MTMW14 2023: Numerical Modelling of the Atmosphere and Oceans

Project 2: A simplified model of ocean gyres and the Gulf Stream [1]

Goals of Project 2

1. Learn, in practice, about:

   fluid motion and numerical analyses methods in two-dimensions

   the fluid response and adjustment to external forcings

2. Experiment with:

   different numerical approaches to the advection problem

   individual aspects of modern geoscientific numerical modelling

3. Develop independent, logical and hypothesis-driven thinking

Problem Description

The model of Stommel (1948) is the simplest dynamical model able to represent a wind-driven circulation in a closed ocean basin, including a western boundary current. It describes the horizontal flow of an incompressible fluid with a free surface and solid lower boundary. The vertical pressure gradient is assumed to be in hydrostatic balance. The model is described by the shallow water equations, linearised about a resting state, with the addition of linear drag and wind stress:

$$\frac{\partial \eta}{\partial t} + H\boldsymbol{\nabla}\cdot\mathbf{V} = 0, \tag{1}$$

$$\frac{\partial \mathbf{V}}{\partial t} + (f_0 + \beta y)\mathbf{k}\times\mathbf{V} = -g\boldsymbol{\nabla}\eta - \gamma\mathbf{V} + \frac{\boldsymbol{\tau}}{\rho H}, \tag{2}$$

where the *prognostic variables* are:

1. the surface elevation $\eta(\mathbf{x}, t)$, and;

2. the depth-averaged horizontal velocity $\mathbf{V}(\mathbf{x}, t)$

where:
$\mathbf{x} = (x, y)$ is the (2D) spatial coordinate;
H is the resting depth of the fluid which is assumed constant;
$f_0 + \beta y$ is the approximation to the Coriolis parameter for a β-plane;
$\mathbf{k}$ is a unit vector in the vertical;
g is the gravitational acceleration;
γ is a linear drag coefficient;
ρ is the uniform density;
$\boldsymbol{\tau}$ is the wind stress acting on the surface of the fluid.

[1] partially based on an earlier practical by Emmanuel Hanert

Eqs. (1) and (2) are solved subject to no-normal flow (kinematic) boundary conditions. The computational domain is square defined by the ranges $[0, L] \times [0, L]$ where $L = 10^6$ m. The physical parameters f_0, β, g, γ, ρ and H are respectively set to 10^{-4} s^{-1}, 10^{-11} m^{-1}s^{-1}, 10 ms^{-2}, 10^{-6} s^{-1}, 1000 kg m^{-3} and 1000 m. The wind stress vector is specified as:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_x \\ \tau_y \end{pmatrix} = \tau_0 \begin{pmatrix} -\cos(\frac{\pi y}{L}) \\ 0 \end{pmatrix},$$

where $\tau_0 = 0.2$ N m^{-2}. Initially, the flow is at rest and velocity and elevation are zero.

Task A [5 points], TO BE HANDED IN AT THE END OF LAB SESSION 6

1. Which waves can this model support? Which wave propagation is most relevant to the calculation of the CFL criterion? Are those waves likely to be spawned: a) by your Initial Conditions (ICs) and/or b) by some other phenomenon during the integration? If a) or b), what will happen to such waves as we perform the integration to steady state requested in Task D?

2. Compute the Rossby radius of deformation, R_D, for this problem.

3. Briefly scan the Stommel (1948) paper, considering the main scientific objective of the simulation, and choose an appropriate grid-length, d, for that main purpose. Now compute the number R_D/d.

   What regime does R_D/d point to? Consequently, which Arakawa grid should be used for solving this problem with the numerical methods learnt so far? The recommended grid is Arakawa-C; if you disagree, justify your decision quantitatively.

   At this stage, identify and use the largest grid-length, d, deemed suitable to representing the dynamics of the problem , because it will save you much time in waiting for the numerical model to run. You will be asked, at a later stage, to increase the resolution of the numerical model, thus to ***re-compute the CFL criterion.***

4. Draw the variables in question on an Arakawa-C grid, including their indices at each locationa, as well as the meridional and zonal boundaries. You will need this detailed drawing in Task B, so leave sufficient space. Now, looking at your drawing, and taking into account the discussions in Lecture 6, understand how the fluid can move in the domain, and compute the CFL criteria for 2D flow.

5. Now, given the constraints you have derived so far, choose the time-step carefully in order to obtain a stable numerical solution and justify your choices. What is the maximum wave phase speed that you anticipate and what is the associated upper bound on the time-step?

   Do remember that the speed in the CFL criterion can refer to a wave phase speed, or to a fluid velocity!. Annotate your diagram explicitly with your final choice of time step Δt and of grid spacing d.

