Body Fitted Grids and Finite Difference Method

The Finite Volume method on rectangular grids can be generalised to complex geometries in two ways:
- by directly adapting the finite volume method to arbitrarily shaped cells,
- or, by considering derivatives along curved gridlines and transforming the Navier Stokes equations according to this new reference frame.

We start with this second method which has the advantage of exhibiting the new problems associated with complex grids, namely the loss of accuracy and stability if the grid is too distorted. Most conclusions regarding mesh quality will also apply to the finite volume method on non-orthogonal grids.

Consider the grid around an elliptic body generated by:

\[
x(\xi, \eta) = sh \xi \cos(\omega \eta) \\
y(\xi, \eta) = h \xi \sin(\omega \eta)
\]

where \( h \) and \( \omega \) are constants. \( s \) is the ratio of the horizontal axis to the vertical axis. The plot below is for \( s = 2, h = \frac{1}{10} \).

With finite differences, variables are stored at nodes where gridlines intersect, corresponding to \( \xi \) and \( \eta \) taking integer values. The programme will contain arrays for pressure, velocity etc. corresponding to those nodes. Here, \( f(i, j) \) is the value of the continuous variable \( f(x, y) \) when \( \xi = i, \eta = j \), that is at point \( x = rh \xi \cos(\omega \eta) \) and \( y = h \xi \sin(\omega \eta) \). In other words, in the computer programme \( f(i, j) = f(\xi = i, \eta = j) \) is the value stored in the array representing values of \( f \) at each grid-point. Similarly the grid-point coordinates are stored in a array: \( x(i, j) = x(\xi = i, \eta = j) \). Note that \( \xi = i \) only at nodes. \( \xi \) is a real number indicating a continuous displacement along the gridlines, whereas \( i \) is an integer.

To sum up, we consider two families of gridlines defined by:

\[
\xi = \xi(x, y) = Cst \quad ; \quad \eta = \eta(x, y) = Cst \quad \text{#}
\]

and we have \( f(x, y) = f(\xi(x, y), \eta(x, y)) = f(\xi, \eta) \) everywhere, but: \( f(x, y) = f(i, j) = f_{ij} \) "is known in the computer programme" only when \( (x, y) \) corresponds to a grid node.

The derivatives of the variable with respect to the gridlines are easily approximated by:
\[ \left( \frac{\partial f}{\partial \xi} \right)_{ij} = \frac{f_{i+1,j} - f_{i-1,j}}{\xi_{i+1,j} - \xi_{i-1,j}} = \frac{f_{i+1,j} - f_{i-1,j}}{2} \]

since \( \xi_{ij} = i \) and \( \xi_{i-1,j} = i - 1 \)

However the equations of the physical phenomenon are given in the Cartesian frame \((x, y)\), so we need to transform the derivatives by the chain rule:

\[
\begin{align*}
\frac{\partial f(\xi, \eta)}{\partial x} &= \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial x} \\
\frac{\partial f(\xi, \eta)}{\partial y} &= \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial y}
\end{align*}
\]

How can we compute \( \frac{\partial \xi}{\partial x} \)? For simple geometries, the mapping \((\xi, \eta) \mapsto (x, y)\) is given by algebraic expressions (e.g. polar coordinates) but for complex geometries the mesh is given as sets of grid points \( x_{ij} = x(\xi = i, \eta = j) \).

From this we can apply finite differences to compute:

\[
\left( \frac{\partial x}{\partial \xi} \right)_{ij} = \frac{x_{i+1,j} - x_{i-1,j}}{\xi_{i+1,j} - \xi_{i-1,j}} = \frac{x_{i+1,j} - x_{i-1,j}}{2}
\]

but what we actually want is the reverse relations giving us \( \left( \frac{\partial \xi}{\partial x} \right) \). Generalizing to 3 dimensions, \((\xi_1, \xi_2, \xi_3) \mapsto (x_1, x_2, x_3)\), we are seeking:

\[
\begin{bmatrix}
\frac{\partial f}{\partial x_1} \\
\frac{\partial f}{\partial x_2} \\
\frac{\partial f}{\partial x_3}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_3}{\partial x_1} \\
\frac{\partial \xi_1}{\partial x_2} & \frac{\partial \xi_2}{\partial x_2} & \frac{\partial \xi_3}{\partial x_2} \\
\frac{\partial \xi_1}{\partial x_3} & \frac{\partial \xi_2}{\partial x_3} & \frac{\partial \xi_3}{\partial x_3}
\end{bmatrix} \begin{bmatrix}
\frac{\partial f}{\partial \xi_1} \\
\frac{\partial f}{\partial \xi_2} \\
\frac{\partial f}{\partial \xi_3}
\end{bmatrix}
\]

Or, using vector-matrix the notations for the previous relation, we seek \( T \) in:

\[ \nabla f = T \tilde{\nabla} f \]

while what we actually know is \( A \):

\[ \tilde{\nabla} f = A \nabla f \]

Obviously, the transformation matrix \( T \) is simply given by \( T = A^{-1} \), that is, we need to inverse matrix:

\[
A = \begin{bmatrix}
\frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_1} \\
\frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_2} \\
\frac{\partial x_1}{\partial \xi_3} & \frac{\partial x_2}{\partial \xi_3} & \frac{\partial x_3}{\partial \xi_3}
\end{bmatrix}
\]

To invert the matrix \( A \), for each entry \( a_{ij} \) of \( A \), let \( c_{ij} \) be the cofactor of \( a_{ij} \), \( c_{ij} = (-1)^{i+j}D_{ij} \), where \( D_{ij} \) is the determinant of the \((n - 1) \times (n - 1)\) matrix obtained from \( A \) by omitting the \( i^{th} \) row and \( j^{th} \) column of \( A \).

The adjunct matrix of \( A \) is defined to be the \( n \times n \) matrix whose entry in the \( i^{th} \) row and \( j^{th} \) column is \( c_{ji} \) (transpose rows and columns).
\[ T = A^{-1} = \frac{1}{J} \cdot \text{adj} A \]

\( J \) is the determinant of \( A \). In the present context of the transformation \((x_1, x_2, x_3) \rightarrow (\xi_1, \xi_2, \xi_3)\), \( J \) is called the Jacobian of the transformation.

Instead of directly using the inverse transform \( T = A^{-1} \), it is usual practice to leave out the Jacobian factor (because simplifications occur, and also \( J \) measures the transform of volumes.

- Hence we define the coefficients:
  \[ T_{kl} = \frac{1}{J} \beta_{kl} \]

  Thus we have
  \[ \frac{\partial f}{\partial x_1} = \frac{1}{J} \left( \frac{\partial f}{\partial \xi_j} \beta_{1j} \right) = \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \]

  Beware that \( \beta_{ij} \) is not symmetric, summation with \( \frac{\partial f}{\partial \xi_j} \) applies to the second or "column" index.

The convection term poses no difficulties since we can write:
\[ u_i \frac{\partial f}{\partial x_i} = u_i \frac{1}{J} \left( \frac{\partial f}{\partial \xi_k} \beta_{ik} \right) = \left( u_i \frac{1}{J} \beta_{ik} \right) \frac{\partial f}{\partial \xi_k} = \bar{u}_k \frac{\partial f}{\partial \xi_k} \]

That is, the convection term in the transformed space retains the usual form once we have computed the convecting velocity in the transformed space: \( \bar{u}_k = u_i \frac{1}{J} \beta_{ik} \) (summation with the first or "row" index) and we can then apply e.g. the QUICK scheme in the transformed space.

