Streamline integration as a method for two-dimensional elliptic grid generation

M. Wiesenberger, M. Held, and L. Einkemmer

February 16, 2021

Abstract

[This writeup is based/copied on the recent article [6]]
We propose a new numerical algorithm to construct a structured numerical elliptic grid of a doubly connected domain. Our method is applicable to domains with boundaries defined by two contour lines of a two-dimensional function. Furthermore, we can adapt any analytically given boundary aligned structured grid, which specifically includes polar and Cartesian grids. The resulting coordinate lines are orthogonal to the boundary. Grid points as well as the elements of the Jacobian matrix can be computed efficiently and up to machine precision. In the simplest case we construct conformal grids, yet with the help of weight functions and monitor metrics we can control the distribution of cells across the domain. Our algorithm is parallelizable and easy to implement with elementary numerical methods. We assess the quality of grids by considering both the distribution of cell sizes and the accuracy of the solution to elliptic problems. Among the tested grids these key properties are best fulfilled by the grid constructed with the monitor metric approach.

1 High precision elliptic grid generation

Given is a function $\psi(x, y)$ in Cartesian coordinates. We want to construct a grid on the region bounded by the two lines $\psi(x, y) = \psi_0$ and $\psi(x, y) = \psi_1$ with $\psi_0 \neq \psi_1$. The derivative of the function ψ may not vanish within this region and on the boundary and we further assume that the region is topologically a ring. Note here that this excludes the description of domains with an X-point (saddle point) or O-point (local extremum). The numerical grid is described by a mapping of the discretization of the rectangular computational domain $(u, v) \in [0, u_1] \times [0, 2\pi]$ to the physical domain (x, y). u_1 is an unknown, which the grid generation process has to provide.

As mentioned in the introduction and illustrated in Fig. 1 the idea of our algorithm are two consecutive coordinate transformations constructed by streamline integration. We denote the first transformation with $x(\zeta, \eta)$, $y(\zeta, \eta)$, where the ζ coordinate is aligned with ψ and η is an angle-like coordinate. The solution of the elliptic equation on this flux-aligned coordinate system is denoted with $\bar{u}(\zeta, \eta)$. The second coordinate transformation is then denoted $\zeta(u, v)$, $\eta(u, v)$ with u aligned to $\bar{u}$.

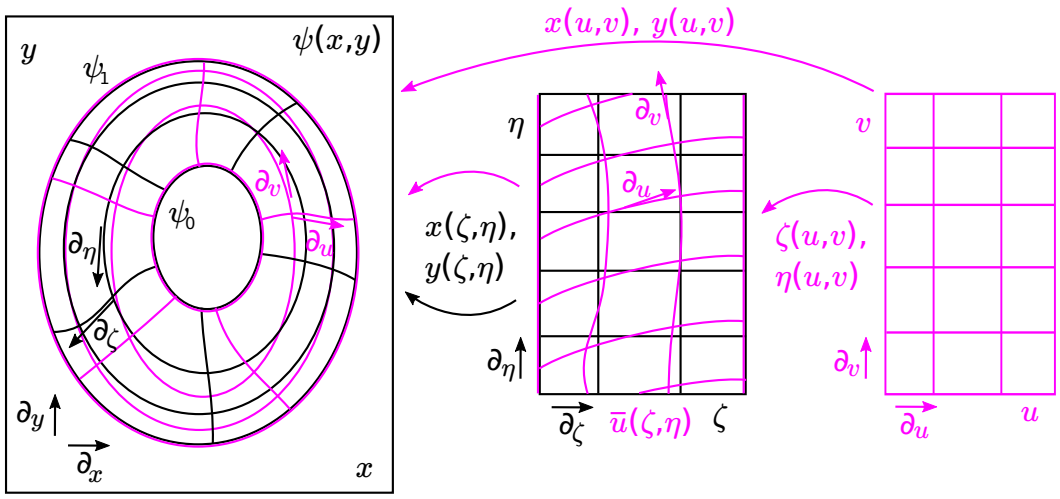

Figure 1: Sketch of the coordinate systems, coordinate lines and basis vector fields involved in our method.

When deriving the algorithm we use basic methods and notational conventions of differential geometry. Here, we recommend the excellent Reference [2] as an introduction to the topic. We do so since in this approach the separate roles of the metric tensor, the coordinate system and its base vector fields are very clear. This is paramount for a concise description of our method.

In this Section we first show how the basis vector fields can be integrated to construct coordinate lines with the example of orthogonal coordinates in Section 1.2. This follows an introduction of some helpful quantities, our notation and streamlines in Section 1.1. In the next step we transform, discretize and solve an elliptic equation on the flux aligned coordinate system. The solution $\bar{u}(\zeta,\eta)$ then takes the role of a new flux function in the flux aligned coordinates. We can therefore repeat the streamline integration method in the flux aligned coordinate system in order to construct coordinate lines of the final u, v coordinates. We discuss three examples of this method. In Section 1.3 we consider the simple Laplace and the Cauchy-Riemann equations (**??**) in order to construct conformal coordinates. In Section 1.4 we modify the elliptic equation to allow grid adaption, before we introduce the monitor metric in Section 1.5. Finally, we present our algorithm in Section 1.6.

