

NAPDE project outline

Matteo Calafà and Federica Botta

June 3, 2021

Contents

1	Monodomain	2
1.1	Analytical models	2
1.2	Semi-discretized numerical method	2
1.3	Totally-discretized numerical method (" θ – <i>method</i> ")	3
2	Bidomain	4
2.1	Analytical models	4
2.2	Semi-discretized numerical methods	4
2.3	Totally-discretized numerical methods	5
2.3.1	Semi-implicit method	5
2.3.2	Quasi-implicit Operator Splitting	6
2.3.3	Godunov Operator Splitting	7
3	Dubiner basis	8
3.1	Basis functions	8
3.1.1	Mapping transformation	8
3.1.2	Jacobian polynomials	8
3.1.3	Dubiner Basis	9
3.1.4	Gradient of Dubiner Basis	9
3.2	Transformation from FEM to Dubiner basis	9
4	Few considerations about well-posedness	11

1 Monodomain

1.1 Analytical models

Monodomain model

$$\begin{cases} \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma \nabla V_m) + \chi_m I_{ion}(V_m, w) = I^{ext} & \text{in } \Omega_{mus} \times (0, T] \\ \frac{\partial w}{\partial t} = g(V_m, w) & \text{in } \Omega_{mus} \times (0, T] \\ \Sigma \nabla V_m \cdot n = b & \text{on } \partial\Omega_{mus} \times (0, T] \end{cases} \quad (1)$$

where the unknowns are:

- $V_m = \Phi_i - \Phi_e$ (difference between internal and external potential)
- w ("gating variable")

and these constants are given : χ_m, C_m, Σ

FitzHugh-Nagumo model

$$\begin{aligned} I_{ion}(V_m, w) &= -kV_m(V_m - a)(V_m - 1) - w \\ g(V_m, w) &= \epsilon(V_m - \gamma w) \end{aligned} \quad (2)$$

1.2 Semi-discretized numerical method

$$\left. \begin{aligned} V_{ij} &= \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \\ I_{i,j}^T &= \sum_{F \in F_h^I} \int_F \{ \{ \nabla \varphi_j \} \} \cdot [[\varphi_i]] \\ I_{i,j} &= \sum_{F \in F_h^I} \int_F [[\varphi_j]] \cdot \{ \{ \nabla \varphi_i \} \} \\ S_{i,j} &= \sum_{F \in F_h^I} \int_F \gamma [[\varphi_j]] \cdot [[\varphi_i]] \end{aligned} \right\} \quad A = \Sigma(V - I^T - \delta I + S) \quad (3)$$

$$M_{ij} = \sum_{K \in \tau_h} \int_K \varphi_j \varphi_i \quad (4)$$

$$C(u_h)_{ij} = - \sum_{K \in \tau_h} \int_K \chi_m k(u_h - 1)(u_h - a) \varphi_j \varphi_i \quad (5)$$

$$F_i = \int_{\Omega} f \varphi_i - \sum_{F \in F_h^B} \int_F b \varphi_i \quad (6)$$

Semi-discretized problem

$$\{\varphi_j\}_{j=1}^{N_h} \text{ base di } V_h^p = \{v_h \in L^2 : v_h|_K \in \mathbb{P}^{p_k}(K) \quad p_k \leq p \quad \forall K \in \tau_h\}$$

$$u_h(t) = \sum_{j=1}^{N_h} u_j(t) \varphi_j, \quad w_h(t) = \sum_{j=1}^{N_h} w_j(t) \varphi_j$$

$$\Rightarrow \begin{array}{c} \boxed{\chi_m C_m M \dot{u} + Au + C(u_h)u - \chi_m M w = F} \\ \boxed{\dot{w} = \epsilon(u - \gamma w)} \end{array} \quad (7)$$

1.3 Totally-discretized numerical method (" θ – method")

Initial form ($\theta \in [0, 1]$)

1.

$$\begin{aligned} \chi_m C_m M \frac{u^{k+1} - u^k}{\Delta t} + A(\theta u^{k+1} + (1 - \theta)u^k) + C(u^k)(\theta u^{k+1} + (1 - \theta)u^k) + \\ - \chi_m M w^{k+1} = \theta F^{k+1} + (1 - \theta)F^k \end{aligned} \quad (8)$$

