NUMERICAL OPTIMIZATION ALGORITHMS

INNOCENT NIYONZIMA*

Abstract. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Key words. Numerical modeling, finite element method, topology optimization, shape optimization, level set method, computational electromagnetics.

MSC codes. 35K55

2. Introduction. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

*Univ. Grenoble Alpes, CNRS, Grenoble INP, G2ELab, F-38000 Grenoble, France, innocent.niyonzima@univ-grenoble-alpes.fr.

Algorithm 2.1 Pseudocode for the density optimization code (to be modified)

INPUT: The mesh, initial and boundary conditions.
OUTPUT: Optimized design.

procedure STREAMER CODE

1. Define parameters used for the resolution and for mesh refinement.

2. Read the mesh and perform uniform refinement (if needed).

3. Choose time steppers for n_e and n_i.

4. Generate initial refined mesh using the initial value n_{i0}.

5. Define FS : $H^1(\Omega_{\text{ele}} : \mathbb{R})$, $H^1(\Omega_{\text{n}_i} : \mathbb{R})$, $H^1(\Omega_{\text{n}_e} : \mathbb{R})$, $\boldsymbol{H}(\mathbf{curl}; \Omega_{\text{ele}} : \mathbb{R}^3)$ and $L^2(\Omega_{\text{ele}} : \mathbb{R})$.

6. Define BCs, the BilinearForm and the LinearForm **for the electrostatic problem**.

7. Define ICs for n_e and n_i, and instantiate *AdvectionDiffusionOperators* objects used for the resolution of ADR equations. For a given ADR PDE:

$$\partial_t n = \operatorname{div}\Big(\mu(E)\,\boldsymbol{E}\,n + D(E)\,\mathbf{grad}\,n\Big) + \bar{\alpha}(E)n,$$

this object represents thr RHS of the discretized ODE :

$$\frac{dn_h}{dt} = \boldsymbol{M}^{-1}\boldsymbol{f}(t, n_h),$$

9. Define the list of estimators, the **ThreshHoldRefiner** and the **ThreshHoldDerefiner**.

10. Define DataCollections (ParaViewDataCollection and GmshDataCollection).

11. Initialize OdeSolvers for n_e and n_i.

12. **Time loop**.

for ($n \leftarrow 1$ To N_{TS}) **do** ▷ *the time loop*

12.1. Set Time and ΔTime.

12.2. Refiner.Reset() and Derefiner.Reset().

12.3. **Refinement loop**.

for ($l \leftarrow 1$ To N_{Ref}) **do** ▷ *The refinement loop*

12.3.1. **Solve the electrostatic problem** (compute total electric charge, compute the BilinearForm and the LinearForm, Linear solve, RecoverSolution, saveSolutions).

12.3.2. Apply the refiner $\longrightarrow$ compute errors. **If Refiner.Stop()** $\longrightarrow$ exit the refinement loop, **else**, update objects (FunctionSpace, BilinearForms LineaForms, Vectors, ...)

end for

12.4. **Derefinement** $\longrightarrow$ update objects.

12.5. Solve the ADR equations for n_e and n_i (project $\mathbf{grad}\,v$ onto $\boldsymbol{H}(\mathbf{curl}; \Omega_{\text{ele}} : \mathbb{R}^3)$, update parameters used in the ADR for $n_e \longrightarrow$ solve for n_e, update parameters used in the ADR for $n_i \longrightarrow$ solve for n_i, compute $n_i - n_e$ and $\bar{\alpha}\mu_e(E)En_e$ and use them in the error estimators, Save solutions).

end for

end procedure