Volume 8 Number 1 2019

STUDENT JOURNAL OF PHYSICS

INTERNATIONAL JOURNAL

INDIAN ASSOCIATION OF PHYSICS TEACHERS

ISSN – 2319-3166

STUDENT JOURNAL OF PHYSICS

This is a quarterly journal published by Indian Association Of Physics Teachers. It publishes research articles contributed by Under Graduate and Post Graduate students of colleges, universities and similar teaching institutions, as principal authors.

Registered Office

Editor-in-Chief, SJP, Institute of Physics, Sainik School, Bhubaneswar, Odisha, India – 751005
(www.iopb.res.in/~sjp/)

STUDENT JOURNAL OF PHYSICS

Scope of the Journal

The journal is devoted to research carried out by students at undergraduate level. It provides a platform for the young students to explore their creativity, originality, and independence in terms of research articles which may be written in collaboration with senior scientist(s), but with a very significant contribution from the student. The articles will be judged for suitability of publication in the following two broad categories:

1. **Project based articles**
   These articles are based on research projects assigned and guided by senior scientist(s) and carried out predominantly or entirely by the student.
2. **Articles based on original ideas of student**
   These articles are originated by the student and developed by him/ her with possible help from senior advisor. Very often an undergraduate student producing original idea is unable to find a venue for its expression where it can get due attention. SJP, with its primary goal of encouraging original research at the undergraduate level provides a platform for bringing out such research works.
   It is an online journal with no cost to the author.

Since SJP is concerned with undergraduate physics education, it will occasionally also publish articles on science education written by senior physicists.

Information for Authors

- Check the accuracy of your references.
- Include the complete source information for any references cited in the abstract. (Do not cite reference numbers in the abstract.)
- Number references in text consecutively, starting with [1].

- Language: Papers should have a clear presentation written in good English. Use a spell checker.

Submission

1. Use the link "Submit" of Website to submit all files (manuscript and figures) together in the submission (either as a single .tar file or as multiple files)
2. Choose one of the Editors in the link "Submit" of Website as communicating editor while submitting your manuscript.

Preparation for Submission

Use the template available at "Submit" section of Website for preparation of the manuscript.

Re-Submission

- For re-submission, please respond to the major points of the criticism raised by the referees.
- If your paper is accepted, please check the proofs carefully.

Scope

- SJP covers all areas of applied, fundamental, and interdisciplinary physics research.

STUDENT JOURNAL OF PHYSICS

Exploring Symbolic Neural Networks for Multiscale Applications

Elijah Sheridan[1,2], **David Rimel**[2] **and Michael S. Murillo**[2]
[1] 2nd Year, Undergraduate, Department of Physics and Astronomy, College of Arts and Science, Vanderbilt University, Nashville, Tennessee 37235
[2] Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, Michigan 48824

Abstract. Many scientific and engineering applications involve dynamics across wide ranges of length and time scales. Modeling such multiscale systems is challenging when the known macroscopic models are approximate versions of the underlying microscopic dynamics. The nonexistence of general mathematical descriptions of some such systems motivates the use of additional tools, such as machine learning, in computational modeling. Additional goals of scientific objectivity, interpretability, and potential for extrapolation lead us to consider the use of deep symbolic neural networks, in particular, in the hope of learning analytic forms representing the macroscopic equations of motion in a multiscale system directly from the microscopic, atomistic dynamics. In this paper, we implement the previously developed EQL$^{\div}$ (EQuation Learner) model—an end-to-end differentiable, feed-forward function-learning network—in multiscale modeling and evaluate the model using simpler physical systems, with broader future applications in mind. We find that EQL$^{\div}$ is capable of exactly solving systems in some instances, but is able only to approximate more complex physical relations. We also identify hyperparameter sensitivities and structural idiosyncrasies that result in potentially undesirable behavior, leading us to consider particular potential future improvements.

Keywords: Neural Network, Multiscale Application, Equation Learner.

1. INTRODUCTION

A multiscale system is a physical system whose structure and evolution relies on information from different temporal and spatial scales. Such systems are studied across scientific and engineering disciplines; we are interested in the computational modeling of multiscale systems. A prototypical example of a microscopic model is molecular dynamics (MD), which models atomic and molecular behavior typically at the sub-micron scale. Conversely, hydrodynamics is a standard macroscopic model that can describe phenomena on very large length and times scales; however, hydrodynamics relies on microscopic closure information that can only be obtained from microscale models. This closure information is, in fact, the information that distinguishes one material from another in the model. In many cases, the structure of the closure is known, and the closure information can be computed in advance; for example, MD can be used to precompute the equation of state needed to inform the pressure closure term in a hydrodynamics model. In some cases, however, the closure and/or its data are not known in advance.

Here, we examine a particular class of multiscale model, known as heterogeneous multiscale models (HMMs), that employs two separate methods at the two scales. Typically [18], the two methods are run sequentially; the microscale model, usually MD, runs first and produces closure data, and the macroscale model, usually hydrodynamics, runs next and uses that closure data until it becomes stale, at which point the cycle repeats. It is desirable that multiscale models depend as little as possible on information contained at the smaller scales, as smaller-scale systems are computationally much more expensive because of both the quantity of items to be simulated and the system complexity. However, a minimal level of microscale computation is typically required to maintain a target level of accuracy. In general, it is not desirable to employ advanced, complex problem-solving approaches when such approaches are not strictly necessary because of expense or impracticality. However, the inherent complexities of some multiscale systems warrant a search for alternative modeling methods.

We chose to explore machine learning (ML) approaches in this work for several reasons. First, we—and other researchers [12][13]—hope to use ML to resolve several difficulties, including the closure problem. Second, we observe that with extensive research into multiscale systems, traditions and customs have evolved that result in inevitable biases, which incentivize the selection of an objective, data-driven approach, and ML approaches can be designed to fulfill these conditions. Finally, ML is an emerging tool that is proving to be of increasing utility in the sciences [14], and we hope to use ML to maximize the accuracy of our results. We note that ML was by no means our only possible choice of method, and that other data-driven methods [16] have been developed to mitigate the closure problem and other unresolved problems as well.

