Journal of Computational and Applied Mathematics 260 (2014) 135–148

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

hp-FEM electromechanical transduction model of ionic polymer–metal composites

D. Pugal [a,d,*], P. Solin [b,c], K.J. Kim [e], A. Aabloo [d,*]

[a] *Mechanical Engineering Department, University of Nevada, Reno, NV, USA*
[b] *FEM group, Department of Mathematics and Statistics, University of Nevada, Reno, NV, USA*
[c] *Institute of Thermomechanics, Prague, Czech Republic*
[d] *Institute of Technology, University of Tartu, Tartu, Estonia*
[e] *Department of Mechanical Engineering, University of Nevada, Las Vegas, NV, USA*

ARTICLE INFO

Article history:
Received 30 April 2013

Keywords:
hp-FEM
Nernst–Planck
Poisson
Weak form

ABSTRACT

We propose a mathematically explicit model of ionic polymer–metal composite (IPMC) materials that consist of a coupled system of Poisson–Nernst–Planck–Euler equations, discretized by means of adaptive higher-order finite element methods (*hp*-FEM). We show that due to the transient character of the problem and different fields in terms of the locations of field gradients in the domain it is efficient to use adaptive algorithms that are capable of changing the mesh dynamically in time. We also show that due to large qualitative and quantitative differences between the four solution components – cation concentration, electric potential, *x*-directional displacement, and *y*-directional displacement – it is efficient to approximate them on different meshes using a novel adaptive multi-mesh *hp*-FEM. The study is accompanied with computations and comparisons of the adaptive multi-mesh *hp*-FEM with a number of adaptive FEM algorithms.

1. Introduction

Ionic polymer–metal composites (IPMC) have been studied for their potential to serve as noiseless mechanoelectrical and electromechanical transducers (see Basu and Sharma [1], Shahinpoor and Kim [2], Newbury and Leo [3], Wallmersperger et al. [4], Porfiri [5], Chen and Tan [6], Pugal et al. [7,8], Zangrilli and Weiland [9], Rossiter et al. [10]). The advantages of IPMC over other electroactive polymer actuators are low voltage bending, high strains (over 1%), and an ability to work in wet environments.

An IPMC consists of a thin ionomeric polymer membrane that is typically Nafion [11,12]. The membrane is electrolessly coated with a thin layer of a noble metal electrode, such as platinum or gold. The polymer membrane contains fixed anions and solvent with freely movable cations, so the overall charge of the material is balanced. The model of the material is shown in Fig. 1. Upon applying a voltage to the electrodes, the cations start migrating due to the imposed electric field. By dragging along the solvent, the osmotic pressure difference near the electrodes results in bending of the material (see Fig. 2).

In this study we model IPMC materials via a multiphysics coupled problem consisting of the Poisson and Nernst–Planck equations (further abbreviated by PNP) and Navier's equations for displacement. These equations are used to calculate charge transport and resulting electromechanical transduction of the material.

* Corresponding authors at: Institute of Technology, University of Tartu, Tartu, Estonia.
E-mail addresses: david.pugal@gmail.com (D. Pugal), solin@unr.edu (P. Solin), kwang.kim@unlv.edu (K.J. Kim), alvo.aabloo@ut.ee (A. Aabloo).

0377-0427/$ – see front matter
http://dx.doi.org/10.1016/j.cam.2013.09.011

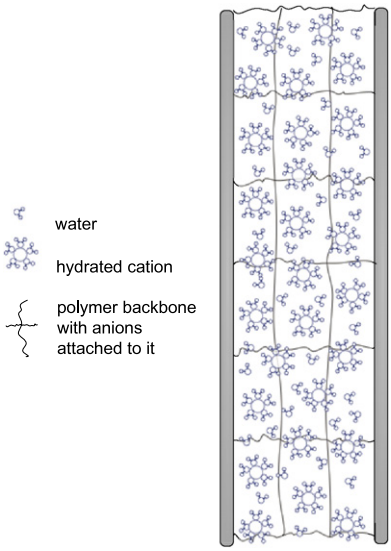


Fig. 1. IPMC model—no voltage is applied to the electrodes.

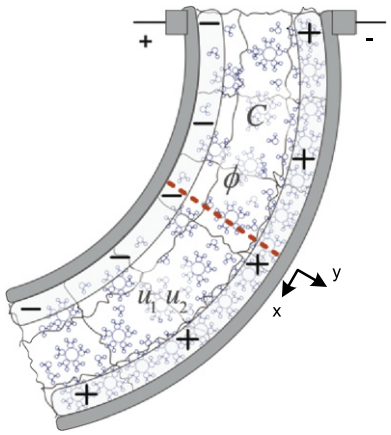


Fig. 2. IPMC model—a voltage is applied to the electrodes. Four physical fields: cation concentration C, electric potential ϕ, x-directional displacement u_1, and y-directional displacement u_2 are used to describe the electromechanical transduction of IPMC.

1.1. Motivation

The PNP system is highly nonlinear and for a typical domain with two electrodes, the largest differences in cation concentration occur in a very narrow region near the boundary. The computing power required for a full scale problem can be significant. Fig. 3 shows cation concentration C, electric potential ϕ, x-directional displacement u_1, and y-directional displacement u_2 in the cross-section of an IPMC (see the dashed line in Fig. 2) in response to an applied voltage.

A solution of the PNP system (Fig. 3, main plot) has some notable characteristics. For the most part, gradient of cation concentration (∇C) is equal to zero. Close to the cathode, ∇C is very large but rather steady in time in terms of Debye length or spatial distribution. Furthermore, ∇C is relatively small near the anode boundary but it shifts in time. Secondly, ϕ is for the most part a smooth function but it has a large gradient near the anode boundary. At the same time, displacement field components in the calculation domain are rather smooth functions (Fig. 3, sub-plot). This makes the choice of an optimal mesh problematic. Even in case of the stationary solution, an optimal mesh for C could be never optimal for ϕ, u_1, and u_2.

That is the reason why we are interested in exploring adaptive algorithms [13–15]—we hope that using these algorithms results in the meshes that are optimal in terms of calculation time and calculation error in a 2D domain.

2. Model

We will model IPMC materials via a multiphysics coupled problem consisting of the Poisson and Nernst–Planck equations (abbreviated by PNP in the following) coupled to Navier's equations. First, the PNP system is used to calculate time dependent

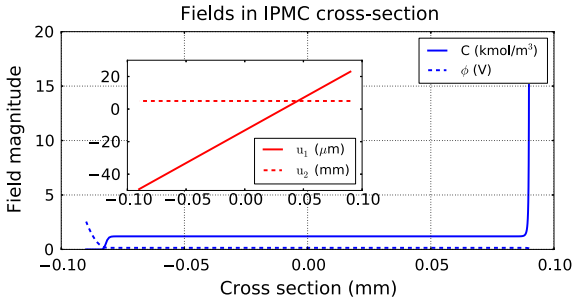


Fig. 3. Calculated fields C, ϕ (on the main graph), u_1, and u_2 (on the subgraph) in IPMC cross-section (see Fig. 2).

spatial charge in IPMC polymer domain—the charge transport is the key underlying mechanism of electromechanical transduction of IPMC.

The Nernst–Planck equation for the mobile cations has the form

$$\frac{\partial C}{\partial t} + \nabla \cdot (-D\nabla C - \mu F C \nabla \phi) = 0, \tag{1}$$

where C stands for cation concentration with the initial value of C_0, D is diffusion constant, μ mobility, F Faraday constant, and ϕ voltage. The Poisson equation has the form

$$-\nabla^2 \phi = \frac{F\rho}{\varepsilon} \tag{2}$$

where ε is the absolute dielectric permittivity. The charge density $\rho = C - C_0$ where C_0 is a constant anion concentration.

