Contents

Acknowledgements

My years at NTNU have passed by so fast and in a way it is hard to believe that I now have a degree and a thesis in my hands. There are many thank-you's to be said, but I would like to acknowledge those who have been particularly significant to me - for my engineering degree, my thesis and beyond.

My supervisor **Bjørn Holmedal** for your encouragement and continuous guiding through my work. I was very lucky to have a supervisor who cared a lot about my work and who was always available for my questions.

Karin Aslaksen and **Rune Rønvik** for being among the reasons for which I decided to study engineering. Thank you for making engineering available and more understandable to me when I was still at school.

The **Lycée Francais René Cassin** in Oslo for my primary and secondary education. For making me thrive at school for 12 years, and for interesting me in most subjects. Yet I would like to thank my physics and math teachers **Mme Cordon** and **Mr Magagnin** in particular.

My dear friends **Kine Karlsen**, **Jørgen Westrum Thorsen**, **Lotte Skolem**, **Eivind Berstad**, **Eirik Vikum**, **Elin Kathrine Saunes**, **Karoline Fossland** and **Stina Låstad**, for some of my best time in Trondheim. I was lucky and so happy to work with such a dedicated, brilliant and hilarious dream team, and I still am to have you as my friends. Thank you Kine for recruiting me.

To the **MTNANO class of 2008**, for making my last year at NTNU a memorable one, and for making me feel as a part of your great class. Every lunch and dinner at SiT, every laugh and every discussion has been enriching and an enjoyable break from the thesis.

Oda Toreskås, **Ida Hjorth**, **Eirik Gjelsvik Medbø**, **Reidun Dahl Schlanbusch**, **Marianne Sandvold**, **Inthu Rathnavel** and **Ingrid Hallsteinsen**, for every moment I've spent with you during these years in Trondheim. If looking back at these years will evoke so many joyful memories, I know it is because I have been so lucky to have some of my best friends around me every day. Thank you Marianne and Oda for being there during spring 2008, when it was particularly needed.

Anna Cécile Cole, my dearest friend. "L'amitié se confirme avec le temps."

Mamie Annie, my greatest role model, thank you for believing so much in me for as long as I can remember.

My parents and two sisters, for everything.

Abstract

The goal of this project was to implement a model to describe the evolution of precipitate distribution during precipitation that could be used for all aspect ratios of the precipitate needles. The first step consists of solving the diffusion equation in cartesian coordinates using the finite elements method to calculate the concentration at each point of the matrix. The concentration gradient at the interface is used to calculate the flux through the cuboid/matrix interface, which is in turn related to the growth/dissolution rate through mass conservation. However the diffusion equation becomes an unsolvable singular equation at some of the boundary conditions, because some of the terms of the diffusion equation become equal to zero. An approximation was made to find a solution, and the concentration profile through the transformed volume was calculated. The flux was also calculated for a number of different aspect ratios of the needles and described qualitatively.

1 Introduction

Heat treatment of aluminium alloys is used to enhance the hardness properties of the alloy. Precipitates form in the alloy during the heat treatment that inhibit the motion of dislocations through the material. The microstructural morphology of the precipitates is governed by the interfacial and strain energies of the precipitate/matrix system [**?**]. The precipitates that are mainly responsible for the enhancement of the hardness properties in Al 6xxx alloys, β'', are cuboid needles. As the Al 6xxx alloys are widely used it is of great interest to desbribe the precipitation process as accurately as posssible. Precipitate growth takes place by diffusion. In this project the goal was to implement a model to predict the evolution of precipitate distribution during precipitation in aluminium alloys, that could be used for all aspect ratios of the precipitate needles. The first step of the model is to solve the diffusion equation at each point of the transformed volume, to find the concentration profile through the volume. This is done by solving the diffusion equation numerically using the finite elements method. The concentration gradient at each point of the interface of the cuboid is then calculated to find the flux through each point. The flux is in turn related to the growth or dissolution rate of the precipitate through mass conservation, and the growth/dissolution rate describes the evolution of precipitate distribution. However the diffusion equation becomes an unsolvable singular equation at some of the boundary conditions, because some of the terms of the diffusion equation become equal to zero. An approximation was made to find a solution, and the concentration profile through the transformed volume was calculated. The flux was also calculated for a number of different aspect ratios of the needles and described qualitatively.

2 Theory

We want to study the evolution of precipitate distribution during precipitation. The evolution of precipitate distribution is given by:

$$\frac{dN}{dt} + v\frac{dN}{dr} = 0 \tag{1}$$

2.1 Finite difference approximations to derivatives

If a function U and its derivatives are single-valued, finite and continuous functions of x, then by Taylor's theorem [**?**],

$$U(x+h) = U(x) + hU'(x) + \frac{1}{2}h^2U''(x) + \frac{1}{6}h^3U'''(x) + ... \tag{2}$$

$$U(x-h) = U(x) - hU'(x) + \frac{1}{2}h^2U''(x) - \frac{1}{6}h^3U'''(x) + ... \tag{3}$$

Addition of these expansions gives

$$U(x+h) + U(x-h) = 2U(x) + h^2U''(x) + O(h^4)$$

where $O(h^4)$ denotes terms containing fourth and higher powers of h.
Subtracting these two expansions gives:

$$U(x+h) - U(x-h) = 2hU'(x) + O(h^3)$$

where $O(h^3)$ denotes terms containing third and higher powers of h.
Assuming that $O(h^4)$ and $O(h^3)$ are negligible in comparison with lower powers of h it follows that

$$U''(x) \simeq \frac{1}{h^2}\left(U(x+h) - 2U(x) + U(x+h)\right)$$

with an error order of h^3 , and that

$$U'(x) = \left(\frac{dU}{dx}\right) \simeq \frac{1}{2h}\left(U(x+h) - U(x-h)\right)$$

with an error of order h^2. These equations are called central-difference approximations. By using only equation 2 or 3 we can also derive a forward-difference formula

$$U'(x) \simeq \frac{1}{h}\left(U(x+h) - U(x)\right) \tag{4}$$

or a backward-difference formula

$$U'(x) \simeq \frac{1}{h} \left(U(x) - U(x-h) \right) \tag{5}$$

The leading errors in these forward and backward formulae are both $O(h)$, which is substantially higher than the leading error in the central difference formula.

2.2 Diffusional Transformations in metals

The precipitation reaction modelled in this project is a diffusional transformation induced by a change of temperature in a alloy of fixed bulk composition. Understanding the mechanisms behind the precipitation reaction is important to be able to model it. Consider for instance the binary phase diagram presented in figure 1 below. This could be the phase diagram for the Al-Mg system, or the Al-Si system for example. The α phase to the far left and the β phase to the far right represent supersaturated solid solutions, whereas the phase in between them is composed of both α and β phases. We want to study the rection that occurs when quenching the alloy from a high temperature, where it is in a supersaturated solid solution, to a temperature below the solvus line. The red arrow in the figure illustrates this process.

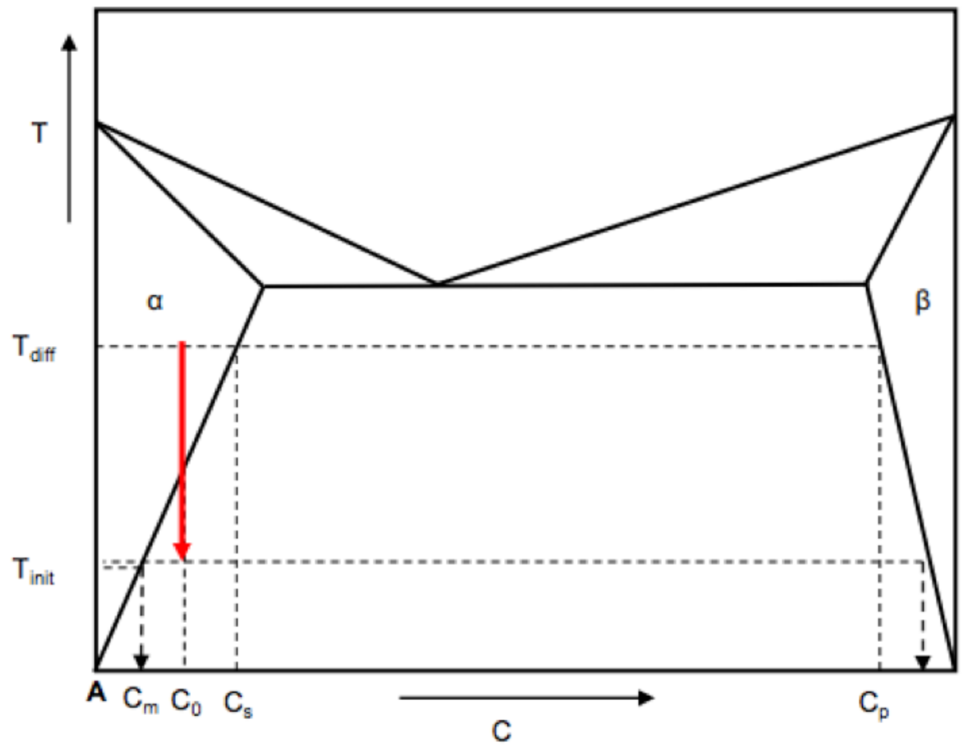


Figure 1: Quenching / growth

Homogeneous nucleation in pure metals:
Consider figure 2 below.

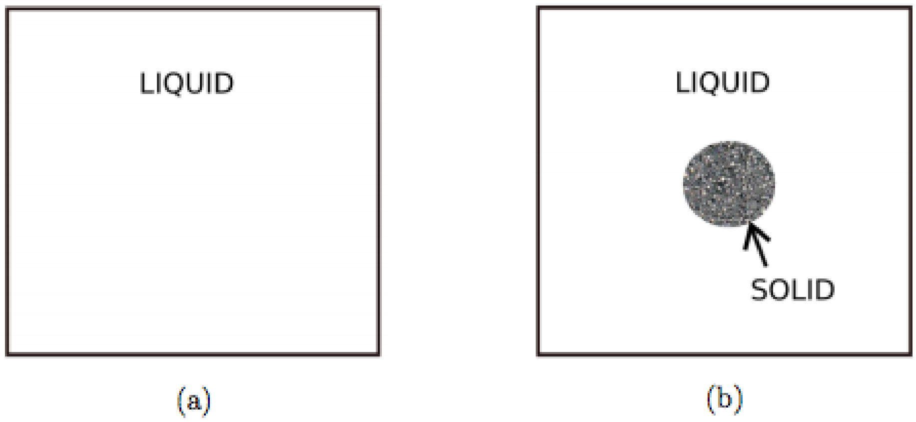


Figure 2: Homogeneous nucleation in a pure solid. Based on [?]

Let us define G_v^L as the free energy per unit volume of liquid, and G_v^s as the free energy per unit volume of solid. V_L and V_S are defined as the volumes of liquid and solid in system b). A_{SL} is defined as the solid/liquid interfacial area and γ_{SL} as the solid/liquid interfacial free energy. We can calculate the free energy of system b) [?]:

$$G_2 = V_l G_V^L + V_S G_V^S + A_{SL} * \gamma_{SL}$$

and the free energy of system a):

$$G_1 = (V_L + V_S) * G_V^L$$

Hence the fomation of a solid volume in a liquid, as in b) , results in a free energy change :

$$\Delta G = G_2 - G_1 = -V_S \Delta G_V + A_{SL} \gamma_{SL} \tag{6}$$

and

$$\Delta G_v = G_v^L - G_v^S \tag{7}$$

For and undercooling ΔT, ΔG_V is given by

$$\Delta G_V = \frac{L_V \Delta T}{T_M} \tag{8}$$

where L_v is the latent heat of fusion per unit volume. Below the melting temperature ΔT is positive, thus $-V_s \Delta G_V$ is negative. Therefore the free energy per unit volume associated with the formation of a small volume has a negative contribution due to the lower free energy of the bulk volume, and a positive contribution due to the creation of a solid/liquid interface. This interfacial free energy can be minimized by the correct choice of particle shape. If γ_{SL} is isotropic, the shape that will minimize the interfacial free energy is a sphere of radius r . The particles that form in Al 6xxx alloys are not spheres, but needles, hence the interfacial energy is not isotropic, and the needles represent the shape minimizing the interfacial free energy.

Precipitation in age-hardenable alloys

The theory of nucleation presented in previous section forms the basis for understanding the transformations in age-hardenable alloys, and we will here look at the Al-Mg-Si system. If the alloy is heated to a temperature above the solvus line the solute atoms will dissolve into the Al matrix. By quenching the sample rapidly into water, there is no time for a transformation to occur and the metal will remain in supersaturated solid solution. Now if the alloy is held at room temperature for a period of time, precipitation of Mg- and Si-rich coherent GP-zones will occur. Let us try to understand why. The β phase is incoherent with the α phase, and therefore has a high interfacial energy term. The GP zones, on the other hand, are coherent with the α phase and therefore they have a much lower interfacial energy term, and thus a lower activation barrier. That is the reason for why the first precipitate to nucleate is not β but GP zones.

Precipitation of transition phases usually follow precipitation of GP zones. They form because, like GP zones, they have a lower activation barrier of formation than the equilibrium phase. Each new transition phase is more stable than the preceding one and therefore has lower free energy. The equilibrium phase has lowest free energy and this is the driving force for the transfomations. The lower energy barriers are achieved because each new transition phase is partly coherent with the previous one, which results in a low interfacial energy contribution to the free energy (which means a lower activation barrier). The equilibrium phase is incoherent with the matrix and therefore has a very high interfacial energy contribution to the free energy (which means a high activation barrier of formation)[**?**].

Precipitation sequence in the Al-Mg-Si system

The precipitation sequence of the Al-Mg-Si system is given by [**?**] :

Supersaturated solid solution (SSSS) $\rightarrow$ Atomic clusters $\rightarrow$ GP zones $\rightarrow$ β'' $\rightarrow$ β', Ui, U2, B' $\rightarrow$ β/Si

β'' precipitates

The β'' precipitates (sometimes called GP-II zones) are semicoherent, larger needles with a typical size of 4x4x50 nm^3. Their unit cell is monoclinic, space group C2/m, with unit cell parameters a=1.516 nm, b=0.405 nm, c=0.674 and β = 105.3 degrees . The needles are elongated along the $\langle 100 \rangle$ direction of aluminium, and their orientation relationships can be expressed as $(010)_{\beta''}//100_{Al}$, $[100]_{\beta''}//\langle 310 \rangle_{Al}$, and $[100]_{\beta''}//\langle 230 \rangle_{Al}$. These precipitates are the main responsible for the hardening effect in Al-Mg-Si alloys.

2.3 Gibbs Thomson effects in phase transformations

Nucleation and growth during precipitation depend strongly on interfacial effects [?]. The corrected solubility limit X_{eq}, of B atoms in a α matrix in equilibrium with β phase occurring as spherical particles of radius r is given as [?]:

$$X^{\alpha}_{eq} = X^{\alpha}_{eq\infty} exp\left(\frac{2\gamma V_m}{rRT}\right) \tag{9}$$

where T is the temperature, γ the surface energy, R the molar gas constant and V_m is the molar volume.

2.4 The model

The model used to predict the evolution of precipitates in the alloy is characterized by:

- The continuous time evolution of the particle distribution is considered in terms of discreet time steps
- A nucleation model that calculates the number of stable nuclei that form at each time step, using classical nucleation theory
- The mean matrix concentration is calculated and updated at each time step, using mass balance
- A rate law calculating the growth or dissolution rate of the particles, assuming diffusion-controlled growth/dissolution
- The particle size distribution and volume fraction is updated at each time step
- A continuity equation

2.4.1 Nucleation rate

Classical nucleation theory is used to predict the number of stable nuclei that form at each time step. The steady-state nucleation rate j can be expressed as [**?**]

$$j = j_0 exp\left(\frac{\Delta G^*_{het}}{RT}\right) exp\left(-\frac{Q_d}{RT}\right) \tag{10}$$

And the energy barrier for heterogeneous nucleation is given by the following equation [**?**]

$$\Delta G^*_{het} = \frac{(A_0)^3}{(RT)^2[ln(\bar{C}/C_e)]^2} \tag{11}$$

Where $\hat{C}$ is the mean solute content in the matrix, and C_e is the equilibrium solute content at the particle/matrix interface (given by the phase diagram). Because $\Delta G{*}_{het}$ is defined on a mole basis, the parameter A_0 has the same dimension as the activation energy for nucleation. The nucleation rate can be expressed as a combination of equation 10 and 11:

$$j = j_0 exp[\left(-\frac{A_0^3}{RT}\right)\left(\frac{1}{ln(\bar{C}/C_e)}\right)] * exp\left(-\frac{Q_d}{RT}\right) \tag{12}$$

This equation implies that the nucleation rate j starts to drop off when the matrix is depleted with repect to solute. The nucleation process eventually stops when $\bar{C}$ approaches C_e.

2.4.2 Precipitate growth

When a spherical particle of radius r and solute concentration C_p is embedded in a super-satursated solid solution of a mean concentration $\bar{C}$, it will either dissolve or grow, depending on whether the particle/matrix interface concentration C_i exceeds $\bar{C}$ or not. The rate at which this occurs can be expressed as [**?**]

$$v = \frac{dr}{dt} = \frac{\bar{C} - C_i}{C_p - C_i}\frac{D}{r} \tag{13}$$

The interface concentration C_i is, in turn, related to the equilibrium value C_e through the Gibbs-Thomson equation, equation 9

$$C_i = C_e exp\left(\frac{2\sigma V_m}{rRT}\right)$$

where σ is the particle-matrix interface energy, and V_m is the molar volume of the particle.

