Modeling Ionic Polymer-Metal Composites with Space-Time Adaptive Multimesh *hp*-FEM

Deivid Pugal[1,4], Pavel Solin[2,3,*], Kwang J. Kim[1], and Alvo Aabloo[4]

[1] *Mechanical Engineering Department, University of Nevada, Reno, NV, U.S.A.*
[2] *Department of Mathematics and Statistics, University of Nevada, Reno, NV, U.S.A.*
[3] *Institute of Thermomechanics, Prague, Czech Republic*
[4] *Institute of Technology, Tartu University, Estonia*

Abstract. We are concerned with a model of ionic polymer-metal composite (IPMC) materials that consists of a coupled system of the Poisson and Nernst-Planck equations, discretized by means of the finite element method (FEM). We show that due to the transient character of the problem 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 two solution components, it is efficient to approximate them on different meshes using a novel adaptive multimesh *hp*-FEM. The study is accompanied with numerous computations and comparisons of the adaptive multimesh *hp*-FEM with several other adaptive FEM algorithms.

AMS subject classifications: 35Q84, 35Q05, 35J47

Key words: Ionic polymer-metal composites, IPMC, Nernst-Planck equation, Poisson equation, finite element method, FEM, adaptive multimesh *hp*-FEM

1 Introduction

Ionic Polymer-Metal Composites (IPMC) have been studied during the past two decades for their potential to serve as noiseless mechanoelectrical and electromechanical transducers [1, 3–7, 12]. The advantages of IPMC over other electroactive polymer actuators are low voltage bending, high strains ($>1\%$), and an ability to work in wet environments. A typical IPMC consists of a thin sheet of polymer (often Nafion or Teflon) which is sandwiched between noble metal electrodes such as platinum or gold. When fabricated, the polymer membrane is saturated with certain solvent and ions such as water and H^+. When a voltage is applied to the electrodes, the counter ions 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. 1).

*Corresponding author. *Email addresses:* david.pugal@gmail.com (D. Pugal), solin@unr.edu (P. Solin), kwangkim@unr.edu (K. Kim), alvo@ut.ee (A. Aabloo)

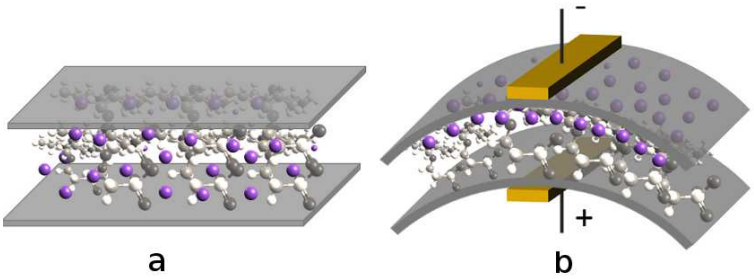

Figure 1: Conceptual model of the actuation of IPMC. Initial counter ion distribution (a) and the distribution and resulting bending after applying a voltage (b).

In this study we will model IPMC materials via a multiphysics coupled problem consisting of the Poisson and Nernst-Planck equations (abbreviated by PNP in the following). These equations are used to model charge transport in materials that includes ionic migration, diffusion, and convection. The charge transport process is a key mechanism for electromechanical transduction.

The PNP system is highly nonlinear and for a typical domain with two electrodes, largest differences in charge concentration occur in a very narrow region near the boundary. The computing power required for a full scale problem is significant. This is why we are interested in exploring adaptive algorithms – we hope to obtain meshes that are optimal in terms of calculation time and calculation error.

The Nernst-Planck equation for a mobile species — in our case for counter ions — has the form

$$\frac{\partial C}{\partial t}+\nabla\cdot(-D\nabla C-z\mu FC\nabla\phi)=0. \tag{1.1}$$

Here C stands for the counter ion concentration, D is diffusion, μ mobility, F Faraday constant, ϕ voltage, and z the charge number. We have neglected the velocity of the species as in our case it can be assumed zero. The Poisson equation has the form

$$-\nabla^2\phi=\frac{F\rho}{\varepsilon} \tag{1.2}$$

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

The outline of the paper is as follows: Section 2 shows that the solution components C and ϕ have very different behavior, which is the reason why it is difficult to find a common mesh that would be optimal for both of them. This explains why we are interested in approximation them on individual meshes equipped with mutually independent adaptivity mechanisms. The PNP model is presented in Section 3 where also its weak formulation for the Newton's method is derived. Section 4 presents a brief overview of a novel adaptive multimesh hp-FEM method [2,9–11] that is used to solve the problem numerically. Numerical results and comparisons are presented in Section 5, and conclusion and outlook are drawn in Section 6.

2 Motivation

In this section we use a simplified one-dimensional PNP model to illustrate the principal difficulties encountered in the numerical solution. Table 1 shows relevant constants.

Table 1: Constants used in the Poisson-Nernst-Planck system of equations.

Constant	Value	Unit	Description
D	10×10^{-11}	$\frac{m^2}{s}$	Diffusion constant
z	1	-	Charge number
F	96,485	$\frac{C}{mol}$	Faraday number
R	8.31	$\frac{J}{mol\cdot K}$	The gas constant
$\mu\ \left(=\frac{D}{RT}\right)$	4.11×10^{-14}	$\frac{s}{mol\cdot K}$	Mobility
C_0	1,200	$\frac{mol}{m^3}$	Anion concentration
ε	0.025	$\frac{F}{m}$	Electric permittivity

Fig. 2 shows a typical solution for C and ϕ at $t=0.1\ s$ and $t=3.0\ s$.

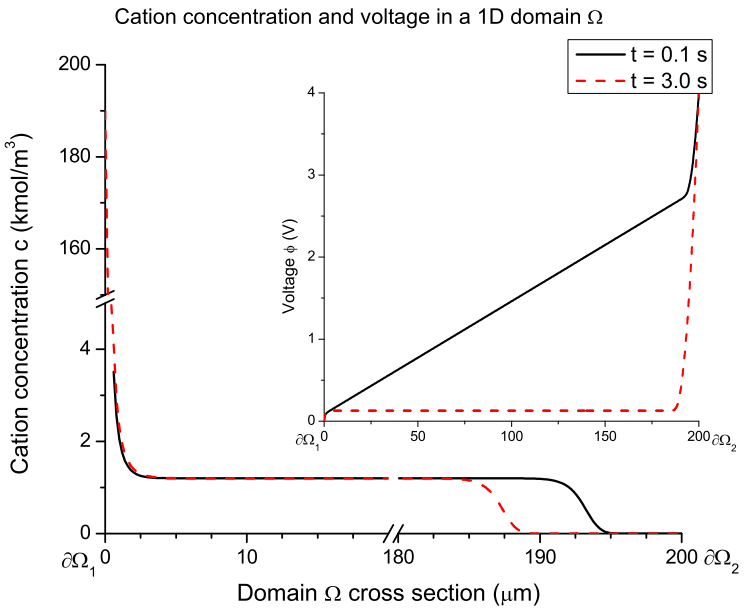

Figure 2: Sample concentration C and voltage ϕ in a 1D domain $\Omega\subset\mathbb{R}$. Dirichlet boundary conditions ($V_{\partial\Omega_1}=0\ V$ and $V_{\partial\Omega_2}=4\ V$) were applied to the Poisson equation (1.2) and Neumann conditions to the Nernst-Planck equation (1.1).