Among the many types of ML, we chose to develop deep neural networks for our multiscale modeling. ML models, in general, are extremely sensitive to the form of their input data, and because of this sensitivity, feature engineering, i.e., the construction and preparation of input parameters, is of the utmost importance. To escape existing scientific biases as much as possible, we must choose to be as hands-off as possible in the implementation of our ML method; we should let even the process of feature engineering be controlled by the ML model. For this reason, we chose to use deep neural networks in this work; in these models, each layer receives features produced, customized, and optimized by the layer before it. Using deep neural networks, we hope to minimize scientific bias in our results and thereby to increase the chance of producing innovative results as well.

Among all possible ML approaches, we examined ML models that learn analytic mathematical expressions—in particular, macroscale equations of motion. Generic neural networks are often viewed as uninterpretable black boxes, as they are often applied to problems with no presumed underlying mathematical laws. In the physical world, however, much of behavior is governed by mathematical laws; thus, we chose to employ symbolic, function-learning neural networks in this work [11]. This selection ensures that our ML model learns something that is structurally similar to the goal (a function), thereby enabling higher accuracy. Furthermore, because actual equations describe phenomena not merely at individual points but across regions, we can expect to be able to extrapolate results obtained with a function-learning network more accurately as well. Finally,

working with a symbolic network gives us the luxury of being able to critically examine and interpret our model's output and its components.

In this paper, we explored the use of the Equation Learner with Division (EQL$^\div$) model [2], which is a modification of the original Equation Learner (EQL) model [1] and a general-purpose end-to-end differentiable, feed-forward function-learning network. It is worth noting that many other symbolic function-learning neural networks exist [8–10] and that EQL$^\div$ is also not inherently physics-informed, unlike alternative ML approaches that take the form of both symbolic [5] and non-symbolic [6, 7, 15] neural networks.

In Section 2, we introduce the EQL$^\div$ model in depth, beginning with a survey of the general theory and vocabulary of neural networks. Next, in Section 3, we provide experimental evaluation of EQL$^\div$ motivated by our particular interest in computational modeling. Finally, in Section 4, we offer conclusions about our findings and a perspective on the path forward for incorporating ML into the study and simulation of multiscale systems.

2. NEURAL NETWORKS AND EQL$^\div$

In this section, we discuss the theory behind the EQL$^\div$ model. The first subsection briefly introduces neural networks and their architecture and training. With this framework, we are then able to construct a comprehensive, rigorous description of the mathematical architecture of the EQL$^\div$ model in the second subsection. The third and fourth subsections motivate and establish the loss function and training schedule used with the EQL$^\div$ model, respectively.

2.1 *A Brief Introduction to Artificial Neural Networks*

Artificial neural networks (ANNs) are a broad class of ML algorithms whose design is based on the structure and behavior of biological neurons. The simplest ANNs use as their fundamental unit a "neuron," whose architecture accepts n input variables and returns one output value. The neuron works by computing a linear combination of those input variables and applying an "activation function" to this sum, which is a nonlinear function (e.g., a sigmoid function).

Mathematically, the ANN can be expressed in terms of a vector of input variables $\mathbf{x} \in \mathbb{R}^n$, a "row vector" or $1 \times n$ matrix of weights $\mathbf{w} \in \mathbb{M}_{1,n}$, a bias $b \in \mathbb{R}$, and a activation function $f : \mathbb{R} \to \mathbb{R}$. Here, the "bias" is simply a constant that is independent of the input vector; it can be thought of as a weight that multiplies $x_0 = 1$. In this framework, for a given set of inputs, our neuron computes and returns $f(\mathbf{w}\mathbf{x} + b)$; in this way, our neuron is really a function $P : \mathbb{R}^n \to \mathbb{R}$.

The neuron learns by changing its weights and bias, the elements of the row vector $\mathbf{w}$ and the number b, respectively. For every input $\mathbf{x}$ that we give our neuron, we have some value $y \in \mathbb{R}$ that we wish our neuron to return, and we penalize the neuron based on how far off its result is from this y, i.e., how large $|f(\mathbf{w}\mathbf{x} + b) - y|$ is. We desire that our neuron chooses weights to minimize this penalty, which we call a "loss function," across all training data. Multivariable minimization problems can be treated with calculus-based methods, in this case usually with what are called

"backpropagation" algorithms, which use the negative gradient to enable our neuron to efficiently "learn" by consistently updating its weights to improve accuracy with respect to a dataset. We refer to the input vectors in this data set as "predictors" and the desired output values as "labels." Given a model with ℓ weights and biases, our loss function can be understood as $L : \mathbb{R}^\ell \to \mathbb{R}$, and we can think about this learning process as the traversal of the graph of L (an ℓ-dimesional hypersurface in $\mathbb{R}^{\ell+1}$) toward local minima, analogous to how a marble would roll to the nearest crevice on a wrinkled sheet. We use standard backpropagation methods [19], which we won't discuss further.

We generalize the single neuron first by recognizing that we can produce m outputs (instead of just one) by merely sending our input $\mathbf{x}$ through m distinct neurons, each with its own weights and bias. Instead of considering many separate row vectors $\mathbf{w_1}, \ldots, \mathbf{w_m}$, we can stack them on top of each other as rows in a single matrix $\mathbf{W} \in \mathbb{M}_{m,n}$, and similarly, we can define a vector $\mathbf{b} \in \mathbb{R}^m$ that stacks the individual biases associated with each neuron on top of each other to make a vector. The matrix $\mathbf{W}$ and vector $\mathbf{b}$ then act on our input vector $\mathbf{x}$, and we obtain the final output of our "neuron group" by applying our activation function f to each component of the resulting vector $\mathbf{W}\mathbf{x} + \mathbf{b} \in \mathbb{R}^m$. Note that our network is now a function $P : \mathbb{R}^n \to \mathbb{R}^m$.

