Hat Optimization of Redox Flow Battery Flow Fields Through Physics-Informed Reinforcement Learning

Vivaan Shah, Colin Lopes, Ganeshan Venu

Abstract

The realization of a low-carbon economy and the large-scale transition to renewable energy encompasses some of the most pressing challenges of the era. A developing technology in the energy sector is the vanadium redox flow battery (VRFB). This is a promising long-term energy storage solution that can alleviate the intermittency of renewable sources while boasting benefits over the traditional lithium-ion centered grid structure. However, VRFBs require further enhancement before becoming dominant market players; one target area of improvement is the design of the flow fields (FF) that facilitate pressure consistency, distributed diffusion, and uniform electron transfer between the electrodes and active species ions in an electrolyte solution. Guided by their increasingly pervasive nature, I plan to analyze the effectiveness of physics-informed machine learning techniques to design and maximize these favorable characteristics in a FF, with the end goal being a method to improve VRFB performance without being cost-intensive. Specifically, I will implement a SARSA network that will iteratively improve FF design through reinforcement learning. I will train the network using a FF path generation algorithm and physical simulation of each iteration with components derived from relevant fluid dynamics. Then, I will evaluate it against the three standard FF designs. With guidance, I would love to be able to construct the optimized FF and VRFB in real life.

I. Idea

In past decades, a slew of technologies have been proposed and implemented to reach sustainability goals and broaden the scope of renewable energy sources. These technologies seek to curtail inefficiencies of renewable energy compared to carbon systems, such as its intermittency and unpredictable availability; the sun doesn't always shine and the wind doesn't always blow. Therefore, there is a need for long-term energy storage architecture to underlie the grid. Current solutions are based on lithium-ion, lead-acid, zinc, hydropower, compressed air, and vanadium redox flow (Xu, 2022; Zimmerman, 2014), the latter of which is the focus of this project. Vanadium redox flow batteries (VRFB) differ from conventional galvanic cells due to the external pumping action of electrolyte solutions into and out of the battery cell (Zimmerman, 2014), as shown in Fig. 1. Thus, by increasing the storage space of the electrolyte solutions, VRFBs are easily scaled and allow for decoupling of power and energy, a feature not possible in traditional lithium-ion systems (Viswanathan et al., 2022).

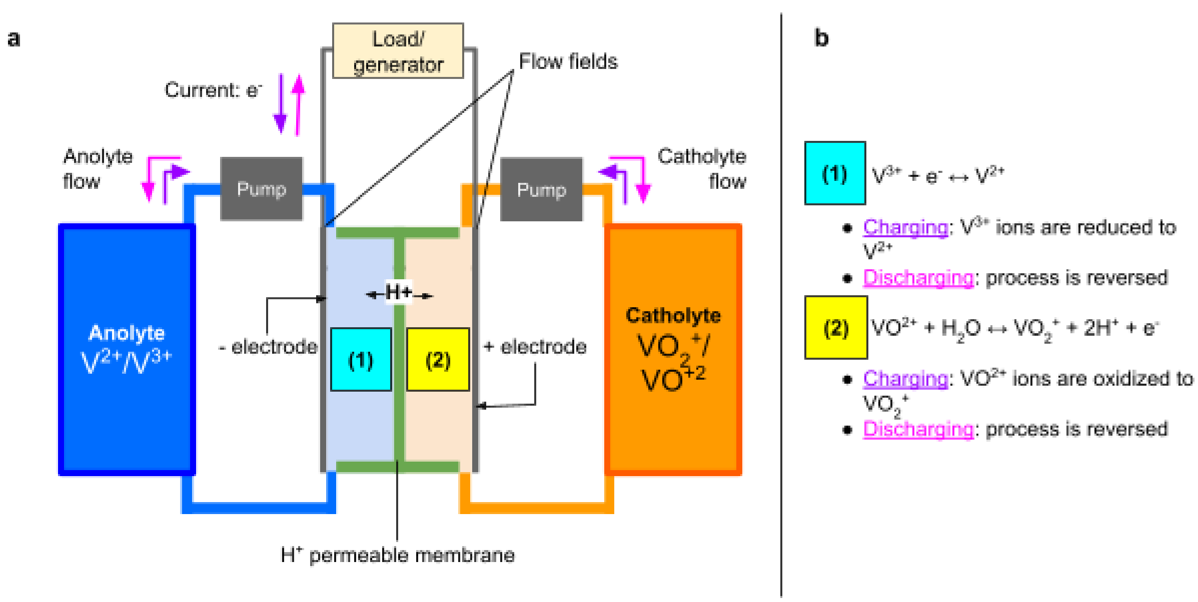

Fig 1. a. Diagram of a VRFB. **b.** The relevant redox reactions for charging (purple) and discharging (pink).

