Министерство науки и высшего образования Российской Федерации
ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»
Кафедра «школы бакалавриата (школа)»

Оценка работы___100___
Руководитель от УрФУ Кошелев А.А.

Тема задания на практику

Методы решения дифференциальных уравнений на основе искусственных нейронных сетей.

ОТЧЕТ
Вид практики Производственная практика
Тип практики Производственная практика, Научно-исследовательская работа

Руководитель практики от предприятия (организации) Кошелев АА
ФИО руководителя **Подпись**

Студент Кузнецов И.А.
ФИО студента

Специальность (направление подготовки) 01.03.01 Математика

Группа МЕН-490102

Екатеринбург 2023

РЕФЕРАТ

Кузнецов Игорь александрович «Возможность использования искусственных нейронных сетей для решения задач математической физики»: работа содержит: страниц 18, иллюстраций 5, использованных источников 11

Ключевые слова: PINN, дифференциальные уравнения, электрокинетика, нейронные сети, python

PINN (Physics-Informed Neural Networks) - это метод, который сочетает в себе преимущества нейронных сетей и физических моделей для решения задач научного моделирования. В данном дипломном проекте исследуется применение метода PINN для решения задачи обратной задачи электрокинетики. В работе проводится анализ эффективности метода PINN в сравнении с классическими методами решения обратных задач. Также исследуется влияние различных параметров на точность рения задачи. Результаты исследования показывают, что метод PINN может быть эффективным инструментом для решения задач научного моделирования, особенно в случаях, когда классические методы неэффективны или недостаточно точны. Были использованы следующие технологии:

- язык программирования Python;
- библиотека для работы с глубокими нейронными сетями Tensorflow;
- библиотека ESPResSo, для проведения традиционных симуляций;

Содержание

РЕФЕРАТ **2**

1 Введение **4**

2 Постановка задачи **6**

2.1 Цель работы 6

2.2 Описание задачи 6

3 Используемые технологии **7**

4 Теоретическое описание PINN **8**

5 Техническая реализация **10**

6 Эксперименты **11**

6.1 Установившееся распределение тепла в кольце 11

6.2 Задача электрокинетики 13

6.3 Упрощённый случай 14

6.4 Полный случай 15

7 Заключение **16**

8 Список литературы **17**

1 Введение

В последние годы нейронные сети получили широкое распространение, они широко используются для анализа и генерации изображений и видео, обработки естественных языков (перевод, чат-боты), медицинской диагностике, финансовых прогнозах и так далее.

Одним из перспективных направлений в этой области являются так называемые PINN – Physics-Informed Neural Networks, физически-информированные нейронные сети. Классические нейронные сети используют большую выборку реальных данных, однако в естественно-научных областях, таких как физика, химия, биология и т.д. зачастую может просто не хватать нужного объёма данных для обучения. PINN способны обойти это ограничения, используя в обучении знания законов физики, описываемые дифференциальными уравнениями в частных производных. Это позволяет использовать неполные и зашумленные данные, что делает их полезными в реальных научных задачах. Однако, вычислительная сложность PINN выше, чем у классических нейронных сетей, что требует большого количества вычислительных мощностей.

Впервые термин PINN был введён в статье [1]. В ней автор дал формальное определение PINN'ам и рассмотрел решение нескольких задач: уравнение Шрёдингера, Навье-Стокса, Ален-Чана.

В настоящее время PINN широко применяются моделировании, анализе широкого спектра физических явлений:

В статье [2] рассматривается задача симуляции циклической вольтаметрии, исследователями было рассмотрено несколько случая: одномерная вольтаметрия на дисковом электроде с полубесконечными или тонкослойными граничными условиями, двумерная вольтаметрия на микрополосковом электроде и наконец вольтаметрия на края квадратного электрода, количественно определяя неравномерное распределение тока вблизи угла электрода. Для моделирования был использован перцептрон использующий от трёх до шести скрытых слоёв, и гиперболический тангенс в качестве функции активации. Полученные исследователями данные хорошо согласуются с решениями этих же задач, полученными другими способами.

