NATURALLY CURVED QUADRILATERAL MESH GENERATION USING AN ADAPTIVE SPECTRAL ELEMENT SOLVER

Julian Marcon[1] David A. Kopriva[2,3] Spencer J. Sherwin[4] Joaquim Peiró[5]

[1] *Imperial College London, London, U.K. julian.marcon14@imperial.ac.uk*
[2] *The Florida State University, Tallahassee, FL, U.S.A. kopriva@math.fsu.edu*
[3] *San Diego State University, San Diego, CA, U.S.A.*
[4] *Imperial College London, London, U.K. s.sherwin@imperial.ac.uk*
[5] *Imperial College London, London, U.K. j.peiro@imperial.ac.uk*

ABSTRACT

We describe an adaptive version of a method for generating valid naturally curved quadrilateral meshes. The method uses a guiding field, derived from the concept of a cross field, to create block decompositions of multiply connected two dimensional domains. The *a priori* curved quadrilateral blocks can be further split into a finer high-order mesh as needed. The guiding field is computed by a Laplace equation solver using a continuous Galerkin or discontinuous Galerkin spectral element formulation. This operation is aided by using p-adaptation to achieve faster convergence of the solution with respect to the computational cost. From the guiding field, irregular nodes and separatrices can be accurately located. A first version of the code is implemented in the open source spectral element framework *Nektar++* and its dedicated high order mesh generation platform *NekMesh*.

Keywords: cross field, quadrilateral meshing, high order, spectral element method, adaptation

1. INTRODUCTION

Finite element [1], finite volume, (block structured) finite difference [2] and spectral element methods [3, 4] all benefit from using quadrilateral meshes. Yet fully automatic quadrilateral mesh generation is still difficult [5], especially for high order (curved) elements used in spectral element methods.

A method was described in [6] that generates large block structures with naturally curved boundaries that can be used as is, say for spectral element or block structured finite difference methods, or further subdivided into small quadrilaterals for finite element or finite volume methods. The idea behind the approach came from the computer graphics community, which used cross fields [1, 5]. The procedure in [6] used a high order approach to compute a guiding field, accurately locating the irregular nodes, and accurately integrating the separatrices used to subdivide the domain. It used a high order continuous or discontinuous Galerkin spectral element method (SEM or DGSEM) on a base triangular mesh, the approximation depending on the regularity of the boundary conditions needed to solve Laplace problems that determine the guiding field. Unlike previous work, accuracy was preserved because all operations were performed on the guiding field rather than a cross field. That new procedure was implemented in the open source spectral/hp element framework *Nektar++* and its high order mesh generation utility *NekMesh*.

One of the significant costs in the high resolution field guided approach is the solution to the Laplace problems on the triangular mesh. As the polynomial order of the approximation gets high, the condition number of the matrices that must be solved (either directly or

iteratively) becomes large, leading to increased cost to get the field solution. The approximation order was chosen *ad hoc*, and there was no way to tell whether or not the chosen approximation order was sufficient.

One can avoid the *ad hoc* nature of the choice of approximation order, and the slow convergence rates associated with choosing one that is higher than is necessary, by using an adaptive solver that adjusts the approximation to satisfy a desired error tolerance. With a spectral element solver, adaptation can be done either by subdividing the underlying triangular elements (h-adaptation) and/or by increasing the polynomial order (p-adaptation). In this work, we employ p-adaptation because of its best accuracy, compared to h-adaptation, with respect to the computational cost for smooth fields without discontinuities [7]. We use a dynamic p-adaptive strategy previously devised for compressible flow simulation [8, 9] and implemented in *Nektar++*.

The methodology consists of three main steps. First, a guiding field is computed. This requires the generation of a high order triangular mesh of the domain and the solution to the Laplace problem using a p-adaptive SEM or DGSEM solver. Then, analysis of the guiding field must be carried out to construct a separatrix graph. This construction process includes the location of interior critical points, the calculation of their valence as well as the valence of corners, and then the integration of streamlines and their possible manipulation. Finally, a high order quadrilateral mesh can be generated. To achieve this, streamlines are splined and used to split the CAD model into quadrilateral blocks, which can then be trivially meshed and refined.

This paper is structured as follows. Section 2 gives an outline of the mathematical formulation of the problem, including a definition of crosses, the guiding field and the partial differential equations to solve, a summary of the SEM/DGSEM formulation, and a overview of the p-adaptation strategy. Section 3 covers some of the implementation details and Section 4 illustrates the methodology with examples.

2. FORMULATION

The goal is to subdivide any simply or multiply connected two dimensional domain, Ω, with piecewise smooth boundaries into quadrilateral subdomains. The decomposition is called regular if all the corners of the quadrilaterals have valence four, that is, each corner node connects four edges. Especially in multiply connected domains, there will be *irregular* nodes, where the valence will be larger or smaller than four. The problem is how to generate a quadrilateral decomposition that has a minimum number of irregular nodes.

2.1 The Guiding Field

An approach that has been used in the computer graphics community to generate quadrilateral meshes over surfaces relies on the idea of a *cross field*. For full discussion of cross fields and methods based on them, see, for example, references [1, 10, 11, 12]. We will present the basic idea.

A cross has four way rotational symmetry and can be represented by the four vectors

$$C(\psi) = \left\{\vec{c}_k = (c_x, c_y)_k = (\cos(\psi_k), \sin(\psi_k))\right\}, \quad (1)$$

where

$$\psi_k = \psi + k\frac{\pi}{2} \quad (2)$$

for, $k = 0, 1, 2, 3$. In turn, the tangent angle, or phase, ψ, can be computed from the four quadrant inverse tangent,

$$\psi = \frac{1}{4}\mathrm{atan2}\,(\mathrm{v}, \mathrm{u}) \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]. \quad (3)$$

In other words, a cross is represented by four unit vectors $\vec{c}_k$ at any point $\vec{x} \in \Omega$ in a vector field $\vec{v} = (u(\vec{x}), v(\vec{x}))$ at which $\vec{v} \neq 0$. A cross rotated by an angle $\pi/2$ is still the same cross.

The function ψ is either tangent or orthogonal (due to the jump in the $atan2$ function) to the streamlines of the field $\vec{v}$. It is parallel to one of the branches of a cross and is undefined at a critical point of the field, i.e. where $\vec{v} = (0, 0)$. Given the way ψ is computed from the arctangent, there will be a jump in value $\pi/2$ in ψ depending on the signs of u and v even if $\vec{v}$ is smooth. We will call lines in the field across which ψ jumps *jump lines*.

The method rests on finding an appropriate guiding field $\vec{v}$. The only known values are on the boundary $\partial\Omega$ of Ω. There, the field is aligned with the boundary to ensure that the mesh to be generated is aligned there. Along the boundary,

$$\vec{v}_b = (u_b, v_b) = (\cos(4\theta_b), \sin(4\theta_b)), \quad (4)$$

where θ_b is the tangent angle of the boundary. The vector $\vec{v}_b$ at any point along the boundary defines a cross (1) at that point. The factor of four in eq. (4) ensures the same u, v values for each $90°$ rotation of the angle θ_b, and hence the rotational symmetry.