Finally, we are able to realize the most general form of neural network by chaining these groups of individual neurons together. This is accomplished by feeding the output of one neuron group as input into another neuron group. The number of neuron groups chained together in this way is the number of layers L of the model; for example, an architecture with $L = 3$ layers has three separate $\mathbf{W}$ matrices, $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$, and $\mathbf{W^{(3)}}$, and three separate $\mathbf{B}$ vectors, $\mathbf{b^{(1)}}$, $\mathbf{b^{(2)}}$, $\mathbf{b^{(3)}}$. Such models are called "multi-layer networks."

Because each layer of the ANN has its own weight matrix $\mathbf{W_i}$ and bias vector $\mathbf{b_i}$, chaining neurons together in this way introduces another variable. We usually have a fixed number of inputs n and outputs m that we wish to use and produce, respectively, but if $L \geq 3$, then we have what are called "hidden layers," which neither directly receive the input variables nor output the final result of the multi-layer neuron. We refer to ANNs with $L \geq 3$ as "deep." Before, our values n and m determined the dimensions of the weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$; now, we potentially have many weight matrices and bias vectors, and those associated with hidden layers can theoretically be of any dimension, as long as each $\mathbf{W^{(i)}}$ and $\mathbf{b^{(i)}}$ is compatible with $\mathbf{W^{(i-1)}}$, $\mathbf{W^{(i+1)}}$ and $\mathbf{b^{(i-1)}}$, $\mathbf{b^{(i+1)}}$, respectively, for matrix multiplication and vector addition. We call the number of elements in the vector output of an individual layer the "width" z_i of that layer, with $z_L = m$ by definition.

We distinguish between two types of parameters in our model. Several variables in our model are predetermined and are thus established before training begins (e.g., n, m, L), while others are "learned" by the model during training (e.g., the elements of $\mathbf{W^{(i)}}$ and $\mathbf{b^{(i)}}$). We call the former "hyperparameters" and the latter simply "parameters," or alternatively, weights or trained variables.

EQL$^\div$, the ML model we utilize in this paper and introduce in the next section, is an extension of the ANN approach we have established here. This means that the vocabulary we develop here can be applied to the EQL$^\div$ as well. We demonstrate this in Fig. 1, which gives a diagram of an example of an EQL$^\div$ model architecture (and of its predecessor, EQL). The EQL$^\div$ model alternates

between the application of weights/biases to input vectors (labeled as the functions $T^{(1)}$, $T^{(2)}$, and $T^{(3)}$ in the diagram) and a set of non-linear functions to each individual output component (labeled collectively $F^{(1)}$ and $F^{(2)}$ in the diagram) that behaves like a activation function. In this way, Fig. 1 demonstrates how the EQL$^{\div}$ model is an extension of a basic neural network: in particular, we see an example EQL$^{\div}$ model with the hyperparameters $n = 3$, $m = 3$, and $L = 3$ (recall that $T^{(i)}$ and $F^{(i)}$ are grouped together for each i to form an individual layer), with $z_i = 6$ throughout.

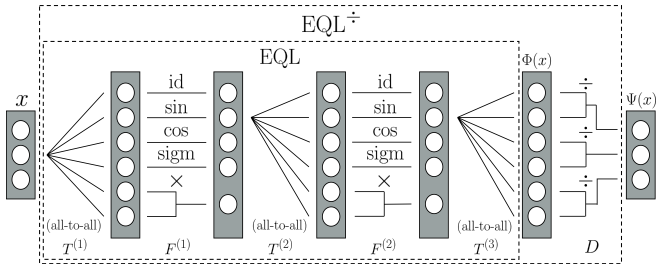

Figure 1. Architecture of an instance of the EQL$^{\div}$ model, demonstrating how it builds on both generally on the general neural network architecture described in Sec. 2.1 and specifically on its predecessor, the EQL model. In the language of Sec. 2.1, this instance has hyperparameters $n = 3$, $m = 3$, $L = 3$, and $z_i = 6$. In the language of Sec. 2.2, this instance has hyperparameters $n = 3$, $m = 3$, $L = 3$, $u_i = 4$, $v_i = 1$, $H = \{\mathrm{id}, \sin, \cos, \mathrm{sigm}\}$, and $I^{(i)} = \{1, 2, 3, 4\}$. Note that in this diagram, nodes are values, while lines are operations; in contrast, conventionally, the opposite is true.

2.2 *EQL$^{\div}$ Architecture*

We now introduce the architecture of the EQL$^{\div}$ model [2]. An EQL$^{\div}$ model is defined by the following hyperparameters: the number of inputs n, the number of outputs m, the number of layers L, the two $(L-1)$-tuples that together define layer widths $U = \{u^{(1)}, \ldots, u^{(L-1)}\}, V = \{v^{(1)}, \ldots, v^{(L-1)}\} \subset \mathbb{N}$, the ordered hypothesis set $H = \{\mathcal{F}_1, \ldots, \mathcal{F}_\ell\} \subset (\mathbb{R} \to \mathbb{R})$, the $(L-1)$-tuple of unary function index tuples $\mathcal{I} = \{\mathbf{I^{(1)}}, \ldots, \mathbf{I^{(L-1)}}\}$ (where $\mathbf{I^{(j)}} \subset \mathbb{N}/\ell\mathbb{N}$ is a $u^{(j)}$-tuple, $1 \leq j \leq L-1$), and the division threshold $\theta \in \mathbb{R}$ (described below). Such a model learns a function $\Psi : \mathbb{R}^n \to \mathbb{R}^m$ defined as $\Psi(x) = (\psi^{(1)} \circ \ldots \circ \psi^{(L)})(x)$, where $\psi^{(i)}$ denotes the function applied by the ith layer.

