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

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

[a]*Mechanical Engineering Department, University of Nevada, Reno, NV, U.S.A.*
[b]*Department of Mathematics and Statistics, University of Nevada, Reno, NV, U.S.A.*
[c]*Institute of Thermomechanics, Prague, Czech Republic*
[d]*Institute of Technology, Tartu University, Estonia*

Abstract

In this paper we study a multiphysics coupled problem consisting of the Poisson and Nernst-Planck equations, that is used to model ionic polymer-metal composite (IPMC) materials. The problem is discretized using the finite element method (FEM) and solved using space-time adaptive hp-FEM algorithms provided by the Hermes library. Numerical results are presented and discussed, including comparison of various versions of adaptive hp-FEM and conventional low-order FEM. In particular, we discuss the performance of a novel adaptive multimesh hp-FEM algorithm.

Keywords: Ionic polymer-metal composites, IPMC, Nernst-Planck equation, Poisson equation, finite element method, space-time adaptivity, 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. 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).
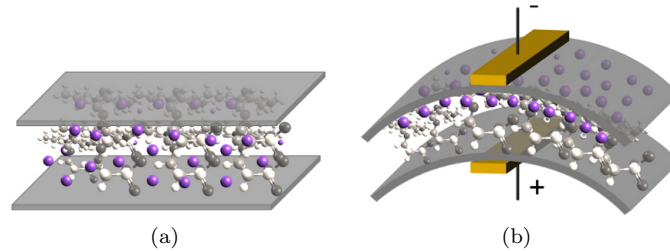

(a)　　　　　(b)

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).

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

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 [1, 2, 3, 4, 5, 6, 7].

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

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

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

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

where ε is the absolute dielectric permittivity. The charge density ρ is defined via

$$\rho = C - C_0 \tag{3}$$

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 each of them. This explains why we are interested in applying the novel multimesh hp-FEM method [ADD COUPLE OF REFERENCES TO SOLIN-MULTIMESH]. The PNP model is presented in Section 3 where also its weak formulation for the Newton's method is derived. Section 4 describes briefly the open source adaptive hp-FEM library Hermes 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 model to illustrate the principal difficulties encountered in the numerical solution of the PNP system. Table 1 shows constants that we will use in computations in this section as well as in the rest of the paper:

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. 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 extremely difficult. Even if the solution was stationary, an optimal mesh for C could never be optimal for ϕ and vice versa.
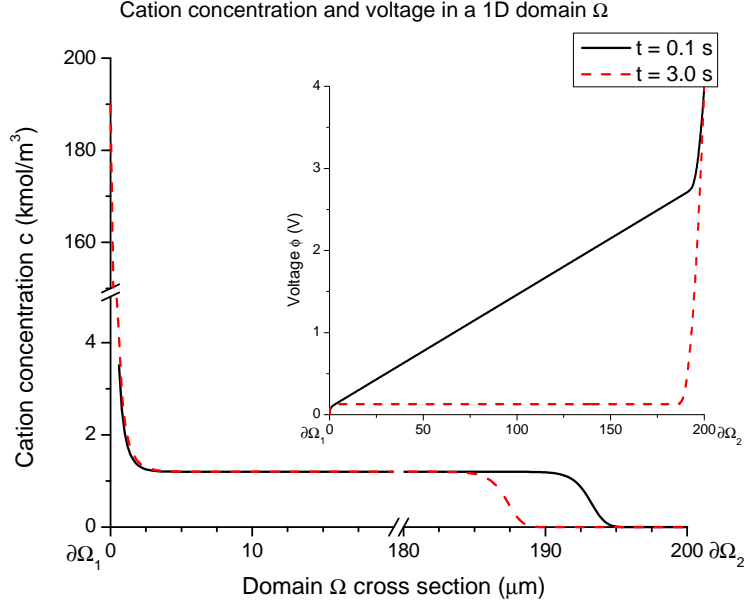
2

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 (2) and Neumann conditions to the Nernst-Planck equation (1).

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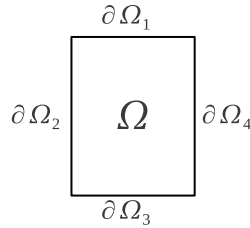
Figure 3: Calculation domain $\Omega \subset \mathbb{R}^2$ with boundaries $\partial\Omega_{1\ldots4} \subset \partial\Omega$.

As there is no flow through the domain's boundary, Eq. (1) is equipped with a Neumann boundary condition

$$-D\frac{\partial C}{\partial n} - z\mu FC\frac{\partial \phi}{\partial n} = 0. \tag{4}$$

3

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

$$\phi_{\partial\Omega_1} = V_{pos}, \tag{5}$$
$$\phi_{\partial\Omega_3} = 0. \tag{6}$$

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

$$\frac{\partial\phi_{\Omega_2}}{\partial n} = \frac{\partial\phi_{\Omega_4}}{\partial n} = 0. \tag{7}$$

3.1. Weak form of Poisson-Nernst-Planck system

To make our results easily reproducible, we present a rather detailed derivation of weak forms of Eqs. (1) and (2), as well as formulas for the Jacobian matrix and residual vector that are used in actual computations. To facilitate the derivation, we introduce two additional constants

$$K = z\mu F, \tag{8}$$
$$L = \frac{F}{\varepsilon}. \tag{9}$$

After using Eq. (3) and the constants K and L, Eq. (1) and Eq. (2) become

$$\frac{\partial C}{\partial t} - D\nabla^2 C - K\nabla \cdot (C\nabla\phi) = 0, \tag{10}$$
$$-\nabla^2\phi - L(C - C_0) = 0. \tag{11}$$

The boundary condition Eq. (4) now has the form

$$-D\frac{\partial C}{\partial n} - KC\frac{\partial\phi}{\partial n} = 0. \tag{12}$$

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. (10), we first multiply it with a test function $v^C \in V$ and integrate over the domain Ω,

$$\int_\Omega \frac{\partial C}{\partial t}v^C d\mathbf{x} - \int_\Omega D\nabla^2 C v^C d\mathbf{x} - \int_\Omega K\nabla C \cdot \nabla\phi v^C d\mathbf{x} - \int_\Omega KC\nabla^2\phi v^C d\mathbf{x} = 0. \tag{13}$$