The reader can observe that the solution has two notable characteristics: For the most part of the domain Ω, the gradient $\nabla C=0$. Close to $\partial\Omega_2$, ∇C is nonzero and moving

in time, and ∇C is very large at $\partial\Omega_1$. At the same time, ϕ is a "nice" smooth function for the most part of Ω but it has a large gradient at $\partial\Omega_2$. This makes the choice of an optimal mesh highly problematic. Even if the solution was stationary, an optimal mesh for C could never be optimal for ϕ and vice versa.

Furthermore, the shape of the solution in Fig. 2 suggests that the polynomial degree of finite elements in the middle of the domain Ω and near the boundaries $\partial\Omega_1$, $\partial\Omega_2$ should be different — large high-degree elements should be used in the middle of the domain while small low-degree ones should be used in the boundary layers. The qualitative differences in the solution components C and ϕ also suggest that using different meshes would be beneficial.

3 Model

We consider a rectangular 2D domain $\Omega \subset \mathbb{R}^2$ with boundaries $\partial\Omega_{1\ldots4} \subset \partial\Omega$, shown in Fig. 3.

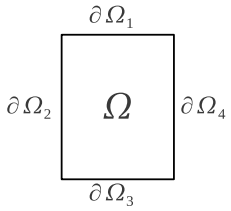

Figure 3: Calculation domain $\Omega \subset \mathbb{R}^2$ with boundaries $\partial\Omega_{1\ldots4} \subset \partial\Omega$.

As there is no flow through the domain's boundary, Eq. (1.1) is equipped with a Neumann boundary condition

$$-D\frac{\partial C}{\partial n} - z\mu FC\frac{\partial \phi}{\partial n} = 0. \tag{3.1}$$

Furthermore, we prescribe a positive constant voltage V_{pos} on Ω_1 and zero voltage on Ω_3:

$$\phi_{\partial\Omega_1} = V_{pos}, \tag{3.2}$$
$$\phi_{\partial\Omega_3} = 0. \tag{3.3}$$

On the rest of the boundary, ϕ has zero normal derivatives, and thus we prescribe a Neumann boundary condition

$$\frac{\partial \phi_{\Omega_2}}{\partial n} = \frac{\partial \phi_{\Omega_4}}{\partial n} = 0. \tag{3.4}$$

3.1 Weak form of the PNP system

To make our results easily reproducible, in the following we present the derivation of weak forms of Eqs. (1.1) and (1.2), as well as formulas for the Jacobian matrix and residual

vector that are used in actual computations. To simplify notation, we define

$$K = z\mu F, \quad L = \frac{F}{\varepsilon}. \tag{3.5}$$

Eqs. (1.1) and (1.2) yield

$$\begin{aligned} \frac{\partial C}{\partial t} - D\nabla^2 C - K\nabla\cdot(C\nabla\phi) &= 0, & (3.6)\\ -\nabla^2\phi - L(C - C_0) &= 0. & (3.7) \end{aligned}$$

Boundary condition Eq. (3.1) has the form

$$-D\frac{\partial C}{\partial n} - KC\frac{\partial \phi}{\partial n} = 0. \tag{3.8}$$

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

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

In order to derive the weak form of the Nernst-Planck equation Eq. (3.6), we first multiply it with a test function $v^C \in V$ and integrate over the domain Ω,

$$\int_\Omega \frac{\partial C}{\partial t} v^C d\mathbf{x} - \int_\Omega D\nabla^2 C v^C d\mathbf{x} - \int_\Omega K\nabla C\cdot\nabla\phi v^C d\mathbf{x} - \int_\Omega KC\nabla^2\phi v^C d\mathbf{x} = 0. \tag{3.9}$$

Applying the Green's first identity to the terms that contain second derivatives, we obtain

$$\begin{aligned} &\int_\Omega \frac{\partial C}{\partial t} v^C d\mathbf{x} + D\int_\Omega \nabla C\cdot\nabla v^C d\mathbf{x} - K\int_\Omega \nabla C\cdot\nabla\phi v^C d\mathbf{x}\\ &+K\int_\Omega \nabla\left(Cv^C\right)\cdot\nabla\phi d\mathbf{x} - D\int_{\partial\Omega}\frac{\partial C}{\partial n}v^C d\mathbf{S} - \int_{\partial\Omega} K\frac{\partial\phi}{\partial n}Cv^C d\mathbf{S} = 0. \end{aligned} \tag{3.10}$$

Expanding the nonlinear term and using the boundary condition (3.8), we have

$$\begin{aligned} &\int_\Omega \frac{\partial C}{\partial t} v^C d\mathbf{x} + D\int_\Omega \nabla C\cdot\nabla v^C d\mathbf{x} - K\int_\Omega \nabla C\cdot\nabla\phi v^C d\mathbf{x}\\ &+K\int_\Omega \nabla\phi\cdot\nabla C v^C d\mathbf{x} + K\int_\Omega C\left(\nabla\phi\cdot\nabla v^C\right) d\mathbf{x} = 0. \end{aligned} \tag{3.11}$$

After the second and third terms cancel out, we obtain the final weak form of the Nernst-Planck equation

$$\int_\Omega \frac{\partial C}{\partial t} v^C d\mathbf{x} + D\int_\Omega \nabla C\cdot\nabla v^C d\mathbf{x} + K\int_\Omega C\left(\nabla\phi\cdot\nabla v^C\right) d\mathbf{x} = 0. \tag{3.12}$$

Analogously we derive also the weak form of the Poisson equation (3.7),

$$-\int_\Omega \nabla^2\phi v^\phi d\mathbf{x}-\int_\Omega LCv^\phi d\mathbf{x}+\int_\Omega LC_0v^\phi d\mathbf{x}=0. \tag{3.13}$$

Performing integration by parts and taking into account the boundary conditions for ϕ, we obtain

$$\int_\Omega \nabla\phi\cdot\nabla v^\phi d\mathbf{x}-\int_\Omega LCv^\phi d\mathbf{x}+\int_\Omega LC_0v^\phi d\mathbf{x}=0. \tag{3.14}$$

3.2 Jacobian matrix and residual vector for the Newton's method

To employ the Newton's method for the nonlinear system (3.12), (3.14), formulas for the Jacobian matrix and residual vector need to be derived. Time discretization will be performed using the second-order Crank-Nicolson method. The unknown solution components C^{n+1} and ϕ^{n+1} at the end of the time step are expressed as linear combinations of finite element basis functions v_k^C and v_k^ϕ with unknown coefficients,

$$C^{n+1}=C(Y^{n+1}) = \sum_{k=1}^{N^C} y_k^C v_k^C, \tag{3.15}$$

$$\phi^{n+1}=\phi(Y^{n+1}) = \sum_{k=1}^{N^\phi} y_k^\phi v_k^\phi. \tag{3.16}$$

Here Y^{n+1} is a coefficient vector of length N^C+N^ϕ comprising the unknown solution coefficients y_k^C and y_k^ϕ (in this order). We will also be using $C^n=C(Y^n)$ and $\phi^n=\phi(Y^n)$ for the previous time step solutions.