Despite their advantages, VRFBs are faced with a plethora of challenges. For instance, the element vanadium itself, used for its large oxidation state space (Asadipour, 2021) and ability to reduce the effect of cross-contamination between the anolyte and catholyte (UNSW, 2009), is limited in abundance and contributes to a significant portion of the capital cost (Viswanathan et al., 2022), prompting endeavors to create metal-free redox flow batteries (Huskinson et al., 2014; Lin, Chen et al., 2015; Lin, Gómez-Bombarelli et al., 2016; Maloney, 2016). Another issue is pump power consumption, which itself exceeds $100/kWh (Asadipour, 2021).

To compensate for such points, the flow field (FF) of electrolyte solution, meaning the area through which it is pumped to diffuse into the electrodes, can be studied and optimized, representing a simple and cost-effective way to improve VRFB performance. Common FF designs, as shown in Fig. 2a, are interdigitated, serpentine, or parallel (Asadipour, 2021), which all involve a mix of interconnected vertical and horizontal flow channels. Recently, topology optimization (Lin, Baker et al., 2022) and machine learning (Wan et al., 2022) have been researched as methods to improve FF design. For instance, Wan et al. (2022) trained a convolutional neural network (CNN) on a space of generated FFs, each having been evaluated by a multiphysics simulation, and they propose five primary defining characteristics of FFs. While I plan to use this paper as a baseline, its employed techniques may not have yielded the most optimal solution to the FF design problem, and alternative routes should be explored.

In my project, I plan to use another machine learning approach: reinforcement learning (RL) directed by physics-informed techniques. Physics-informed machine learning (PIML) is a powerful, interpolative approach when modeling a dynamical system governed by partial differential equations (PDE) with little experimental data (Hans & Bilionis, 2021). PIML, broad in its techniques, can improve the accuracy of neural networks (Moseley, 2021) and has been used to solve inverse problems such as the one at hand (Willard et al., 2022). PIML has proven itself in RL for robot simulation (Jiang et al., 2021), haemodynamic studies (Brindise et al., 2019), discovery of underlying physics of a system (Chen et al., 2021; MacDonald, 2022; Udrescu & Tegmark, 2020), and solving various PDEs (Rosofsky et al., 2022); a relevant example are the Navier-Stokes equations, demonstrated by Cuomo et al. (2022) and Jin et al. (2020). As demonstrated, PIML is readily applicable to both RL and fluid simulation, and thus I affirm that it can make an impact in VRFB research. In fact, the union of PIML and VFRBs has already been pioneered by He, Fu et al. (2022), He, Stinis et al. (2022), and Howard & Tartakovsky (2021). In this project, PIML will be implemented through a multiphysics evaluation of generative FF designs, by using the VRFB model outlined by Wan et al. (2022) as shown in Fig. 2b.

