Compute Approximated Geodesic Distance on 3D Mesh by the Heat Method using FEM

Mauricio Cele Lopez Belon

Abstract. Compute approximated geodesic distance on the mesh by the Heat Method using Finite Element Method (FEM).

Mathematics Subject Classification (2010). Parallel algorithms 68W10; Clifford algebras, spinors 15A66.

Keywords. Geometric Algebra, Quaternion Estimation, Mesh Deformation.

1. Introduction

This follows the method described in [3] and [1].

The goal is to compute the approximated geodesic Φ from a points given by boundary conditions to all other points in the mesh. The Heat Method works as follows: first solves the Heat Equation PDE $U\ dt = \triangle U$ with dirichlet boundary conditions using Finite Element Method (FEM)

$$\begin{array}{ll} \text{Solve } \triangle U(\vec{X},t) = \frac{\partial U(\vec{X},t)}{\partial t} & \text{in region } \Omega \\ \text{s.t. } U(\vec{X},t) = G(\vec{X}) & \text{on the region boundary } \partial\Omega \text{ for all times} \\ U(\vec{X},0) = U_0(\vec{X}) & \text{initial condition at time t = 0} \end{array}$$

The resulting function $U(\vec{X})$ is the diffusion of an small instant, which happens in geodesic path. Then computes the negative of the gradient of $U(\vec{X})$ on every triangle element and normalizes it:

$$\text{Solve } X(\vec{X}) = -\nabla U(\vec{X})/\|\nabla U(\vec{X})\| \quad \text{in triangle } T$$

The resulting $X(\vec{X})$ is a "discrete" vector field (potential) with directions towards the geodesic paths (i.e., pointing towards the steepest change). The divergence $\nabla \cdot X(\vec{X})$ is the "strength" of vector field X on a particular point. Then solves a Poisson Equation PDE with dirichlet boundary conditions:

$$\begin{array}{ll} \text{Solve } \triangle\Phi(\vec{X},t) = \nabla \cdot X(\vec{X}) & \text{in region } \Omega \\ \text{s.t. } \Phi(\vec{X},t) = Q(\vec{X}) & \text{on the region boundary } \partial\Omega \text{ for all times} \end{array}$$

Where Φ is the smooth function interpolating the boundary conditions which is interpreted as the geodesic distance. Finally shifts the distance Φ so the lowest value is zero.

In this document we mostly discuss how to solve the Heat PDE using FEM. We don't discuss the Poisson PDE since is very similar to Heat PDE but simpler.

1.1. The Heat PDE

Since $\frac{\partial U(\vec{X},t)}{\partial t}$ is a time derivative we discretize the time using Backguard Euler

$$\frac{\partial U(\vec{X},t)}{\partial t} = \frac{U(\vec{X},t+\Delta t) - U(\vec{X},t)}{\Delta t} \tag{1.1}$$

Replacing on original PDE for backguard euler:

$$\frac{U(\vec{X},t+\Delta t) - U(\vec{X},t)}{\Delta t} = \triangle U(\vec{X},t+\Delta t) \tag{1.2}$$

Re-arranging:

$$U(\vec{X},t+\Delta t) - \Delta t \triangle U(\vec{X},t+\Delta t) = U(\vec{X},t) \tag{1.3}$$

or better:

$$U(\vec{X},t_i) - \Delta t \triangle U(\vec{X},t_i) = U(\vec{X},t_{i-1}) \tag{1.4}$$

The integral form (notice $\int_\Omega$ is a double integral):

$$\int_\Omega U(\vec{X},t_i)\ dA - \Delta t \int_\Omega \triangle U(\vec{X},t_i)\ dA = \int_\Omega U(\vec{X},t_{i-1})\ dA \tag{1.5}$$

2. Weak Form

The weak form comes from the "variational method". Here one tries to get a solution to the PDE by translating it into a minimization problem. That's done by definig an energy function that is a linear functional on a function space Ω. The linear functional is artificial i.e., pure mathematical device to enable optimization. In consequence, solving the variational problem may require less continuity than that of the actual solution (hence the name WEAK form).