Boundary values defined with (4) are smoothly propagated to the interior. Solving a Laplace problem for u and v

$$\begin{cases} \nabla^2\vec{v} = 0, & \vec{x} \in \Omega \\ \vec{v} = \vec{v}_b, & \vec{x} \in \partial\Omega \end{cases} \quad (5)$$

guarantees that the field $\vec{v}$ is smooth in the interior of the domain.

2.1.1 Critical Points

To obtain a valid block decomposition of an arbitrary simply or multiply connected domain, *irregular nodes* must be found where the valence $\mathcal{V} \neq 4$. In the cross field literature, these correspond to *singular points* of the cross field where crosses are undetermined. A similar phenomenon is observed in this guiding field method. We return to Eq. (3) where ψ is undefined when $u = v = 0$. Points where $\vec{v} = \vec{0}$ are called *critical points* of the guiding field and are a direct analogy of singular points of cross fields.

Critical points are tied to null isocontours in the domain. Zeroes in u and v emerge from boundary conditions and propagate into the interior of the domain due to the Laplace equation. This is shown in Fig. 1 where null isocontours are shown intersecting at critical points. Although ψ is undefined at critical points, $\vec{v}$ remains smooth across the domain

We further need to calculate the valence of these irregular nodes, which will correspond to the number of quads connected to them. We categorize them analytically, as recognizable topologies in the $\vec{v}$ field [13], by the Poincaré index

$$i_\gamma = \frac{1}{2\pi} \oint_\gamma d\phi, \tag{6}$$

where γ is a simple arbitrary closed curve around a critical point and

$$\phi = \mathrm{atan2}\,(\mathrm{v}, \mathrm{u}) = 4\psi. \tag{7}$$

A regular node in the domain would yield an index of zero while irregular nodes have an index of ± 1. The index can, in turn, be used to calculate the number of separatrices connected to this irregular node, and therefore its valence. The index $i_\gamma = -1$ leads to $\mathcal{V} = 5$ and $i_\gamma = +1$ to $\mathcal{V} = 3$ [1, Lemma 5.1], [14].

We adapt the Poincaré index (6), using relation (7) to the guiding field to obtain

$$I_c = \frac{1}{\pi/2} \oint_c \frac{d\psi}{d\theta} d\theta, \tag{8}$$

where c is a (small) counter-clockwise circular contour centered on the critical point. From this, the *valence* $\mathcal{V}$ of any point in the domain is defined as

$$\mathcal{V} = 4 - I_c. \tag{9}$$

The contour radius c must be chosen so that the closed curve encloses only one critical point. If c encloses multiple critical points, an incorrect combined valence will be computed. For example, two enclosed $\mathcal{V} = 5$ critical points will produce a combined valence of 6, while two opposite (one $\mathcal{V} = 5$ and one $\mathcal{V} = 3$) enclosed critical points will cancel out and produce a combined regular node ($\mathcal{V} = 4$). In practice, critical points are far enough from each other that such problems do not arise, and one can always compute the radius around any critical point from the distance to the nearest of the remaining critical points.

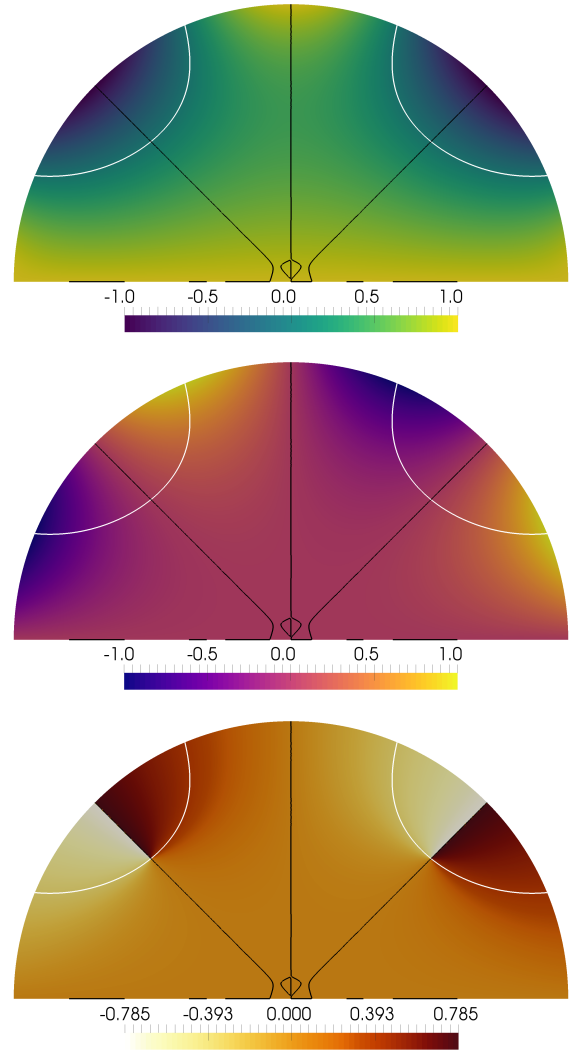

Figure 1: Solution u (top) and v (middle) fields and computed ψ field (bottom) on the half disc geometry. Isocontours of $u = 0$ and $v = 0$ shown in white and black respectively.

Eq. (9) can be generalized to an open line circle integral to handle boundary points too [12], leading to

$$I\,(\theta_0, \theta_f) = \frac{1}{\pi/2} \int_{\theta_0}^{\theta_f} \frac{d\psi}{d\theta} d\theta, \tag{10}$$

such that $I_c = I\,(0, 2\pi)$. Similarly, the valence of any

boundary point is defined as

$$\mathcal{V} = \frac{\Delta\theta}{\pi/2} - I\left(\theta_0, \theta_f\right), \tag{11}$$

where $\Delta\theta = \theta_f - \theta_0$. $I\left(\theta_0, \theta_f\right)$ can be viewed as a correcting factor for the valence towards an integer value.

2.1.2 Streamline Integration

Separatrices of the block decomposition, in the form of *streamlines* of the guiding field, must be traced from irregular nodes and boundary corners throughout the domain. The guiding field, akin to cross fields, can be used for this purpose too. We define a streamline as $\vec{x}(t)$ such as

$$\frac{d\vec{x}}{dt} = \tilde{v}\left(\psi'(\vec{x})\right) \tag{12}$$

where t is the integration parameter and $\tilde{v}\left(\psi'(\vec{x})\right)$ is the adjusted guiding field vector.

Because ψ is discontinuous, defined on $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ by (3), whereas crosses are invariant of rotations of $\pi/2$, we use an *adjusted* guiding direction ψ'. The guiding direction must indeed be adjusted by a multiple of $\pi/2$ every time a streamline crosses a jump line, defined in Sec. 2.1. This allows us to retain the consistent directionality of a smooth cross field.

Eq. (12) can be integrated with respect to t given appropriate initial conditions

$$\vec{x}(t_0) = \vec{x}_0, \tag{13}$$

where t_0 is the initial integration time and $\vec{x}_0$ the location of the start of the streamline. The start of the streamline may either be an irregular node or a boundary corner. The ordinary differential equation (12) can be integrated using any traditional numerical scheme. Sec. 3.2.3 will more specifically cover the integration strategy.