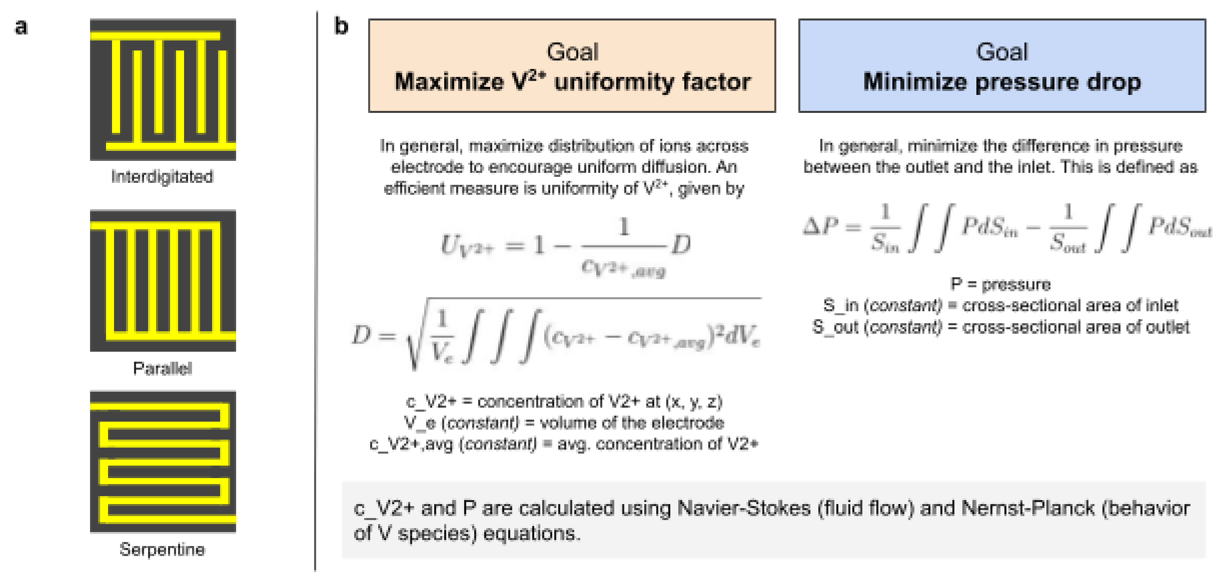

Fig 2. a. Common single-channel FF designs. **b.** VRFB model from Wan et al. (2022). In their paper, a software called COMSOL Multiphysics is used for modeling. As I don't have access to this, I will use SolidWorks or an open-source simulator.

II. Plan

Step 1: Path Representation Algorithm - by Jan 30

The goal of this step is to develop and refine an algorithm that can be used to define a new FF path, given an action to edit an existing one. Each FF will be represented using a 10x10 graph. Three simple algorithms I have tested for path editing are node jumping, single-node traversal, and multi-node traversal (the quickest so far). I will also use this timeframe to learn more about reinforcement learning and VRFBs.

Algorithm 1
Node jumping
Input: 10x10 matrix with a marked cell
Output: new FF
1: new FF ← old FF + input

Algorithm 2
Single-node traversal
Input:
- A: 1x3 matrix of 0s except for 1 cell
- B: Current direction (up, down, left, or right)

Output: new FF
1: Initialize new cell (0, 0)
2: switch A:
3:     case [1, 0, 0]: // straight
4:         new cell ← old cell 1 unit moved in B
5:     case [0, 1, 0]: // right
6:         new cell ← old cell 1 unit moved in (B + 1)
7:     case [0, 0, 1]: // left
8:         new cell ← old cell 1 unit moved in (B - 1)
9: new FF ← old FF with new cell

Algorithm 3
Multi-node jumping
Input:
A: 1x10 matrix of 0s except for 1 cell
B: Current direction (row or column)
Output: new FF
1: Initialize new cell
2: i = index of A where A[i] = 1
3: if B = row:
4:     new cell ← (i, y_{last})
5: else: // column
6:     new cell ← (x_{last}, i)
7: new direction ← opposite of B
8: new FF ← old FF with new cell and cells between new cell and old cell

Step 2: Multiphysics Simulation - by Feb 15

The goal of this step is, given an FF in graph representation, to convert it to 3D so it can be evaluated using a physics simulation that returns the uniformity factor U_V^{2+} (to be maximized) and pressure drop ΔP (to be minimized). I plan to implement this step in 2D first and consider only the ΔP, using Python to model the fluid flow. Then, I will upscale it using the 3D VRFB model in Fig. 2b, or alternatively the 0d model proposed by Pugach et al. (2018), which incorporates uniformity of all ions. Problems that may arise during this step include the simulation being too intensive and time-consuming, which can be fixed by lowering the FF dimensions, keeping the simulation in 2D, or simplifying the RFB model by evaluating fewer parameters.

Step 3: Reinforcement Learning - by Mar 15

The goal of this step is to produce a successful SARSA (state, action, reward, state, action) model to iteratively improve FF design. SARSA is a suitable choice for this task as it requires no data preparation/preprocessing and fits well to graph-based tasks. Iterative improvement of the FF follows a Markov Decision process by applying actions to the FF using the path representation algorithm. The actions will start out random but gradually be overridden by the developing model, as shown in Fig. 3. After ample training, the model must learn a decent correlation between FF design and efficiency. The entire system is summarized in Fig. 4.

