SIMULATION AND MODELING OF PHASE TRANSITION

In this project we are trying to model the crystal structure of a two phase solid.

Assumptions : There are only two phases in the solid and those two phases have some dislocations(represented by transformation matrix A).

Aim : Our aim is to minimize the energy functional in order to find the optimal crystal structure, given the total number of atoms and the two types of phases.

ENERGY FUNCTIONAL

$$F = F1 + F2 + F3 + F4$$

where

$$F1 = \int_{\Omega} \sum_{i} W(A(x)(y_i - X_y(x))) * \eta(\tfrac{y_i - x}{\delta})dx$$

$$F2 = \int_{\Omega} tr((id - A^t A)^2)dx$$

$$F3 = \int_{\Omega} \Psi(\sum_{i} \eta(\tfrac{y_i - x}{\delta}) - \frac{\det(A)}{v/n} * \int_{0}^{\delta} 2\pi r \eta(\tfrac{r}{\delta})dr)dx$$

$$F4 = \int_{\Omega} \sum_{i \neq j} \hat{\phi}(|y_i - y_j|)dx$$

where

A is the dislocation transformation matrix, it is the inverse of the product of all the shear, expansion and compression etc to which the crystal structure is subject to. So in order to transform the atomic positions to a more analysable form we translate it to appropriate position by X_y and then transform it by A.

η is a function s.t.

$$\eta(x) = \begin{cases} 1 & \forall x < 0.5 \\ 0 & \forall x > 1 \\ -\tfrac{16}{11}x^3 + \tfrac{36}{11}x^2 - \tfrac{48}{11}x + \tfrac{28}{11} & \forall 0.5 < x < 1 \end{cases}$$

In the range 0.5 to 1 I took η such that it satisfies:

$$\eta(\tfrac{1}{2}) = 1, \eta(1) = 0, \eta'(\tfrac{1}{2}) = 0, \eta'(1) = 0$$

to be fairly smooth at the ends.

x is any point on the domain(two dimensional in our case).

y_i are the positions of the atoms in space.

δ is the interaction neighbourhood, we consider that after this distance the interaction between the atoms is negligible.

ANALYSING F1

In this we penalise the η-sum of the measure of dislocation of all the atoms at every point in the domain.

W is a measure of how far the configuration of the atoms(after taking the dislocation transformation) is from the actual crystal structure.We need to calculate beforehand that which atoms are in which phase.

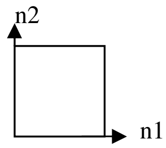
In our example we are considering a two dimensional crystal configuration whose two phases are :

1.Cubic structure

2.Hexagonal structure

Since we just want to penalise the amount by which the atomic positions are off the perfect crystal lattice positions, the function $\tau(x) = 1 + \sin(2\pi(x - \tfrac{1}{4}))$ will be helpful.

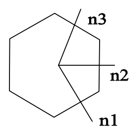
Cubic



$$W(x) = \tfrac{1}{2} * (\tau(\frac{x.n1}{h1^{\perp}}) + \tau(\frac{x.n2}{h1^{\perp}}))$$

where $n1 = \hat{i}$ and $n2 = \hat{j}$

Hexagonal



$$W(x) = \tfrac{1}{3} * (\tau(\frac{x.n1}{h2^{\perp}}) + \tau(\frac{x.n2}{h2^{\perp}}) + \tau(\frac{x.n3}{h2^{\perp}}))$$

where $n1 = \tfrac{1}{2}\hat{i} - \tfrac{\sqrt{3}}{2}\hat{j}$, $n2 = \hat{i}$ and $n3 = \tfrac{1}{2}\hat{i} + \tfrac{\sqrt{3}}{2}\hat{j}$

Deciding about the phase at a point in the domain Ω and calculating $X_y(x)$ is done simultaneously since these are interrelated.
Let us say it is phase 1 at point x

Let
$f_1 = \sum_i W(A(x)(y_i - X_y)) * \eta(\frac{y_i - x}{\delta})$ be the function in consideration ,

since we want to minimize this functional so we differentiate it w.r.t. X_y and equate to 0

$$\partial_{X_y} f_1 = \sum_i -A^t * \nabla W(A(x)(y_i - X_y(x))) * \eta(\tfrac{y_i - x}{\delta}) = 0$$

then solve this nonlinear system using newton-raphson to get X_y .

i.e. $\partial^2_{X_y} f_1 X_y = -\partial_{X_y} f_1$ where

$$\partial^2_{X_y} f_1 = \sum_i A^t \partial^2 W A * \eta(\tfrac{y_i - x}{\delta})$$

now we calculate $\sum_i W(A(x)(y_i - X_y(x))) * \eta(\frac{y_i - x}{\delta})$ assuming phase 1 all over(actually only at point x) and then with phase 2 .If its value is less with phase 1 we are done otherwise calculate $X_y(x)$ using phase 2 and then check again.If again there is a contradiction then assume the phase 1 if the difference in values of the above function was less than that after recalculation with phase 2.

<u>ANALYSING F2</u>

It penalises the measure of deformation of the crystal structure(transformation matrix A) other than rotation. If A is just a rotation matrix then F2 is 0 since if A is a pure rotation matrix then its transpose is also a rotation matrix but in the opposite direction so their product is an identity matrix. This function is just a measure of the difference between $A^t A$ and Id .