1.1 Preliminaries

In a curvilinear coordinate system (ζ,η) the components of the metric tensor and its inverse take the form

$$\boldsymbol{g}(\zeta,\eta) = \begin{pmatrix} g_{\zeta\zeta} & g_{\zeta\eta} \\ g_{\zeta\eta} & g_{\eta\eta} \end{pmatrix} \quad \boldsymbol{g}^{-1}(\zeta,\eta) = \begin{pmatrix} g^{\zeta\zeta} & g^{\zeta\eta} \\ g^{\zeta\eta} & g^{\eta\eta} \end{pmatrix} \tag{1}$$

We denote $g = \det \boldsymbol{g} = g_{\zeta\zeta} g_{\eta\eta} - g_{\zeta\eta}^2$. For Cartesian coordinates (x, y) the elements of the inverse metric tensor are transformed by

$$g^{\zeta\zeta} = \zeta_x^2 + \zeta_y^2 \tag{2a}$$
$$g^{\zeta\eta} = \zeta_x \eta_x + \zeta_y \eta_y \tag{2b}$$
$$g^{\eta\eta} = \eta_x^2 + \eta_y^2 \tag{2c}$$

since $g^{xx} = g^{yy} = 1$ and $g^{xy} = 0$. Given $\zeta(x, y)$, $\eta(x, y)$ and its inverse $x(\zeta, \eta)$, $y(\zeta, \eta)$ recall that their Jacobian matrices are related by

$$\begin{pmatrix} x_\zeta & x_\eta \\ y_\zeta & y_\eta \end{pmatrix} \Bigg|_{\zeta(x,y),\eta(x,y)} = \frac{1}{\zeta_x \eta_y - \zeta_y \eta_x} \begin{pmatrix} \eta_y & -\zeta_y \\ -\eta_x & \zeta_x \end{pmatrix} \Bigg|_{x,y} \tag{3}$$

With the rules of tensor transformation it is easy to prove that the element of the volume form $\sqrt{g}$ is related to the Jacobian via

$$\sqrt{g} = (x_\zeta y_\eta - y_\zeta x_\eta) = (\eta_y \zeta_x - \eta_x \zeta_y)^{-1} \tag{4}$$

The components of the gradient operator and the divergence in an arbitrary coordinate system read

$$(\nabla f)^i = g^{ij} \partial_j f \tag{5a}$$
$$\nabla \cdot \boldsymbol{A} = \frac{1}{\sqrt{g}} \partial_i \left(\sqrt{g} A^i \right), \tag{5b}$$

where we sum over repeated indices $i, j \in \{\zeta, \eta\}$ and define $\partial_\zeta \equiv \partial/\partial\zeta$, $\partial_\eta \equiv \partial/\partial\eta$. Finally, we introduce the geometrical poloidal angle

$$\theta(x, y) = \begin{cases} +\arccos\left(\frac{x - x_0}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \right) & \text{for } y \geq y_0 \\ -\arccos\left(\frac{x - x_0}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} \right) & \text{for } y < y_0 \end{cases} \tag{6}$$

such that the differential 1-form

$$\mathrm{d}\theta = -\frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2} \mathrm{d}x + \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} \mathrm{d}y, \tag{7}$$

where (x_0, y_0) is any point inside the region bounded by ψ_0.

Streamline integration is the central part of our algorithm. Recall that given a vector field $v(x, y) = v^x(x, y)\partial_x + v^y(x, y)\partial_y$ its streamlines are given by the equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v^x(x, y)|_{x(t),y(t)} \tag{8a}$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} = v^y(x, y)|_{x(t),y(t)} \tag{8b}$$

where t is a parameter. Recall here that in differential geometry the directional derivatives ∂_x and ∂_y are the base vector fields of the coordinate system[1]. Now recall that if we have a function $f(x,y)$ such that $f(x(t),y(t))$ is a one-to-one map from t to f we can re-parameterize Eq. (8) by

$$\frac{\mathrm{d}x}{\mathrm{d}f} = \frac{\mathrm{d}x/\mathrm{d}t|_{t(f)}}{\mathrm{d}f/\mathrm{d}t|_{t(f)}} = \left.\frac{v^x(x,y)}{(v^x\partial_x f + v^y\partial_y f)(x,y)}\right|_{x(f),y(f)} \tag{9a}$$

$$\frac{\mathrm{d}y}{\mathrm{d}f} = \frac{\mathrm{d}y/\mathrm{d}t|_{t(f)}}{\mathrm{d}f/\mathrm{d}t|_{t(f)}} = \left.\frac{v^y(x,y)}{(v^x\partial_x f + v^y\partial_y f)(x,y)}\right|_{x(f),y(f)} \tag{9b}$$

Furthermore, the derivative of any function $g(x,y)$ along the streamlines of v parameterized by f reads

$$\left.\frac{\mathrm{d}g}{\mathrm{d}f}\right|_{x(f),y(f)} = \left.\frac{(v^x\partial_x g + v^y\partial_y g)(x,y)}{(v^x\partial_x f + v^y\partial_y f)(x,y)}\right|_{x(f),y(f)} \tag{10}$$