Consider the weight function $\vartheta(x,y)$ (a.k.a. test function) with compact support defined on Ω. The "Galerkin" variational method states that the inner product of $\vartheta(x,y)$ with a residual function $R(x,y)$ should be zero.

$$\int_\Omega \vartheta(\vec{X}) R(\vec{X})\ dA = 0$$

where $R(x,y)$ is the residual or error of the function i.e.,

$$R(x,y) = U(\vec{X},t_i) - \Delta t \triangle U(\vec{X},t_i) - U(\vec{X},t_{i-1})$$

The weak form of the PDE is:

$$\begin{aligned} \vartheta(\vec{X})R(\vec{X}) &= 0 \\ \vartheta(\vec{X})(U(\vec{X},t_i) - \Delta t \triangle U(\vec{X},t_i) - U(\vec{X},t_{i-1})) &= 0 \\ U(\vec{X},t_i)\vartheta(\vec{X}) - \Delta t\vartheta(\vec{X}) \triangle U(\vec{X},t_i) &= U(\vec{X},t_{i-1})\vartheta(\vec{X}) \end{aligned}$$

and the integral (variational) form of the PDE is

$$\int_\Omega U_t\vartheta \; dA - \Delta t \int_\Omega \vartheta \triangle U_t \; dA = \int_\Omega U_{t-1}\vartheta \; dA \tag{2.1}$$

NOTE on inner products: The integral form is more important since integral of the product of two functions $f(x,y)$ and $g(x,y)$ is the "inner product" $\int_\Omega f(x,y)g(x,y) \; dA$. If function $g(x,y)$ is a "test function" then the inner product is way to "measure" the function $f(x,y)$ on location where $g(x,y)$ is defined. Since $g(x,y)$ is not the Dirac Delta function then the measure is an averaged one.

Consider the vector field

$$E(\vec{X}) = \vartheta(\vec{X})\nabla U(\vec{X}) \tag{2.2}$$

Taking $\nabla \cdot E(\vec{X})$ by Green's first identity we get:

$$\begin{aligned} \nabla \cdot E &= \nabla \cdot (\vartheta \; \nabla U) \\ \nabla \cdot E &= \nabla\vartheta \cdot \nabla U + \vartheta \triangle U \end{aligned} \tag{2.3}$$

By applying the Divergence theorem:

$$\int_\Omega \nabla \cdot E \; dA = \int_{\partial\Omega} E \cdot n \; dS \tag{2.4}$$

where $n(\vec{X})$ is the normal to the boundary $\partial\Omega$

Replacing (2.3) on (2.4) we get:

$$\int_\Omega \nabla\vartheta \cdot \nabla U + \vartheta \triangle U \; dA = \int_{\partial\Omega} E \cdot n \; dS$$

By definition of $E(\vec{X}) = \vartheta(\vec{X})\nabla U(\vec{X})$ the RHS becomes:

$$\begin{aligned} \int_\Omega \nabla\vartheta \cdot \nabla U + \vartheta \triangle U \; dA &= \int_{\partial\Omega} \vartheta\nabla U \cdot n \; dS \\ &= \int_{\partial\Omega} \vartheta \left(\nabla U \cdot n\right) dS \end{aligned} \tag{2.5}$$

since test function $\vartheta(\vec{X})$ can be considered "constant" at the boundary.

So we can re-arrange the left-hand-side to get:

$$\int_\Omega \vartheta \triangle U \; dA = \int_{\partial\Omega} \vartheta \left(\nabla U \cdot n\right) \; dS - \int_\Omega \nabla\vartheta \cdot \nabla U \; dA \tag{2.6}$$

Replacing the left-hand-side of our variational form of PDE (2.1) with the above we get:

$$\int_\Omega U_t\vartheta \; dA - \Delta t \left(\int_{\partial\Omega} \vartheta \left(\nabla U_t \cdot n\right) \; dS - \int_\Omega \nabla\vartheta \cdot \nabla U_t \; dA \right) = \int_\Omega U_{t-1}\vartheta \; dA \tag{2.7}$$