ANALYSING F3

It helps in avoiding holes in the solution crystal structure as it takes care of the fact that in every δ -neighbourhood , the $\eta-$weighted difference, between the no of atoms actually present and what should be there in perfect crystal, should not be more.

$\sum_i \eta(\frac{y_i - x}{\delta})$ is the η-weighted no of atoms actually present.

For calculating the η-weighted no of atoms present in perfect crystal:

$$\eta(x) = \int_0^\infty \chi_{\eta(x)>\sigma} d\sigma \quad \text{where } \chi_{cond} = \begin{cases} 1 & if \quad cond \\ 0 & otherwise \end{cases}$$

required value= $\sum_i \eta(\frac{x_i - x}{\delta})$ where x_i is the actual lattice position transformed by $(A(x))^{-1}$

$$\therefore \sum_i \eta(\frac{x_i - x}{\delta}) = \sum_i \int_0^\infty \chi_{\eta(z_i)>\sigma} d\sigma \quad \text{where } z_i = \frac{x_i - x}{\delta}$$

$$= \int_0^\infty \sum_i \chi_{\eta(z_i)>\sigma} d\sigma$$

let B^σ be the ball s.t. for every atom inside it $\eta(z_i) > \sigma$

$$\sum_i \chi_{\eta(z_i)>\sigma} = \text{no of atoms inside } B^\sigma = \frac{\det(A).areaOfB^\sigma}{areaCoveredBy1atom}$$

Let r be the radius of B^σ

$$\therefore \eta\left(\tfrac{r}{\delta}\right) = \sigma \Rightarrow \eta'\left(\tfrac{r}{\delta}\right)\tfrac{1}{\delta} dr = d\sigma$$

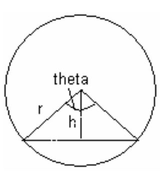
putting these things in the above equation and integrating by parts we get

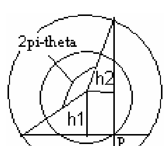
$$\sum_i \eta(\tfrac{x_i - x}{\delta}) = \frac{\det(A)}{vol\_per\_atom} * \int_0^\delta 2\pi r \eta(\tfrac{r}{\delta}) dr$$

Putting the value of η in the above expression, we get

$$\sum_i \eta(\tfrac{x_i - x}{\delta}) = \frac{\det(A)}{vol\_per\_atom} * \frac{38\pi}{55} \delta^2$$

if the δ -ball intersects the boundary, then we should integrate $\eta(\frac{r}{\delta})$ only on the area inside the boundary.





Owing tho the dimensions, the delta ball can intersect atmost two lines.In the case shown in first figure, theta(θ)=$\cos^{-1}$(h/r) and the integral becomes

Integral= $\int_0^\delta 2\pi r \eta(\frac{r}{\delta}) dr - \int_h^\delta \theta r \eta(\frac{r}{\delta}) dr$, if we have two non-intersecting chords subtending angles θ_1 and θ_2 then

Integral= $\int_0^\delta 2\pi r \eta(\frac{r}{\delta}) dr - \int_{h_1}^\delta \theta_1 r \eta(\frac{r}{\delta}) dr - \int_{h_2}^\delta \theta_2 r \eta(\frac{r}{\delta}) dr$

For case shown in second figure, we first integrate inside the point p(radius r1) which is same as the first case with two chords and then we integrate from r1 to delta as

$\int_{r_1}^\delta (\pi - \phi + \cos^{-1}(h1/r) + \cos^{-1}(h2/r)) r \eta(\frac{r}{\delta}) dr$ where ϕ is the angle(inside the valid region) between the intersecting boundary lines which in our case is always 90^0 because of square domain.

The only integral evaluation is of the form

$$\int_a^\delta \cos^{-1}(h/r)\, r\,\eta(\tfrac{r}{\delta})\,dr$$

Integrating by parts

$$= c1 - \tfrac{h}{2}\int_a^\delta \frac{2r}{\sqrt{1-(\frac{h}{r})^2}}\left(\tfrac{-16}{11.5}(\tfrac{r}{\delta})^3 + \tfrac{36}{11.4}(\tfrac{r}{\delta})^2 + \tfrac{-48}{11.3}(\tfrac{r}{\delta}) + \tfrac{28}{11.2}\right)dr$$

Again integrating by parts

$$= c2 + h\int_a^\delta \sqrt{r^2-h^2}\left(\tfrac{-16.3}{11.5}(\tfrac{r^2}{\delta^3}) + \tfrac{36.2}{11.4}(\tfrac{r}{\delta^2}) + \tfrac{-48}{11.3\delta}\right)dr \quad \text{where}$$

$$c1 = \cos^{-1}(\tfrac{h}{r})\left(\tfrac{-16}{11.5}(\tfrac{r^5}{\delta^3}) + \tfrac{36}{11.4}(\tfrac{r^4}{\delta^2}) + \tfrac{-48}{11.3}(\tfrac{r^3}{\delta}) + \tfrac{28r^2}{11.2}\right)\Big|_a^\delta$$

$$c2 = c1 - h\sqrt{r^2-h^2}\left(\tfrac{-16}{11.5}(\tfrac{r}{\delta})^3 + \tfrac{36}{11.4}(\tfrac{r}{\delta})^2 + \tfrac{-48}{11.3}(\tfrac{r}{\delta}) + \tfrac{28}{11.2}\right)\Big|_a^\delta$$

