

# Simple finite element methods in Python

Roland Becker

March 22, 2021

## Contents

<b>1</b>	<b>Heat equation</b>	<b>2</b>
1.1	Computation of the matrices for $\mathcal{P}_h^1(\Omega)$	2
<b>A</b>	<b>Python implementation</b>	<b>4</b>
<b>B</b>	<b>Finite elements on simplices</b>	<b>5</b>
B.1	Simplices	5
B.2	Barycentric coordinates	5
B.3	Finite elements	6
B.3.1	$\mathcal{P}_h^1(\Omega)$	6
B.3.2	$\mathcal{CR}_h^1(\Omega)$	6
B.3.3	$\mathcal{RT}_h^0(\Omega)$	7
<b>C</b>	<b>Discretization of the transport equation</b>	<b>8</b>
C.1	$\mathcal{P}_h^1(\Omega)$	8

# 1 Heat equation

Let  $\Omega \subset \mathbb{R}^d$ ,  $d = 2, 3$  be the computational domain. We suppose to have a disjointed partition of its boundary:  $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ . We consider the parabolic equation for the temperature  $T$ , heat flux  $\vec{q}$  and heat release  $\dot{q}$

Heat equation (strong formulation)

$$\left\{ \begin{array}{ll} \vec{q} = -k\nabla T \\ \rho C_p \frac{\partial T}{\partial t} + \operatorname{div}(\vec{v}T) + \operatorname{div} \vec{q} = \dot{q} & \text{in } \Omega \\ T = T^D & \text{in } \Gamma_D \\ k \frac{\partial T}{\partial n} = q^N & \text{in } \Gamma_N \\ c_R T + k \frac{\partial T}{\partial n} = q^R & \text{in } \Gamma_R \end{array} \right. \quad (1.1)$$

Heat equation (weak formulation)

Let  $H_f^1 := \{u \in H^1(\Omega) \mid T|_{\Gamma_D} = f\}$ . The standard weak formulation looks for  $T \in H_{T^D}^1$  such that for all  $\phi \in H_0^1(\Omega)$

$$\int_{\Omega} \rho C_p \frac{\partial T}{\partial t} \phi - \int_{\Omega} \vec{v}T \cdot \nabla \phi + \int_{\Omega} k \nabla T \cdot \nabla \phi + \int_{\Gamma_R} c_R T \phi + \int_{\Gamma_N \cup \Gamma_R} \vec{v}_n T \phi = \int_{\Omega} \dot{q} \phi + \int_{\Gamma_R} q^R \phi \quad (1.2)$$

We can derive (1.2) from (1.1) by the divergence theorem

$$\int_{\Omega} \operatorname{div} \vec{F} = \int_{\partial\Omega} \vec{F}_n \quad \xLeftrightarrow{F \rightarrow F\phi} \quad \int_{\Omega} (\operatorname{div} \vec{F}) \phi = - \int_{\Omega} \vec{F} \cdot \nabla \phi + \int_{\partial\Omega} \vec{F}_n \phi,$$

which gives with  $\vec{F} = \vec{v} + \vec{q}$

$$\int_{\Omega} \operatorname{div}(\vec{v} + \vec{q}) \phi = - \int_{\Omega} \vec{v} \cdot \nabla \phi + \int_{\Omega} k \nabla T \cdot \nabla \phi + \int_{\partial\Omega} \vec{F}_n \phi.$$

Using that  $\phi$  vanishes on  $\Gamma_D$  we have

$$\int_{\partial\Omega} \vec{F}_n \phi = \int_{\Gamma_N \cup \Gamma_R} \vec{F}_n \phi = \int_{\Gamma_N \cup \Gamma_R} \vec{v}_n \phi + \int_{\Gamma_N \cup \Gamma_R} \vec{q}_n \phi,$$

and then with the different boundary conditions, we find

$$\int_{\Gamma_N \cup \Gamma_R} \vec{q}_n \phi = \int_{\Gamma_D} q^N \phi + \int_{\Gamma_R} (q^R - c_R T) \phi$$