With the notation (3.15), (3.16), the time discretized Eq. (3.12) leads to the formula for the first part F^C of the residual vector F,

$$\begin{aligned} F_i^C(Y) = & \int_\Omega \frac{C(Y)}{\tau}v_i^C d\mathbf{x}-\int_\Omega \frac{C^n}{\tau}v_i^C d\mathbf{x} \\ & +\frac{1}{2}\left[D\int_\Omega \nabla C(Y)\cdot\nabla v_i^C d\mathbf{x}+D\int_\Omega \nabla C^n\cdot\nabla v_i^C d\mathbf{x}\right] \\ & +\frac{1}{2}\left[K\int_\Omega C(Y)\left(\nabla\phi(Y)\cdot\nabla v_i^C\right)d\mathbf{x}+K\int_\Omega C^n\left(\nabla\phi^n\cdot\nabla v_i^C\right)d\mathbf{x}\right] \end{aligned} \tag{3.17}$$

where $i=1,2,\ldots,N^C$. Analogously, Eq. (3.14) defines the second part F^ϕ of the residual vector F,

$$F_i^\phi(Y)=\int_\Omega \nabla\phi(Y)\cdot\nabla v_i^\phi d\mathbf{x}-\int_\Omega LC(Y)v_i^\phi d\mathbf{x}+\int_\Omega LC_0v_i^\phi d\mathbf{x} \tag{3.18}$$

where $i=N^C+1,N^C+2,\ldots,N^C+N^\phi$. The nonlinear discrete problem that needs to be solved at the end of each time step thus has the form $F(Y)=0$.

The Jacobian matrix $J(Y) = DF/DY$ has a 2×2 block structure,

$$J(Y) = \begin{pmatrix} \frac{\partial F_i^C}{\partial y_j^C} & \frac{\partial F_i^C}{\partial y_j^\phi} \\ \frac{\partial F_i^\phi}{\partial y_j^C} & \frac{\partial F_i^\phi}{\partial y_j^\phi} \end{pmatrix}, \tag{3.19}$$

and its entries are obtained by calculating the partial derivatives of F with respect to the components of the coefficient vector Y. For this it is useful to realize that

$$\frac{\partial C(Y)}{\partial y_j^C} = v_j^C, \quad \frac{\partial \nabla C(Y)}{\partial y_j^C} = \nabla v_j^C, \quad \text{etc..}$$

We obtain

$$\frac{\partial F_i^C}{\partial y_j^C}(Y) = \int_\Omega \frac{1}{\tau} v_j^C v_i^C d\mathbf{x} + \frac{1}{2} D \int_\Omega \nabla v_j^C \cdot \nabla v_i^C d\mathbf{x} + \frac{1}{2} K \int_\Omega v_j^C \left(\nabla \phi(Y) \cdot \nabla v_i^C \right) d\mathbf{x}, \tag{3.20}$$

$$\frac{\partial F_i^C}{\partial y_j^\phi}(Y) = \frac{1}{2} K \int_\Omega C(Y) \left(\nabla v_j^\phi \cdot \nabla v_i^C \right) d\mathbf{x}, \tag{3.21}$$

$$\frac{\partial F_i^\phi}{\partial y_j^C}(Y) = -\int_\Omega L v_j^C v_i^\phi d\mathbf{x}, \tag{3.22}$$

$$\frac{\partial F_i^\phi}{\partial y_j^\phi}(Y) = \int_\Omega \nabla v_j^\phi \cdot \nabla v_i^\phi d\mathbf{x}. \tag{3.23}$$

3.3 Newton's iteration

At the beginning of the $(n+1)$st time step we set $Y_0^{n+1} = Y^n$, where Y^n is the coefficient vector that was calculated in the nth time step (or coming from the initial condition if $n=0$). We set $k=0$ and run the Newton's iteration

$$\begin{aligned} J(Y_k^{n+1}) \delta Y_{k+1}^{n+1} &= -F(Y_k^{n+1}), \\ Y_{k+1}^{n+1} &= Y_k^{n+1} + \delta Y_{k+1}^{n+1}, \\ k &:= k+1 \end{aligned}$$

over k until it converges. Then we set $Y^{n+1} := Y_k^{n+1}$. We use a combined stopping criterion that makes sure that both the norm of the residual vector $\|F(Y^{n+1})\|$ as well as the norm of the increment $\|\delta Y^{n+1}\|$ are sufficiently small.

4 Adaptive *hp*-FEM and the Open-Source Library Hermes

In traditional low-order FEM (based on piecewise-linear or piecewise quadratic elements), refining an element is not algorithmically complicated, and so the most difficult part is to find out what elements should be refined. To do this, various techniques ranging from rigorous guaranteed a-posteriori error estimates to heuristic criteria such as residual error indicators or error indicators based on steep gradients are employed.

However, none of these approaches is suitable for multiphysics coupled problems or higher-order finite element methods: Rigorous guaranteed error estimates only exist for very simple problems (such as linear elliptic PDE) and only for low-order finite elements. Heuristic techniques are usually somehow doable for all problems, but they fail in more complicated situations. Moreover, they lack a transparent relation to the true approximation error and thus they may give wrong results.

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. Secondly, a higher-order element can be refined in many different ways, as illustrated in Fig. 4.

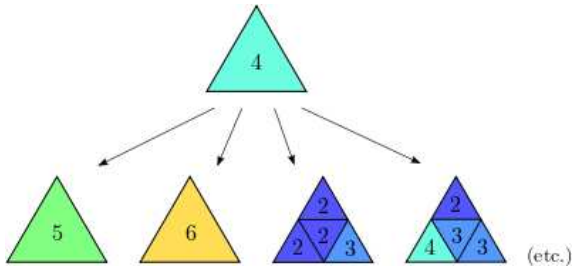

Figure 4: Many possible refinement candidates for a fourth-order element.

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.

4.1 The Hermes Library

Hermes[†] 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. The isotropic refinements are *h*-isotropic (H_ISO), *p*-isotropic (P_ISO), *hp*-isotropic (HP_ISO). Anisotropic refinement modes are *h*-anisotropic (H_ANISO), *hp*-anisotropic-*h* (HP_ANISO_H), *p*-anisotropic (P_ANISO), *hp*-anisotropic-p

[†]http://hpfem.org/hermes

(HP_ANISO_P), and *hp*-anisotropic (HP_ANISO). The eight adaptivity modes are summarized in Fig. 5.