In order to calculate deformation of the material, the PNP system of equation is coupled with Navier's equation system for linear elastic material. For low frequency actuation, the time independent Navier's equation can be used to calculate deformation of IPMC as a function of time and local charge density in the material. Based on the strain–displacement relation

$$e_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{3}$$

and equilibrium equation

$$\tau_{ij,j} + F_i = 0, \tag{4}$$

the constitutive equation of the linear elasticity is as follows:

$$\tau_{ij} = \lambda \delta_{ij} e_{kk} + 2\mu e_{ij}. \tag{5}$$

From there, Navier's equation in the vector form can be expressed as

$$(\lambda + \mu)\, u_{k,ki} + \mu u_{i,jj} + F_i = 0, \tag{6}$$

where u_i is a component of the displacement vector and F_i is a component of body force ($\frac{\mathrm{N}}{\mathrm{m}^3}$) [16]. Constants λ and μ are Lame's constants

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{7}$$

where E is Young's modulus and ν Poisson's ratio. In 2D Cartesian coordinates (6) takes the form

$$(\lambda + \mu)\left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_2}{\partial y \partial x}\right) + \mu\left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2}\right) + F_1 = 0, \tag{8}$$

$$(\lambda + \mu)\left(\frac{\partial^2 u_1}{\partial x \partial y} + \frac{\partial^2 u_2}{\partial y^2}\right) + \mu\left(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}\right) + F_2 = 0. \tag{9}$$

In case of IPMC, $F_2 = 0$ and F_1 can be expressed as a function of cation concentration $F_1 = A(C - C_0)$, where A is a constant [17].

2.1. Calculation domain

We consider a rectangular 2D domain $\Omega \subset \mathbb{R}^2$ with boundaries $\partial\Omega_{1\ldots4} \subset \partial\Omega$, shown in Fig. 4. As there is no flow through the domain's boundary, (1) is equipped with a Neumann boundary condition (BC)

$$-D\frac{\partial C}{\partial n} - \mu F C \frac{\partial \phi}{\partial n} = 0. \tag{10}$$

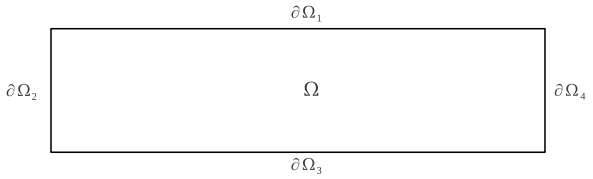


Fig. 4. Calculation domain $\Omega \subset \mathbb{R}^2$ with boundaries $\partial\Omega_{1\ldots4} \subset \partial\Omega$.

We prescribe a positive constant voltage V_{pos} on $\partial\Omega_1$ and zero voltage on $\partial\Omega_3$:

$$\phi_{\partial\Omega_1} = V_{pos}, \qquad \phi_{\partial\Omega_3} = 0. \tag{11}$$

On the rest of the boundary, ϕ has zero normal derivatives, and are thus equipped with Neumann BCs

$$\frac{\partial\phi_{\Omega_2}}{\partial n} = \frac{\partial\phi_{\Omega_4}}{\partial n} = 0. \tag{12}$$

For Navier's equations (8)–(9), the following Dirichlet BCs are applied:

$$u_{1_{\partial\Omega_2}} = u_{2_{\partial\Omega_2}} = 0. \tag{13}$$

As no external forces are considered, zero Neumann BCs are applied on $\partial\Omega$:

$$\tau_{ij} n_j|_{\partial\Omega} = 0. \tag{14}$$

To make the results easily reproducible, in the following we present the derivation of weak forms of (1), (2), (8), (9) as well as formulas for the Jacobian matrix and residual vector that are used in actual computations.

2.2. Weak form of the PNP system

To simplify the notation, we use a dimensionless formulation of (1) and (2). The following new notations for the independent variables x, y, t and for the dependent variables C and ϕ are used [18]:

$$X = \frac{x}{l}, \qquad Y = \frac{y}{l}, \qquad \tau = \frac{tD}{\lambda_D l}, \qquad \varphi = \frac{\phi F}{RT}, \qquad c = \frac{C}{C_0}. \tag{15}$$

Here λ_D is the Debye screening length and it is expressed as follows [18]:

$$\lambda_D = \sqrt{\frac{\varepsilon RT}{2F^2 C_0}}. \tag{16}$$

After inserting variables (15) into (1) the PNP system of equations becomes:

$$\frac{DC_0}{\lambda_D l}\frac{\partial c}{\partial \tau} + \frac{1}{l}\nabla_d \cdot \left(-\frac{DC_0}{l}\nabla_d c - c\frac{DC_0}{l}\nabla_d \varphi\right) = 0 \tag{17}$$

$$-\frac{\varepsilon RT}{l^2 F^2 C_0}\nabla_d^2 \varphi = c - 1, \tag{18}$$

where

$$\nabla_d = \left(\frac{\partial}{\partial X}, \frac{\partial}{\partial Y}\right). \tag{19}$$

After simplifying (17) and (18) and denoting

$$\epsilon = \frac{\lambda_D}{l}, \tag{20}$$

the dimensionless form of the PNP system of equations is

$$\frac{\partial c}{\partial \tau} - \epsilon\nabla_d^2 c - \epsilon\nabla_d \cdot (c\nabla_d \varphi) = 0, \tag{21}$$

$$-\nabla_d^2 \varphi = \frac{c - 1}{2\epsilon^2}. \tag{22}$$

Boundary condition (10) has the form

$$-\frac{\partial c}{\partial n} - c\frac{\partial \varphi}{\partial n} = 0. \tag{23}$$

As the second derivatives of both c and φ are present in the equations, the appropriate function space for them is the Sobolev space $V = H^1(\Omega)$ where

$$H^1(\Omega) = \left\{ v \in L^2(\Omega);\ \nabla_d v \in \left[L^2(\Omega)\right]^2 \right\}. \tag{24}$$

In order to derive the weak form of the Nernst–Planck equation (21), we first multiply it with a test function $v^c \in V$ and integrate over the domain Ω. After applying Green's first identity to the terms that contain second derivatives, we obtain

$$\int_\Omega \frac{\partial c}{\partial \tau} v^c d\mathbf{x} + \epsilon \int_\Omega \nabla_d c \cdot \nabla_d v^c d\mathbf{x} - \epsilon \int_\Omega \nabla_d c \cdot \nabla_d \varphi v^c d\mathbf{x} + \epsilon \int_\Omega \nabla_d \left(c v^c\right) \cdot \nabla_d \varphi d\mathbf{x} - \epsilon \int_{\partial\Omega} \frac{\partial c}{\partial n} v^c d\mathbf{S} - \int_{\partial\Omega} \epsilon \frac{\partial \varphi}{\partial n} c v^c d\mathbf{S} = 0. \tag{25}$$

After expanding the nonlinear term and using the boundary condition (23), we obtain the final weak form of the Nernst–Planck equations:

$$\int_\Omega \frac{\partial c}{\partial \tau} v^c d\mathbf{x} + \epsilon \int_\Omega \nabla_d c \cdot \nabla_d v^c d\mathbf{x} + \epsilon \int_\Omega c \left(\nabla_d \varphi \cdot \nabla_d v^c\right) d\mathbf{x} = 0. \tag{26}$$

Analogously we derive also the weak form of the Poisson equation (22),

$$\int_\Omega \nabla_d \varphi \cdot \nabla_d v^\varphi d\mathbf{x} - \frac{1}{2\epsilon^2} \left[\int_\Omega c v^\varphi d\mathbf{x} - \int_\Omega v^\varphi d\mathbf{x} \right] = 0. \tag{27}$$

For more detailed derivation, see [19].