2.

$$\frac{w^{k+1} - w^k}{\Delta t} = \epsilon(u^k - \gamma w^{k+1}) \quad (9)$$

Expanded form ($\theta \in [0, 1]$)

1.

$$\begin{aligned} [\chi_m C_m M + \theta \Delta t A + \theta \Delta t C(u^k)] \mathbf{u}^{k+1} = \theta \Delta t F^{k+1} + (1 - \theta) \Delta t F^k + \\ [\chi_m C_m M - (1 - \theta) \Delta t A - (1 - \theta) \Delta t C(u^k)] u^k + \chi_m \Delta t M w^{k+1} \end{aligned} \quad (10)$$

2.

$$[1 + \epsilon \gamma \Delta t] \mathbf{w}^{k+1} = w^k + (\epsilon \Delta t) u^k \quad (11)$$

2 Bidomain

2.1 Analytical models

Bidomain model

$$\begin{cases} \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \phi_i) + \chi_m I_{ion}(V_m, w) = I_i^{ext} & \text{in } \Omega_{mus} \times (0, T] \\ -\chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_e \nabla \phi_e) - \chi_m I_{ion}(V_m, w) = -I_e^{ext} & \text{in } \Omega_{mus} \times (0, T] \\ \frac{\partial w}{\partial t} = g(V_m, w) & \text{in } \Omega_{mus} \times (0, T] \\ \Sigma_i \nabla \phi_i \cdot n = b_i & \text{on } \partial\Omega_{mus} \times (0, T] \\ \Sigma_e \nabla \phi_e \cdot n = b_e & \text{on } \partial\Omega_{mus} \times (0, T] \end{cases} \quad (12)$$

where the unknowns are:

- ϕ_i, ϕ_e ($V_m = \phi_i - \phi_e$)
- w ("gating variable")

and these constants are given: $\chi_m, C_m, \Sigma_i, \Sigma_e$

FitzHugh-Nagumo model

$$\begin{aligned} I_{ion}(V_m, w) &= -kV_m(V_m - a)(V_m - 1) - w \\ g(V_m, w) &= \epsilon(V_m - \gamma w) \end{aligned} \quad (13)$$

2.2 Semi-discretized numerical methods

$$\left. \begin{aligned} V_{ij} &= \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \\ I_{i,j}^T &= \sum_{F \in F_h^I} \int_F \{ \{ \nabla \varphi_j \} \} \cdot [[\varphi_i]] \\ I_{i,j} &= \sum_{F \in F_h^I} \int_F [[\varphi_j]] \cdot \{ \{ \nabla \varphi_i \} \} \\ S_{i,j} &= \sum_{F \in F_h^I} \int_F \gamma [[\varphi_j]] \cdot [[\varphi_i]] \end{aligned} \right\} \quad \begin{aligned} A &= (V - I^T - \theta I + S) \\ A_i &= \Sigma_i A \\ A_e &= \Sigma_e A \end{aligned} \quad (14)$$

$$M_{ij} = \sum_{K \in \tau_h} \int_K \varphi_j \varphi_i \quad (15)$$

$$C(u_h)_{ij} = - \sum_{K \in \tau_h} \int_K \chi_m k (u_h - 1)(u_h - a) \varphi_j \varphi_i \quad (16)$$

$$\begin{aligned} F_{i,k} &= \int_{\Omega} I_i^{ext} \varphi_k - \sum_{F \in F_h^B} \int_F b_i \varphi_k \\ F_{e,k} &= - \int_{\Omega} I_e^{ext} \varphi_k - \sum_{F \in F_h^B} \int_F b_e \varphi_k \end{aligned} \quad (17)$$

