Battery Computing

An MMSC Case Study on Scientific Computing
Candidate Number: 1072462

Abstract

This work shall attempt to

Our Goal: Numerically obtain the solution $\{a(x,T), b(x,T)\}$ of

$$\begin{cases} \frac{\partial a}{\partial t} = D_a \frac{\partial^2 a}{\partial x^2}, & a : \mathbb{R}^+ \times [0,T] \mapsto [0,1],\ T \in \mathbb{R}^+,\ D_a \in \mathbb{R}^+, \\ \frac{\partial b}{\partial t} = D_b \frac{\partial^2 b}{\partial x^2}, & b : \mathbb{R}^+ \times [0,T] \mapsto [0,1],\ D_b \in \mathbb{R}^+, \\ a(\infty,t) = 1,\ b(\infty,t) = 0 & \forall t \in [0,T] \\ a(x,0) = 1,\ \ b(x,0) = 0 & \forall x \in (0,\infty) \\ a(0,t) = 0,\ \frac{\partial a}{\partial x} + D\frac{\partial b}{\partial x} = 0 & \end{cases}$$

The implementation bla bla

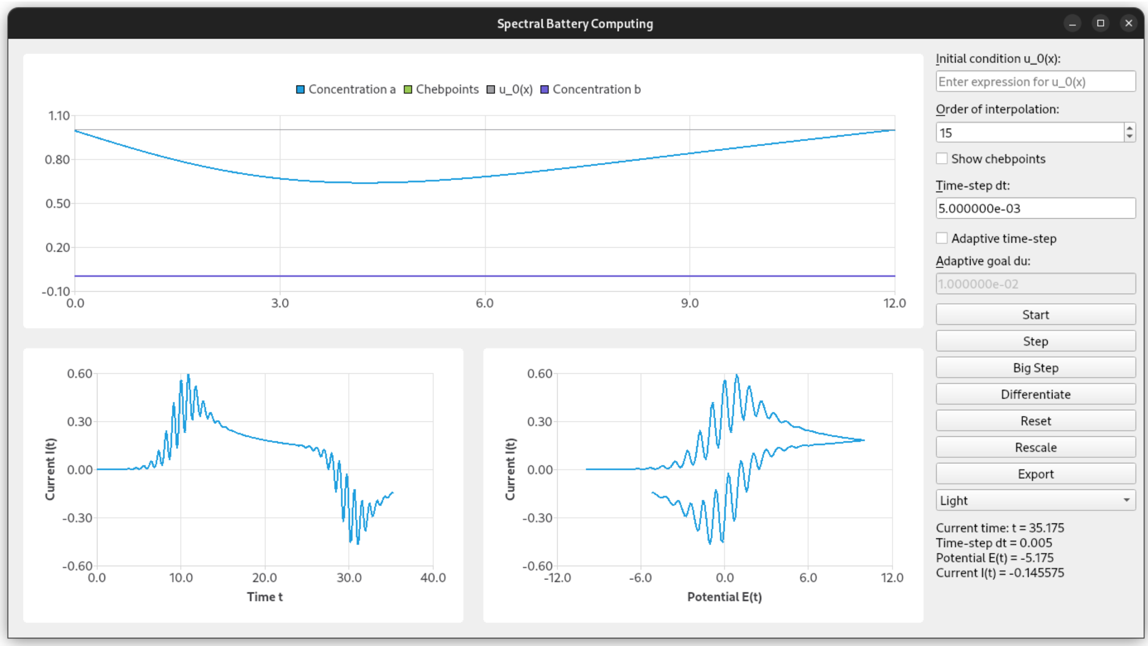

Figure 1: Graphical User Interface

Contents

1 Problem Introduction

Energy storage and its associated challenges are clearly among the most relevant questions, not only for the industrial but also the private sector. Politically, many nations in the world are steering towards greener energy supplies. Renewable energy sources such as wind and sun usually have a fundamental issue however, their availability is subject to an immense amount of fluctuation, which the energy grid must compensate through short- and long-term energy storage.

Long-term solutions include for example pumped-storage hydroelectricity facilities, but these must be complemented with short-term storage approaches such as Lithium-Ion or Lithium-Iron-Phosphate ($LiFePO_4$) batteries. Most modern batteries exploit electrochemical reactions to relate electrical potentials with chemical potentials and their associated difference ($\rightarrow$ voltage). The oxidation reaction we considere here is

$$A \Rightarrow B + e^{-}?$$

where A and B can be any chemicals and e^{-} is an electron.

More bla bla later on.