2.2 The Spectral Element Method

We compute the BVPs (5) with either a continuous Galerkin (CG) or a discontinuous Galerkin (DG) spectral element method on a triangular mesh [3]. Spectral element methods are spectrally accurate, meaning that the convergence rate depends only on the smoothness of the solution. They are high resolution in that they use a large number of degrees of freedom within an element. Unlike traditional low order finite element methods, the high order polynomial expansion of the solution inside each element allows us to locate critical points, identify their valence, and finally trace streamlines with high order accuracy.

In this work, we use the spectral/*hp* element methods formulation described in detail in reference [3] and implemented in *Nektar++* [15]. We briefly survey the fundamentals of this method here for the numerical solution of PDEs, $\mathcal{L}u = 0$, over a domain Ω. The domain Ω is covered by a mesh of triangular finite elements, Ω_e so that $\Omega = \cup\Omega_e$ and $\Omega_{e_1} \cap \Omega_{e_2} = \partial\Omega_{e_1 e_2}$ is either an empty set or the interface between two elements, and is of one dimension less than the mesh. We solve the PDE problem weakly and require that $u|_{\Omega_e}$ is in the Sobolev space $H^k(\Omega_e)$ $k = 0, 1$. In the CG formulation, we require the solution to be in H^1, H^0 for DG.

We formulate the solution to (5) in weak form: *find* $\vec{v} \in H^k(\Omega)$ *such that*

$$a(\vec{v}, \vec{w}) = l(\vec{w}) \quad \forall \vec{w} \in H^k(\Omega), \tag{14}$$

where $a(\cdot,\cdot)$ is a symmetric bilinear form, $l(\cdot)$ is a linear form and $H^k(\Omega)$ is the Sobolev space formally defined as

$$H^k(\Omega) := \{\vec{w} \in L^2(\Omega) \mid D^\alpha \vec{w} \in L^2(\Omega) \ \forall \ |\alpha| \leq k\}. \tag{15}$$

We solve problem (14) numerically and therefore consider solutions in a finite dimensional subspace $V_N \subset H^k(\Omega)$. In the finite dimensional subspace, the problem becomes: *find* $\vec{v}^\delta \in V_N$ *such that*

$$a(\vec{v}^\delta, \vec{w}^\delta) = l(\vec{w}^\delta) \quad \forall \vec{w}^\delta \in V_N, \tag{16}$$

augmented with boundary conditions, (4). In the CG formulation, we also enforce the condition $V_N \subset C^0$.

To represent $\vec{v}^\delta(\vec{x}) = \sum_n \hat{u}_n \Phi_n(\vec{x})$, we take a weighted sum of N trial functions $\Phi_n(\vec{x})$ defined on Ω. This transforms the problem of finding the subspace solution to one of finding the coefficients $\hat{u}_n$ that define $\vec{v}^\delta(\vec{x})$ within an element. To obtain a unique choice of coefficients $\hat{u}_n$, we require that the residual $R = \mathcal{L}\vec{v}^\delta$ be orthogonal to the polynomial space, i.e. that its L^2 inner product, with respect to the test functions $\Psi_n(\vec{x})$, is zero. In the Galerkin approximation, the projection the test functions are chosen to be the same as the trial functions, i.e. $\Psi_n = \Phi_n$.

The contributions of each element in the domain must be combined to construct the global basis Φ_n. A parametric mapping $\mathcal{X}_e : \mathcal{E} \to \Omega_e$ exists to each element Ω_e from a standard reference element $\mathcal{E} \subset [-1, 1]^d$. This mapping is given by $\vec{x} = \mathcal{X}_e\left(\vec{\xi}\right)$, where $\vec{x}$ is the vector of the physical coordinates and $\vec{\xi}$ defines the coordinates in the reference space. In *Nektar++*, triangular, tetrahedral, prismatic, and pyramidal elements are created by collapsing one or more of the coordinate directions to create singular vertices. This allows it to support, for this work, more easily generated triangular meshes.

We use a local polynomial basis on the reference element to represent the solution. A one-dimensional order-P basis is a set of polynomials $\Phi_p(\xi), 0 \leq p \leq P$, defined on the reference segment $-1 \leq \xi_1 \leq 1$. A tensor basis is used in two dimensional reference regions, where the polynomial space is constructed as the tensor product of one dimensional bases on segments and quadrilaterals.

Finally, the discrete solution in a physical element Ω_e can be expressed as

$$\vec{v}^{\delta}(\vec{x}) = \sum_n \hat{v}_n \phi_n(\mathcal{X}_e^{-1}(\vec{x})), \tag{17}$$

with $\hat{v}_n$ being the coefficients computed by the Galerkin procedure. We therefore restrict the solution space to

$$V := \{\vec{v} \in H^k(\Omega) \mid \vec{v}|_{\Omega_e} \in \mathcal{P}_P(\Omega_e)\}, \tag{18}$$

where $\mathcal{P}_P(\Omega_e)$ is the space of order P polynomials on Ω_e.

An assembly operator assembles the element contributions to the global solution. In the CG formulation, elemental contributions of neighbours are summed to enforce C^0-continuity. In the DG formulation, flux values are transferred from the element interfaces into the global solution vector. *Nektar++* supports both discretizations, allowing us to trivially switch from one to the other at any time. The DG formulation is necessary for the method to work with arbitrary corner angles in the geometry, as will be explained in Sec. 3.1.

2.2.1 p-Adaptation

Spectral element methods also offer the freedom to locally increase or decrease the resolution of the solution according to an estimated local error. Adaptation is achieved by adding or removing modes — i.e. increasing or decreasing the polynomial order of individual elements. The p-adaptive strategy employed in this paper follows the procedure laid out in references [8] and [9]. The idea is simple: where local error is high, we increase local resolution; where local error is low, we decrease resolution. The procedure is described in Algorithm 1, where e is an individual element, S_e and P_e are its associated error indicator and polynomial order, ϵ_u and ϵ_l are the upper and lower error thresholds, and P_{max} and P_{min} are the maximum and minimum polynomial orders allowed.

We use an estimate of the discretization error from the formulation described in reference [16] for the error indicator (sometimes called *sensor*). We only require to compute the solution u_P once at order P, which we then project onto polynomial space $P-1$ to obtain

```
while any P_e is modified do
    Calculate solution to (5) ;
    foreach e do
        Calculate S_e ;
        if S_e > ε_u ∧ P_e < P_max then
            increment P_e ;
        else if S_e < ε_l ∧ P_e > P_min then
            decrement P_e ;
        else
            maintain P_e ;
        end
    end
end
```

Algorithm 1: The p-adaptive procedure.

u_{P-1}. From these solutions at different polynomial orders, we calculate our sensor:

$$S_e = \frac{\|u_P - u_{P-1}\|_{2,e}^2}{\|u_P\|_{2,e}^2}, \tag{19}$$

where $\|\cdot\|_2$ is the L_2-norm. For a modal expansion, this sensor is cheap to compute. It phyically represents the ratio between the energy of the highest mode and the rest of modal energy spectrum. The sensor S_e can be computed for either component of the solution $\vec{v}$ in (5).

3. IMPLEMENTATION OF THE PROCEDURE

The process is split into three main steps, each associated with different libraries and utilities of the *Nektar++* platform [15, 17]:

1. Computation of a guiding field by solving the Laplace problem (5). Carried out in the *solver*.

2. Analysis of the guiding field, including locating critical points, calculating their valence and the valence of corners, and the integration of streamlines. Carried out in the post-processing utility *FieldConvert*.