We now define that all $\vartheta(\vec{X})$ at the boundary $\partial\Omega$ has zero value, which is a necessary condition for the variational problem to converge e.g., test function must have compact support.

$$\vartheta(\vec{X}) = 0 \quad \forall \vec{X} \in \partial\Omega \tag{2.8}$$

So the weak form is now:

$$\int_\Omega U_t \vartheta \, dA + \Delta t \int_\Omega \nabla\vartheta \cdot \nabla U_t \, dA = \int_\Omega U_{t-1} \vartheta \, dA \tag{2.9}$$

3. Discretization

The domain Ω has to be meshed in N nodes and T triangles. For linear elements the nodes coincides with mesh vertices (but for quadratic elements there will be nodes defined on the triangle's edge mid-points, so there can be more nodes than vertices).

The first discretisation is easy. We discretize the test functions

$$\vartheta(\vec{X}) = \sum_i \varphi_i(\vec{X}) \tag{3.1}$$

as a linear combination of $\varphi_i(\vec{X})$ over the entire domain Ω. $\varphi_i(\vec{X})$ is a function with compact support defined at every node of Ω to be equal to 1 at center node $\vec{N}_i$, equal to 0 at sorrounding nodes N_j and linear everywhere else inside triangles in triangle-set Ω_i a.k.a., a tent function:

$$\begin{aligned} \varphi_j(\vec{N}_j) &= 1 \quad \forall j = i \\ \varphi_j(\vec{N}_j) &= 0 \quad \forall j \neq i \\ \varphi_j(\vec{X}) &= linear \ in \ \Omega_i \end{aligned}$$

when restricted to individual triangle, test functions defined at nodes are also known as barycentric coordinates of triangles.

Plugging (3.1) into (2.9) we get a system of integral equations, one equation per triangle-set Ω_i corresponding to the compact support of test function $\varphi_i(\vec{X})$:

$$\begin{aligned} \int_\Omega U_t \sum_i \varphi_i \, dA + \Delta t \int_\Omega \nabla \sum_i \varphi_i \cdot \nabla U_t \, dA &= \int_\Omega U_{t-1} \sum_i \varphi_i \, dA \\ \int_\Omega \sum_i U_t \ \varphi_i \, dA + \Delta t \int_\Omega \sum_i \nabla\varphi_i \cdot \nabla U_t \, dA &= \int_\Omega \sum_i U_{t-1} \ \varphi_i \, dA \\ \sum_i \int_\Omega U_t \ \varphi_i \, dA + \sum_i \Delta t \int_\Omega \nabla\varphi_i \cdot \nabla U_t \, dA &= \sum_i \int_\Omega U_{t-1} \ \varphi_i \, dA \end{aligned}$$

where the integrals are now much smaller i.e., per triangle-set Ω_i around nodes $\vec{N}_i$ or tent function $\varphi_i(\vec{X})$

The second discretization is of solution $U(\vec{X}, t)$. We define the global solution $U(\vec{X}, t)$ to be a finite sum:

$$U(\vec{X}, t) = \sum_{j}^{N} U_t^j \psi_j(\vec{X}) \tag{3.2}$$

where U_t^j are coefficients to be determined at each node on entire domain Ω and $\psi_j(\vec{X})$ are the so-called "shape functions" (i.e., interpolation functions) also defined with compact support of triangle-sets Ω_i.

Functions defined on triangle-set Ω_i around node $\vec{N}_i$ to be equal to 1 at center node $\vec{N}_i$, equal to 0 at sorrounding nodes N_j and linear everywhere else at triangles in Ω_i a.k.a., a tent function also known as P1 linear functions.

$$\begin{aligned} \psi_j(\vec{N}_j) &= 1 \quad \forall j = i \\ \psi_j(\vec{N}_j) &= 0 \quad \forall i \neq j \\ \psi_j(\vec{X}) &= linear\ in\ \Omega_i \end{aligned}$$

when restricted to individual triangles, shape functions are also known as barycentric coordinates of triangles.

At this point we can see $\varphi_i(\vec{X})$ and $\psi_i(\vec{X})$ are the same, but they don't have to be the same. When those that the same the FEM is so-called "Galerkin FEM".