2.3. Weak form of the Navier's equation system

As in case of the PNP system, a dimensionless formulation of Navier's equations (8) and (9) is derived by denoting the components of displacement field:

$$U_1 = \frac{u_1}{l}, \qquad U_2 = \frac{u_2}{l}. \tag{28}$$

It could be observed from (15) and (28) that in the dimensionless formulation, variables with the unit of m are in the upper case, whereas other variables are in the lower case. After inserting variables (28) as well as variable c into (8) and (9), Navier's equations become:

$$(\lambda + \mu) \left(\frac{\partial^2 U_1}{\partial X^2} + \frac{\partial^2 U_2}{\partial X \partial Y} \right) + \mu \left(\frac{\partial^2 U_1}{\partial X^2} + \frac{\partial^2 U_1}{\partial Y^2} \right) = AlC_0(1 - c), \tag{29}$$

$$(\lambda + \mu) \left(\frac{\partial^2 U_1}{\partial X \partial Y} + \frac{\partial^2 U_2}{\partial Y^2} \right) + \mu \left(\frac{\partial^2 U_2}{\partial X^2} + \frac{\partial^2 U_2}{\partial Y^2} \right) = 0. \tag{30}$$

As in the case of deriving the weak form of the PNP system, the Sobolev space is used for the test functions. First, we multiply (29) with a test function $v^{U_1} \in V$ and integrate over the domain Ω. By applying the zero external boundary conditions, the resulting final weak form of (29) is:

$$(2\mu + \lambda) \int_\Omega \frac{\partial U_1}{\partial X} \frac{\partial v^{U_1}}{\partial X} d\mathbf{x} + \lambda \int_\Omega \frac{\partial U_2}{\partial Y} \frac{\partial v^{U_1}}{\partial X} d\mathbf{x} + \mu \int_\Omega \frac{\partial U_1}{\partial Y} \frac{\partial v^{U_1}}{\partial Y} + \frac{\partial U_2}{\partial X} \frac{\partial v^{U_1}}{\partial Y} d\mathbf{x} + lAC_0 \int_\Omega (1 - c)\, v^{U_1} d\mathbf{x} = 0. \tag{31}$$

Dimensionless formulation of the equation can be obtained by first writing constants μ and λ explicitly in terms of ν and E and then multiplying the equation by $(1 + \nu)/E$:

$$\left(1 + \frac{\nu}{1 - 2\nu}\right) \int_\Omega \frac{\partial U_1}{\partial X} \frac{\partial v^{U_1}}{\partial X} d\mathbf{x} + \frac{\nu}{1 - 2\nu} \int_\Omega \frac{\partial U_2}{\partial Y} \frac{\partial v^{U_1}}{\partial X} d\mathbf{x} + \frac{1}{2} \int_\Omega \frac{\partial U_1}{\partial Y} \frac{\partial v^{U_1}}{\partial Y} + \frac{\partial U_2}{\partial X} \frac{\partial v^{U_1}}{\partial Y} d\mathbf{x} + \frac{1 + \nu}{E} C_0 lA \int_\Omega (1 - c)\, v^{U_1} d\mathbf{x} = 0. \tag{32}$$

Now the LHS consists only dimensionless constants and variables and the displacements are governed solely by the RHS term that consists of material properties and is also a function of cation concentration c. Similarly, the weak form of (30) is:

$$\left(1 + \frac{\nu}{1 - 2\nu}\right) \int_\Omega \frac{\partial U_2}{\partial Y} \frac{\partial v^{U_2}}{\partial Y} d\mathbf{x} + \frac{\nu}{1 - 2\nu} \int_\Omega \frac{\partial U_1}{\partial X} \frac{\partial v^{U_2}}{\partial Y} d\mathbf{x} + \frac{1}{2} \int_\Omega \frac{\partial U_2}{\partial X} \frac{\partial v^{U_2}}{\partial X} + \frac{\partial U_1}{\partial Y} \frac{\partial v^{U_2}}{\partial X} d\mathbf{x} = 0. \tag{33}$$

2.4. PNP Jacobian matrix and residual vector for Newton's method

To employ Newton's method for the nonlinear system (26), (27), (32) and (33), formulas for the Jacobian matrix and residual vector need to be derived. Time discretization will be performed using the second-order Crank–Nicolson method.

For instance, in case of cation concentration c, the unknown solution components c^{n+1} at the end of the time step $\delta\tau$ are expressed as linear combinations of finite element basis functions v_k^c:

$$c^{n+1} = c(\mathcal{Y}^{n+1}) = \sum_{k=1}^{N^c} y_k^c v_k^c. \tag{34}$$

Similar notation is used for φ, U_1, and U_2. Here $\mathcal{Y}^{n+1}$ is a coefficient vector of length $N^c + N^\varphi + N^{U_1} + N^{U_2}$ comprising the unknown solution coefficients $y_k^c, y_k^\varphi, y_k^{U_1}$, and $y_k^{U_2}$ (in this order). We will also be using $c^n = c(\mathcal{Y}^n), \varphi^n = \varphi(\mathcal{Y}^n), U_1^n = U_1(\mathcal{Y}^n)$, and $U_2^n = U_2(\mathcal{Y}^n)$ for the previous time step solutions. Based on that, (26) leads to the formula for the first part $\mathcal{F}^c$ of the residual vector $\mathcal{F}$,

$$\begin{aligned}\mathcal{F}_i^c(\mathcal{Y}) = &\int_\Omega \frac{c(\mathcal{Y})}{\delta\tau} v_i^c d\mathbf{x} - \int_\Omega \frac{c^n}{\delta\tau} v_i^c d\mathbf{x} + \frac{1}{2}\epsilon\left[\int_\Omega \nabla_d c(\mathcal{Y}) \cdot \nabla_d v_i^c d\mathbf{x}\right.\\ &\left. + \int_\Omega \nabla_d c^n \cdot \nabla_d v_i^c d\mathbf{x} + \int_\Omega c(\mathcal{Y})\left(\nabla_d \varphi(\mathcal{Y}) \cdot \nabla_d v_i^c\right) d\mathbf{x} + \int_\Omega c^n\left(\nabla_d \varphi^n \cdot \nabla_d v_i^c\right) d\mathbf{x}\right],\end{aligned} \tag{35}$$

(27) defines the second part $\mathcal{F}^\varphi$,

$$\mathcal{F}_i^\varphi(\mathcal{Y}) = \int_\Omega \nabla_d \varphi(\mathcal{Y}) \cdot \nabla_d v_i^\varphi d\mathbf{x} - \frac{1}{2\epsilon^2}\left[\int_\Omega c(\mathcal{Y}) v_i^\varphi d\mathbf{x} - \int_\Omega v_i^\varphi d\mathbf{x}\right], \tag{36}$$

and the 3rd and the 4th components of the residual vector can be respectively written:

$$\begin{aligned}\mathcal{F}_i^{U_1}(\mathcal{Y}) = &\left(1 + \frac{\nu}{1-2\nu}\right)\int_\Omega \frac{\partial U_1(\mathcal{Y})}{\partial X}\frac{\partial v^{U_1}}{\partial X} d\mathbf{x} + \frac{\nu}{1-2\nu}\int_\Omega \frac{\partial U_2(\mathcal{Y})}{\partial Y}\frac{\partial v^{U_1}}{\partial X} d\mathbf{x}\\ &+ \frac{1}{2}\int_\Omega \frac{\partial U_1(\mathcal{Y})}{\partial Y}\frac{\partial v^{U_1}}{\partial Y} + \frac{\partial U_2(\mathcal{Y})}{\partial X}\frac{\partial v^{U_1}}{\partial Y} d\mathbf{x} + \frac{1+\nu}{E} C_0 lA \int_\Omega (1 - c(\mathcal{Y}))\, v^{U_1} d\mathbf{x},\end{aligned} \tag{37}$$

$$\begin{aligned}\mathcal{F}_i^{U_2}(\mathcal{Y}) = &\left(1 + \frac{\nu}{1-2\nu}\right)\int_\Omega \frac{\partial U_2(\mathcal{Y})}{\partial Y}\frac{\partial v^{U_2}}{\partial Y} d\mathbf{x} + \frac{\nu}{1-2\nu}\int_\Omega \frac{\partial U_1(\mathcal{Y})}{\partial X}\frac{\partial v^{U_2}}{\partial Y} d\mathbf{x}\\ &+ \frac{1}{2}\int_\Omega \frac{\partial U_2(\mathcal{Y})}{\partial X}\frac{\partial v^{U_2}}{\partial X} + \frac{\partial U_1(\mathcal{Y})}{\partial Y}\frac{\partial v^{U_2}}{\partial X} d\mathbf{x}.\end{aligned} \tag{38}$$