There are two main integrals in the above formula

$$\int_a^\delta r^2\sqrt{r^2-h^2}\,dr \text{ ,put } \tfrac{r}{h} = \sec\theta$$

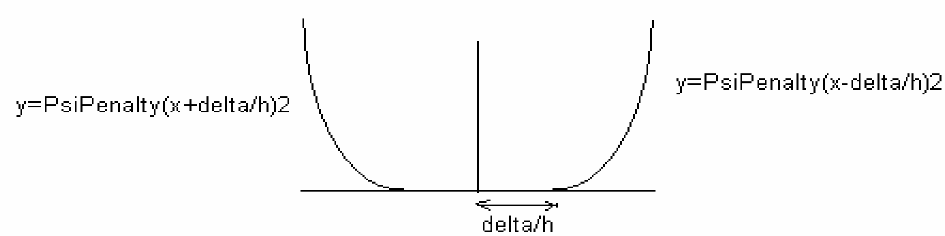
$$= h^4\left[\frac{14\tanh^{-1}(\tan(\theta/2)) + \sec\theta(2\sec^2\theta+7)\tan\theta}{8}\right]_{\sec^{-1}(\frac{a}{h})}^{\sec^{-1}(\frac{\delta}{h})}$$

$$\int \sqrt{r^2-h^2}\,dr = \tfrac{-h^2}{2}\cosh^{-1}(\tfrac{r}{a}) + \tfrac{x}{2}\sqrt{x^2-a^2}$$

Hence the above can be evaluated according to the case.

But if you want to avoid these messy calculations with a little approximation, we can just consider only those delta-balls that are completely inside the domain while calculating F3 and its derivative provided there are no 'silver edges' of the domain but it might lead to bad solution-configuration at the boundary.
Or we have a drichlet boundary condition i.e. we generate atoms outside the domain upto a width equal to the delta, these atoms will remain fixed as being a boundary condition.

Since the error in the calculations is of order δ/h where h is the lattice parameter so Ψ should be 0 upto that and then increase somewhat symmetrically w.r.t. 0.



ANALYSING F4

This term takes care of the fact that no two atoms come very close to each other as in that case their electrostatic potential due to the induced dipole moment will increase highly.

$\hat{\phi}(t) = \beta e^{-\alpha t}$ where α and β are very large constants.
As the parameter of the function we take the euclidean distance between two atoms.

EXAMPLE SITUATION OF PHASE TRANSITION

DOMAIN : A square of size 1x1 with origin at the lower left corner. This domain is discretized by a square grid having 21 points in both the directions so grid parameter $t = \frac{1}{20} = 0.05$.

INITIAL ATOMIC POSITIONS : There are 121 atoms initially placed on a regular square grid over the whole domain. Hence its surface density in the interior is $\frac{1}{(\frac{1}{10})^2}$ (assuming unit atomic weight).

In order to see some results from the program, the density of one of the phases(square) should be lower and and that of other(hex.) should be higher than this.

Density of hexagonal grid(hcp)= $\dfrac{\frac{1}{2}}{\frac{1}{2}(\frac{h_2^{\perp}}{\sin(60^0)})h_2^{\perp}} = \dfrac{\sqrt{3}}{2(h_2^{\perp})^2}$

Density of square grid(ccp)= $\dfrac{1}{(h_1^{\perp})^2}$

Hence $\dfrac{\sqrt{3}}{2(h_2^{\perp})^2} > 100$

i.e. $h_2^{\perp} < 0.093$

let $h_2^{\perp} = 0.09$ then $h_1^{\perp} = \frac{2}{\sqrt{3}} h_2^{\perp} = 0.104$ which satisfy the condition.

INITIAL 'A' MATRIX : Since initially the atomic positions are just a scaled version of perfect square lattice positions so A should be a scaling matrix everywhere in the domain whose off diagonal elements are 0 and the diagonal elements are the scaling in one dimension by which we have to scale to get the perfect lattice sites i.e. $\frac{0.104}{0.1}$

DRICHLET BOUNDARY CONDITION : We can set up the boundary condition by just generating some atomic positions outside the domain upto a distance of delta. We should have some different phase from that inside the doamin in initially so that it is not at a local minima initially as in that case the program might not do anything as it only tries to reach a local minima.

Also we hsould give some nucleation points inside the domain where the phase is different over a small region, these small regions will act as a source of low threshold energy path for the global minima.

Other parameters are set as :
δ =0.2
PsiPenalty=10
α =1000
β =100

MINIMIZING THE FUNCTIONAL

The variables over which this functional is to be minimized are the atomic positions y_i and the transformation matrices $A(x)$ for all $x \in \Omega$.

The variable vector is defined as :

$$\left[y_0^0, y_1^0, y_0^1, y_1^1, \ldots y_0^{noOfY-1}, y_1^{noOfY-1}, A_{00}^0, A_{01}^0, A_{10}^0, A_{11}^0, \ldots, A_{00}^{noOfX-1}, A_{01}^{noOfX-1}, A_{10}^{noOfX-1}, A_{11}^{noOfX-1}\right]$$

For minimization, we approximate the function by its second order taylor series expansion

$$\tilde{F}(X) = F(X_0) + \partial X^t \nabla F(X_0) + (\partial X)^t \nabla^2 F(X_0)(\partial X)$$

For minima, its gradient should be 0

$$\nabla \tilde{F}(X) = \nabla F + \nabla^2 F(\partial X) = 0$$

This system of linear equations we solve using conjugate gradient method for ∂X .