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

$$\int_\Omega \frac{\partial C}{\partial t}v^C d\mathbf{x} + D\int_\Omega \nabla C \cdot \nabla v^C d\mathbf{x} - K\int_\Omega \nabla C \cdot \nabla\phi v^C d\mathbf{x} + K\int_\Omega \nabla\left(Cv^C\right) \cdot \nabla\phi d\mathbf{x} \tag{14}$$
$$-D\int_{\partial\Omega} \frac{\partial C}{\partial n}v^C d\mathbf{S} - \int_{\partial\Omega} K\frac{\partial\phi}{\partial n}Cv^C d\mathbf{S} = 0. \tag{15}$$

Expanding the nonlinear term and using the boundary condition (12), we have

$$\int_\Omega \frac{\partial C}{\partial t}v^C d\mathbf{x} + D\int_\Omega \nabla C \cdot \nabla v^C d\mathbf{x} - K\int_\Omega \nabla C \cdot \nabla\phi v^C d\mathbf{x} + K\int_\Omega \nabla\phi \cdot \nabla C v^C d\mathbf{x} + K\int_\Omega C\left(\nabla\phi \cdot \nabla v^C\right) d\mathbf{x} = 0. \tag{16}$$

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

$$\int_\Omega \frac{\partial C}{\partial t}v^C d\mathbf{x} + D\int_\Omega \nabla C \cdot \nabla v^C d\mathbf{x} + K\int_\Omega C\left(\nabla\phi \cdot \nabla v^C\right) d\mathbf{x} = 0. \tag{17}$$

4

Analogously we derive also the weak form of the Poisson equation (11),

$$-\int_\Omega \nabla^2\phi v^\phi d\mathbf{x} - \int_\Omega LCv^\phi d\mathbf{x} + \int_\Omega LC_0 v^\phi d\mathbf{x} = 0. \tag{18}$$

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

$$\int_\Omega \nabla\phi \cdot \nabla v^\phi d\mathbf{x} - \int_\Omega LCv^\phi d\mathbf{x} + \int_\Omega LC_0 v^\phi d\mathbf{x} = 0. \tag{19}$$

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

To employ the Newton's method for the nonlinear system (17), (19), 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,

$$C^{n+1} = C(Y^{n+1}) = \sum_{k=1}^{N^C} y_k^C v_k^C, \tag{20}$$

$$\phi^{n+1} = \phi(Y^{n+1}) = \sum_{k=1}^{N^\phi} y_k^\phi v_k^\phi. \tag{21}$$

Here Y^{n+1} is a coefficient vector of length $N^C + N^\phi$ 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 (20), (21), the time discretized Eq. (17) leads to the formula for the first part F^C of the residual vector F,

$$\begin{aligned}
F_i^C(Y) &= \int_\Omega \frac{C(Y)}{\tau} v_i^C d\mathbf{x} - \int_\Omega \frac{C^n}{\tau} v_i^C d\mathbf{x} \\
&+ \frac{1}{2}\left[D\int_\Omega \nabla C(Y) \cdot \nabla v_i^C d\mathbf{x} + D\int_\Omega \nabla C^n \cdot \nabla v_i^C d\mathbf{x} \right] \\
&+ \frac{1}{2}\left[K\int_\Omega C(Y)\left(\nabla\phi(Y)\cdot\nabla v_i^C\right) d\mathbf{x} + K\int_\Omega C^n\left(\nabla\phi^n\cdot\nabla v_i^C\right) d\mathbf{x} \right]
\end{aligned} \tag{22}$$

where $i = 1, 2, \ldots, N^C$. Analogously, Eq. (19) defines the second part F^ϕ of the residual vector F,

$$F_i^\phi(Y) = \int_\Omega \nabla\phi(Y)\cdot\nabla v_i^\phi d\mathbf{x} - \int_\Omega LC(Y)v_i^\phi d\mathbf{x} + \int_\Omega LC_0 v_i^\phi d\mathbf{x} \tag{23}$$

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,

$$J(Y) = \begin{pmatrix} \dfrac{\partial F_i^C}{\partial y_j^C} & \dfrac{\partial F_i^C}{\partial y_j^\phi} \\ \dfrac{\partial F_i^\phi}{\partial y_j^C} & \dfrac{\partial F_i^\phi}{\partial y_j^\phi} \end{pmatrix}, \tag{24}$$

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

$$\frac{\partial F_i^C}{\partial y_j^C}(Y) = \int_\Omega \frac{1}{\tau} v_j^C v_i^C d\mathbf{x} + \frac{1}{2} D \int_\Omega \nabla v_j^C \cdot \nabla v_i^C d\mathbf{x} + \frac{1}{2} K \int_\Omega v_j^C \left(\nabla \phi(Y) \cdot \nabla v_i^C \right) d\mathbf{x}, \qquad (25)$$

$$\frac{\partial F_i^C}{\partial y_j^\phi}(Y) = \frac{1}{2} K \int_\Omega C(Y) \left(\nabla v_j^\phi \cdot \nabla v_i^C \right) d\mathbf{x}, \qquad (26)$$

$$\frac{\partial F_i^\phi}{\partial y_j^C}(Y) = -\int_\Omega L v_j^C v_i^\phi d\mathbf{x}, \qquad (27)$$

$$\frac{\partial F_i^\phi}{\partial y_j^\phi}(Y) = \int_\Omega \nabla v_j^\phi \cdot \nabla v_i^\phi d\mathbf{x}. \qquad (28)$$

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 $Y^{n+1} := Y_{k+1}^{n+1}$. 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 a traditional low-order FEM, 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, error indicators based on steep gradients, etc are employed. Unfortunately, none of these approaches are suitable for real-life 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 moreover only for low-order finite elements (such as piecewise linear approximations). These heuristic techniques may fail in more complicated situations, and moreover they lack a transparent relation to the true approximation error.

In order to obtain fast, usable adaptivity, one has to resort to higher-order finite elements and adaptive hp-FEM. Automatic adaptivity in the hp-FEM is substantially different from adaptivity in low-order FEM, since every element can be refined in many different ways. For illustration, Fig. 4 shows several illustrative refinement candidates for a fourth-order element.
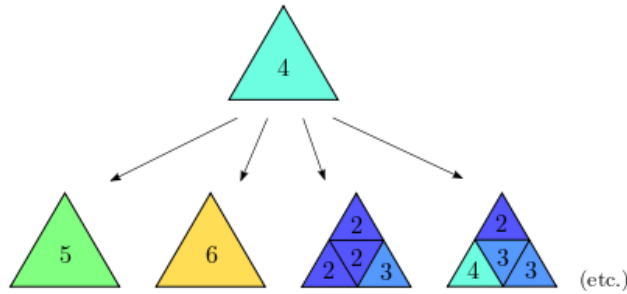
Figure 4: Many possible refinement candidates for a fourth-order element.