To illustrate this we will derive the growth rate in 1 D.

Growth rate - 1D

Consider figure 3 below. A slab of solute-rich precipitate has grown from zero thickness. Since the solute concentration in the precipitate C_β is higher than the concentration of solute in the bulk, C_0, the matrix is depleted with solute at the interface. Also, since equilibrium at the interface can be assumed, the concentration of solute in the matrix at the interface will be the equilibrium concentration C_e. The flux of B through a unit area in time dt is given by $D(dC/dx)dt$, where D is the interdiffusion coefficient. For a unit area of interface to advance dx, a volume $1 * dx$ of material must be converted from α to β . $(C_\beta - C_e)dx$ moles of B must be supplied by diffusion through α . Using mass balance, these two quantities can be equated, which yields

$$(C_\beta - C_e)dx = D\frac{\mathrm{d}C}{dx}dt \tag{14}$$

$$v = \frac{dx}{dt} = \frac{D}{C_\beta - C_e}\frac{dC}{dx} \tag{15}$$

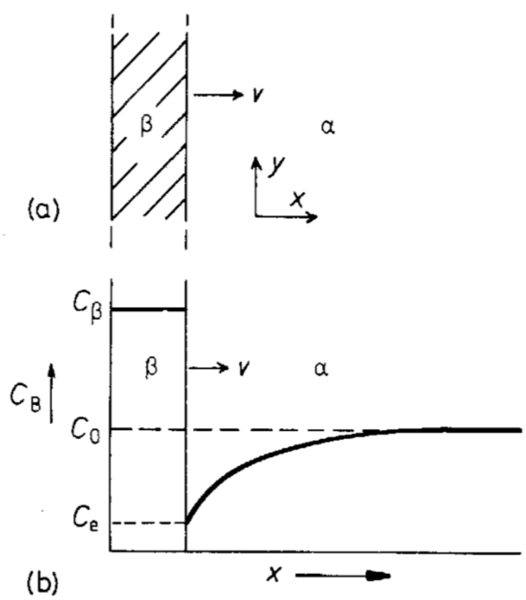


Figure 3: Growth of a particle in 1 D

2.4.3 Mean solute concentration in matrix - Continuity equation

Relating the amount of solute in the particles to the amount of solute drained from the matrix through mass balance, the mean solute concentration in the matrix can be expressed as

$$\bar{C} = C_0 - (C_p - \bar{C}) \int \frac{4}{3}\pi r^3 \varphi dr \tag{16}$$

Where φ is the size distribution function. The mean solute concentration is calculated at each time step and used to calculate the growth or dissolution rate from equation 13.

2.4.4 Evolution of particle distribution

The particle distribution is divided into a series of small "radius" elements, each with a size Δr. As in the diffusion problem in subsection above, the growth or dissolution of particles that occur during a small time increment Δt is also a flux of matter in or out the particles. Let J denote the particle flux, while N is the number density of particles within Δr. The evolution of particle distribution can be expressed as

$$\frac{\partial N}{\partial t} = -\frac{\partial J}{\partial r} + S \tag{17}$$

through mass conservation, where S describes the formation of new particles at each time step. The particle flux can be expressed as

$$J = Nv \tag{18}$$

where v is the growth or dissolution rate of the particle, given by equation 13. Inserting equation 18 into equation 17, the evolution of particle distribution is given by the following equation

$$\frac{\partial N}{\partial t} = -\frac{\partial (Nv)}{\partial r} + S \tag{19}$$

2.5 Diffusion problem

2.5.1 Poisson's equation

Poissons's equation is an elliptic partial differential equation, often written as $\nabla^2\varphi = f$. In three-dimensional cartesian coordinates this takes the form:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\varphi(x,y,z) = f(x,y,z)$$

When f is equal to zero this equation becomes the Laplace's equation

$$\nabla^2\varphi = 0$$

The Poisson's equation is also known as the diffusion equation.

2.5.2 Diffusion problem for spherical particles

$$\nabla^2\varphi = 0$$

The diffusion equation in polar coordinates is given by:

$$\frac{d^2\varphi}{dr^2} + \frac{2}{r}\frac{d\varphi}{dr} = 0$$

We define $\hat{r}$:

$$\hat{r} = \frac{R}{r}\ ,\ \frac{d\hat{r}}{dr} = -\frac{R}{r^2} = -\frac{\hat{r}^2}{R}$$

Thus

$$\frac{\mathrm{d}\phi}{\mathrm{d}r} = \frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}}\frac{\mathrm{d}\hat{r}}{\mathrm{d}r} = -\frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}}\frac{\hat{r}^2}{R}$$

And

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}r^2} = -\frac{\mathrm{d}}{\mathrm{d}r}\left(\frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}}\frac{\hat{r}^2}{R}\right) = \left(-\hat{r}^2\frac{\mathrm{d}^2\phi}{\mathrm{d}\hat{r}^2}(-\hat{r}^2) - 2\hat{r}\frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}}(-\hat{r}^2)\right)\frac{1}{R^2}$$

Insert into the diffusion equation:

$$\left(+2\hat{r}^3\frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}} + \hat{r}^4\frac{\mathrm{d}^2\phi}{\mathrm{d}\hat{r}^2}\right) + 2\hat{r}(-\hat{r}^2)\frac{\mathrm{d}\phi}{\mathrm{d}\hat{r}} = 0$$

$$\hat{r}^4\frac{d^2\varphi}{d\hat{r}} = 0$$

This equation has an analytical solution:

$$\phi = \hat{r} = \frac{R}{r}$$

The boundary conditions are given by

$$\phi(\hat{r} = 0) = 0 \ \text{ and } \ \phi(\hat{r} = 1) = 1$$

2.5.3 Diffusion problem for cuboid particles

Consider the cuboid in figure 4

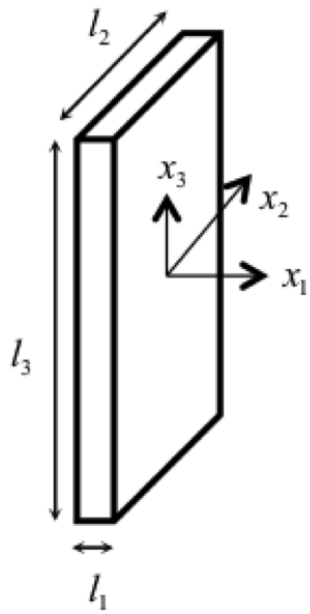


Figure 4: Cuboid

The diffusion problem is given as

$$\frac{dc}{dt} = D\frac{\partial^2 c}{\partial x_i^2}$$

At steady-state

$$\frac{dc}{dt} = 0$$

Hence

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} = 0 \tag{20}$$

The boundary conditions are given by: $C = C_0$ at $x_i \to \infty$

$$c = \left\{ \begin{array}{lll} C_i^{(1)} & \text{at} & x = \pm\frac{1}{2}l_x, |y| \leq \frac{1}{2}l_y, |z| \leq \frac{1}{2}l_z \\ C_i^{(2)} & \text{at} & y = \pm\frac{1}{2}l_y, |x| \leq \frac{1}{2}l_x, |z| \leq \frac{1}{2}l_z \\ C_i^{(3)} & \text{at} & z = \pm\frac{1}{2}l_z, |x| \leq \frac{1}{2}l_x, |y| \leq \frac{1}{2}l_y \end{array} \right\}$$

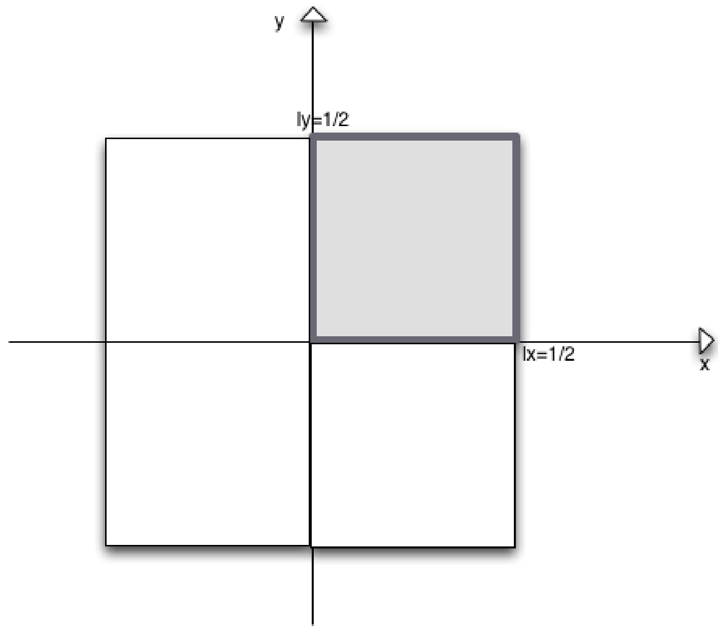


Figure 5: The cuboid included in our model represents only 1/8 of the actual volume of the cuboid This is represented in 2D in this figure.

Initial condition: $c = C_0$ at $t = 0$
We define

$$\phi = \frac{C - C_0}{C_i - C_0} \tag{21}$$

Thus

$$C = (C_i - C_0)\phi + C_0 \tag{22}$$

Since this problem is linear, the general solution can be built from unit solutions for each type of surface.

$$\phi = \phi^{(1)} + \phi^{(2)} + \phi^{(3)}$$

The boundary condition for each such basis problem is

Right and left surface

$$\phi^{(1)} = 0 \quad at \quad x \; \rightarrow \; \infty$$

$$\phi^{(1)} = \left\{ \begin{array}{lll} 1 & \text{at} & x = \pm\frac{1}{2}l_x, y \leq \frac{1}{2}l_y, |z| \leq \frac{1}{2}l_z \\ 0 & \text{at} & y = \pm\frac{1}{2}l_y, |x| \leq \frac{1}{2}l_x, l_z \leq \frac{1}{2}l_z \\ 0 & \text{at} & z = \pm\frac{1}{2}l_z, |x| \leq \frac{1}{2}l_x, |l_y| \leq \frac{1}{2}l_y \end{array} \right\}$$

Front and back surfaces:

$$\phi^{(3)} = 0 \ \ at \ \ y \ \rightarrow \ \infty$$

$$\phi^{(2)} = \left\{ \begin{array}{lll} 0 & \text{at} & x = \pm\frac{1}{2}l_x, |y| \leq \frac{1}{2}l_y, |z| \leq \frac{1}{2}l_z \\ 1 & \text{at} & y = \pm\frac{1}{2}l_y, |x| \leq \frac{1}{2}l_x, |z| \leq \frac{1}{2}l_z \\ 0 & \text{at} & z = \pm\frac{1}{2}l_z, |x| \leq \frac{1}{2}l_x, |l_y| \leq \frac{1}{2}l_y \end{array} \right\}$$

Top and bottom surfaces:

$$\phi^{(3)} = 0 \ \ at \ \ z \ \rightarrow \ \infty$$

$$\phi^{(3)} = \left\{ \begin{array}{lll} 0 & \text{at} & x = \pm\frac{1}{2}l_x, |y| \leq \frac{1}{2}l_y, |z| \leq \frac{1}{2}l_z \\ 0 & \text{at} & y = \pm\frac{1}{2}l_y, |x| \leq \frac{1}{2}l_x, |z| \leq \frac{1}{2}l_z \\ 1 & \text{at} & z = \pm\frac{1}{2}l_z, |x| \leq \frac{1}{2}l_x, |y| \leq \frac{1}{2}l_y \end{array} \right\}$$

2.5.4 Flux

$$I = 8{*}\left(-\iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\,\mathrm{d}z - \iint_{x,z=0}^{l_x,l_y} \frac{\partial \phi}{\partial y}\bigg|_{y=l_y} \mathrm{d}x\,\mathrm{d}z - \iint_{x,y=l_x,l_z}^{\frac{1}{2}} \frac{\partial \phi}{\partial z}\bigg|_{z=l_z} \mathrm{d}x\,\mathrm{d}y \right) \tag{23}$$

Having obtained the basis solution the fluxes through the particle surfaces can also be compiled as:

$$i^{(x)} = -\iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(1)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\,\mathrm{d}z - \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(2)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\,\mathrm{d}z - \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(3)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\,\mathrm{d}z$$

$$i^{(y)} = -\iint_{x,z=0}^{l_x,l_z} \frac{\partial \phi^{(1)}}{\partial y}\bigg|_{y=l_y} \mathrm{d}x\,\mathrm{d}z - \iint_{x,z=0}^{l_x,l_z} \frac{\partial \phi^{(2)}}{\partial y}\bigg|_{y=l_y} \mathrm{d}x\,\mathrm{d}z - \iint_{x,z=0}^{l_x,l_z} \frac{\partial \phi^{(3)}}{\partial y}\bigg|_{y=l_y} \mathrm{d}x\,\mathrm{d}z$$

$$i^{(z)} = -\iint_{x,y=0}^{l_x,l_y} \frac{\partial \phi^{(1)}}{\partial z}\bigg|_{z=l_z} \mathrm{d}x\,\mathrm{d}y - \iint_{x,y=0}^{l_x,l_y} \frac{\partial \phi^{(2)}}{\partial z}\bigg|_{z=l_z} \mathrm{d}x\,\mathrm{d}y - \iint_{x,y=0}^{l_x,l_y} \frac{\partial \phi^{(3)}}{\partial z}\bigg|_{z=l_z} \mathrm{d}x\,\mathrm{d}y$$

So that the total flux through the particle surface is:

$$i = 8 * (i^{(x)} + i^{(y)} + i^{(z)})$$

Dimensionless variables

By using dimensionless variables, the solution is not specific to a particular problem, but general enough to be applied to all sets of problems.

$$\tilde{x} = \frac{x}{l_x}\ , \quad \tilde{y} = \frac{y}{l_x}\ , \ \tilde{z} = \frac{z}{l_x}$$
$$\tilde{l}_x = \frac{l_x}{l_x}\ , \quad \tilde{l}_y = \frac{l_y}{l_x}\ , \ \tilde{l}_z = \frac{l_z}{l_x}$$

3 Experimental

3.1 Calculating the equivalen flux through a sphere

In order to predict the flux through the interface of the cuboid we calculated it analytically for a sphere, as this will give us an idea of the order of magnitude of the flux through the cuboid. The flux I is given by

$$I = -A\frac{\partial C}{\partial r}\bigg|_{r=R} = -4\pi r^2 * (C_i - C_0)\frac{\partial \phi}{\partial r}$$

$$= -\frac{4\pi R^2 (C_i - C_0)}{\hat{r}^2}\frac{\partial \phi}{\partial \hat{r}}\frac{d\hat{r}}{dr} = 4\pi R(C_i - C_0)\frac{\partial \phi}{\partial \hat{r}}\bigg|_{\hat{r}=1}$$

As

$$\hat{r} = \frac{R}{r}, \quad C = (C_i - C_0)\frac{\partial \phi}{\partial r}, \quad \text{and} \quad \frac{\partial \phi}{\partial r} = \frac{\partial \phi}{\partial \hat{r}}\frac{d\hat{r}}{dr} = \frac{\partial \phi}{\partial \hat{r}} * \left(-\frac{\hat{r}^2}{R}\right)$$

The derivative at the interface is given as

$$\frac{\partial \phi}{\partial \hat{r}}\bigg|_{\hat{r}=1} = 1$$

Hence

$$I = R(C_i - C_0)\hat{i}, \quad \text{where} \quad \hat{i} = 4\pi\Big(\frac{3}{4\pi}\Big)^{1/3} \approx 7.8$$

Therefore we should expect a flux around 8 through the cuboid.

3.2 Transformation

In this project the goal is to solve the diffusion problem for cuboid particles, hence we must turn to cartesian coordinates. As described in previous sections, the diffusion equation for cuboid particle in three dimensions is given by the following equation

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\phi(x, y, z) = 0$$

Unlike the diffusion problem for spherical precipitates this equation does not have an analytical solution, hence it must be solved numerically. The goal is to model the concentration outside the cuboid, which is from a finite

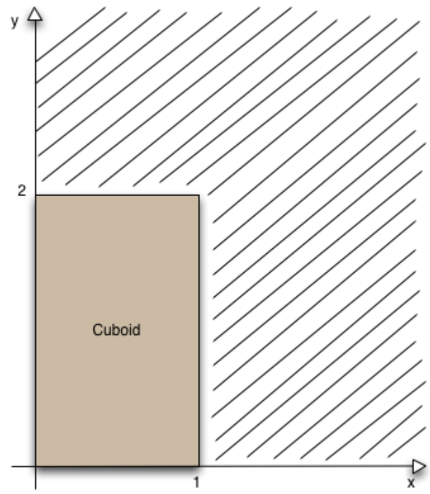


Figure 6: In the xy-plane figures the cuboid, and the area ouside it where we want to model the concentration. Hence we want to model the concentration from the cuboid/matrix interface to "infinity".

boundary (the cuboid interface) to infinity. This is not possible to model numerically, hence we need to introduce a transformation which makes us model/simulate from a finite boundary to another. Initially we used the tranformation given by

$$\hat{x} = \frac{1}{\tilde{x}} \ , \ \hat{y} = \frac{1}{\tilde{y}} \ , \ \hat{z} = \frac{1}{\tilde{z}} \tag{24}$$

Figure 7 illustrates this transfomation. Using this transformation follows that

$$\frac{d\hat{x}}{dx} = \frac{d\hat{x}}{d\tilde{x}}\frac{d\tilde{x}}{dx} = -\frac{1}{(\tilde{x})^2} * \frac{1}{l_x} = -\frac{\hat{x}^2}{l_x}$$
$$\frac{d\hat{y}}{dy} = \frac{d\hat{y}}{d\tilde{y}}\frac{d\tilde{y}}{dy} = -\frac{1}{(\tilde{y})^2} * \frac{1}{l_x} = -\frac{\hat{y}^2}{l_x}$$
$$\frac{d\hat{x}}{dz} = \frac{d\hat{z}}{d\tilde{z}}\frac{d\tilde{z}}{dz} = -\frac{1}{(\tilde{z})^2} * \frac{1}{l_x} = -\frac{\hat{z}^2}{l_x}$$

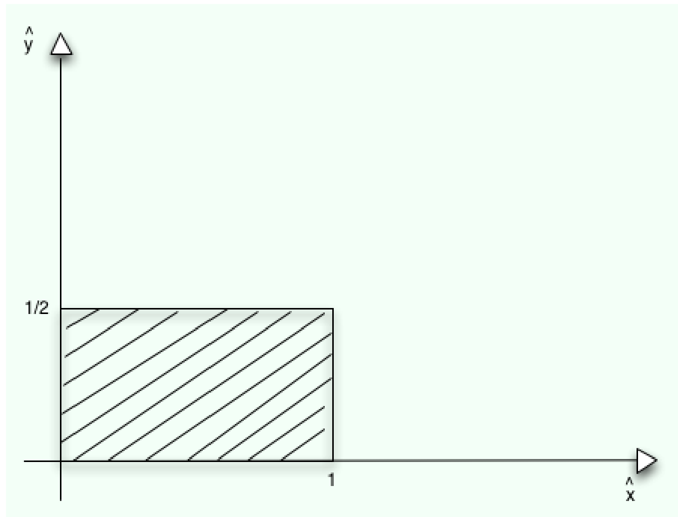


Figure 7: Transformed volume. We want the entire chopped area in figure 6, the area ouside the cuboid, to be included in the chopped area in this figure.