3. Generation of a quadrilateral mesh by splitting the CAD model into blocks and splitting these further into a fine mesh. Carried out in the meshing utility *NekMesh*.

This section provides an overview of the process, illustrated on the geometry of a half disc geometry (see Fig. 2), commonly used in the cross field literature [14, 10]. The reader should refer to reference [6] for implementation details.

Figure 2: Geometry of a half disc for illustration of the quadrilateral meshing method.

3.1 Computation of a Guiding Field

A standard Laplace solver for the boundary value problem (5) is used in the open source spectral/*hp* element framework *Nektar++* [15, 17] with boundary conditions (4). The geometry is passed onto the guiding field solver in the form of a high order triangular mesh previously generated in *NekMesh* [17] following the procedure laid out in reference [18]. For the reference geometry, such a mesh is illustrated in Fig. 3.

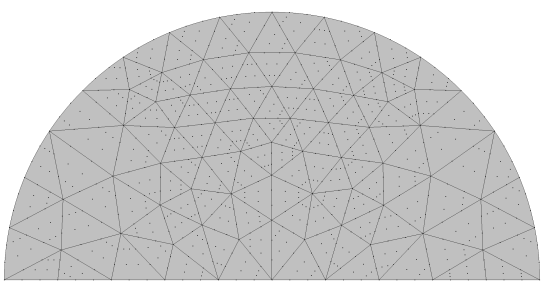

Figure 3: Third order triangular spectral element mesh for the half disc geometry.

For performance purposes, the triangular mesh should be coarse for fast computation of the guiding field. Resolution will be added in areas of interest by use of p-adaptation, as described in Sec. 2.2.1. A coarse mesh also proved to be lighter to process in the analysis of the guiding field, as will seen in Sec. 3.2.

Notice that the boundary conditions (BCs) (4) depend on the geometry and may therefore be discontinuous where the geometry is not smooth. To be exact, BCs are continuous if the boundaries satisfy one of two conditions:

- The curve at each point is C^1-continuous, i.e. a smooth curve; or
- The boundary curve is only C^0-continuous, i.e. a corner, and the angle is a multiple of $\pi/2$.

Otherwise, they are discontinuous.

A discontinuous Galerkin formulation, available in *Nektar++*, offers a discretization consistent way of handling discontinuous BCs without the need for *ad hoc* smoothing of a corner's BCs as used in traditional low order techniques [1].

As an example of a geometry that requires a DG solve, we present the polygon geometry shown in Fig. 4, which shows the solution and ψ fields. We note that the discontinuous BCs are naturally imposed with the Dirichlet BCs enforced through fluxes when needed. Considerations beyond the scope of this paper are made to address the discontinuous nature of the field solution, but they however do not affect the rest of the procedure.

Following the calculation of a solution, p-adaptation can be applied iteratively to locally increase or decrease the local polynomial order inside each element. Using the approach described in Sec. 2.2.1, a finer solution is obtained without the need to uniformly increase the resolution and therefore the cost of the calculation. Fig. 5 shows the final distribution of local number of modes when adapting based on either u (top) or v (bottom). Higher polonomial orders are found near the boundaries where the gradient of the solution is stronger.

3.2 Analysis of the Guiding Field

The mesh generation procedure continues with an analysis of the guiding field, in a way where crosses are never generated. The analysis is carried out in *FieldConvert*, *Nektar++*'s post-processing utility [17], and consists of three main parts:

1. Location of interior critical points.
2. Calculation of the valence of interior critical points and corners.
3. Integration of streamlines.

The goal of this analysis is to construct a separatrix graph to pass on to *NekMesh* for the generation of a coarse high order quadrilateral mesh.

3.2.1 Location of Interior Critical Points

As described in Sec. 2.1.1, interior critical points are located at $\vec{v} = \vec{0}$. To locate all of them we take a simple narrowing down approach. We first identify elements

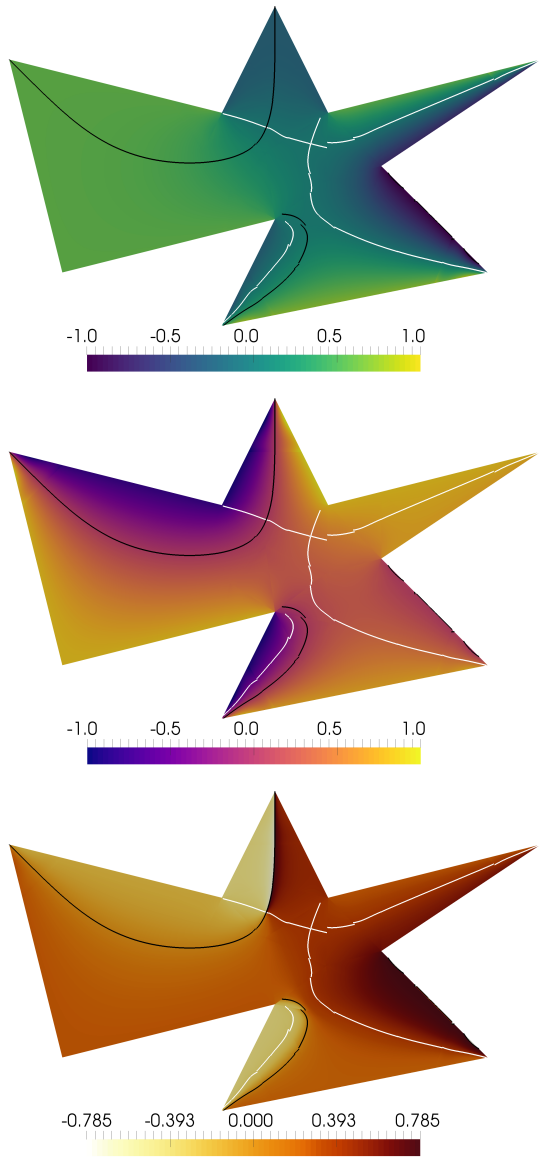

Figure 4: Solution u (top) and v (middle) fields and computed ψ field (bottom) on a polygon with acute and obtuse corners using a DG discretization. Isocontours of $u = 0$ and $v = 0$ are shown in white and black respectively.

that are likely to contain a $\vec{v} = \vec{0}$ point. These elements should have at least one quadrature point with each of the values $u \leq 0$, $u \geq 0$, $v \leq 0$ and $v \geq 0$. These elements either contain a critical point or are close to an element that does. For each of these elements, we perform a Newton search for a point satisfying $\vec{v} = \vec{0}$. A critical point may be found outside of the element boundaries, in which case we ignore it. Fig. 1 shows the location of the two interior critical points of

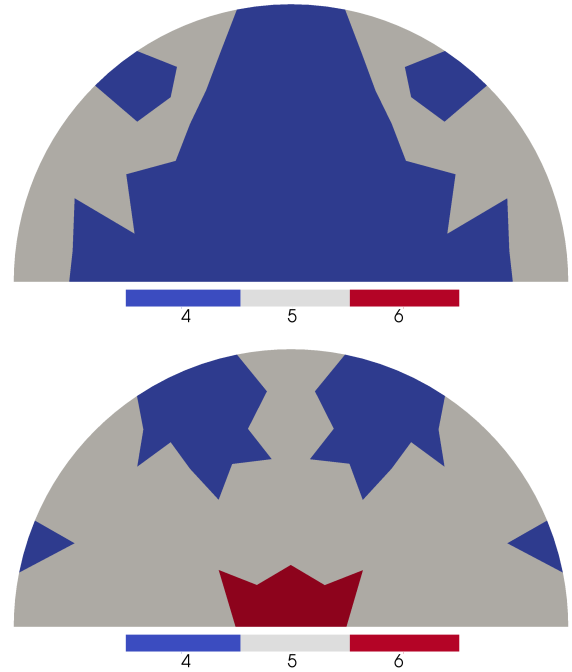

Figure 5: Distribution of local number of modes ($= P + 1$) for half disc geometry. Adaptation based on u (top) and v (bottom).

the reference geometry as the crossing of the u and v zero contours, consistent with the literature [14].