CAND_LIST	*H-candidates*	*ANISO-candidates*		*P-candidates*
H_ISO	h h v v h h v v			h v
H_ANISO	h h v v h h v v	h v h v	h h v v	h v
P_ISO				$h+\delta_0$ $v+\delta_0$
P_ANISO				$h+\alpha_0$ $v+\beta_0$
HP_ISO	$\frac{1}{2}h+\delta_3$ $\frac{1}{2}h+\delta_2$ $\frac{1}{2}v+\delta_3$ $\frac{1}{2}v+\delta_2$ $\frac{1}{2}h+\delta_0$ $\frac{1}{2}h+\delta_1$ $\frac{1}{2}v+\delta_0$ $\frac{1}{2}v+\delta_1$			$h+\delta_0$ $v+\delta_0$
HP_ANISO_H	$\frac{1}{2}h+\delta_3$ $\frac{1}{2}h+\delta_2$ $\frac{1}{2}v+\delta_3$ $\frac{1}{2}v+\delta_2$ $\frac{1}{2}h+\delta_0$ $\frac{1}{2}h+\delta_1$ $\frac{1}{2}v+\delta_0$ $\frac{1}{2}v+\delta_1$	$h+\delta_1$ $\frac{1}{2}v+\delta_1$ $h+\delta_0$ $\frac{1}{2}v+\delta_0$	$\frac{1}{2}h+\delta_0$ $\frac{1}{2}h+\delta_1$ $v+\delta_0$ $v+\delta_1$	$h+\delta_0$ $v+\delta_0$
HP_ANISO_P	$\frac{1}{2}h+\alpha_3$ $\frac{1}{2}h+\alpha_2$ $\frac{1}{2}v+\beta_3$ $\frac{1}{2}v+\beta_2$ $\frac{1}{2}h+\alpha_0$ $\frac{1}{2}h+\alpha_1$ $\frac{1}{2}v+\beta_0$ $\frac{1}{2}v+\beta_1$			$h+\alpha_0$ $v+\beta_0$
HP_ANISO	$\frac{1}{2}h+\alpha_3$ $\frac{1}{2}h+\alpha_2$ $\frac{1}{2}v+\beta_3$ $\frac{1}{2}v+\beta_2$ $\frac{1}{2}h+\alpha_0$ $\frac{1}{2}h+\alpha_1$ $\frac{1}{2}v+\beta_0$ $\frac{1}{2}v+\beta_1$	$h+\alpha_1$ $\frac{1}{2}v+\beta_1$ $h+\alpha_0$ $\frac{1}{2}v+\beta_0$	$\frac{1}{2}h+\alpha_0$ $\frac{1}{2}h+\alpha_1$ $v+\beta_0$ $v+\beta_1$	$h+\alpha_0$ $v+\beta_0$

Figure 5: Refinement candidates for every refinement mode for quad type elements.

Note that triangular elements do not support anisotropic refinements. 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. The initial coarse mesh is read from the mesh file, and the ini-

tial 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 [8]. Note that this approach to automatic adaptivity is PDE-independent and thus naturally applicable to a large variety of multiphysics coupled problems.

4.2 Multimesh *hp*-FEM

In multiphysics PDE systems such as Poisson-Nernst-Planck it can happen that one physical field is very smooth where others are not, as we illustrated in Fig. 2. 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.

Hermes implements a novel adaptive multimesh *hp*-FEM [2,9,11] 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 master mesh is there for algorithmic purposes only and it may not even be used for discretization purposes. Every mesh in the system is obtained from the master mesh via an arbitrary sequence of elementary refinements. Assembling is done on a *union mesh*, a geometrical union of all meshes in the system (imagine printing all meshes on transparencies and positioning them on top of each other).

The union mesh is not constructed physically in the computer memory – it merely serves as a hint to correctly transform the integration points while integrating over subelements of elements in the existing meshes. As a result, the multimesh discretization of the PDE system is monolithic in the sense that no physics is lost — all integrals in the discrete weak formulations are evaluated exactly up to the error in the numerical quadrature. The exact preservation of the coupling structure of multiphysics coupled problems makes the multimesh *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.

5 Numerical Results and Comparisons

The solutions to the PNP problem exhibit a specific behavior that was described above. In order to find the best adaptive method to deal with this type of problems, we performed numerous computations using all adaptivity modes in both the single-mesh and multi-mesh regimes. In the numerical experiments we paid attention to the relative error, cumulative CPU time, and problem size in terms of number of degrees of freedom (DOF) in each time step.

We used two types of initial meshes — a finer mesh shown in Fig. 6 (c) was used for *p*-adaptivity and a very coarse initial mesh shown in Fig. 6 (a) was used for *h*-adaptivity and *hp*-adaptivity.

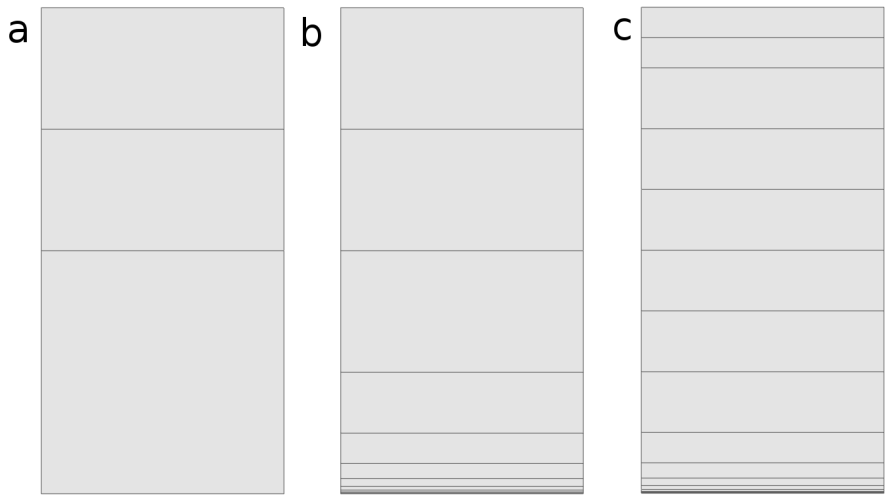

Figure 6: Initial coarse mesh (a), half refined mesh (b) and refined mesh (c). The coarse mesh (a) and refined mesh (c) were used in the initial calculations, the latter one in case of *p*-adaptivity (including HP_ANISO_P). The half-refined mesh (b) was used later to optimize the *hp*-adaptive refinement solutions.

An example of the solution at $t = 0.7\ s$ and $t = 3.0\ s$ calculated with the HP_ANISO refinement mode is shown in Figs. 7 and 8.

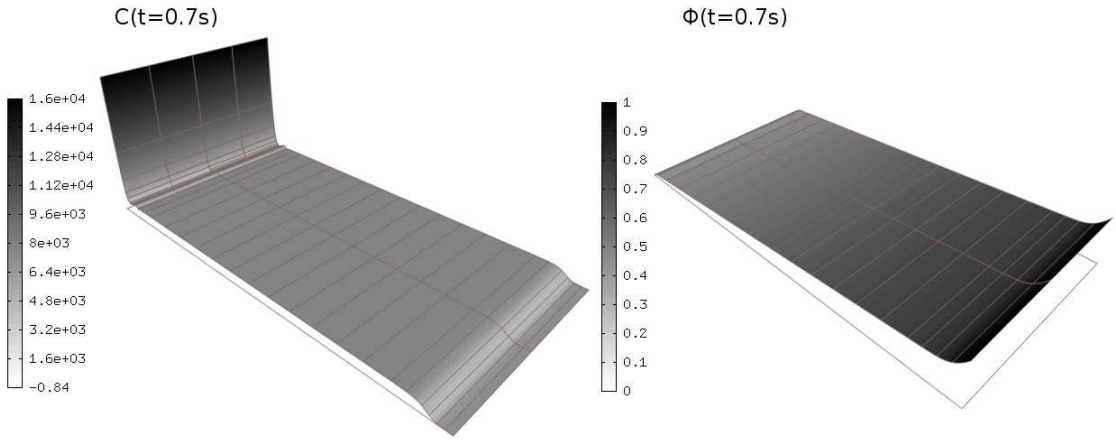

Figure 7: Concentration C and voltage ϕ at $t = 0.7\ s$.