And therefore

$$\frac{d\phi}{dx} = \frac{d\phi}{d\hat{x}}\frac{d\hat{x}}{dx} = -\frac{\hat{x}^2}{l_x} * \frac{d\phi}{d\hat{x}}\ , \quad \frac{d\phi}{dy} = \frac{d\phi}{d\hat{y}}\frac{d\hat{y}}{dy} = -\frac{\hat{y}^2}{l_x} * \frac{d\phi}{d\hat{y}},$$

$$\frac{d\phi}{dz} = -\frac{\hat{z}^2}{l_x} * \frac{d\phi}{d\hat{z}}$$

The second-derivative becomes

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial\phi}{\partial x} = \frac{\partial}{\partial\hat{x}}\frac{\partial\hat{x}}{\partial x}\left(-\frac{\hat{x}^2}{l_x}\frac{\partial\phi}{\partial\hat{x}}\right) = -\frac{\hat{x}^2}{l_x}\left[-\frac{2\hat{x}}{l_x} * \frac{\partial\phi}{\partial\hat{x}} + \left(-\frac{\hat{x}^2}{l_x}\right) * \frac{\partial^2\phi}{\partial\hat{x}^2}\right]$$

$$= \frac{2\hat{x}^3}{l_x^2}\frac{\partial\phi}{\partial\hat{x}} + \frac{\hat{x}^4}{l_x^2}\frac{\partial^2\phi}{\partial\hat{x}^2}$$

Similarly

$$\frac{\partial^2 \phi}{\partial y^2} = \frac{\partial}{\partial y}\frac{\partial \phi}{\partial y} = \frac{\partial}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y}\left(-\frac{\hat{y}^2}{l_x}\frac{\partial \phi}{\partial \hat{y}}\right) = -\frac{\hat{y}^2}{l_x}\left[-\frac{2\hat{y}}{l_x} * \frac{\partial \phi}{\partial \hat{y}} + \left(-\frac{\hat{y}^2}{l_x}\right) * \frac{\partial^2 \phi}{\partial \hat{y}^2}\right]$$

$$= \frac{2\hat{y}^3}{l_x^2}\frac{\partial \phi}{\partial \hat{y}} + \frac{\hat{y}^4}{l_x^2}\frac{\partial^2 \phi}{\partial \hat{y}^2}$$

And

$$\frac{\partial^2 \phi}{\partial z^2} = \frac{\partial}{\partial z}\frac{\partial \phi}{\partial z} = \frac{\partial}{\partial \hat{z}}\frac{\partial \hat{z}}{\partial z}\left(-\frac{\hat{z}^2}{l_x}\frac{\partial \phi}{\partial \hat{z}}\right) = -\frac{\hat{z}^2}{l_x}\left[-\frac{2\hat{z}}{l_x} * \frac{\partial \phi}{\partial \hat{z}} + \left(-\frac{\hat{z}^2}{l_x}\right) * \frac{\partial^2 \phi}{\partial \hat{z}^2}\right]$$

$$= \frac{2\hat{z}^3}{l_x^2}\frac{\partial \phi}{\partial \hat{z}} + \frac{\hat{z}^4}{l_x^2}\frac{\partial^2 \phi}{\partial \hat{z}^2}$$

Inserting these derivatives into the diffusion equation becomes, it becomes

$$\nabla^2 \phi = \frac{1}{l_x^2}\left(\hat{x}^4\frac{\partial^2 \varphi}{\partial \hat{x}^2} + 2\frac{\partial \varphi}{\partial \hat{x}}\hat{x}^3 + \hat{x}^4\frac{\partial^2 \varphi}{\partial \hat{x}^2} + 2\frac{\partial \varphi}{\partial \hat{y}}\hat{y}^3 + \hat{z}^4\frac{\partial^2 \varphi}{\partial \hat{z}^2} + 2\frac{\partial \varphi}{\partial \hat{z}}\hat{z}^3\right) = 0 \tag{25}$$

Problem

However the transformation describes above turned out to be insufficient. Take a look at figure 8 below. To illustrate this point a figure is used in 2D, but the principle remains the same for 3D. The brown rectangle represents the cuboid, thus the area for which we want to model the concentration is all the area outside this rectangle, limited by the xy-axes. The grey/shaded rectangle represents the transformed volume, hence all the area ouside the cuboid should be included in the transformed volume after transformations. But using the transformation described above, only the chopped area is included in the transformed volume after the transformation. Look at the point A (2,0.1): A'(0.5, 10). A' does not belong to the transformed volume! Only the chopped area is included in the transformed volume. Thus we had to introduce a new transformation.

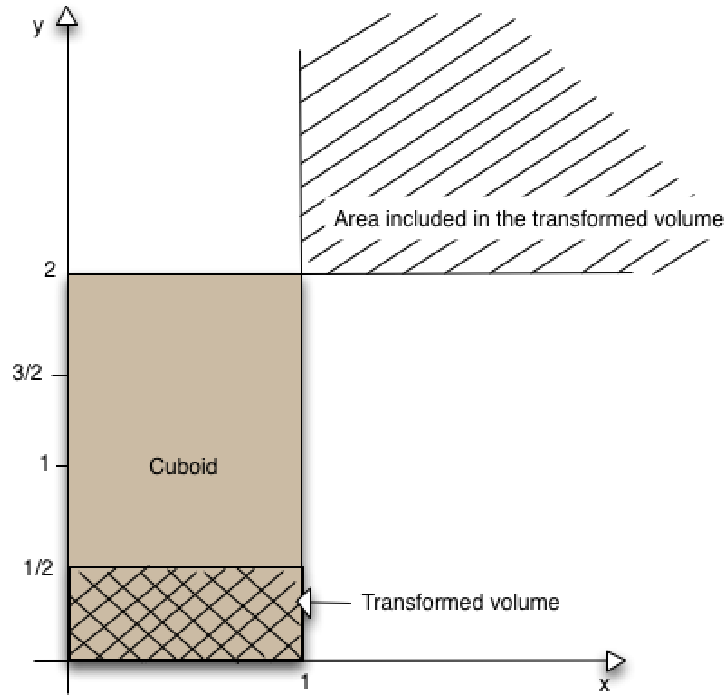


Figure 8: Insufficient transformation

3.3 New transformation

To overcome this problem we used a new transformation, which is given by:

$$\hat{x} = \frac{1}{\tilde{x} + \tilde{l_x}}, \quad \hat{y} = \frac{1}{\tilde{y} + \tilde{l_y}}, \quad \hat{z} = \frac{1}{\tilde{z} + \tilde{l_z}}$$

Using dimensionless variables

$$\hat{x} = \frac{1}{\tilde{x} + 1}, \quad \hat{y} = \frac{1}{\tilde{y} + \frac{l_y}{l_x}}, \quad \hat{z} = \frac{1}{\tilde{z} + \frac{l_z}{l_x}}$$

Therefore

$$\frac{\mathrm{d}\hat{x}}{\mathrm{d}\tilde{x}} = -\frac{1}{\left(\tilde{x} + \tilde{l}_x\right)^2} = -\hat{x}^2$$

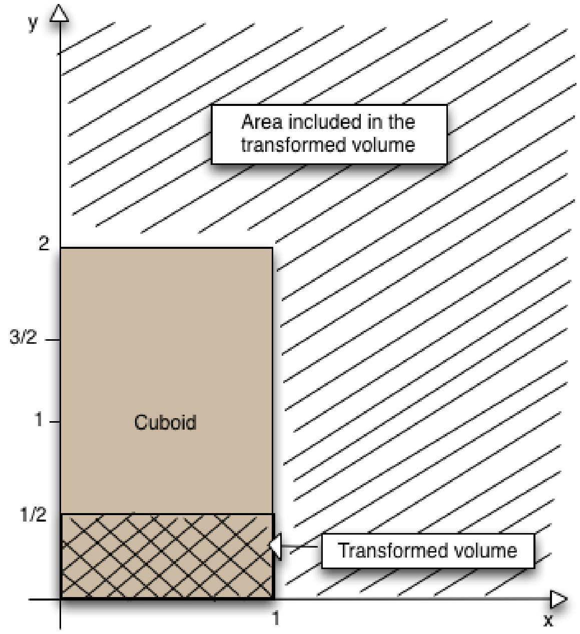


Figure 9: New transformation

and as with the previous transformation

$$\frac{d\phi}{dx} = -\frac{\hat{x}^2}{l_x}\frac{d\phi}{d\hat{x}}, \quad \frac{d\phi}{dy} = -\frac{\hat{y}^2}{l_x}\frac{d\phi}{d\hat{y}}, \quad \frac{d\phi}{dz} = -\frac{\hat{z}^2}{l_x}\frac{d\phi}{d\hat{z}}$$

$$\frac{\partial^2\phi}{\partial x^2} = \frac{1}{l_x^2}\left(\hat{x}^4\frac{\partial^2\varphi}{\partial\hat{x}^2} + 2\frac{\partial\varphi}{\partial\hat{x}}\hat{x}^3\right), \quad \frac{\partial^2\phi}{\partial y^2} = \frac{1}{l_x^2}\left(\hat{y}^4\frac{\partial^2\varphi}{\partial\hat{y}^2} + 2\frac{\partial\varphi}{\partial\hat{y}}\hat{y}^3\right), \quad \frac{\partial^2\phi}{\partial z^2} = \frac{1}{l_x^2}\left(\hat{z}^4\frac{\partial^2\varphi}{\partial\hat{z}^2} + 2\frac{\partial\varphi}{\partial\hat{z}}\hat{z}^3\right)$$

Thus we see that the diffusion equation remains the same as for the previous transformation and is given by:

$$\nabla^2\phi = \frac{1}{l_x^2}\left(\hat{x}^4\frac{\partial^2\varphi}{\partial\hat{x}^2} + 2\frac{\partial\varphi}{\partial\hat{x}}\hat{x}^3 + \hat{x}^4\frac{\partial^2\varphi}{\partial\hat{x}^2} + 2\frac{\partial\varphi}{\partial\hat{y}}\hat{y}^3 + \hat{z}^4\frac{\partial^2\varphi}{\partial\hat{z}^2} + 2\frac{\partial\varphi}{\partial\hat{z}}\hat{z}^3\right) = 0$$

Figure 10, 11 and 12 show the untransformed cuboid, the transformed volume sketched from side, and the transformed volume sketched from the front respectively. The figures below show the transformed volume, / transformed cuboid.

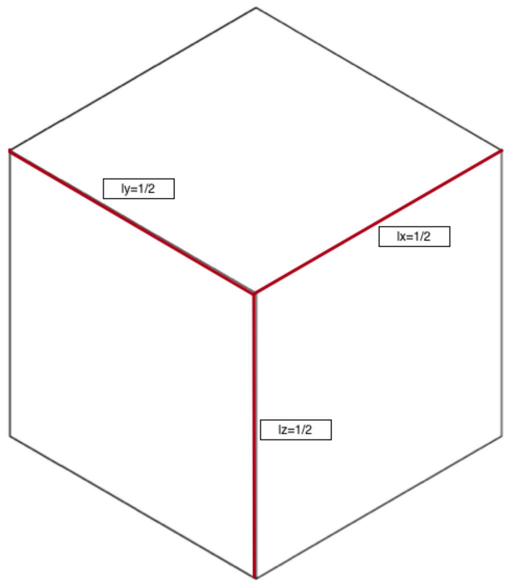


Figure 10: Illustration of the cuboid shape we used for the transformation. lx=ly=lz=1, thus we have a cube.

3.4 Finite elements

$$\frac{\partial \phi_{ijk}}{\partial \hat{x}_i} = \frac{\phi_{i+1jk} - \phi_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}}$$
$$\frac{\partial \phi_{ijk}}{\partial \hat{y}_j} = \frac{\phi_{ij+1k} - \phi_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}}$$
$$\frac{\partial \phi_{ijk}}{\partial \hat{z}_k} = \frac{\phi_{ijk+1} - \phi_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}}$$

And

$$\frac{\partial^2 \phi_{ijk}}{\partial \hat{x}_i^2} = \frac{\phi_{i+1jk} + \phi_{i-1jk} - 2\phi_{ijk}}{\frac{(\hat{x}_{i+1} - \hat{x}_{i-1})^2}{4}}$$
$$\frac{\partial^2 \phi_{ijk}}{\partial \hat{y}_j^2} = \frac{\phi_{ij+1k} + \phi_{ij-1k} - 2\phi_{ijk}}{\frac{(\hat{y}_{j+1} - \hat{y}_{j-1})^2}{4}}$$
$$\frac{\partial^2 \phi_{ijk}}{\partial \hat{z}_k^2} = \frac{\phi_{ijk+1} + \phi_{ijk-1} - 2\phi_{ijk}}{\frac{(\hat{z}_{k+1} - \hat{z}_{k-1})^2}{4}}$$

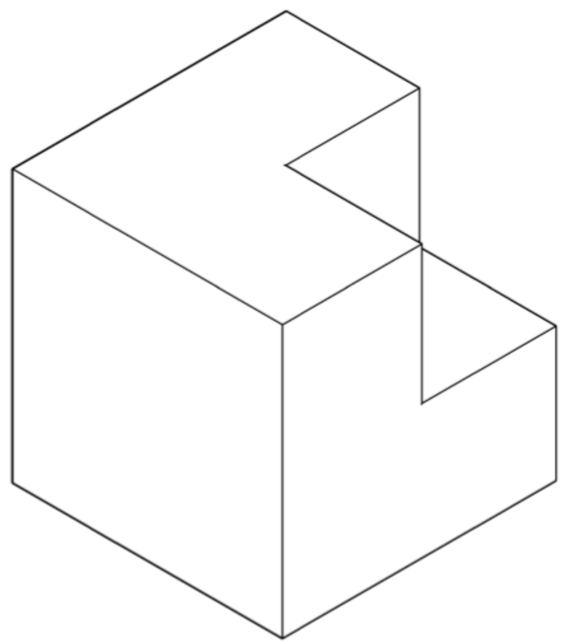


Figure 11: The transformed volume, sketched from the side in this illustration.

Insert into the diffusion equation:

$$\begin{aligned}\nabla^2\phi =&\frac{1}{l_x}\left(4\hat{x}_i^4\frac{\phi_{i+1jk}+\phi_{i-1jk}-2\phi_{ijk}}{(\hat{x}_{i+1}-\hat{x}_{i-1})^2}+2\hat{x}_i^3\frac{\phi_{i+1jk}-\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}+4\hat{y}_j^4\frac{\phi_{ij+1k}+\phi_{ij-1k}-2\phi_{ijk}}{(\hat{y}_{j+1}-\hat{y}_{j-1})^2}\right.\\&\left.+2\hat{y}_j^3\frac{\phi_{ij+1k}-\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}+4\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{(\hat{z}_{k+1}-\hat{z}_{k-1})^2}+2\hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}\right)=0\end{aligned}$$

Thus

$$\begin{aligned}\nabla^2\phi =&4\hat{x}_i^4\frac{\phi_{i+1jk}+\phi_{i-1jk}-2\phi_{ijk}}{(\hat{x}_{i+1}-\hat{x}_{i-1})^2}+2\hat{x}_i^3\frac{\phi_{i+1jk}-\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}+4\hat{y}_j^4\frac{\phi_{ij+1k}+\phi_{ij-1k}-2\phi_{ijk}}{(\hat{y}_{j+1}-\hat{y}_{j-1})^2}\\&+2\hat{y}_j^3\frac{\phi_{ij+1k}-\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}+4\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{(\hat{z}_{k+1}-\hat{z}_{k-1})^2}+2\hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}=0\end{aligned}$$

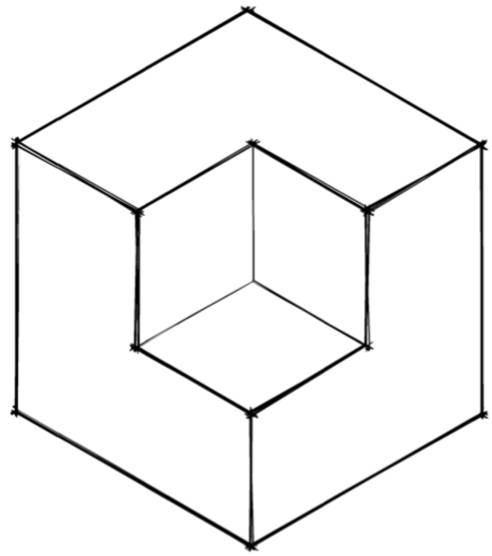


Figure 12: The transformed volume, sketched from the front in this illustration.