We accelerate the search inside each element by performing Netwon's method in reference space. Along with the initial identification of likely elements, all operations are performed in reference space at negligible computational cost.

3.2.2 Calculation of Valences

After the interior critical points are located, valences of these as well as of corners can be computed using (9) and (11) respectively.

Here again, crosses are not generated and valences can be computed from the guiding field alone. We note that the closed curve circle integral I_c in (8) evaluates to zero when ψ is continuous. This is the case of regular nodes in the domain. Where a critical point appears, lines of ψ jumps appear and I_c can be evaluated based on the number and direction of these. In a counter-clockwise manner, a positive jump (i.e. from $-\frac{\pi}{4}$ to $\frac{\pi}{4}$) leads to a negative index $I_c = -1$, and vice versa. The sign of the jump can itself be determined solely by values of $\vec{v}$, without the need to compute ψ or the cross field.

Using this knowledge, we can calculate the valence of

interior critical points from (9) by marching on a circle of small radius c around each critical point. The same reasoning is applied to the calculation of corner valences with (11) where θ_0 and θ_f must also be taken into account and are found trivially. From this logic, we find that the two critical points of the reference geometry are characterized by a single negative jump line each, and therefore have a valence $\mathcal{V} = 3$. Likewise, the two corners have a trivially evaluated valence $\mathcal{V} = 1$ due to $\Delta\theta = \pi/2$ and the absence of any jump line in ψ. See Fig.1 (bottom) for reference.

Although these evaluations are performed in physical space at a negligible computational cost, they can further be accelerated by performing them in reference space.

3.2.3 Integration of Streamlines

Streamlines emanate from the interior irregular nodes and corners. Because $\vec{v} = \vec{0}$ at irregular nodes, ψ is undefined. Likewise, ψ at corners may be ambiguous if BCs are discontinuous. It is therefore necessary to identify the initial direction of each streamline by looking at the value of ψ nearby the irregular nodes, at a small distance c from them. The initial direction can be found iteratively by progressively refining an initial guess, akin to a fixed point search. For the first streamline of an interior irregular node, any initial guess may be taken. For subsequent streamlines as well as for streamlines coming from corners, an initial guess can be take at angles multiples of $\frac{\Delta\theta}{\mathcal{V}}$ where $\Delta\theta = 2\pi$ for an interior irregular node.

Subsequently, all streamlines are *synchronously* advanced throughout the domain by integrating (12). At any point, two streamlines may meet head on. For that purpose, we check the proximity of all streamline front points as well as their direction. If two front points are within merging distance and if they are pointing in opposite directions, their respective streamlines are merged. This merging distance is typically taken as the integration step size but it can also be artificially increased to perform *aggressive* merging, therefore reducing the total number of streamlines and simplifying the separatrix graph. Fig. 6 illustrates the difference between normal (top) and aggressive (bottom) merging. When two streamlines are flagged for merging, they are advanced up to each other's starting point and merged pointwise using trigonometric weight functions to retain an optimal streamline direction. These trigonometric weight functions deflect streamlines as is illustrated in Fig. 6 (bottom). Reference [6] contains more technical details regarding the procedure.

A 4th order multi step Adams-Bashforth integrator is used in physical space. This integrator requires a search for the element that contains every new point,

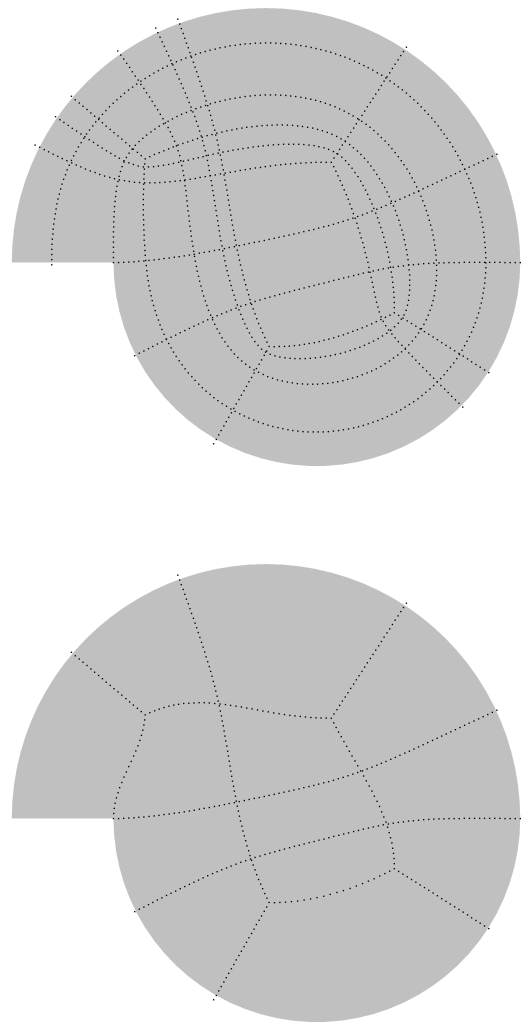

Figure 6: Example of normal (top) and aggressive (bottom) merging. A distance threshold of 5 times the step size was used for aggressive merging.

an operation that is expensive on large meshes. An inverse mapping is used to transform to parametric coordinates before the high order interpolation fo $\vec{v}$ is computed via (17). This process can be accelerated by integrating streamlines in reference space in a similar fashion to the work of Coppola *et al.* [19]. A standard Runge-Kutta scheme can then used while retaining the speed of reference space based integration.

3.2.4 Midpoint Division

A note should be made about corners with very shallow angles. Eq. (11) can lead to a valence of zero

when $\Delta\theta$ and $\Delta\psi$ cancel each other. This represents the situation of all streamlines converging towards a corner. We call this corner *degenerate* as it will produce a triangular block, i.e. a collapsed quadrilateral block. This topology is not valid for the purpose of quadrilateral-only mesh generation, so we resort to *ad hoc* manipulations. The triangular block can easily be split into three quadrilateral blocks using a midpoint division approach [20]. Practically, a 3-valence irregular node is introduced at the center of the triangular block *a posteriori*. Two of the three streamlines can be extended towards the domain boundaries in a straight line while the third streamline is integrated away from the degenerate corner towards the inside of the domain. An example of an *ad hoc* midpoint division manipulation is shown in Fig. 7.