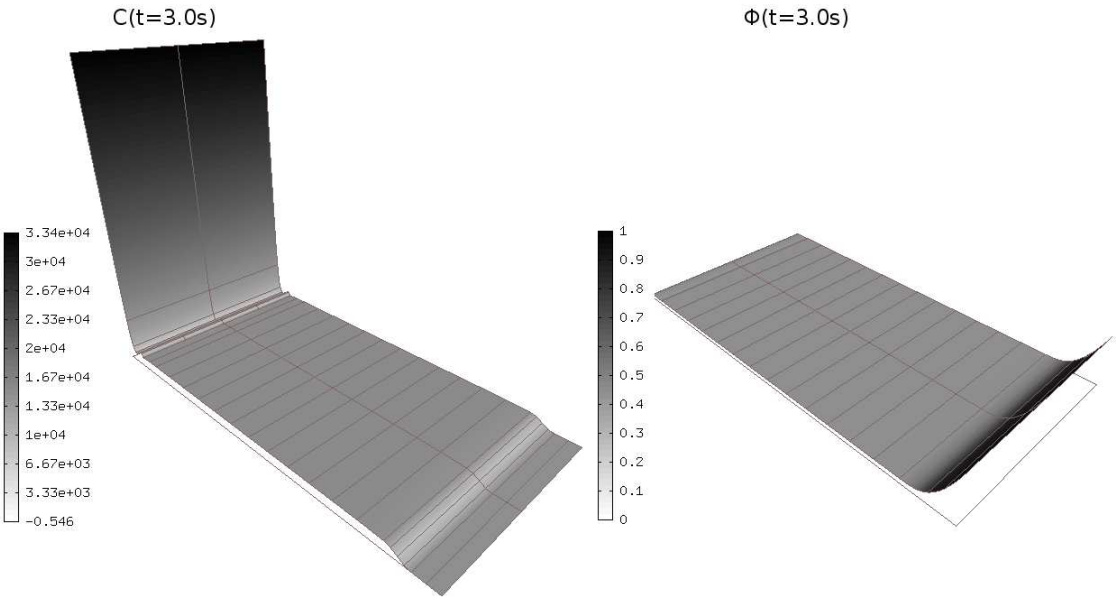

Figure 8: Concentration C and voltage ϕ at $t = 3.0\ s$.

The reader can see that at $t = 0.7\ s$ some ionic migration has already taken place and large concentration gradients near the boundaries $\partial\Omega_1$ and $\partial\Omega_3$ have formed. The figures also show that the meshes at $t = 0.7\ s$ and $t = 3.0\ s$ are different.

5.1 Comparison of single mesh low-order FEM and hp-FEM

Comparison of single mesh H_ANISO with p=1, single mesh H_ANISO with p=2 and single mesh HP_ANISO should be added here to make the point that hp-FEM is much better than low-order FEM. Otherwise the reader does not know why we are just using hp-FEM in the following. Or is there a reason why you do not want to add this here – long computing time for hp-FEM or something else?

5.2 Comparison of single-mesh and multi-mesh hp-FEM

Running the simulation with different adaptivity modes and meshes showed that the multi-mesh hp-FEM configuration resulted into the smallest problems, shortest computing times, and better or the same error convergence compared to any single-mesh configuration. This is illustrated in Figs. 9 and Fig. 10.

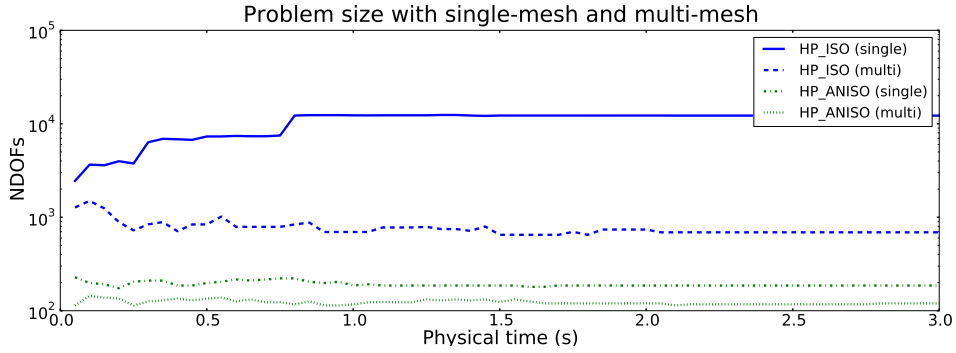

Figure 9: Number of degrees of freedom (DOF) as a function of physical time for single-mesh and multi-mesh configurations with the HP_ISO and HP_ANISO adaptivity modes. Note logarithmic scale on the vertical axis.

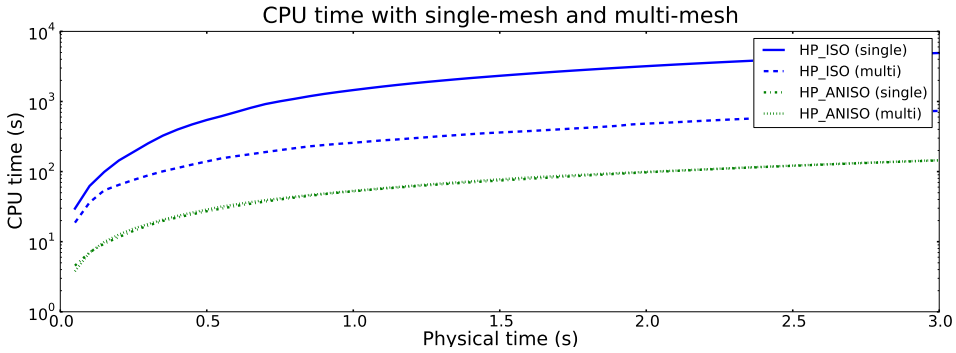

Figure 10: Cumulative CPU time as a function of physical time for single-mesh and multi-mesh configurations with the HP_ISO and HP_ANISO adaptivity modes. Note logarithmic scale on the vertical axis.

Fig. 11 shows higher-order meshes in the adaptive multimesh hp-FEM computation for C and ϕ at $t = 0.7$ s. Different colors mean different polynomial degrees. A diagonal

pattern inside an element tells that the element has different polynomial degrees in the horizontal and vertical directions.

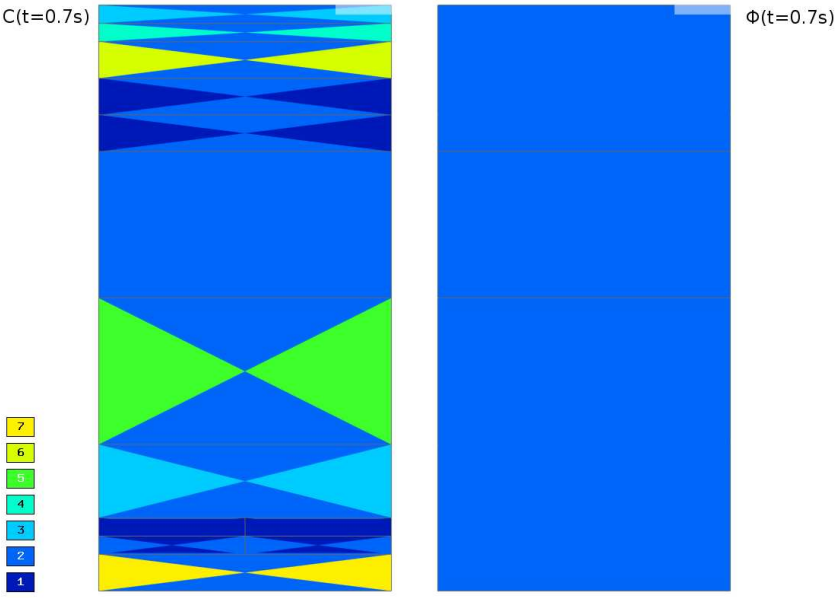

Figure 11: Higher-order FEM mesh for C and ϕ at $t = 0.7\ s$.

