Patterson & James Fairbanks (2022): *Operadic Modeling of Dynamical Systems: Mathematics and Computation*. Electronic Proceedings in Theoretical Computer Science 372, pp. 192–206, doi:10.4204/EPTCS.372.14. Available at http://arxiv.org/abs/2105.12282v3.

[7] Mamdouh S. Mohamed, Anil N. Hirani & Ravi Samtaney (2016): *Discrete exterior calculus discretization of incompressible Navier–Stokes equations over surface simplicial meshes*. Journal of Computational Physics 312, pp. 175–191, doi:10.1016/j.jcp.2016.02.028. Available at https://linkinghub.elsevier.com/retrieve/pii/S0021999116000929.

[8] Evan Patterson, Andrew Baas, Timothy Hosgood & James Fairbanks (2022): *A diagrammatic view of differential equations in physics*. Mathematics in Engineering 5(2), pp. 1–59, doi:10.3934/mine.2023036. Available at http://www.aimspress.com/article/doi/10.3934/mine.2023036.

[9] Evan Patterson, Owen Lynch & James Fairbanks (2022): *Categorical Data Structures for Technical Computing*. Compositionality 4, p. 5, doi:10.32408/compositionality-4-5. Available at https://compositionality-journal.org/papers/compositionality-4-5/.

[10] David I. Spivak (2014): *Database queries and constraints via lifting problems*. Mathematical Structures in Computer Science 24(6), p. e240602, doi:10.1017/S0960129513000479. Available at https://www.cambridge.org/core/product/identifier/S0960129513000479/type/journal_article.