Semi-discretized problem

$$\{\varphi_j\}_{j=1}^{N_h} \text{ base di } V_h^k = \{v_h \in L^2 : v_h|_{\mathcal{K}} \in \mathbb{P}^k(\mathcal{K}) \quad \forall \mathcal{K} \in \tau_h\}$$

$$\Phi_h(t) = \begin{bmatrix} \Phi_i^h(t) \\ \Phi_e^h(t) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N_h} \Phi_{i,j}(t) \varphi_j \\ \sum_{j=1}^{N_h} \Phi_{e,j}(t) \varphi_j \end{bmatrix}, \quad w_h(t) = \sum_{j=1}^{N_h} w_j(t) \varphi_j$$

$$\Rightarrow \begin{bmatrix} \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \dot{\Phi}_i^h(t) \\ \dot{\Phi}_e^h(t) \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \Phi_i^h(t) \\ \Phi_e^h(t) \end{bmatrix} + \\ \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \Phi_i^h(t) \\ \Phi_e^h(t) \end{bmatrix} - \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w_h(t) \\ w_h(t) \end{bmatrix} = \begin{bmatrix} F_i^h \\ F_e^h \end{bmatrix} \end{bmatrix} \quad (18)$$

$$\boxed{\dot{w}_h(t) = \epsilon(V_m^h(t) - \gamma w_h(t))} \quad (19)$$

2.3 Totally-discretized numerical methods

2.3.1 Semi-implicit method

Initial form

$$\begin{aligned} & \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \frac{\Phi_i^{k+1} - \Phi_i^k}{\Delta t} \\ \frac{\Phi_e^{k+1} - \Phi_e^k}{\Delta t} \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \Phi_i^{k+1} \\ \Phi_e^{k+1} \end{bmatrix} + \\ & \begin{bmatrix} C(V_m^k) & -C(V_m^k) \\ -C(V_m^k) & C(V_m^k) \end{bmatrix} \begin{bmatrix} \Phi_i^{k+1} \\ \Phi_e^{k+1} \end{bmatrix} - \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{k+1} \\ w^{k+1} \end{bmatrix} = \begin{bmatrix} F_i^{k+1} \\ F_e^{k+1} \end{bmatrix} \end{aligned} \quad (20)$$

$$\frac{w^{k+1} - w^k}{\Delta t} = \epsilon(V_m^k - \gamma w^{k+1}) \quad (21)$$

Expanded form

$$\begin{aligned} & \left(\frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} + \begin{bmatrix} C(V_m^k) & -C(V_m^k) \\ -C(V_m^k) & C(V_m^k) \end{bmatrix} \right) \begin{bmatrix} \Phi_i^{k+1} \\ \Phi_e^{k+1} \end{bmatrix} = \\ & \begin{bmatrix} F_i^{k+1} \\ F_e^{k+1} \end{bmatrix} + \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{k+1} \\ w^{k+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \Phi_i^k \\ \Phi_e^k \end{bmatrix} \end{aligned} \quad (22)$$

$$(1 + \epsilon \gamma \Delta t) w^{k+1} = w^k + \epsilon \Delta t V_m^k \quad (23)$$

2.3.2 Quasi-implicit Operator Splitting

Initial form

I

$$\begin{aligned} \chi_m C_m M \frac{\tilde{V}_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^{n+1} - \chi_m M w^{n+1} &= 0 \\ \frac{w^{n+1} - w^n}{\Delta t} &= \epsilon(V_m^{n+1} - \gamma w^{n+1}) \end{aligned} \quad (24)$$

II

$$\begin{aligned} \chi_m C_m M \frac{V_m^{n+1} - \tilde{V}_m^{n+1}}{\Delta t} + A_i \Phi_i^{n+1} &= F_i^{n+1} \\ -\chi_m C_m M \frac{V_m^{n+1} - \tilde{V}_m^{n+1}}{\Delta t} + A_e \Phi_e^{n+1} &= F_e^{n+1} \end{aligned} \quad (25)$$

Expanded form

$$\begin{cases} \chi_m C_m M \frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^{n+1} - \chi_m M w^{n+1} + A_i \Phi_i^{n+1} = F_i^{n+1} \\ \chi_m C_m M \frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^{n+1} - \chi_m M w^{n+1} - A_e \Phi_e^{n+1} = -F_e^{n+1} \\ \frac{w^{n+1} - w^n}{\Delta t} = \epsilon(V_m^{n+1} - \gamma w^{n+1}) \end{cases} \quad (26)$$

$$\begin{aligned} \bullet \quad Q_n &:= \frac{\chi_m C_m}{\Delta t} M + C(V_m^n) - \frac{\epsilon \chi_m \Delta t}{1 + \epsilon \gamma \Delta t} M \\ \bullet \quad R_n &:= \frac{\chi_m C_m}{\Delta t} M V_m^n + \frac{\chi_m}{1 + \epsilon \gamma \Delta t} M w^n \end{aligned} \quad (27)$$