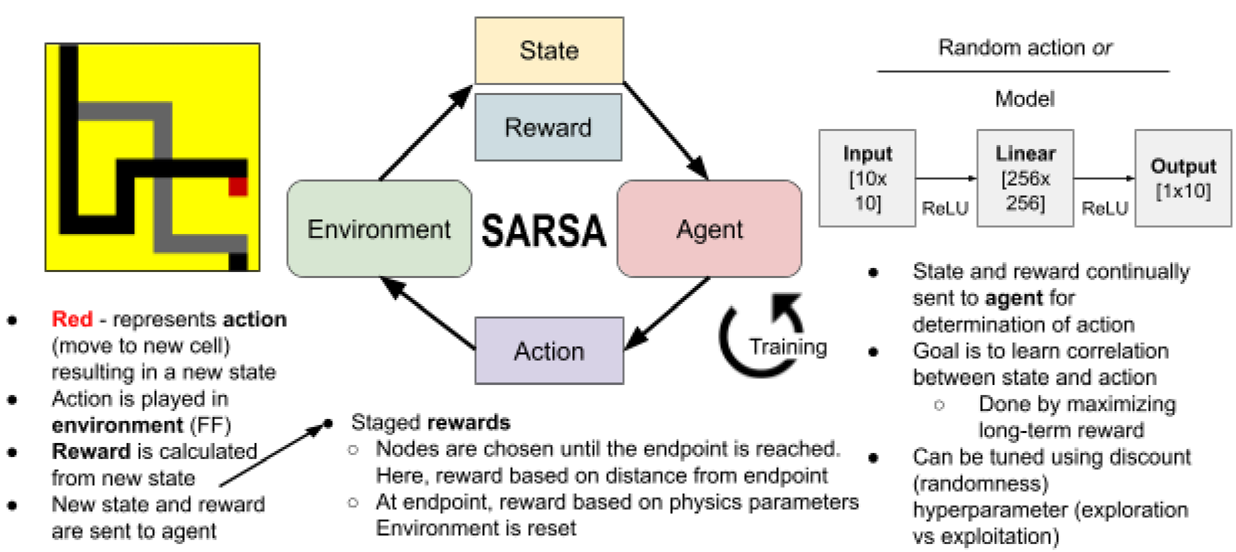

Fig 3. SARSA network applied to FF design.

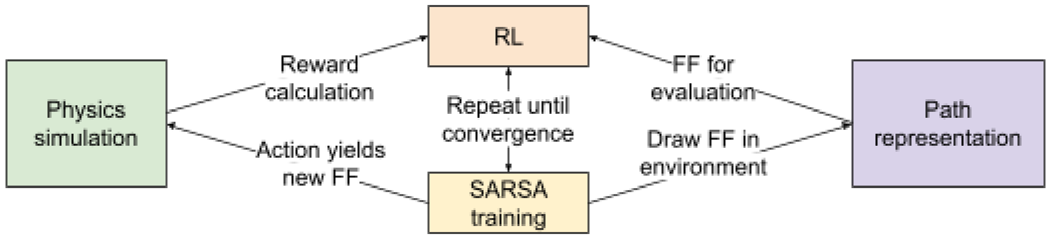

Fig 4. Full system flowchart

To demonstrate the concept, I have adapted code from Loeber (2021), initially an RL tutorial to play snake, to reach an underlying FF path. In my first test, I used the system in Fig. 3 along with the node jumping algorithm to generate a single-channel path in a small 4x4 FF. The results of the training are shown in Fig. 5; after a sufficient amount of time, the model was successfully able to find the path and remain consistent.

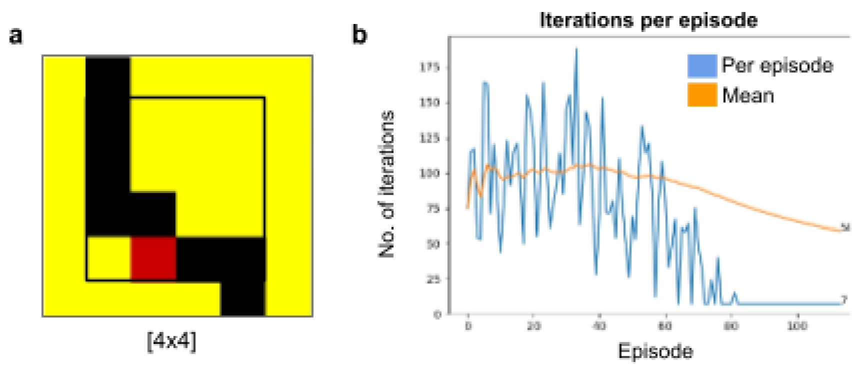

Fig 5. a. 4x4 FF design goal. **b.** Number of iterations to find path. After ~80 episodes, the model reaches the minimum number of iterations (7).