The nonlinear discrete problem that needs to be solved at the end of each time step has the form $\mathcal{F}(\mathcal{Y}) = 0$. The Jacobian matrix $\mathcal{J}(\mathcal{Y}) = D\mathcal{F}/D\mathcal{Y}$ has a 4×4 block structure, and its entries are obtained by calculating the partial derivatives of $\mathcal{F}$ with respect to the components of the coefficient vector $\mathcal{Y}$. For this it is useful to realize that

$$\frac{\partial c(\mathcal{Y})}{\partial y_j^c} = v_j^c, \qquad \frac{\partial \nabla_d c(\mathcal{Y})}{\partial y_j^c} = \nabla_d v_j^c, \quad \text{etc.}$$

The non-zero components of the matrix are as follows:

$$\frac{\partial \mathcal{F}_i^c}{\partial y_j^c}(\mathcal{Y}) = \int_\Omega \frac{1}{\delta\tau} v_j^c v_i^c d\mathbf{x} + \frac{1}{2}\epsilon\left[\int_\Omega \nabla_d v_j^c \cdot \nabla_d v_i^c d\mathbf{x} + \int_\Omega v_j^c\left(\nabla_d \varphi(\mathcal{Y}) \cdot \nabla_d v_i^c\right) d\mathbf{x}\right], \tag{39}$$

$$\frac{\partial \mathcal{F}_i^c}{\partial y_j^\varphi}(\mathcal{Y}) = \frac{1}{2}\epsilon\int_\Omega c(\mathcal{Y})\left(\nabla_d v_j^\varphi \cdot \nabla_d v_i^c\right) d\mathbf{x}, \tag{40}$$

$$\frac{\partial \mathcal{F}_i^\varphi}{\partial y_j^c}(\mathcal{Y}) = -\frac{1}{2\epsilon^2}\int_\Omega v_j^c v_i^\varphi d\mathbf{x}, \tag{41}$$

$$\frac{\partial \mathcal{F}_i^\varphi}{\partial y_j^\varphi}(\mathcal{Y}) = \int_\Omega \nabla_d v_j^\varphi \cdot \nabla_d v_i^\varphi d\mathbf{x}. \tag{42}$$

$$\frac{\partial \mathcal{F}_i^{U_1}}{\partial y_j^c}(\mathcal{Y}) = -\frac{1+\nu}{E} C_0 lA \int_\Omega v_j^c v_i^{U_1} d\mathbf{x}, \tag{43}$$

$$\frac{\partial \mathcal{F}_i^{U_1}}{\partial y_j^{U_1}}(Y) = \left(1 + \frac{\nu}{1-2\nu}\right)\int_\Omega \frac{\partial v_j^{U_1}}{\partial X}\frac{\partial v_i^{U_1}}{\partial X} d\mathbf{x} + \frac{1}{2}\int_\Omega \frac{\partial v_j^{U_1}}{\partial Y}\frac{\partial v_i^{U_1}}{\partial Y} d\mathbf{x}, \tag{44}$$

$$\frac{\partial \mathcal{F}_i^{U_1}}{\partial y_j^{U_2}}(\mathcal{Y}) = \frac{\nu}{1-2\nu}\int_\Omega \frac{\partial v_j^{U_2}}{\partial Y}\frac{\partial v_i^{U_1}}{\partial X} d\mathbf{x} + \frac{1}{2}\int_\Omega \frac{\partial v_j^{U_2}}{\partial X}\frac{\partial v_i^{U_1}}{\partial Y} d\mathbf{x}, \tag{45}$$

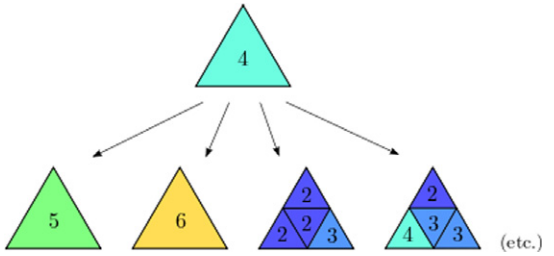


Fig. 5. Some of the possible refinement candidates for a fourth-order element.
Source: From [19].

$$\frac{\partial \mathcal{F}_i^{U_2}}{\partial y_j^{U_1}}(\mathcal{Y}) = \frac{\nu}{1-2\nu}\int_\Omega \frac{\partial v_j^{U_1}}{\partial X}\frac{\partial v_i^{U_2}}{\partial Y}d\mathbf{x} + \frac{1}{2}\int_\Omega \frac{\partial v_j^{U_1}}{\partial Y}\frac{\partial v_i^{U_2}}{\partial X}d\mathbf{x}, \tag{46}$$

$$\frac{\partial \mathcal{F}_i^{U_2}}{\partial y_j^{U_2}}(\mathcal{Y}) = \left(1+\frac{\nu}{1-2\nu}\right)\int_\Omega \frac{\partial v_j^{U_2}}{\partial Y}\frac{\partial v_i^{U_2}}{\partial Y}d\mathbf{x} + \frac{1}{2}\int_\Omega \frac{\partial v_j^{U_2}}{\partial X}\frac{\partial v_i^{U_2}}{\partial X}d\mathbf{x}. \tag{47}$$

See [19] for applying the derived residual vector and matrix in Newton's iteration calculations.

3. *hp*-FEM

The *hp*-FEM is a modern version of the finite element method (FEM) that is capable of exponential convergence (the approximation error drops exponentially as new degrees of freedom are added during adaptivity) while standard FEM can only attain algebraic (polynomial) convergence rates which are much slower [20].

Automatic adaptivity in higher-order finite element methods (*hp*-FEM) is much different from adaptivity in low-order FEM. Firstly, analytical error estimates capable of guiding adaptive *hp*-FEM do not exist even for the simplest linear elliptic equations, not speaking about nonlinear multiphysics coupled systems [15]. Secondly, a higher-order element can be refined in many different ways, as illustrated in Fig. 5. The number of possible element refinements is implementation dependent. In general it is very low in *h*-adaptivity and *p*-adaptivity, and much higher in *hp*-adaptivity. Moreover, this number grows very fast when anisotropic refinements are enabled.

3.1. The Hermes library

Hermes[1] is a free and open-source C++ library that implements higher-order finite elements approximations and adaptive *hp*-FEM. It supports 8 different adaptivity modes—three isotropic and five anisotropic. It has been shown that for the given problem, the isotropic adaptivities are rather inefficient [19]. The supported anisotropic refinement modes are *h*-anisotropic (H_ANISO), *hp*-anisotropic-*h* (HP_ANISO_H), *p*-anisotropic (P_ANISO), *hp*-anisotropic-p (HP_ANISO_P), and *hp*-anisotropic (HP_ANISO). It must be noted that in case of HP_ANISO_H, only element size is adapted anisotropically whereas polynomial degree is adapted isotropically. The opposite holds true for HP_ANISO_P. The adaptivity modes are shown in Fig. 6.

Due to the large number of refinement options, classical error estimators that provide a constant error estimate per element, cannot be used to guide automatic *hp*-adaptivity. For this, one needs to know the shape of the approximation error. Hermes uses a pair of approximations with different orders of accuracy to obtain this information: coarse mesh solution and fine mesh solution [21]. The initial coarse mesh is read from the mesh file, and the initial fine mesh is created through its global refinement both in *h* and *p*. The fine mesh solution is the approximation of interest both during the adaptive process and at the end of computation. Global orthogonal projection of the fine mesh solution on the coarse mesh is used to extract the low-order part from the reference solution. The adaptivity algorithm is guided by the difference between the reference solution and its low-order part. Note that this approach to automatic adaptivity is PDE-independent and thus naturally applicable to a large variety of multiphysics coupled problems.