CONJUGATE GRADIENT METHOD

Let we have to solve the system of equation

$$AX_L + b = 0$$

which is equivalent to minimization of the function $\overline{F}(X_L) = X_L{}^t A X_L + b X_L$

The initial solution should be 0 since we have to find the increment outer loop(Newton-Raphson).

Let d_i be the direction at the i^{th} step r_i be the gradient of $\overline{F}$ at the i^{th} step.

$r_i = \nabla\overline{F}(X_{Li}) = AX_{Li} + b$

Let $X_{Li} = X_{Li-1} + \lambda_{i-1}d_{i-1}$

Since λ_i minimizes $\overline{F}$ so its derivative at λ_i should be 0

i.e. $\partial_{\lambda i}\overline{F}(X_{Li} + \lambda_i d_i) = d_i{}^t A(X_{Li} + \lambda_i d_i) + d_i^t b = 0$

i.e. $\lambda_i = -\dfrac{d_i^t(AX_{Li} + b)}{d_i^t A d_i}$

We can also consider minimizing $\overline{F}$ at i+1th step w.r.t. λ_i and λ_{i-1} i.e.

$(X_{i-1} + \lambda_{i-1}d_{i-1} + \lambda_i d_i)^t A(X_{i-1} + \lambda_{i-1}d_{i-1} + \lambda_i d_i) + b^t(X_{i-1} + \lambda_{i-1}d_{i-1} + \lambda_i d_i) \rightarrow \min$

differentiating w.r.t. λ_i and λ_{i-1} we get

$d_{i-1}{}^t AX_{i+1} + b^t d_{i-1} = 0$ --1

and

$d_i{}^t AX_{i+1} + b^t d_i = 0$ --2

also from the ith step

$d_{i-1}{}^t AX_i + b^t d_{i-1} = 0$ --3

subtracting 3 from 1, we get

$\lambda_i d_{i-1} A d_i = 0$

hence the consecutive directions should be A-orthogonal, so we take

$$d_i = r_i - d_{i-1}\frac{r_i^t A d_{i-1}}{d_{i-1}Ad_{i-1}}$$

The first step is just the gradient step so $d_0 = r_0$

Since we need to minimize the function so at every iteration of the conjugate gradient step we should check that $X_{Lk}{}^t\nabla^2 F(X_0)X_{Lk} > 0$ otherwise we should update the X and evaluate the $\nabla^2 F$ and ∇F at the new point to minimize the error.

The second order approximation of the function F is good as far as we are getting a value lower than the previous value in each iteration of the outer loop(Newton) otherwise we have to do a trust region step.

TRUST REGION METHOD

Since errors might accumulate up due to second order taylor approximationand there might be case when value of function starts increasing instead of decreasing, we have to do a third order approximation and find the optimal point in the direction of the increment given by the conjugate gradient.

Let $\Phi(s) = F(X_0 + s\partial X)$,we do its third order taylor series approximation wrt s.

$\tilde{\Phi}(s) = \Phi(0) + \Phi'(0)s + \tfrac{1}{2}\Phi''(0)s^2 + \tfrac{1}{6}\Phi'''(0)s^3$

For minima

$\tilde{\Phi}'(s) = \Phi'(0) + \Phi''(0)s + \tfrac{1}{2}\Phi'''(0)s^2 = 0$ and

$\tilde{\Phi}''(s) = \Phi''(0) + \Phi'''(0)s > 0$

i.e. $s = \dfrac{-\Phi''(0) \pm \sqrt{[\Phi''(0)]^2 - 2\Phi'''(0)\Phi'(0)}}{\Phi'''(0)}$ and

$s > -\dfrac{\Phi''(0)}{\Phi'''(0)}$

where

$$\Phi'(0) = \nabla F(X_0)(\partial X)^t$$

$$\Phi''(0) = (\partial X)^t \nabla^2 F(X_0)(\partial X)$$

Either we can explicitly evaluate the third derivative

$$\Phi'''(0) = \sum_{i,j,k=1}^{noOfVar} \nabla^3 F(X_0)_{/i,j,k} (\partial X)_{/i} (\partial X)_{/j} (\partial X)_{/k}$$

or we can approximate it as $\Phi'''(0) = \frac{\Phi''(\varepsilon) - \Phi''(0)}{\varepsilon}$

Even after this if the value of the functional is not decreasing, we should successively reduce the increments by truncating each dimension of the increments by a certain number(γ) and divide γ by 4 to further reduce the region.

Another approach is we approximate Φ by

$$\tilde{\Phi}(s) = \Phi(0) + \Phi'(0)s + \tfrac{1}{2}\Phi''(0)s^2 + C\,|\,s\,|^3$$

where C is a rough estimate for Φ'''

initially C can be approximated by $100h^{-3}\,|\,\partial Y\,|^3 + 100\,|\,\partial A\,|^3$

and evaluate the function value, if it doesn't decrease instead of taking all the Y's in $|\,\partial Y\,|^3$ we should add only those Y's d2F w.r.t. whom is negative i.e. we should check the diagonal blocks of rank equal to the dimension of the domain of d2F corresponding to Y's and check whether they are positive definite or notand take those which are not.

If again it doesn't decrease, we should change the C by dividing by 4 and then evaluate the function value to see if it is decreased.

