Exercise 3, TFY4235 Computational physics

Martin Johnsrud

Introduction

This paper documents the implementation and results of the simulation as given in [2]. By using a finite difference scheme, the process of absorption of CO_2 by the global ocean in the form of dissolved inorganic carbon (DIC). A simplified model of the physics is used, where both the atmospheric and oceanic concentration of carbon is constant at a given height, meaning there is in effect only one spatial dimension. The method is tested against known solutions, and the convergence is monitored. It is then used to extract results of how the ocean absorbs carbon, and at what rate.

Theory and implementation

The implementation of the simulation largely follows [2], as laid out in the appendix. This section describes the particular choices made, and in a notation consistent with the code. The diffusion equation can be written as

$$\Delta t \frac{\partial}{\partial t} C(z,t) = \Delta t \left(K(z) \frac{\partial^2}{\partial z^2} + \frac{\mathrm{d}K(z)}{\mathrm{d}z} \frac{\partial}{\partial z} \right) C(z,t) = \mathcal{D} C(z,t).$$

$\mathcal{D}$ is thus a linear operator which can be approximated by a discretization. With the scheme as descirbed in [2], and including the boundary condition this gives

$$\Delta t \frac{\partial}{\partial t} C_n(t) = \mathcal{D}_{nm} C_n(t) + S_n(t),$$

where summation over repeated indices are implied, and

$$\mathcal{D} = \begin{pmatrix} -4\alpha K_0 - 2\Gamma & 4\alpha K_0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & -\frac{\alpha}{2}K_i' + 2\alpha K_i & -4\alpha K_i & \frac{\alpha}{2}K_i' + 2\alpha K_i & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 4\alpha K_N & -4\alpha K_N \end{pmatrix}, \quad \alpha = \frac{\Delta t}{2\Delta z^2},$$

$$S(t) = \begin{pmatrix} 2\Gamma C_{\mathrm{eq}}(t) & 0 & \dots & 0 \end{pmatrix}^T \quad \Gamma = 2\frac{\alpha k_w \Delta z}{K_0} \left(K_0 - \frac{1}{2}(-\frac{3}{2}K_0 + 2K_1 - \frac{1}{2}K_2) \right), \quad K_n' = K_{n+1} - K_{n-1}.$$

Using the Cranck-Nichelson scheme to discretize the time part yields

$$C_n^{i+1} = C_n^i + \frac{1}{2}(\mathcal{D}_{nm} C_m^i + S_n^i) + \frac{1}{2}(\mathcal{D}_{nm} C_m^{i+1} + S_n^{i+1}),$$

so the equation to be solved to get the next timestep is

$$A_{nm} C_m^{i+1} = V_n^i, \quad V_n^i = \left(\delta_{nm} + \frac{1}{2}\mathcal{D}_{nm} \right) C_m^i + \frac{1}{2}(S_n^i + S_n^{i+1}), \quad A_{mn} = \left(\delta_{nm} - \frac{1}{2}\mathcal{D}_{nm} \right).$$

The implementation of this system of equation uses SciPy's sparse matrix library. After creating sparse realizations of A, SciPy's `splu` is used to generate the LU decomposition `LU` of A. This is an object with

methods such as `.solve()`, which utilizes the LU decomposition. The wrapper `simulate` then loops over $N_t - 1$ steps, using `solve(V) = lambda V: LU.solve(V)`, where `V` is as given above. The results of the simulation of C is kept in a NumPy array. However, it is not necessary to keep all steps made in time. `simulate` can be passed a integer argument `save`, the number of time steps to be kept in the array. This entails that $N_t - 1$ is divisible by save $- 1$, which is checked by an `assert` statement.