These result are in good agreement with Fig. 8 – in the vicinity of the boundaries $\partial\Omega_1$ and $\partial\Omega_3$, the concentration gradient is much greater than the voltage gradient. Therefore the multimesh hp-FEM adaptivity algorithm has increased the maximum polynomial degree for the C-space to 7 while the maximum polynomial degree for the ϕ-space is 2. One can also see that the mesh is significantly more refined for C. Since these results are representative for all adaptivity modes, only multi-mesh configurations are considered in the following.

5.3 Comparison of isotropic and anisotropic refinements

Next we would like to illustrate the role of anisotropic mesh refinements. Figs. 12 and 13 show typical results for the H_ISO, H_ANISO, HP_ISO, HP_ANISO adaptivity modes in terms of DOF and cumulative CPU time.
Figs. 14 and 15 present a similar comparsion for the P_ISO, P_ANISO, and HP_ANISO_P modes. Recall that these computations use a different initial mesh that was a-priori refined in space.

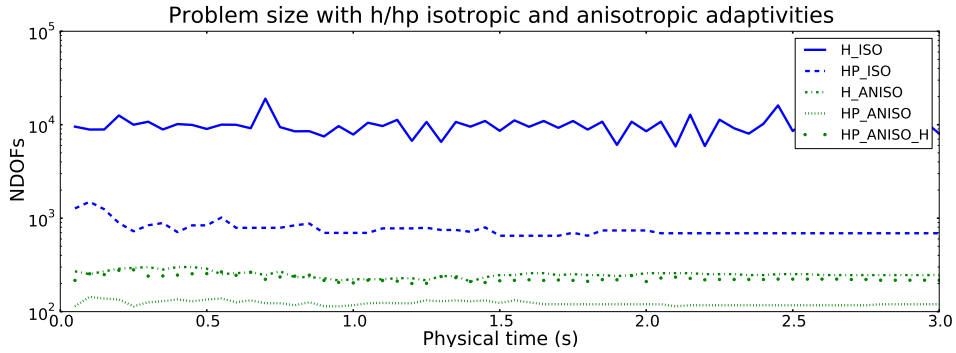

Figure 12: Number of DOF as a function of physical time for multi-mesh configurations with H_ISO, H_ANISO, HP_ISO, and HP_ANISO adaptivity modes.

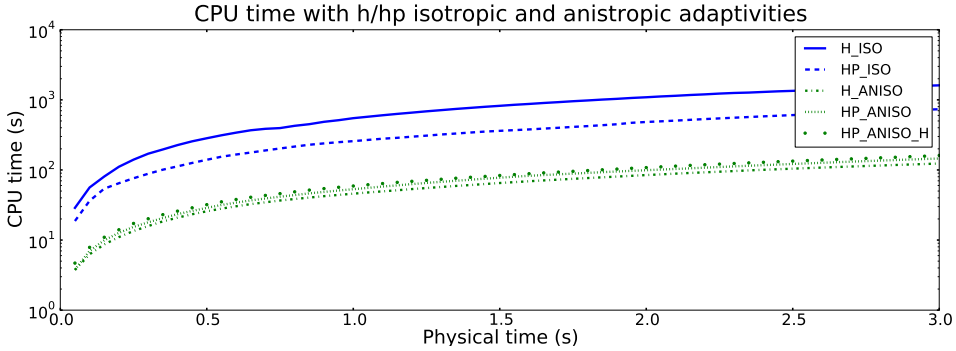

Figure 13: Cumulative CPU time as a function of physical time for multi-mesh configurations with H_ISO, H_ANISO, HP_ISO, and HP_ANISO adaptivity modes.

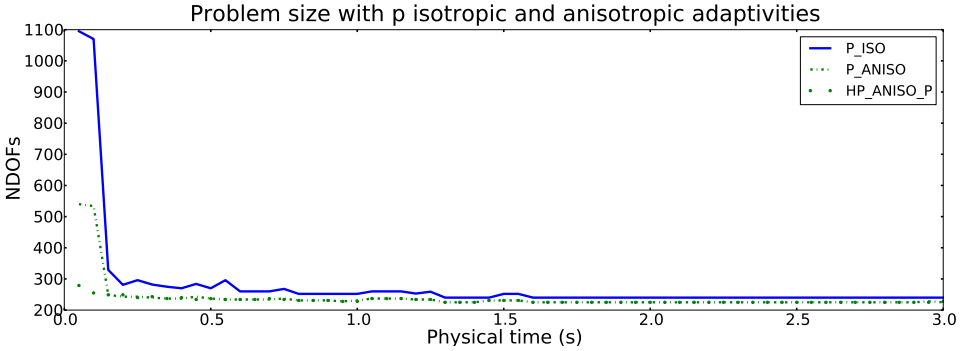

Figure 14: Number of DOF as a function of physical time for multi-mesh configurations with P_ISO, P_ANISO, and HP_ANISO_P adaptivity modes.

As a conclusion, the reader can see that the anisotropic adaptivity modes always perform better than the isotropic ones. In particular, HP_ANISO results into the smallest problem size. In the *p*-adaptivity group, HP_ANISO_P leads to a small problem size con-

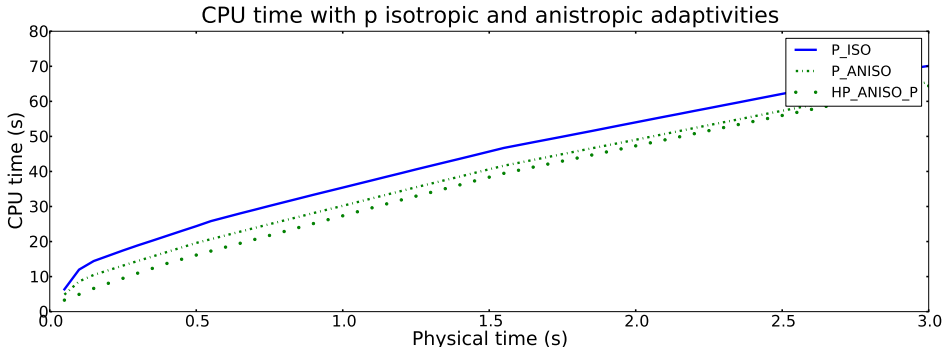

Figure 15: Cumulative CPU times as a function of physical time for multi-mesh configurations with P_ISO, P_ANISO, and HP_ANISO_P adaptivity modes.

sistently in each time step, whereas P_ISO and P_ANISO yield large problems during the first few time steps.