Layer i first applies a (fully trainable) affine map $T^{(i)}(\mathbf{x}) = \mathbf{W^{(i)}}\mathbf{x} + \mathbf{b^{(i)}}$. If $i < L$, the layer then applies a nonlinear map $F^{(i)} : \mathbb{R}^{u^{(i)}+2v^{(i)}} \to \mathbb{R}^{u^{(i)}+v^{(i)}}$ composed of $u^{(i)}$ unary functions $f_1^{(i)}, \ldots, f_u^{(i)} : \mathbb{R} \to \mathbb{R}$, sampled from H through $\mathcal{I}$ such that $f_j^{(i)} = \mathcal{F}_{\mathbf{I_j^{(i)}}}$, and $v^{(i)}$ instances of multiplication, as follows:

$$F^{(i)}(\mathbf{x}) = \left(f_1^{(i)}(x_1), \ldots, f_{u^{(i)}}^{(i)}(x_{u^{(i)}}), x_{u^{(i)}+1} \cdot x_{u+2}, \ldots, x_{u^{(i)}+2v^{(i)}-1} \cdot x_{u^{(i)}+2v^{(i)}}\right). \quad (1)$$

Where x_i gives the ith component of the input vector $\mathbf{x}$. If $i = L$, then the layer instead follows the affine map with a special element-wise division function $D : \mathbb{R}^{2m} \to \mathbb{R}^m$, defined as

$$D(\mathbf{x}) = (h^\theta(x_1, x_2), \ldots, h^\theta(x_{2m-1}, x_{2m})), \tag{2}$$

where the binary function $h^\theta : \mathbb{R}^2 \to \mathbb{R}$ is, in turn, a division operation modified to avoid divergences during training, defined as

$$h^\theta(a, b) = \begin{cases} \frac{a}{b} & b > \theta, \\ 0 & b < \theta. \end{cases} \tag{3}$$

The net transformation $\psi^{(i)}$ associated with a given layer is then

$$\psi^{(i)}(\mathbf{x}) = \begin{cases} \left(F^{(i)} \circ T^{(i)}\right)(\mathbf{x}) & i < L, \\ \left(D \circ T^{(i)}\right)(\mathbf{x}) & i = L. \end{cases} \tag{4}$$

Eq. 4 allows us to establish the domain and codomain of $\psi^{(i)}$:

$$\psi^{(i)} : \begin{cases} n \to u^{(i)} + v^{(i)} & i = 1, \\ u^{(i-1)} + v^{(i-1)} \to u^{(i)} + v^{(i)} & 1 < i < L, \\ u^{(i-1)} + v^{(i-1)} \to m & i = L. \end{cases} \tag{5}$$

The domains and codomains for each component of $\psi^{(i)}$ can then be inferred from Eq. 5 and the definitions of $T^{(i)}$ and $F^{(i)}$.

The relationship between EQL$^\div$ and EQL, from which EQL$^\div$ originates, is shown in Fig. 1. Each EQL$^\div$ model can be thought of as containing an EQL model within it: if we define $\Phi(\mathbf{x}) = (T^{(L)} \circ \psi^{(L-1)} \circ \ldots \circ \psi^{(1)})(\mathbf{x})$, then $\Phi(\mathbf{x})$ and all of the hyperparameters it depends on precisely define the architecture of the EQL model. EQL$^\div$, then, is simply EQL with division.

2.3 *The EQL$^\div$ Loss Function*

We now consider the qualitative goals of EQL$^\div$ and use these to motivate the terms of the loss function we employ with our model [2].

Most importantly, we want these models to be accurate and capable of learning equations that represent data well. This desire mandates the presence of a term in our loss function that captures the error of the model with respect to training data; in our loss function, we use the mean squared error for this purpose.

Next, and more subtly, because we use these models to learn real, physical equations, it is often desirable for a model to use only a small fraction of its weights. EQL$^\div$ has the capacity to develop learned functions that are hundreds of terms long, but in most cases, actually learning such long functions would be an example of overfitting the data. Thus, we chose to include a term in our loss

function that encourages sparsity: an L1 regularizer [19]. The combination of the mean squared error with L1 regularization is commonly referred to as a Lasso-like objective [3].

Finally, we have constructed our division layer in such a way that we avoid divergences by ignoring (setting to zero) division nodes with denominators below a threshold θ. Thus, in the event that a division node settles on a sufficiently small or negative denominator at any point in training, it not only will be zeroed out but will also possess no gradient to allow weights and biases to evolve in such a way that the denominator ever becomes positive again. This situation prompts us to impose an artificial gradient as a part of our EQL$^{\div}$ loss function that penalizes denominators in a manner proportional to how far below θ they are.

We now present the loss function $\mathcal{L}$ for our EQL$^{\div}$ model. First, consider an EQL$^{\div}$ model learning a function $\Psi : \mathbb{R}^n \to \mathbb{R}^m$ with L layers. We define $\Phi : \mathbb{R}^n \to \mathbb{R}^{2m}$ to be $\Phi(\mathbf{x}) = (T^{(L)} \circ \psi^{(L-1)} \circ \cdots \circ \psi^{(1)})(\mathbf{x})$; note that this definition coincides with that in the previous section, giving the function whose output is input into D. $\mathcal{L}$ for this model with respect to a dataset X of length N with predictors $\mathbf{x_i}$ and labels $\mathbf{y_i}$ (where the subscript now indexes the data point within the data set, instead of indicating a vector component) can now be written as

$$\mathcal{L}(\mathbf{X}) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} \left\|\Psi(\mathbf{x_i}) - \mathbf{y_i}\right\|_2^2}_{\substack{\text{Mean Squared Error} \\ \text{(Accuracy)}}} + \underbrace{\lambda \sum_{l=1}^{L} \left(\left\|\mathbf{W}^{(\mathbf{l})}\right\|_1 + \left\|\mathbf{b}^{(\mathbf{l})}\right\|_1 \right)}_{\substack{\text{L1 Regularization} \\ \text{(Sparsity)}}} + \underbrace{\sum_{i=1}^{N}\sum_{j=1}^{m} \max(\theta - \Phi(\mathbf{x_i})_{2j}, 0)}_{\substack{\text{Artificial Gradient} \\ \text{(Positive Denominators)}}}. \tag{6}$$

Note that the even-numbered indices of $\Phi(\mathbf{x_i})$ are precisely the denominators of that division layer. We can now also see that the regularization coefficient $\lambda \in \mathbb{R}$ is another hyperparameter for EQL$^{\div}$.