The number of possible element refinements is implementation dependent. In general it is very low in h or p adaptivity, and higher in hp-adaptivity. It rises very fast when anisotropic refinements are considered.

4.1. 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. 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 (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 / v, h / v	h / v, h / v			h, v
H_ANISO	h / v, h / v	h / v, h / 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	$\tfrac{1}{2}h+\delta_3$, $\tfrac{1}{2}h+\delta_2$ / $\tfrac{1}{2}v+\delta_3$, $\tfrac{1}{2}v+\delta_2$	$\tfrac{1}{2}h+\delta_0$, $\tfrac{1}{2}h+\delta_1$ / $\tfrac{1}{2}v+\delta_0$, $\tfrac{1}{2}v+\delta_1$			$h+\delta_0$, $v+\delta_0$
HP_ANISO_H	$\tfrac{1}{2}h+\delta_3$, $\tfrac{1}{2}h+\delta_2$ / $\tfrac{1}{2}v+\delta_3$, $\tfrac{1}{2}v+\delta_2$	$\tfrac{1}{2}h+\delta_0$, $\tfrac{1}{2}h+\delta_1$ / $\tfrac{1}{2}v+\delta_0$, $\tfrac{1}{2}v+\delta_1$	$h+\delta_1$ / $\tfrac{1}{2}v+\delta_1$, $h+\delta_0$ / $\tfrac{1}{2}v+\delta_0$	$\tfrac{1}{2}h+\delta_0$, $\tfrac{1}{2}h+\delta_1$ / $v+\delta_0$, $v+\delta_1$	$h+\delta_0$, $v+\delta_0$
HP_ANISO_P	$\tfrac{1}{2}h+\alpha_3$, $\tfrac{1}{2}h+\alpha_2$ / $\tfrac{1}{2}v+\beta_3$, $\tfrac{1}{2}v+\beta_2$	$\tfrac{1}{2}h+\alpha_0$, $\tfrac{1}{2}h+\alpha_1$ / $\tfrac{1}{2}v+\beta_0$, $\tfrac{1}{2}v+\beta_1$			$h+\alpha_0$, $v+\beta_0$
HP_ANISO	$\tfrac{1}{2}h+\alpha_3$, $\tfrac{1}{2}h+\alpha_2$ / $\tfrac{1}{2}v+\beta_3$, $\tfrac{1}{2}v+\beta_2$	$\tfrac{1}{2}h+\alpha_0$, $\tfrac{1}{2}h+\alpha_1$ / $\tfrac{1}{2}v+\beta_0$, $\tfrac{1}{2}v+\beta_1$	$h+\alpha_1$ / $\tfrac{1}{2}v+\beta_1$, $h+\alpha_0$ / $\tfrac{1}{2}v+\beta_0$	$\tfrac{1}{2}h+\alpha_0$, $\tfrac{1}{2}h+\alpha_1$ / $v+\beta_0$, $v+\beta_1$	$h+\alpha_0$, $v+\beta_0$

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

---

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

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 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 [ADD CITATION SOLIN-SIMKO]. 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 [ADD REFERENCES] 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 sub-elements 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

Calculation for all the refinement modes were performed in both single-mesh and multi-mesh configurations. The following numerical results were recorded for each refinement mode: converged relative error, cumulative CPU time, and the problem size in terms of number of degrees of freedoms (NDOFs) at each time step. Two types of initial meshes were used — in case of only p-adaptivity, more refined mesh was used (Fig. 6 (c)) to ensure the error convergence. When the element size refinement was also enabled (all h/hp refinement modes), very coarse initial mesh was used (see Fig. 6 (a)) to let the adaptivity algorithm find the most optimal mesh. So coarse initial mesh might not be suitable for all the practical applications, as will be demonstrated in the end of this section, however, it provides a good insight into the adaptivity performance of Hermes. In both cases, the initial mesh was loaded at each time step of the calculation.

Example of the solution at $t = 0.7\ s$ and $t = 3.0\ s$ with refinement mode HP_ANISO is shown in Fig. 7. The time $t = 0.7\ s$ was chosen because by that time, some ionic migration has already taken place, i.e. concentration gradient near the $\partial \Omega_1$ and $\partial \Omega_3$ has formed. The automatic mesh refinements at different time steps are clearly visible in the figure — especially near the top boundary, where the concentration gradient is moving in time (as was also seen in Fig. 2).

The following subsections provide a detailed comparison of the different refinement modes and try to estimate the most suitable one for the given problem.
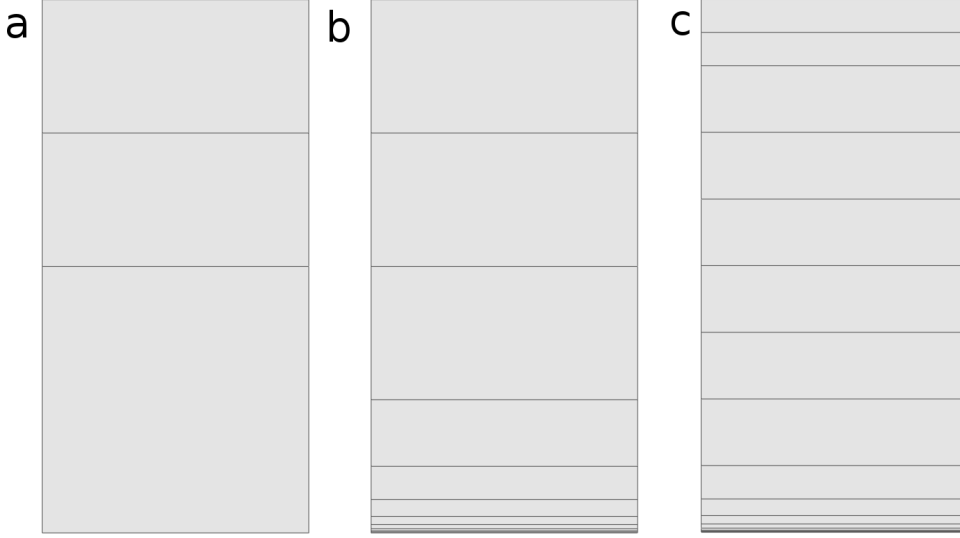
Figure 6: Initial coarse mesh (a), half refined mesh (b) and refined mesh (c). The coarse mesh and refined mesh were used in the initial calculations, the latter one in case of p-adaptivity (including HP_ANISO_P). The half-refined mesh was used later to optimize the hp-adaptive refinement solutions.