Так же PINN применяются для: анализа литий-ионных батарей[3, 4], для моделирование теплопереноса в системах со сложной геометрией [5, 6], решения уравнения Навье-Стокса для моделирования турбулентности [7], химической кинематике [8, 9]. Для изучения биологических процессов существует разновидность PINN'ов – BINN (Biologically-informed neural network) [10]

2 Постановка задачи

2.1 Цель работы

Рассмотреть уже решённую физическую задачу, решить её с помощью PINN и сравнить полученные данные с изначальным решением, оценить целесообразность применения PINN к задаче.

2.2 Описание задачи

На вход нейросети подаются пространственно-временные координаты. На выходе хотим получить различные характеристики исследуемого физического процесса.

3 Используемые технологии

Для разработки будем использовать язык Python 3.10.6 – высокоуровневый язык программирования общего назначения, один из наиболее популярных языков в области машинного обучения и Tensorflow – библиотеку для создания и обучения нейронных сетей. Выбор данной библиотеки обусловлен простотой создания нейронных сетей, высокой производительностью, а так же встроенным автоматическим дифференцированием, которое и позволит нам обучить нейросеть дифференциальным уравнениям в частных производных.

4 Теоретическое описание PINN

Пусть система описывается некой системой дифференциальных уравнений:

$$F_j(t, x, u, \lambda_j) = 0, x \in \Omega, t > 0, j = \overline{1, N} \tag{1}$$

где t – время, x – пространственные координаты, Ω – некоторая область в пространстве $\mathbb{R}^n$, $u(t, x)$ – искомая функция описывающая интересующие нас свойства системы (скорость, концентрация, потенциал и т.п.), λ_j – векторы постоянных параметров системы, такие как плотность вещества, заряд частиц, теплопроводность материала, температура окружающей среды и тому подобное.

Определим $f(t, x)$ следующим образом:

$$f(t, x) := \begin{pmatrix} F_1(t, x, u) \\ F_2(t, x, u) \\ \dots \\ F_N(t, x, u) \end{pmatrix} \tag{2}$$

где $u(t, x)$ аппроксимируется с помощью глубокой нейронной сети. $f(t, x)$ назовём физически-информированной нейросетью, или же PINN, она может быть получена с помощью автоматического дифференцирования сложных функций. Данная имеет все те же параметры, что и сеть $u(t, x)$, а так же дополнительно набором параметров λ.

Для обучения нейросети составим следующую функцию потерь:

$$MSE = MSE_f + MSE_u \tag{3}$$

где

$$MSE_f = \sum_{j=1}^{N} \frac{1}{N_f} \sum_{i=1}^{N_f} F_j^2(t_f^i, x_f^i, u(t_f^i, x_f^i)) \tag{4}$$

требует соблюдения дифференциальных уравнений, описывающих процесс, здесь $\left\{t_f^i, x_f^i\right\}_{i=1}^{N_f}$ – точки коллокации для F_j, N_f – количеств этих точек и

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u}((u(t_u^i, x_u^i)) - u_0^i)^2 \tag{5}$$

требует соблюдения экспериментальных данных и/или граничных условий $\left(t_0^i, x_0^i, u_0^i\right)_{i=1}^{N_0}$ для функции $u(t,x)$.