Which means

$$\phi_{ijk} = \frac{1}{\frac{4\hat{x}_i^4}{(\hat{x}_{i+1}-\hat{x}_{i-1})^2} + \frac{4\hat{y}_j^4}{(\hat{y}_{j+1}-\hat{y}_{j-1})^2} + \frac{4\hat{z}_k^4}{(\hat{z}_{k+1}-\hat{z}_{k-1})^2}} * \tag{26}$$

$$(2\hat{x}_i^4 \frac{\phi_{i+1jk} + \phi_{i-1jk}}{(\hat{x}_{i+1} - \hat{x}_{i-1})^2} + 2\hat{y}_j^4 \frac{\phi_{ij+1k} + \phi_{ij-1k}}{(\hat{y}_{j+1} - \hat{y}_{j-1})^2} + 2\hat{z}_k^4 \frac{\phi_{ijk+1} + \phi_{ijk-1}}{(\hat{z}_{k+1} - \hat{z}_{k-1})^2} \tag{27}$$

$$+ \hat{x}_i^3 \frac{\phi_{i+1jk} - \phi_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} + \hat{y}_j^3 \frac{\phi_{ij+1k} - \phi_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} + \hat{z}_k^3 \frac{\phi_{ijk+1} - \phi_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}}) \tag{28}$$

3.5 New problem

At the boundary condition where $x, y,$ or $z = \infty$ thus where $\hat{x}, \hat{y}$ or $\hat{z} = 0$, ϕ is given as

$$\phi = 0$$

But when $\hat{x} = 0$ the derivatives with respect to $\hat{y}$ and $\hat{z}$ are zero, and the diffusion equation becomes

$$\nabla^2\phi = 4\hat{x}_i^4 \frac{\phi_{i+1jk} + \phi_{i-1jk} - 2\phi_{ijk}}{(\hat{x}_{i+1} - \hat{x}_{i-1})^2} + 2\hat{x}_i^3 \frac{\phi_{i+1jk} - \phi_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} = 0$$

Similarly, when $\hat{y} =$ the diffusion equation becomes

$$\nabla^2\phi = 4\hat{y}_i^4 \frac{\phi_{ij+1k} + \phi_{ij-1k} - 2\phi_{ijk}}{(\hat{y}_{j+1} - \hat{y}_{j-1})^2} + 2\hat{y}_j^3 \frac{\phi_{ij+1k} - \phi_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} = 0$$

And when $\hat{z} = 0$

$$\nabla^2\phi = 4\hat{z}_k^4 \frac{\phi_{ijk+1} + \phi_{ijk-1} - 2\phi_{ijk}}{(\hat{z}_{k+1} - \hat{z}_{k-1})^2} + 2\hat{z}_k^3 \frac{\phi_{ijk+1} - \phi_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} = 0$$

These are singular equations, and the solutions are on the form

$$\phi(x) = -\frac{C_1}{x}$$
$$\phi(z) = -\frac{C_2}{y}$$
$$\phi(z) = -\frac{C_3}{z}$$

where C_1 , C_2 , C_3 are constants. These solutions can never be equal to zero, hence the diffusion equation is unsolvable at the boundary conditions where $\hat{x}, \hat{y}$ or $\hat{z} = 0$.

Hence we had to find a way where the diffusion equation did not become singular at the boundary condition. We chose to solve the equation from $\hat{x},\ \hat{y},\ \hat{z} = \frac{1}{nx},\ \frac{1}{ny},\ \frac{1}{nz}$ instead of 0 , where $i, j, k = 1$ instead of 0. This means that the solution we get is an approximation, as the condition where x, y or $z = \infty$ isn't modelled, but we consider $\hat{x},\ \hat{y},\ \hat{z} = \frac{1}{nx},\ \frac{1}{ny},\ \frac{1}{nz}$ to be small enough to be considered as the transformation of infinity.

The flux out the cuboid is approximatively the same as the flux out of a sphere with equivalent volume, calculated in section xx. In addition, the concentration profile out of the cuboid can be considered a sphere, when x, y and z are very large. Through Stoke's theorem, the flux of solute from a spherical precipitate is constant as the distance from the interface of the precipitate changes, if we assume no loss of material. Hence the flux at $\hat{x},\ \hat{y},\ \hat{z} = \frac{1}{nx},\ \frac{1}{ny},\ \frac{1}{nz}$ should be equivalent to the flux through the surfaces of the cuboid.

From section 3.1:

$$\left.\frac{\partial \phi}{\partial \hat{r}}\right|_{\hat{r}=1} = 1 \tag{29}$$

$$I = 4\pi R(C_i - C_0)\ ,\ R = \frac{I}{4\pi}$$

because $C_i - C_0 \approx 0$

$$\phi \rightarrow \frac{R}{r} \ \text{ and } \hat{r}^2 = \hat{x}^2 + \hat{y}^2 + \hat{z}^2 \tag{30}$$

$$\text{thus} \ \ \phi_{1jk} = \frac{I}{4\pi\sqrt{\hat{x(1)}^2 + \hat{y(j)}^2 + \hat{z(k)}^2}} \tag{31}$$

$$\phi_{i1k} = \frac{I}{4\pi\sqrt{\hat{x}(i)^2 + \hat{y}(1)^2 + \hat{z}(k)^2}} \tag{32}$$

$$\phi_{ij1} = \frac{I}{4\pi\sqrt{\hat{x}(i)^2 + \hat{y}(i)^2 + \hat{z}(1)^2}} \tag{33}$$

ϕ_{1jk} , ϕ_{i1k} , ϕ_{ij1} is calculated using equations 31, 32, and 33 at each each iteration, and inserted at these boundary conditions. ϕ is then calculated numerically at each point of the matrix using equation 28.

3.6 Calculating (ϕ) through the transformed volume

The concentration profile out of the cuboid was modelled by calculating ϕ_{ijk} numerically at each point of the matrix. The method used is characterized by the steps listed below:

- ϕ_{ijk} is set to zero at all th epoints in the matrix, so that all the points have a value
- The boundary conditions are set
- ϕ_{1jk} , ϕ_{i1k}, and $\phi ij1$ are calculated using equations 31 , 32, and 33
- ϕ_{ijk} is calculated numerically at all points of the matrix using equation (28), starting from $i,\ j,\ k = 2\ \rightarrow\ \hat{x},\ \hat{y},\ \hat{z} = 2/nx,\ 2/ny,\ 2/nz$.
- The old ϕ_{ijk} is substracted to the updated ϕ_{ijk} at each point, so that the error through the matrix is calculated
- Until the error gets smaller than a defined boundary, this is repeated.

Boundary conditions

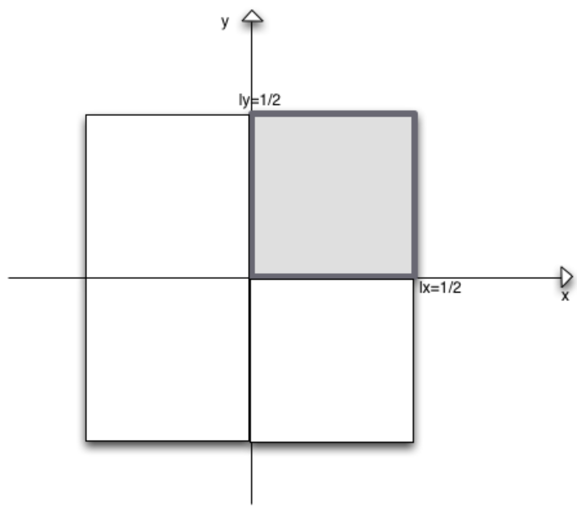


Figure 13: This illustration in two dimensions illustrates that only a fraction of the real cuboid is modelled in this model. In two dimensions this corresponds to 1/4 of the real cuboid, whereas in three dimensions this corresponds to 1/8 of the cuboid.

There are three boundary conditions in our problem. The two first correspond to the situation where x, y or z is at the surface of the cuboid, or very far away: $x,\ y,\ z = lx,\ ly,\ lz$ or $x,\ y,\ z = \infty$, whereas the last one sets a symmetry constraint alon the three axes $x = 0$, $y = 0,\ z = 0$. This boundary condition is illustrated in figure 13. This means that the first two constraints define the concentration at the corresponding surfaces in the transformed volume, whereas the last boundary condition sets a constraint inside equation 28.

- $x,\ y,$ or $z = \infty$.
  This boundary condition corresponds to the situation where $x,\ y,$ or $z = \infty$, in which case the concentration is zero.

$$\phi = 0 \begin{cases} \text{at } x = \infty,\ 0 \leq y, z \leq \infty \rightarrow \hat{x} = 0, & 0 \leq \hat{y}, \hat{z} \leq \frac{l_x}{l_y}, \frac{l_x}{l_z} \\ \text{at } y = \infty,\ 0 \leq x, z \leq \infty \rightarrow \hat{y} = 0, & 0 \leq \hat{x}, \hat{z} \leq \frac{l_x}{l_x}, \frac{l_x}{l_z} \\ \text{at } z = \infty,\ 0 \leq x, y \leq \infty \rightarrow \hat{z} = 0, & 0 \leq \hat{x}, \hat{y} \leq \frac{l_x}{l_x}, \frac{l_x}{l_y} \end{cases}$$

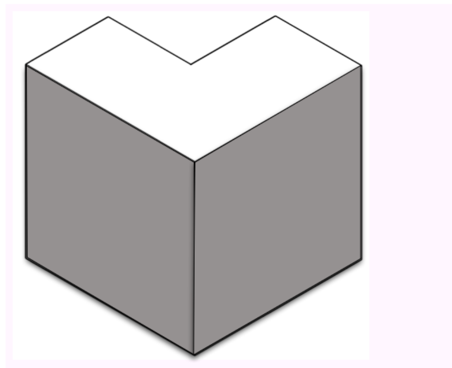


Figure 14: The transformed volume. The shaded surfaces are the surfaces where the concentration is zero.

- $x,\ y,$ or $z = lx\ ly\ lz$.
  This boundary condition corresponds to the situation where $x,\ y,$ or $z = lx\ ly\ lz$, in which case the concentration is one.

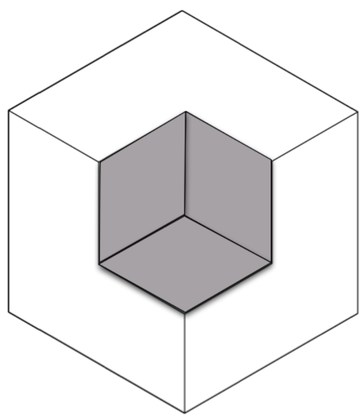


Figure 15: The transformed volume. The shaded surfaces represent a boundary condition and they correspond to the surfaces where the concentration is one.

$$\phi = 1 \begin{cases} \text{at } x = l_x,\ 0 \le y, z \le l_y, l_z \rightarrow \hat{x} = \frac{1}{2}, \quad \frac{l_x}{2l_y} \le \hat{y}, \hat{z} \le \frac{l_x}{l_y} \\ \text{at } y = l_y,\ 0 \le x, z \le l_x, l_z \rightarrow \hat{y} = \frac{l_x}{2l_y}, \quad \frac{l_x}{2l_z} \le \hat{x}, \hat{z} \le \frac{l_x}{l_z} \\ \text{at } z = l_z,\ 0 \le x, y \le l_x, l_y \rightarrow \hat{z} = \frac{l_x}{2l_z}, \quad \frac{l_x}{2l_y} \le \hat{x}, \hat{y} \le \frac{l_x}{l_y} \end{cases}$$

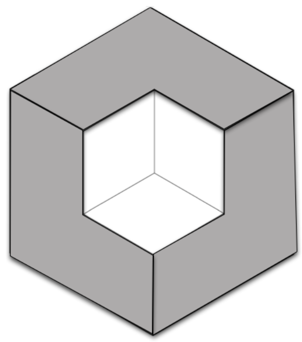


Figure 16: The transformed volume. The shaded surfaces represent a boundary condition, they correspond to the symmertry faces.

- Symmetry surfaces:
  This is the case that corresponds to the situation where x=0 or y=0 in figure 13. This is not a boundary condition that defines the concentration at these points, as the two previous ones, but it set a constraint on the derivatives in equation 28. These constaints will be included when calculating the concentration at each point in the matrix.

$$\left(\frac{\partial\phi}{\partial x}\right)\bigg|_{x=0} = 0 \quad\rightarrow\quad \left(\frac{\partial\phi}{\partial \hat{x}}\right)\bigg|_{\hat{x}=1} = \frac{\phi_{i+1jk} - \phi_{i-1jk}}{x_{i+1} - x_{i-1}} = 0 \;:\; \phi_{i+1jk} = \phi_{i-1jk}$$

$$\left(\frac{\partial\phi}{\partial y}\right)\bigg|_{y=0} = 0 \quad\rightarrow\quad \left(\frac{\partial\phi}{\partial \hat{y}}\right)\bigg|_{\hat{y}=1} = \frac{\phi_{ij+1k} - \phi_{ij-1k}}{y_{j+1} - y_{j-1}} = 0 \;:\; \phi_{ij+1k} = \phi_{ij-1k}$$

$$\left(\frac{\partial\phi}{\partial z}\right)\bigg|_{z=0} = 0 \quad\rightarrow\quad \left(\frac{\partial\phi}{\partial \hat{z}}\right)\bigg|_{\hat{z}=1} = \frac{\phi_{ijk+1} - \phi_{ijk-1}}{z_{k+1} - z_{k-1}} = 0 \;:\; \phi_{ijk+1} = \phi_{ijk-1}$$

- Edges between two symmetry surfaces:
  This is a special case in the symmetry face boundary conditions. It corresponds to the edge between to symmetry faces, meaning that there are two constraints on the derivatives in equation 28.

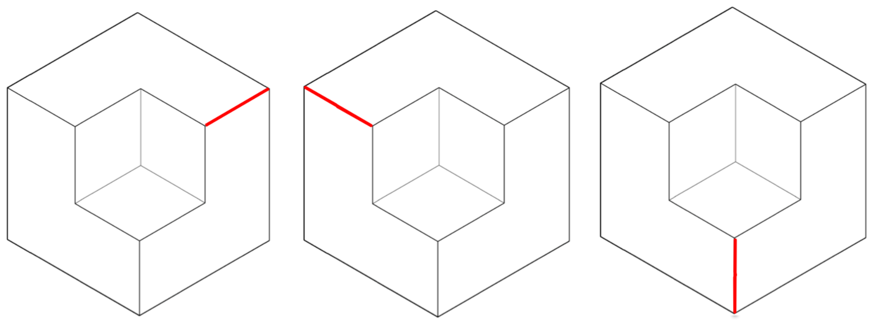


Figure 17: The transformed volume. The highlighted edges correspond to boundary conditions, they correspond to the edges between two symmetry surfaces.

$$\left(\frac{\partial\phi}{\partial x}\right)\Bigg|_{x=0}, \left(\frac{\partial\phi}{\partial y}\right)\Bigg|_{y=0} = 0 \;\rightarrow\; \left(\frac{\partial\phi}{\partial \hat{x}}\right)\Bigg|_{\hat{x}=1}, \left(\frac{\partial\phi}{\partial \hat{y}}\right)\Bigg|_{\hat{y}=1} = 0$$
$$: \phi_{i+1jk} = \phi_{i-1jk},\; \phi_{ij+1k} = \phi_{ij-1k}$$

$$\left(\frac{\partial\phi}{\partial y}\right)\Bigg|_{y=0}, \left(\frac{\partial\phi}{\partial z}\right)\Bigg|_{z=0} = 0 \;\rightarrow\; \left(\frac{\partial\phi}{\partial \hat{y}}\right)\Bigg|_{\hat{y}=1}, \left(\frac{\partial\phi}{\partial \hat{z}}\right)\Bigg|_{\hat{z}=1} = 0$$
$$: \phi_{ij+1k} = \phi_{ij-1k},\; \phi_{ijk+1} = \phi_{ijk+1}$$

$$\left(\frac{\partial\phi}{\partial x}\right)\Bigg|_{x=0}, \left(\frac{\partial\phi}{\partial z}\right)\Bigg|_{z=0} = 0 \;\rightarrow\; \left(\frac{\partial\phi}{\partial \hat{x}}\right)\Bigg|_{\hat{x}=1} = 0, \left(\frac{\partial\phi}{\partial \hat{z}}\right)\Bigg|_{\hat{z}=1} = 0$$
$$: \phi_{i+1jk} = \phi_{i-1jk},\; \phi_{ijk+1} = \phi_{ijk-1}$$

Calculating ϕ_{ijk} at all points in the matrix

ϕ_{ijk} is then calculated at every point of the transformed volume using equation 28.