To check convergence of the method, both in time and space, a way of measuring error is needed. In this project, relative root mean square (RMS) error is used. Given a concentration $C(z)$ and a reference $C_0(z)$ at a given time, sampled at a set of points $\{z_i\}_{i=1}^N \subset [0, L]$, relative RMS error is given by

$$\Delta_{\text{rms}} = \sqrt{\frac{1}{N}\sum_i \left(\frac{C(z_i) - C_0(z_i)}{C_0(z_i)}\right)^2}.$$

When comparing concentrations with different values for N_z, as is necessary to find the convergence of the method in space, one must make sure that the points of comparison are the same for both C and C_0. This is done by setting $N_z = 5 \cdot 2^N a + 1$ for C_0, and $N_z = 2^n a + 1$ for C. Here, a and N are integers, and n is some integer less or equal to N. This ensures that the set of points $\{z_i\}_{i=1}^{2^n a+1}$ that $C(z)$ is simulated on, $C_0(z)$ is as well, and they can be compared. This is implemented by `get_rms` in `utilities.py`, which takes a list of arrays and compare each with the last element. The task also requires the spatial integral of the concentration an related quantities. This is done with the SciPy implementation of Simpsons method, `simpson`.

Tests

Make sure the implementation gives accurate answers, a series of tests are run, where the answer is known or there is a method to check error. The method used in this implementation has quadratic convergence, both in time and space. A convergence test was implemented for a simple test case with constant diffusivity K and a Gaussain initial concentration. The result is shown in Figure 1. A constant concentration of CO_2 should remain constant, regardless of $K(z)$. This test is shown in Figure 2, with a both a constant $K(z)$, and a smooth step function between two values of K as described in [2], problem 2. For both, the constant initial concentration is close to unchanged, save for numerical errors. The systems should also, given $k_w = 0$, conserve mass. To test this, a initial distribution of two Gaussian functions were evolved in time, both with constant and non-constant diffusivity. The result is shown in Figure 3, where the mass is conserved to a good approximation.

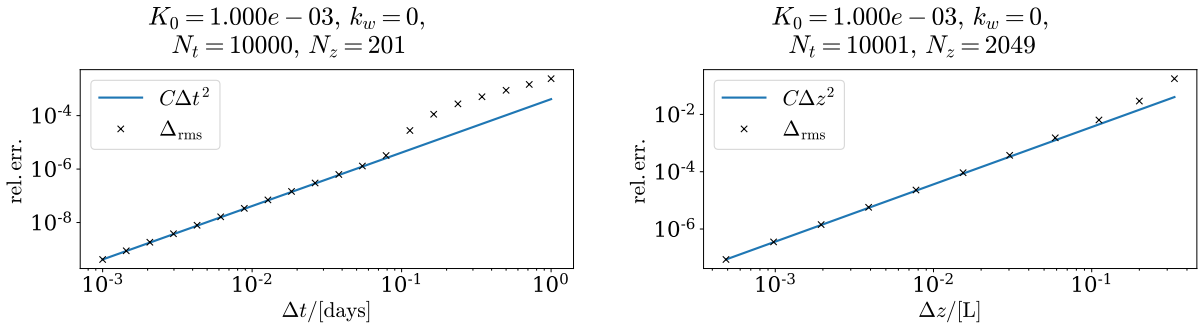

Figure 1: Error, measured as the relative RMS deviation from a reference value, after 1 day simulation of an Gaussian initial concentration.