The third discretization is of the gradients $\nabla U(\vec{X}, t)$ and $\nabla \vartheta(\vec{X})$ using (3.2) and (3.1). The grandient of $U(\vec{X}, t)$ is just the linear combination of gradients at the triangle nodes:

$$\begin{aligned} \nabla U(\vec{X}, t) &= \nabla \left(\sum_i U_t^i \psi_i(\vec{X}) \right) \\ \nabla U(\vec{X}, t) &= \sum_i U_t^i \nabla \psi_i(\vec{X}) \end{aligned} \tag{3.3}$$

where the las equality holds by product rule. The gradient is defined in terms of $\nabla \psi_i(\vec{X})$ and U_t^i can be treated as a "constant" coefficient that need to be determined at every node. Note that triangles sharing vertices also shares the same U_i coefficient. $\nabla U(\vec{X}, t)$ is a smooth function inside triangles so it gives us a "continuous" gradient at the whole Ω.

The gradient of test functions, is defined in the same manner:

$$\nabla \vartheta(\vec{X}) = \sum_i \nabla \varphi_i(\vec{X}) \tag{3.4}$$

Now we can discretize the inner product $\langle \nabla\vartheta(\vec{X}), \nabla U(\vec{X},t)\rangle$ which is characteristic of the Laplacian:

$$\begin{aligned}\nabla\vartheta\cdot\nabla U_t &= \sum_i \nabla\varphi_i \cdot \sum_j U_t^j \nabla\psi_j \\ &= \sum_{i,j} \nabla\varphi_i \cdot U_t^j \nabla\psi_j \\ &= \sum_{ij} U_t^j \left(\nabla\varphi_i \cdot \nabla\psi_j\right) \end{aligned} \tag{3.5}$$

summation inidices run for all connected nodes i and j.

Plugging (3.5) into (2.9) we get a system of linear equations

$$\begin{aligned}\int_\Omega \left(\sum_i U_t^i\psi_i\right)\left(\sum_i \varphi_i\right)\, dA + \Delta t \int_\Omega \sum_{ij} U_t^j \left(\nabla\varphi_i\cdot\nabla\psi_j\right)\, dA \\ = \int_\Omega \left(\sum_i U_{t-1}^i\psi_i\right)\left(\sum_i \varphi_i\right)\, dA \\ \sum_{ij} U_t^j \int_\Omega \psi_j\varphi_i\, dA + \sum_{ij} U_t^j \Delta t \int_\Omega \nabla\varphi_i\cdot\nabla\psi_j\, dA = \sum_{ij} U_{t-1}^j \int_\Omega \psi_j\varphi_i\, dA \\ \sum_{ij} U_t^j \left(\int_\Omega \psi_j\varphi_i\, dA + \Delta t \int_\Omega \nabla\varphi_i\cdot\nabla\psi_j\, dA\right) = \sum_{ij} U_{t-1}^j \int_\Omega \psi_j\varphi_i\, dA\end{aligned}$$

Which can be written in matrix form as follows. First define a set of "stiffness" matrices L_T^i per triangle of size $N\times N$ defined as:

$$L_T^k(i,j) = \int_T \nabla\varphi_i\cdot\nabla\psi_j\, dA$$

inidices (i,j) run for all nodes of triangle T.

Then define the "mass"" matrix M_T^i per triangle of size $N\times N$ as:

$$M_T^k(i,j) = \int_T \varphi_i\psi_j\, dA$$

The $N\times N$ stiffness matrix L can be assembled as $L=\sum_k L_T^k$, similarly the mass matrix M ca be assembled as $M=\sum_k M_T^k$ to form a global $N\times N$ matrix system (we choose to have shape functions ψ_i equal to test functions φ_i so we have same number of equations than unknowns)

$$\begin{aligned}(M+\Delta t\ L)U_t &= MU_{t-1} \\ M^{-1}(M+\Delta t\ L)U_t &= U_{t-1} \\ (I+\Delta t\ M^{-1}L)U_t &= U_{t-1}\end{aligned} \tag{3.6}$$

subject to boundary conditions

3.1. Optimization

Linear system can be simplified by approximating the quadrature rule for the "mass" matrix instead of using the exact quadrature rule. Exact quadrature rule for mass matrix (which is a quadratic form) evaluates the integral at mid-point of the triangle edges. If instead we evaluate the integral at the vertices themselves we get a linear approximation. The benefit is that the mass matrix becomes a diagonal matrix where the diagonal values are $\frac{1}{3}$ multiplied by the area of the face (which is constant).