5.1. Optimal refinement modes

Running the simulation with different refinement modes and meshes showed that the multi-mesh configuration generally results in a smaller problem, faster calculation, and better or similar error convergence compared to the single-mesh configuration. This is well illustrated in Fig. 8 and Fig. 9. The result can be understood from Fig. 2 — in the boundary regions $\partial\Omega_1$ and $\partial\Omega_3$ the concentration gradient is greater than voltage gradient, namely $\nabla C >> \nabla\phi$. Therefore refining the mesh for both variables is not reasonable in terms of number of degrees of freedom. For instance, the solution space with corresponding mesh in case of HP_ANISO at $t = 0.7$ s is shown in Fig. 7. The corresponding polynomial degree space is shown in Fig. 10. Notice that the adaptive algorithm has increased the maximum polynomial degree for C space to 7, however, at the same time, the maximum polynomial degree for the ϕ space is one. Furthermore, the mesh is significantly more refined for C. For most cases using the multi-mesh results in similar or better CPU time, i.e. multi-mesh takes less computational resources. Only excpetion was HP_ANISO_H refinement mode for which the single mesh configuration resulted slightly faster calculation time. Based on the results, only multi-mesh configurations will be considered in the following comparisons.

To narrow down the list of refinement modes for the given problem, first the isotropic and anisotropic adaptivities were compared. The p-adaptivity modes were compared separately as they used more refined mesh (Fig. 6 (c)). Fig. 11 and Fig. 12 show the comparison of H_ISO, H_ANISO, HP_ISO, HP_ANISO, and HP_ANISO_H modes in terms of CPU time and problem size. Fig. 13 and Fig. 14 show the similar comparsion for the P_ISO, P_ANISO, and HP_ANISO_P modes. It can be clearly seen that the anisotropic refinement modes result in a reasonable problem size and the problems solve within a reasonable calculation time. It is interesting to note that HP_ANISO results in the smallest problem size. Also, in p-adaptivity group, HP_ANISO_P results in the smallest problem size at each time step, whereas P_ISO and P_ANISO have a very largy problem size during the first time steps of the solution. Here the term "reasonable problem size" means that the number of degrees of freedom in time converges to so that $N_{dof} < 500$, and the term "reasonable calculation time" means that the calculation (step $\tau = 0.01$ s, physical time $t_{end} = 3.0$ s) time t on a givem system was $t < 500$ s. Although these parameters are empirical, they serve as an upper limit, given that the most refinement modes give significantly smaller results: $t << 500$ s and $N_{dof} << 500$.

C(t=0.7s)   Φ(t=0.7s)

C(t=3.0s)   Φ(t=3.0s)
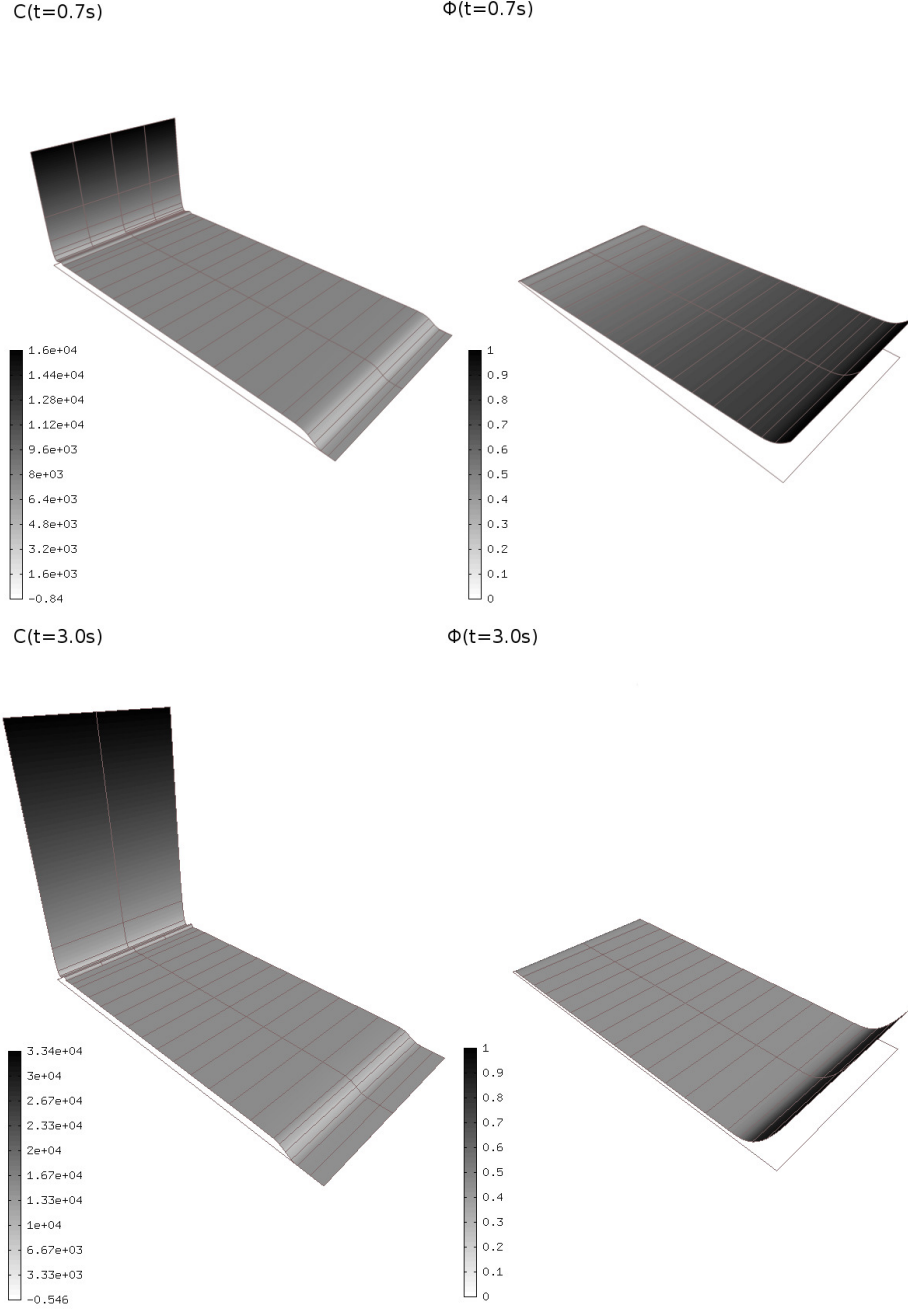
Figure 7: Concentration C and voltage ϕ at two different time steps (HP_ANISO adaptivity was used). This is a 2D solution shown as in 3D, where the height indicates the values of C and ϕ.