Figuratively, in any coordinate system (ζ,η) the 1-forms $\mathrm{d}\zeta$ and $\mathrm{d}\eta$ (the contravariant basis) are visualized by the lines (surfaces in higher dimensions) of constant ζ and η. At the same time the streamlines of the vector fields ∂_ζ and ∂_η (the covariant basis) give the coordinate lines of ζ and η (cf. Fig. 1). For example, if we hold η constant and vary ζ, we go along a streamline of ∂_ζ. This implies that in two dimensions $\mathrm{d}\eta$ and ∂_ζ trace the same line. The vector fields $\nabla\zeta$ and $\nabla\eta$ are associated to $\mathrm{d}\zeta$ and $\mathrm{d}\eta$ through the metric tensor by Eq. (5) and are the vector fields that are everywhere perpendicular to the lines of constant ζ and η, respectively. It is important to realize that in general curvilinear coordinates the vector fields $\nabla\zeta$ and $\nabla\eta$ point in different directions than ∂_ζ and ∂_η. The central point in our algorithm is the realization that once we can express ∂_ζ and ∂_η in terms of ∂_x and ∂_y we can immediately construct the coordinate transformation by integrating streamlines of ∂_ζ and ∂_η using Eq. (8). This holds true even if (x,y) were curvilinear coordinates. The components of the one-forms $\mathrm{d}\zeta$ and $\mathrm{d}\eta$ in terms of $\mathrm{d}x$ and $\mathrm{d}y$ form the elements of the Jacobian matrix of the transformation. These are also necessary in order to transform any tensor (including the metric) from the old to the new coordinate system.

1.2 Orthogonal coordinates

In general, orthogonal coordinates ζ,η with ζ aligned to ψ are described by

$$\mathrm{d}\zeta = \zeta_x\mathrm{d}x + \zeta_y\mathrm{d}y = f(\psi)(\psi_x\mathrm{d}x + \psi_y\mathrm{d}y) \tag{11a}$$

$$\mathrm{d}\eta = \eta_x\mathrm{d}x + \eta_y\mathrm{d}y = h(x,y)(-\psi_y\mathrm{d}x + \psi_x\mathrm{d}y) \tag{11b}$$

With Eq. (11) we have $g^{\zeta\eta} = \zeta_x\eta_x + \zeta_y\eta_y = 0$, $g^{\zeta\zeta} = (\nabla\psi)^2 f^2$, $g^{\eta\eta} = (\nabla\psi)^2 h^2$ and $\sqrt{g}^{-1} = (\nabla\psi)^2 hf$. From Eq. (3) we directly see that the basis vector fields are

$$\partial_\zeta = x_\zeta\partial_x + y_\zeta\partial_y = \frac{1}{(\nabla\psi)^2 f}(\psi_x\partial_x + \psi_y\partial_y) \tag{12a}$$

$$\partial_\eta = x_\eta\partial_x + y_\eta\partial_y = \frac{1}{(\nabla\psi)^2 h}(-\psi_y\partial_x + \psi_x\partial_y) \tag{12b}$$

[1] Some textbooks (e.g. [1]) introduce the notation $\boldsymbol{e}_i := \partial\boldsymbol{x}/\partial x^i$ and $\boldsymbol{e}^i := \nabla x^i$. While this formulation is suitable in many situations we refrain from using it since it mixes the metric into the basis vectors through the use of the gradient (cf. Eq. (5)). This is unpractical for our purposes.

i.e. ∂_ζ points into the direction of the gradient of ψ and ∂_η into the direction of constant $\psi =$ const surfaces. Now, the coordinate system is defined up to the functions $f(\psi)$ and $h(x,y)$. Note that f must be a function of ψ only since the restriction $\mathrm{d}(\mathrm{d}\zeta) = 0$ must hold. Furthermore, $f(\psi) = \mathrm{d}\zeta/\mathrm{d}\psi \neq 0$ is in principle an arbitrary function, yet we choose $f(\psi) = f_0 =$ const. With this choice we directly get

$$\zeta(x,y) = f_0(\psi(x,y) - \psi_0) \tag{13}$$

Note that $\zeta_0 = 0$ and $\zeta_1 = f_0(\psi_1 - \psi_0)$. The up to now undefined function $h(x,y)$ is not arbitrary since $\mathrm{d}(\mathrm{d}\eta) = 0$ must hold. This is the requirement that $\mathrm{d}\eta$ must be a closed form in order for the potential η to exist. We can express this as

$$(\psi_x\partial_x + \psi_y\partial_y)h = f(\nabla\psi)^2\partial_\zeta h = -h\Delta\psi \tag{14}$$

where $\Delta\psi = \psi_{xx} + \psi_{yy}$ is the two-dimensional Laplacian. Let us remark here that Eq. (14) can be written as $\nabla\cdot(h\nabla\psi) = 0$, which makes the orthogonal grid an elliptic grid with adaption function h as becomes evident later. In order to integrate this equation we need an initial condition for h. We choose to first discretize the line given by $\psi(x,y) = \psi_0$.