**Transformation of a Laplacian:**

The diffusion terms become quite complex. To show this let us expand only the first term:
\[ \frac{\partial f}{\partial x_1} = \frac{1}{J} \left( \frac{\partial f}{\partial \xi_m} \beta_{1m} \right) = \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \]
\[ \frac{\partial}{\partial x_1} \gamma \frac{\partial f}{\partial x_1} = \frac{\partial}{\partial x_1} \left( \gamma \frac{1}{J} \sum_{k=1}^{3} \left( \frac{\partial f}{\partial \xi_k} \beta_{1k} \right) \right) \]
\[ = \frac{\partial}{\partial x_1} \left( \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \right) \]
\[ = \frac{1}{J} \sum_{j=1}^{3} \left( \frac{\partial}{\partial \xi_j} \left[ \gamma \frac{1}{J} \sum_{k=1}^{3} \left( \frac{\partial f}{\partial \xi_k} \beta_{1k} \right) \right] \beta_{1j} \right) \]
\[ = \frac{1}{J} \frac{\partial}{\partial \xi_1} \left[ \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \right] \beta_{11} \]
\[ + \frac{1}{J} \frac{\partial}{\partial \xi_2} \left[ \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \right] \beta_{12} \]
\[ + \frac{1}{J} \frac{\partial}{\partial \xi_3} \left[ \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} \beta_{11} + \frac{\partial f}{\partial \xi_2} \beta_{12} + \frac{\partial f}{\partial \xi_3} \beta_{13} \right) \right] \beta_{13} \]
The generic term is \( \frac{\partial^2 f}{\partial \xi_i \partial \xi_k} \beta_{1k} \beta_{ij} \) where the first index of \( \beta \)'s is one, because we only considered the second derivative on the first \( (\xi_1) \) direction in the diffusion operator. The diffusion operator will also have \( \frac{\partial}{\partial x_2} \gamma \frac{\partial f}{\partial x_2} \) in the Cartesian frame, which will lead to generic term \( \frac{\partial^2 f}{\partial \xi_j \partial \xi_k} \beta_{2k} \beta_{2j} \). The full diffusion terms can be written:

\[
\frac{\partial}{\partial x_i} \gamma \frac{\partial f}{\partial x_i} = \frac{1}{J} \left( \frac{\partial}{\partial \xi_j} \left[ \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_k} \beta_{ik} \right) \right] \beta_{ij} \right)
\]

The right hand side has three implicit summation on \( k, j \) and \( i \), leading to 27 terms, but there are only 9 derivative terms such as \( \frac{\partial^2 f}{\partial \xi^2} \), so one can regroup terms by summing on the \( i \) index first.

For instance consider \( \frac{\partial^2 f}{\partial \xi_2 \partial \xi_3} \beta_{12} \beta_{13} \). When summing up with the other 2 terms, \( \frac{\partial}{\partial x_2} \gamma \frac{\partial f}{\partial x_2} \) and \( \frac{\partial}{\partial x_3} \gamma \frac{\partial f}{\partial x_3} \), we can expect to find additional contributions such as \( \frac{\partial^2 f}{\partial \xi_2 \partial \xi_3} \beta_{22} \beta_{23} \) and \( \frac{\partial^2 f}{\partial \xi_2 \partial \xi_3} \beta_{32} \beta_{33} \).

If we now introduce:

\[
B_{ij} = \beta_{ik} \beta_{il} = \beta_{1k} \beta_{1l} + \beta_{2k} \beta_{2l} + \beta_{3k} \beta_{3l}
\]

(note that the summation is on the first index and this is important because \( \beta_{ik} \) is not symmetrical, but \( B \) is symmetrical: \( B_{ij} = B_{ji} \))

then, the complete diffusion term can be rearranged as:

\[
\sum_{j=1}^{3} \frac{\partial}{\partial x_j} \gamma \frac{\partial f}{\partial x_i} = \frac{1}{J} \frac{\partial}{\partial \xi_1} \left( \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} B_{11} + \frac{\partial f}{\partial \xi_2} B_{21} + \frac{\partial f}{\partial \xi_3} B_{31} \right) \right)
\]

\[
+ \frac{1}{J} \frac{\partial}{\partial \xi_2} \left( \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} B_{12} + \frac{\partial f}{\partial \xi_2} B_{22} + \frac{\partial f}{\partial \xi_3} B_{32} \right) \right)
\]

\[
+ \frac{1}{J} \frac{\partial}{\partial \xi_3} \left( \gamma \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} B_{13} + \frac{\partial f}{\partial \xi_2} B_{23} + \frac{\partial f}{\partial \xi_3} B_{33} \right) \right)
\]

The \( \beta \) and \( B \) coefficients are calculated only once at the beginning of the Navier Stokes solver, or within the mesh generator since these geometrical coefficients are constant, not time dependant.

