METR3334 Programming Assignment No. 4.

Assigned on Monday April 18, 2022
Due on Friday May 6

Grade: 20 points out of 100 of total course grade

In this assignment, you will complete several missing pieces of codes within a 2D cloud model and run several experiments for cold and hot bubble initialization. We discussed the codes within class and you were asked to complete them as part of in-class exercise. Equations for the codes are now also included within the Jupyter notebook provided as part of the assignment.

You will start from jupyter notebook downloaded from http://twister.ou.edu/METR3334/2DCloudModel/2DCloud_assignment.ipynb, which is similar to the the 2D_Cloud_template.ipynb notebook we worked on in class.

1. **Complete codes for the following:**

    a) Within function poisson2d, added codes for calculating cx, cz, and C, and codes for calculating residual R to complete the function. Refer to notes in http://twister.ou.edu/METR3334/2DCloudModel/SOR.pdf or formula in the notebook.

    b) For the time integration of vorticity and potential temperature equations, add codes for calculating advection terms, and for calculating vorticity generation term, and use them properly in the time integration equations. We wrote codes together in class for 1D advection based on 1DAdvection_template.ipynb so you may want to reference that code. Please make sure that you consider all combination of positive and negative u and w cases. You can either use if-else conditional blocks according to Eqs.(26a-d) or the pseudo codes given in pages 14 and 15, or use the u+|u| form according to Eqs.(27) and (28) in http://twister.ou.edu/METR3334/2DCloudModel/2D_equations.pdf.

    c) Finally add codes to calculate u and w from stream function according to equation (32) in the above notes.

2. **Run the cloud model for 4 different configurations**

    a) A 10 K initial warm bubble (delpt=10 K) centered at thermz = 0.2*lenz for nx = nz = 65 and dt = 2.0
    b) A -10 K initial cold bubble (delpt=-10 K) centered at thermz = 0.3*lenz for nx = nz = 65 and dt = 2.0.
    c) A 10 K initial warm bubble (delpt=10 K) centered at thermz = 0.2*lenz for nx = nz = 129 and dt = 2.0. This run has twice the spatial resolution as the nx=65 case.
    d) A -10 K initial cold bubble (delpt=-10 K) centered at thermz = 0.3*lenz for nx = nz = 129 and dt = 2.0. This run has twice the spatial resolution as the nx=65 case.

Keep in mind the high resolution runs have 4 times the number of grid points so the program will take longer to run.

e) Plot potential temperature, stream function, vorticity contours and wind vectors using the provided codes as template, adjust the contour intervals and wind vector intervals as necessary to avoid too few or too many contours/vectors. The printout of max/min values of the fields can help you specifying contour intervals. Remember you can regenerate the plots without reruning the model if the model has just completed the time integration.

3. Prepare a short report with figures and submit code and output files

a) For each of the runs above, create a picture in a Word or Latex file consisting of four panels of plots at 300, 600, 900 and 1200 s, and include figure caption for each figure. Discuss the solutions, including the effect of resolution on the solutions.
b) Print the Jupyter notebook for each of the above four runs with all output messages and graphics to PDF files. Submit the PDF files, and the Jyputer notebook code to Canvas.

4. Bonus (for 2 bonus points)

Repeat runs 3 and 4 above with nx=nz=257 and dt=1.0 as runs 5 and 6, and include the corresponding figures in the report and discuss the results. Submit the corresponding output PDF files also.