Solving the Helmholtz equation using multigrid

May 1, 2018

The code is the same as in Tutorials/LinearSolvers/ABecLaplacian_F except that the variables in the equation are changed. The code has two options in the inputs file – prob_type 1 solves a Poisson problem with Dirichlet boundary conditions, and prob_type 2 solves a Helmholtz problem with Neumann boundary conditions. In this document, the Helmholtz problem with Neumann BCs is described.

1 Governing equation

This tutorial solves the Helmholtz equation (the Poisson equation is a special case) which takes the form

$$a\alpha(\boldsymbol{x})\phi - b\nabla\cdot(\beta(\boldsymbol{x})\nabla\phi) = \text{rhs}$$

where a, b, α and β are scalar functions of space. Neumann boundary conditions $\dfrac{\partial\phi}{\partial(\cdot)} = 0$ are applied in all directions.

2 Test

The computational domain is (0,1) in the x, y and z directions, and the solution is chosen to be $\phi = \cos(\pi x)\cos(\pi y)\cos(\pi z)$, since it satisfies the Neumann boundary conditions. Then we have for $a = 1$, $b = 2$, $\alpha(\boldsymbol{x}) = x^3 y$, and $\beta(\boldsymbol{x}) = xyz$,

$$\begin{aligned}\text{rhs} = x^3 y\cos(\pi x)\cos(\pi y)\cos(\pi z) + 2\pi(yz\cos(\pi y)\cos(\pi z)(\pi x\cos(\pi x) + \sin(\pi x)) \quad &+\\ xz\cos(\pi x)\cos(\pi z)(\pi y\cos(\pi y) + \sin(\pi y)) \quad &+\\ xy\cos(\pi x)\cos(\pi y)(\pi z\cos(\pi z) + \sin(\pi z))).\end{aligned}$$

The variables and the rhs are defined in rhs_helmholtz.f90.

3 Run the code

This example builds the code as a third party by just linking to the amrex library in tmp_install_dir/lib. It is done using an automated makefile generation procedure using mkmf (https://github.com/NOAA-GFDL/mkmf) which contains a perl script that creates a makefile taking care of all dependencies (see the git repo for more details). Currently the makefile is already built. If changes are made, then do the following.

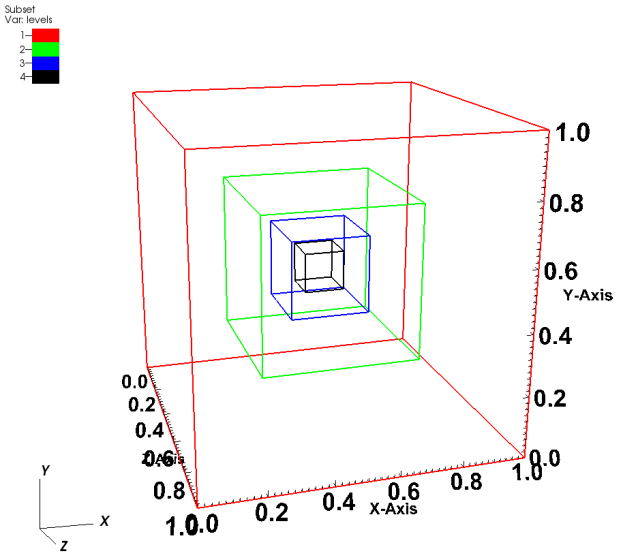
1. Change the path to tmp_install_dir in template.mk (tmp_install_dir is located in amrex)
2. In the source code directory execute – ./mkmf/bin/mkmf -t template.mk -p main3d.gnu.MPI.ex – this will build the make file
3. make – will build the executable main3d.gnu.MPI.ex
4. mpirun -np 4 ./main3d.gnu.MPI.ex inputs

Currently run_Helmholtz.sh has all the commands required to compile and execute the code – just do sh run_Helmholtz.sh.

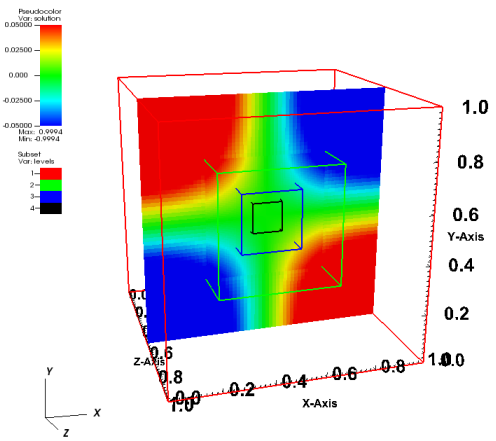
Note: The last line in template.mk contains flags from the verbose result of make using AMReX GNUMakefile. There could be more of these flags. These maybe important for optimization.

4 Result

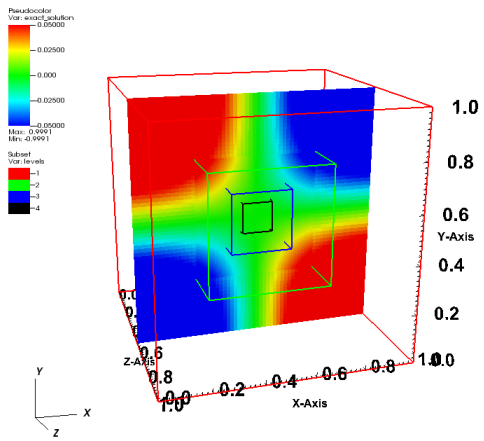
The number of cells (n_cells) and refinement levels (max_level) are taken as inputs from the inputs file. The subroutine init_grids generates the grid. Each grid level is half the size of the previous and is located the center of the domain. Fig. 1 shows the different grid levels for 3 refinement levels – including the level 0 grid there are 4 levels. The comparison of the numerical and the exact solution is also shown in Fig. 1. See the visualization section in the AMReX (https://amrex-codes.github.io/amrex/docs_html/Chapter11.html) for more details on visualization. The current results are from VisIt.



(a)



(b)



(c)

Figure 1: (a) AMR grid levels and comparison of (b) numerical and (c) exact solution.