General solution

Below figures an extract from the code that shows how ϕ is calculated through the surfaces. It is extracted from the code of my supervisor Bjørn Holmedal, as my code did not give correct results.

```
C Beregner konsentrasjon i det transformerte volumet
c indre pkt:
     do i=2,nx-1
       do j=2,ny-1
         do k=2,nz/2-1
           tmp=C(i,j,k)
c
           tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
           tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
           C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
           tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
           tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
           C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
           tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
           tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
           C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
           tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
           tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
           tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
           C(i,j,k) = C(i,j,k)/tmp1
c
c
           err=err+abs(C(i,j,k)-tmp)
           errmax=max(errmax,abs(C(i,j,k)-tmp))
         enddo
       enddo
     enddo


     do i=2,nx-1
       do j=2,ny/2-1
         do k=nz/2,nz-1
           tmp=C(i,j,k)
c
           tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
           tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
           C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
           tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
           tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
           C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
           tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
           tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
           C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
```

```
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo
    enddo

    do i=2,nx/2-1
      do j=ny/2,ny-1
        do k=nz/2,nz-1
          tmp=C(i,j,k)
c
          tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
          tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
          tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
          tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
          tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo
    enddo
```

At the symmetry surfaces

These points had to be treated separately, as there is a constraint on on the derivatives in equation (28), namely:

$$\phi_{i+1jk} = \phi_{i-1jk} \text{ when } i = nx$$

$$\phi_{ij+1k} = \phi_{ij-1k} \text{ when } j = ny$$

$$\phi_{i+1jk} = \phi_{i-1jk} \text{ when } k = nz$$

For each of these cases, this constraint had to be specified before calculating ϕ.

```
c Symmetriflatene:
c    dC/dx = 0:
     i=nx
     do j=2,ny-1
       do k=2,nz/2-1
         C(i+1,j,k)=C(i-1,j,k)
         tmp=C(i,j,k)
c
         tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
         tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
         C(i,j,k) =   x(i)**3*tmp1/(x(i+1)-x(i-1))
c
         tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
         tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
         C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
         tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
         tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
         C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
         tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
         tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
         tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
         C(i,j,k) = C(i,j,k)/tmp1
c
         err=err+abs(C(i,j,k)-tmp)
         errmax=max(errmax,abs(C(i,j,k)-tmp))
       enddo
     enddo
```

At the edges between two symmetry surfaces

These points also had to be treated separately, as there are two constratints on the derivatives of equation 28, namely

$$\phi_{i+1jk} = \phi_{i-1jk},\ \phi_{ij+1k} = \phi_{ij-1k} \text{ when } i = nx \text{ and } j = ny$$

$$\phi_{i+1jk} = \phi_{i-1jk},\ \phi_{ijk+1} = \phi_{ijk-1} \text{ when } i = nx \text{ and } k = nz$$

$$\phi_{ij+1k} = \phi_{ij-1k},\ \phi_{ijk+1} = \phi_{ijk-1} \text{ when } j = ny \text{ and } k = nz$$

Below figures an extract from the code that shows how ϕ is calculated through the edges.

```
c Kanter mellom symmetriflatene
c Kanten hvor dC/dx=0 og hvor dC/dy=0
      i=nx
      j=ny
      do k=2,nz/2-1
        C(i+1,j,k)=C(i-1,j,k)
        C(i,j+1,k)=C(i,j-1,k)
        tmp=C(i,j,k)
c
        tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
        tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
        tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
        tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
        tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
        tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
        tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
        tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
        tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
        C(i,j,k) = C(i,j,k)/tmp1
c
        err=err+abs(C(i,j,k)-tmp)
        errmax=max(errmax,abs(C(i,j,k)-tmp))
      enddo
```

Error

The difference between the old and the new ϕ at each point was calculatd, then summed over all the points. This is the error , and corresponds to the difference between the old and the new values of ϕ through the entire matrix at each simulation. This error decreases for each simulation, and eventually converges to zero. The simulation is run again and again, until the error gets below a certain barrier.

3.7 Flux

From equation 23:

$$I = 8 * \Bigg(- \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\ \mathrm{d}z\ -\ \iint_{x,z=0}^{l_x,l_z} \frac{\partial \phi}{\partial y}\bigg|_{y=l_y} \mathrm{d}x\ \mathrm{d}z\ -\ \iint_{x,y=0}^{l_x,l_y} \frac{\partial \phi}{\partial z}\bigg|_{z=l_z} \mathrm{d}x\ \mathrm{d}y \Bigg)$$

$$= 8 * l_x * \Bigg(\int_{\frac{l_x}{2l_y}}^{\frac{l_y}{l_x}} \int_{\frac{l_x}{2l_z}}^{\frac{l_z}{l_x}} \frac{\hat{x}^2}{\hat{y}^2\hat{z}^2} \frac{\partial \phi}{\partial \hat{x}}\bigg|_{\hat{x}=\frac{1}{2}} \mathrm{d}\hat{y}\ \mathrm{d}\hat{z}\ +\ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} \frac{\hat{y}^2}{\hat{x}^2\hat{z}^2} \frac{\partial \phi}{\partial \hat{y}}\bigg|_{\hat{y}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{z}$$

$$+\ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} \frac{\hat{z}^2}{\hat{x}^2\hat{y}^2} \frac{\partial \phi}{\partial \hat{z}}\bigg|_{\hat{z}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{y} \Bigg)$$

Or, the flux can be calculated using the basis solutions $\phi^{(1)},\ \phi^{(2)},\ \phi^{(3)}$:

$$i^{(x)} = 8 * \Bigg(- \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(1)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\ \mathrm{d}z\ -\ \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(2)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\ \mathrm{d}z\ -\ \iint_{y,z=0}^{l_y,l_z} \frac{\partial \phi^{(3)}}{\partial x}\bigg|_{x=l_x} \mathrm{d}y\ \mathrm{d}z \Bigg)$$

$$= 8 * l_x * \Bigg(\int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{x}^2}{\hat{y}^2\hat{z}^2}\ \frac{\partial \phi^{(1)}}{\partial \hat{x}}\bigg|_{\hat{x}=\frac{1}{2}} \mathrm{d}\hat{y}\ \mathrm{d}\hat{z}\ +\ \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{x}^2}{\hat{y}^2\hat{z}^2}\ \frac{\partial \phi^{(2)}}{\partial \hat{x}}\bigg|_{\hat{x}=\frac{1}{2}} \mathrm{d}\hat{y}\ \mathrm{d}\hat{z}$$

$$+\ \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{x}^2}{\hat{y}^2\hat{z}^2}\ \frac{\partial \phi^{(3)}}{\partial \hat{x}}\bigg|_{\hat{x}=\frac{1}{2}} \mathrm{d}\hat{y}\ \mathrm{d}\hat{z} \Bigg)$$

$$i^{(y)} = 8 * \Bigg(- \int_{x=0}^{l_x}\int_{z=0}^{l_z} \frac{\partial \phi^{(1)}}{\partial y}\bigg|_{y=1/2} \mathrm{d}x\ \mathrm{d}z\ -\ \int_{x=0}^{l_x}\int_{z=0}^{l_z} \frac{\partial \phi^{(2)}}{\partial y}\bigg|_{y=1/2} \mathrm{d}x\ \mathrm{d}z\ -\ \int_{x=0}^{l_x}\int_{z=0}^{l_z} \frac{\partial \phi^{(3)}}{\partial y}\bigg|_{y=1/2} \mathrm{d}x\ \mathrm{d}$$

$$= 8 * l_x *\ \Bigg(\int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{y}^2}{\hat{x}^2 \hat{z}^2}\ \frac{\partial \phi^{(1)}}{\partial \hat{y}}\bigg|_{\hat{y}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{z}\ +\ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{y}^2}{\hat{x}^2 \hat{z}^2}\ \frac{\partial \phi^{(2)}}{\partial \hat{y}}\bigg|_{\hat{y}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{z}$$

$$+ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_z}}^{\frac{l_x}{l_z}} l_x\ \frac{\hat{y}^2}{\hat{x}^2 \hat{z}^2}\ \frac{\partial \phi^{(3)}}{\partial \hat{y}}\bigg|_{\hat{y}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{z} \Bigg)$$

$$i^{(z)} = 8 * \Bigg(- \int_{x=0}^{l_x}\int_{y=0}^{l_y} \frac{\partial \phi^{(1)}}{\partial z}\bigg|_{z=1/2} \mathrm{d}x\ \mathrm{d}y\ -\ \int_{x=0}^{l_x}\int_{y=0}^{l_y} \frac{\partial \phi^{(2)}}{\partial z}\bigg|_{z=1/2} \mathrm{d}x\ \mathrm{d}y\ -\ \int_{x=0}^{l_x}\int_{y=0}^{l_y} \frac{\partial \phi^{(3)}}{\partial z}\bigg|_{z=1/2} \mathrm{d}x\ \mathrm{d}$$

$$= 8 * l_x *\ \Bigg(\int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} l_x\ \frac{\hat{z}^2}{\hat{x}^2 \hat{y}^2}\ \frac{\partial \phi^{(1)}}{\partial \hat{z}}\bigg|_{\hat{z}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{y}\ +\ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} l_x\ \frac{\hat{z}^2}{\hat{x}^2 \hat{y}^2}\ \frac{\partial \phi^{(2)}}{\partial \hat{z}}\bigg|_{\hat{z}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{y}$$

$$+\ \int_{\frac{1}{2}}^{1} \int_{\frac{l_x}{2l_y}}^{\frac{l_x}{l_y}} l_x\ \frac{\hat{z}^2}{\hat{x}^2 \hat{y}^2}\ \frac{\partial \phi^{(3)}}{\partial \hat{z}}\bigg|_{\hat{z}=\frac{1}{2}} \mathrm{d}\hat{x}\ \mathrm{d}\hat{y} \Bigg)$$

$$I = i^{(x)} + i^{(y)} + i^{(z)}$$

3.8 Calculating the flux numerically

$f(i,j,k)$ is calculated at all points of the three surfaces where $\phi = 1$ of the transformed precipitate, and then summed over all the points. This sum is multiplied with 8 to get the flux through the entire precipitate.

Concentration beyond surfaces

In order to calculate the concentration gradient using the central difference formula, we need to calculate the concentration at one point beyond the surface. We want to use the central difference formula as this gives us a higher precision than the forward or backward difference formulas. To calculate the concentration at the points beyond the surfaces we solve the diffusion equation with respect to $\phi_{i+1,j,k}$, ϕ_{ij+1k} and ϕ_{ijk+1}. From equation 28

$$-\frac{4\hat{x}_i^4 * \phi_{i+1jk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2} - \frac{2\hat{x}_i^3\phi_{i+1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}$$

$$= 4\hat{x}_i^4\frac{\phi_{i-1jk}-2\phi_{ijk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2} - 2\hat{x}_i^3\frac{\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}} + 4\hat{y}_j^4\frac{\phi_{ij+1k}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}$$

$$+ 2\hat{y}_j^3\frac{\phi_{ij+1k}-\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}} + 4\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2} + 2\hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_k-1}$$

Thus

$$\phi_{i+1jk} = -\frac{1}{\frac{2\hat{x}_i^4}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2}+\frac{\hat{x}_i^3}{\hat{x}_{i+1}-\hat{x}_{i-1}}} * \left[2\hat{x}_i^4\frac{\phi_{i-1jk}-2\phi_{ijk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2} - \hat{x}_i^3\frac{\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}\right.$$

$$+ 2\hat{y}_j^4\frac{\phi_{ij+1k}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2} + \hat{y}_j^3\frac{\phi_{ij+1k}-\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}} + 2\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2}$$

$$\left.+ \hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}\right]$$

The concentration at these points need to be modelled even though they are inside the untransformed precipitate (eq outside the transformed precipitate). The concentration at these poins is necessary to be able to model the flux at the interfaces. or we could have used a forward difference formula, but that would have led to less precision.

Also

$$
-4\hat{y}_j^4\frac{\phi_{ij+1k}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}-2\hat{y}_j^3\frac{\phi_{ij+1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}
$$

$$
=4\hat{x}_i^4\frac{\phi_{i+1jk}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2}+2\hat{x}_i^3\frac{\phi_{i+1jk}-\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}+4\hat{y}_j^4\frac{\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}
$$

$$
-2\hat{y}_j^3\frac{\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}+4\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2}+2\hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}
$$

Thus

$$
\phi_{ij+1k}=-\frac{1}{\frac{2\hat{y}_j^4}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}+\frac{\hat{y}_j^3}{\hat{y}_{j+1}-\hat{y}_{j-1}}}*\left[2\hat{x}_i^4\frac{\phi_{i+1jk}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2}\right.
$$

$$
+\hat{x}_i^3\frac{\phi_{i+1jk}-\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}+2\hat{y}_j^4\frac{\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}
$$

$$
\left.-\hat{y}_j^3\frac{\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}+2\hat{z}_k^4\frac{\phi_{ijk+1}+\phi_{ijk-1}-2\phi_{ijk}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2}+\hat{z}_k^3\frac{\phi_{ijk+1}-\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}\right]
$$

Also

$$
-4\hat{z}_k^4\frac{\phi_{ijk+1}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2}-2\hat{z}_k^3\frac{\phi_{ijk+1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}
$$

$$
=4\hat{x}_i^4\frac{\phi_{i+1jk}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{x}_{i+1}-\hat{x}_{i-1}\right)^2}+2\hat{x}_i^3\frac{\phi_{i+1jk}-\phi_{i-1jk}}{\hat{x}_{i+1}-\hat{x}_{i-1}}+4\hat{y}_j^4\frac{\phi_{ij+1k}+\phi_{ij-1k}-2\phi_{ijk}}{\left(\hat{y}_{j+1}-\hat{y}_{j-1}\right)^2}
$$

$$
+2\hat{y}_j^3\frac{\phi_{ij+1k}-\phi_{ij-1k}}{\hat{y}_{j+1}-\hat{y}_{j-1}}+4\hat{z}_k^4\frac{\phi_{ijk-1}-2\phi_{ijk}}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2}-2\hat{z}_k^3\frac{\phi_{ijk-1}}{\hat{z}_{k+1}-\hat{z}_{k-1}}
$$

Thus

$$\phi_{ijk+1} = -\frac{1}{\frac{2\hat{z}_k^4}{\left(\hat{z}_{k+1}-\hat{z}_{k-1}\right)^2} + \frac{\hat{z}_k^3}{\hat{z}_{k+1}-\hat{z}_{k-1}}} * \Bigg[2\hat{x}_i^4 \frac{\phi_{i+1jk} + \phi_{ij-1k} - 2\phi_{ijk}}{\left(\hat{x}_{i+1} - \hat{x}_{i-1}\right)^2}$$

$$+ \hat{x}_i^3 \frac{\phi_{i+1jk} - \phi_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} + 2\hat{y}_j^4 \frac{\phi_{ij+1k} + \phi_{ij-1k} - 2\phi_{ijk}}{\left(\hat{y}_{j+1} - \hat{y}_{j-1}\right)^2}$$

$$+ \hat{y}_j^3 \frac{\phi_{ij+1k} - \phi_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} + 2\hat{z}_k^4 \frac{\phi_{ijk-1} - 2\phi_{ijk}}{\left(\hat{z}_{k+1} - \hat{z}_{k-1}\right)^2} - \hat{z}_k^3 \frac{\phi_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} \Bigg]$$

Flux through surfaces

The flux through the surface $i = nx/2$ is calculated numerically as

$$j^{(x)} = l_x * \Bigg[\sum_{j=\frac{ny}{2}+1}^{ny-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(1)}_{i+1jk} - \phi^{(1)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{1}{4} \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2}$$

$$+ \sum_{j=\frac{ny}{2}+1}^{ny-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(2)}_{i+1jk} - \phi^{(2)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{1}{4} \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2}$$

$$+ \sum_{j=\frac{ny}{2}+1}^{ny-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(3)}_{i+1jk} - \phi^{(3)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{1}{4} \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \Bigg]$$

The flux through the surface $j = ny/2$ is calculated numerically as

$$
\begin{aligned}
j^{(y)} =& l_x \Bigg[\sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(1)}_{ij+1k} - \phi^{(1)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \\
&+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(2)}_{ij+1k} - \phi^{(2)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \\
&+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(3)}_{ij+1k} - \phi^{(3)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \Bigg]
\end{aligned}
$$

The flux through the surface $k = nz/2$ is calculated numerically as

$$
\begin{aligned}
j^{(z)} =& l_x \Bigg[\sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(1)}_{ijk+1} - \phi^{(1)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \\
&+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(2)}_{ijk+1} - \phi^{(2)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \\
&+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(3)}_{ijk+1} - \phi^{(3)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{1}{4} \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \Bigg]
\end{aligned}
$$

Below figures an extract from the code that shows how the flux is calculated through the surfaces.

```
c___________________________________
c NB
c flux beregnes her først for lx=(c_i-c_0)=1
c og kun for en åttendedel av kuben.
c
C Beregner flux gjennom flatene som tilsvarer overflatene
c i den utransformerte kuben
c bruker ligningen til å beregne fiktivt punkt
c utenfor flata (fysisk inni kuben).
c___________________________________
c for flatenormal i x-retning.
      fluxx = 0.
c
c Flux indre flatepunkt
      i=nx/2
      do j=ny/2+1,ny-1
        do k=nz/2+1,nz-1
c
          tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
          tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
          tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
          C(i+1,j,k) = C(i,j,k)*tmp1
c
          tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &         ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
          C(i+1,j,k) = C(i+1,j,k) - tmp1
c
          tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &         + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
          c(i+1,j,k) = C(i+1,j,k) - tmp1
c
          tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &         + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
          c(i+1,j,k) = C(i+1,j,k) - tmp1
c
          C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &         /(x(i)**3*x(i+1))
c
          f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
          f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
          f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(y(j+1)-y(j-1))
c
          fluxx=fluxx+f(i,j,k)
        enddo
      enddo
```