## 1.1 Computation of the matrices for $\mathcal{P}_h^1(\Omega)$

For the convection, we suppose that  $\vec{v} \in \mathcal{RT}_h^0(\Omega)$  and let for given  $K \in \mathcal{K}_h$   $\vec{v} = \sum_{k=1}^{d+1} v_k \Phi_k$ . Using

$$x_k = x_{S_k}^K, \quad h_k = h_{S_k}^K, \quad \sigma_k = \sigma_{S_k}^K, \quad n_k = n_{S_k}$$

we compute

$$\int_K \lambda_j \vec{v} \cdot \nabla \lambda_i = \sum_{k=1}^{d+1} v_k \int_K \lambda_j \Phi_k \cdot \nabla \lambda_i$$

$$\int_K \lambda_j \Phi_k \cdot \nabla \lambda_i = -\frac{\sigma_k \sigma_i}{h_k h_i} \int_K \lambda_j (x - x_k) \cdot n_i = -\frac{\sigma_k \sigma_i}{h_k h_i} \sum_{l=1}^{d+1} (x_l - x_k) \cdot n_i \int_K \lambda_j \lambda_l$$

## A Python implementation

We suppose to have a `class SimplexMesh` containing the following elements

```
class SimplexMesh():
    dimension, nnodes, ncells, nfaces
    simplices # np.array((ncells, dimension+1))
    faces     # np.array((nfaces, dimension))
    points, pointsc, pointsf # np.array((nnodes,3)), np.array((ncells,3)
    ), np.array((nfaces,3))
    normals, sigma # np.array((nfaces,dimension)), np.array((ncells,
    dimension+1))
    dV # np.array((ncells))
    bdrylabels # dictionary(keys: colors, values: id's of boundary
    faces)
```

The norm of the 'normals'  $\widetilde{\vec{n}}$  is the measure of of the face

$$\widetilde{\vec{n}}_i = |S_i| \vec{n}_i$$

## References Section A

- [0] M. A. Eisenberg and L. E. Malvern. "On finite element integration in natural co-ordinates". In: *Int. J. of Numer. Meth. in Engrg.* 7 (1973), pp. 574–575.
- [0] F. J. Vermolen and A. Segal. "On an integration rule for products of barycentric coordinates over simplexes in  $\mathbb{R}^n$ ". In: *J. Comput. Appl. Math.* 330 (2018), pp. 289–294.

## B Finite elements on simplices

### B.1 Simplices

We consider an arbitrary non-degenerate simplex  $K = (x_0, x_1, \dots, x_d)$ . The volume of  $K$  is given by

$$|K| = \frac{1}{d!} \det(x_1 - x_0, \dots, x_d - x_0) = \frac{1}{d!} \det(1, x_0, x_1, \dots, x_d) \quad 1 = (1, \dots, 1)^T. \quad (\text{B.1})$$

The  $d+1$  sides  $S_k$  (co-dimension one,  $d-1$ -simplices or facets) are defined by  $S_k = (x_0, \dots, x_k, \dots, x_d)$ . The height is  $h_k = |P_{S_k} x_k - x_k|$ , where  $P_S$  is the orthogonal projection on the hyperplane associated to  $S_k$ . We have  $P_{S_k} x_k = x_k + h_k \vec{n}_k$  and  $S_k = \{x \in \mathbb{R}^d \mid \vec{n}_k^T x = h_k\}$  and

$$\begin{aligned} 0 &= \int_K \operatorname{div}(\vec{c}) = \sum_{i=0}^d \int_{S_i} \vec{c} \cdot \vec{n}_i = \vec{c} \cdot \sum_{i=0}^d |S_i| \vec{n}_i \Rightarrow \sum_{i=0}^d |S_i| \vec{n}_i = 0 \\ d|K| &= \int_K \operatorname{div}(x) = \sum_{i=0}^d \int_{S_i} x \cdot \vec{n}_i = \sum_{i=0}^d |S_i| h_i \end{aligned}$$

Height formula

$$h_k = d \frac{|K|}{|S_k|}$$

### B.2 Barycentric coordinates

The barycentric coordinate of a point  $x \in \mathbb{R}^d$  give the coefficients in the affine combination of  $x = \sum_{i=0}^d \lambda_i x_i$  ( $\sum_{i=0}^d \lambda_i = 1$ ) and can be expressed by means of the outer unit normal  $\vec{n}_i$  of  $S_i$  or the signed distance  $d^s$  as

$$\lambda_i(x) = \frac{\vec{n}_i^T (x_j - x)}{\vec{n}_i^T (x_j - x_i)} \quad (j \neq i), \quad \lambda_i(x) = \frac{d^s(x, H)}{h_i}. \quad (\text{B.2})$$