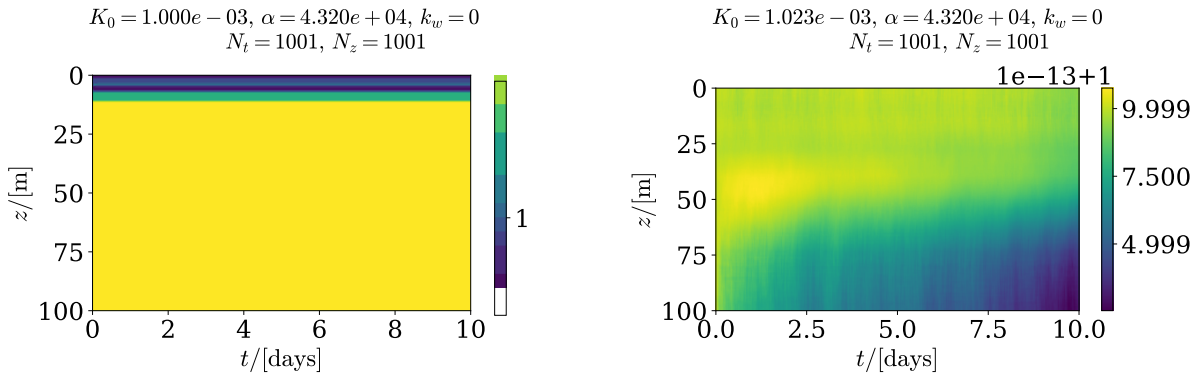

Figure 2: The time evolution of an constant concentration. $K(z)$ is constant in the system on the left, and varies in the system on the right. The largest deviation from the inital calues of the systems are of order 10^{-15} and 10^{-13}, respectively.

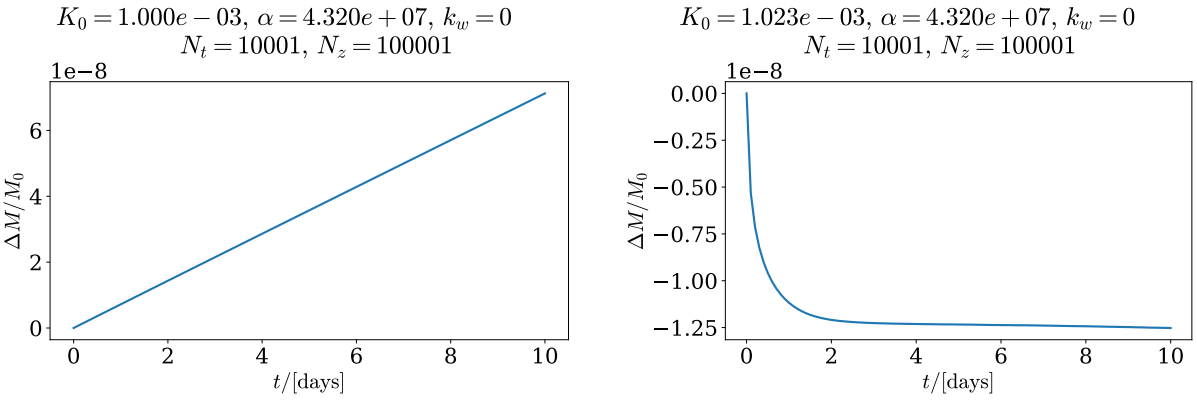

Figure 3: Relative change in mass of a system with constant (left) and varying (right) diffusivity.

A sharply peaked Gaussian package should have a variance that increases linearly with time, then approach a steady state. This is shown in Figure 4. For a small Biot nuber and given a zero partial pressure in the atmosphere, the depletion of CO_2 from the ocean should follow an exponential decay. Lastly, Figure 6 shows how the ocean reaches a equilibrium with the atmosphere, given a non-zero, positive mass-transfer coefficient k_w and either constant or non-constant diffusivity.