5.2. Quantitative analysis of the refinement modes

Based on the results in the previous subsection, H_ANISO, HP_ANISO, and HP_ANISO_H from the hp/h-adaptivity group and HP_ANISO_P from p-adaptivity group will be compared in terms of problem size and cumulative CPU time. In all of the cases, the relative error at each time step remained below the threshold which were set to $e_{th} = 0.5\%$ between the coarse mesh and fine mesh solutions, therefore the error-time plot will not be considered.
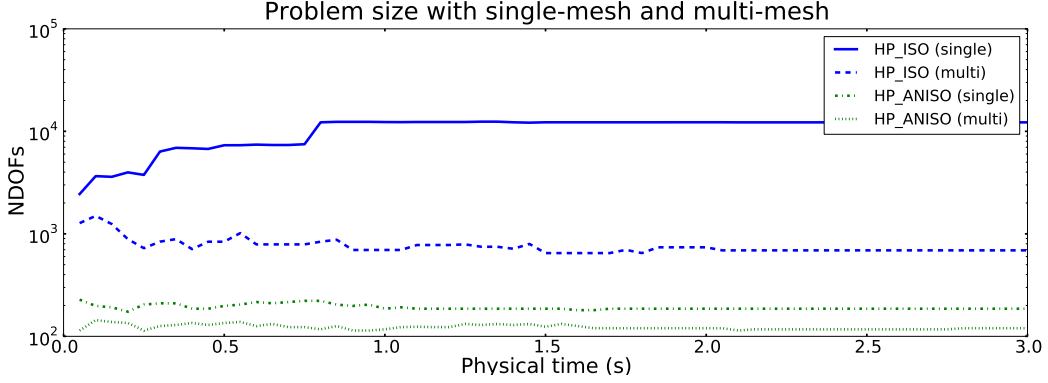
Figure 8: NDOFs in case of single-mesh and multi-mesh solutions with HP_ISO and HP_ANISO refinements. (Notice log Y scale)
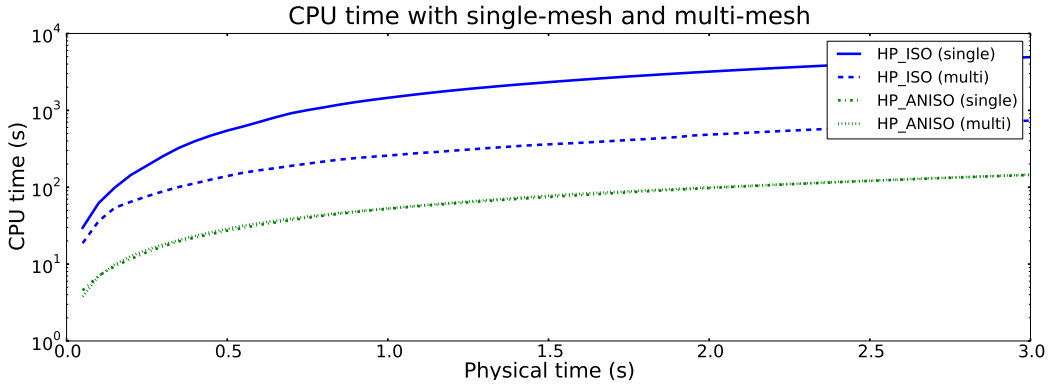
Figure 9: CPU times in case of single-mesh and multi-mesh solutions with HP_ISO and HP_ANISO refinements. (Notice log Y scale)

Fig. 15 shows the cumulative CPU time for different refinement modes at each time step. All the calculations were done on the same computer. Here we see that HP_ANISO_H and HP_ANISO require the most resources which can be understood from the fact that these refinement modes have the largest number of candidates (see Section. 3) from which the refinement method is chosen. At the same time, HP_ANISO_P is the fastest among all the other refinement modes. Fig. 16 shows the NDOFs at each time step. It can be seen that the HP_ANISO results in the smallest problem size — $N_{dof} \approx 125$. All the other refinement modes result in a problem size of approximately ($N_{dof} \approx 250$). So we have two notable refinement modes — HP_ANISO_P because of the fast solution, and HP_ANISO because of the small problem size. In the following subsection, optimization of those two modes will be considered.

5.3. Optimizations of HP_ANISO and HP_ANISO_P refinement modes

So far we have seen that HP_ANISO results in the smallest problem size, but requires quite a lot CPU time compared to HP_ANISO_P that is the fastest. When it comes to a large domain or 3D modeling the problem size becomes the most important factor. Therefore, we consider HP_ANISO the most suitable refinement mode to the given problem. Thus some ways to optimize the time factor 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 somewhat more refined initial mesh. Other way one could think to optimize the problem would be to change the refinement frequency during the solving process. Recall that up to this point, the initial mesh has been loaded in the beginning of each time step. True, by employing the
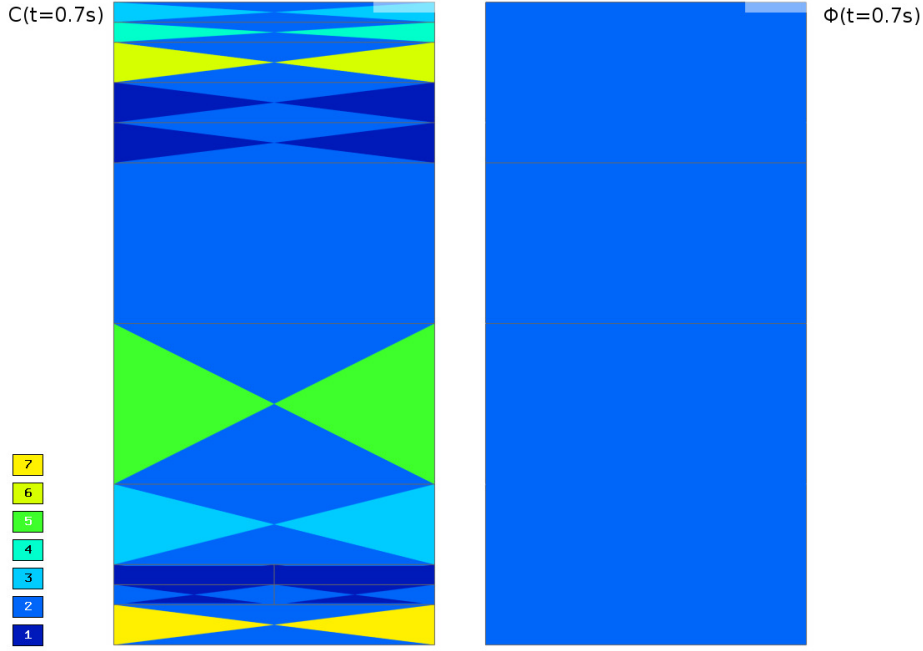
Figure 10: Polynomial degree space for C and ϕ at $t = 0.7$ s The color indicates the maximum polynomial degree of the corresponding element.
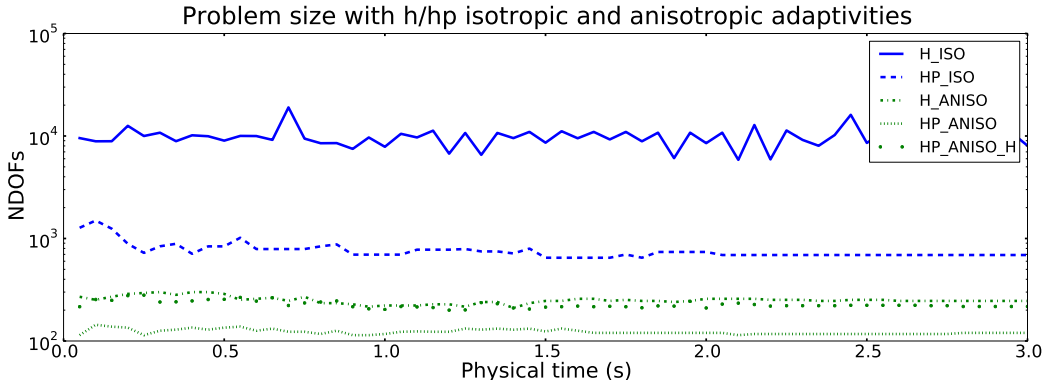
Figure 11: NDOFs in case of multi-mesh solutions with H_ISO, H_ANISO, HP_ISO, HP_ANISO, and single-mesh solution with HP_ANISO_H refinement modes. (Notice log Y scale)