PRECONDITIONING CONJUGATE GRADIENT

Preconditioning means changing variable or maybe the number of variables also, the new variables are some linear combination of the original variables. So preconditioned variable vector can be formed by multiplying the old variable vector by a matrix.

In our case the equation to be solved is $\nabla^2 F(\partial X) = -\nabla F$, let the new variable space is Y so $\partial X = CY$ where C is the precondition matrix. Wecan convert gradient and second derivative into the new variable space by multiplying C appropriately.

So the equation becomes

$$C^t \nabla^2 FCX = -C^t \nabla F$$

1.For Atomic Positions

Since the change in atomic positions is a combination of the actual atomic change and the change due to change in the transformation matrix A so we subtract the change due to A to form the new variables. Hence we need to subtract $(A^k)^{-1} \Delta A^k (y_i - X_y{}^k)$ from each y_i where k is the index of the nearest point to y_i in the discretization of the domain.

So preconditioning matrix for the atomic positions is given by:

$$C^y = \begin{bmatrix} 100----0 \frac{a_{11}^k (y_0^0 - X_{y0}^k)}{|A|} \frac{a_{11}^k (y_1^0 - X_{y1}^k)}{|A|} \frac{-a_{01}^k (y_0^0 - X_{y0}^k)}{|A|} \frac{-a_{01}^k (y_1^0 - X_{y1}^k)}{|A|} 0---0 \\ 010----0 \frac{-a_{10}^k (y_0^0 - X_{y0}^k)}{|A|} \frac{-a_{10}^k (y_1^0 - X_{y1}^k)}{|A|} \frac{a_{00}^k (y_0^0 - X_{y0}^k)}{|A|} \frac{a_{00}^k (y_1^0 - X_{y1}^k)}{|A|} 0---0 \\ ---------------------------------- \\ \end{bmatrix}$$

2.BPX(hierarchical) Preconditioning for the transformation matrices

During the minimization process, the atomic positions might come outside the domain so we have to truncate the increments.

For numerical solvability we descretize the domain into squares of size h which are the ‘voronoi regions’ of the particular discrete point and we assume the function to be piecewise constant over the region of a point.
The solution vector consists of the coordinates of all the atoms and the transformation matrices at all the points in the discretised domain.

For calculating the ∇F :

$$\partial_{y_i} F1 = \int_\Omega A^t(x)\nabla W(A(x)(y_i - X_y))\eta(\tfrac{y_i - x}{\delta}) + \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - x}{\delta})W(A(x)(y_i - X_y))dx$$

$$\partial_{y_i} F3 = \int_\Omega \Psi'(\sum_i \eta(\tfrac{y_i - x}{\delta}) - \frac{\det(A)}{vol\_per\_atom} * \frac{38\pi}{55}\delta^2)\tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - x}{\delta})dx$$

$$\partial_{y_i} F4 = \int_\Omega \sum_{j\neq i}\hat{\phi}'(|y_i - y_j|)\frac{(y_i - y_j)}{|y_i - y_j|}dx$$

$$\partial_A F1_{/B(x)} = \sum_i \nabla W(A(x)(y_i - X_y))^t B(y_i - X_y)\eta(\tfrac{y_i - x}{\delta})dx$$

$$\partial_A F2_{/B(x)} = tr(-2A^tB - 2B^tA + B^tAA^tA + A^tBA^tA + A^tAB^tA + A^tAA^tB)dx$$

$$\partial_A F3_{/B(x)} = -\Psi' * \frac{\det(A)tr(A^{-1}B)}{v/n} * \frac{38\pi}{55}\delta^2 dx$$

For calculating $\nabla^2 F$

$$\partial_{y_i y_i} F1 = \int_\Omega A^t\nabla^2 WA\ \eta(\tfrac{y_i - x}{\delta}) + A^t\nabla W\ \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - x}{\delta})^t + \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - x}{\delta})\nabla W^t A + \tfrac{1}{\delta^2}\partial^2\eta(\tfrac{y_i - x}{\delta})Wdx$$

$$\partial_{y_i y_j} F3 = \int_\Omega \Psi''\tfrac{1}{\delta^2}\nabla\eta(\tfrac{y_i - x}{\delta})\nabla\eta(\tfrac{y_j - x}{\delta})^t dx$$

$$\partial_{y_i y_i} F3 = \int_\Omega \Psi''\tfrac{1}{\delta^2}\nabla\eta(\tfrac{y_i - x}{\delta})\nabla\eta(\tfrac{y_i - x}{\delta})^t + \Psi'\tfrac{1}{\delta^2}\partial^2\eta(\tfrac{y_i - x}{\delta})dx$$

$$\partial_{y^k{}_i y^l{}_j} F4 = \int_\Omega \hat{\phi}''(|y_i - y_j|)\frac{(y_j^l - y_i^l)(y_i^k - y_j^k)}{|y_i - y_j|^2} + \hat{\phi}'(|y_i - y_j|)\frac{|y_i - y_j|^2\frac{-\partial y_j^k}{\partial y_j^l} - (y_i^k - y_j^k)(y_j^l - y_i^l)}{|y_i - y_j|^3}dx$$

$$\partial_{y^k{}_i y^l{}_i} F4 = \int_\Omega \sum_{j\neq i}\hat{\phi}''(|y_i - y_j|)\frac{(y_i^l - y_j^l)(y_i^k - y_j^k)}{|y_i - y_j|^2} + \hat{\phi}'(|y_i - y_j|)\frac{|y_i - y_j|^2\frac{\partial y_i^k}{\partial y_i^l} - (y_i^l - y_j^l)(y_i^k - y_j^k)}{|y_i - y_j|^3}dx$$