2.4 *EQL$^{\div}$ Training Schedule*

As outlined by the creators of this model [1, 2], we used a stochastic gradient-descent algorithm with mini-batches and optimized the loss function using the Adam method [4] as follows:

$$\mathbf{W_{t+1}} = \mathbf{W_t} + \text{Adam}(-\nabla\mathcal{L}(\mathbf{X}), \alpha), \tag{7}$$

where $\mathbf{W_t}$ denotes a vector containing all trainable weights and biases at an arbitrary training time step $\mathbf{t} \in \mathbb{N}$. The learning rate $\alpha \in \mathbb{R}$ used with Adam is another model hyperparameter.

We can motivate the EQL$^{\div}$ training schedule [1, 2] by identifying a particular problem implicitly contained in our loss function $\mathcal{L}$. Presently, our regularization penalty will both minimize the number of active (non-zero) weights and biases (i.e., impose sparsity) and minimize the magnitude of those weights and biases (i.e., impose "small" coefficients). This latter behavior is undesirable, as it means that our model will have some difficulty learning functions with large coefficients.

This problem with our loss function was mitigated by imposing the following three-phase dynamic training schedule. Given a training duration T, phase one of training lasts for $\frac{1}{4}T$, during

which we let $\lambda = 0$ (i.e., we impose no regularization). The goal of this phase of training is for the weights and biases to acquire plausible values when given some time to settle without restriction. Phase two of training lasts $\frac{7}{10}T$, and in this period we set λ to its predetermined, non-zero value and impose regularization on the trainable variables to establish sparsity. Finally, in phase three, weights and biases are allowed to reach previously inaccessible magnitudes through the brief re-elimination of regularization (i.e., setting $\lambda = 0$ once again) for the final $\frac{1}{20}T$ of training. To prevent trainable variables with small or zero values from reemerging and undoing the benefits of sparsity imposed during phase two, we impose the additional constraint that those small variables must maintain their total L0 norm; that is, we require all trainable variables with values greater than or equal to a threshold $c \in \mathbb{R}$ to remain non-zero, and all variables with values strictly less than c to become and remain zero.

As a final note, the original creators of EQL$^{\div}$, desiring the division threshold θ to be as small as possible while still recognizing the risk of divergence associated with sufficiently small values, chose to make θ a dynamic hyperparameter that varies with the epoch number $\mathbf{t}$, as follows:

$$\theta(\mathbf{t}) = \frac{1}{\sqrt{\mathbf{t}+1}}. \tag{8}$$

Eq. 8 allows θ to shrink as the model learns to avoid small and negative denominators, thanks to the "artificial gradient" in $\mathcal{L}$, as defined in Eq. 6.

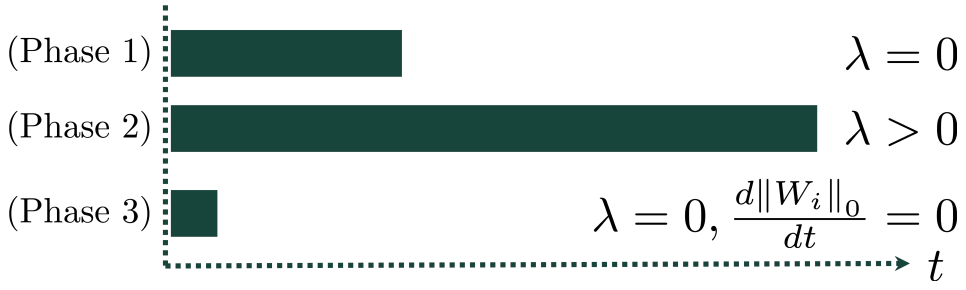

Figure 2. Visualization of the training schedule for EQL and EQL$^{\div}$, showing the dynamic regularization progression and the relative length of the three training periods.

3. RESULTS

To test our models, we chose two examples of physical systems that exhibit behavior that is simultaneously well understood and complex. In particular, we examined the single and double pendula. The equations of motion for both systems are well known and are not difficult to derive using the methods of Lagrangian mechanics, but in both cases, the equations of motion are nonlinear and do not possess general analytic solutions. In practice, the single and double pendula can be modeled through numerical integration methods.

Computational numerical integrators typically depend on a function $f : \mathbb{R}^n \to \mathbb{R}^n$ that captures the first-order ordinary differential equation (ODE) decomposition of a system's equations of motion. This function maps a "state vector," which contains complete information about a given system's state, to a vector which gives the time rate of change of each component in state vector, evaluated at that particular state; i.e., the ith component of $f(\mathbf{a})$ is $\frac{dx_i}{dt}|_{x_i=a_i}$ for a system defined by n variables $x_1, x_2, \ldots, x_n$. The state of a system is usually specified as a complete set of $\frac{1}{2}n$ generalized coordinates and either their time derivatives or their associated generalized momenta.

It is this function f, which efficiently contains the equations of motion of a given system, that we chose to ask our EQL$^\div$ model to learn for both the single and double pendula. Thus, we were able to evaluate the accuracy of our trained models by numerically integrating their learned equations and comparing the resulting time series data with those resulting from the numerical integration of the true equations of motion.

We find the EQL$^\div$ model to be extremely sensitive to hyperparameters, especially the regularization coefficient λ. Even in cases where EQL$^\div$ is tasked with learning a relatively simple function, we find that setting λ too low causes the EQL$^\div$ model to tend to learn functions with many nonlinear terms (i.e., overfitting the data), while setting λ too high restricts the number of terms in the learned function and eliminates the model's ability to capture subtleties and complexities (i.e., underfitting the data). We illustrate this point in Fig. 3 with an invented function and three functions learned by three separate trained instances of EQL$^\div$, each using a different value of λ. The function learned by the instance with the greatest regularization merely captures a global trend inside the training region (indicated by the vertical dotted lines), while the function learned by the instance with the smallest value of λ overfits the training region and cannot be extrapolated. Only the function learned by the model instance in which λ takes on an intermediate value is able to capture both local structure in the training region and a global trend.