Any polynomial in the barycentric coordinates can be integrated exactly. For  $\alpha \in \mathbb{N}_0^{d+1}$  we let  $\alpha! = \prod_{i=0}^d \alpha_i!$ ,  $|\alpha| = \sum_{i=0}^d \alpha_i$ , and  $\lambda^\alpha = \prod_{i=0}^d \lambda_i^{\alpha_i}$

Integration on  $K$

$$\int_K \lambda^\alpha = |K| \frac{d! \alpha!}{(|\alpha| + d)!} \quad (\text{B.3})$$

see [EisenbergMalvern73], [VermolenSegal18].

Gradient of  $\lambda_i$

$$\nabla \lambda_i = -\frac{1}{h_i} \vec{n}_i.$$

<sup>1</sup><https://en.wikipedia.org/wiki/Simplex#Volume>

### B.3 Finite elements

We consider a family  $\mathcal{H}$  of regular simplicial meshes  $h$  on a polyhedral domain  $\Omega \subset \mathbb{R}^d$ . The set of simplices of  $h \in \mathcal{H}$  is denoted by  $\mathcal{K}_h$ , and its  $d - 1$ -dimensional sides by  $\mathcal{S}_h$ , divided into interior and boundary sides  $\mathcal{S}_h^{\text{int}}$  and  $\mathcal{S}_h^{\partial}$ , respectively. The set of  $d + 1$  sides of  $K \in \mathcal{K}_h$  is  $\mathcal{S}_h(K)$ . To any side  $S \in \mathcal{S}_h$  we associate a unit normal vector  $n_S$ , which coincides with the unit outward normal vector  $n_{\partial\Omega}$  if  $S \in \mathcal{S}_h^{\partial}$ .

For  $K \in \mathcal{K}_h$  and  $S \in \mathcal{S}_h$ , or  $S \in \mathcal{S}_h(K)$  we denote

$$\begin{aligned} x_K &: \text{barycenter of } K & x_S &: \text{barycenter of } S \\ x_S^K &: \text{vertex opposite to } S \text{ in } K & h_S^K &: \text{distance of } x_S^K \text{ to } S \\ \sigma_S^K &:= \begin{cases} +1 & \text{if } n_S = n_K, \\ -1 & \text{if } n_S = -n_K. \end{cases} & \lambda_S^K &: \text{barycentric coordinates of } K \end{aligned}$$

For  $k \in \mathbb{N}_0$  we denote by  $\mathcal{C}_h^k(\Omega)$  the space of piecewise  $k$ -times differential functions with respect to  $\mathcal{K}_h$ . The subspace of piecewise polynomial functions of order  $k \in \mathbb{N}_0$  in  $\mathcal{C}_h^k(\Omega)$  is denoted by  $\mathcal{D}_h^k(\Omega)$  and the  $L^2(\Omega)$ -projection by  $\pi_h^k: L^2(\Omega) \rightarrow \mathcal{D}_h^k(\Omega)$ .

#### B.3.1 $\mathcal{P}_h^1(\Omega)$

We have  $\mathcal{P}_h^1(\Omega) = \mathcal{D}_h^1(\Omega) \cap C(\overline{\Omega})$ , but the FEM definition also provides a basis. The restrictions of the basis functions of  $\mathcal{P}_h^1(\Omega)$  to the simplex  $K$  are the barycentric coordinates  $\lambda_S^K$  associated to the node opposite to  $S$  in  $K$ .