$$\partial^2{}_A F1_{/B_1(x)B_2(x)} = \sum_i (B^1(y_i - X_y))^t\nabla^2 W(B^2(y_i - X_y))\eta(\tfrac{y_i - x}{\delta})dx$$

$$\partial^2{}_A F2_{/B_1(x)B_2(x)} = tr(-2B_2^tB_1 - 2B_1^tB_2 + B_1^tB_2A^tA + B_1^tAB_2^tA + B_1^tAA^tB_2 + B_2^tB_1A^tA + A^tB_1B_2^tA + A^tB_1A^tB_2$$
$$+ B_2^tAB_1^tA + A^tB_2B_1^tA + A^tAB_1^tB_2 + B_2^tAA^tB_1 + A^tB_2A^tB_1 + A^tAB_2^tB_1)dx$$

$$\partial_A^2 F3_{/B_1(x)B_2(x)} = \Psi'' * (\frac{\det(A)}{v/n} * \frac{38\pi}{55}\delta^2)^2 tr(A^{-1}B_1)tr(A^{-1}B_2) +$$
$$\Psi' * \frac{\det(A)}{v/n} * \frac{38\pi}{55}\delta^2(A^{-1}B_1^tA^{-1}{}_{/B_2} - tr(A^{-1}B_1)tr(A^{-1}B_2))dx$$

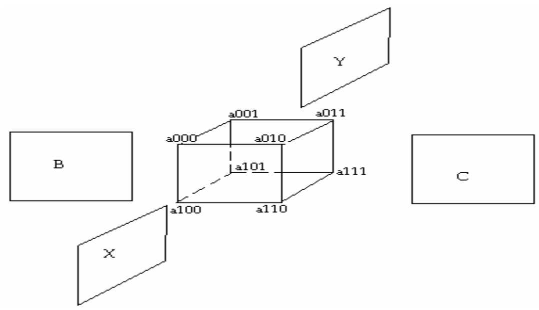
$$\partial_{Ay_i} F1_{/B(x)} = (A^t\nabla^2 W^t B(y_i - X_y) + B^t\nabla W)\eta(\tfrac{y_i - x}{\delta}) + \nabla W^t B(y_i - X_y)\tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - x}{\delta})dx$$

$$\partial_{Ay_i} F3_{/B(x)} = -\Psi'' \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i-x}{\delta})\frac{\det(A)tr(A^{-1}B)}{v/n} * \frac{38\pi}{55}\delta^2 dx$$

For calculating $\nabla^3 F$

NOTATION

Since $\nabla^3 F$ has 2^3 elements and three dimensions so it can be thought of having a cubical structure(3-dimensional hypercube) or a box and we can multiply other matrices to it in the same way as we multiply matrices with matrices but here we can multiply a matrix with a box in 4 different ways i.e. left, right, back and front.



Each matrix can be multiplied to box by multiplying the matrix to the two matrices(faces) of box in the plane of the given matrix separately but if any of the matrices is box or any other portion of box then if it is in position of B or C then its front face is multiplied with front face of box and its back face with back of box, using normal matrix multiplication and likewise in the other direction. The $\perp$ sign over any vector indicates that it has been rotated by 90^0 to lie along the third direction(into the plane of paper), if this sign is over a second or higher derivative then this means that the last differential was done along the direction into the plane of paper so the vector should be places properly e.g. $\nabla^2\eta(\frac{y_i-X_y}{\delta})^{t\perp}$ has the same configuration as the top of the box. And the above structure can be written as

$$B\underset{X}{\overset{Y}{A}}C$$

using this notation the formulae for third derivative of F are written as under

$$\begin{aligned}
\partial_{y_iy_iy_i}F1 = \int_\Omega\, & A^t\,\overset{A}{\nabla^3 W}\,A\eta(\tfrac{y_i-X_y}{\delta}) + \tfrac{1}{\delta}A^t\nabla^2 WA\nabla\eta(\tfrac{y_i-X_y}{\delta})^{t\perp} + \tfrac{1}{\delta}\nabla^2\underset{A^t}{\overset{A}{W}}{}^{\perp}\nabla\eta(\tfrac{y_i-X_y}{\delta})^t + \\
& \tfrac{1}{\delta^2}A^t\nabla W\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^{t\perp} + \tfrac{1}{\delta^2}\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^{\perp}\nabla W^t A + \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla^2 W^{t\perp}A + \\
& \tfrac{1}{\delta^3}\nabla^3\eta(\tfrac{y_i-X_y}{\delta})W + \tfrac{1}{\delta^2}\nabla^2\eta(\tfrac{y_i-X_y}{\delta})\nabla\overset{A}{W}{}^{t\perp}\, dx
\end{aligned}$$

$$\partial_{y_iy_jy_k}F3 = \int_\Omega \Psi'''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla\eta(\tfrac{y_j-X_y}{\delta})^t\nabla\eta(\tfrac{y_k-X_y}{\delta})^{t\perp}dx$$