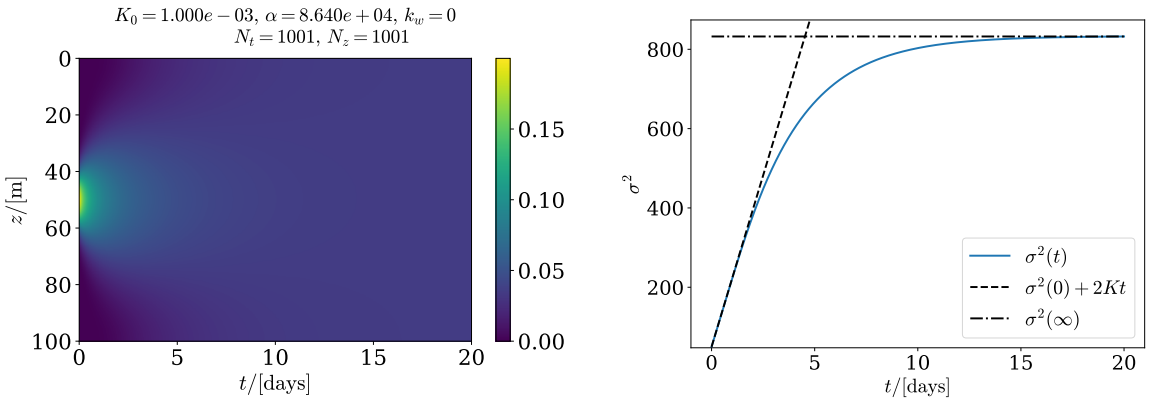

Figure 4: A system with an initial Gaussain distribution. On the right, the variance as a function of time is shown. The increase is initially constant, but then approaches a steady state.

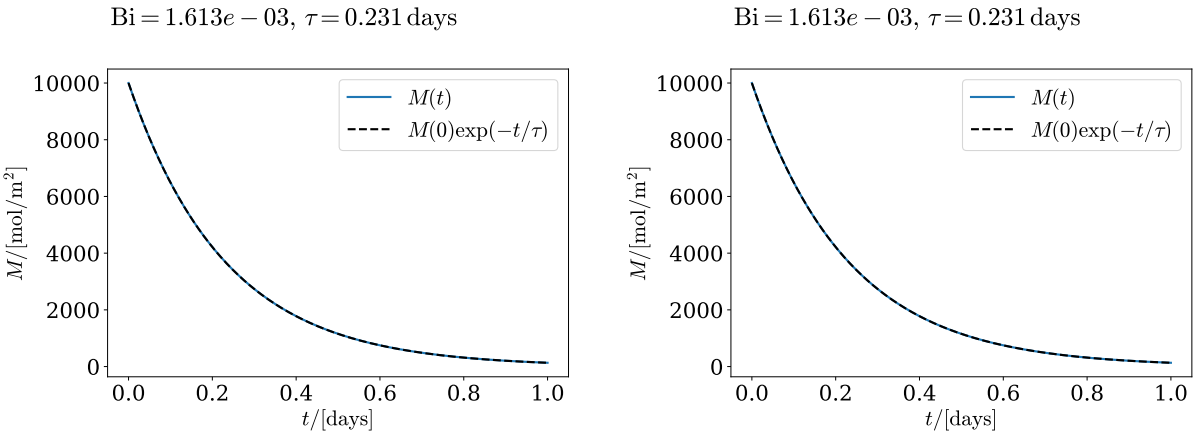

Figure 5: The slow removal of CO_2 from the ocean, when the atmosphere contains a partial pressure of 0. Constant (left) and varying (right) diffusivity.