3.1 *Single Pendulum*

Adopting overdot notation to denote derivatives taken with respect to time, the motion of the ideal single pendulum of unit length and unit mass is given by the following linear first-order ordinary differential equation for the angle θ that the pendulum makes with the vertical:

$$\ddot{\theta} + g\sin(\theta) = 0. \tag{9}$$

If we define $\omega := \dot{\theta}$, we observe that a single-pendulum state is uniquely determined by the vector (θ, ω). In this case, the function f, which captures the defining first-order differential equations, takes the form $f(\theta, \omega) : \mathbb{R}^2 \to \mathbb{R}^2$ and is defined as

$$f(\theta, \omega) = \begin{bmatrix} \omega \\ -g\sin(\theta) \end{bmatrix}. \tag{10}$$

However, we recall that one of the anticipated weaknesses of the EQL$^\div$ architecture is its difficulty with learning large coefficients. As a consequence, we follow a modification imposed

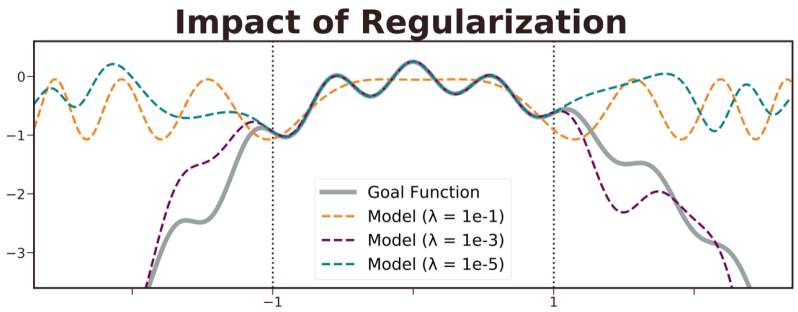

Figure 3. Visualizing the sensitivity of EQL$^{\div}$ to the regularization hyperparameter λ using an arbitrary function as the goal and plotting that goal along with three functions learned by three EQL$^{\div}$ models, each with a different λ value. In particular, we see how models with values for λ that are either slightly too high (10^{-1}) or too low (10^{-5}) end up underfitting or overfitting the data, respectively. This illustrates the sensitivity of EQL$^{\div}$ to regularization, which in turn necessitates the fine-tuning of the hyperparameter λ to each data set. Dotted vertical lines give the bounds of the region from which training data was sampled.

by the creators of EQL$^{\div}$, who used the single pendulum to test its predecessor, EQL [1, 2], and divide both components of f by g, giving the following, scaled-down function f' with easier-to-learn coefficients:

$$f'(\theta, \omega) = \begin{bmatrix} \frac{1}{g}\omega \\ -\sin(\theta) \end{bmatrix}. \tag{11}$$

Eq. 11 gives the function we trained the EQL$^{\div}$ model to learn. We also followed a data-generation procedure designed by the original creators of the EQL$^{\div}$ model [1]; this procedure involves sampling the hypercube $[-h, h] \times [-h, h] \subset \mathbb{R}^2,\ h \in \mathbb{R}$ for N predictors and using the goal function f' to compute their corresponding N labels.

Because of the simplicity of this function, we find that EQL$^{\div}$ is effective at learning it exactly, without too many data points or training epochs, across certain ranges of hyperparameter values. In an attempt to pinpoint the minimal amount of training data and the minimal training duration required, we completed a grid search across a range of values for the number of data points N and the training epoch T; our results are shown in Fig. 4 in the form of "heat maps" that depict the accuracy of individual trained models using color. As a metric, we used the root-mean-squared error (RMSE, the square root of the first term in Eq. 6) on an "extrapolation" data set containing 10000 data points, with predictors sampled from the rectangular hyper-shell $[-2h, 2h] \times [-2h, 2h] \setminus [-h, h] \times [-h, h] \subset \mathbb{R}^2$. We made the following hyperparameter selections for this grid search: $L = 2$, $4u^{(i)} = v^{(i)} = 4$, $H = \{\mathrm{id}, \sin, \cos, \mathrm{sigm}\}$, $I^{(i)}$ chosen by evenly sampling $\{1, 2, 3, 4\}$, $\theta = 10^{-3}$, $\lambda = 10^{-3}$, and $\alpha = 0.005$. Fig. 4 shows that as N increases, the variability in the accuracy of our training models also increases.

Further inspection of the training process reveals that in the third phase of training (the final $\frac{1}{20}T$), models often quickly converged to nearly perfect functions but failed to remain there, continuing to tweak coefficients, thereby varying in their root-mean-squared error scales.

To counter this tendency, we implemented an early stopping functionality during the final stage of training that automatically cut off training when the model stopped improving and restored the previous best set of weights and biases $\mathbf{W}_{\mathbf{t}_{\mathbf{best}}}$ from that stage. We find this modification to be immensely successful. Fig. 4 offers a comparison between models trained with and without early stopping, with one heat map shown for each category.

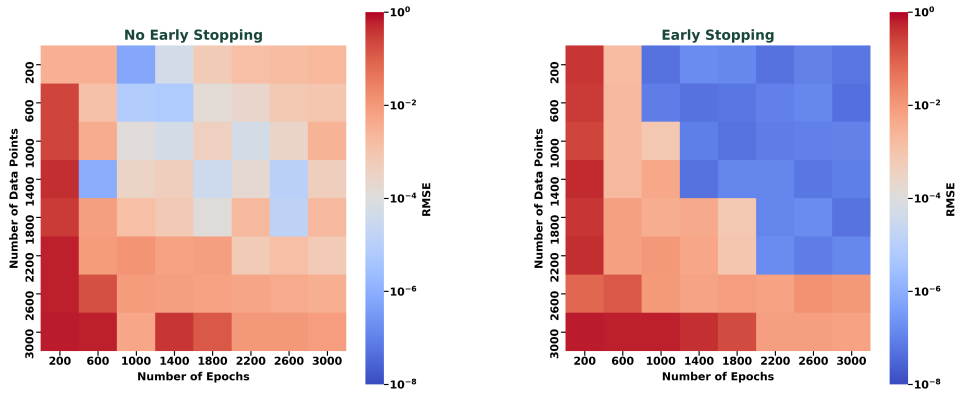

Figure 4. Visualizing the accuracy of equations learned by EQL$^{\div}$ models for different training data set sizes N and different training durations T through a heat map where each cell gives the average total RMSE on an extrapolation test data set across five EQL$^{\div}$ models trained at those hyperparameter settings. On the left we give a heatmap for models trained without early stopping, while on the right we give a heatmap for models trained with early stopping (in the final stage of training), demonstrating the effectiveness of early stopping and suggesting the propensity of the EQL$^{\div}$ model to locate and then stray away from absolute minima during the final stage of training.