We have seen that the diffusion term should be calculated implicitly in order to gain stability and second order accuracy, but the system above contains cross derivatives such as \( \frac{\partial}{\partial \xi_1} \frac{\partial f}{\partial \xi_2} B_{21} \) which contribute to off diagonal terms in the final linear system to be solved by an iterative method, and this makes the convergence all the more slower when these terms are large. Indeed, to discretise this term, one needs first the evaluate \( \frac{\partial f}{\partial \xi_2} \) on the two neighbouring nodes along the \( \xi_1 \) line, and then take their difference as shown below.
Nodes involved in computing cross-derivative \( \frac{\partial}{\partial \xi_1} \frac{\partial f}{\partial \xi_2} \)

\[
\left[ \frac{\partial}{\partial \xi_1} \frac{\partial f}{\partial \xi_2} \right]_{ij} \approx \frac{(f_{i+1,j+1} - f_{i+1,j-1})/2 - (f_{i-1,j+1} - f_{i-1,j-1})/2}{2}
\]

The above expression is involved in the evaluation of the diffusion at node \((i,j)\) yet doesn't involve the value of \(f\) at this node. In other words, if one assembles the matrix corresponding to only these terms, then it would have zeros on the diagonal and could not be inverted. Fortunately when the gridlines are orthogonal the off diagonal coefficients such as \(B_{21}\) are zero, but this is not always easy to impose. Also when we introduce:

\[
\left( \frac{\partial x}{\partial \xi} \right)_{ij} = \frac{x_{i+1,j} - x_{i-1,j}}{\zeta_{i+1,j} - \zeta_{i-1,j}}
\]

this is second order accurate only when point \(x_{i,j}\) is in the middle of \(x_{i+1,j}\) and \(x_{i-1,j}\). When the grid is very irregular there is a risk of introducing more finite difference errors in evaluating the geometrical coefficients \(\frac{\partial x}{\partial \xi}\) than for the gradients of the physical variables \(\frac{\partial f}{\partial \xi}\) themselves.

**Recommendations:**

To conclude, although the above presentation shows that it is fairly simple to apply the standard FD methods in the transformed space, which can then be mapped back to any complex geometry, one must be aware that strong skewing or stretching of the mesh can introduce large errors and convergence difficulties. It is thus very important to take care during the mesh generation to avoid irregular mesh spacings (only gradually increase mesh step a a few percent from one cell to the next), and to avoid far from orthogonal grid-lines.

With this precaution, the body-fitted FD method works fine for simple geometries such as flow around a cylinder or a wing profile. However ensuring high quality of grids is not always possible, for example mapping a square mesh in to a circular pipe inevitably introduces 60° angles, or a multi-element airfoil can lead to conflicting constraints on the gridlines.

To resolve this dilemma, advanced or industrial methods combine a range of cell shapes (rectangles, triangles and even general polygonal cells). These grids are then called "unstructured" since the structure or simple correspondence \(\xi = i, \eta = j\) is lost.

Discretisation is then introduced either by finite elements which will again use the Jacobian transformation extensively, or by direct finite volume discretisation on the polygonal cells. However, distorted cells, or rapidly varying cell sizes also rapidly degrade the quality of
the FV discretisation, and to some extent this also applies to finite elements.
Example

Consider the grid generated by

\[
x_1 = R \frac{\xi_1}{10} \cos(\frac{\pi \xi_2}{10})
\]

\[
x_2 = R \frac{\xi_1}{10} \sin(\frac{\pi \xi_2}{10})
\]

for \(\xi_1\) and \(\xi_2\) varying from 0 to 10. When \(\xi_2\) varies while \(\xi_1\) is kept constant, this generates semi-circular gridlines around a half cylinder of radius \(R \frac{\xi_1}{10}\). These are the \(\xi_1\)-constant gridlines. Similarly the \(\xi_2\)-constant gridlines correspond to rays.