Formulae for  $\mathcal{P}_h^1(\Omega)$

$$\nabla \lambda_S^K = -\frac{\sigma_S^K}{h_S^K} n_S, \quad \frac{1}{|K|} \int_K \lambda_S^K = \frac{1}{d+1}. \quad (\text{B.4})$$

For the computation of matrices we use (B.3), for example for  $i, j \in \llbracket 0, d \rrbracket$

$$\int_K \lambda_i \lambda_j = |K| \frac{d! \alpha!}{(|\alpha| + d)!} \quad \text{with} \quad \begin{cases} \alpha = (1, 1, 0, \dots, 0) & (i \neq j) \\ \alpha = (2, 0, \dots, 0) & (i = j) \end{cases}$$

so

$$\int_K \lambda_i \lambda_j = \frac{|K|}{(d+2)(d+1)} (1 + \delta_{ij}) \quad (\text{B.5})$$

More generally, we have for  $i_l \in \llbracket 0, d \rrbracket$  with  $1 \leq l \leq k$

$$\int_K \lambda_{i_1} \cdots \lambda_{i_k} = \frac{|K| \alpha!}{(d+k) \cdots (d+1)}, \quad \alpha_l = \# \{j \in \llbracket 0, d \rrbracket \mid i_j = l\}, \quad 1 \leq l \leq k. \quad (\text{B.6})$$

#### B.3.2 $\mathcal{CR}_h^1(\Omega)$

$$\mathcal{CR}_h^k(\Omega) := \left\{ q \in \mathcal{D}_h^k(\Omega) \mid \int_S [q] p = 0 \quad \forall S \in \mathcal{S}_h^{\text{int}}, \forall p \in P^{k-1}(S) \right\}. \quad (\text{B.7})$$

Denote in addition the basis of  $\mathcal{CR}_h^1(\Omega)$  by  $\psi_S$ , we have

Formulae for  $\mathcal{CR}_h^1$

$$\psi_{S|_K} = 1 - d \lambda_S^K, \quad \nabla \psi_{S|_K} = \frac{|S| \sigma_S^K}{|K|} n_S, \quad \frac{1}{|K|} \int_K \psi_S = \frac{1}{d+1}. \quad (\text{B.8})$$

### B.3.3 $\mathcal{RT}_h^0(\Omega)$

The Raviart-Thomas space for  $k \geq 0$  is given by

$$\mathcal{RT}_h^k(\Omega) := \left\{ \mathbf{v} \in \mathbf{D}_h^k(\Omega, \mathbb{R}^d) \oplus X_h^k \mid \int_S [\mathbf{v}_n] \mathbf{p} = 0 \ \forall S \in \mathcal{S}_h^{\text{int}}, \forall \mathbf{p} \in P^k(S) \right\} \quad (\text{B.9})$$

where  $X_h^k := \{ \mathbf{x} \mathbf{p} \mid \mathbf{p}|_K \in P_{\text{hom}}^k(K) \ \forall K \in \mathcal{K}_h \}$  with  $P_{\text{hom}}^k(K)$  the space of  $k$ -th order homogenous polynomials.

Then the Raviart-Thomas basis function of lowest order is given by

Formulae for  $\mathcal{RT}^0$

$$\Phi_{S|_K} := \sigma_S^K \frac{\mathbf{x} - \mathbf{x}_S^K}{h_S^K}, \quad \int_K \text{div } \Phi_{S|_K} = \sigma_S^K \frac{d|K|}{h_S^K} = \sigma_S^K |S|, \quad \frac{1}{|K|} \int_K \Phi_S = \sigma_S^K \frac{\mathbf{x}_K - \mathbf{x}_S^K}{h_S^K}. \quad (\text{B.10})$$

For the [python](#) implementation of the projection on  $\mathcal{D}_h^0(\Omega, \mathbb{R}^d)$  we have with the height formula

$$\pi_h(\vec{v})|_K = \sum_{i=1}^d v_i \frac{1}{|K|} \int_K \Phi_i(\mathbf{x}) = \sum_{i=1}^d v_i \sigma_i^K (\mathbf{x}_K - \mathbf{x}_{S_i}) \frac{|S_i|}{d|K|}$$