Our experimentation with EQL$^{\div}$ in the single-pendulum system also reveals several interesting tendencies of the model, specifically with respect to the algebraic equivalencies or approximations it converges upon as local minima. In Fig. 5, we present examples of learned functions, which demonstrate a few of the more notable instances of such approximations. In this figure, the first component of f_2 illustrates the model's use of $\mathrm{sigm}(0)$, which equals $\frac{1}{2}$, but avoids incurring the loss-function cost of a non-zero weight resulting from regularization; the use of $\mathrm{sigm}(0)$ was consistently encountered. The first component of f_1 exemplifies how EQL$^{\div}$ approximates a line with small slope by a low-frequency sine curve; the second component of f_1 shows how EQL$^{\div}$ discovers and takes advantage of the identity $\sin(\theta) = \cos(\theta - \frac{\pi}{2})$. Finally, the second component of f_3 depicts how

EQL$^{\div}$ can locally approximate one period of $-\sin(\theta)$ somewhat effectively with a rational function.

$$f_1(\theta,\omega) = \begin{bmatrix} 1.08\sin\left(0.0946\omega\right) \\ -1.0\cos\left(1.0\theta - 1.57\right) \end{bmatrix}$$
$$f_2(\theta,\omega) = \begin{bmatrix} \frac{0.0591\omega}{0.177\,\mathrm{sigm}\,(0)+0.491} \\ -1.0\sin\left(1.0\theta\right) \end{bmatrix}$$
$$f_3(\theta,\omega) = \begin{bmatrix} 0.102\omega \\ -\frac{0.134\theta}{0.0367\theta^2+0.125} \end{bmatrix}$$

Figure 5. A few examples of close approximations to the goal function learned by instances of the EQL$^{\div}$ model trained for the single-pendulum system. These represent local minima that the model was unable to escape during the training process. We identify the capacity and tendency of insufficiently-trained EQL$^{\div}$ models to discover functional approximations, trigonometric identities, and loss function loopholes.

3.2 *Double Pendulum*

The motion of the double pendulum is significantly more complicated. The system often acts as a canonical example of chaos (which occurs when similar initial conditions do not necessarily result in similar time evolution), and the differential equations constituting its equations of motion represent a considerable step up in complexity and nonlinearity. As before, let us consider each individual pendulum in the double pendulum to have unit length and unit mass. If we let θ_1, θ_2 be the two angles which the two rods constituting the pendulum make with respect to the vertical (numbered from the top down) and define $\omega_1 := \dot{\theta}_1, \omega_2 := \dot{\theta}_2$, then we can write the function f that captures the first-order decomposition of the equations of motion for the double pendulum as

$$f(\theta_1,\omega_1,\theta_2,\omega_2) = \begin{bmatrix} \omega_1 \\ \frac{\omega_1^2\cos(\theta_1-\theta_2)\sin(\theta_1-\theta_2)+g\sin(\theta_2)\cos(\theta_1-\theta_2)-\omega_2^2\sin(\theta_1-\theta_2)-2g\sin(\theta_1)}{2-\cos^2(\theta_1-\theta_2)} \\ \omega_2 \\ \frac{\omega_2^2\cos(\theta_1-\theta_2)\sin(\theta_1-\theta_2)+2g\sin(\theta_1)\cos(\theta_1-\theta_2)-2\omega_1^2\sin(\theta_1-\theta_2)-2g\sin(\theta_2)}{2-\cos^2(\theta_1-\theta_2)} \end{bmatrix}. \tag{12}$$

Our data-sampling method mirrors that of the single pendulum, except that the hypercube now resides in $\mathbb{R}^4$.

We find that the double-pendulum case requires us to modify our original method for assessing the accuracy of our model results, the learned equations. Our experimentation—which included training on data sets with size up to $N = 50000$ and durations up to $T = 10000$, requiring up to 16 hours to complete—failed to reveal a hyperparameter configuration for which an EQL$^{\div}$ model

could exactly learn the desired equation. As a result, our evaluation must involve the qualification of the accuracy of the approximations to f learned by our models. We recall that the ultimate goal of our project is to discover learned equations to be used in computational modeling; it seems natural, then, to evaluate our models based on the accuracy of the computational models that use the learned equations. Thus, in the case of the double pendulum, we established a metric $\mathcal{E}(t)$ for the accuracy of a computational model at time t to be the Euclidean distance between the true and learned locations of the second mass (the one on the very end of the double pendulum) at time t, given by the numerical integration of the learned equation and the numerical integration of the true equations of motion at time t.

In Fig. 6, we select three of our trained models, differing with respect to the training-time variables N and T, and plot $\mathcal{E}(t)$ vs. time for two different initial conditions: one we call "low-energy" ($[\theta_1, \omega_1, \theta_2, \omega_2]^\top = [\frac{\pi}{4}, 0, \frac{\pi}{4}, 0]^\top$), and one we label "high-energy" ($[\theta_1, \omega_1, \theta_2, \omega_2]^\top = [\frac{\pi}{2}, 0, \frac{\pi}{2}, 0]^\top$). For the remainder of this section, "Model 1," "Model 2," and "Model 3" will refer to the models labeled as such in Fig. 6. The hyperparameters for the models were $L = 4$, $4u^{(i)} = v^{(i)} = 4$, $H = \{\mathrm{id}, \sin, \cos, \mathrm{sigm}\}$, $I^{(i)}$ chosen by evenly sampling $\{1, 2, 3, 4\}$, $\theta = 10^{-3}$, $\lambda = 10^{-3}$ and $\alpha = 10^{-3}$; the models were trained on data generated with the choice $h = 3$.