5 Техническая реализация

В качестве архитектуры нейронной сети, аппроксимирующей функцию $u(t, x)$ возьмём многослойный перцептрон, точное число слоёв и нейронов в каждом слое будем выбирать экспериментально, в качестве функции активации слоя будем использовать $\tanh$. Для создания нейронной сети воспользуемся классом `tensorflow.keras.Model`, для создания слоёв классом `tensorflow.keras.layers.Dense`. PINN $f(t, x)$ будет иметь всего один слой. Параметрами этого слоя и есть параметры λ_j из 1. Внутри этого слоя мы считаем частные производные $u(t, x)$ с помощью `tensorflow.GradientTape`, и составлять из них и параметров λ_j систему уравнений 1. На выход из данного слоя будем выдавать значения $u(t, x)$ и сами эти уравнения. В силу вида системы 1 все выходы, соответствующие уравнениям системы 1 должны быть равны 0. В качестве оптимизатора будем использовать Adam, а в качестве метрики MSE

6 Эксперименты

6.1 Установившееся распределение тепла в кольце

Рассмотрим для начала относительно простую задачу: определить распространение тепла, установившееся в кольце $1 < r < 2$. $0 < \phi < 2\pi$, с граничными условиями:

$$\begin{aligned} u(1,\phi) &= \cos\phi + \sin\phi + \sin(2\phi) + 5\sin(3\phi) + 1 \\ u(2,\phi) &= \sin(2\phi) + \sin(3\phi) + \cos(4\phi) \end{aligned} \tag{6}$$

Установившееся распространение тепла описывается уравнением Пуассона, учитывая что внутренних источников тепла нет получаем:

$$\Delta u = 0 \tag{7}$$

Для полярных координат оно принимает вид

$$\frac{\partial^2 u}{\partial r} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \phi^2} = 0 \tag{8}$$

Аналитическое решение данной задачи:

$$\begin{aligned} u(r,\phi) = 1 &- \frac{\ln r}{\ln 2} \\ &+ \left(\frac{-r}{3} + \frac{4}{3r}\right)\sin(\phi) + \left(\frac{-r}{3} + \frac{4}{3r}\right)\cos(\phi) \\ &+ \left(\frac{r^2}{5} + \frac{4}{5r^2}\right)\sin(2\phi) \\ &+ \left(\frac{3r^3}{63} + \frac{312}{64r^3}\right)\sin(3\phi) \\ &+ \left(\frac{16r^4}{255} - \frac{16}{255r^4}\right)\cos(4\phi) \end{aligned} \tag{9}$$

В данном случае функция потерь будем выглядеть следующим образом:

$$
\begin{aligned}
MSE = &\frac{1}{N_f}\sum_{i=1}^{N_f}\left(\frac{\partial^2 u}{\partial r}(r_i,\phi_i) + \frac{1}{r}\frac{\partial u}{\partial r}(r_i,\phi_i) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \phi^2}(r_i,\phi_i)\right)^2 + \\
&\frac{1}{N_u}\sum_{i=1}^{N_u}(u(1,\phi_i) - \cos\phi + \sin\phi + \sin(2\phi_i + 5\sin(3\phi_i+1)^2 + \\
&\frac{1}{N_u}\sum_{i=1}^{N_u}(u(2,\phi_i) - \sin(2\phi_i + \sin(3\phi_i + \cos(4\phi_i)^2
\end{aligned}
\tag{10}
$$

Запустим обучение со следующими конфигурациями скрытых слоёв:

- 20, 20, 20
- 50, 50, 50

Для одной итерации обучения возьмём по одной точке для каждого граничного условия 10 и одну внутреннюю точку для уравнения 9, всего 3 точки на итерацию. Рассмотри случаи с 1000, 10000 и 100000 точек. Обучать будем в течении 1000 эпох или пока ошибка не будет улучшаться в течении 10 эпох хотя бы на 0.0001, размер батча 32. Результаты обучения изображён на рисунке 1, время обучения указано в таблице 1

Итераций / Слои	1000	10000	100000
[20, 20, 20]	10.06	85.62	401.14
[50, 50, 50]	12.68	51.69	173.97