3.2. Multi-mesh hp-FEM

In multiphysics PDE systems such as PNP, it can happen that one physical field is very smooth whereas others are not. If all the fields are approximated on the same mesh, then unnecessary refinements will be present in smooth areas where they are not necessary. This can be very wasteful.

[1] http://hpfem.org/hermes.

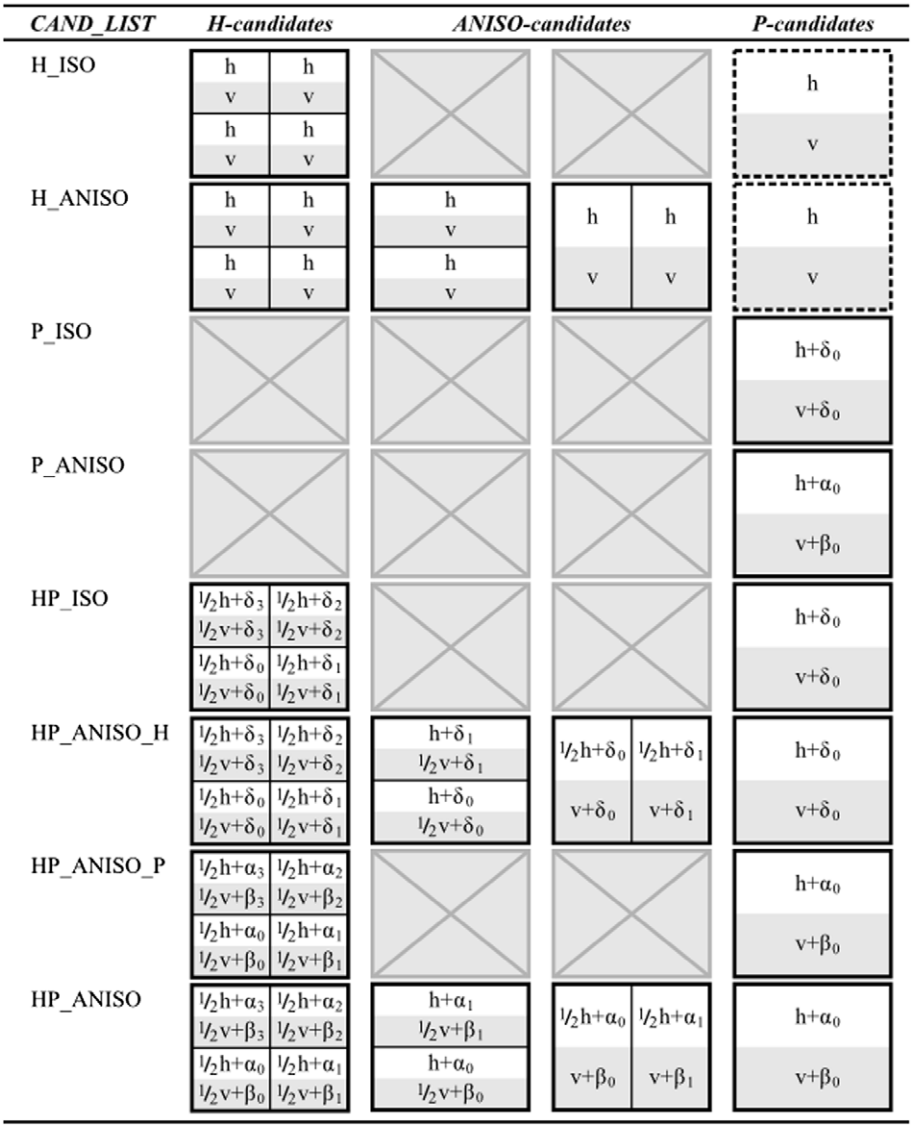


Fig. 6. Refinement candidates for the refinement modes.
Source: From [19].

Hermes implements a novel adaptive multi-mesh *hp*-FEM [22,13,23] that makes it possible to approximate different fields on individual meshes, without breaking the monolithic structure of the coupling mechanism. For practical reasons, the meshes in the system are not allowed to be completely independent—they have a common coarse mesh that we call *master mesh*. The exact preservation of the coupling structure of multiphysics coupled problems makes the multi-mesh *hp*-FEM very different from various interpolation and projection based methods that suffer from errors made while transferring data between different meshes in the system.

4. *hp*-FEM calculations of IPMC electromechanical transduction

The PNP–Navier problem was implemented in Hermes and a large number of calculations were carried out in order to find the best adaptivity mode. Furthermore, multi-mesh and single-mesh calculations were compared for CPU time and problem size in terms of number of degrees of freedom (DOF). A deformation of IPMC calculated with Hermes is illustrated in Fig. 7.

In order to find the best adaptive mode for this type of problems, we performed numerous computations using various adaptivity modes in (a) single-mesh, (b) multi-mesh with common mesh for displacement fields, and (c) multi-mesh for all fields configurations. In the numerical experiments we paid attention to the relative error and problem size in terms of DOF

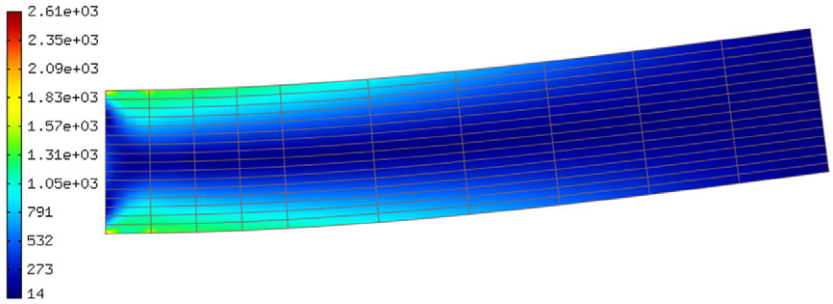


Fig. 7. Example deformation of IPMC calculated with Hermes *hp*-FEM software package. The color indicates local von-Mises stress. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

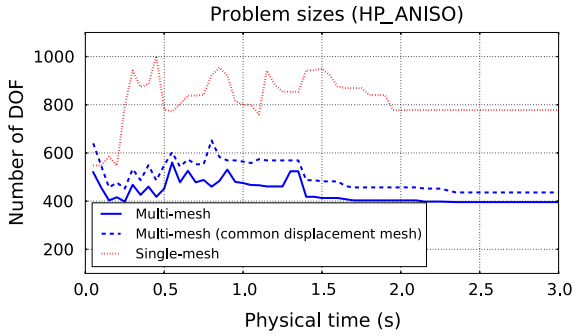


Fig. 8. Number of DOFs as a function of physical time for single-mesh and multi-mesh configurations with HP_ANISO adaptivity mode.

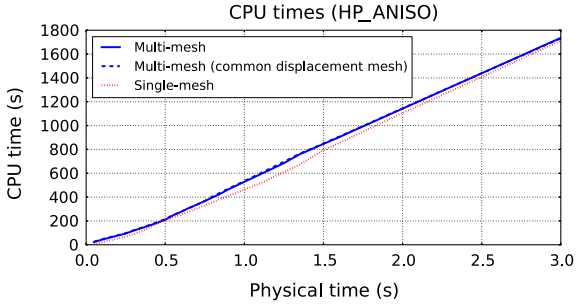


Fig. 9. Cumulative CPU time as a function of physical time for single-mesh and multi-mesh configurations with HP_ANISO adaptivity mode.

at each time step. The maximum relative error was fixed to 1.0%. Following, a number of comparative calculation results are presented.

