A spectral element method to compute approximations of the anisotropic diffusion operator with bidimensional tensor coefficient

E A Calderon-Barreto¹, C A Acosta-Minoli¹, and J E Quiroga-Méndez²
¹ Grupo de Investigación GEDES, Universidad del Quindío, Armenia, Colombia
² Escuela de Ingeniería Mecánica, Universidad Industrial de Santander, Bucaramanga, Colombia
E-mail: cminoli@uniquindio.edu.co

Abstract. We derived a continuous Galerkin spectral element method to compute approximations of the anisotropic diffusion operator with the tensorial coefficient of order two. After writing the operator in the weak form, we calculated spatial integration using the Legendre base. Then, we applied three different methods for the solution of the associated linear system: The Lower–Upper factorization, the biconjugate gradient method and the biconjugate gradient stabilized method with the Richardson preconditioner. To validate the algorithm, we present a convergence study for the Poisson equation when varying the functions of the tensor coefficient. Results show how the algorithm can be used to construct solvers for more complex problems like the anisotropic diffusion-reaction equation.

1. Introduction
For some phenomena described with the diffusion-reaction equation, there is the need to consider particular characteristics from the medium to model the diffusion coefficient accurately. For example, the cardiac electrical activity with the two-domain or monodomain model where a tensor of order two represents the diffusion coefficient, there, the electrical potential propagates with higher speed in the direction of the muscle fibers [1,2]. We can find another example in ecology, particularly in models of populations of species where the diffusion coefficient is also a tensor of order two, that changes according to food paths, temperature or other factors [3]. Equation (1), presents a general form of the two-dimensional reaction anisotropic diffusion models.

\[
\frac{\partial u}{\partial t} = \nabla \cdot \sigma \nabla u + S(u, x, t) \quad \rightarrow \quad u(x, t = 0) = u_0 \quad \rightarrow \quad \vec{n} \cdot \nabla u = 0, \quad (1)
\]

where \( u \) is the state variable, \( S \) represents the reactive term, \( t \) is the temporal variable, \( x = (x, y) \) the spatial variable and \( \sigma \) is a space-dependent matrix of order two that represents the tensor diffusion coefficient.

On the other hand, the computational effort of diffusion-reaction models in some cases is demanding; this is because the integration steps must be of the order of hundredths of a millisecond and the spatial discretization must be very fine in order to obtain reliable results. Therefore, some features that would justify the use of spectral methods are:
While the expansions of the orthogonal polynomials can be of order two or three in the methods of finite differences and finite elements, spectral methods can be of the order of hundreds [4]. Spectral methods assume a global interpolation while finite element methods use a local interpolation [5,6].

Approximations with spectral methods can have very high convergence rates, which allow fewer degrees of freedom for the desired level of accuracy [4]. Spectral methods can be implemented with a "p-refinement and an h-refinement", although finite element codes that have been developed with this type of refinement are far from universal [5].

In this work, we derived a continuous Galerkin spectral element method to compute approximations of the anisotropic diffusion operator with tensorial coefficient of order two. After writing the operator of the anisotropic diffusion operator with tensorial coefficient of order two. After writing the operator in the weak form, we computed spatial integration using the continuous Galerkin spectral element method. Then we offer a convergence study when compare to an analytical solution for the Poisson equation.

2. Tensorial diffusion model
We use a particular case of the diffusion tensor, which depends on a vector field mainly used as a conductivity tensor model. Applications can be found when modelling the electrical activity in the heart muscle using partial differential equations since muscle has a fiber architecture, where the electrical conductivity is greater along the direction of the fibers; Equation (2) shows the model.

\[ \bar{\sigma}_{Le} = \sigma_{le}^\text{g} + (\sigma_{le}^1 - \sigma_{le}^0) \mathbf{r} \otimes \mathbf{r}, \]  

where \( \bar{\sigma}_{le} \) denotes the metric tensor; This component is taken as the identity matrix in an orthogonal coordinate system, the vector field \( \mathbf{r} \) is unitary and in the direction of the fibers, and \( \sigma_{le}^1 \) and \( \sigma_{le}^0 \) are the coefficients of the intracellular medium along and across the membrane respectively [7].