On each vertex that amounts to the sum of the areas of the sorrounding faces multiplied by $\frac{1}{3}$.

$$M_T^k(i,i) = \frac{1}{3}\sum_{ij} area\ of\ face\ j\ indident\ to\ vertex\ i \tag{3.7}$$

That diagonal matrix can be easily inverted by inverting the diagonal entries so we get:

$$(I + \Delta t\ M^{-1}L)U_t = U_{t-1} \tag{3.8}$$

where $M^{-1}L$ amounts to multiply the entire ith row of L with the ith diagonal value of M. This is convinient as there is no need to explcitly find M^{-1} or multiply the right-hand-side times M

4. Assembly

The process of creating the stiffness matrix M is called assembly. It consist of creating the matrix M_k for each triangle T_k and then add it to the global M matrix.

The global matrix M is $N \times N$ where N is number of mesh nodes. In previous section 3 the matrix M_k is a $N \times N$ super-sparse matrix just with a few non-zero entries which corresponds to a single triangle contribution.

$$M = \sum_k M_k \tag{4.1}$$

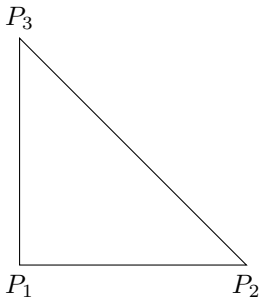
In practice the M_k matrix is not $N \times N$ super-sparse, but a 3×3 matrix (in case of linear triangle elements). There is a map from indices of the 3×3 matrix to the $N \times N$ matrix as follows.

Each triangle's nodes has a global index assigned in the mesh, we also establish a local indexing of triangle like 1, 2 and 3. We define a trivial bijective map between global and local indices.

5. Reference Elements

One of the novelties of FEM is to be able to compute the integrals per individual elements on an easy and generic way by using so called Reference Elements.

For triangle elements we just need a single reference triangle



$$P_1 = (0,0)$$
$$P_2 = (1,0)$$
$$P_3 = (0,1)$$

The P1 (linear) 2D Lagrange shape functions $\psi(u,v)$ has the form

$$\psi(u,v) = Au + Bu + C$$

For the reference trignale defined above the $\psi(u,v)$ takes the simple form:

$$\psi_1(u,v) = 1 - u - v$$
$$\psi_2(u,v) = u$$
$$\psi_3(u,v) = v$$

We can check that $\psi_i \cdot \psi_j = \delta_{ij}$

$$\psi_1(P_1) = 1 \quad \psi_1(P_2) = 0 \quad \psi_1(P_3) = 0$$
$$\psi_2(P_1) = 0 \quad \psi_2(P_2) = 1 \quad \psi_2(P_3) = 0$$
$$\psi_3(P_1) = 0 \quad \psi_3(P_2) = 0 \quad \psi_3(P_3) = 1$$

We define a function $F(\vec{U})$, a.k.a., *pushforward*, that maps reference 2D triangle to "physical" 2D triangle on the mesh (the inverse $F^{-1}(\vec{X})$, a.k.a., *pullback*, maps triangle from physical to reference frame).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Which cab be expressed in vector form

$$F(\vec{U}) = B\vec{U} + \vec{X}_1$$
$$F^{-1}(\vec{X}) = B^{-1}(\vec{X} - \vec{X}_1)$$

We denote ψ_i^k the function ψ_i defined on physical triangle K.

$$\psi_i^k(\vec{X}) = \psi_i \circ F^{-1}(\vec{X})$$
$$\psi_i^k(\vec{X}) = \psi_i(F^{-1}(\vec{X}))$$