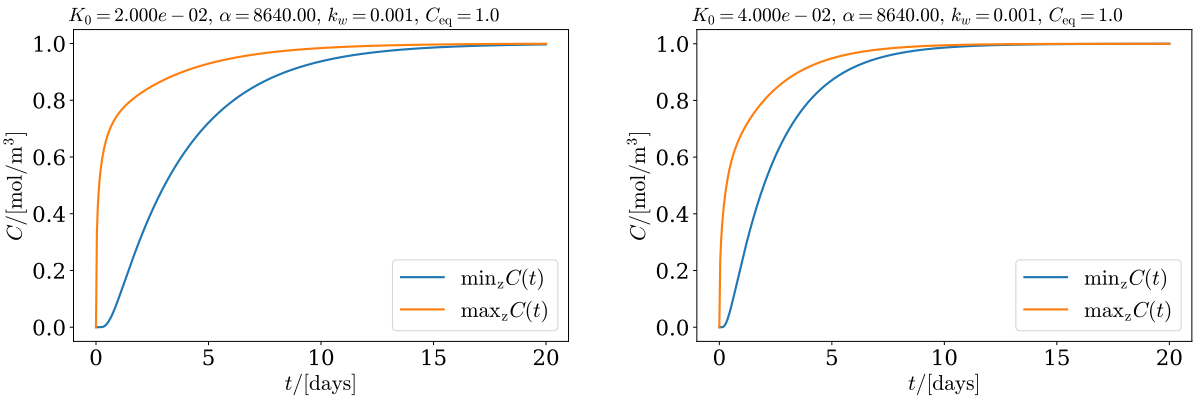

Figure 6: Equilibration of the ocean with a atmosphere with $C_{\mathrm{eq}} = 0.5$. The figure on the left is a system with constant diffusivity, the left side has a oscillating but always positive diffusivity.

Results and discussion

Shallow waters

To simulate the uptake of CO_2 in shallow areas, $L = 100\,\mathrm{m}$, diffusivity $K(z)$ is used as given in equation (11) in [2]. The diffusivity is illustrated in Figure 8. Figure 7 illustrates the convergence of the simulation, which shows that the simulation still has quadratic convergence both in time and space. The convergence test are run for a time of 10 days.

The simulation is then run for 180 days, with $N_t = 10001$, $N_z = 100001$. This corresponds to $\Delta t = 180/10000 = 1.8 \cdot 10^{-2}$days, and $\Delta z = L/100000 = 10^{-5}L$. This gives error well bellow 0.1%, and takes very little time. The results are shown in Figure 8, together with the form of the diffusivity, $K(z)$. Figure 9 shows how the minimum and maximum concentration evolves as a function of time, together with some snap shots of the concentration as it evolves. The surface of the ocean, where wind dominates the diffusion, first absorbs the CO_2. The deeper parts of the ocean lacks behind some, as there low diffusivity around 40 meters are a bottle neck, but it does not take long before it penetrates down to the ocean floor. This shows that after 90 days, the ocean is nearly in equilibrium with the atmosphere, and at 180 it is so close as does not make a difference. The model therefore suggests that, at least in shallow waters, the concentration of DIC in the ocean is in sync with the changes in atmosphere. It will in less than a year catch up to differences that are as large as the actual concentration of DIC.

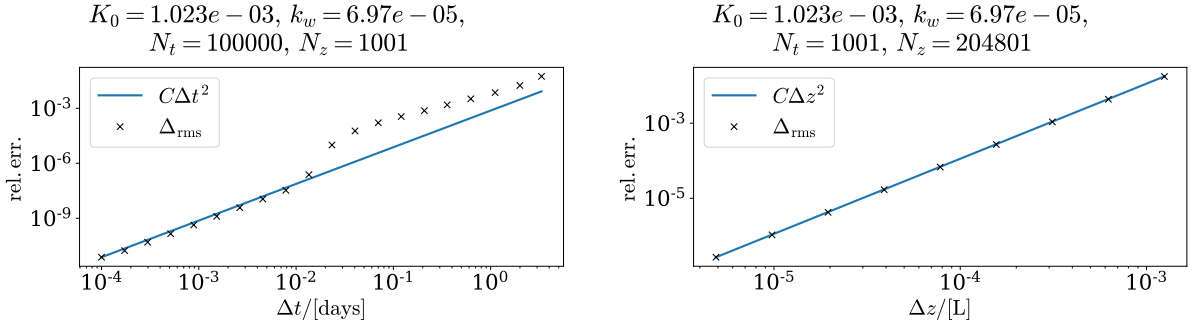

Figure 7: Error, as a function of step length in both time and space.

