AME60714: Advanced Numerical Methods
Final Project

1. Overview

The course will culminate with a significant final project that includes the implementation and study of a numerical method. Students can work alone or in teams; each team will propse the topic and scope of their project to be negotiated/approved by the instructor. The project is expected to utilize concepts covered in the course; however, the most interesting projects will incorporate knowledge/methods beyond the material covered. For example, the project can relate to one's research or implement a method from a research paper. The final project will be a microcosm of a formal research project from inception to dissemination. As such, it will include a proposal, manuscript, and presentation, all of which will be counted toward the project grade.

2. Project anatomy

The final project "deliverables" will consist of a proposal, midterm checkpoint, technical report, presentation, and source code. Each deliverable will be graded and together will comprise the project grade; see Section 3.3 for details.

2.1. Proposal

The final project will begin with a 1-2 page proposal that describes the scope of the intended project and the team (Section 3.2). The proposal should include a description of the project including the objectives and goals, analytical work and implementation required, and tests that will be run to verify the correctness of the implementation (e.g., unit tests, order of accuracy tests, reproduce results from paper). It should also detail the steps that need to be taken to get from your current state to the completion of the project. Include any knowledge gaps that you will need to fill to complete the project, e.g., "X will be critical to the project for reason Y, but we currently have no idea how to do it", and how you intend to fill these gaps. Teams of more than one person must also include a collaboration plan that identifies how each team member will contribute to the project. Finally, include a rough timeline for the project and identify tangible milestones.

2.2. Midterm checkpoint

While not necessarily part of a research project, a midterm checkpoint is included to ensure reasonable, sustained progress on the final project is maintained throughout the semester. In an attempt to not increase the course workload unnecessarily, the checkpoint will involve no deliverables. This will simply be a 20 minute meeting with the instructor to discuss the team's progress and remaining work during the week of October 19, 2020. The midterm checkpoint will constitute 5% of the project grade, credit will be assigned on an all-or-nothing basis, and grading will be individualized to the team's project.

2.3. Technical report

All teams must submit a final report that should emulate a research paper. The report should contain an introduction that provides some motivation for the work (why did you choose the project) and an overview of project (work completed and results from the numerical experiments). One or more sections that detail the mathematical foundations of the method(s) and their implementation should be included. The main section(s) of the report must detail all the numerical experiments run, including tests used to verify the correctness of the code. Teams of more than one person must include a section titled "author statement" that details each team members contribution to the project. Finally, a short section should be included that concludes the project and reflects on the project/course. The instructor is particularly interested on what you learned from the project/course, how did the self-defined nature of the project facilitate or hinder your learning, and any other reflections you find relevant.

Teams are strongly encouraged to typeset in LaTeX. Be sure to cite any relevant literature and include a bibliography. Grades will be assigned based on the clarity and detail of the method description, the soundness of the verification tests and relevance of the numerical experiments, the scope and difficulty of the project, and overall quality of the report.

2.4. Presentation

All teams will give a 15-20 minute presentation to the class during the regularly scheduled final exam to describe their project. The presentation should cover the mathematical foundations of the method, implementation details, and numerical results; all team members are expected to speak. The grade for the presentation will be based on the team's ability to communicate the details of their project to their peers and the instructor, and answer questions.

2.5. Source code review

Finally, all teams must schedule a 30 minute meeting with the instructor to describe and demonstrate their code. Each team member will be expected to describe their contributions to the code. All code must be submitted to the instructor, including all unit and system tests used for verification, prior to the meeting. The grade for this deliverable will be based on team's ability to answer instructor questions, the design of the code, and rigor of verification tests.

3. Logistics

3.1. Code/implementation

Any programming language can be used for the coding aspects of the final project. Teams are welcome and encouraged to use the course MATLAB code `FEdu`; however, no other starter code will be provided.

3.2. Teams

Students can work alone or in teams. There is no explicit limit on team size; the scope of the project is expected to be proportional to team size (project scope will be negotiated with the instructor once the proposal is submitted). Teams should include a collaboration plan in the proposal (see details in Section 2.1) and an author statement in the report (see details in Section 2.3). Each student is expected to contribute to the proposal, implementation, midterm checkpoint, technical report, and presentation. For example, it is not acceptable for one student to do all coding tasks and another to do all writing.