5.4 Optimization of the HP_ANISO and HP_ANISO_P adaptivity modes

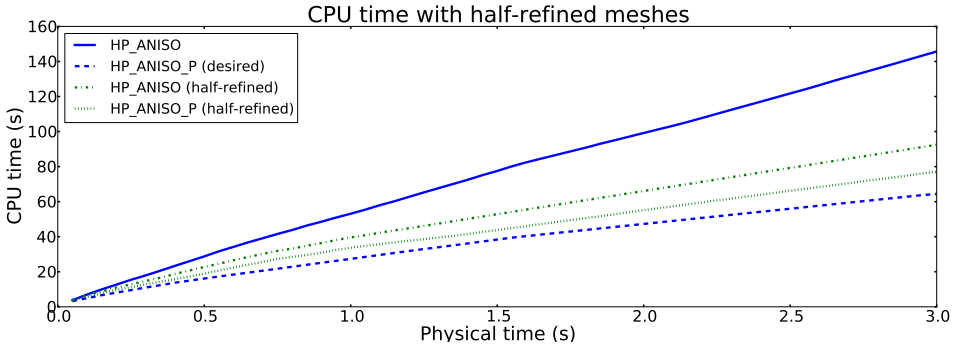

Figure 16: Cumulative CPU time for HP_ANISO and HP_ANISO_P with different initial meshes.

When it comes to a large domain or 3D modeling, the problem size becomes a very important factor. Namely, 3D full scale solutions tend to use a lot of memory. Therefore, we consider HP_ANISO the most suitable refinement mode for the given problem. Hence, some ways to optimize the HP_ANISO model to improve the CPU time factor without significantly compromising the NDOFs will be considered. The desirable output would be HP_ANISO problem size close to HP_ANISO_P CPU time. One way to optimize the problem is to choose more refined initial mesh. Another way would be to change the refinement frequency during the solving process. Theoretically it could result in a fewer adaptivity steps which are rather expensive in terms of CPU time. Recall that up to this point, the initial mesh was loaded in the beginning of each time step. True, by employing the optimizations, one must know something about the problem and its

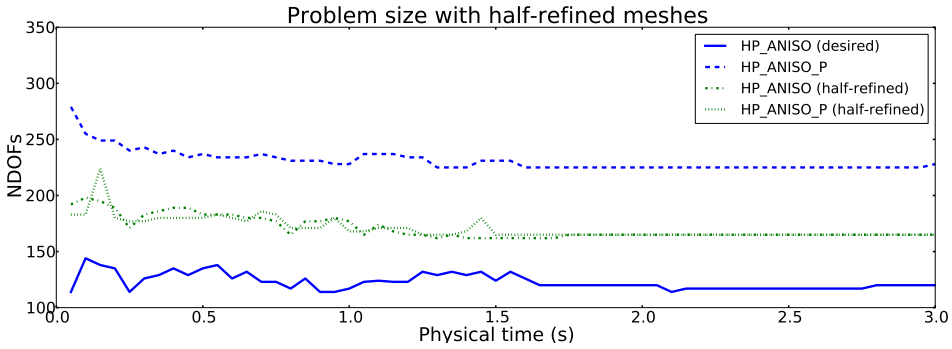

Figure 17: NDOFs at each time step for HP_ANISO and HP_ANISO_P with different meshes.

solution beforehand. However, it could still be practial when solving a real problem in a large domain.

The problem size and CPU time with HP_ANISO and HP_ANISO_P adaptivities on more refined initial mesh (see Fig. 6 (c)) compared to the coarse initial mesh (Fig. 6 (a)) and HP_ANISO_P solution are shown in Fig. 16 and Fig. 17. By using initially more refined mesh, the problem solving time can be reduced in case of HP_ANISO, at the same time, the problem size increases as the initial mesh is not necessarily the most optimal one. However, in this situation, HP_ANISO_P and HP_ANISO perform equally well.

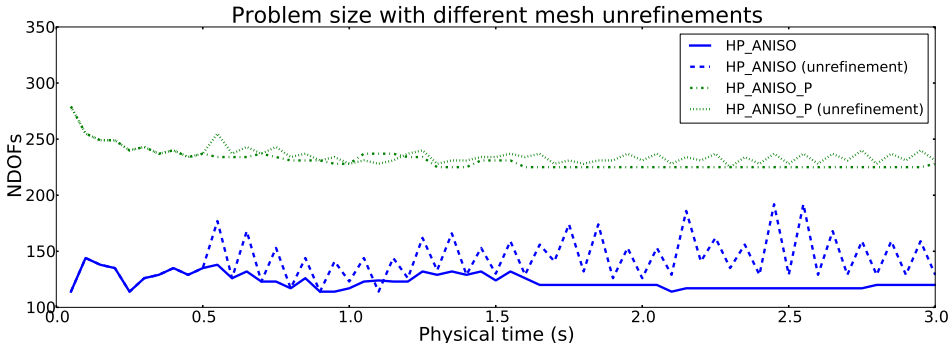

Figure 18: NDOFs at each time step for HP_ANISO and HP_ANISO_P with mesh unrefinement at each time step and at over each time step after first $0.5\ s$ physical solution time.

The next proposed optimization involved changing the unrefinement frequency. It is known that the concentration gradient ∇C changes the most in the initial phase of the calculation, therefore, the unrefinement after each time step was performed until $t = 0.5\ s$ (physical time). After that, the unrefinement was performed in $\Delta t = 0.10\ s$ interval. However, this optimization does not appear to result in a stable solution, i.e. the problem size does not remain steady, but starts to oscillate depending on the unrefinement frequency. This is shown in Fig. 18.

Therefore varying the unrefinement frequency will likely not result in desired results

in real applications for given system of equation. At the same time, by varying the initial mesh size, optimal initial mesh could be found for both HP_ANISO and HP_ANISO_P refinement modes.

5.5 More general results

Based on the results, cation concentration and voltage was calculated for different boundary conditions. For instance, when voltage is applied as follows

$$\phi_{\Omega_1} = 0.5\frac{x}{width_{\Omega_1}} + 0.5, \tag{5.1}$$

the concentration gradient ∇C and the voltage gradient $\nabla\phi$ are no longer effectively 1D. The calculated C and ϕ in Ω and corresponding meshes and polynomial degrees of the elements are shown in Fig. 19. HP_ANISO refinement mode was used. Notice that the solution is different to the one in Fig. **??** and the adapted mesh and the polynomial degrees are also more complicated than in Fig. 11. It must be noted that in case of non uniform boundary conditions which results in 2D problem, refined initial mesh was more efficient to use.

6 Conclusion and Outlook

In this work the system of Nernst-Planck-Poisson equations was solved using *hp*-finite element method with adaptive multimesh configuration. The weak form, residuals and the Jacobian matrix of the system were explicitly derived and implemented in Hermes *hp*-FEM adaptive solver. The solution for Nernst-Planck-Poisson problem with two field variables C and ϕ results in very different field gradients in the space and time. Conventional finite element solvers do not necessarily provide the means how to deal with such problems such that both the error of the solution and problem size remain small throughout the time dependent solving process. This is largely due to the fact that it is very difficult to find an optimal initial mesh.