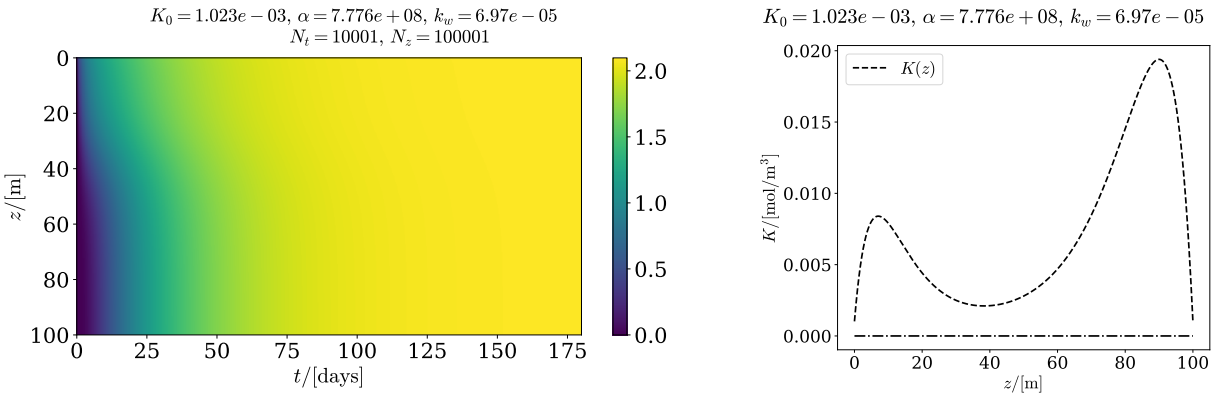

Figure 8: Concentration as a function of time is shown on the left, the diffusivity on the right

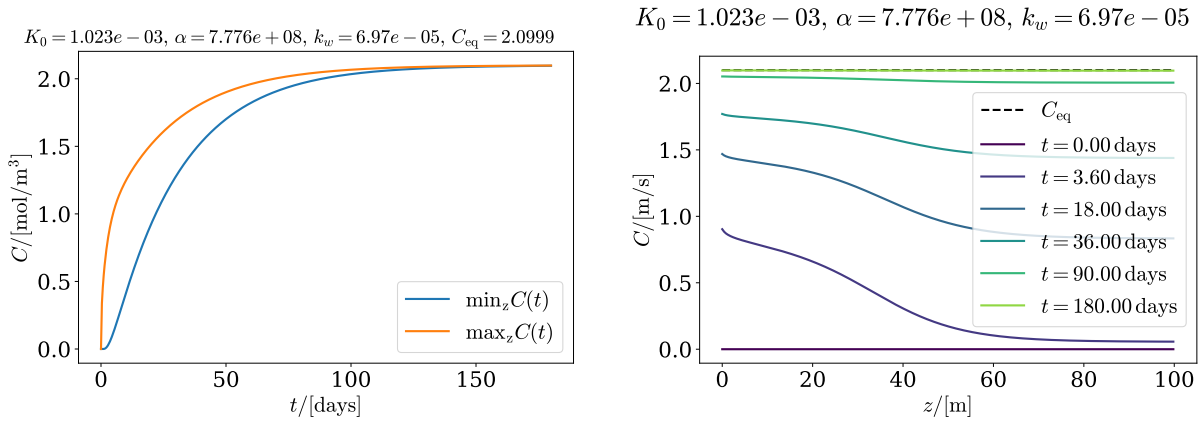

Figure 9: Minimum and maximum concentrations, as a function on time is shown on the left, snapshots of the concentration at different times on the right.

Deep waters

In this section, the ocean is modeled as 4000 m deep, and with a 2-layered diffusivity. This model is run for a longer time of several years, and therefore includes the increase in CO_2 in the atmosphere, as described in [2]. Figure 10 shows the convergence of the simulation, confirming the simulation still converges quadratically. The results in this section is obtained using $N_t = N_z = 10001$, giving a step length of $\Delta z = L/10000 = 10^{-4}L$, and $\Delta t = 10 \cdot 365\,\text{days}/10000 \approx 4 \cdot 10^{-1}\,\text{days}$. The convergence test suggests this should give high enough precision, considering the crudeness of the model, and running time is around 3 seconds.