$$\partial_{y_iy_jy_i}F3 = \int_\Omega \Psi'''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla\eta(\tfrac{y_j-X_y}{\delta})^t\nabla\eta(\tfrac{y_i-X_y}{\delta})^{t\perp} + \Psi''\tfrac{1}{\delta^3}\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^{\perp}\nabla\eta(\tfrac{y_j-X_y}{\delta})^t dx$$

$$\partial_{y_iy_jy_j}F3 = \int_\Omega \Psi'''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla\eta(\tfrac{y_j-X_y}{\delta})^t\nabla\eta(\tfrac{y_j-X_y}{\delta})^{t\perp} + \Psi''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})^{\perp}\nabla^2\eta(\tfrac{y_j-X_y}{\delta})^{t\perp}dx$$

$$\partial_{y_iy_iy_j}F3 = \int_\Omega \Psi'''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla\eta(\tfrac{y_i-X_y}{\delta})^t\nabla\eta(\tfrac{y_j-X_y}{\delta})^{t\perp} + \Psi''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_j-X_y}{\delta})^{t\perp}\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^t dx$$

$$\begin{aligned}
\partial_{y_iy_iy_i}F3 = \int_\Omega\, & \Psi'''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla\eta(\tfrac{y_i-X_y}{\delta})^t\nabla\eta(\tfrac{y_i-X_y}{\delta})^{t\perp} + \Psi''\tfrac{1}{\delta^3}\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^{\perp}\nabla\eta(\tfrac{y_i-X_y}{\delta})^t + \\
& \Psi''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})\nabla^2\eta(\tfrac{y_i-X_y}{\delta})^{t\perp} + \Psi''\tfrac{1}{\delta^3}\nabla\eta(\tfrac{y_i-X_y}{\delta})^{t\perp}\nabla^2\eta(\tfrac{y_i-X_y}{\delta}) + \Psi'\tfrac{1}{\delta^3}\nabla^3\eta(\tfrac{y_i-X_y}{\delta})dx
\end{aligned}$$

$$
\partial_{y_i^k y_i^l y_i^m} F4 = \int_\Omega \sum_{j\neq i}
\begin{array}{l}
\hat{\phi}''' \dfrac{(y_i^k - y_j^k)(y_i^l - y_j^l)(y_i^m - y_j^m)}{\left|y_i - y_j\right|^3} + \\
\hat{\phi}'' \dfrac{\left|y_i - y_j\right|^2 \left[(y_i^l - y_j^l)\frac{\partial y_i^k}{\partial y_i^m} + (y_i^k - y_j^k)\frac{\partial y_i^l}{\partial y_i^m}\right] - 2(y_i^k - y_j^k)(y_i^l - y_j^l)(y_i^m - y_j^m)}{\left|y_i - y_j\right|^4} \\
+ \hat{\phi}'' \dfrac{(y_i^m - y_j^m)\left[\left|y_i - y_j\right|^2 \frac{\partial y_i^k}{\partial y_i^l} - (y_i^k - y_j^k)(y_i^l - y_j^l)\right]}{\left|y_i - y_j\right|^4} + \\
\hat{\phi}' \left[\dfrac{-\frac{\partial y_i^k}{\partial y_i^l}(y_i^m - y_j^m)}{\left|y_i - y_j\right|^3} - \dfrac{\left|y_i - y_j\right|^2 \left[(y_i^l - y_j^l)\frac{\partial y_i^k}{\partial y_i^m} + (y_i^k - y_j^k)\frac{\partial y_i^l}{\partial y_i^m}\right] - 3(y_i^k - y_j^k)(y_i^l - y_j^l)(y_i^m - y_j^m)}{\left|y_i - y_j\right|^5} \right]
\end{array}
dx
$$

$$
\partial_{y_i^k y_i^l y_j^m} F4 = \int_\Omega
\begin{array}{l}
\hat{\phi}''' \dfrac{(y_i^k - y_j^k)(y_i^l - y_j^l)(y_j^m - y_i^m)}{\left|y_i - y_j\right|^3} + \\
\hat{\phi}'' \dfrac{\left|y_i - y_j\right|^2 \left[(y_i^l - y_j^l)\frac{-\partial y_j^k}{\partial y_j^m} + (y_i^k - y_j^k)\frac{-\partial y_j^l}{\partial y_j^m}\right] - 2(y_i^k - y_j^k)(y_i^l - y_j^l)(y_j^m - y_i^m)}{\left|y_i - y_j\right|^4} \\
+ \hat{\phi}'' \dfrac{(y_j^m - y_i^m)\left[\left|y_i - y_j\right|^2 \frac{\partial y_i^k}{\partial y_i^l} - (y_i^k - y_j^k)(y_i^l - y_j^l)\right]}{\left|y_i - y_j\right|^4} + \\
\hat{\phi}' \left[\dfrac{-\frac{\partial y_i^k}{\partial y_i^l}(y_j^m - y_i^m)}{\left|y_i - y_j\right|^3} - \dfrac{\left|y_i - y_j\right|^2 \left[(y_i^l - y_j^l)\frac{-\partial y_j^k}{\partial y_j^m} + (y_i^k - y_j^k)\frac{-\partial y_j^l}{\partial y_j^m}\right] - 3(y_i^k - y_j^k)(y_i^l - y_j^l)(y_j^m - y_i^m)}{\left|y_i - y_j\right|^5} \right] dx
\end{array}
$$

$$
\partial_A^3 F1_{/B_1 B_2 B_3} = \sum_i (B_1 y_i)^t \overset{(B_3 y_i)}{\nabla^3 W} (B_2 y_i)\, \eta(\tfrac{y_i - X_y}{\delta})\, dx
$$