4.1. Advantages of multi-mesh hp-FEM

Simulations with different adaptivity modes and meshes showed that the multi-mesh *hp*-FEM configuration results in the smallest problems compared to any single-mesh configuration. At the same time, the computing time and error convergence were similar for all cases. For instance, single vs. multi-meshes comparison for the HP_ANISO adaptivity mode are shown in Figs. 8 and 9.

Displacement fields U_1 and U_2 are similar in terms of gradient and very different from c and φ. Therefore, one of the multi-mesh configuration was based on three different meshes—a mesh for c, a mesh for φ, and a common mesh for U_1 and U_2. Interestingly, the "true" multi-mesh (separate mesh for each field) turned out to be the most efficient configuration. Calculating the multi-mesh costs more CPU time, but simulations run on other meshes are more expensive, resulting very similar CPU-time cuves (Fig. 9). However, the problem size of the multi-mesh is the smallest (Fig. 8). Thus, in all of the following simulations, the multi-mesh configuration is used.

4.2. Anisotropic refinements

The calculations were carried out with three different anisotropic adaptivity modes—HP_ANISO, HP_ANISO_H, and HP_ANISO_P. We have already showed that the rest of the adaptivity types (isotropic, P_ANISO, H_ANISO) do not perform

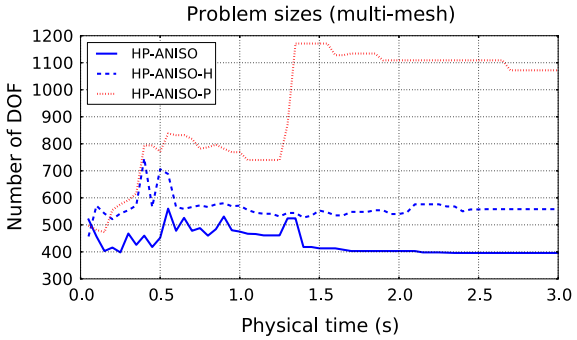


Fig. 10. Problem size with different adaptivity modes in the multi-mesh configuration.

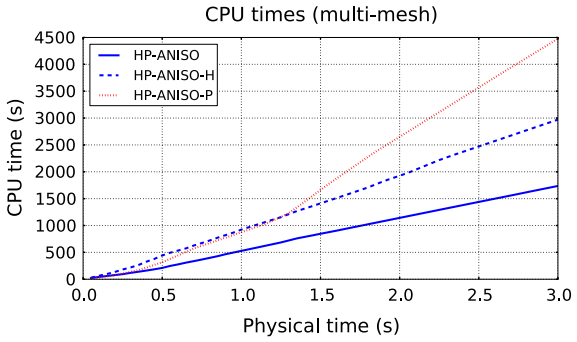


Fig. 11. CPU time with different adaptivity modes in the multi-mesh configuration.

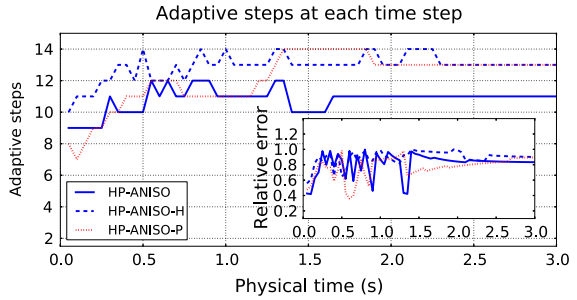
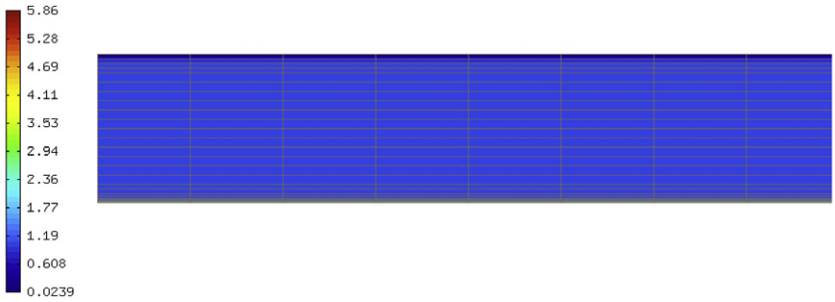


Fig. 12. Number of adaptive steps at each time (main plot) and final relative error between a coarse and fine mesh solutions (sub-plot).

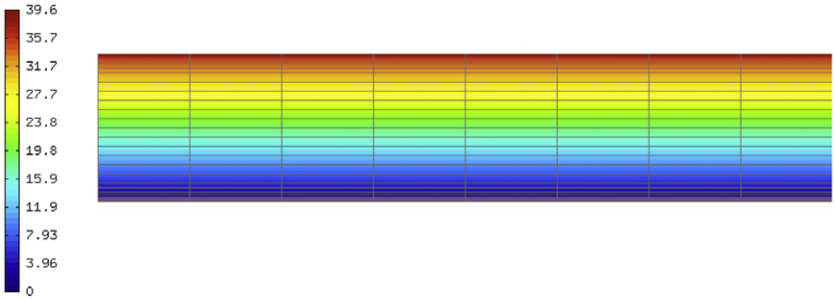
well for this type of multiphysics problems [19]. For instance, in case of *p*-adaptivities it is necessary to know the problem and gradients rather well to start with a suitable initial mesh or the error might never converge below a pre-set limit. At the same time, HP_ANISO, HP_ANISO_H, and HP_ANISO_P all perform relatively well. Thus, in this study we tried to determine which one of them is the most suitable for such a coupled problem as is the time dependent IPMC actuation.

It can be seen in Figs. 10 and 11 that both the problem size and CPU time are consistently smallest for HP_ANISO adaptivity. Interestingly, in case of HP_ANISO_P, both CPU time and number of DOF start rapidly increasing at the point where the other modes stabilize. It can be attributed to the isotropic *h*-refinements for the steep boundary gradients. Namely, Fig. 12 shows the number of adaptive steps (main plot) it takes to reach to the desired error level (Fig. 12 subplot). Although the number of steps for HP_ANISO_P stabilizes close to HP_ANISO_H (13 adaptivity steps per time step), the latter is more efficient—apparently, isotropic *h*-adaptivity is a rather expensive way to reduce the error of a solution.

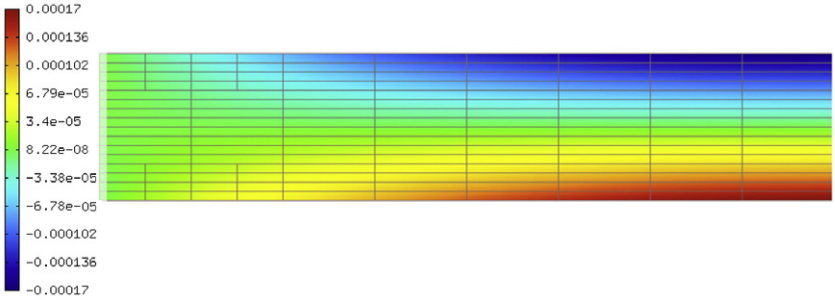
To illustrate the effectiveness of HP_ANISO, a snapshot of the meshes and polynomial degrees were recorded for each of the fields. Fig. 13 shows the magnitudes and meshes for c, φ, U_1 and U_2 at $t = 0.1$ s. The mesh is significantly adapted near the $\partial\Omega_1$ and $\partial\Omega_3$ boundaries for c and φ—this can be explained by the steep gradients of these fields (as shown in Fig. 3). On the other hand, the displacement field mesh is rather coarse. In case of U_1, some of the elements are split vertically whereas in most of the elements, no vertical splits occur. Fig. 14 shows local polynomial degrees for the fields at the same time. Again, due to the steep gradient of c near $\partial\Omega_3$, the polynomial degree near that boundary is 8. For other fields, the maximum polynomial degrees are mostly equal or less than 4. Overall, the HP_ANISO adaptivity mode appears to be the most suitable for the such multi-field multiphysics problem due to the resulting small problem size and relative fastness compared to the other anisotropic adaptivities.