Figure 11 shows the concentration of CO_2 as a function of time, as given by the simulation. Figure 2 in [2] illustrates the absorption mechanism of the ocean, where a thin layer at the top of the ocean, the mixed layer, absorbs CO_2 easier by the help of wind and turbulence. Then, it slowly seeps into the deep ocean. This phenomenon is illustrated in Figure 11, where the upper 100 meters of the ocean is more or less in equilibrium with the atmosphere, while the deep ocean lags behind. The slight slope of 10 year line in the figure to the right also shows that shallow water above deep waters might be less in sync than the shallow water that was synced earlier, as it does not have no-flux boundary condition, but rather always have a flux down in to the deep.

The units of C in this exercise is $\mathrm{mol/m^{-3}}$, so the "mass" given by the formula $M = \int \mathrm{d}z C(z)$ is mol per square meter of ocean. Using $A = 0.36 \cdot 10^{15}\mathrm{m}^2$ as the value of the surface area of the global oceans, and $\mathrm{Mm} = 12\,\mathrm{g/mol}$ as the molar weight of carbon in CO_2, the total weight of the carbon stored in the ocean as a function of time is given by

$$M(t) = \mathrm{Mm}\, A \int_0^L \mathrm{d}z\, C(z,t).$$

The mass given by the simulation is plotted in Figure 12. This simulation results in a uptake of $\Delta M = 120.8 \cdot 10^{15}\mathrm{g}$ carbon over 10 years, i.e. an average absorption rate of $12.08 \cdot 10^{15}\mathrm{g/year}$.

Gruber et al. [1] esitmates that the ocean had an annual uptake of carbon of around $2.6 \cdot 10^{15}\mathrm{g}$ from 1994 to 2007. This simulation obtained a result that is about 4.6 times large. This error is likely due to the many simplifications made to the model, for example modeling the ocean as having the same depth everywhere, modeling it and the atmosphere above it as 1 dimensional, and the diffusivity as constant in time and a simple Sigmoid curve. Taking into account all these simplifications, the result seems to indicate that this simulation is a good stepping stone towards a more accurate one.

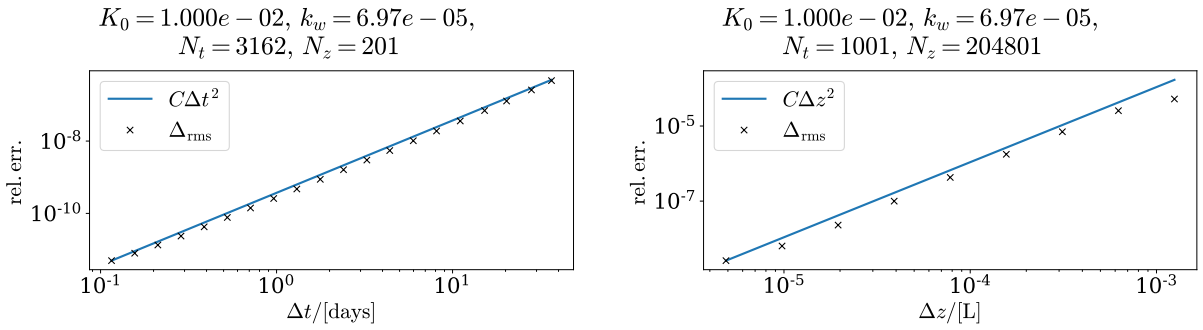

Figure 10: Error, as a function of step length in both time and space.