It was shown that using time dependent adaptivity, multi-mesh configuration, and anisotropic *hp* refinements, the problem size remains acceptably small throughout the solving process. Namely, Hermes refinement modes HP_ANISO and HP_ANISO_P resulted the smallest and fastest problem, respectively. At the same time, relative error of the solution was known. Furthermore, using the multi-mesh configuration for the variables C and ϕ was justified — the adaptivity algorithm did not refine the mesh of ϕ nor did increase the polynomial degree throughout the adaptivity process. However, the mesh was significantly refined for C and also the maximum polynomial degree was varied in the range of 2…9. So it is efficient to use multi-mesh in terms of the number of degrees of freedom.

Conclusively, by using *hp*-FEM with adaptive multi-mesh configuration we can possibly reduce the problem size of the Nernst-Planck-Poisson equation system significantly

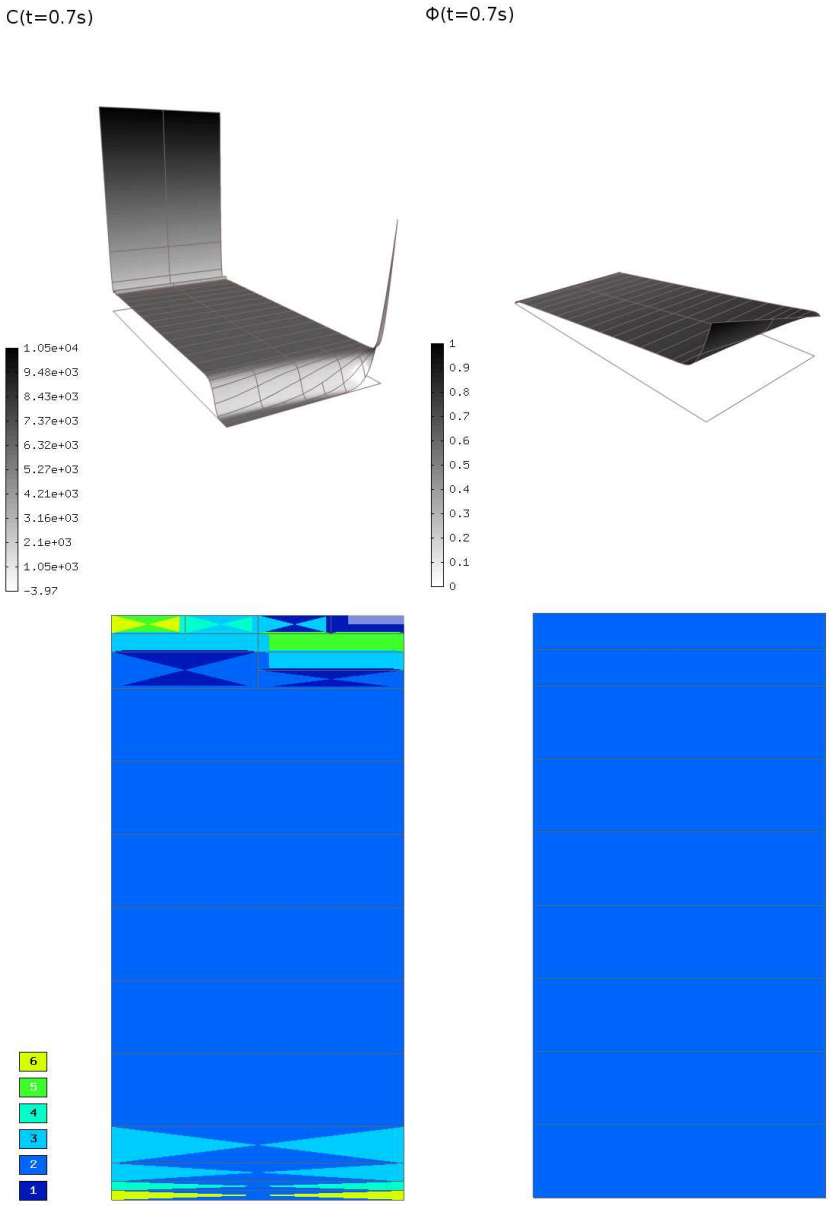

Figure 19: Solutions C and ϕ and corresponding polynomial degrees of the elements at $t = 0.7\ s$. HP_ANISO refinement mode was used. The height in the solution graphs indicates the value.

while still maintaining prescribed precision of the solution. We believe, and this is yet to be demonstrated, that this becomes especially important when dealing with 3D problems in a large physical domain with non-uniform boundary conditions.

Acknowledgments

The second author was partially supported by the Grant Agency of the Academy of Sciences of the Czech Republic under Grant No. IAA100760702, and by the U.S. Department of Energy Research Subcontract No. 00089911. The third author acknowledges the financial support of the U.S. Office of Naval Research under Award N000140910218.

References

[1] S. Basu and M.M. Sharma, *An improved space-charge model for flow through charged microporous membranes*, Journal of Membrane Science **124** (1997), no. 1, 77–91.

[2] L. Dubcova, P. Solin, J. Cerveny, and P. Kus, *Space and Time Adaptive Two-Mesh hp-Finite Element Method for Transient Microwave Heating Problems*, Electromagnetics **30** (2010), no. 1, 23–40.

[3] Sia Nemat-Nasser, *Micromechanics of actuation of ionic polymer-metal composites*, Journal of Applied Physics **92** (2002), no. 5, 2899–2915.

[4] K.M. Newbury and D.J. Leo, *Linear electromechanical model of ionic polymer transducers-Part I: Model Development*, Journal of Intelligent Material Systems and Structures **14** (2003), no. 6, 333.

[5] D. Pugal, K. Jung, A. Aabloo, and K.J. Kim, *Ionic polymer-metal composite mechanoelectrical transduction: review and perspectives*, Polymer international **59** (2010), no. 3, 279–289.

[6] Deivid Pugal, Kwang J. Kim, Andres Punning, Heiki Kasemagi, Maarja Kruusmaa, and Alvo Aabloo, *A self-oscillating ionic polymer-metal composite bending actuator*, Journal of Applied Physics **103** (2008), no. 8, 084908.

[7] Mohsen Shahinpoor and Kwang J Kim, *Ionic polymer-metal composites: I. fundamentals*, Smart Materials and Structures **10** (2001), no. 4, 819.

[8] P. Solin, D. Andrs, J. Cerveny, and M. Simko, *PDE-independent adaptive hp-FEM based on hierarchic extension of finite element spaces*, Journal of computational and applied mathematics **233** (2010), no. 12, 3086–3094.

[9] P. Solin, J. Cerveny, L. Dubcova, and D. Andrs, *Monolithic discretization of linear thermoelasticity problems via adaptive multimesh hp-FEM*, Journal of computational and applied mathematics **234** (2010), no. 7, 2350–2357.

[10] P. Solin, L. Dubcova, and I. Dolezel, *Adaptive hp-FEM with arbitrary-level hanging nodes for Maxwells equations*, Advances in Applied Mathematics and Mechanics **2** (2010), no. 4, 518–532.

[11] P. Solin, L. Dubcova, and J. Kruis, *Adaptive hp-FEM with dynamical meshes for transient heat and moisture transfer problems*, Journal of computational and applied mathematics **233** (2010), no. 12, 3103–3112.

[12] Thomas Wallmersperger, Donald J. Leo, and Curt S. Kothera, *Transport modeling in ionomeric polymer transducers and its relationship to electromechanical coupling*, Journal of Applied Physics **101** (2007), no. 2, 024912.