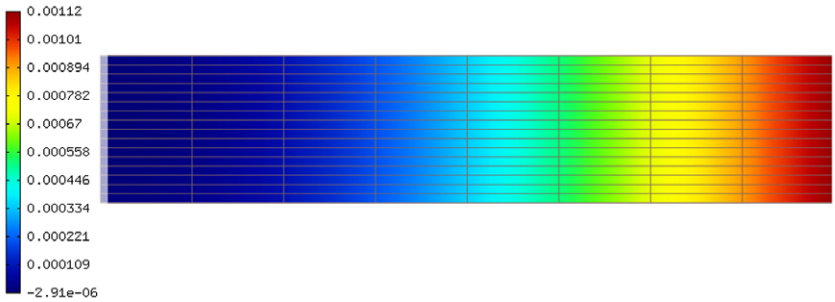
(a) Cation concentration field, magnitude and mesh.



(b) Electric potential fields, magnitude and mesh.



(c) *x*-directional displacement field, magnitude and mesh.

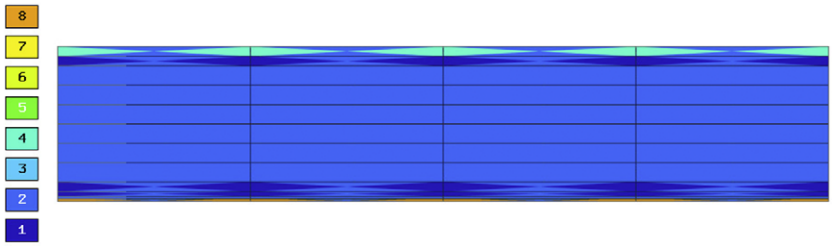


(d) *y*-directional displacement field, magnitude and mesh.

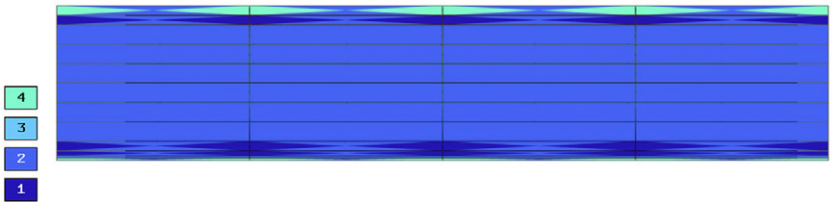
Fig. 13. c, φ, U_1, and U_2 field meshes and magnitudes at $t = 0.1$ s in (a), (b), (c), and (d), respectively. Notice that the meshes are optimized for the particular fields.

4.3. Calculations with more advanced BCs

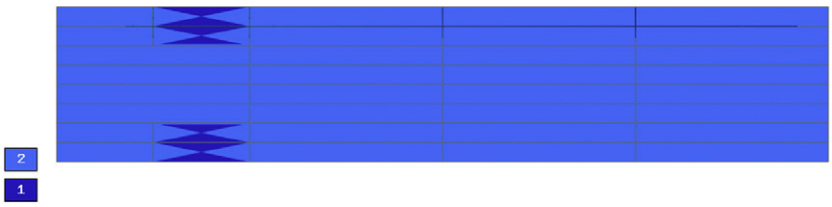
In the previous section, HP_ANISO, HP_ANISO_H, and HP_ANISO_P adaptivity modes were compared in IPMC electromechanical transduction calculation (PNP–Euler system of equations) based on constant voltage BCs. In more advanced calculations, the applied voltages on the boundaries $\partial\Omega_1$ and $\partial\Omega_3$ have to be considered as gradients rather than constants. This



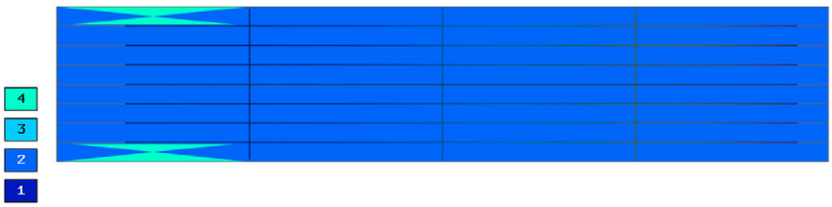
(a) Cation concentration field.



(b) Electric potential field, orders.



(c) x-directional displacement field, orders.



(d) y-directional displacement field, orders.

Fig. 14. c, φ, U_1, and U_2 field polynomial degrees at $t = 0.1$ s in (a), (b), (c), and (d), respectively. Higher order elements are used in case of c and φ due to steep boundary gradients.

can be, for instance, due to the high resistance of the electrodes or underlying electrochemical currents as explained in [17]. In regular FEM calculations, rather fine mesh has to be used as the boundary gradients can result in solution instability—for our configuration, the concentration gradient can peak in the corner of Ω (see Fig. 4). To see how the anisotropic adaptivity modes perform, the potentials on these boundaries were applied as linear gradients. On $\partial\Omega_1$, the potential had linear drop of 25% from $x = 0$ toward the tip. Thus, for 1 V applied voltage, the tip voltage was 0.75 V. The same gradient was applied on $\partial\Omega_3$, i.e. from 0 to 0.25 V at the tip. With gradient BCs the concentration gradient ∇c and the voltage gradient $\nabla\varphi$ are no longer effectively in 1D. The goal of running simulations with multiple adaptivity types was to explore if additional x-directional gradients of c and φ would result in different adaptivity performances. It must be noted that in more advanced calculations, the gradient is also dynamic, however, this would require considering the electrode domains and is not within the scope of this work.

Figs. 15 and 16 show problem sizes and CPU times for different adaptivity modes with potential gradient BCs. It could be reasoned that HP_ANISO_P can be a suitable as the x-directional gradient can require also vertical mesh refinements. However, as in the previous case, the HP_ANISO results in the smallest CPU time and problem size.

To illustrate, calculated φ in Ω and corresponding meshes and polynomial degrees of the elements at $t = 0.1$ s are shown in Fig. 17. Notice that the solution and polynomial degrees are notably different to the ones in Figs. 13(b) and 14(b). The HP_ANISO adaptivity algorithm has particularly increased the polynomial degree and refined the mesh near Ω_1 where a sharp concentration peaks occur due to the x-directional applied potential gradient. Also, the mesh is more refined. The

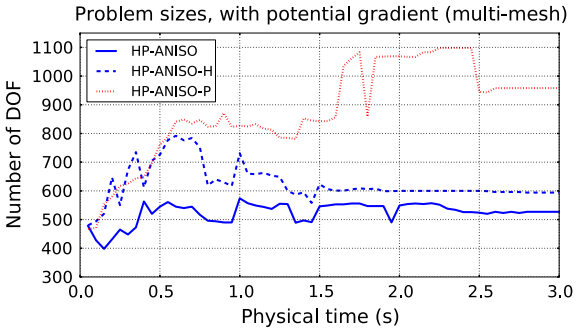


Fig. 15. Problem size with different adaptivity modes in the multi-mesh configuration and applied potential gradient BCs.

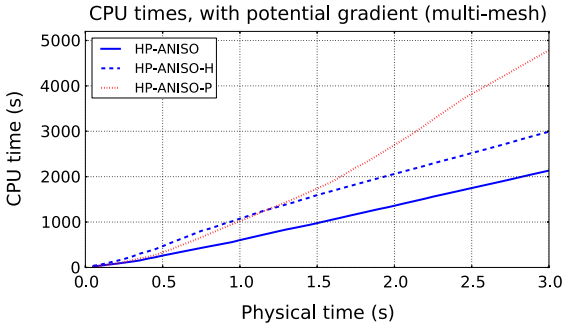


Fig. 16. CPU time with different adaptivity modes in the multi-mesh configuration and applied potential gradient BCs.