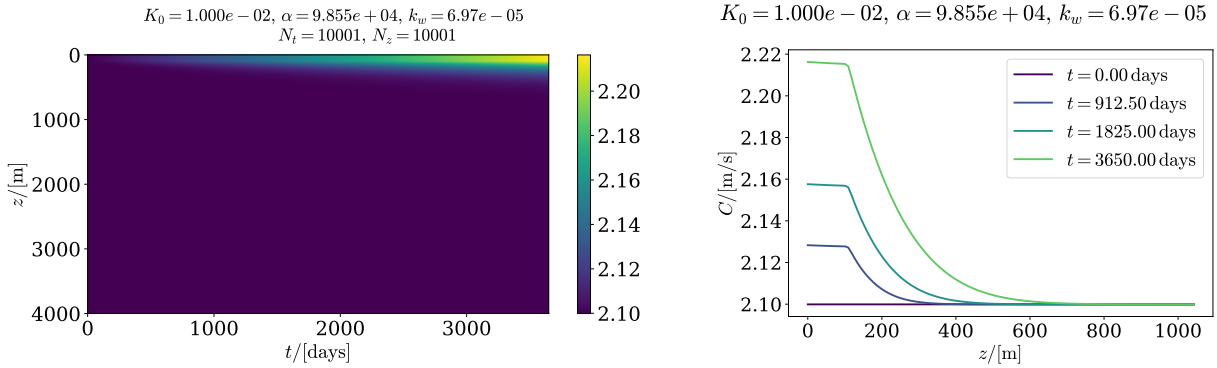

Figure 11: The concentration of CO_2 as a function of time and depth, over 10 years.

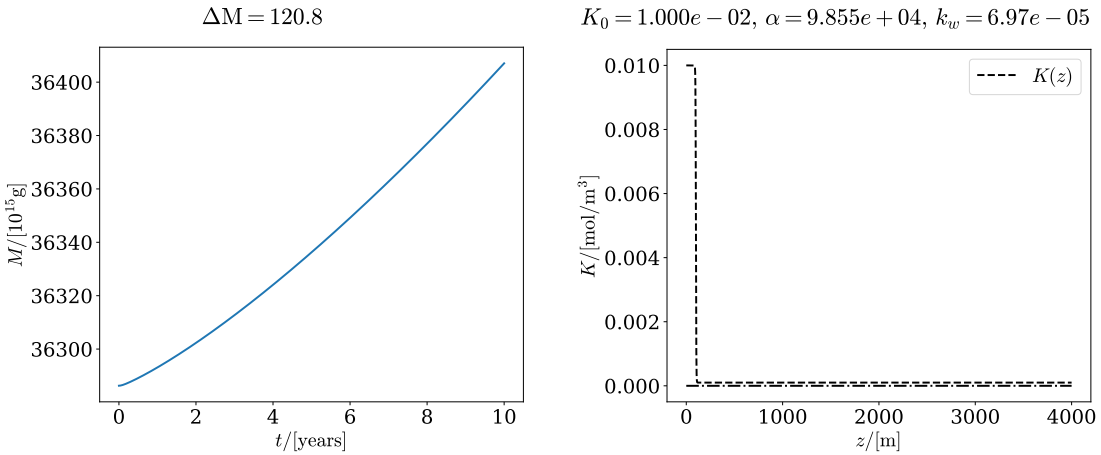

Figure 12: On the left, the total mass of carbon in the form of DIC, as a function of time. On the right the diffusivity as a functio of depth.

References

[1] Nicolas Gruber et al. "The oceanic sink for anthropogenic CO2 from 1994 to 2007". In: *Science* 363.6432 (2019), pp. 1193–1199. ISSN: 0036-8075. DOI: 10.1126/science.aau5153. eprint: https://science.sciencemag.org/content/363/6432/1193.full.pdf. URL: https://science.sciencemag.org/content/363/6432/1193.

[2] NTNU, Institutt for Fysikk. *Exercise 3, TFY4235 Computational Physics*. 2021.