1.

$$\begin{aligned} \chi_m C_m M \frac{\Phi_i^{n+1} - \Phi_e^{n+1} - V_m^n}{\Delta t} + C(V_m^n) (\Phi_i^{n+1} - \Phi_e^{n+1}) + \\ -\chi_m M \left(\frac{w^n + \epsilon \Delta t (\Phi_i^{n+1} - \Phi_e^{n+1})}{1 + \epsilon \gamma \Delta t} \right) + A_i \Phi_i^{n+1} &= F_i^{n+1} \end{aligned} \quad (28)$$

$$\Rightarrow (Q_n + A_i) \Phi_i^{n+1} - Q_n \Phi_e^{n+1} = R_n + F_i^{n+1}$$

2.

$$\begin{aligned} \chi_m C_m M \frac{\Phi_i^{n+1} - \Phi_e^{n+1} - V_m^n}{\Delta t} + C(V_m^n) (\Phi_i^{n+1} - \Phi_e^{n+1}) + \\ -\chi_m M \left(\frac{w^n + \epsilon \Delta t (\Phi_i^{n+1} - \Phi_e^{n+1})}{1 + \epsilon \gamma \Delta t} \right) - A_e \Phi_e^{n+1} &= -F_e^{n+1} \end{aligned} \quad (29)$$

$$\Rightarrow Q_n \Phi_i^{n+1} - (Q_n + A_e) \Phi_e^{n+1} = R_n - F_e^{n+1}$$

3.

$$w^{n+1} = \frac{w^n + \epsilon \Delta t (\Phi_i^{n+1} - \Phi_e^{n+1})}{1 + \epsilon \gamma \Delta t} \quad (30)$$

$$\Rightarrow \begin{cases} \left(\begin{bmatrix} Q_n & -Q_n \\ Q_n & -Q_n \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \Phi_i^{n+1} \\ \Phi_e^{n+1} \end{bmatrix} = \begin{bmatrix} R_n \\ R_n \end{bmatrix} + \begin{bmatrix} F_i^{n+1} \\ -F_e^{n+1} \end{bmatrix} \\ Q_n = \frac{\chi_m C_m}{\Delta t} M + C(V_m^n) - \frac{\epsilon \chi_m \Delta t}{1 + \epsilon \gamma \Delta t} M \\ R_n := \frac{\chi_m C_m}{\Delta t} M V_m^n + \frac{\chi_m}{1 + \epsilon \gamma \Delta t} M w^n \\ w^{n+1} = \frac{w^n + \epsilon \Delta t (\Phi_i^{n+1} - \Phi_e^{n+1})}{1 + \epsilon \gamma \Delta t} \end{cases} \quad (31)$$

2.3.3 Godunov Operator Splitting

Initial form

I

$$\begin{aligned} \chi_m C_m M \frac{\hat{V}_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^n - \chi_m M w^n &= 0 \\ \frac{w^{n+1} - w^n}{\Delta t} &= \epsilon (V_m^n - \gamma w^n) \end{aligned} \quad (32)$$

II

$$\begin{aligned} \chi_m C_m M \frac{V_m^{n+1} - \hat{V}_m^{n+1}}{\Delta t} + A_i \Phi_i^{n+1} &= F_i^{n+1} \\ -\chi_m C_m M \frac{V_m^{n+1} - \hat{V}_m^{n+1}}{\Delta t} + A_e \Phi_e^{n+1} &= F_e^{n+1} \end{aligned} \quad (33)$$