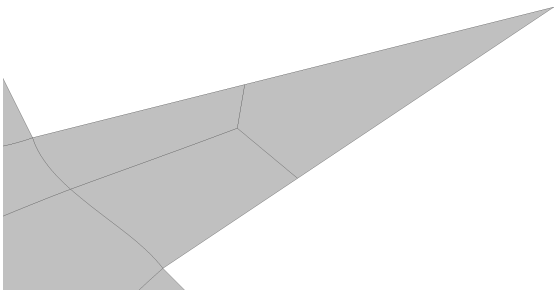

Figure 7: Detail of a domain showing a midpoint division of a degenerate quadrilateral caused by a sharp angled corner. The full domain is shown in Fig. 11.

3.3 Generation of a Quadrilateral Mesh

Once a separatrix graph has been generated, it is passed to *NekMesh*, *Nektar++*'s high order mesh generation utility [17]. *NekMesh* uses OpenCASCADE [21] as its CAD system for high order mesh generation. The OpenCASCADE kernel offers not only CAD query capabilities but also CAD modification tools. The discrete separatrices computed by *FieldConvert* are converted into interpolating splines. These splines can then be used to split the original CAD domains into the desired quadrilateral blocks. This results in a CAD model of topologically connected quadrilateral faces. A coarse high order quadrilateral mesh can then be trivially generated by *NekMesh*.

Such meshes are typically too coarse for spectral element analysis; it is often necessary to split them further. We take advantage of an isoparametric splitting approach described in [22] to refine strings of blocks, extended in references [23, 24] for bidirectional splitting. The method leverages the mapping between physical and reference space coordinates by splitting elements in reference space to obtain valid high order elements in physical space. In this procedure, it also preserves optimal alignment of blocks. Fig. 8 shows the coarse and fine meshes obtained on the reference geometry.

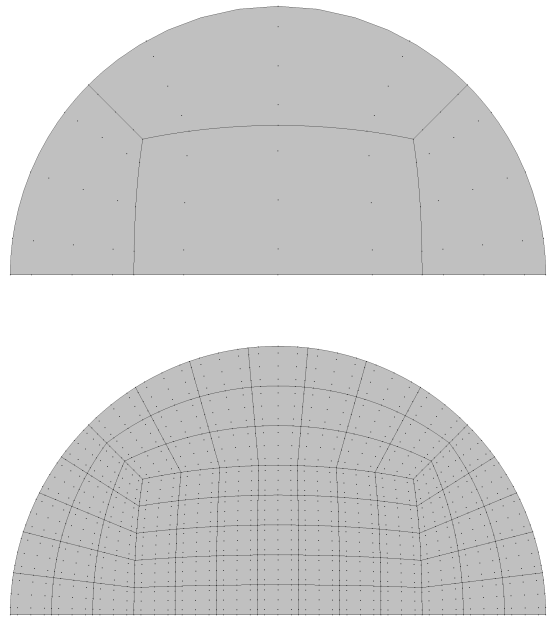

Figure 8: Coarse (top) and subdivided (bottom) quadrilateral meshes obtained on the half disc geometry based on detected critical points and traced and merged streamlines.

4. ADVANTAGES OF THE METHOD

We first illustrate advantages of the method with examples before analysing the use of p-adaptation in the procedure. These example geometries, e.g. the reference geometry of Fig. 2, are typically used as examples in the cross field literature [1, 14, 10]. The advantages that we want to illustrate in this section include minimal number of irregular nodes and possibility to aggressively merge streamlines (Geometry I), validity of the block decompositions and absence of limit cycles (Geometry II), a discretization consistent way of handling corners (Geometry III), the preservation of symmetries and patterns (Geometry IV), and finally the flexibility in generating finer meshes (Geometry V).

Geometry I (Fig. 9) is a multiply connected domain previously meshed using cross fields in [14]. While some methods generate spurious irregular nodes in the upper central part of the domain, the approach described here generates only two 3-valence nodes points to obtain an optimal block decomposition of the do-

main. Aggressive merging can also be applied to connect the two irregular nodes. The reader will observe that optimal angles are preserved at the irregular nodes thanks to the trigonometric weights for merging.

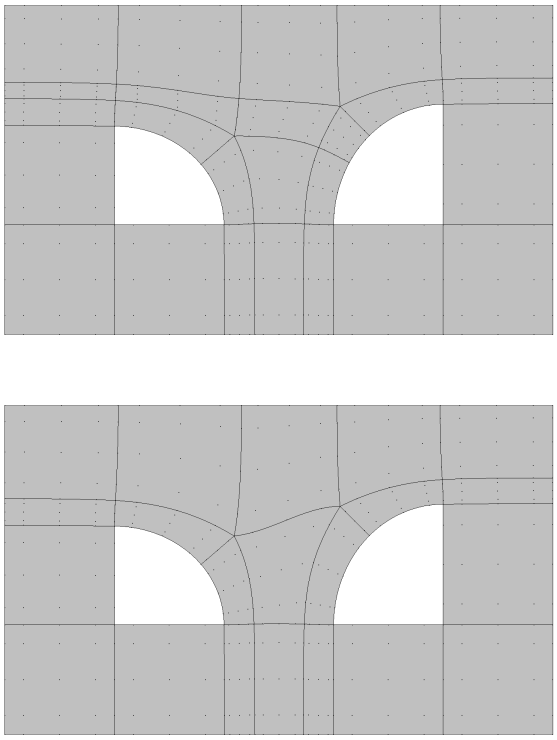

Figure 9: Coarse quadrilateral mesh obtained on Geometry I with normal (top) and aggressive (bottom) merging.

Geometry II (Fig. 10), sometimes called the *nautilus*, is typically prone to a limit cycle of the streamlines in low order methods leading to invalid block decompositions without *ad hoc* manipulations [1, 10]. The formulation of the problem generates a guiding field in which no limit cycle appears. In fact, the irregular node placement is similar to that of a simple disc, with four 3-valence irregular nodes the optimal number for a valid block decomposition. As noted in Sec. 3.2.3, this improved separatrix graph was subject to aggressive streamline merging, for a visually more pleasing coarse mesh.

Geometry III (Fig. 11) illustrates the use of a DG discretization to handle angles that are not multiples of $\pi/2$ without the need for *ad hoc* smoothing of corners. The rest of the procedure remains however unchanged for easier code implementation and maintenance. This example also features a sharp corner with a valence of zero whose connected degenerate block is split using the midpoint division method.

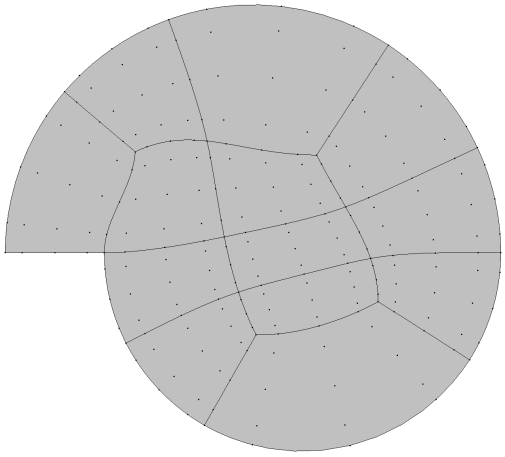

Figure 10: Coarse quadrilateral mesh obtained on Geometry II, a nautilus.