Таблица 1. Время обучения в секундах

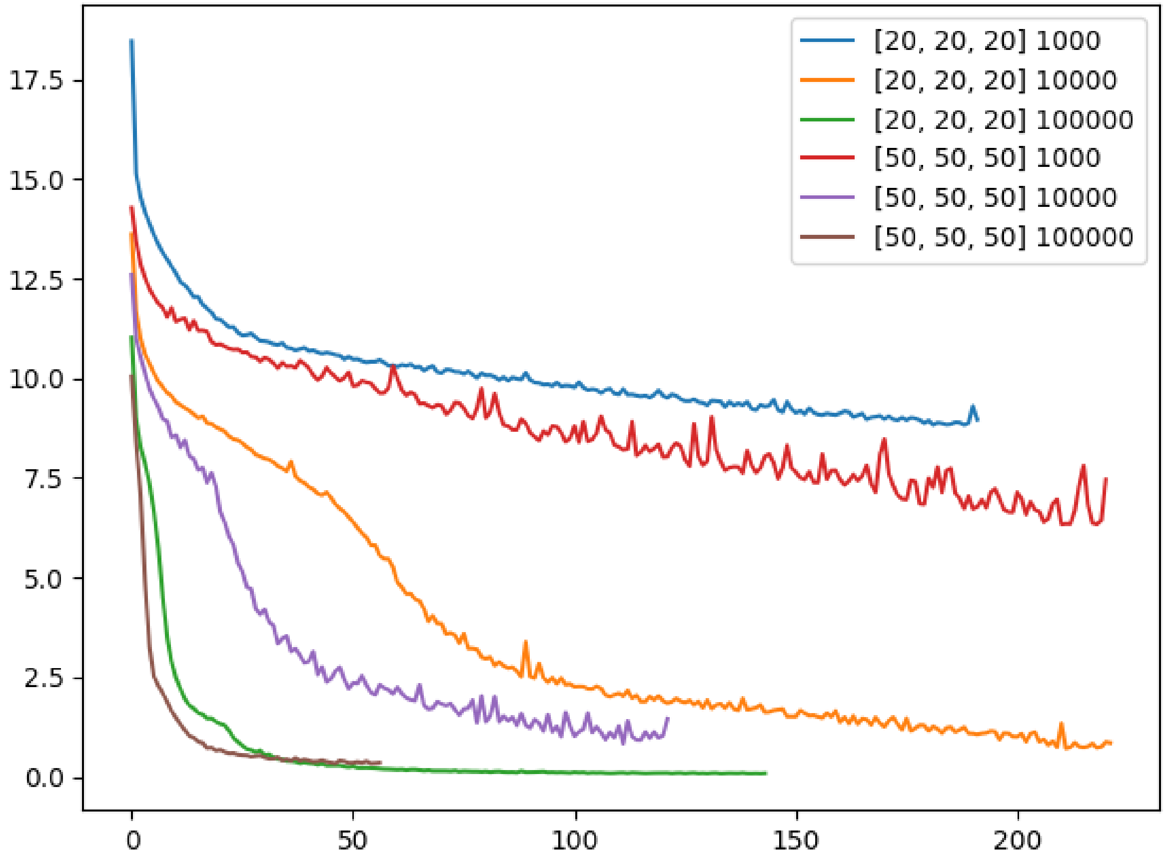

Рис. 1. График loss-функции числа в квадратных скобках – слои, последнее число – количество эпох

6.2 Задача электрокинетики

Для примера возьмём систему из [11]. Она описывается следующими уравнениями

$$
\begin{aligned}
\vec{j} &= -D\nabla c - \xi zec\nabla\Phi + c\vec{u} \\
\partial_t c &= -\nabla \cdot \vec{j} \\
\nabla^2\Phi &= -4\pi l_{\mathrm{B}} k_{\mathrm{B}} T zc \\
\rho\big(\partial_t \vec{u} + (\vec{u}\cdot\nabla)\vec{u}\big) &= -\nabla p_H + \eta\nabla^2\vec{u} - (k_{\mathrm{B}} T\nabla c + zec\nabla\Phi) \\
\nabla \cdot \vec{u} &= 0
\end{aligned}
$$

Здесь первое уравнение в системе описывает поток плотности, второе электростатику, третье гидродинамику с помощью уравнения Навье-Стокса, четвёртое уравнение несжимаемости жидкости.