3. Spatial discretization
The first step in extending the spectral method to complex geometries is by deriving discrete operators in quadrilaterals with curved sides; for this, a transformation from the physical domain (\( \Omega_F \)) to the computational domain (\( \Omega_C \)) is created, which in this case is the square \([-1,1] \times [-1,1]\) since we will use Legendre polynomials.

3.1. Transfinite interpolation
Transfinite is a linear algebraic transformation which assigns to each point \( (\xi, \eta) \) of \( \Omega_C \), a point \((x,y)\) on \( \Omega_F \) by a mapping \( X = (x,y) = X(\xi, \eta) \) as shown in Figure 1 and Equation (3), [8]. This transformation requires information from the four curves \( \Gamma_1 \), \( \Gamma_2 \), \( \Gamma_3 \), \( \Gamma_4 \), opposite of pairs of the following: \( \Gamma_1 \) opposite to \( \Gamma_3 \) and \( \Gamma_2 \) opposed \( \Gamma_4 \) (see Figure 1). With this information the transformation \( X(\xi, \eta): \Omega_C \to \Omega_F \) can be applied as follows, Equation (3).

\[
X(\xi, \eta) = \frac{1}{2} \{ \Gamma_4(\eta)(1 - \xi) + \Gamma_2(\eta)(1 + \xi) + \Gamma_1(\xi)(1 - \eta)\Gamma_3(\xi)(1 + \eta) \} - \frac{1}{4} \{ \Gamma_1(-1)(1 - \xi)(1 - \eta) + \Gamma_3(-1)(1 + \xi) + \Gamma_1(1)(1 + \xi)(1 - \eta) + \Gamma_2(1)(1 + \eta) \},
\]

(\( \xi, \eta \)) must be a parameterization by arc length, this can be calculated by solving Equation (4) when searching for a \( t \) given for a \( s \) given (s is a Legendre zero), where \( s_0 = -1 \) and \( s_N = 1 \), in this way you can find the corresponding transformation of the Legendre nodes \((\xi, \eta) \in \Omega_C \text{ in } (x, y) \in \Omega_F \). Normally the Equation (4) is solved numerically with Newton’s method, [9,10].
\( s - s_0 = \frac{1}{L} \int_{t_0}^{1} \sqrt{(x'(z))^2 + (y'(z))^2} \, dz \).  

(4)

Figure 1. Transfinite interpolation of the nodes of a Gauss Lobato Legendre polynomial of order 6 in 2D to two different physical domains; one concave and another convex.

To discretize the gradient and divergent operators, some metric terms are necessary, such as the covariant and contravariant vectors in each node of the physical domain coordinate system, the Jacobian in each node and the normal vectors to the borders, all this parameterized in \((\xi, \eta)\).

### 3.2. Metric terms

Assuming that \(X(\xi, \eta)\) it is defined by transfinite interpolation, Equation (3), and deriving this expression you get Equation (5).

\[
\frac{\partial X}{\partial \xi} = \frac{1}{2} \left\{ \Gamma_2(\eta) - \Gamma_4(\eta) + (1 - \eta)\Gamma'_1(\xi) + (1 + \eta)\Gamma'_3(\xi) \right\} \\
- \frac{1}{4} \left\{ (1 - \eta)[\Gamma_1(1) - \Gamma_1(-1)] + (1 + \eta)[\Gamma_3(1) - \Gamma_3(-1)] \right\},
\]

\[
\frac{\partial X}{\partial \eta} = \frac{1}{2} \left\{ (1 - \xi)\Gamma'_4(\eta) + (1 + \xi)\Gamma'_2(\eta) + \Gamma_3(\xi) - \Gamma_1(\xi) \right\} \\
- \frac{1}{4} \left\{ (1 - \xi)[\Gamma_3(1) - \Gamma_3(-1)] + (1 + \xi)[\Gamma_3(1) - \Gamma_1(1)] \right\}.  
\]