3.3. Deliverables and evaluation

Project grades will be assigned based on: proposal (5%), checkpoint (5%), technical report (40%), presentation (25%), and source code review (25%). Grades will be assigned based on the scope and difficulty of the project and quality of the work; see individual sections for grading criteria.

3.4. Important dates

Monday, September 7, 2020	proposal due
Friday, September 11, 2020	project finalized
Friday, October 23, 2020	checkpoint
TBD (final exam slot)	presentations
Wednesday, November 25, 2020	all deliverables due

4. Sample topics

The following are suggestions for final project topics; note a complete proposal will require additional information, e.g., work plan, model problems, etc.

- Mesh generation [12, 4]
- Error estimation and adaptation [3, 18]
- Fluid-structure interaction [9] or general multiphysics [8]
- Flow control [5]
- Shape optimization [14, 13, 19]
- Shock tracking [6, 2, 21]
- Discontinuous Galerkin methods [7], finite volume methods [10]

- Conservation laws on deforming domains [11]
- Sensitivities for chaotic problems [15]
- Nonlinear model reduction [1, 17]
- Time-periodic solutions of PDEs [20]
- Magnetohydrodynamics [16]

References

[1] Saifon Chaturantabut and Danny Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[2] Andrew Corrigan, Andrew Kercher, and David Kessler. A moving discontinuous Galerkin finite element method for flows with interfaces. *International Journal for Numerical Methods in Fluids*, 89(9):362–406, 2019.

[3] Krzysztof Fidkowski and David Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.

[4] Meire Fortunato and Per-Olof Persson. High-order unstructured curved mesh generation using the winslow equations. *Journal of Computational Physics*, 307:1–14, 2016.

[5] Max Gunzburger. *Perspectives in flow control and optimization.* SIAM, 2002.

[6] Ami Harten and James Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of computational Physics*, 50(2):235–269, 1983.

[7] Jan Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications.* Springer Science & Business Media, 2007.

[8] Daniel Huang, Per-Olof Persson, and Matthew J. Zahr. High-order, linearly stable, partitioned solvers for general multiphysics problems based on implicit–explicit runge–kutta schemes. *Computer Methods in Applied Mechanics and Engineering*, 346:674–706, 2019.

[9] Emmanuel Lefrançois and Jean-Paul Boufflet. An introduction to fluid-structure interaction: application to the piston problem. *SIAM review*, 52(4):747–767, 2010.

[10] Randall LeVeque. *Finite Volume Methods for Hyperbolic Problems*, volume 31. Cambridge UniversityPpress, 2002.

[11] Per-Olof Persson, Javier Bonet, and Jaime Peraire. Discontinuous Galerkin solution of the navier–stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1585–1595, 2009.

[12] Per-Olof Persson and Gilbert Strang. A simple mesh generator in MATLAB. *SIAM review*, 46(2):329–345, 2004.

[13] Jamshid Samareh. A survey of shape parameterization techniques. In *NASA Conference Publication*, pages 333–344. Citeseer, 1999.

[14] Thomas Sederberg and Scott Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH computer graphics*, volume 20, pages 151–160. ACM, 1986.

[15] Qiqi Wang, Rui Hu, and Patrick Blonigan. Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, 2014.

[16] V. Wheatley, H. Kumar, and P. Huguenot. On the role of Riemann solvers in discontinuous Galerkin methods for magnetohydrodynamics. *Journal of Computational Physics*, 229(3):660 – 680, 2010.

[17] Masayuki Yano. Discontinuous galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Advances in Computational Mathematics*, 45(5-6):2287–2320, 2019.

[18] Masayuki Yano and David Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, 2012.

[19] Matthew J. Zahr and Per-Olof Persson. An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems. *Journal of Computational Physics*, 326(Supplement C):516 – 543, 2016.

[20] Matthew J. Zahr, Per-Olof Persson, and Jon Wilkening. A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints. *Computers & Fluids*, Special Issue on USNCCM13 International Symposium on Spectral and High-Order Methods, 2016.

[21] Matthew J. Zahr, Andrew Shi, and Per-Olof Persson. Implicit shock tracking using an optimization-based high-order discontinuous Galerkin method. *Journal of Computational Physics*, page 109385, 2020.