Рассмотрим систему щелевых пор, состоящую из двух одноимённо заряженных бесконечных пластин. Выпишем для такой системы граничные условия

$$
\begin{aligned}
c(t, X_l) &= 0.01 \\
c(t, X_r) &= 0.01 \\
c(0, X) &= 0.002 \\
\vec{v}(t, X_l) &= 0 \\
\vec{v}(t, X_r) &= 0 \\
\vec{v}(0, X) &= 0 \\
\Phi(t, X_l) &= -0.05 \\
\Phi(t, X_r) &= -0.05 \\
\Phi(0, X) &= -0.009x^2 + 2
\end{aligned}
$$

здесь t – время, X_l – пространственные координаты, соответствующие левой стенке, X_r – правой, x в формул для $\Phi(0, x)$ соответствует оси, перпендикулярной пластинам.

6.3 Упрощённый случай

В начале рассмотрим упрощённую систему уравнений с одной пространственной координатой и без скорости

$$
\begin{aligned}
\frac{dc}{dT} &= -\nabla \cdot (-D\nabla c - \xi zec\nabla\Phi) \\
\nabla^2\Phi &= -4\pi l_{\mathrm{B}} k_{\mathrm{B}} T c
\end{aligned}
$$

С граничными условиями

$$
\begin{aligned}
\frac{dc}{dX}(t, 0) &= 0 \\
\frac{dc}{dX}(t, 1) &= 0 \\
c(0, X) &= 1 \\
\Phi(0) &= 1 \\
\Phi(1) &= 1
\end{aligned}
$$

Координаты в данном случае нормализованны

$$T = t/t_{\max},\ X = (x + x_{\min})/x_{\max}$$

где t_max – время симуляции, x_max – размер системы по x-овой координате, $x_{\min}$ - координата левой стенки. Оператор ∇ действует по пространственным координатам $\nabla = (\frac{d}{dX})$, все константы положим равными 1 и исключим 4π из второго уравнения.

Запустим обучение следующим образом: 100 запусков обучения (функция `fit`) по 10 эпох на 10000 наборах точек (внутренняя, граничная и начальная). Обучение заняло 2 минуты 48 секунд, значение функции потерь: $1.7378 \cdot 10^{-5}$. Результаты симуляции представлены на рис. 2 3:

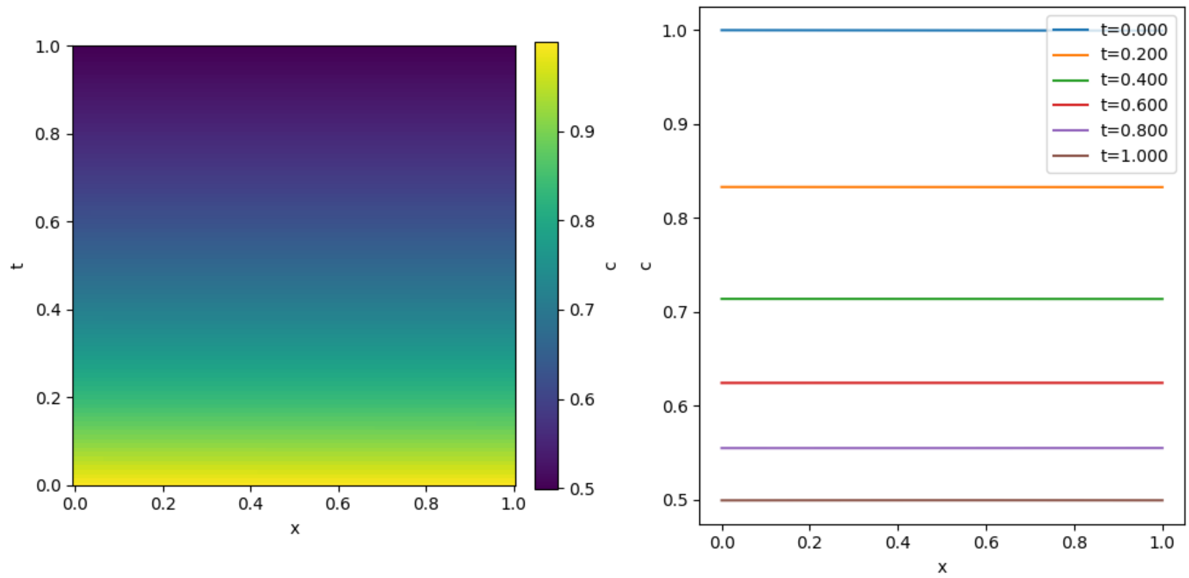

Рис. 2. Упрощённый случай: концентрация

6.4 Полный случай

Рассмотрим двумерный случай. Скрытые слои будут иметь размеры 80 и 40, входной слой 4, один для концентрации, два для скорости и один для потенциала, функция активации tanh (гиперболический тангенс). Результат работы после 5000000 итераций показан на графике 4

Рассмотрим теперь трёхмерный случай. Скрытые слои так же будут иметь размеры 80 и 40, входной слой 5, один для концентрации, три для скорости и один для потенциала, функция активации *relu*. Результат работы после 30000000 итераций показан на графике 5