\[
A = \begin{bmatrix}
\frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} \\
\frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2}
\end{bmatrix}
= \begin{bmatrix}
\frac{R}{10} \cos(\frac{\pi \xi_2}{10}) & \frac{R}{10} \sin(\frac{\pi \xi_2}{10}) \\
-R \frac{\xi_1}{10} \pi \sin(\frac{\pi \xi_2}{10}) & R \frac{\xi_1}{10} \pi \cos(\frac{\pi \xi_2}{10})
\end{bmatrix}

\]

\[
J = \det \left[ \begin{array}{cc}
\frac{R}{10} \cos(\frac{\pi \xi_2}{10}) & \frac{R}{10} \sin(\frac{\pi \xi_2}{10}) \\
-R \frac{\xi_1}{10} \pi \sin(\frac{\pi \xi_2}{10}) & R \frac{\xi_1}{10} \pi \cos(\frac{\pi \xi_2}{10})
\end{array} \right] = \frac{R}{10} \frac{\xi_1}{10} \pi \left( \cos^2 + \sin^2 \right) = \frac{R}{10} \frac{\xi_1}{10} \pi
\]

Let \(h = \frac{R}{10}\) the radial step, and \(\omega = \frac{\pi}{10}\) the angular step, \(r = \frac{\xi_1}{10} R\) the radial position, and \(\alpha = (\pi \frac{\xi_2}{10})\).

\(J = h(r, \omega)\) is the mesh cell surface in \((x_1, x_2)\) space corresponding to the cell generated by taking unit increments from point \((\xi_1, \xi_2)\). Then:

\[
A^{-1} = \frac{1}{h \omega} \begin{bmatrix}
ro \cos \alpha & -h \sin \alpha \\
ro \sin \alpha & h \cos \alpha
\end{bmatrix}
\]

\[
\beta_{11} \quad \beta_{12} \\
\beta_{21} \quad \beta_{22}
\]

\[
= \begin{bmatrix}
ro \cos \alpha & -h \sin \alpha \\
ro \sin \alpha & h \cos \alpha
\end{bmatrix}
\]

Using:

\[
B_{kl} = \beta_{kl} = \beta_{1k} \beta_{l1} + \beta_{2k} \beta_{l2}
\]

\[
B_{11} = \beta_{11} \beta_{11} + \beta_{21} \beta_{21} = (ro \cos \alpha)^2 + (ro \sin \alpha)^2 = (ro)^2
\]

\[
B_{22} = \beta_{12} \beta_{12} + \beta_{22} \beta_{22} = (h \sin \alpha)^2 + (h \cos \alpha)^2 = h^2
\]

\[
B_{12} = \beta_{11} \beta_{12} + \beta_{21} \beta_{22} = roh \sin \alpha \cos \alpha - roh \sin \alpha \cos \alpha = 0
\]

\[
\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} = \frac{1}{J} \frac{\partial}{\partial \xi_1} \left( \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} B_{11} + \frac{\partial f}{\partial \xi_2} B_{21} \right) \right) + \frac{1}{J} \frac{\partial}{\partial \xi_2} \left( \frac{1}{J} \left( \frac{\partial f}{\partial \xi_1} B_{12} + \frac{\partial f}{\partial \xi_2} B_{22} \right) \right)
\]