(5)

These vectors help build the base of covariant vectors and the Jacobian transformation, as shown in Equation (6).

\[
\mathbf{a}_1 = \frac{\partial X}{\partial \xi} = X_\xi \hat{x} + Y_\xi \hat{y}, \quad \rightarrow \quad \mathbf{a}_2 = \frac{\partial X}{\partial \eta} = X_\eta \hat{x} + Y_\eta \hat{y},
\]

(6)

where \(J\) is the Jacobian defined in Equation (7).

\[
J = \mathbf{a}_1 \cdot (\mathbf{a}_1 \times \mathbf{a}_k) = X_\xi Y_\eta - X_\eta Y_\xi.  
\]

(7)

Using the base of contravariant vectors Equation (6) to obtain the non-conservative form of the divergent of a vector field \(\mathbf{F}\) and the gradient of a scalar field \(f\), from the formulas presented in Equation (8).

\[
\nabla \cdot \mathbf{F} = \frac{1}{J} \sum_{i=1}^{3} (\mathbf{a}_i \times \mathbf{a}_k) \frac{\partial F}{\partial \xi^i} \quad \rightarrow \quad \nabla f = \frac{1}{J} \sum_{i=1}^{2} (\mathbf{a}_i \times \mathbf{a}_k) \frac{\partial f}{\partial \xi^i}.
\]

(8)
For Neumann boundary conditions it is necessary to know the normal vectors to the boundary curves. As shown in Equation (9).

$$\vec{n}^1 = \frac{\| Y_\xi X_\eta \|}{\sqrt{Y_\xi^2 - X_\eta^2}} \quad \rightarrow \quad \vec{n}^2 = \frac{\| -Y_\xi X_\eta \|}{\sqrt{Y_\xi^2 - X_\eta^2}}.$$  
(9)

3.3. Potential approximation

To analyze the approximation of the spectral method, the solutions of a modified Poisson problem Equation (10) were studied, where $\sigma$ it is a tensor coefficient dependent on spatial variables and $S$ is the source.

$$\nabla \cdot \sigma \nabla u = S(u, x) \quad \rightarrow \quad u(\partial \Omega) = g(x, y).$$  
(10)

The approximation will be given by the degree interpolation $n$ of the potential $u$ and the test function $v$ as show in Equation (11).

$$u \approx \Phi(x, y) = \sum_{i=0}^{n} \Phi_{ij} l_i(x) l_j(y) \quad \rightarrow \quad v \approx \phi(x, y) = \sum_{i=0}^{n} \phi_{ij} l_i(x) l_j(y).$$  
(11)

Starting from the weak formulation of Equation (10) then integrating and multiplying by $\phi$ and $W$, where $\phi$ satisfies the boundary conditions and $W$ is a function appropriate to the polynomial base, you will get Equation (12).

$$- \sum_{i=0}^{n} \nabla \phi_{ij} \cdot \sigma \nabla \phi_{ij} l_i^F w_i^\eta + \int_{\partial \Omega_c} n \sigma \nabla \phi \cdot \nabla \phi dH = \sum_{i=0}^{n} \phi_{ij} s_{ij} l_i^F w_i^\eta,$$  
(12)

where $J$ is the Jacobian of transformation, Equation (7), $w_i^\xi$ y $w_i^\eta$ are Legendre’s weights after integration by parts and applying the quadrature of Gauss Legendre Lobatto. To calculate the operator $\sigma \nabla \Phi$ we define de tensor $\sigma$ in Equation (13).