optimizations, we already must know something about the problem and its 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 is shown in Fig. 17 and Fig. 18. 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. In this situation, HP_ANISO_P and HP_ANISO perform equally well.

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

12
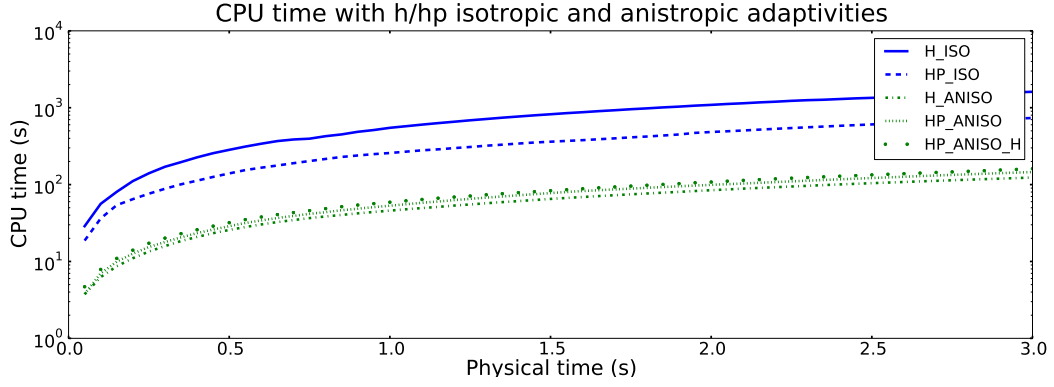
Figure 12: CPU times in case of multi-mesh solutions with H_ISO, H_ANISO, HP_ISO, HP_ANISO, and single-mesh solution with HP_ANISO_H refinement modes. (notice log Y scale)
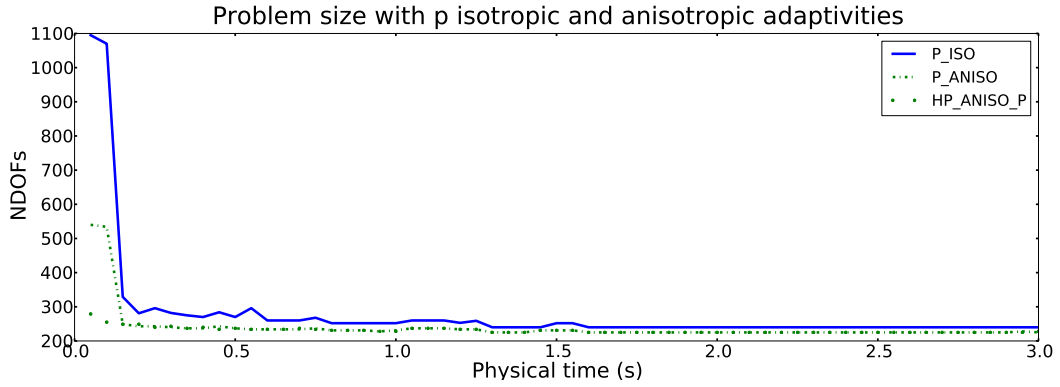
Figure 13: NDOFs in case of multi-mesh solutions with P_ISO, P_ANISO, and HP_ANISO_P refinement modes.
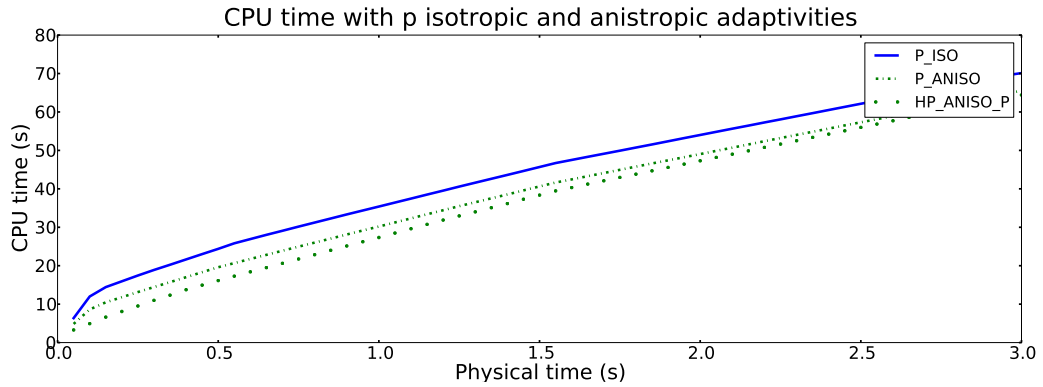
Figure 14: CPU times in case of multi-mesh solutions with P_ISO, P_ANISO, and HP_ANISO_P refinement modes.

performed in $\Delta t = 0.10$ s interval. However, this optimization did not 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. 19.