As already mentioned ∂_η is the vector field the streamlines of which give the coordinate lines for η. We choose $h(x,y) =$ const on ψ_0. To this end we parameterize the coordinate line by θ (cf. Eq. (9))

$$\left.\frac{\mathrm{d}x}{\mathrm{d}\theta}\right|_{\zeta=0} = \frac{x_\eta}{\theta_\eta} = \frac{-\psi_y}{\psi_x\theta_y - \psi_y\theta_x} \tag{15a}$$

$$\left.\frac{\mathrm{d}y}{\mathrm{d}\theta}\right|_{\zeta=0} = \frac{y_\eta}{\theta_\eta} = \frac{\psi_x}{\psi_x\theta_y - \psi_y\theta_x} \tag{15b}$$

$$\left.\frac{\mathrm{d}\eta}{\mathrm{d}\theta}\right|_{\zeta=0} = \frac{1}{\theta_\eta} = \frac{(\nabla\psi)^2 h(\psi_0)}{\psi_x\theta_y - \psi_y\theta_x} \tag{15c}$$

Let us define $h(\psi_0)$ such that $\eta \in [0, 2\pi]$, that is,

$$2\pi = \oint_{\psi=\psi_0} \mathrm{d}\eta = \oint_0^{2\pi} \left.\frac{\mathrm{d}\eta}{\mathrm{d}\theta}\right|_{\zeta=0} \mathrm{d}\theta$$

or

$$f_0 := h(\psi_0) = \frac{2\pi}{\int_0^{2\pi}\mathrm{d}\theta\frac{(\nabla\psi)^2}{\psi_x\theta_y - \psi_y\theta_x}} \tag{16}$$

Here, we also fixed the constant f_0 such that our coordinate system ζ, η fulfills the Cauchy–Riemann condition (**??**) on the boundary line ψ_0. As initial point for the integration of Eq. (15) we can use any point with $\psi(x,y) = \psi_0$. We then use $h(\psi_0)$ on the flux-surface ψ_0 as initial condition for the integration of Eq. (14).

We obtain coordinate lines by integrating the vector fields ∂_ζ and ∂_η given in (12). We start the construction by integrating ∂_η for $\psi = \psi_0$, i.e. $\zeta = 0$. This can be done since $h|_{\psi_0}$ is known.

The obtained points serve as starting points for the integration of $\partial_\zeta = f_0^{-1}\partial_\psi$. In order to get h we simply integrate Eq. (14)

$$\left.\frac{\mathrm{d}x}{\mathrm{d}\zeta}\right|_{\eta=\text{const}} = \frac{\psi_x}{f_0(\nabla\psi)^2} \tag{17a}$$

$$\left.\frac{\mathrm{d}y}{\mathrm{d}\zeta}\right|_{\eta=\text{const}} = \frac{\psi_y}{f_0(\nabla\psi)^2} \tag{17b}$$

$$\left.\frac{\mathrm{d}h}{\mathrm{d}\zeta}\right|_{\eta=\text{const}} = -\frac{\Delta\psi}{f_0(\nabla\psi)^2}h \tag{17c}$$

Note that if $\Delta\psi = 0$, we directly get a conformal grid with this algorithm. This can be seen as then $h(\zeta,\eta) = f_0$. In **??** we briefly study the class of functions ψ that are solutions of the Grad–Shafranov equation and satisfy $\Delta\psi = 0$. This, however, is not true in general.

Let us further remark on the sign of f_0. It is our goal to construct a right handed coordinate system and to have $\zeta_1 > \zeta_0 = 0$. The curves of constant θ surround x_0, y_0 in a mathematically positive direction. That means that Eq. (16) implies that $f_0 > 0$ if $\nabla\psi$ points away from x_0, y_0. If this is not the case, we obtain $f_0 < 0$. On the other hand if ζ should increase from ψ_0 to ψ_1, we need $f_0 < 0$ for $\psi_1 < \psi_0$ and $f_0 > 0$ for $\psi_1 > \psi_0$. We thus simply take the absolute value of Eq. (16) and multiply by -1 if $\psi_1 < \psi_0$. For ease of notation we do so also in the following without further notice.

Let us finally summarize the grid generation in the following algorithm; we assume that the ζ coordinate is discretized by a list of not necessarily equidistant values ζ_i with $i = 0, 1, \dots N_\zeta - 1$ and η is discretized by a list of η_j with $j = 0, 1, \dots N_\eta - 1$:

1. Find an arbitrary point (x, y) with $\psi(x, y) = \psi_0$ and a point x_0, y_0 within the region bound by $\psi(x, y) = \psi_0$ for the definition of θ in Eq. (6)
2. Integrate Eq. (15) with $h = 1$ over $\Theta = [0, 2\pi]$ and use Eq. (16) to compute $f \equiv f_0$ and $h(\psi_0)$. Use any convenient method for the integration of ordinary differential equations.
3. Integrate one streamline of Eq. (12b) with $h = f_0$ from $\eta = 0 \dots \eta_j$ for all j. The result is a list of N_η coordinates $x(0, \eta_j), y(0, \eta_j)$ on the ψ_0 surface.
4. Using this list and $h = f_0$ as starting values integrate Eq. (17) from $\zeta = 0 \dots \zeta_i$ for all i and all η_j. This gives the map $x(\zeta_i, \eta_j), y(\zeta_i, \eta_j)$ as well as $h(\zeta_i, \eta_j)$ for all i and j.
5. Last, using these results and Eq. (11) evaluate the derivatives $\zeta_x(\zeta_i, \eta_j)$, $\zeta_y(\zeta_i, \eta_j)$, $\eta_x(\zeta_i, \eta_j)$, and $\eta_y(\zeta_i, \eta_j)$ for all i and j.