$$\sigma = \begin{pmatrix} f_1(x, y) & f_2(x, y) \\ f_3(x, y) & f_4(x, y) \end{pmatrix},$$  
(13)

and with this, the operator $\sigma \nabla u$, is discretized as shown in Equation (14):

$$\sigma \nabla \Phi_{ij} = \left[ \begin{array}{c} F_{ij} \\ G_{ij} \end{array} \right] = \begin{pmatrix} A \sum_{k=0}^{N} D_{ik}^F \Phi_{ik} - B \sum_{k=0}^{N} D_{ik}^H \Phi_{ik} \sum_{i=0}^{n} \phi_{ij} s_{ij} l_i^F w_i^\eta \\ D \sum_{k=0}^{N} D_{ik}^H \Phi_{ik} - C \sum_{k=0}^{N} D_{ik}^F \Phi_{ik} \sum_{i=0}^{n} \phi_{ij} s_{ij} l_i^F w_i^\eta \end{pmatrix}.$$  
(14)

where $D_{ij}^F = l_i^F(\xi_j)$ y $D_{ij}^H = l_i^H(\eta_j)$ are matrices derived and A, B, C, D are defined in Equation (15):

$$A = f_1 Y_\xi - f_2 X_\xi Y_\eta - f_3 X_\eta + f_4 X_\eta,$$
$$B = f_1 Y_\xi - f_2 X_\xi Y_\eta - f_3 X_\eta + f_4 X_\eta,$$
$$C = f_1 Y_\xi - f_2 X_\xi Y_\eta - f_3 X_\eta + f_4 X_\eta,$$
$$D = f_1 Y_\xi - f_2 X_\xi Y_\eta - f_3 X_\eta + f_4 X_\eta.$$  
(15)

By replacing Equation (15) in Equation (12) you will get, Equation (16).
\[ w_j^N \sum_{k=0}^{N} D_{jk}^n F_{k+l} + w_j^\xi \sum_{k=0}^{N} D_{jk}^n G_{lk} = I_{ll} S_{lj} w_l^\xi w_j^N - \int_{\partial \Omega} \mathbf{n} \sigma \nabla \Phi \, dH. \] (16)

The right part of the Equation (16) defines a linear transformation with which it is possible to apply an iterative method; however, to apply a direct method it is necessary to find the matrix associated with the transformation by identifying each of its coefficients, as shown in Equation (17).

\[-\nabla \cdot \sigma \nabla \Phi_{ij} = - \sum_{k=0}^{N} \left( \sum_{s=0}^{N} D_{s}^\xi w_{i} A_{s} D_{sk}^\xi w_{s} \right) \Phi_{ik} + \sum_{k=0}^{N} \sum_{s=0}^{N} \left( D_{s}^\xi w_{i} B_{sk} D_{jk}^\xi w_{s} \right) \Phi_{sk} \]

\[-\sum_{k=0}^{N} \left( \sum_{s=0}^{N} D_{s}^\xi w_{i} D_{ls}^\xi w_{s} \right) \Phi_{ik} + \sum_{k=0}^{N} \sum_{s=0}^{N} \left( D_{s}^\xi w_{i} C_{ls} D_{jk}^\xi w_{s} \right) \Phi_{sk}, \]

Finally, the boundary is approximated as shown in Equation (18),

\[ \int_{\partial \Omega} \mathbf{n} \sigma \nabla \Phi \, dH = \sum_{k=0}^{N} \mathbf{n}^{-\frac{g}{h+4}} \cdot \sigma \nabla \Phi_{hk} w_{k} + \sum_{k=0}^{N} \mathbf{n}^{\frac{g}{h+4}} \cdot \sigma \nabla \Phi_{lk} w_{l}. \] (18)

Here, index \( h \) and index \( l \) can take the values \( \{0, N\} \).