Flux through edges

The flux through the edge where $i = nx/2$ and j=ny:

$$
\begin{aligned}
j^{(x)} =& l_x \Bigg[\sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(1)}_{i+1jk} - \phi^{(1)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_j - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \\
&+ \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(2)}_{i+1jk} - \phi^{(2)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_j - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \\
&+ \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(3)}_{i+1jk} - \phi^{(3)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_j - \hat{y}_{j-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \Bigg]
\end{aligned}
$$

The flux through the edge where $i = nx/2$ and k=nz:

$$
\begin{aligned}
j^{(x)} =& l_x \Bigg[\sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(1)}_{i+1jk} - \phi^{(1)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \\
&+ \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(2)}_{i+1jk} - \phi^{(2)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \\
&+ \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(3)}_{i+1jk} - \phi^{(3)}_{i-1jk}}{\hat{x}_{i+1} - \hat{x}_{i-1}} * \hat{x}_i^2 * \frac{(\hat{y}_{j+1} - \hat{y}_{j-1})(\hat{z}_{k} - \hat{z}_{k-1})}{\hat{y}_j^2 \ \hat{z}_k^2} \Bigg]
\end{aligned}
$$

The flux through the edge where $j = ny/2$ and i=nx:

$$
\begin{aligned}
j^{(y)} =& l_x \Bigg[\sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(1)}_{ij+1k} - \phi^{(1)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \\
&+ \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(2)}_{ij+1k} - \phi^{(2)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \\
&+ \sum_{k=\frac{nz}{2}+1}^{nz-1} \frac{\phi^{(3)}_{ij+1k} - \phi^{(3)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \Bigg]
\end{aligned}
$$

The flux through the edge where $j = ny/2$ and k=nz:

$$j^{(y)} = l_x \left[\sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(1)}_{ij+1k} - \phi^{(1)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \right.$$

$$+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(2)}_{ij+1k} - \phi^{(2)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2}$$

$$\left. + \sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(3)}_{ij+1k} - \phi^{(3)}_{ij-1k}}{\hat{y}_{j+1} - \hat{y}_{j-1}} * \hat{y}_j^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{z}_{k+1} - \hat{z}_{k-1})}{\hat{x}_i^2 \ \hat{z}_k^2} \right]$$

The flux through the edge where $k = nz/2$ and i=nx:

$$j^{(z)} = l_x \left[\sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(1)}_{ijk+1} - \phi^{(1)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \right.$$

$$+ \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(2)}_{ijk+1} - \phi^{(2)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2}$$

$$\left. + \sum_{j=\frac{ny}{2}+1}^{ny-1} \frac{\phi^{(3)}_{ijk+1} - \phi^{(3)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \right]$$

The flux through the edge where $k = nz/2$ and j=ny:

$$j^{(z)} = l_x \left[\sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(1)}_{ijk+1} - \phi^{(1)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \right.$$

$$+ \sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(2)}_{ijk+1} - \phi^{(2)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2}$$

$$\left. + \sum_{i=\frac{nx}{2}+1}^{nx-1} \frac{\phi^{(3)}_{ijk+1} - \phi^{(3)}_{ijk-1}}{\hat{z}_{k+1} - \hat{z}_{k-1}} * \hat{z}_k^2 * \frac{(\hat{x}_{i+1} - \hat{x}_{i-1})(\hat{y}_{j+1} - \hat{y}_{j-1})}{\hat{x}_i^2 \ \hat{y}_j^2} \right]$$

Below figures and extract of the code that shows how the flux is calculated through the edges. Note that the edges belong to the symmetry face boundary conditions, hence the constraint on the derivative has to be specified in the code.

```
c kanter
c
    i=nx/2
    k=nz
    do j = ny/2+1,ny-1
      C(i,j,k+1) = C(i,j,k-1)
c
      tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
      tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
      tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
      C(i+1,j,k) = C(i,j,k)*tmp1
c
      tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &      ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
      C(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &      + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
      tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &      + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
      tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &      /(x(i)**3*x(i+1))
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j+1)-y(j-1))
      fluxx=fluxx+f(i,j,k)
    enddo
```

Flux through corners

Below figures and extract of the code that shows how the flux is calculated through the corners. Note that the corners belong to the symmetry face boundary conditions, and that the constraint on the derivative has to be specified in the code.

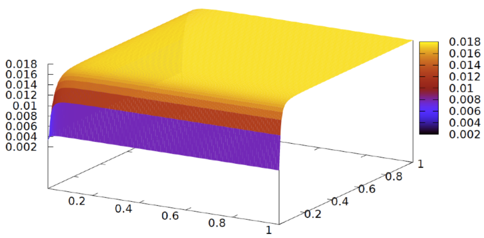
```
c hjørner
c
      i=nx/2
      j=ny/2
      k=nz/2
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(y(j+1)-y(j))
c
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny
      k=nz/2
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(y(j)-y(j-1))
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny
      k=nz
c
      C(i,j+1,k) = C(i,j-1,k)
      C(i,j,k+1) = C(i,j,k-1)
c
      tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
      tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
      tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
      C(i+1,j,k) = C(i,j,k)*tmp1
c
      tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &    ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
      C(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &    + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
      tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &    + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
      tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &    /(x(i)**3*x(i+1))
```

```
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j)-y(j-1))
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny/2
      k=nz
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j+1)-y(j))
      fluxx = fluxx + f(i,j,k)
```

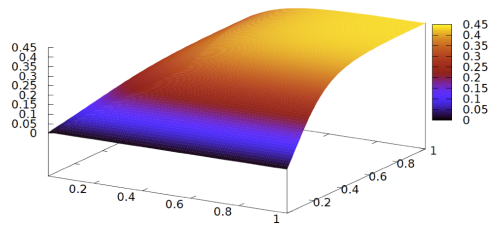
4 Results and discussion

4.1 Concentration profiles

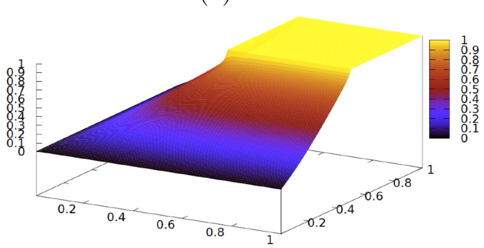
The concentration profile through a transformed volume in which $\tilde{l}_x = \tilde{l}_y = \tilde{l}_z$ (a cube) is plotted in figure 18, using a 3-D shaded surface plot in Matlab. The concentration is plotted at different heights, corresponding to different values of k. The concentration was modelled using nx=ny=nz=120.



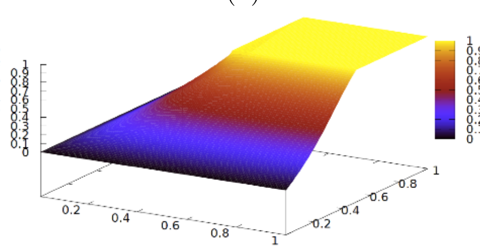
(a) k=2



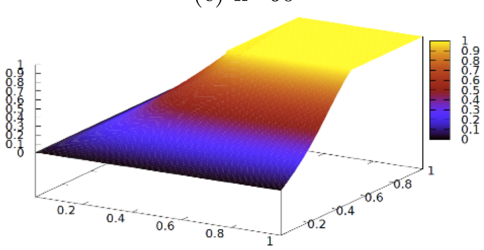
(b) k=30



(c) k=60



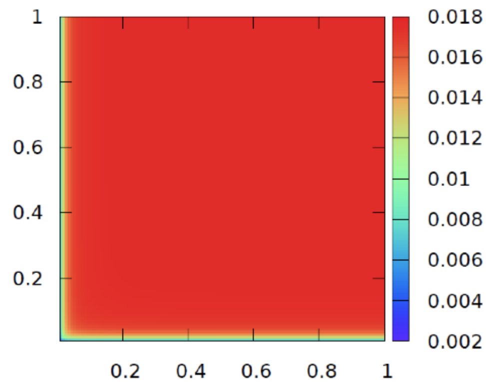
(d) k=90



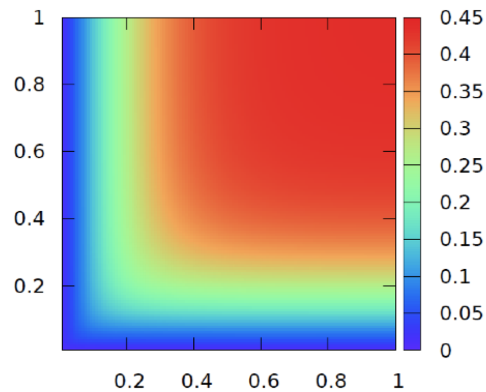
(e) k=120

Figure 18: k=2, 30,60, 90, 120

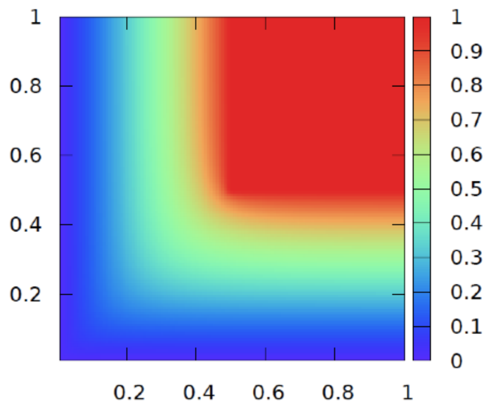
The concentration profile through the transformed volume is plotted in figure 19, using a contour plot displaying isolines in Matlab. The concentration is plotted at different heights, corresponding to different values of k. The concentration was modelled using nx=ny=nz=120.



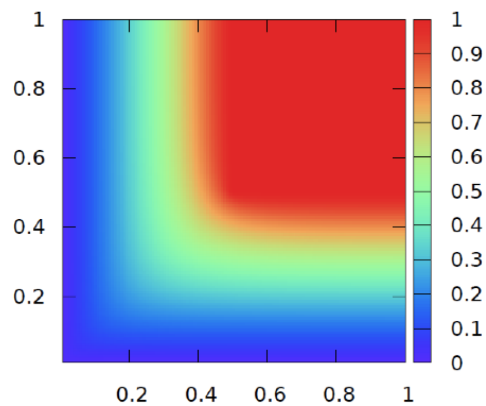
(a) k=2



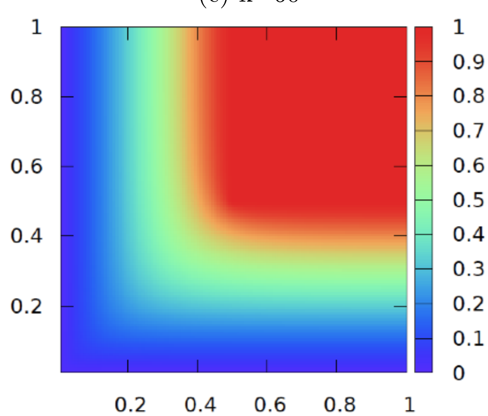
(b) k=30



(c) k=60



(d) k=90



(e) k=120

Figure 19: k=2, 30,60, 90, 120

4.2 Flux

4.2.1 flux(N)

The flux is plotted against N in figure 20 below for a precipitate in which lx/lx=ly/lx=lz/lx, a cube. As N increases the flux seems to converge. The green dots show the flux obtained when the program is run until the error gets smaller than 1e-7, the red dots show the flux obtained when the program is run until the error gets smaller than 1e-8. Hence we can see that the flux hasn't converged completely yet when the error is 1e-7.

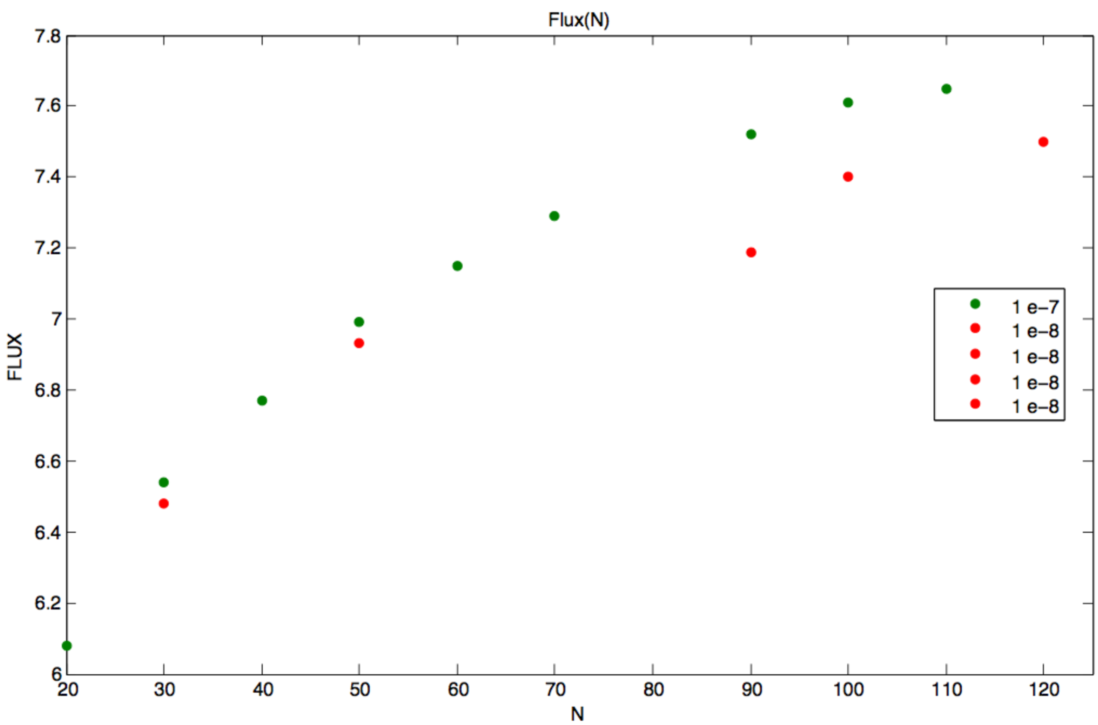


Figure 20: Flux versus N for a precipitate in which lx/lx=ly/lx=lz/lx. N=nx=ny=nz.

4.2.2 Needles

The flux is plotted against the aspect ratio lz/lx in figure 21 below. N was set to be 70 and the error 1e-7, even if we can see from figure 21 that the flux hasn't converged either at N=70 or at error 1e-7. We chose these values for N and the error because of lack of time, as running the program for many points and a smaller error takes much time. Hence the figure below can be used to understand the evolution of the flux qualitatively. We see that the flux initially decreases distinctively when the aspect ratio increases, and then it starts to increase from an aspect ratio of approximatively 20. The reason for which the flux initially decreases could be related to the fact that the precipitate is stretched. As the aspect ratio increases, the total surface area of the precipitate increases, so that could be the reason for why the flux starts to increase again.

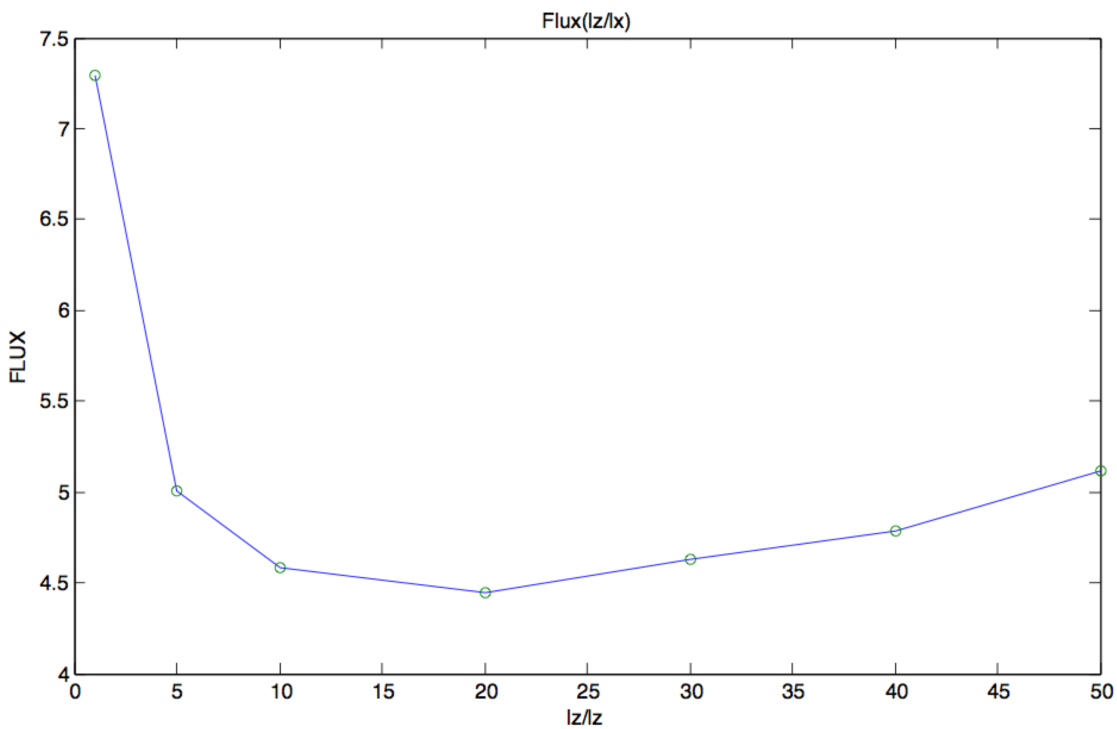


Figure 21: Flux versus lz/lx for precipitate needles. nx=ny=nz=70 and the error is 1e-7.

4.2.3 Plates

The flux is plotted against the aspect ratio lz/lx=ly/lx for plates in figure 21 below. N was set to be 70 and the error 1e-7 again, for the same reason as for the needles. We see that the flux initially decreases steeply when the aspect ratios increase, and then it decreases slower. The reason for why the flux decreases is most probable to the new shape.