Expanded form

$$\begin{aligned} &\begin{cases} \chi_m C_m M \frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^n - \chi_m M w^n + A_i \Phi_i^{n+1} = F_i^{n+1} \\ \chi_m C_m M \frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n) V_m^n - \chi_m M w^n - A_e \Phi_e^{n+1} = -F_e^{n+1} \\ w^{n+1} = (1 - \epsilon \gamma \Delta t) w^n + \epsilon \Delta t V_m^n \end{cases} \\ \Rightarrow &\begin{cases} \left(\frac{\chi_m C_m}{\Delta t} M + A_i \right) \Phi_i^{n+1} - \frac{\chi_m C_m}{\Delta t} M \Phi_e^{n+1} = F_i^{n+1} + \chi_m M w^n + \left(\frac{\chi_m C_m}{\Delta t} M - C(V_m^n) \right) V_m^n \\ \frac{\chi_m C_m}{\Delta t} M \Phi_i^{n+1} - \left(\frac{\chi_m C_m}{\Delta t} M + A_e \right) \Phi_e^{n+1} = -F_e^{n+1} + \chi_m M w^n + \left(\frac{\chi_m C_m}{\Delta t} M - C(V_m^n) \right) V_m^n \\ w^{n+1} = (1 - \epsilon \gamma \Delta t) w^n + \epsilon \Delta t V_m^n \end{cases} \end{aligned} \quad (34)$$

$$\Rightarrow \begin{cases} \left(\frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ M & -M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \Phi_i^{n+1} \\ \Phi_e^{n+1} \end{bmatrix} = \begin{bmatrix} F_i^{n+1} \\ -F_e^{n+1} \end{bmatrix} + \\ \chi_m \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} w^n \\ w^n \end{bmatrix} + \left(\frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} C(V_m^n) & 0 \\ 0 & C(V_m^n) \end{bmatrix} \right) \begin{bmatrix} V_m^n \\ V_m^n \end{bmatrix} \\ w^{n+1} = (1 - \epsilon \gamma \Delta t) w^n + \epsilon \Delta t V_m^n \end{cases} \quad (35)$$

3 Dubiner basis

3.1 Basis functions

3.1.1 Mapping transformation

On the reference triangle

$$\hat{K} = \{(\xi, \eta) : \xi, \eta \geq 0, \xi + \eta \leq 1\} \quad (36)$$

we consider the transformation between the reference square and the reference triangle given by

$$\xi := \frac{(1+a)(1-b)}{4}, \eta := \frac{(1+b)}{2} \quad (37)$$

and the inverse transformation is

$$a := \frac{2\xi - 1 + \eta}{1 - \eta} = \frac{2\xi}{1 - \eta} - 1, b := 2\eta - 1 \quad (38)$$

3.1.2 Jacobian polynomials

Evaluation of the polynomial in $z \in \mathbb{R}^n$:

– $n = 0$

$$J_0^{\alpha, \beta}(z) = \overbrace{[1 \quad 1 \quad \dots \quad 1]}^{n \text{ times}} \quad (39)$$

– $n = 1$

$$J_1^{\alpha, \beta}(z) = \frac{1}{2}(\alpha - \beta + (\alpha + \beta + 2) \cdot z); \quad (40)$$

– $n \geq 2$

$$\begin{aligned} J_n^{\alpha, \beta}(z) = \sum_{k=2}^n & \left[\frac{(2k + \alpha + \beta - 1)(\alpha^2 - \beta^2)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} + \right. \\ & \frac{(2k + \alpha + \beta - 2)(2k + \alpha + \beta - 1)(2k + \alpha + \beta)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} J_{k-1}^{\alpha, \beta}(z) + \\ & \left. - \frac{2(k + \alpha - 1)(k + \beta - 1)(2k + \alpha + \beta)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} J_{k-2}^{\alpha, \beta}(z) \right] \end{aligned} \quad (41)$$

Proposition. $J_i^{\alpha, \beta}(\cdot)$ is orthogonal w.r.t. the Jacobi weight $w(x) = (1-x)^\alpha(1+x)^\beta$:

$$\int_{-1}^1 (1-x)^\alpha(1+x)^\beta J_m^{\alpha, \beta} J_q^{\alpha, \beta}(x) dx = \frac{2}{2m+1} \delta_{mq} \quad (42)$$