The [python](#) implementation reads

## C Discreization of the transport equation

For  $k \in \mathbb{N}_0$  we denote by  $\mathcal{C}_h^k(\Omega)$  the space of piecewise  $k$ -times differential functions with respect to  $\mathcal{K}_h$ , and piecewise differential operators  $\nabla_h : \mathcal{C}_h^l(\Omega) \rightarrow \mathcal{C}_h^{l-1}(\Omega, \mathbb{R}^d)$  ( $l \in \mathbb{N}$ ) by  $\nabla_h q|_K := \nabla(q|_K)$  for  $q \in \mathcal{C}_h^l(\Omega)$  and similarly for  $\text{div}_h : \mathcal{C}_h^l(\Omega, \mathbb{R}^d) \rightarrow \mathcal{C}_h^{l-1}(\Omega)$ . We frequently use the piecewise Stokes formula

$$\int_{\Omega} (\nabla_h q) v + \int_{\Omega} q (\text{div}_h v) = \int_{\mathcal{S}_h^{\text{int}}} [q v_n] + \int_{\mathcal{S}_h^{\partial}} q v_n, \quad (\text{C.1})$$

where  $\int_{\mathcal{S}_h} = \sum_{S \in \mathcal{S}_h} \int_S$  and  $n$  in the sum stands for  $n_S$ .

The subspace of piecewise polynomial functions of order  $k \in \mathbb{N}_0$  in  $\mathcal{C}_h^k(\Omega)$  is denoted by  $\mathcal{D}_h^k(\Omega)$  and the  $L^2(\Omega)$ -projection by  $\pi_h^k : L^2(\Omega) \rightarrow \mathcal{D}_h^k(\Omega)$ .

Suppose  $u$  satisfies

$$\text{div}(\beta u) = f \quad \text{in } \Omega, \quad \beta_n^-(u - u^D) = 0 \quad \text{on } \partial\Omega. \quad (\text{C.2})$$

From the integration by parts formula

$$\int_{\Omega} \text{div}(\beta u) v = - \int_{\Omega} \beta u \cdot \nabla v + \int_{\partial\Omega} \beta_n u v \quad (\text{C.3})$$

it then follows that  $u$  satisfies

$$a(u, v) = l(v) \quad \forall v \in V$$

with

$$a(u, v) := - \int_{\Omega} u \beta \cdot \nabla v + \int_{\partial\Omega} \beta_n^+ u v, \quad l(v) := \int_{\Omega} f v - \int_{\partial\Omega} \beta_n^- u^D v. \quad (\text{C.4})$$

We also have

$$\begin{aligned} a(u, v) &= - \frac{1}{2} \int_{\Omega} u \beta \cdot \nabla v + \frac{1}{2} \int_{\Omega} \text{div}(\beta u) v - \frac{1}{2} \int_{\partial\Omega} \beta_n u v + \int_{\partial\Omega} \beta_n^+ u v \\ &= \frac{1}{2} \int_{\Omega} \text{div}(\beta) u v + \frac{1}{2} \left( \int_{\Omega} \beta \cdot \nabla u v - \int_{\Omega} u \beta \cdot \nabla v \right) + \frac{1}{2} \int_{\partial\Omega} |\beta_n| u v \end{aligned}$$

### C.1 $\mathcal{P}_h^1(\Omega)$

Let  $K \in \mathcal{K}_h$ ,  $\beta_K = \pi_K \beta$ ,  $x_K$  be the barycenter of  $K$  and  $x_K^{\sharp} \in \partial K$  such that with  $\delta_K \geq 0$

$$x_K^{\sharp} = x_K + \delta_K \beta_K \quad (\text{C.5})$$

If we know  $\tilde{n}_i^T \beta_K$ , we can compute  $x_K^{\sharp}$  as follows.

$$\lambda_i(x_K^{\sharp}) = \lambda_i(x_K) + \delta_K \nabla \lambda_i^T \beta_K = \frac{1}{d+1} - \delta_K \frac{\tilde{n}_i^T \beta_K}{h_i} = \frac{1}{d+1} - \delta_K \frac{\tilde{n}_i^T \beta_K |S_i|}{d|K|}$$

It follows that

$$\delta_K = \max \left\{ \frac{d|K|}{(d+1)|S_i|(\tilde{n}_i^T \beta_K)^+} \mid 0 \leq i \leq d \right\}. \quad (\text{C.6})$$

The stabilized bilinear form is

$$a^{\text{supg}}(u, v) := \int_{\Omega} (\beta \cdot \nabla u) v - \int_{\partial\Omega} \beta_n^- u v + \int_{\Omega} \delta(\beta \cdot \nabla u)(\beta \cdot \nabla v) \quad (\text{C.7})$$

Then we have

$$a^{\text{supg}}(u, v) =$$