Then, I used multi-node traversal to find a similar path in a 10x10 FF, as node jumping and single-node traversal proved to be time consuming. Unfortunately, this model does not converge to the desired path but rather a shorter one, shown in Fig. 6. A good first step would be to either implement a new path representation algorithm, experiment with new model architectures, or tailor the reward function to better represent deviations from the underlying FF in a given state.

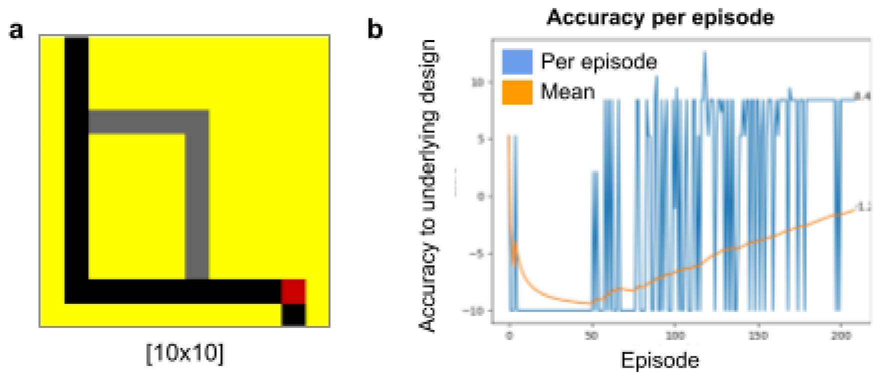

Fig 6. a. Desired (grey) vs predicted (black). **b.** Accuracy metric per episode (reward). After ~200 episodes, the model converges, but not to the desired path.

<u>Step 4: Testing and Tweaking</u> - by Apr 30

By this time, the RL model should be capable of producing its own designs independent of templates. The goal of this step is to test and evaluate the FFs produced by the model against three common FF designs shown in Fig. 2a. In addition, I will use this time to improve the model, such as by experimenting with the hyperparameters or trying different models if the one outlined above does not yield optimal results. If the model does not converge, I will try adding a target network (Deep RL) to curtail volatility.

<u>Step 5: Battery Construction</u> - by May 30

If a suitable FF is found, I would like to construct a small RFB to test it. The FF generated by the model, as well as controls, would be milled into graphite plates and assembled into a battery using components listed in the budget. This may also serve as time for testing and tweaking of the model.

<u>Budget & Funding</u>

I expect the system, particularly the multiphysics simulation, to be both time intensive and computationally expensive for my personal laptop. Thankfully, there are many hardware accelerators available online, like Google Colab. I would dedicate my budget towards a Colab Pro subscription, which offers access to faster GPUs/TPUs. Additionally, if I am able to fabricate the battery, I would purchase the necessary materials, excluding those already available at my house, school or a local lab. The total budget remains below $1000.

Category	Item / Purpose	Cost
Software	Google Colab Pro Subscription - faster computational resources	$9.99/month for 3-4 months = ~$50
VRFB construction	Graphite milling surfaces (2) - FF	$14.99 each = $29.98
	Water pumps (2)	$29.99 each = $59.98
	Nafion™ 212 - Proton membrane	$66.99
	600g Vanadium(V) oxide - electrolyte solution	$33.06 ("Buy 2" bulk pack)
	Carbon paper pack - electrodes	$75.00
	Total	= ~$265

Energy is a well-established field, yet new advancements are being made everyday, as they should be if sustainability goals are to be met. I first came across flow batteries in a YouTube video (Xu, 2022), and after some research, I wanted to try and apply PIML to VRFB design. While I have a fairly confident handle on reinforcement learning and Python programming, I will need to improve (or even learn for the first time) my understanding of the necessary physics and chemistry, as well as all the intricate workings of a flow battery, perhaps including those which use non-vanadium electrolytes (Lin, Gómez-Bombarelli et al., 2016).