3.1.3 Dubiner Basis

$$\begin{aligned}\phi_{ij}(\xi, \eta) &:= c_{ij}(1-b)^j J_i^{0,0}(a) J_j^{2i+1,0}(b) = \\ &= c_{ij} 2^j (1-\eta)^j J_i^{0,0}\left(\frac{2\xi}{1-\eta} - 1\right) J_j^{2i+1,0}(2\eta - 1)\end{aligned}\quad (43)$$

for $i, j = 1, \dots, p$ and $i + j \leq p$, where

$$c_{ij} := \sqrt{\frac{2(2i+1)(i+j+1)}{4^i}} \quad (44)$$

and $J_i^{\alpha,\beta}(\cdot)$ is the i -th Jacobi polynomial

3.1.4 Gradient of Dubiner Basis

– $i = 0, j = 0$

$$\begin{aligned}\phi_{00}^\xi(\xi, \eta) &= 0 \\ \phi_{00}^\eta(\xi, \eta) &= 0\end{aligned}\quad (45)$$

– $i = 0, j \neq 0$

$$\begin{aligned}\phi_{0j}^\xi &= 0 \\ \phi_{0j}^\eta &= c_{0j}(j+2)J_{j-1}^{2,1}(b)\end{aligned}\quad (46)$$

– $i \neq 0, j = 0$

$$\begin{aligned}\phi_{i0}^\xi(\xi, \eta) &= c_{i0} 2^i (1-\eta)^{i-1} (i+1) J_{i-1}^{1,1}(a) \\ \phi_{i0}^\eta(\xi, \eta) &= c_{i0} 2^i (-i(1-\eta)^{i-1} J_i^{0,0}(a) + \xi(1-\eta)^{i-2} (i+1) J_{i-1}^{1,1}(a))\end{aligned}\quad (47)$$

– $i \neq 0, j \neq 0$

$$\begin{aligned}\phi_{ij}^\xi(\xi, \eta) &= c_{ij} 2^i (1-\eta)^{i-1} (i+1) J_{i-1}^{1,1}(a) J_j^{2i+1,0}(b) \\ \phi_{ij}^\eta(\xi, \eta) &= c_{ij} 2^i (-i(1-\eta)^{i-1} J_i^{0,0}(a) J_j^{2i+1,0}(b) + \xi(1-\eta)^{i-2} (i+1) J_{i-1}^{1,1}(a) J_j^{2i+1,0}(b) \\ &\quad + (1-\eta)^i (2i+j+2) J_i^{0,0}(a) J_{j-1}^{2i+2,1}(b))\end{aligned}\quad (48)$$

3.2 Transformation from FEM to Dubiner basis

One of the many advantages of the FEM basis is that the evaluation of a basis function in a point of the mesh is equal to 1 only if that point is the one associated to the basis, 0 otherwise:

$$\psi_i(x_j) = \delta_{ij} \quad (49)$$

This property cannot be satisfied by Dubiner basis (although other good properties hold in this case, for instance regularity and especially *orthogonality*). Indeed these basis have not localized support and they are neither normalized on the mesh edges. This means that the coefficients of the solution of the Dubiner system are *not* the evaluation over the mesh points of the discretized function itself. They have a completely different meaning, they are now *modal* values instead of being *nodal*. For this reason we introduced two

new functions that best transform the coefficients of the solution w.r.t. FEM basis to the coefficients w.r.t. Dubiner basis and viceversa.

Consider an element $\mathcal{K} \in \tau_h$ and $\{\psi_i\}_{i=1}^p, \{\phi_j\}_{j=1}^q$ as, respectively, the set of FEM functions and the set of Dubiner functions with support in \mathcal{K} . In addition, consider as $\{\hat{u}_i\}_{i=1}^p, \{\tilde{u}_j\}_{j=1}^q$ as, respectively, the FEM and Dubiner coefficients of the solution.

Let us start from the transformation to the FEM coefficients. We now exploit the property 49, i.e. the coefficient \hat{u}_i is nothing else but the evaluation of u_h on the i -th mesh point, then:

$$\hat{u}_i = \sum_{j=1}^q \tilde{u}_j \phi_j(x_i) \quad (50)$$

where x_i is the point associated to the ψ_i basis function.