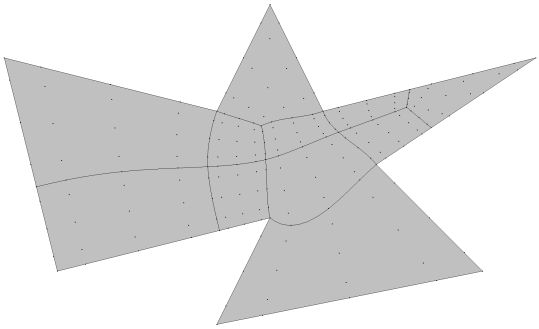

Figure 11: Coarse quadrilateral mesh obtained on Geometry III, a polygon.

Geometry IV (Fig. 12) illustrates the preservation of symmetries and patterns. The geometry is that of a simple gear and we observe the same irregular node pattern as seen in the cross field literature [1].

Whereas Geometries I–III illustrate the generation of naturally curved very coarse meshes, finer meshes are typically required for SEM analysis with user defined element size requirements. Geometry V (Fig. 13) illustrates the use of the isoparametric splitting to obtain a finer mesh with a certain distribution of elements. The geometry is that of NACA 0012 aerofoil, a typical geometry used in academic CFD studies.

We next demonstrate the use of p-adaptation. For Geometry I, we show the distribution of the local number of modes ($= P + 1$) after p-adaptation in Fig. 14. The local polynomial order is higher mostly near bound-

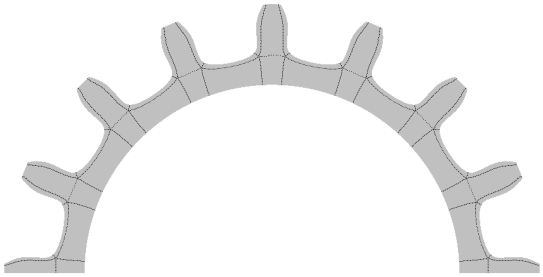

Figure 12: Block decomposition obtained on Geometry IV, a gear.

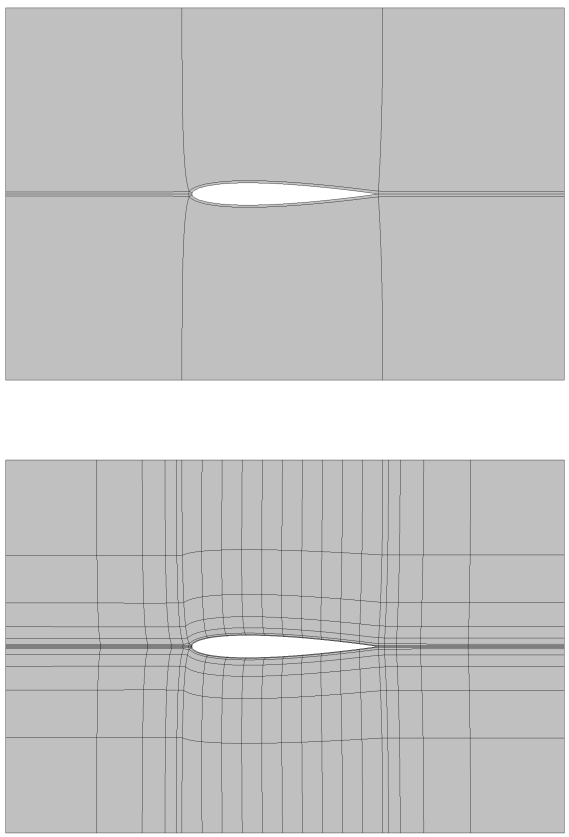

Figure 13: Coarse (top) and split (bottom) quadrilateral meshes obtained on Geometry V, a NACA 0012 profile. High order interior degrees of freedom have been hidden for clearer visualization.

aries and in the central area where rapid change in the solution is observed. The local polynomial order also seems unexpectedly high in the v-based case but this can be easily explained by the actual solution. Because the geometry is dominated by Cartesian-aligned straight-sided boundaries, a large part of the field (especially the areas in the bottom left and right) has values close to $\vec{v} = (1, 0)$. This means that the low order modes have no energy for the v component. Therefore, for the v based sensor, any energy appearing in the higher order modes will trigger an increase in the local polynomial order. For such examples dominated by Cartesian-aligned boundaries, it is therefore best to base the sensor on u since $u_b = 1$ in Cartesian directions. The same phenomenon is retrospectively observed for the half disc geometry in Fig. 5.

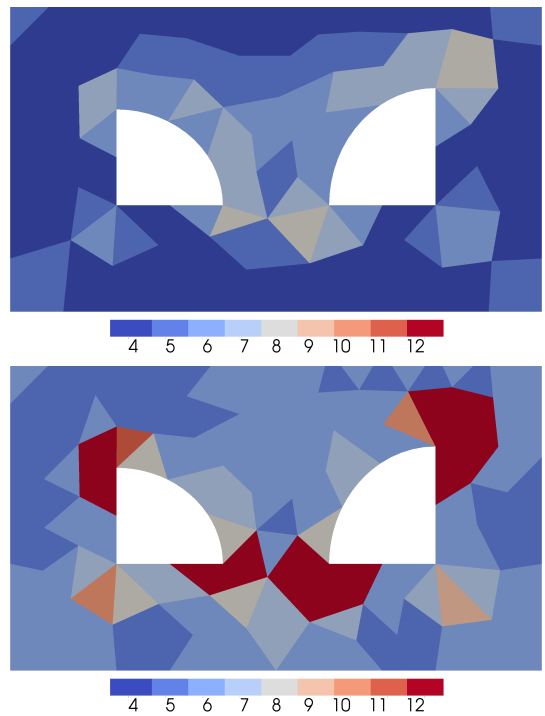

Figure 14: Distribution of local number of modes ($= P + 1$) for Geometry I. Adaptation based on u (top) and v (bottom).

The use of high order methods might intuitively sound prohibitively expensive for this method. We find, however, that the computational time remains low, with all examples in this paper taking $\mathcal{O}(10^0)$ seconds to process. The main cost in *NekMesh* is the computation of the streamlines because that is currently done in physical, not reference, space. Streamline integration currently takes between roughly 50% and 90% of the time of the whole procedure, depending on the size of the domain and of the base mesh, the integration step size, and how far streamlines have to travel. All other tasks — each solution of the Laplace problem, the analysis of the critical points and corners and the generation of a quad mesh — take about the same share of the remaining 10% to 50% of computational

time.

5. SUMMARY AND CONCLUSIONS

We have demonstrated the benefits of a field guided method to generate quadrilateral block decompositions, and meshes, of multiply connected two dimensional domains. The guiding field is inspired by cross fields but crosses are, in fact, never generated. This method generates naturally curved elements suitable for spectral element methods, and other related methods, and it does so by using a spectral element solver. The guiding field is obtained from a boundary value problem using a Laplace solver, with boundary conditions determined by local geometrical information. This information can be discontinuous, in which case a discontinuous Galerkin formulation is used for a discretization consistent way of imposing these discontinuous boundary conditions. The separatrix graph, obtained from the analysis of the guiding field, is later used to split the two dimensional model. From this block decomposed geometry, a coarse, valid, curved mesh is generated that can be refined to users' needs. The procedure is implemented in the open source spectral/*hp* element framework *Nektar++* and its high order mesh generation tool *NekMesh*.