The gradient of $\psi_i^k(\vec{X})$ in physical triangle is:

$$\begin{aligned}\nabla_{\vec{X}}\psi_i^k(\vec{X}) &= \nabla_{\vec{X}}\psi_i(F^{-1}(\vec{X})) \\ &= \nabla_{\vec{U}}\psi_i(\vec{U}) \cdot \nabla_{\vec{X}}F^{-1}(\vec{X}) \\ &= B^{-T}\nabla_{\vec{U}}\psi_i(\vec{U})\end{aligned}$$

where we used $\vec{U} = F^{-1}(\vec{X})$ and $\nabla_{\vec{X}}F^{-1}(\vec{X}) = B^{-T}$.

The gradient of $\psi_i(\vec{U})$ in reference triangle is:

$$\begin{aligned}\nabla\psi_1 &= (-1,-1) \\ \nabla\psi_2 &= (1,0) \\ \nabla\psi_3 &= (0,1)\end{aligned}$$

So the gradient of $\psi_i^k(\vec{X})$ in physical triangle k is "constant":

$$\begin{aligned}\nabla\psi_1^k &= B^{-T}(-1,-1) \\ \nabla\psi_2^k &= B^{-T}(1,0) \\ \nabla\psi_3^k &= B^{-T}(0,1)\end{aligned}$$

5.1. For 3D Triangles

For 3D triangles the map function F changes as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

In matrix algebra it can be expressed as:

$$\begin{aligned}F(\vec{U}) &= B\vec{U} + \vec{X}_1 \\ F^{-1}(\vec{X}) &= (B^TB)^{-1}B^T(\vec{X} - \vec{X}_1)\end{aligned}$$

or

$$F^{-1}(\vec{X}) = B^{-*}(\vec{X} - \vec{X}_1)$$

where $B^{-*} = (B^TB)^{-1}B^T$ is the Moore-Penrose pseudo-inverse of B.

The same analysis as before holds so the gradient in "physical" trignale k is:

$$\begin{aligned}\nabla\psi_1^k &= B^{-*T}(-1,-1) \\ \nabla\psi_2^k &= B^{-*T}(1,0) \\ \nabla\psi_3^k &= B^{-*T}(0,1)\end{aligned}$$

6. Numerical Quadrature

The entries of the matrix M_T^k are the following double integrals:

$$M_T^k(i,j) = \int_{T^k} \nabla\psi_i^k(\vec{X}) \cdot \nabla\psi_j^k(\vec{X})\ dA$$

inidices (i,j) run for all nodes of triangle T.

Let define the inner product of basis functions as

$$I_{ij}^k(\vec{X}) = \nabla\psi_i^k(\vec{X}) \cdot \nabla\psi_j^k(\vec{X})$$

The 3-point Gauss quadrature rule for the linear triangle is:

$$M_T^k(i,j) = \frac{area(T^k)}{3}(I_{ij}^k(P_1) + I_{ij}^k(P_2) + I_{ij}^k(P_3)) \tag{6.1}$$

evaluated at the vertices of the triangle P_1, P_2 and P_3. Which is an exact quadrature formula for linear function on a triangle. However since $\nabla\psi_i^k(\vec{X})$ does not depends on $\vec{X}$ the quadrature expression reduces to:

$$M_T^k(i,j) = area(T^k)\nabla\psi_i^k(\vec{X}) \cdot \nabla\psi_j^k(\vec{X})$$

7. Triangle's Grandient

Recall the discretised function $U(\vec{X},t)$ is the finite sum:

$$U(\vec{X},t) = \sum_j^N U_t^j \psi_j(\vec{X})$$

The grandient in the reference triangle is:

$$\nabla U(\vec{X}) = \nabla \sum_j^N U^j \psi_j(\vec{X})$$

$$\nabla U(\vec{X}) = \sum_j^N U^j \nabla\psi_j(\vec{X})$$

So

$$\nabla\psi_1(\vec{U}) = (-1,-1)\nabla\psi_2(\vec{U}) = (1,0)\nabla\psi_3(\vec{U}) = (0,1) \tag{7.1}$$