$$
\partial_A^3 F2_{/B_1 B_2 B_3} = tr\begin{pmatrix}
B_1^t B_2 B_3^t A + B_1^t B_2 A^t B_3 + B_1^t B_3 B_2^t A_3 + B_1^t A\, B_2^t B_3 + B_1^t B_3 A^t B_2 + B_1^t A\, B_3^t B_2 + \\
B_2^t B_1 B_3^t A \ + B_2^t B_1 A^t B_3 + B_3^t B_1 B_2^t A \ + A^t B_1 B_2^t B_3 + B_3^t B_1 A^t B_2 + A^t B_1 B_3^t B_2 + \\
B_2^t B_3 B_1^t A \ + B_2^t A\, B_1^t B_3 + B_3^t B_2 B_1^t A \ + A^t B_2 B_1^t B_3 + B_3^t A\, B_1^t B_2 + A^t B_3 B_1^t B_2 + \\
B_2^t B_3 A^t B_1 + B_2^t A\, B_3^t B_1 + B_3^t B_2 A^t B_1 + A^t B_2 B_3^t B_1 + B_3^t A\, B_2^t B_1 + A^t B_3 B_2^t B_1 +
\end{pmatrix} dx
$$

$$
\partial_{y_i y_i A} F2_{/B} = A^t \overset{B(y_i - X_y)}{\nabla^3 W} A\,\eta(\tfrac{y_i - X_y}{\delta}) + A^t \overset{B(y_i - X_y)}{\nabla^2 W^\perp} \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta})^t + B^t \nabla W \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta}) +
$$

$$
B^t \nabla^2 W A\,\eta(\tfrac{y_i - X_y}{\delta}) + A^t \nabla^2 W B\,\eta(\tfrac{y_i - X_y}{\delta}) + \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta}) \overset{B(y_i - X_y)}{\nabla^2 W^{t\perp}} A + \tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta}) \nabla W^t B +
$$

$$
\tfrac{1}{\delta^2}\nabla^2\eta(\tfrac{y_i - X_y}{\delta}) \overset{B(y_i - X_y)}{\nabla W^{t\perp}}\, dx
$$

$$\partial_{y_i y_j A} F3_{/B} = \Psi'''(\tfrac{-\det(A)tr(A^{-1}B)}{v/n}\int_0^{\delta} 2\pi r\eta(\tfrac{r}{\delta})dr)\tfrac{1}{\delta^2}\nabla\eta(\tfrac{y_i - X_y}{\delta})\nabla\eta(\tfrac{y_j - X_y}{\delta})^t\,dx$$

$$\partial_{y_i y_j A} F3_{/B} = \Psi'''(\tfrac{-\det(A)tr(A^{-1}B)}{v/n}\int_0^{\delta} 2\pi r\eta(\tfrac{r}{\delta})dr)\tfrac{1}{\delta^2}\nabla\eta(\tfrac{y_i - X_y}{\delta})\nabla\eta(\tfrac{y_i - X_y}{\delta})^t +$$

$$\Psi''(\tfrac{-\det(A)tr(A^{-1}B)}{v/n}\int_0^{\delta} 2\pi r\eta(\tfrac{r}{\delta})dr)\tfrac{1}{\delta^2}\nabla^2\eta(\tfrac{y_i - X_y}{\delta})dx$$

$$\partial_{AAy_i} F1_{/B_1B_2} = \sum_i (B_1 y_i)^t \overset{A}{\nabla^3} W(B_2 y_i)\eta(\tfrac{y_i - X_y}{\delta})dx$$

$$\partial_{AAy_i} F3_{/B_1B_2} = \Psi'''\tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta})^{t\perp}(\tfrac{\det(A)}{v/n}\int_0^{\delta} 2\pi r\eta(\tfrac{r}{\delta})dr)^2 tr(A^{-1}B_1)tr(A^{-1}B_2) +$$

$$\Psi''\tfrac{1}{\delta}\nabla\eta(\tfrac{y_i - X_y}{\delta})^{t\perp}(\tfrac{\det(A)}{v/n}\int_0^{\delta} 2\pi r\eta(\tfrac{r}{\delta})dr)(A^{-1}B_1^t A^{-1}{}_{/B_2} - tr(A^{-1}B_1)tr(A^{-1}B_2))dx$$

where B's are the direction matrices in which all the elements are 0 except the one with respect to which we have to differentiate.

Since we assume every function to be piecewise constant over the discretised domain so we replace the integral sign in the above equations with summation over all the discretiation points of domain and dx with the area of the corresponding voronoi region inside the domain.

Computational aspects:
As we see in the above formulas particualrly in F1 and F3 that we have to evaluate the functions on the delta ball for each discretization point of domain, so we can save some computation power on spatial locality grounds i.e. we can maintain a datastructure which gives the information of all the atoms present in a specified region. For that we have a 2-d array structure(better for near rectangular domains) whose 1st and 2nd dimension correspond to the x and y direction of the domain and we discretise the domain into small rectangles, the number of rectangles along a direction is equal to the number of elements in the array in that dimension. In this way the rectangles are mapped to corresponding to element in the array i.e. all the points of the discretization of the domain that come in a particular rectangle are there in that element in the form of a linked list whose initial pointer is given by the respective element of the array.