\[
J = h \omega
\]
\[
\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} = \frac{1}{h \omega} \frac{\partial}{\partial \xi_1} \left( \frac{1}{h \omega} f \omega, 0 \right) + \frac{1}{h \omega} \frac{\partial}{\partial \xi_2} \left( \frac{1}{f} \left( 0 + \frac{\partial f}{\partial \xi_2} h^2 \right) \right) \tag{#}
\]
\[
\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} = \frac{1}{h \omega} \frac{\partial}{\partial \xi_1} \left( \frac{\omega}{h} \frac{\partial f}{\partial \xi_1} \right) + \frac{1}{h \omega} \frac{\partial}{\partial \xi_2} \left( \frac{\omega}{h} \frac{\partial f}{\partial \xi_2} \right) \tag{#}
\]
\[
= \frac{1}{h^2} \frac{\partial^2 f}{\partial \xi_1 \partial \xi_1} + \frac{1}{h} \frac{\partial r}{\partial \xi_1} \frac{\partial f}{\partial \xi_1} + \frac{1}{(\rho \omega)^2} \frac{\partial^2 f}{\partial \xi_2 \partial \xi_2} \tag{#}
\]
\[
\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} = \frac{1}{h^2} \frac{\partial^2 f}{\partial \xi_1 \partial \xi_1} + \frac{1}{h^2} \frac{\partial^2 f}{\partial \xi_1 \partial \xi_2} + \frac{1}{h^2} \frac{\partial^2 f}{\partial \xi_2 \partial \xi_2} + \frac{1}{(\rho \omega)^2} \frac{\partial^2 f}{\partial \xi_2 \partial \xi_2} \tag{#}
\]

Remark 1: This is comparable to the Laplacian in cylindrical coordinates:
\[
\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} + \frac{1}{r^2} \frac{\partial^2 f}{\partial \theta^2} \tag{#}
\]

since \(dr = \frac{\rho}{\alpha} d\xi_1 = h d\xi_1\) and \(d\theta = \omega d\xi_2\)

Remark 2: There are no cross derivative terms such as \(\frac{\partial^2 f}{\partial \xi_1 \partial \xi_2}\) because the gridlines are locally orthogonal to each other.

The other method to obtain the laplacian:
\[
A^{-1} = \frac{1}{h} \left[ \begin{array}{cc} \frac{1}{h} \cos \alpha & -\frac{1}{\omega \rho} \sin \alpha \\ \frac{1}{h} \sin \alpha & \frac{1}{\omega \rho} \cos \alpha \end{array} \right] = \left[ \begin{array}{cc} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_2}{\partial x_1} \\ \frac{\partial \xi_1}{\partial x_2} & \frac{\partial \xi_2}{\partial x_2} \end{array} \right] \tag{#}
\]

Remark 3: We can also apply twice the chain rule:
\[
\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial \xi_1} \frac{\partial \xi_1}{\partial x_1} + \frac{\partial f}{\partial \xi_2} \frac{\partial \xi_2}{\partial x_1} \tag{#}
\]
\[
\frac{\partial f}{\partial x_1} = \frac{1}{h} \cos \alpha \frac{\partial f}{\partial \xi_1} - \frac{1}{\omega \rho} \sin \alpha \frac{\partial f}{\partial \xi_2} = g \tag{#}
\]
\[
\frac{\partial g}{\partial x_1} = \frac{1}{h} \cos \alpha \frac{\partial g}{\partial \xi_1} - \frac{1}{\omega \rho} \sin \alpha \frac{\partial g}{\partial \xi_2} \tag{#}
\]
\[
\frac{\partial}{\partial x_1} \left( \frac{\partial f}{\partial x_1} \right) = \frac{1}{h} \cos \alpha \frac{\partial}{\partial \xi_1} \left( \frac{1}{h} \cos \alpha \frac{\partial f}{\partial \xi_1} - \frac{1}{\omega \rho} \sin \alpha \frac{\partial f}{\partial \xi_2} \right) \tag{#}
\]
\[
- \frac{1}{\omega \rho} \sin \alpha \frac{\partial}{\partial \xi_2} \left( \frac{1}{h} \cos \alpha \frac{\partial f}{\partial \xi_1} - \frac{1}{\omega \rho} \sin \alpha \frac{\partial f}{\partial \xi_2} \right) \tag{#}
\]

Then similarly for \(\frac{\partial^2 f}{\partial x_2^2}\) and adding the result, one obtains the laplacian.