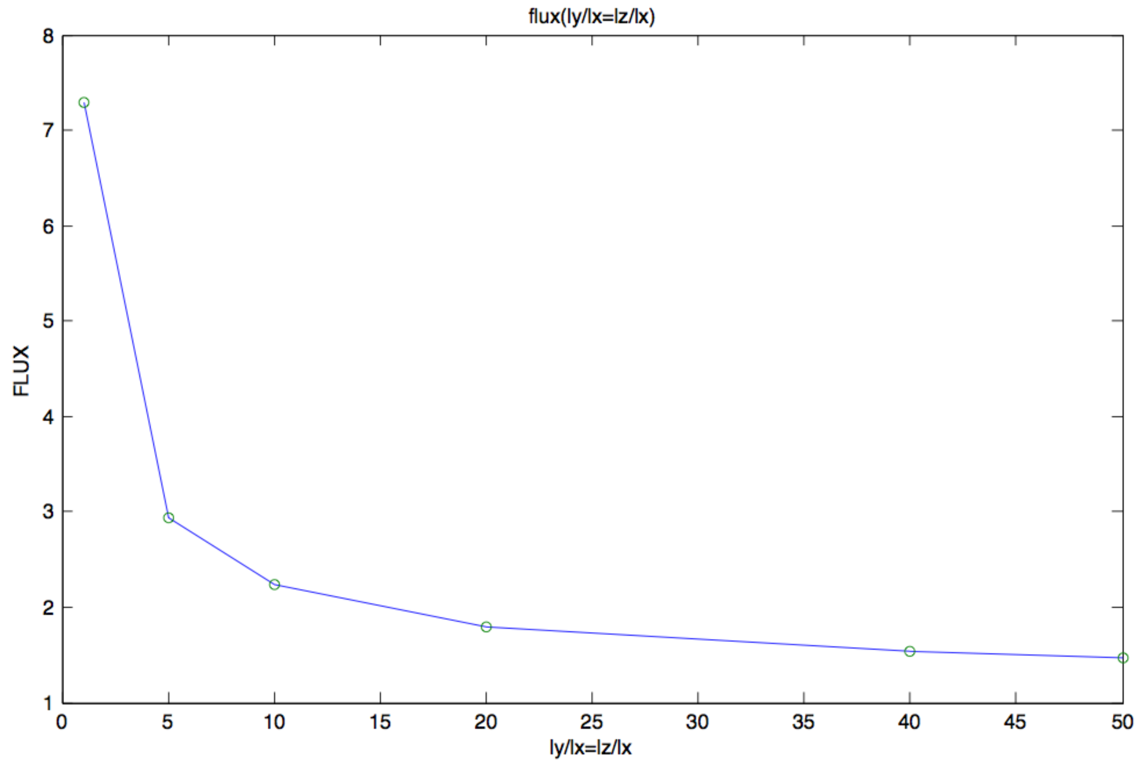


Figure 22: Flux versus ly/lx=lz/lx

5 Conclusion

In this project we did the first step to implement a model to describe the evolution of precipitate distribution during precipitation that could be used for all aspect ratios of the precipitate needles. We solved the diffusion equation in cartesian coordinates using the finite elements method to calculate the concentration at each point of the matrix. The concentration gradient at the interface is used to calculate the flux through the cuboid/matrix interface. The flux was calculated for different number of points N, for different aspect ratios and for different errors. The results could be used to study the flux qualitatively, but for more accurate results the program should be run until the flux has converged completely.

6 Further work

The results of this project open for further work:

- To study the evolution of the flux more precisely the program should be run with parameters for which the flux converges completely, N=120 and error=1e-8 for example.
- In this project, we chose to keep nx=ny=nz. As the needle are stretched, it could be beneficial to increase the number of points in the axis in which the neelde is stretched. If the needle is stretched along the z axix for example, nz could be larger than nx and ny.