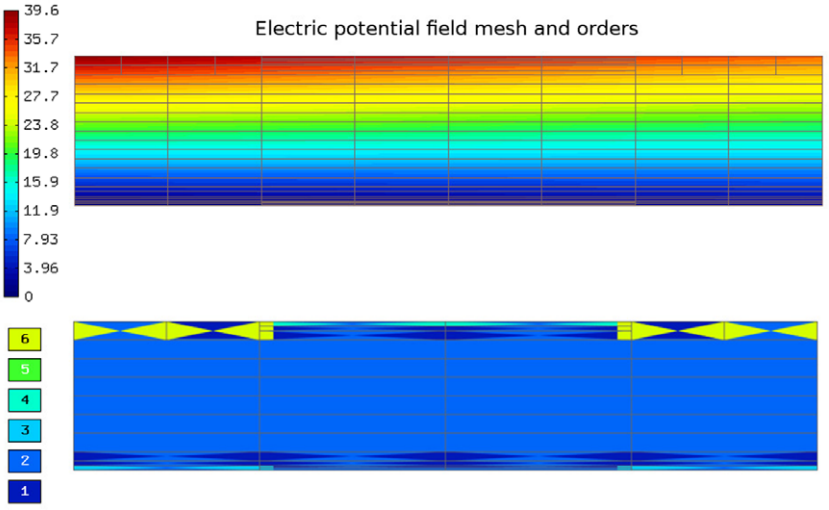


Fig. 17. Calculated scaled φ field magnitude, mesh, and polynomial degrees with applied potential gradients along $\partial\Omega_1$ and $\partial\Omega_3$ at $t = 0.1$ s.

effect is similar in case of c; however, it is easier to observe φ as the boundary is wider for the latter. This example clearly illustrates how the solution of the equations with non-uniform electric potential boundary conditions is dynamic and how the HP_ANISO time dependent adaptivity calculates an optimal mesh and polynomial space to adapt to the dynamics of the problem.

5. Conclusions

In this work, IPMC electromechanical transduction was modeled by solving the system of Nernst–Planck–Poisson–Euler equations using *hp*-finite element method with adaptive multi-mesh configuration. The weak form, residuals and the Jacobian matrix of the system were explicitly derived—the result can be used in a number of finite element software packages, however, here we demonstrated using it with Hermes *hp*-FEM time dependent adaptive solver. The solution for the

problem with four dimensionless field variables – cation concentration c, electric potential φ, x-directional displacement U_1, and y-directional displacement U_2 – results in very different field gradients in the space and time. When using a conventional low order FEM, finding an optimal mesh for this type of problem such that both the error of the solution and problem size remain small throughout the time dependent solving process is difficult.

We showed that using the time dependent adaptivity, multi-mesh configuration, and anisotropic *hp* refinements, the problem size remains very small throughout the solving process while maintaining a pre-set relative error of the solution. Namely, Hermes refinement mode HP_ANISO resulted in the smallest and fastest problem solution. The meshes were significantly refined for c and φ and also the maximum polynomial degree was varied in the range of $2 \ldots 8$ whereas for the displacement fields U_1 and U_2, the mesh variations were smaller; however, the polynomial degrees of the elements were increased where necessary.

Conclusively, by using *hp*-FEM with adaptive multi-mesh configuration we can possibly reduce the problem size of the Nernst–Planck–Poisson–Euler equation system significantly while still maintaining prescribed precision of the solution. We believe, and this is yet to be shown, that the effect is especially noticeable when dealing with the same problem in a 3D domain.

Acknowledgments

KJK thanks partial financial support from Office of Naval Research (N000141210387). PS acknowledges Grant No. P102/11/0498 of the Grant Agency of the Czech Republic. AA and DP acknowledge ETF Grant ETF 8553.

References

[1] S. Basu, M. Sharma, An improved space-charge model for flow through charged microporous membranes, Journal of Membrane Science 124 (1997) 77–91.
[2] M. Shahinpoor, K.J. Kim, Ionic polymer–metal composites: I. fundamentals, Smart Materials and Structures 10 (2001) 819.
[3] K. Newbury, D. Leo, Linear electromechanical model of ionic polymer transducers-part I: model development, Journal of Intelligent Material Systems and Structures 14 (2003) 333.
[4] T. Wallmersperger, D.J. Leo, C.S. Kothera, Transport modeling in ionomeric polymer transducers and its relationship to electromechanical coupling, Journal of Applied Physics 101 (2007) 024912.
[5] M. Porfiri, Charge dynamics in ionic polymer metal composites, Journal of Applied Physics 104 (2008) 104915.
[6] Z. Chen, X. Tan, A control-oriented and physics-based model for ionic polymer–metal composite actuators, Mechatronics, IEEE/ASME Transactions on 13 (2008) 519–529.
[7] D. Pugal, K.J. Kim, A. Punning, H. Kasemagi, M. Kruusmaa, A. Aabloo, A self-oscillating ionic polymer–metal composite bending actuator, Journal of Applied Physics 103 (2008) 084908.
[8] D. Pugal, K. Jung, A. Aabloo, K. Kim, Ionic polymer–metal composite mechanoelectrical transduction: review and perspectives, Polymer International 59 (2010) 279–289.
[9] U. Zangrilli, L.M. Weiland, Prediction of the ionic polymer transducer sensing of shear loading, Smart Materials and Structures 20 (2011) 094013.
[10] J. Rossiter, B. Stoimenov, T. Mukai, A bistable artificial muscle actuator, in: International Symposium on Micro-NanoMechatronics and Human Science, 2006, pp. 1–6.
[11] S. Nemat-Nasser, Micromechanics of actuation of ionic polymer–metal composites, Journal of Applied Physics 92 (2002) 2899–2915.
[12] S. Nemat-Nasser, Y. Wu, Comparative experimental study of ionic polymer–metal composites with different backbone ionomers and in various cation forms, Journal of Applied Physics 93 (2003) 5255–5267.
[13] P. Solin, L. Dubcova, J. Kruis, Adaptive *hp*-FEM with dynamical meshes for transient heat and moisture transfer problems, Journal of Computational and Applied Mathematics 233 (2010) 3103–3112.
[14] S. Schnepp, E. Gjonaj, T. Weiland, A hybrid finite integration–finite volume scheme, Journal of Computational Physics 229 (2010) 4075–4096.
[15] S.M. Schnepp, T. Weiland, Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method, Journal of Computational and Applied Mathematics 236 (2012) 4909–4924. FEMTEC 2011: 3rd International Conference on Computational Methods in Engineering and Science, May 9–13, 2011.
[16] J. Heinbockel, Introduction to Tensor Calculus and Continuum Mechanics, Trafford Publishing, 2001.
[17] D. Pugal, K.J. Kim, A. Aabloo, An explicit physics-based model of ionic polymer–metal composite actuators, Journal of Applied Physics 110 (2011) 084904.
[18] M. Bazant, K. Thornton, A. Ajdari, Diffuse-charge dynamics in electrochemical systems, Physical Review E 70 (2004) 21506.
[19] D. Pugal, P. Solin, K.J. Kim, A. Aabloo, Modeling ionic polymer–metal composites with space–time adaptive multimesh hp-fem, Communications in Computational Physics 11 (2012) 249–270.
[20] P. Solin, K. Segeth, I. Dolezel, Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, 2003.
[21] P. Solin, D. Andrs, J. Cerveny, M. Simko, PDE-independent adaptive *hp*-FEM based on hierarchic extension of finite element spaces, Journal of Computational and Applied Mathematics 233 (2010) 3086–3094.
[22] P. Solin, J. Cerveny, L. Dubcova, D. Andrs, Monolithic discretization of linear thermoelasticity problems via adaptive multimesh hp-FEM, Journal of Computational and Applied Mathematics 234 (2010) 2350–2357.
[23] L. Dubcova, P. Solin, J. Cerveny, P. Kus, Space and time adaptive two-mesh *hp*-finite element method for transient microwave heating problems, Electromagnetics 30 (2010) 23–40.