1.3 Conformal coordinates

A conformal mapping $u(x, y), v(x, y)$ has to satisfy the Cauchy–Riemann equations given in Eq. (**??**). A direct consequence is that u and v are harmonic functions

$$\Delta u = \Delta v = 0 \tag{18}$$

Here, $\Delta = \nabla^2$ is the two-dimensional Laplacian with the divergence and gradient operators defined in (5). First, we note that Eq. (18) holds in every coordinate system. Let us assume

that we have constructed flux aligned coordinates (ζ, η). These can be, but not necessarily have to be, the orthogonal coordinates introduced in the last section. Now, in order to construct conformal coordinates u, v we first define

$$u(\zeta, \eta) := c_0(\bar{u}(\zeta, \eta) - \psi_0) \tag{19}$$

and thus

$$\Delta \bar{u}(\zeta, \eta) = 0 \tag{20}$$

where $\bar{u}(0, \eta) = \psi_0$ and $\bar{u}(\zeta_1, \eta) = \psi_1$ fulfills Dirichlet boundary conditions in ζ. In η we have periodic boundary conditions.

Note the analogy between Eq. (19) and Eq. (13). Now, $\bar{u}$ is equal to ψ at the boundaries and its Laplacian vanishes in between. In fact, $\bar{u}$ takes the role of ψ in the following coordinate transformation. We introduce c_0 as a normalization constant with the same role as f_0 in the orthogonal coordinate transformation. Having $\bar{u}(\zeta, \eta)$, our idea is to construct the basis one-forms $\mathrm{d}u$ and $\mathrm{d}v$ in terms of $\mathrm{d}\zeta$ and $\mathrm{d}\eta$ by transforming the Cauchy-Riemann equations to the ζ, η coordinate system. Analogues to the algorithm in Section 1.2 we can then construct the basis vector fields ∂_u and ∂_v, appropriately choose a normalization and then use streamline integration in the ζ, η coordinate system to construct the coordinates. From the basis one-forms $\mathrm{d}u$ and $\mathrm{d}v$ we get the elements of the Jacobian matrix of the transformation. Before we do this in detail however, let us first discuss some alternative elliptic equations to the simple Eq. (18).

1.4 Grid adaption

Although the conformal grid is advantageous for elliptic equations (due to the vanishing metric coefficients) the cell distribution is not very flexible; once the boundary is set the conformal map is unique. We therefore have little control over the distribution of cells. We can use grid adaption techniques to overcome this restriction. The idea is to modify the elliptic equations that u and v have to fulfill. That is, we choose

$$\nabla \cdot \left(\frac{\nabla u}{w} \right) = \nabla \cdot (w \nabla v) = 0 \tag{21}$$

where w is an appropriately chosen weight function. The cell size will be small in regions where w is large and spread out in regions where w is small. The Cauchy–Riemann equations (**??**) are changed accordingly to

$$v_x = -\frac{u_y}{w} \quad v_y = \frac{u_x}{w} \tag{22}$$

Let us remark here that it is straightforward to implement the weight function in the orthogonal grid generation. In Section 1.2 we simply replace the function h by h/w. Then we have

$$\partial_\zeta = \frac{1}{f_0(\nabla\psi)^2}(\psi_x \partial_x + \psi_y \partial_y) \tag{23a}$$

$$\partial_\eta = \frac{w}{h(\nabla\psi)^2}(-\psi_y \partial_x + \psi_x \partial_y) \tag{23b}$$

and $\nabla\psi \cdot \nabla(h/w) = -h/w\Delta\psi$. A suitable choice for w is

$$w = |\nabla\psi| \tag{24}$$

as then the angle-like coordinate η becomes the arc length on the ψ_0 line.