Appendix

The code used to calculate the flux figures below. This code is coded by my supervisor Bjørn Holmedal, as my code did not give the correct results.

```
c Program i 3D
      program laplaceNytrans

      implicit none
      Integer i, j, k, nx, ny, nz, NxMax, NyMax, NzMax, count
      Integer Stopp,itmax
      Double precision Lx, Ly, Lz, L0
      Parameter (NxMax=200, NyMax=200, NzMax=200)
      Double precision C(NxMax,NyMax,NzMax), err, tmp, flux
      Double precision x(NxMax), y(NyMax), z(NzMax)
      Double precision aa(NxMax), bb(NyMax), cc(NzMax), dd(NzMax)
      Double precision f(NxMax,NyMax,NzMax)
      Double Precision tmp1, fluxx, fluxy,fluxz, errmax,minerr,pi


c     NB ratio-er
      Lx=1.
      Ly=1.
      Lz=5.
c Nx, Ny og Nz må være partall!!
      nx=70
      ny=70
      nz=70
      pi = 3.141592654
      minerr = 1.0E-7
      itmax = nx*ny*nz

c førstegjetning for flux
      flux=7.

c     Definerer partisering
      do i=1,nx+1
        x(i)=i*Lx/Nx
      enddo
      do j=1,ny+1
        y(j)=j*Ly/Ny
      enddo
      do k=1,nz+1
        z(k)=k*Lz/Nz
      enddo

c     Gjetter initialfelt
      do i=1,nx
        do j=1,ny
          do k=1,nz
            tmp = y(j)**2*z(k)**2+x(i)**2*z(k)**2+x(i)**2*y(j)**2
            C(i,j,k)=flux*x(i)*y(j)*z(k)/(4.*pi*sqrt(tmp))
            C(i,j,k)=min(tmp,1.)
            C(i,j,k)=0.
```

```
          enddo
        enddo
      enddo

      count = 0
      stopp = 0

      open(16,file="err.txt")

 100  continue
      err=0.
      errmax=0.

C Definerer concentrasjon på grensebetingelsene
c Flatene hvor C=analystisk asymptotisk løsning:
      i=1
      do j=1,ny
        do k=1,nz
c          C(i,j,k)=0.
          tmp = y(j)**2*z(k)**2+x(i)**2*z(k)**2+x(i)**2*y(j)**2
          C(i,j,k)=flux*x(i)*y(j)*z(k)/(4.*pi*sqrt(tmp))
        enddo
      enddo

      j=1
      do i=1,nx
        do k=1,nz
c          C(i,j,k)=0.
          tmp = y(j)**2*z(k)**2+x(i)**2*z(k)**2+x(i)**2*y(j)**2
          C(i,j,k)=flux*x(i)*y(j)*z(k)/(4.*pi*sqrt(tmp))
        enddo
      enddo

      k=1
      do i=1,nx
        do j=1,ny
c          C(i,j,k)=0.
          tmp = y(j)**2*z(k)**2+x(i)**2*z(k)**2+x(i)**2*y(j)**2
          C(i,j,k)=flux*x(i)*y(j)*z(k)/(4.*pi*sqrt(tmp))
        enddo
      enddo

c Flatene hvor C=1:
      j=ny/2
      do i=nx/2,nx
        do k=nz/2,nz
          C(i,j,k)=1.
        enddo
      enddo
```

```
      k=nz/2
      do i=nx/2,nx
        do j=ny/2,ny
          C(i,j,k)=1.
        enddo
      enddo

      i=nx/2
      do j=ny/2,ny
        do k=nz/2,nz
          C(i,j,k)=1.
        enddo
      enddo

c Symmetriflatene:
c     dC/dx = 0:
      i=nx
      do j=2,ny-1
        do k=2,nz/2-1
          C(i+1,j,k)=C(i-1,j,k)
          tmp=C(i,j,k)
c
          tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
          tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
          tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
          tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
          tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo

      i=nx
      do j=2,ny/2-1
        do k=nz/2,nz-1
```

```
          C(i+1,j,k)=C(i-1,j,k)
          tmp=C(i,j,k)
c
          tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
          tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
          tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
          tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
          tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo


c dC/dy= 0:
      j=ny
      do i=2,nx-1
        do k=2,nz/2-1
          C(i,j+1,k)=C(i,j-1,k)
          tmp=C(i,j,k)
c
          tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
          tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
          tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
          tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
          tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
```

```
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo


      j=ny
      do i=2,nx/2-1
        do k=nz/2,nz-1
          C(i,j+1,k)=C(i,j-1,k)
          tmp=C(i,j,k)
c
          tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
          tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
          tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
          tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
          tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
          tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
          tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
          tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
          C(i,j,k) = C(i,j,k)/tmp1
c
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo


c dC/dz = 0:
      k=nz
      do i=2,nx/2-1
        do j=2,ny-1
          C(i,j,k+1)=C(i,j,k-1)
          tmp=C(i,j,k)
c
          tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
          tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
```

```
c
        tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
        tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
        tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
        tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
        tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
        tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
        tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
        C(i,j,k) = C(i,j,k)/tmp1
c
        err=err+abs(C(i,j,k)-tmp)
        errmax=max(errmax,abs(C(i,j,k)-tmp))
      enddo
    enddo


    k=nz
    do i=nx/2,nx-1
      do j=2,ny/2-1
        C(i,j,k+1)=C(i,j,k-1)
        tmp=C(i,j,k)
c
        tmp1 =       x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
        tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
        tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
        tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
        tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
        tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
        tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
        tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
        tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
        C(i,j,k) = C(i,j,k)/tmp1
c
        err=err+abs(C(i,j,k)-tmp)
        errmax=max(errmax,abs(C(i,j,k)-tmp))
      enddo
    enddo
```

```
c Kanter mellom symmetriflatene
c Kanten hvor dC/dx=0 og hvor dC/dy=0
     i=nx
     j=ny
     do k=2,nz/2-1
       C(i+1,j,k)=C(i-1,j,k)
       C(i,j+1,k)=C(i,j-1,k)
       tmp=C(i,j,k)
c
       tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
       tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
       C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
       tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
       tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
       C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
       tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
       tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
       C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
       tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
       tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
       tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
       C(i,j,k) = C(i,j,k)/tmp1
c
       err=err+abs(C(i,j,k)-tmp)
       errmax=max(errmax,abs(C(i,j,k)-tmp))
     enddo


c Kanten hvor dC/dy=0 og hvor dC/dz=0
     j=ny
     k=nz
     do i=2,nx/2-1
       C(i,j+1,k)=C(i,j-1,k)
       C(i,j,k+1)=C(i,j,k-1)
       tmp=C(i,j,k)
c
       tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
       tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
       C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
       tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
       tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
       C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
```

```
c
        tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
        tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
        tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
        tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
        tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
        C(i,j,k) = C(i,j,k)/tmp1
c
        err=err+abs(C(i,j,k)-tmp)
        errmax=max(errmax,abs(C(i,j,k)-tmp))
      enddo

c Kanten hvor dC/dx=0 og hvor dC/dz=0
      i=nx
      k=nz
      do j=2,ny/2-1
        C(i+1,j,k)=C(i-1,j,k)
        C(i,j,k+1)=C(i,j,k-1)
        tmp=C(i,j,k)
c
        tmp1 =        x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
        tmp1 = tmp1 + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        C(i,j,k) =    x(i)**3*tmp1/(x(i+1)-x(i-1))
c
        tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
        tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        C(i,j,k) = C(i,j,k) + y(j)**3*tmp1/(y(j+1)-y(j-1))
c
        tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
        tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        C(i,j,k) = C(i,j,k) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
        tmp1 = x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
        tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
        tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
        C(i,j,k) = C(i,j,k)/tmp1
c
        err=err+abs(C(i,j,k)-tmp)
        errmax=max(errmax,abs(C(i,j,k)-tmp))
      enddo


C Beregner konsentrasjon i det transformerte volumet
c indre pkt:
      do j=2,ny-1
```

```
        do k=2,nz/2-1
          bb(1) = 1.
          cc(1) = 0.
          dd(1) = c(1,j,k)
          do i=2,nx-1
c             tmp=C(i,j,k)
c
            tmp1 = x(i)**3/(x(i+1)-x(i-1))
            aa(i) = -tmp1*x(i-1)/(x(i)-x(i-1))
            cc(i) = -tmp1*x(i+1)/(x(i+1)-x(i))
c
            tmp1 =       x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
            tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
            tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
            bb(i) = tmp1
c
            tmp1 =       y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
            tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
            dd(i) =  y(j)**3*tmp1/(y(j+1)-y(j-1))
c
            tmp1 =       z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
            tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
            dd(i) = dd(i) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
c
            tmp1 = aa(i)/bb(i-1)
            bb(i) = bb(i)-tmp1*cc(i-1)
            dd(i) = dd(i)-tmp1*dd(i-1)
c
c             c(i,j,k)=(dd(i)-aa(i)*C(i-1,j,k)-cc(i)*C(i+1,j,k))
c             C(i,j,k)=C(i,j,k)/bb(i)
c
c             err=err+abs(C(i,j,k)-tmp)
c             errmax=max(errmax,abs(C(i,j,k)-tmp))
          enddo
          dd(nx-1)=dd(nx-1)-cc(nx-1)*C(nx,j,k)
          tmp=C(nx-1,j,k)
          C(nx-1,j,k)=dd(nx-1)/bb(nx-1)
          err=err+abs(C(nx-1,j,k)-tmp)
          errmax=max(errmax,abs(C(nx-1,j,k)-tmp))
          do i=nx-2,2,-1
            tmp=C(i,j,k)
            C(i,j,k)=(dd(i)-cc(i)*C(i+1,j,k))/bb(i)
            err=err+abs(C(i,j,k)-tmp)
            errmax=max(errmax,abs(C(i,j,k)-tmp))
          enddo
        enddo
      enddo
```

```
      do j=2,ny/2-1
        do k=nz/2,nz-1
          bb(1) = 1.
          cc(1) = 0.
          dd(1) = c(1,j,k)
          do i=2,nx-1
c             tmp=C(i,j,k)
c
            tmp1 = x(i)**3/(x(i+1)-x(i-1))
            aa(i) = -tmp1*x(i-1)/(x(i)-x(i-1))
            cc(i) = -tmp1*x(i+1)/(x(i+1)-x(i))
c
            tmp1 =        x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
            tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
            tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
            bb(i) = tmp1
c
            tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
            tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
            dd(i) =  y(j)**3*tmp1/(y(j+1)-y(j-1))
c
            tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
            tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
            dd(i) = dd(i) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
c
            tmp1 = aa(i)/bb(i-1)
            bb(i) = bb(i)-tmp1*cc(i-1)
            dd(i) = dd(i)-tmp1*dd(i-1)
c
c             c(i,j,k)=(dd(i)-aa(i)*C(i-1,j,k)-cc(i)*C(i+1,j,k))
c             C(i,j,k)=C(i,j,k)/bb(i)
c
c             err=err+abs(C(i,j,k)-tmp)
c             errmax=max(errmax,abs(C(i,j,k)-tmp))
          enddo
          dd(nx-1)=dd(nx-1)-cc(nx-1)*C(nx,j,k)
          tmp=C(nx-1,j,k)
          C(nx-1,j,k)=dd(nx-1)/bb(nx-1)
          err=err+abs(C(nx-1,j,k)-tmp)
          errmax=max(errmax,abs(C(nx-1,j,k)-tmp))
          do i=nx-2,2,-1
            tmp=C(i,j,k)
            C(i,j,k)=(dd(i)-cc(i)*C(i+1,j,k))/bb(i)
            err=err+abs(C(i,j,k)-tmp)
            errmax=max(errmax,abs(C(i,j,k)-tmp))
```

```
          enddo
        enddo
      enddo


      do j=ny/2,ny-1
        do k=nz/2,nz-1
          bb(1) = 1.
          cc(1) = 0.
          dd(1) = c(1,j,k)
          do i=2,nx/2-1
c             tmp=C(i,j,k)
c
            tmp1 = x(i)**3/(x(i+1)-x(i-1))
            aa(i) = -tmp1*x(i-1)/(x(i)-x(i-1))
            cc(i) = -tmp1*x(i+1)/(x(i+1)-x(i))
c
            tmp1 =        x(i)**4/(x(i+1)-x(i))/(x(i)-x(i-1))
            tmp1 = tmp1 + y(j)**4/(y(j+1)-y(j))/(y(j)-y(j-1))
            tmp1 = tmp1 + z(k)**4/(z(k+1)-z(k))/(z(k)-z(k-1))
c
            bb(i) = tmp1
c
            tmp1 =        y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
            tmp1= tmp1 +  y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
            dd(i) =  y(j)**3*tmp1/(y(j+1)-y(j-1))
c
            tmp1 =        z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
            tmp1 = tmp1 + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
            dd(i) = dd(i) + z(k)**3*tmp1/(z(k+1)-z(k-1))
c
c
            tmp1 = aa(i)/bb(i-1)
            bb(i) = bb(i)-tmp1*cc(i-1)
            dd(i) = dd(i)-tmp1*dd(i-1)
c
c             c(i,j,k)=(dd(i)-aa(i)*C(i-1,j,k)-cc(i)*C(i+1,j,k))
c             C(i,j,k)=C(i,j,k)/bb(i)
c
c             err=err+abs(C(i,j,k)-tmp)
c             errmax=max(errmax,abs(C(i,j,k)-tmp))
          enddo
          dd(nx/2-1)=dd(nx/2-1)-cc(nx/2-1)*C(nx/2,j,k)
          tmp=C(nx/2-1,j,k)
          C(nx/2-1,j,k)=dd(nx/2-1)/bb(nx/2-1)
          err=err+abs(C(nx/2-1,j,k)-tmp)
          errmax=max(errmax,abs(C(nx/2-1,j,k)-tmp))
          do i=nx/2-2,2,-1
            tmp=C(i,j,k)
```

```
          C(i,j,k)=(dd(i)-cc(i)*C(i+1,j,k))/bb(i)
          err=err+abs(C(i,j,k)-tmp)
          errmax=max(errmax,abs(C(i,j,k)-tmp))
        enddo
      enddo
    enddo

    count=count+1

    if(count.gt.itmax) then
      Stopp=1
      Goto 300
    EndIf
    if(err/(nx*ny*nz).LT.minerr) then
      Stopp=1
      Goto 300
    Endif
c     If((count/50)*50.eq.count) Goto 300
    goto 300
    Goto 100

 300  Continue

c_______________________________________
c NB
c flux beregnes her først for lx=(c_i-c_0)=1
c og kun for en åttendedel av kuben.
c
C Beregner flux gjennom flatene som tilsvarer overflatene
c i den utransformerte kuben
c bruker ligningen til å beregne fiktivt punkt
c utenfor flata (fysisk inni kuben).
c_______________________________________
c for flatenormal i x-retning.
    fluxx = 0.
c
c Flux indre flatepunkt
    i=nx/2
    do j=ny/2+1,ny-1
      do k=nz/2+1,nz-1
c
        tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i+1,j,k) = C(i,j,k)*tmp1
c
        tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &        ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
        C(i+1,j,k) = C(i+1,j,k) - tmp1
```

```
c
          tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &         + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
          c(i+1,j,k) = C(i+1,j,k) - tmp1
c
          tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &         + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
          tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
          c(i+1,j,k) = C(i+1,j,k) - tmp1
c
          C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &         /(x(i)**3*x(i+1))
c
          f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
          f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
          f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(y(j+1)-y(j-1))
c
          fluxx=fluxx+f(i,j,k)
        enddo
      enddo
c
c kanter
c
      i=nx/2
      k=nz
      do j = ny/2+1,ny-1
        C(i,j,k+1) = C(i,j,k-1)
c
        tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i+1,j,k) = C(i,j,k)*tmp1
c
        tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &       ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
        C(i+1,j,k) = C(i+1,j,k) - tmp1
c
        tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &       + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
        c(i+1,j,k) = C(i+1,j,k) - tmp1
c
        tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &       + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
        c(i+1,j,k) = C(i+1,j,k) - tmp1
c
        C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
```

```
     &       /(x(i)**3*x(i+1))
c
        f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
        f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j+1)-y(j-1))
        fluxx=fluxx+f(i,j,k)
      enddo
c
      i=nx/2
      k=nz/2
      do j = ny/2+1,ny-1
        f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
        f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(y(j+1)-y(j-1))
        fluxx=fluxx+f(i,j,k)
      enddo
c
      i=nx/2
      j=ny
      do k = nz/2+1,nz-1
        C(i,j+1,k)=C(i,j-1,k)
c
        tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i+1,j,k) = C(i,j,k)*tmp1
c
        tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &       ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
        C(i+1,j,k) = C(i+1,j,k) - tmp1
c
        tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &       + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
        c(i+1,j,k) = C(i+1,j,k) - tmp1
c
        tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &       + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
        c(i+1,j,k) = C(i+1,j,k) - tmp1
c
        C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &       /(x(i)**3*x(i+1))
c
        f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
        f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(y(j)-y(j-1))
        fluxx = fluxx + f(i,j,k)
      enddo
```

```
c
      i=nx/2
      j=ny/2
      do k = nz/2+1,nz-1
        f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
        f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(y(j+1)-y(j))
        fluxx = fluxx + f(i,j,k)
      enddo
c
c hjørner
c
      i=nx/2
      j=ny/2
      k=nz/2
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(y(j+1)-y(j))
c
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny
      k=nz/2
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(y(j)-y(j-1))
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny
      k=nz
c
      C(i,j+1,k) = C(i,j-1,k)
      C(i,j,k+1) = C(i,j,k-1)
c
      tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
      tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
      tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
      C(i+1,j,k) = C(i,j,k)*tmp1
c
      tmp1 = x(i)**3*x(i-1)*C(i-1,j,k)/
     &     ((x(i+1)-x(i-1))*(x(i)-x(i-1)))
      C(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &     + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
```

```
      tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &    + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
      tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
      c(i+1,j,k) = C(i+1,j,k) - tmp1
c
      C(i+1,j,k) = C(i+1,j,k)*(x(i+1)-x(i-1))*(x(i+1)-x(i))
     &    /(x(i)**3*x(i+1))
c
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j)-y(j-1))
      fluxx = fluxx + f(i,j,k)
c
      i=nx/2
      j=ny/2
      k=nz
      f(i,j,k)=(C(i+1,j,k)-C(i-1,j,k))/(x(i+1)-x(i-1))
      f(i,j,k)=f(i,j,k)*x(i)**2/y(j)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(y(j+1)-y(j))
      fluxx = fluxx + f(i,j,k)
c
c_____________________________________
c for flatenormal i y-retning.
c
      fluxy = 0.
c Flux indre flatepunkt
      j=ny/2
      do i=nx/2+1,nx-1
        do k=nz/2+1,nz-1
c
          tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
          tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
          tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
          C(i,j+1,k) = C(i,j,k)*tmp1
c
          tmp1 = y(j)**3*y(j-1)*C(i,j-1,k)/
     &        ((y(j+1)-y(j-1))*(y(j)-y(j-1)))
          C(i,j+1,k) = C(i,j+1,k) - tmp1
c
          tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &        + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
          c(i,j+1,k) = C(i,j+1,k) - tmp1
c
          tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &        + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
```

```
          tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
          c(i,j+1,k) = C(i,j+1,k) - tmp1
c
          C(i,j+1,k) = C(i,j+1,k)*(y(j+1)-y(j-1))*(y(j+1)-y(j))
     &       /(y(j)**3*y(j+1))
c
          f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
          f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
          f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(x(i+1)-x(i-1))
          fluxy = fluxy + f(i,j,k)
        enddo
      enddo
c
c kanter
      j=ny/2
      k=nz
      do i = nx/2+1,nx-1
        C(i,j,k+1)=C(i,j,k-1)
c
        tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i,j+1,k) = C(i,j,k)*tmp1
c
        tmp1 = y(j)**3*y(j-1)*C(i,j-1,k)/
     &      ((y(j+1)-y(j-1))*(y(j)-y(j-1)))
        C(i,j+1,k) = C(i,j+1,k) - tmp1
c
        tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &      + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
        c(i,j+1,k) = C(i,j+1,k) - tmp1
c
        tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &      + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
        c(i,j+1,k) = C(i,j+1,k) - tmp1
c
        C(i,j+1,k) = C(i,j+1,k)*(y(j+1)-y(j-1))*(y(j+1)-y(j))
     &       /(y(j)**3*y(j+1))
c
        f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
        f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(x(i+1)-x(i-1))
        fluxy = fluxy + f(i,j,k)
      enddo
c
      j=ny/2
      k=nz/2
```

```
      do i = nx/2+1,nx-1
        f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
        f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(x(i+1)-x(i-1))
        fluxy = fluxy + f(i,j,k)
      enddo
c
      j=ny/2
      i=nx
      do k = nz/2+1,nz-1
        C(i+1,j,k)=C(i-1,j,k)
c
        tmp1 =       x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i,j+1,k) = C(i,j,k)*tmp1
c
        tmp1 = y(j)**3*y(j-1)*C(i,j-1,k)/
     &       ((y(j+1)-y(j-1))*(y(j)-y(j-1)))
        C(i,j+1,k) = C(i,j+1,k) - tmp1
c
        tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &       + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
        c(i,j+1,k) = C(i,j+1,k) - tmp1
c
        tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &       + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
        tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
        c(i,j+1,k) = C(i,j+1,k) - tmp1
c
        C(i,j+1,k) = C(i,j+1,k)*(y(j+1)-y(j-1))*(y(j+1)-y(j))
     &         /(y(j)**3*y(j+1))
c
        f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
        f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(x(i)-x(i-1))
        fluxy = fluxy + f(i,j,k)
      enddo
c
      j=ny/2
      i=nx/2
      do k = nz/2+1,nz-1
        f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
        f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
        f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k-1))*(x(i+1)-x(i))
        fluxy = fluxy + f(i,j,k)
      enddo
c
```

```
c hjørner
      j=ny/2
      i=nx/2
      k=nz/2
c
      f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
      f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(x(i+1)-x(i))
      fluxy = fluxy + f(i,j,k)
c
      j=ny/2
      i=nx
      k=nz/2
c
      f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
      f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k+1)-z(k))*(x(i)-x(i-1))
      fluxy = fluxy + f(i,j,k)
c
      j=ny/2
      i=nx
      k=nz
      C(i+1,j,k)=C(i-1,j,k)
      C(i,j,k+1)=C(i,j,k-1)
c
      tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
      tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
      tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
      C(i,j+1,k) = C(i,j,k)*tmp1
c
      tmp1 = y(j)**3*y(j-1)*C(i,j-1,k)/
     &     ((y(j+1)-y(j-1))*(y(j)-y(j-1)))
      C(i,j+1,k) = C(i,j+1,k) - tmp1
c
      tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &     + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
      tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
      c(i,j+1,k) = C(i,j+1,k) - tmp1
c
      tmp1 = z(k+1)*C(i,j,k+1)/(z(k+1)-z(k))
     &     + z(k-1)*C(i,j,k-1)/(z(k)-z(k-1))
      tmp1 = tmp1*z(k)**3/(z(k+1)-z(k-1))
      c(i,j+1,k) = C(i,j+1,k) - tmp1
c
      C(i,j+1,k) = C(i,j+1,k)*(y(j+1)-y(j-1))*(y(j+1)-y(j))
     &     /(y(j)**3*y(j+1))
c
      f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
      f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
```

```
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(x(i)-x(i-1))
      fluxy = fluxy +f(i,j,k)
c
      j=ny/2
      i=nx/2
      k=nz
c
      f(i,j,k)=(C(i,j+1,k)-C(i,j-1,k))/(y(j+1)-y(j-1))
      f(i,j,k)=f(i,j,k)*y(j)**2/x(i)**2/z(k)**2
      f(i,j,k)=f(i,j,k)*0.25*(z(k)-z(k-1))*(x(i+1)-x(i))
      fluxy = fluxy + f(i,j,k)
c
c___________________________________
c for flatenormal i z-retning.
c
      fluxz = 0.
c Flux indre flatepunkt
      k=nz/2
      do i=nx/2+1,nx-1
        do j=ny/2+1,ny-1
c
          tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
          tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
          tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
          C(i,j,k+1) = C(i,j,k)*tmp1
c
          tmp1 = z(k)**3*z(k-1)*C(i,j,k-1)/
     &         ((z(k+1)-z(k-1))*(z(k)-z(k-1)))
          C(i,j,k+1) = C(i,j,k+1) - tmp1
c
          tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &         + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
          tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
          c(i,j,k+1) = C(i,j,k+1) - tmp1
c
          tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &         + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
          tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
          c(i,j,k+1) = C(i,j,k+1) - tmp1
c
          C(i,j,k+1) = C(i,j,k+1)*(z(k+1)-z(k-1))*(z(k+1)-z(k))
     &         /(z(k)**3*z(k+1))
c
          f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
          f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
          f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j-1))*(x(i+1)-x(i-1))
          fluxz = fluxz + f(i,j,k)
        enddo
      enddo
```

```
c
c kanter
      k=nz/2
      j=ny
      do i = nx/2+1,nx-1
        C(i,j+1,k)=C(i,j-1,k)
c
        tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
        tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
        tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
        C(i,j,k+1) = C(i,j,k)*tmp1
c
        tmp1 = z(k)**3*z(k-1)*C(i,j,k-1)/
     &        ((z(k+1)-z(k-1))*(z(k)-z(k-1)))
        C(i,j,k+1) = C(i,j,k+1) - tmp1
c
        tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &        + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
        tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
        c(i,j,k+1) = C(i,j,k+1) - tmp1
c
        tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &        + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
        tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
        c(i,j,k+1) = C(i,j,k+1) - tmp1
c
        C(i,j,k+1) = C(i,j,k+1)*(z(k+1)-z(k-1))*(z(k+1)-z(k))
     &        /(z(k)**3*z(k+1))
c
        f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
        f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
        f(i,j,k)=f(i,j,k)*0.25*(y(j)-y(j-1))*(x(i+1)-x(i-1))
        fluxz = fluxz + f(i,j,k)
      enddo
c
      k=nz/2
      j=ny/2
      do i = nx/2+1,nx-1
        f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
        f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
        f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j))*(x(i+1)-x(i-1))
        fluxz = fluxz + f(i,j,k)
      enddo
c
      k=nz/2
      i=nx
      do j = ny/2+1,ny-1
        C(i+1,j,k)=C(i-1,j,k)
```

```
c
         tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
         tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
         tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
         C(i,j,k+1) = C(i,j,k)*tmp1
c
         tmp1 = z(k)**3*z(k-1)*C(i,j,k-1)/
     &       ((z(k+1)-z(k-1))*(z(k)-z(k-1)))
         C(i,j,k+1) = C(i,j,k+1) - tmp1
c
         tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &       + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
         tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
         c(i,j,k+1) = C(i,j,k+1) - tmp1
c
         tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &       + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
         tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
         c(i,j,k+1) = C(i,j,k+1) - tmp1
c
         C(i,j,k+1) = C(i,j,k+1)*(z(k+1)-z(k-1))*(z(k+1)-z(k))
     &       /(z(k)**3*z(k+1))
c
         f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
         f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
         f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j-1))*(x(i)-x(i-1))
         fluxz = fluxz + f(i,j,k)
      enddo
c
      k=nz/2
      i=nx/2
      do j = ny/2+1,ny-1
         f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
         f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
         f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j-1))*(x(i+1)-x(i))
         fluxz = fluxz + f(i,j,k)
      enddo
c
c hjørner
      k=nz/2
      i=nx/2
      j=ny/2
c
      f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
      f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
      f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j))*(x(i+1)-x(i))
      fluxz = fluxz + f(i,j,k)
c
      k=nz/2
```

```
      i=nx
      j=ny/2
c
      f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
      f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
      f(i,j,k)=f(i,j,k)*0.25*(y(j+1)-y(j))*(x(i)-x(i-1))
      fluxz = fluxz + f(i,j,k)
c
      k=nz/2
      i=nx
      j=ny
c
      C(i+1,j,k)=C(i-1,j,k)
      C(i,j+1,k)=C(i,j-1,k)
c
      tmp1 =        x(i)**4/((x(i+1)-x(i))*(x(i)-x(i-1)))
      tmp1 = tmp1 + y(j)**4/((y(j+1)-y(j))*(y(j)-y(j-1)))
      tmp1 = tmp1 + z(k)**4/((z(k+1)-z(k))*(z(k)-z(k-1)))
      C(i,j,k+1) = C(i,j,k)*tmp1
c
      tmp1 = z(k)**3*z(k-1)*C(i,j,k-1)/
     &    ((z(k+1)-z(k-1))*(z(k)-z(k-1)))
      C(i,j,k+1) = C(i,j,k+1) - tmp1
c
      tmp1 = x(i+1)*C(i+1,j,k)/(x(i+1)-x(i))
     &    + x(i-1)*C(i-1,j,k)/(x(i)-x(i-1))
      tmp1 = tmp1*x(i)**3/(x(i+1)-x(i-1))
      c(i,j,k+1) = C(i,j,k+1) - tmp1
c
      tmp1 = y(j+1)*C(i,j+1,k)/(y(j+1)-y(j))
     &    + y(j-1)*C(i,j-1,k)/(y(j)-y(j-1))
      tmp1 = tmp1*y(j)**3/(y(j+1)-y(j-1))
      c(i,j,k+1) = C(i,j,k+1) - tmp1
c
      C(i,j,k+1) = C(i,j,k+1)*(z(k+1)-z(k-1))*(z(k+1)-z(k))
     &    /(z(k)**3*z(k+1))
c
      f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
      f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
      f(i,j,k)=f(i,j,k)*0.25*(y(j)-y(j-1))*(x(i)-x(i-1))
      fluxz = fluxz + f(i,j,k)
c
      k=nz/2
      i=nx/2
      j=ny
c
      f(i,j,k)=(C(i,j,k+1)-C(i,j,k-1))/(z(k+1)-z(k-1))
      f(i,j,k)=f(i,j,k)*z(k)**2/x(i)**2/y(j)**2
      f(i,j,k)=f(i,j,k)*0.25*(y(j)-y(j-1))*(x(i+1)-x(i))
```

```
      fluxz = fluxz + f(i,j,k)
c
c________________________________________
c     symmetrifaktor 8 og lx=0.5:
      flux=8.*0.5*(fluxx+fluxy+fluxz)
      write(*,'(i7,1p,6E10.2)')
     &     count, err/nx/ny/nz, errmax, fluxx, fluxy, fluxz, flux
      write(16,'(i7,1p,6E14.4)')
     &     count, err/nx/ny/nz, errmax, fluxx, fluxy, fluxz, flux
      If(stopp.ne.1) Goto 100
c
c utskrift
c
      open(10,file="LaplaceNytrans.txt")
      write(10,*)nx,ny,nz
      do i=1,nx
        do j=1,ny
           do k=1,nz
              if((i.gt.nx/2).and.(j.gt.ny/2).and.(k.gt.nz/2))
     &              C(i,j,k)=1.
              write(10,'(1p,4E14.4)') C(i,j,k)
           enddo
           write(10,*)
        enddo
        write (10,*)
      enddo
      write (10,*)
      close(10)
      close (14)
      end
```