Task B [2 points] TO BE HANDED IN AT THE END OF LAB SESSION 6

Expand on your drawing in the previous task, by reasoning out and adding:

1. What kind of boundary conditions will you need for η and what kind will you need for **V**? Draw the boundary conditions for the prognostic variables (including their values) on each boundary of your domain.

2. a flowchart illustrating all the stages in your model initialisation and integration, including the names, inputs and outputs of all functions you are going to design.

Draw neatly with pencil and ruler, and scan, else use electronic drawing tools, to form a (beautiful and useful!) diagram, to be included in your report. Do not start coding anything until you have completed this diagram.

Task C [3 points]

Mushgrave (1985) derived an analytical solution of Eqns. (1) and (2) for steady state:

$$
\begin{aligned}
u_{st}(x,y) &= -\frac{\tau_0}{\pi\gamma\rho H} f_1(\frac{x}{L}) \cos(\frac{\pi y}{L}), & (3)\\
v_{st}(x,y) &= \frac{\tau_0}{\pi\gamma\rho H} f_2(\frac{x}{L}) \sin(\frac{\pi y}{L}), & (4)\\
\eta_{st}(x,y) &= \eta_0 + \frac{\tau_0}{\pi\gamma\rho H} \frac{f_0 L}{g} \left[\frac{\gamma}{f_0 \pi} f_2(\frac{x}{L}) \cos(\frac{\pi y}{L}) \right. \\
&\quad \left. + \frac{1}{\pi} f_1(\frac{x}{L}) \left(\sin(\frac{\pi y}{L})(1 + \frac{\beta y}{f_0}) + \frac{\beta L}{f_0 \pi} \cos(\frac{\pi y}{L}) \right) \right], & (5)
\end{aligned}
$$

where

$$
\begin{aligned}
f_1(x) &= \pi \left(1 + \frac{(e^a - 1)e^{bx} + (1 - e^b)e^{ax}}{e^b - e^a} \right), \\
f_2(x) &= \frac{(e^a - 1)be^{bx} + (1 - e^b)ae^{ax}}{e^b - e^a},
\end{aligned}
$$

$$
a = \frac{-1 - \sqrt{1 + (2\pi\epsilon)^2}}{2\epsilon}, \quad b = \frac{-1 + \sqrt{1 + (2\pi\epsilon)^2}}{2\epsilon}, \quad \epsilon = \frac{\gamma}{L\beta}.
$$

Note that the elevation is determined up to an unknown constant of integration, η_0. Since $\eta_{st}(0, L/2) = \eta_0$ in the exact solution, the final value of $\eta(0, L/2)$ from the model is a good estimate of η_0. In practice, at the very start of the simulation (as the gyre is just starting to form, you can use $\eta_0 = 0$, but in order to compare to the model after it has achieved its final state, please take the η_0 from that model's state and re-compute the analytic solution.

This exact solution can be used as a **benchmark test for your numerical model:** you will compare it with your numerical model result (Tasks D, F, G) at the final time (approximate steady state) using contour plots of surface elevation, η.

Task D [5 points]

Write a finite difference model using the Arakawa-C grid in space with the following forward-backward time scheme (Matsuno (1966); Beckers and Deleersnijder (1993)), which alternates the order in which the two momentum equations are solved. First u before v:

$$
\begin{aligned}
\eta^{n+1} &= \eta^n - H\Delta t\left(\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y}\right)\\
u^{n+1} &= u^n + (f_0 + \beta y)\Delta t\, v^n - g\Delta t\frac{\partial \eta^{n+1}}{\partial x} - \gamma\Delta t\, u^n + \frac{\tau_x}{\rho H}\Delta t,\\
v^{n+1} &= v^n - (f_0 + \beta y)\Delta t\, u^{n+1} - g\Delta t\frac{\partial \eta^{n+1}}{\partial y} - \gamma\Delta t\, v^n + \frac{\tau_y}{\rho H}\Delta t,
\end{aligned}
$$

and then v before u

$$
\begin{aligned}
\eta^{n+2} &= \eta^{n+1} - H\Delta t\left(\frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y}\right),\\
v^{n+2} &= v^{n+1} - (f_0 + \beta y)\Delta t\, u^{n+1} - g\Delta t\frac{\partial \eta^{n+2}}{\partial y} - \gamma\Delta t\, v^{n+1} + \frac{\tau_y}{\rho H}\Delta t,\\
u^{n+2} &= u^{n+1} + (f_0 + \beta y)\Delta t\, v^{n+2} - g\Delta t\frac{\partial \eta^{n+2}}{\partial x} - \gamma\Delta t\, u^{n+1} + \frac{\tau_x}{\rho H}\Delta t.
\end{aligned}
$$

then returning to the first set of equations, and so on.