13
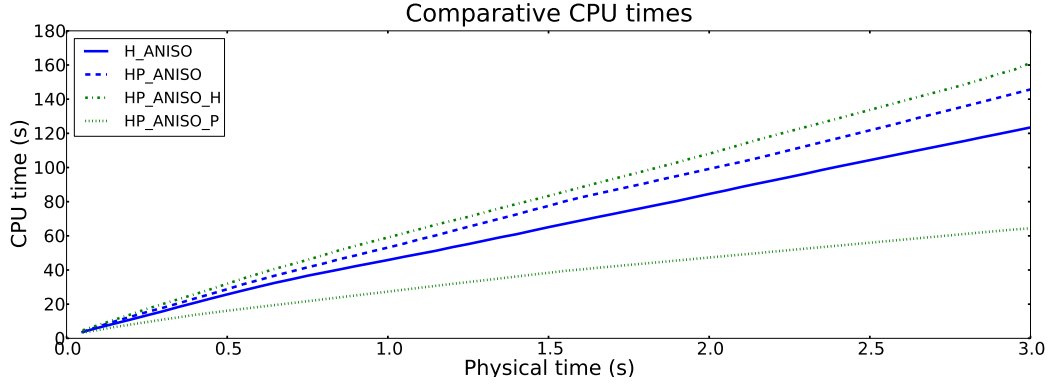
Figure 15: Comparative CPU time for different refinement modes.
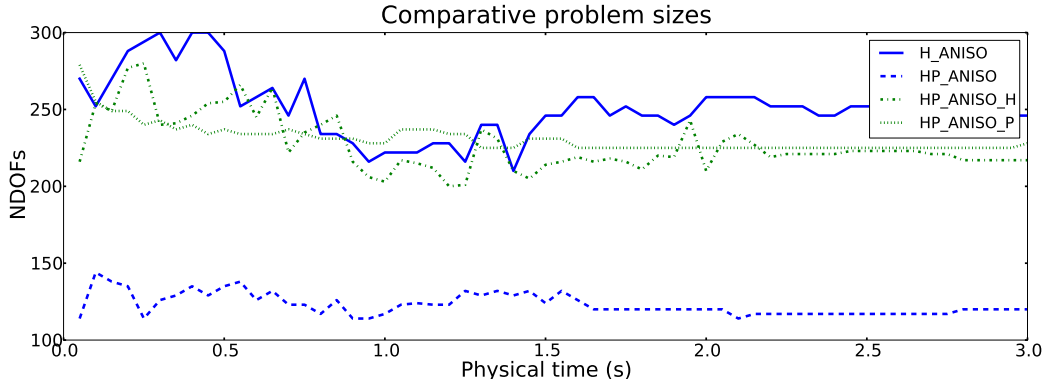
Figure 16: Comparative NDOFs at each time step for different refinement modes.
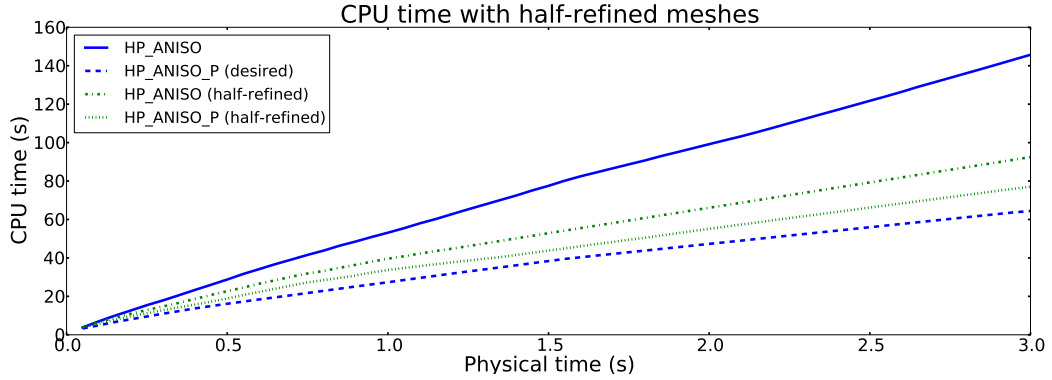
Figure 17: Cumulative CPU time for HP_ANISO and HP_ANISO_P with different initial meshes.

Therefore varying the unrefinement frequency will not likely result in desired results in real applications for given system of equation. At the same time, by varying a mesh size, optimal initial mesh could be found for both HP_ANISO and HP_ANISO_P refinement modes.

14
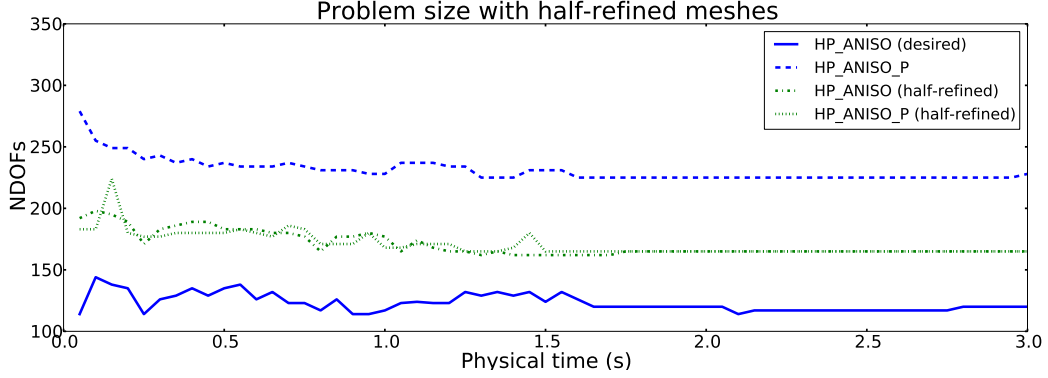
Figure 18: NDOFs at each time step for HP_ANISO and HP_ANISO_P with different meshes.
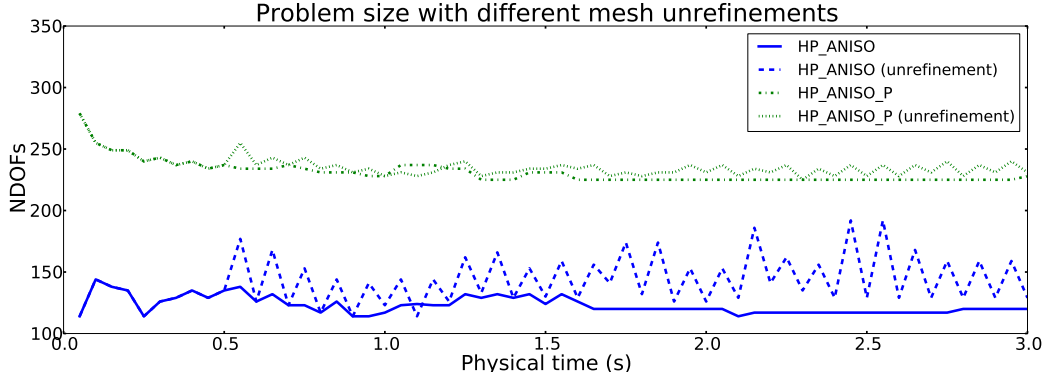
Figure 19: NDOFs at each time step for HP_ANISO and HP_ANISO_P with mesh unrefinement at each time step and at over each time step after first 0.5 s physical solution time.