1.5 Monitor metric and the heat conduction tensor

We follow Reference [3, 4, 5] and replace the canonical metric tensor $\boldsymbol{g}$ by a specifically tailored tensor $\boldsymbol{G}$ that takes the form

$$\boldsymbol{G}(x,y) = \boldsymbol{T}\boldsymbol{T} + k^2\boldsymbol{N}\boldsymbol{N} + \varepsilon(x,y)\boldsymbol{I} \tag{25}$$

with $\boldsymbol{T} = (-\psi_y, \psi_x)$ and $\boldsymbol{N} = -(\psi_x, \psi_y)$. The vector $\boldsymbol{T}$ is tangential to the contour lines of ψ while $\boldsymbol{N}$ is normal to it. We have

$$\sqrt{G} = \left[(\varepsilon + k^2(\psi_x^2 + \psi_y^2))(\varepsilon + (\psi_x^2 + \psi_y^2))\right]^{-1/2} \tag{26}$$

where $k < 1$ is a constant and ε is a function that is nonzero in the neighborhood of singularities, i.e. where $\nabla\psi(x,y) = 0$. In our work we choose $k = 0.1$ and $\varepsilon(x,y) \equiv \varepsilon = 0.001$. Note that the scalar product induced by $\boldsymbol{G}$ conserves perpendicularity with respect to $\boldsymbol{T}$, i.e. if and only if any vector $\boldsymbol{v} \perp \boldsymbol{T}$ in the canonical metric, then it is also perpendicular to $\boldsymbol{T}$ in $\boldsymbol{G}$. In our application this is important at the boundary.
Now, we consider the elliptic equation

$$\begin{aligned} \nabla \cdot (\sqrt{G}\boldsymbol{G}\nabla u) &= 0 \\ \partial_x(\sqrt{G}(G^{xx}\partial_x u + G^{xy}\partial_y u)) + \partial_y(\sqrt{G}(G^{yx}\partial_x u + G^{yy}\partial_y u)) &= 0 \end{aligned} \tag{27}$$

with Dirichlet boundary conditions. The resulting grid coordinate u is almost perfectly aligned in regions far away from singularities and breaks the alignment in regions where $|\nabla\psi|$ is small or vanishes.

We now take a slightly more general approach and rewrite Eq. (27)

$$\nabla \cdot (\boldsymbol{\chi}\nabla u) = 0 \tag{28}$$

where $\chi(x,y)$ is a symmetric positive-definite contravariant tensor. Then the conformal grid from Section 1.3, the grid adaption from Section 1.4 as well as the monitor metric can be considered special cases. The grid adaption is recovered by setting $\boldsymbol{\chi} = 1/w\boldsymbol{I}$, while the monitor metric is simply $\boldsymbol{\chi} = \sqrt{G}\boldsymbol{G}$. The true conformal case is, of course, recovered by setting $\boldsymbol{\chi} = \boldsymbol{I}$.

This allows us to provide a commonly known physical interpretation of Eq. (28). If u is a temperature, then the Dirichlet boundary condition fixes a temperature at the boundary of our domain. The tensor $\boldsymbol{\chi}(x,y)$ is then the anisotropic heat conduction tensor and the coordinate lines for u are the isothermal lines of the steady state solution to the heat diffusion problem. This interpretation allows us to intuitively estimate how the coordinate lines look like when a specific $\boldsymbol{\chi}$ is chosen. In the case of grid adaption, if the weight function is large, the heat conduction is low resulting in small temperature gradients and thus closely spaced grid cells. On the other hand, let us reconsider Eq. (25) for the case $\varepsilon = 0$. Then we can write

$$\boldsymbol{\chi} = \frac{1}{k}\hat{t}\hat{t} + k\hat{n}\hat{n} = \chi_{\parallel}\hat{t}\hat{t} + \chi_{\perp}\hat{n}\hat{n} \tag{29}$$

which for $k < 1$ simply means that the heat conduction parallel to the magnetic field is far stronger than perpendicular to it, which is in fact the case in an actual fusion reactor. The coordinate lines will thus tend to align with the magnetic flux surfaces with the degree of alignment given by k resulting in an almost aligned grid. In the limit of vanishing k the alignment should be perfect. If $|\nabla\psi|$ vanishes, the tensor (25) reduces to $\boldsymbol{\chi} = \boldsymbol{I}$.

1.6 The elliptic grids

Suppose that we have constructed a boundary aligned grid (ζ, η), which may but not necessarily has to be the orthogonal grid from Section 1.2. Now, we solve the general elliptic equation

$$\nabla \cdot (\boldsymbol{\chi} \nabla \bar{u}) = \partial_i (\sqrt{g} \chi^{ij} \partial_j \bar{u}) = 0 \tag{30}$$

in the transformed coordinate system with boundary conditions $\bar{u}|_{\partial\Omega} = \psi$. We set $u = c_0(\bar{u} - \psi_0)$. This means that we have to transform the conduction tensor χ from Cartesian to flux coordinates, which is done by the well known rules of tensor transformation

$$\chi^{\zeta\zeta}(\zeta,\eta) = (\zeta_x\zeta_x\chi^{xx} + 2\zeta_x\zeta_y\chi^{xy} + \zeta_y\zeta_y\chi^{yy})|_{x(\zeta,\eta),y(\zeta,\eta)} \tag{31a}$$
$$\chi^{\zeta\eta}(\zeta,\eta) = (\zeta_x\eta_x\chi^{xx} + (\zeta_x\eta_y + \eta_x\zeta_y)\chi^{xy} + \zeta_y\eta_y\chi^{yy})|_{x(\zeta,\eta),y(\zeta,\eta)} \tag{31b}$$
$$\chi^{\eta\eta}(\zeta,\eta) = (\eta_x\eta_x\chi^{xx} + 2\eta_x\eta_y\chi^{xy} + \eta_y\eta_y\chi^{yy}))|_{x(\zeta,\eta),y(\zeta,\eta)} \tag{31c}$$