For the code implementation you are encouraged to write a class of operators for defining your domain, your finite differences etc., with the advantage that well-written functions will optimise very strongly when using "@jit" directives with Numba or, even better, with JAX. Compare to the basic Numpy performance (how fast the model runs once compiled with "@jit" versus standard Numpy).

STEP D.1: SANITY CHECK: run the model for a day and check that it is stable and that it is starting to produce a smooth clockwise gyre (anticyclonic in the NH) within the ocean basin. Produce plots, showing at 1 day:

1. u versus x along the grid, closest to the southern edge of the basin,
2. v versus y along the grid, closest to the western edge of the basin,
3. η versus x through the middle of the gyre,
4. a 2D contour plot showing elevation η.

STEP D.2: STEADY STATE: now continue the simulation to steady-state: this will take **N days**, with N to be determined by your investigation in Task E. Now produce these plots once again, now at steady state.

1. u versus x along the grid, closest to the southern edge of the basin,

2. v versus y along the grid, closest to the western edge of the basin,

3. η versus x through the middle of the gyre,

4. a 2D contour plot showing elevation η.

STEP D.3: Differences between steady state and analytical model: now you will produce difference plots to assess how good or bad your numerical model is against the analytical model.

After you have achieved steady state, calculate the differences between your numerical solution at steady state in this task and the exact solution (what you developed in Task C): $u' = u - u_{st}$, $v' = v - v_{st}$ and $\eta' = \eta - \eta_{st}$.

Next, compute the energy calculated from the difference fields, $E(u', v', \eta')$. Refer to the instructions for the energy integral in Task E.

Task E: the Energy method [3 points]

Calculate numerically the total energy of the perturbation from the resting ocean:

$$E(u, v, \eta) = \int_0^L \int_0^L \frac{1}{2}\rho(H(u^2 + v^2) + g\eta^2)\, \mathrm{d}x\, \mathrm{d}y$$

and store the result for each time-step of the model in an array.

Plot the evolution of energy with time: what does this tell you about the quality of your decisions in Tasks A, B? Extend your model run until the energy appears to reach a steady state and show the final energy versus time plot. How long does it take to approach approximate steady state?

Now halve the grid-spacing of your model and demonstrate, using the energy error measure, that the model tends towards the exact solution as resolution is increased. CAREFUL: do remember to re-consider the 2D CFL criteria chosen in Task A.

Task F: the Semi-Lagrangian method [6 points]. Do not attempt this task until you have successfully completed all previous tasks and written them up.

Now go back to the notes from lecture 4 and re-derive analytically the **linearised SWEs**: start again with slide 11 and see how we finally obtained the Eulerian form in slide 14.

Modify your numerical model (your implementation of equations (1) and (2) in this Project 2) to now use a *semi-Lagrangian* method to solve the **full non-linear SWEs**, where the local rate of change in equations (1) and (2) is replaced by Lagrangian derivatives. Make sure that you make a safe copy of your linear model first! Please keep it as simple as you can: make use of the template on slides 17,18 of the Semi-Lagrangian lecture notes (Lecture 6, equations 11,12), with simple linear interpolation in space and time. Note that this is a crude simplification, and will therefore present limitations when compared to a state-of-the-art semi-Lagrangian scheme.

Run your nonlinear model for 1 day and compare with the results from Task D (using the same grid-length and time-step). Then run to 30 days and plot the final surface elevation. Compare with the exact steady state solution of the linearised equations.

As in Task D, calculate the difference between your numerical solution in this task and the exact solution (Task C): $u' = u - u_{st}$, $v' = v - v_{st}$ and $\eta' = \eta - \eta_{st}$ and then compute the energy calculated from the difference fields, $E(u', v', \eta')$. Refer to the instructions for the energy integral in Task E.

What do you observe overall? Note that your nonlinear model is likely to be much slower to run than the linear one: if you cannot reach 30 days, do try a shorter run (or else try to optimise with cython, numba and the like).

OPTIONAL Task G1: for students who are interested in the physical oceanography problem [bonus 5 points]

Refer to Remi Tailleux's oceanography notes (ask PLV) and set up a classic boundary current problem. Compare the solutions under the two different forcings and explain, in terms of a chain of mechanisms, what happens in each, by comparison and by physical insight.

OPTIONAL Task G2: for students who are interested in computational optimisation aspects [bonus 5 points]

Experiment with the compilation directives that include Intel-specific optimisations, such as vectorisation. Further experiment with advanced uses of JAX "decorators", and use the more advanced commands in your code, particularly the gradient operators. What speed gains can you achieve? Is your code easier or harder to understand after using JAX operators, instead of explicit computation?