3.4. Spectral element generalization
To approximate Equation (10) in the internal nodes of an element \( k \), for \( i \) and \( j \) different from 0 or \( N \), you'll have Equation (19).

\[ (-\nabla \cdot \sigma \nabla \Phi_{ij})^{[k]} = (I_{ll} S_{lj} w_l^\xi w_j^N)^{[k]} - \left( \int_{\partial \Omega} \mathbf{n} \sigma \nabla \Phi \, dH \right)^{[k]}. \] (19)

For the nodes that are in the border of an element, that is, when any of the indexes \( i, j \), take any of the values \( 0, N \) and assuming that the node is shared by the family of elements \( k_1, k_2, k_3, ..., k_m \) the approximation of Equation (10) is shown in Equation (20).

\[ \sum_{l=1}^{m} (-\nabla \cdot \sigma \nabla \Phi_{ij})^{[kl]} = \sum_{l=1}^{m} \left( (I_{ll} S_{lj} w_l^\xi w_j^N)^{[kl]} - \left( \int_{\partial \Omega} \mathbf{n} \sigma \nabla \Phi \, dH \right)^{[kl]} \right). \] (20)

4. Numerical results for Poisson equation
In this subsection, we validate approximations obtained by Equation (18) and Equation (19), considering a study of the convergence of the method by varying the polynomial order and quantity of elements together with the conditionality when changing the diffusion tensor. The system was solved with three different methods; the first is a direct method using the partial-pivot Lower-Upper (LU) decomposition using the DGESV function of the LAPACK library, the second is the biconjugated gradient method and the third is the biconjugated gradient method stabilized with the Richardson preconditioner [4,11]. Here, an exact analytical solution in closed form was used to verify the algorithm, the solution that we use is shown in Equation (21).

\[ u(x, y) = \cos(2\pi x) \sin(2\pi y). \] (21)

For the parameter \( \sigma \) we used the model Equation (2), varying the vector field \( \mathbf{r} = (r_1, r_2) \) and the coefficients \( \sigma^1 \) and \( \sigma^2 \), as defined in Equation (22).

\[ \sigma(x, y) = \begin{pmatrix} f_1 = \sigma^1 + (\sigma^1 - \sigma^2) r_2^2(x, y) & f_2 = (\sigma^2 - \sigma^1) r_1(x, y) r_2(x, y) \\ f_3 = (\sigma^2 - \sigma^1) r_1(x, y) r_2(x, y) & f_4 = \sigma^1 + (\sigma^2 - \sigma^1) r_2^2(x, y) \end{pmatrix}. \] (22)
The solution Equation (20) next to the tensor Equation (21) modifies the source term, as shown in Equation (23).

\[ S(x, y) = f_1 u_x + f_1 u_{xx} + f_2 u_y + f_2 u_{yx} + f_3 u_x + f_3 u_{xy} + f_4 u_y + f_4 u_{yy}. \]  

(23)

The vector fields used in simulation are shown in Equation (24).

\[ \begin{align*}
\tau^1 &= \left( x^8 - y^3 \right) \\
\tau^2 &= \left( -\sin(x) \right) \\
\tau^3 &= \left( \exp(-x^2) \right) \exp(-y^2) \\
\end{align*} \]  

(24)

Plot in Figure 2 shows the convergence of the error as a function of the polynomial order approximation for each the vector fields proposed. We observe spectral converge in space as expected from a smooth solution [6,12]. The error was computed using max norm when compared with the exact solution. It is known from theory that the conditional number of matrices associated with the spectral element method grows with increasing polynomial order \( N \), [6]. The incorporation of the tensor coefficient generates a slow convergence; we study this consequence by analyzing the number of iterations in the iterative methods and computation time, see Table 1.

![Figure 2](image-url)

**Figure 2.** The logarithm of maximum error when comparing the approximate solution using algorithm Equation (17) with the exact solution Equation (19) for four different vector fields.