5.4. 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

$$\phi_{\Omega_1} = 0.5 \frac{x}{width_{\Omega_1}} + 0.5, \tag{29}$$

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. 20. HP_ANISO refinement mode was used. Notice that the solution is different to the one in Fig. 7 and the adapted mesh and the polynomial degrees are also more complicated than in Fig. 10. 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 provide the means how to deal with such problems such

C(t=0.7s)   Φ(t=0.7s)
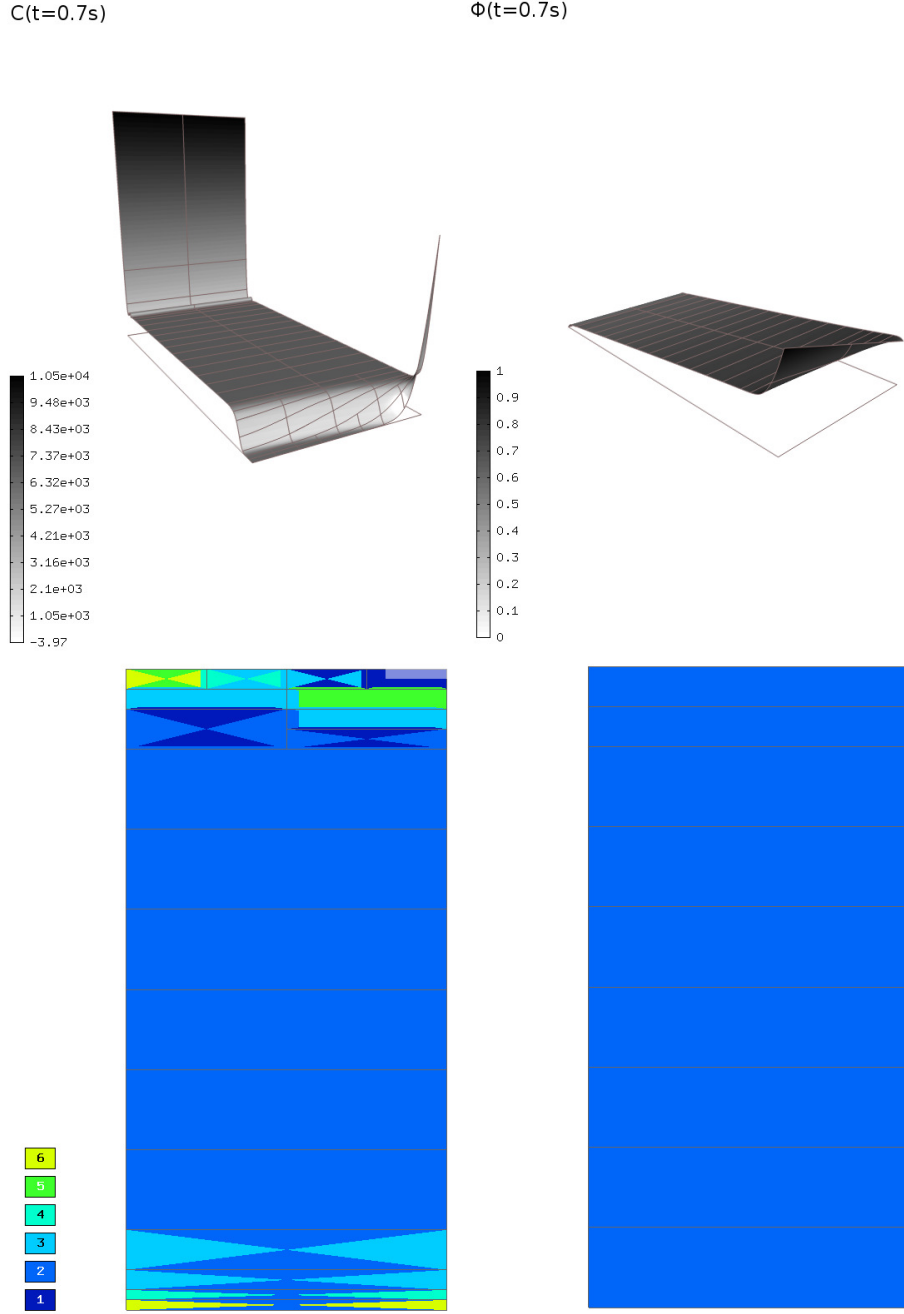
Figure 20: Solutions C and ϕ and corresponding polynomial degrees of the elements at $t = 0.7\ s$. HP_ANISO refinement mode was used. The height in the solution graphs indicates the value.

that both the error of the solution and problem size remain small throughout the time dependent solving process.

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

16

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 $2 \ldots 9$. 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 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 with a large physical domain.

Acknowledgments

[1] S. Basu, M. Sharma, An improved space-charge model for flow through charged microporous membranes, Journal of Membrane Science 124 (1) (1997) 77–91.

[2] M. Shahinpoor, K. J. Kim, Ionic polymer-metal composites: I. fundamentals, Smart Materials and Structures 10 (4) (2001) 819.

[3] S. Nemat-Nasser, Micromechanics of actuation of ionic polymer-metal composites, Journal of Applied Physics 92 (5) (2002) 2899–2915.

[4] K. Newbury, D. Leo, Linear electromechanical model of ionic polymer transducers-Part I: Model Development, Journal of Intelligent Material Systems and Structures 14 (6) (2003) 333.

[5] 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 (2) (2007) 024912.

[6] 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 (8) (2008) 084908.

[7] D. Pugal, K. Jung, A. Aabloo, K. Kim, Ionic polymer-metal composite mechanoelectrical transduction: review and perspectives, Polymer international 59 (3) (2010) 279–289.