We emphasize some of the benefits of this method including some advantages over traditional cross field based approaches as well. Whereas high order mesh generation techniques typical suffer from invalid elements, the method presented here creates coarse *a priori* curved meshes that are naturally valid. It does so by using a spectral element solver, which not only benefits from spectral convergence but is also suitable to p-adaptation for better computational resources. This creates a high resolution guiding field, due to the high order information of the solution, used for the creation of an optimal separatrix graph: small number of irregular nodes, preservation of symmetries, absence of limit cycles, etc. The analysis of the guiding field is performed in reference space at a low computational cost. The refinement of the coarse mesh is later performed using an isoparametric splitting approach that preserves the high order validity of the mesh.

The authors envision two improvements of this proof of concept for better relevance in the CFD community. On the one hand, an extension to three dimensional hexahedral block decompositions must be sought. This extension requires a reformulation of the problem based on three dimensional frame arithmetic. Promising work of Ray *et al.* [25] is paving the way in that direction. On the other hand, the generation of metric aligned quadrilateral blocks, as was done in reference [26], seems possible through the use of a forcing term. The solution of a Poisson problem with adequate forcing would enable modifications of the separatrix graph with respect to a metric field. Such a solution would also be appropriate for the relocation of irregular nodes away from boundaries with strong curvature (see for example Fig. 13).

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675008. This work was supported by a grant from the Simons Foundation (#426393, David Kopriva). The first two authors would also like to thank Prof. Gustaaf Jacobs of the San Diego State University for his hospitality. Finally, JM would like to thank the committee of the 28th International Meshing Roundtable for their travel financial support.

References

[1] Viertel R., Osting B. "An Approach to Quad Meshing Based on Harmonic Cross-Valued Maps and the Ginzburg-Landau Theory." 2017

[2] Kozdon J., Dunham E., Nordström J. "Simulation of Dynamic Earthquake Ruptures in Complex Geometries Using High-Order Finite Difference Methods." *Journal of Scientific Computing*, vol. 55, no. 1, 92–124, 2013

[3] Karniadakis G.E., Sherwin S.J. *Spectral/hp Element Methods for Computational Fluid Dynamics (Second Edition)*. Oxford University Press, 2005

[4] Kopriva D.A. *Implementing Spectral Methods for Partial Differential Equations*. Scientific Computation. Springer, May 2009

[5] Bommes D., Lévy B., Pietroni N., Puppo E., Silva C., Tarini M., Zorin D. "Quad-Mesh Generation and Processing: A Survey." *Computer Graphics Forum*, vol. 32, no. 6, 51–76, 2013

[6] Marcon J., Kopriva D., Sherwin S., Peiró J. "A High Resolution PDE Approach to Quadrilateral Mesh Generation." *Journal of Computational Physics*, 2019. In press

[7] Li Y., Premasuthan S., Jameson A. "Comparison of Adaptive h and p Refinements for Spectral Difference Methods." *40th Fluid Dynamics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Reston, Virigina, jun 2010

[8] Ekelschot D., Moxey D., Sherwin S., Peiró J. "A p-adaptation method for compressible flow problems using a goal-based error indicator." *Computers & Structures*, apr 2016

[9] Moxey D., Cantwell C., Mengaldo G., Serson D., Ekelschot D., Peiró J., Sherwin S., Kirby R.M. "Towards p-Adaptive Spectral/hp Element Methods for Modelling Industrial Flows." *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pp. 63–79. Springer, Cham, 2017

[10] Viertel R., Staten M., Osting B. "Toward a paver replacement." *26th International Meshing Roundtable, IMR26*. 2017

[11] Bunin G. "A Continuum Theory for Unstructured Mesh Generation in Two Dimensions." Tech. rep., sep 2006

[12] Bunin G. "Towards Unstructured Mesh Generation Using the Inverse Poisson Problem." *ArXiv e-prints*, Feb. 2008

[13] Tricoche X., Scheuermann G., Hagen H. "Continuous Topology Simplification of Planar Vector Fields." *Proceedings of the Conference on Visualization '01*, VIS '01, pp. 159–166. IEEE Computer Society, Washington, DC, USA, 2001

[14] Kowalski N., Ledoux F., Frey P. "A PDE based approach to multi-domain partitioning and quadrilateral meshing." *21th Int. Meshing Roundtable*. 2012

[15] Cantwell C., Moxey D., Comerford A., Bolis A., Rocco G., Mengaldo G., De Grazia D., Yakovlev S., Lombard J.E., Ekelschot D., Jordi B., Xu H., Mohamied Y., Eskilsson C., Nelson B., Vos P., Biotto C., Kirby R., Sherwin S. "Nektar++: An open-source spectral/hp element framework." *Computer Physics Communications*, vol. 192, 205–219, 2015

[16] Persson P.O., Peraire J. "Sub-Cell Shock Capturing for Discontinuous Galerkin Methods." *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Reston, Virigina, jan 2006

[17] Moxey D., Cantwell C., Bao Y., Cassinelli A., Castiglioni G., Chun S., Juda E., Kazemi E., Lackhove K., Marcon J., Mengaldo G., Serson D., Turner M., Xu H., Peiró J., Kirby R., Sherwin S. "Nektar++: enhancing the capability and application of high-fidelity spectral/*hp* element methods." *Computer Physics Communications*, 2019. Under review

[18] Sherwin S., Peiró J. "Mesh generation in curvilinear domains using high-order elements." *International Journal for Numerical Methods in Engineering*, vol. 53, no. 1, 207–223, jan 2002

[19] Coppola G., Sherwin S., Peiró J. "Nonlinear Particle Tracking for High-Order Elements." *Journal of Computational Physics*, vol. 172, no. 1, 356–386, sep 2001

[20] Li T., McKeag R., Armstrong C. "Hexahedral meshing using midpoint subdivision and integer programming." *Computer Methods in Applied Mechanics and Engineering*, vol. 124, no. 1-2, 171–193, jun 1995

[21] Open Cascade SAS. "Open Cascade.", 2018

[22] Moxey D., Green M., Sherwin S., Peiró J. "An isoparametric approach to high-order curvilinear boundary-layer meshing." *Computer Methods in Applied Mechanics and Engineering*, vol. 283, 636–650, jan 2015

[23] Marcon J., Turner M., Peiró J., Moxey D., Pollard C., Bucklow H., Gammon M. "High-order curvilinear hybrid mesh generation for CFD simulations." *2018 AIAA Aerospace Sciences Meeting*, January. American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2018

[24] Marcon J., Peiró J., Moxey D., Bergemann N., Bucklow H., Gammon M. "A semi-structured approach to curvilinear mesh generation around streamlined bodies." *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, Reston, Virginia, jan 2019

[25] Ray N., Sokolov D., Lévy B. "Practical 3D frame field generation." *ACM Transactions on Graphics*, vol. 35, no. 6, 1–9, nov 2016

[26] Fogg H.J., Armstrong C.G., Robinson T.T. "Multi-Block Decomposition Using Cross-Fields." J.P.M. de Almeida, P. Díez, C. Tiago, N. Pares, editors, *VI International Conference on Adaptive Modeling and Simulation*. 2013