Instead, to compute the coefficients conversely, we need to exploit the fact that the Dubiner Basis are L^2 -orthonormal (property obtained thanks to 42). We then need to compute a L^2 scalar product between the FEM discretized function and each Dubiner basis function. That means:

$$\tilde{u}_j = \int_{\mathcal{K}} u_h(x) \phi_j(x) dx = \int_{\mathcal{K}} \sum_{i=1}^p \hat{u}_i \psi_i(x) \phi_j(x) dx = \sum_{i=1}^p \left(\int_{\mathcal{K}} \psi_i(x) \phi_j(x) dx \right) \hat{u}_i \quad (51)$$

If the Dubiner functions are chosen as Galerkin basis, both the transformations are needed for the code implementation. Formula 50 is needed to plot and compute errors after the resolution of the system (otherwise solely Dubiner coefficients are useless). Formula 51 is instead needed to convert the FEM initial data u_0 into a vector of Dubiner coefficients before the resolution of the system.

For the sake of both simplicity and logic, we have decided to implement these transformations only from P_n to D_n , $n = 1, 2, 3$ and viceversa. Indeed, from one side, the degree of FEM is here less "important" since it contributes only to the number of points to which evaluate the computed solution. Then, increasing n for P does not substantially improve the quality of the solution. On the other side, choosing the same degree for P and D means having the same number of local nodes (nln). For this reason, both p and q are replaced with nln in the code.

Finally, these are the coordinates of the evaluation points over the reference square:

- $n=1$

$$a = \begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$$

- $n=2$

$$a = \begin{bmatrix} 1 & 0 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} -1 & -1 & -1 & 0 & 1 & 0 \end{bmatrix}$$

- $n=3$

$$a = \begin{bmatrix} -1 & -0.5 & 0.5 & 1 & 1 & 1 & -1 & -1 & -1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} -1 & -1 & -1 & -1 & -0.5 & 0.5 & 1 & 0.5 & -0.5 & -1/3 \end{bmatrix}$$

4 Few considerations about well-posedness

All the previous Bidomain schemes treat the non-linear term in a semi-implicit way (see 18). It means that at time t^{n+1} we can solve a linear problem if data at time t^n are provided as parameters. From another perspective, at each time step t^{n+1} , we can fix the data of the previous time-step and solve a linear variational problem. In this case, the bilinear form at the left-hand side is the sum of some simpler bilinear forms: the standard mass L^2 inner product for the time derivative, the standard stiffness bilinear form and a pseudo-mass bilinear form for the non-linear term.

$$a_c(u, v) = - \int_{\Omega} \chi_m k (V_m^n - a) (V_m^n - 1) uv \, dx \quad (52)$$

We want to check that this bilinear form is coercive w.r.t $L^2(\Omega)$ (without considering the others for now). That means to check that:

$$\exists \alpha > 0 : a_c(u, u) \geq \alpha \|u\|^2 \quad \forall u \in L^2(\Omega) \quad (53)$$

Then:

$$a_c(u, u) = \int_{\Omega} -\chi_m k (V_m^n - a) (V_m^n - 1) u^2 \, dx \geq \inf_{x \in \Omega} [-\chi_m k (V_m^n - a) (V_m^n - 1)] \|u\|_{L^2}^2 \quad (54)$$

Since all parameters are positive and $a \ll 1$, $\alpha = \inf_{x \in \Omega} [-\chi_m k (V_m^n - a) (V_m^n - 1)]$ is positive only if:

$$(V_m^n - 1)(V_m^n - a) < 0 \Leftrightarrow a < V_m^n < 1 \quad (55)$$

Finally, we can say that if the exact solution is between a and 1, a_c is coercive.

Obviously, this is only a sufficient condition for the total coercivity (remember that the other bilinear forms can contribute for the positiveness of $a(u, u)$). However, if this property fails, there may be bad consequences for the stability, especially in presence of physiological parameters. For the first simulations, indeed, we considered the exact solution:

$$V_m(x, t) = \sin(2\pi x) \sin(2\pi y) e^{-5t} \quad (56)$$

whose image is in $[-1, 1]$. Choosing physiological parameters, we obtained some instabilities. Translating the exact solution, in order to replace the image into $[0, 1]$ (remember that a is a small parameter), we found instead convergence:

$$V_m(x, t) = \frac{1}{2} \sin(2\pi x) \sin(2\pi y) e^{-5t} + \frac{1}{2} \quad (57)$$