**Table 1.** We Show the computation time in milliseconds \((t_c)\) and the number of iterations \((I_t)\) of two iterative methods.

| \( N \) | \( t_c \) | \( I_t \) | \( t_c \) | \( I_t \) | \( t_c \) |
|---|---|---|---|---|---|
| 6  | 39 | 52 | 10 | 47 | 83 |
| 7  | 37 | 64 | 10 | 60 | 47 |
| 8  | 39 | 80 | 14 | 77 | 58 |
| 9  | 44 | 94 | 22 | 89 | 90 |
| 10 | 51 | 116 | 29 | 104 | 120 |
| 11 | 63 | 129 | 39 | 112 | 228 |
| 15 | 171 | 200 | 110 | 170 | 688 |
| 20 | 394 | 289 | 295 | 235 | 2744 |
| 25 | 981 | 427 | 695 | 325 | 8573 |
| 29 | 1721 | 524 | 1256 | 403 | 19021 |
| 30 | 1965 | 551 | 1329 | 392 | 23731 |
Finally, in Figure 3, it can be seen how the conditional number increases as the polynomial order increases, this fact is normal in spectral methods, on the other hand two important aspects can be evidenced: first the influence of the increase in the amount of elements and secondly the influence of the diffusion tensor. Additionally, the Figure 3 show of the variation of the reciprocal logarithm of the conditional number of the three matrices associated to the linear operator versus polynomial order, for the three vector fields $\tau^1$, $\tau^2$ and $\tau^3$. Here 1E means an element and 4E are four elements. In Table 1 we show the computation time in milliseconds ($t_c$) and the number of iterations ($I_t$) of two iterative methods: BiCG biconjugated gradient and stabilized biconjugated gradient with the Richardson BiCGSTAB preconditioner. Together with the computation time in milliseconds of a DIRECT method using the DGESV function of the LAPACK library. This approximate solution of Equation (10), diffusion tensor Equation (20) and vector field $\tau^3$.

![Figure 3. Variation of the reciprocal logarithm of the conditional number.](image)

5. Conclusions
In this work, we derived a continuous Galerkin spectral element method to compute approximations of the anisotropic diffusion operator with tensorial coefficient of order two. Numerical convergence study based on the analytic solution, allowed us to validate the algorithm evidencing spectral convergence in space. The presented results agreed with those observed in the literature, which allows concluding that from the computational point of view, we implemented the approach correctly.

Acknowledgments
This research was supported by Universidad del Quindío, Armenia, Quindío, Colombia.

References
[1] Streeter D 1979 *Handbook of Physiology* (Baltimore: Williams & Wilkins)
[2] Schlemmer A, Berg S, Shajahan T K, Parritz U 2015 *Entropy* 17(3) 950-967
[3] Simon T and Koya P R 2016 *American Journal of Applied Mathematics* 3(6) 335-340
[4] Kopriva D A 2009 *Implementing Spectral Methods for Partial Differential Equations* (Springer Netherlands)
[5] Boyd J P 2000 *Chebyshev and Fourier Spectral Methods* (New York: Dover Publications, Inc.)
[6] Canuto C, Hussaini M Y, Quarteroni A and Zang T A 2006 *Spectral Methods: Fundamentals in Single Domains* (New York: Springer Berlin: Heidelberg)
[7] Collin A 2017 *Asymptotic Analysis in Cardiac Electrophysiology: Applications in Modeling and in Data Assimilation* (Paris: Université Pierre ET Marie Curie)
[8] Acosta-Minoli C A, Kopriva D A 2011 *Journal of Computational Physics* 230(5) 1876-1902
[9] Kopriva D A, Gassner G J 2016 *Applied Mathematics and Computation* 272(2) 274-290
[10] Kopriva D A, Winters A R, Bohm M, Gassner G J 2016 *Computers & Fluids* 139(5) 148-160
[11] Quarteroni A, Sacco R, Saleri F 2001 *Numerical Mathematics* (Berlin: Heidelberg Spring-Verlag)
[12] Rasetarinera P, Kopriva D A, Hussaini M Y 2000 Discontinuous spectral element solution of aeroacoustic problems *Third Computational Aeroacoustics (CAA)* (Cleveland: NASA)