The equivalent of the Cauchy–Riemann equations in this formulation reads

$$v_\zeta = -\sqrt{g}(\chi^{\eta\zeta}u_\zeta + \chi^{\eta\eta}u_\eta) \tag{32a}$$
$$v_\eta = +\sqrt{g}(\chi^{\zeta\zeta}u_\zeta + \chi^{\zeta\eta}u_\eta) \tag{32b}$$

These are constructed such that $\nabla \cdot ((\boldsymbol{\chi}/\det\chi)\nabla v) = 0$. The interested reader might notice that Eq. (32) just defines the components of the Hodge dual $\mathrm{d}v = \star\mathrm{d}u$ if χ is interpreted as a metric. We note that these equations are now valid for any grid that we use to solve Eq. (30). If we find a boundary aligned grid analytically, we can start the grid construction directly with the solution of Eq. (30) and then proceed with the conformal grid generation.

The relevant equations for the streamline integration now read

$$\mathrm{d}u = c_0(\bar{u}_\zeta \mathrm{d}\zeta + \bar{u}_\eta \mathrm{d}\eta) \tag{33a}$$
$$\mathrm{d}v = c_0\sqrt{g}(-(\chi^{\eta\zeta}\bar{u}_\zeta + \chi^{\eta\eta}\bar{u}_\eta)\mathrm{d}\zeta + (\chi^{\zeta\zeta}\bar{u}_\zeta + \chi^{\zeta\eta}\bar{u}_\eta)\mathrm{d}\eta) \tag{33b}$$

which just means that v is orthogonal to u in the scalar product generated by the symmetric tensor $\boldsymbol{\chi}$, which we denote by $\langle ., . \rangle_\chi$. We have $J = c_0^2\sqrt{g}\langle\nabla\bar{u}, \nabla\bar{u}\rangle_\chi := c_0^2\sqrt{g}(\bar{u}_\zeta^2\chi^{\zeta\zeta} + 2\bar{u}_\zeta\bar{u}_\eta\chi^{\zeta\eta} + \bar{u}_\eta^2\chi^{\eta\eta})$ and

$$\partial_u = \frac{1}{c_0\langle\nabla\bar{u}, \nabla\bar{u}\rangle_\chi}(\chi^{\zeta\zeta}\bar{u}_\zeta + \chi^{\zeta\eta}\bar{u}_\eta)\partial_\zeta + (\chi^{\eta\zeta}\bar{u}_\zeta + \chi^{\eta\eta}\bar{u}_\eta)\partial_\eta \tag{34a}$$
$$\partial_v = \frac{1}{c_0\sqrt{g}\langle\nabla\bar{u}, \nabla\bar{u}\rangle_\chi}(-\bar{u}_\eta\partial_\zeta + \bar{u}_\zeta\partial_\eta) \tag{34b}$$

As for the orthogonal coordinates we have to integrate these two vector fields to construct our coordinates. We begin with the integration of ∂_v along the $\zeta = 0$ line. It is important to note that $\bar{u}_\eta|_{\zeta=0} = 0$ and thus

$$\eta_v(0,\eta) = \left(c_0\sqrt{g}\bar{u}_\zeta\chi^{\zeta\zeta}\right)|_{\zeta=0,\eta} \tag{35}$$

We can use Eq. (35) to define c_0 such that $v \in [0, 2\pi]$. In order to do so we simply integrate

$$v_1 = \int_0^{2\pi} \left.\frac{\mathrm{d}v}{\mathrm{d}\eta}\right|_{\zeta=0} \mathrm{d}\eta = \int_0^{2\pi} \frac{1}{\eta_v}\mathrm{d}\eta = c_0\int_0^{2\pi} \sqrt{g}\chi^{\zeta\zeta}\bar{u}_\zeta(0,\eta)\mathrm{d}\eta := 2\pi \tag{36}$$

with $v_0 = 0$ and choose c_0 such that $v_1 = 2\pi$. This is the analogous equation to Eq. (16), with the difference that we can integrate Eq. (36) directly using numerical quadrature. Having done this we integrate, analogous to Section 1.2, ∂_v on $\zeta = 0$ to get starting points for the integration of ∂_u from $\bar{u}_0 = 0$ to $\bar{u}_1 = c_0\zeta_1$. Note, that we can compute the components of $\mathrm{d}u$ and $\mathrm{d}v$ in terms of $\mathrm{d}x$ and $\mathrm{d}y$ by using the transformation

$$u_x(\zeta,\eta) = u_\zeta\zeta_x + u_\eta\eta_x,\ u_y(\zeta,\eta) = u_\zeta\zeta_y + u_\eta\eta_y \tag{37a}$$
$$v_x(\zeta,\eta) = v_\zeta\zeta_x + v_\eta\eta_x,\ v_y(\zeta,\eta) = v_\zeta\zeta_y + v_\eta\eta_y \tag{37b}$$

as soon as the constant c_0 becomes available.