Figure

Figure 2: Wohoo

1.1 Chronoamperometry

1.2 DC Voltammetry

1.3 AC Voltammetry

2 Mathematical Background

Let $\mathbb{N}$ denote the nonnegative integers, so $0 \in \mathbb{N}$. Similarly, let $\mathbb{R}^{+} = [0, \infty)$ denote the nonnegative real numbers. Figure 2.

2.1 Laplace Integral Transform

What is Laplace?

Proof for Laplace's differentiation theorem.

2.2 Chebyshev Polynomials

Proof of $U_k(-1)$'s value.

3 Finite Differences

Construct $A\boldsymbol{x} = \boldsymbol{b}$.

3.1 Results

4 Analytical Approaches

4.1 Similarity Solution

4.2 Integral Equation

4.2.1 Derivation

4.2.2 Numerical Solution

5 Spectral Method

From the definition of Chebyshev polynomials $T_k(x) = \cos(k\theta)$, we can derive that

$$\frac{\mathrm{d}T_k}{\mathrm{d}x} = \frac{\mathrm{d}T_k}{\mathrm{d}\theta}\frac{\mathrm{d}\theta}{\mathrm{d}x} = ... = kU_{k-1}(x)\,,$$

where $U_k : [-1,1] \mapsto \mathbb{R}$ denote the Chebyshev polynomials of the second kind, which in turn are defined by

$$U_k(\cos\theta)\sin(\theta) = \sin\left((n+1)\theta\right)\,.$$

In order to enforce a von-Neumann boundary condition on the left and a Dirichlet boundary condition on the right, we are interested in explicitly setting coefficients a_k such that

$$a_x(-1,t) = \left.\frac{\mathrm{d}a}{\mathrm{d}x}\right|_{x=-1} = \tilde{l} \quad \text{and} \quad a(1) = r, \quad \text{where} \quad \tilde{l}, r \in \mathbb{R}\,.$$

Using the Chebyshev series ansatz

$$a(x,t) = \sum_{k=0}^{N-1} a_k^{(t)} T_k(x)$$

we have that
$$\frac{\mathrm{d}a}{\mathrm{d}x} = \sum_{k=0}^{N-1} a_k^{(t)} \frac{\mathrm{d}T_k}{\mathrm{d}x}(x)\,,$$
so we are interested in
$$a_x(-1,t) = \left.\frac{\mathrm{d}a}{\mathrm{d}x}\right|_{x=-1} = \sum_{k=0}^{N-1} a_k^{(t)} \left.\frac{\mathrm{d}T_k}{\mathrm{d}x}\right|_{x=-1} = \sum_{k=0}^{N-1} a_k^{(t)} k U_{k-1}(-1)\,.$$
Following from TODO (explained on Wikipedia), we know that
$$U_k(-1) = (-1)^k(k+1) \quad \text{and} \quad T_k(1) = 1 \quad \forall k \in \mathbb{N}\,,$$
which turns our conditions into algebraic conditions w.r.t. the coefficients $a_k^{(t)}$,
$$a_x(-1,t) = \left.\frac{\mathrm{d}a}{\mathrm{d}x}\right|_{x=-1} = \sum_{k=0}^{N-1} a_k^{(t)} k^2 (-1)^{k-1} \overset{!}{=} \tilde{l} \quad \text{and} \quad a|_{x=1} = \sum_{k=0}^{N-1} a_k^{(t)} \overset{!}{=} r\,.$$
Knowing that the heat equation Forward Euler numerical scheme modifies all but the two highest-degree coefficients in the series, we expand:
$$\begin{aligned} a_x(-1,t) = \sum_{k=0}^{N-1} a_k^{(t)} T_k'(-1) &= \overbrace{-\sum_{k=0}^{N-3} a_k^{(t)} k^2 (-1)^k}^{:=\Sigma_3} - (N-2)^2(-1)^{N-2} a_{N-2} \\ &\quad - (N-1)^2(-1)^{N-1} a_{N-1} = l\,, \\ a(1,t) = \sum_{k=0}^{N-1} a_k^{(t)} T_k(1) &= \underbrace{\sum_{k=0}^{N-3} a_k^{(t)}}_{:=\Sigma_2} + a_{N-2} + a_{N-1} = r\,, \end{aligned}$$

5.1 Enforcing Boundary Conditions

Von Neumann on the left

5.2 Implicit Euler

5.3 Implementation

The solver was implemented in C++.

5.4 Results

Chronoamperometry, DC Voltammetry, AC Voltammetry

6 Conclusion