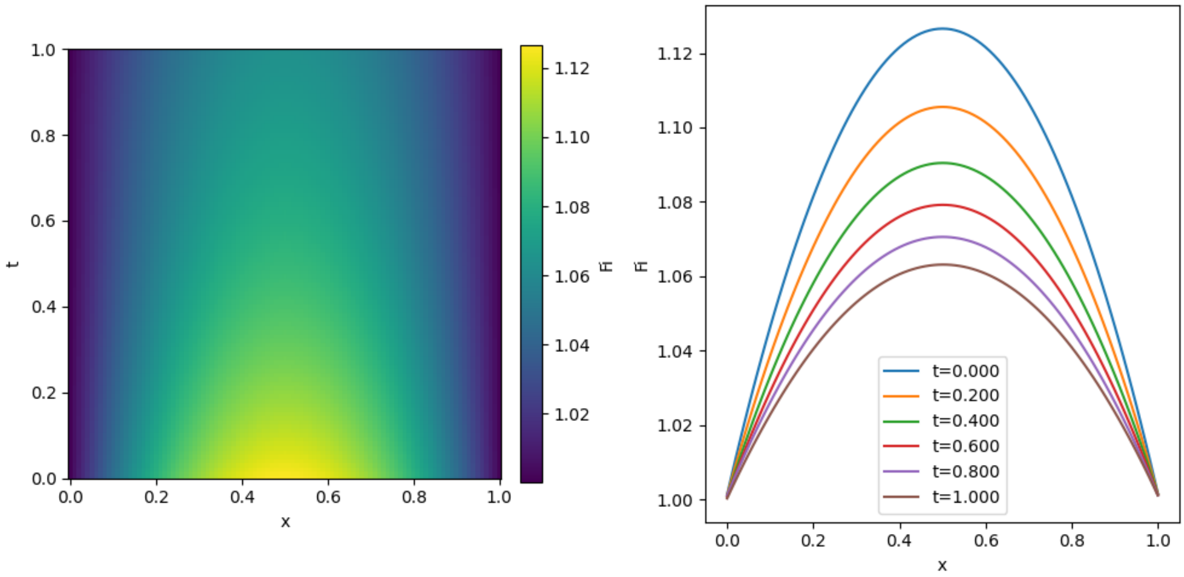

Рис. 3. Упрощённый случай: потенциал

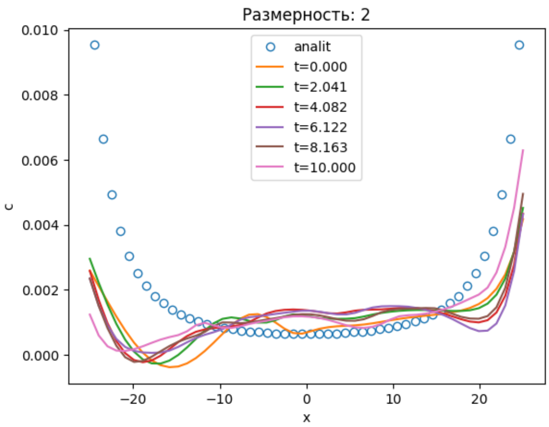

Рис. 4

7 Заключение

Мы написали и протестировали PINN, который решает задачу электрокинетики о системе щелевых пор, состоящих из двух одноимённо заряженных бесконечных пластин. На данном этапе не удалось достичь достаточно точных результатов. Связанно это может быть как со сложностью самих исходных уравнений, так и с недостаточным временем обучения модели. Так же возможно что при задании уравнений была допущена ошибка

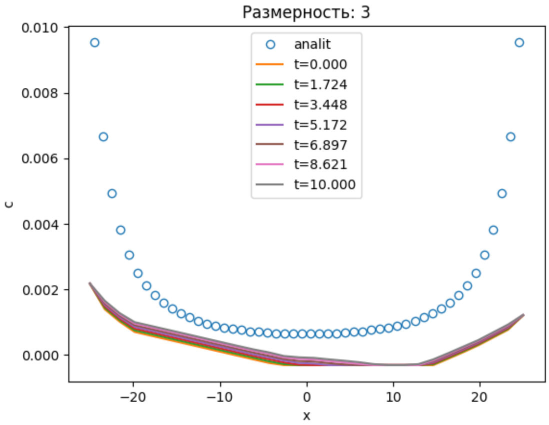

Рис. 5

8 Список литературы

[1] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[2] Haotian Chen, Enno Kätelhön, and Richard G. Compton. Predicting voltammetry using physics-informed neural networks. *The Journal of Physical Chemistry Letters*, 13(2):536–543, 2022. PMID: 35007069.

[3] Muratahan Aykol, Chirranjeevi Balaji Gopal, Abraham Anapolsky, Patrick K. Herring, Bruis van Vlijmen, Marc D. Berliner, Martin Z. Bazant, Richard D. Braatz, William C. Chueh, and Brian D. Storey. Perspective-combining physics and machine learning to predict battery lifetime. *Journal of the Electrochemical Society*, 168(3), 2021.

[4] Renato G. Nascimento, Matteo Corbetta, Chetan S. Kulkarni, and Felipe A. C. Viana. Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *Journal of Power Sources*, 1(513), 2021.

[5] Navid Zobeiry and Keith D. Humfeld. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101:104232, 2021.

[6] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, 143(6), 04 2021. 060801.

[7] Fabian Pioch, Jan Hauke Harmening, Andreas Maximilian Müller, Franz-Josef Peitzmann, Dieter Schramm, and Ould el Moctar. Turbulence modeling for physics-informed neural networks: Comparison of different rans models for the backward-facing step flow. *Fluids*, 8(2), 2023.

[8] Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36):8098–8106, 2021. PMID: 34463510.

[9] Gabriel S. Gusmão, Adhika P. Retnanto, Shashwati C. da Cunha, and Andrew J. Medford. Kinetics-informed neural networks. *Catalysis Today*, 2022.

[10] John H. Lagergren, John T. Nardini, Ruth E. Baker, Matthew J. Simpson, and Kevin B. Flores. Biologically-informed neural networks guide mechanistic modeling from sparse experimental data, 2020.

[11] Jean-Noël Grad. Electrokinetics.