Note that the resulting grid will, in general, not be orthogonal in the Euclidean metric. However, for all the cases we discuss in this paper, the grid is orthogonal at the boundary.

The final algorithm now reads, assuming that u is discretized by not necessarily equidistant points u_i with $i = 0, 1, \dots N_u - 1$ and v by v_j with $j = 0, 1, \dots N_v - 1$ and that $x(\zeta,\eta)$, $y(\zeta,\eta)$ as well as the components of the Jacobian $\zeta_x(\zeta,\eta)$, $\zeta_y(\zeta,\eta)$, $\eta_x(\zeta,\eta)$ and $\eta_y(\zeta,\eta)$ are available from the first coordinate transformation (e.g. Section 1.2):

1. Choose either $\boldsymbol{\chi} = \boldsymbol{I}$, $\boldsymbol{\chi} = 1/w\boldsymbol{I}$ or $\boldsymbol{\chi} = \sqrt{G}\boldsymbol{G}$ depending on whether a conformal, adapted or monitor grid is desired.
2. Discretize and solve the elliptic equation (30) on the ζ,η grid for $\bar{u}(\zeta,\eta)$ with any method that converges. Use Eq. (31) to transform χ from Cartesian to the ζ,η coordinate system. The chosen resolution determines the accuracy of the subsequent streamline integration.
3. Numerically compute the derivatives $\bar{u}_\zeta$ and $\bar{u}_\eta$ and construct $\eta_v^{-1}(0,\eta)$ using Eq. (35) for $c_0 = 1$
4. Integrate Eq. (36) to determine c_0.
5. On the ζ,η grid compute η_v, ζ_u and ζ_v according to Eq. (34) as well as u_ζ, u_η, v_ζ and v_η according to Eq. (33). Use Eq. (37) and the Jacobian of the ζ,η coordinates to compute u_x, u_y, v_x and v_y.
6. Integrate the streamline of ∂_v for $\zeta = 0$ from $v = 0 \dots v_j$ for all j on the ζ,η grid using the normalized component η_v. Interpolate $\eta_v(\zeta,\eta)$ when necessary.
7. Using the resulting points as start values integrate ∂_u from $u = 0 \dots u_i$ for all i and all points. The result is the list of coordinates $\zeta(u_i, v_j)$, $\eta(u_i, v_j)$ for all i and j.

8. Interpolate $x(\zeta,\eta)$, $y(\zeta,\eta)$, $u_x(\zeta,\eta)$, $u_y(\zeta,\eta)$, $v_x(\zeta,\eta)$ and $v_y(\zeta,\eta)$ on this list.

There are two differences in this algorithm from the one presented in Section 1.2. For the integration of the vector fields Eq. (34) and the evaluation of derivatives Eq. (33) we need to evaluate its components at arbitrary points. An interpolation method is thus needed. In order to avoid out-of-bound errors we can artificially make the ζ,η box periodic. Second, the existence of the coordinate v is guaranteed by the Cauchy–Riemann equations and thus the h function does not appear.

A suitable test for the implementation is e.g. the volume/area of the domain. Being an invariant the volume must be the same regardless of the coordinate system in use.

Finally, as already mentioned several times the ζ,η coordinates in the algorithm can be replaced by any structured, boundary aligned grid. In especially this means that for example a Cartesian grid ($\psi(x,y)=y$, $\zeta=x$, $\eta=y$) can be used if the boundary is a rectangle. The periodic boundary condition in η in Eq. (30) can easily be replaced by Neumann boundaries if necessary. The proposed adaption function w or the monitor metric $\boldsymbol{G}$ are in the same way only suggestions, are in principle independent of $\psi(x,y)$ and can be replaced by any suitable quantity.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) W1227-N16 and Y398. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] W. D'haeseleer, W. Hitchon, J. Callen, and J. Shohet. *Flux Coordinates and Magnetic Field Structure*. Springer Series in Computational Physics. Springer-Verlag, 1991.

[2] T. Frankel. *The geometry of physics: an introduction*. Cambridge University Press, second edition, 2004.

[3] A. H. Glasser, V. D. Liseikin, I. A. Vaseva, and Y. V. Likhanova. Some computational aspects on generating numerical grids. *Russ. J. Numer. Anal. Math. Modelling*, 21(6):481–505, 2006. doi:10.1515/rnam.2006.21.6.481.

[4] V. D. Liseikin. *A Computational Differential Geometry Approach to Grid Generation*. Springer-Verlag, second edition, 2007.

[5] I. A. Vaseva, V. D. Liseikin, Y. V. Likhanova, and Y. N. Morokov. An elliptic method for construction of adaptive spatial grids. *Russ. J. Numer. Anal. Math. Modelling*, 24(1):65–78, 2009. doi:10.1515/RJNAMM.2009.006.

[6] M. Wiesenberger, M. Held, and L. Einkemmer. Streamline integration as a method for two-dimensional elliptic grid generation. *Journal of Computational Physics*, 340:435–450, 2017. doi:10.1016/j.jcp.2017.03.056.