So

$$\nabla U(\vec{X}) = U^1(-1,-1) + U^2(1,0) + U^3(0,1)\nabla U(\vec{X}) = (U^2 - U^1, U^3 - U^1) \tag{7.2}$$

Gradient in the "physical" triangle k is:

$$\nabla U(\vec{X}) = \sum_{j}^{N} U^j \nabla \psi_j^k(\vec{X})$$

$$\nabla U(\vec{X}) = U^1 B^{-*T}(-1,-1) + U^2 B^{-*T}(1,0) + U^3 B^{-*T}(0,1)$$

$$\nabla U(\vec{X}) = B^{-*T}(U^2 - U^1, U^3 - U^1)$$

8. Divergence of Triangle's Grandient

The divergence operator acts on a vector field-valued function $A(\vec{X})$. Where $\vec{X}$ is a point on the surface. Therefore $A(\vec{X})$ is a vector field. For 2D case it should be $A(x, y)$.

In FEM the vector-valued function $A(\vec{X})$ can be defined on triangles, nodes or edges. According to some books it is best to define divergence as constant on nodes and vector field-valued function $A(\vec{X})$ be constant on triangles

8.1. Weak Formulation of the Integrated Divergence

In order to discretize the divergence on each vertex we multiply it with a scalar-valued "test function" $\vartheta(\vec{X})$ of compact support so:

$$\nabla \cdot A_i = \int_{\Omega_i} \vartheta \nabla \cdot A \; dA$$

Recall the vector calculus identity, obtained by "product rule":

$$\nabla \cdot (\vartheta A) = \nabla \vartheta \cdot A + \vartheta \nabla \cdot A$$

So

$$\vartheta \nabla \cdot A = \nabla \cdot (\vartheta A) - \nabla \vartheta \cdot A$$

So

$$\int_{\Omega} \vartheta \nabla \cdot A \; dA = \int_{\Omega} \nabla \cdot (\vartheta A) \; dA - \int_{\Omega} \nabla \vartheta \cdot A \; dA$$

Recall Divergence Theorem:

$$\int_{\Omega} \nabla \cdot (\vartheta A) \; dA = \int_{\partial\Omega} \vartheta (A \cdot n) \; dS$$

By "divergence theorem"

$$\int_{\Omega} \vartheta \nabla \cdot A \; dA = \int_{\partial\Omega} \vartheta (A \cdot n) \; dS - \int_{\Omega} \nabla \vartheta \cdot A \; dA$$

If we define $\vartheta(\vec{X})$ to be zero at the boundary $\partial\Omega$ then we have:

$$\int_{\Omega} \vartheta \nabla \cdot A \; dA = - \int_{\Omega} \nabla \vartheta \cdot A \; dA$$

Plugging that to original equation we get:

$$\nabla \cdot A_i = -\int_{\Omega_i} \nabla\vartheta \cdot A \; dA$$

Which can be discretized as usual by definig $\nabla\vartheta(\vec{X}) = \sum_i \nabla\varphi_i(\vec{X})$ and $A(\vec{X}) = \bigcup_k A_k(\vec{X})$ for all triangles k:

$$\nabla \cdot A_i = -\int_{\Omega} \sum_i \nabla\varphi_i \cdot A_k \; dA$$

$$\nabla \cdot A_i = \sum_i -\int_{\Omega} \nabla\varphi_i \cdot A_k \; dA$$

The integral should be evaluated at triangle nodes, so quadrature rule reduces to:

$$\int_{\Omega} \nabla\varphi_i \cdot A_k \; dA = area_k(B^{-*T}(-1,-1) + B^{-*T}(1,0) + B^{-*T}(0,1)) \cdot A_k$$

9. Implementation

Implementation can be found in [2]

References

[1] H. P. Langtangen. Introduction to finite element methods. 2014.

[2] M. C. Lopez Belon. Project on github. Available at https://github.com/mauriciocele/basic-fem-heat-2d, 2024.

[3] F.-J. Sayas. A gentle introduction to the finite element method. 2015.

Mauricio Cele Lopez Belon
Madrid, Spain
e-mail: mclopez@outlook.com