OPTIONAL Task G3: for students who are interested in the concepts behind "Domain Specific Languages" [bonus 5 points]

Experiment with the use of the xgcm, libraries. Re-create your Arakawa grids with the appropriate variable staggering, as well as the boundary conditions using calls to the xgcm library. Further experiment with differential operators. Is your code faster/slower? Is it easier or harder to understand after using this alternative approach to creating and using variables on the grids, instead of your original explicit definition?

OPTIONAL Task G4: for students who are proud of their Runge-Kutta time scheme [bonus 5 points]

Re-write the solver using RK4 instead of the forward-backward scheme proposed in this project. How do the two compare in terms of the quality of the solutions? What time step could you afford to use with RK4, and what is the speed compared to what you did in Task D?

OPTIONAL Task G5: for students who are interested in initialisation and ensemble forecasting [bonus 5 points]

What happens if you alter the Initial Conditions (ICs) that force the system to start from a non-resting state? Does the amplitude (in any spatial direction) matter? What happens if, additionally, you impose a continous stochastic perturbation, as you did at the end of Project 1?

OPTIONAL Task G6: only for advanced students who succeeded with the SL scheme in full [bonus 5 points]

1. Once you are sure that your Task F is correct, make use of all the suggestions in Durran's book to increase the accuracy of the SL scheme: implement quadratic or

cubic interpolation in space, saving velocity increments in time, to improve your SL solutions from Task F.

2. Run your nonlinear model once again for 1 day and compare with the results from Tasks D, E and F (using the same grid-length, but you should now be able to use a substantially longer time-step).

3. Finally run to 30 days and plot the final surface elevation. Compare with the exact steady state solution of the linearised equations.

4. What do you observe when compared with the results from Tasks D, E, F? Why does it happen?

References

Beckers, J. and Deleersnijder, E. (1993). Stability of a FBTCS scheme applied to the propagation of shallow-water inertia-gravity waves on various grids. *J. Computational Phys.*, **108**, 95–104.

Matsuno, T. (1966). Numerical simulation of the primitive equations by a simulated backward difference method. *J. Meteorol. Soc. Japan*, **44**, 76–84.

Mushgrave, D. (1985). A numerical study of the roles of subgyre-scale mixing and the western boundary current on homogenisation of a passive tracer. *J. Geophys. Res.*, **90**, 7037–7043.

Stommel, H. (1948). The westward intensification of wind-driven ocean currents. *Trans. Am. Geophys. Union*, **29**, 202–206.

Project 2 is worth 35% of the mark for this module. Assessment is via a short scientific report describing what you have done. The report must be word-processed, **submitted exclusively as a PDF file**, and:

1. preferentially (NOT compulsory) in the form of a Python notebook (3 bonus points)
2. the scientific discussion should not exceed **six sides of A4 excluding figures**. Extra credit will be given to reports that are concise and clearly written, so follow this discipline: start writing from day 1, and refine each week.
3. your code is integral part of the report and will also be assessed (upload all required to reproduce your results onto BB by the deadline below). It should be highly structured, so that the main program should be at most 1 page. Everything else is to be written as functions, called by the main program (those should be included as an Appendix, and do not count towards the page limit, but concise code will earn full points)
4. declare variable names and units at the top of the main program and of each function
5. all figures and equations must be labelled, numbered and captioned

Start by stating the problem and the fundamental equations, but do not include lengthy background material or a literature review. The emphasis is on the scientific justification of your method to solve this problem numerically and the accuracy and interpretation of the results. Follow the structure: formulation, implementation, evaluation.

The marks will be distributed as follows:

- Numerical implementation of the model to achieve Tasks D,E,F (G). There is no need to repeat the equations given in this problem description. Marks will also be included for the style and legibility of your program (add comments in it). [6/35]
- Derivations in Task A. [5/35]
- Drawings in Task B. [2/35]
- Results from the analytical model, Task C. [3/35]
- Evaluation of numerical model errors in Task D. [5/35]
- Calculation of the energy time series in Tasks D, F. [3/35]
- Task F: implementation of the semi-Lagrangian scheme and results from the non-linear model. [6/35]
- The scientific presentation of the results, including a coherent and concise writing style (remember to label sections, graph axes, units etc). [5/35]

The deadlines for submission of the report are:

1. Tasks A,B derivations and drawings (electronically scanned and clearly labelled with student number): **16:00, Thursday 22 February 2024**

2. Full Report: **12:00, Friday 22 March 2024**.

Late submissions will result in loss of marks. Submit your reports electronically via the BlackBoard submission point. If you decide to submit your code via the Gitlab, please still submit the link to the code via the BB site. *Put all your files in a directory, then tar and gzip that directory prior to submission.* For technical support with BlackBoard, please contact the BB support team.

Figure 1: Project 2 survival guide