Homework 6

Computational Astrophysics (ASTR660)

(Due at the start of class on December 22, 2022)

Exercise 1

[Constrained transport method for MHD (1 pt)]

Using the definitions for the face-averaged magnetic fields and the line-averaged electromotive forces (EMFs) in the constrained transport (CT) method for magnetohydrodynamics (MHD), prove that the following equation will always hold by construction:

$$\nabla \cdot \mathbf{B}^{n+1} = \nabla \cdot \mathbf{B}^{n}. \tag{1.1}$$

That is, the divergence-free constraint of magnetic fields using the CT method can be preserved as long as it is satisfied in the initial conditions.

Exercise 2

[Solving elliptical systems (3 pt)]

In Lectures 13 and 14 we learned how to numerically solve the Laplace equation using the relaxation methods and the multigrid mothod given a boundary condition. Please complete the in-class exercise of implementing the Jacobi and Gauss-Seidel (GS) methods first if you have not done so. You could find a template Python Jupyter Notebook from here: `https://www.dropbox.com/s/r0m7zyy0kcxj89r/laplace_template_L14.ipynb?dl=0`.

(1) Please implement the successive over-relaxation (SOR) method. Run the test using $w = 1.2$ (w is the over-relaxation parameter) and show the results (left figure: 2D plot for the solution; right figure: errors versus the number of iterations N). Does the SOR method converge faster than the Jacobi and GS methods?

(2) Please implement the multigrid method using the full multigrid cycle (see p.18 on the slides of Lecture 14). For each level of grid (either coarse or fine), run the GS method for `NITER=100` iterations as in the template (for simplicity, we don't solve for the residual equation on the coarser grids). Run the code and show the results.

Exercise 3

[Comparisons between grid-based and SPH codes (2 pt)]

In Lecture 14 we learned that there could be significant differences in the numerical solutions of hydrodynamic problems obtained using grid-based and smoothed particle hydrodynamics

(SPH) codes. Please search in the literature (NASA ADS or arXiv) using keywords (e.g., "code comparison", "SPH"). Choose and read one of the articles you are interested in. Be sure to provide a link to the article you read and answer the following questions:

(1) What codes is this work trying to compare? What astrophysical problem are the authors investigating?
(2) After reading the Method section, briefly summarize the numerical techniques used in the study.
(3) What are the main differences of the results among the different codes? You may show a key plot to illustrate the differences.
(4) What is the reason for the differences (which part of the numerical technique used in which code is the cause)?

[**Next week (on 12/22/2022) we will have an in-class discussion about the *pros* and *cons* of grid-based vs. SPH codes, for which you may be able to find relevant information in the Introduction of the paper you chose (or simply consult Google). Be prepared to discuss your answers with your classmates and contribute your opinions!]

Exercise 4

[BONUS – try out an AMR code (3 pt) – due on 1/12/2023]

Let's try out one of the adaptive-mesh-refinement (AMR) codes that are widely used in astrophysical computation. Choose one code from the list below and run one test problem it provides (typically there would be some "quickstart" tutorials in the documentation). Ensure AMR is enabled. Please choose a code (or at least a test problem) that you have not run before.

A list of codes:
(i) `FLASH`: `https://flash.rochester.edu/site/index.shtml`. There is also a copy of version 4.2.2 available on CICA: `/data/hyang/shared/astro660CompAstro/FLASH4.2.2_public`.
(ii) `Athena++`: `https://github.com/PrincetonUniversity/athena-public-version`
(ii) `Enzo`: `http://enzo-project.org`
(iii) `PLUTO`: `http://plutocode.ph.unito.it`
(iv) `RAMSES`: `https://bitbucket.org/rteyssie/ramses`
(v) `GAMER`: `https://github.com/gamer-project/gamer/wiki`

Some of the above codes could be directly downloaded to the CICA cluster using GitHub (see a tutorial: `https://docs.github.com/en/get-started/quickstart/hello-world`). Type "`module load git`" and follow the instructions on their webpages to *git clone* the code.

Please answer the following questions:
(1) Briefly describe the test problem.
(2) What are the adopted grid refinement criteria?
(3) What are the boundary conditions?
(4) Does this test problem have an analytical solution? If yes, how does it compare with your numerical result?
(5) Compare results using different AMR levels (i.e., maximum resolution). Do the results converge?