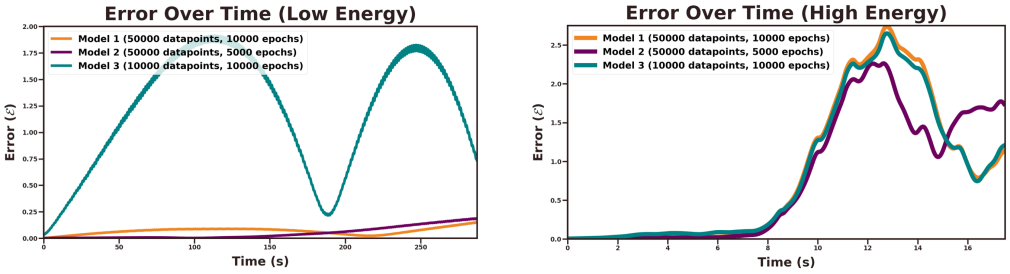

Figure 6. The error $\mathcal{E}(t)$ vs. time t for three instances of trained EQL$^\div$ models for two initial conditions: low energy ($[\frac{\pi}{4}, 0, \frac{\pi}{4}, 0]^\top$, pictured on the left) and high energy ($[\frac{\pi}{2}, 0, \frac{\pi}{2}, 0]^\top$, pictured on the right). We find that equations learned by sufficiently-trained EQL$^\div$ models can effectively approximate the time evolution of a double pendulum beginning in a low-energy configuration for extended periods, but are unable to approximate higher-energy double pendula undergoing chaotic motion are only approximated for longer than a few seconds.

In addition to measuring the accuracy of the computational model, we are also interested in learning to what extent the model discovers functions that adhere to natural laws, such as the conservation of energy. We now introduce two statistics that will enable us to evaluate the extent to which our models conserve energy. First, if we define $E(t)$ to be the total energy of the system at time t as predicted by a computational model, and we let E_0 be the total energy of the initial condition, we can then define the energy drift E_d of the predicted trajectory to be the slope of the linear approximation to the curve $\frac{E(t)-E_0}{E_0}$. This number represents the magnitude and direction of the global trend of the total energy of the modeled system and should ideally be 0.

We note, though, that $E_d = 0$ could be true for a predicted trajectory even if the trajectory does not conserve energy, if the system alternates or oscillates evenly between too much ($E(t) > E_0$) and too little ($E(t) < E_0$) energy. Cases such as this are captured with our second statistic, the energy fluctuation E_f, defined as

$$E_f = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(E(t) - E_0)^2}, \tag{13}$$

where $\mathcal{T}$ is the set of times for which the system state is computed, and $|\mathcal{T}|$ gives the number of elements in $\mathcal{T}$. By squaring the difference between the modeled system energy and the initial energy, we sum a measure of the magnitudes of the discrepancies over time and compute the average thereof. Together, E_d and E_f give an indication of the extent to which a model locally conserves energy from moment to moment for a given initial condition.

Models	Energy Drift E_d (Low Energy)	Energy Fluctuation E_f (Low Energy)	Energy Drift E_d (High Energy)	Energy Fluctuation E_f (High Energy)
Model 1	-1.24e-5 J	0.0694 J	-0.0139 J	4.38 J
Model 2	-4.90e-5 J	0.227 J	-0.0177 J	5.49 J
Model 3	5.93e-4 J	1.756 J	0.0859 J	2.21 J

Table 1. The energy drift E_d and energy fluctuation E_f for each model shown in Fig. 6 for both the low-energy and high-energy initial conditions. We find equations learned by sufficiently-trained models capable of conserving energy for trajectories with low-energy initial configurations of the double pendulum, but fail to achieve this for higher-energy initial configurations that lead to chaotic motion. These measures of energy conservation were computed over the duration indicated in Fig. 6. Videos showing these time evolutions can be found at bit.ly/2lfQ763.

Examination of the time evolution of the actual double pendulum from the low-energy initial condition reveals behavior that is more or less like that of a single pendulum; it oscillates back and forth in a relatively uniform way without exhibiting much chaos. The high-energy initial condition, however, devolves into highly chaotic motion after about eight seconds. This behavior is clearly shown in Fig. 6: in the low-energy plot, we find that the two more thoroughly trained models are able to maintain a high level of accuracy for a long period of time, while Model 3 appears to be moving essentially in and out of phase with the real double-pendulum behavior. However, in the high-energy plot, we observe a complete collapse in accuracy precisely at the eight-second mark; flaws in the approximations of all three models are placed on full display by the inherently chaotic nature of the double pendulum given this particular initial condition. Table 1 reveals a similar story: energy is conserved relatively well in the low-energy case, but we find greater deviations more quickly in the more strenuous test posed by the high-energy initial condition.

4. CONCLUSION

To summarize, we introduced multiscale physical systems and the challenges associated with their study and modeling, namely microscale simulation costs and the closure problem. We then presented some of our goals—objectivity, interpretability, and extrapolation—and used these to determine a suitable external tool to pursue to find a potential solution to our multiscale modeling problems. We chose to use symbolic function-learning neural networks, which we chose to explore, in particular, using the EQL$^\div$ model. We then introduced the architecture of neural networks and the extension of this structure to the EQL$^\div$ model. Next, we used our goals to motivate the terms in our loss function and described our training procedure. Finally, we presented our results generated by testing the EQL$^\div$ model on the equations of motion for the single and double pendula.

In conclusion, we find that EQL$^\div$ succeeds quickly in the simpler case of the single pendulum but can only approximate the more complex, chaotic double pendulum, with both accuracy and the preservation of physically "legal" behavior (in particular, conservation of energy) deteriorating rapidly with a higher-energy, more complex initial condition. In the context of multiscale modeling, the inherent complexity and chaotic nature of multiscale physical systems will likely prove even harder to resolve using EQL$^\div$ in its present state.

In addition to these general results, we identified the sensitivity of EQL$^\div$ to hyperparameters, especially the regularization coefficient λ, which essentially parameterizes the over- or under-fitting of the model; application of EQL$^\div$ to a complex system requires extensive searches for ideal hyperparameters. We also revealed several unexpected idiosyncrasies of the model that result from the nature of its architecture and training, such as the model's propensity to take advantage of its hypothesis set to dodge loss-function penalties, e.g., by using $\mathrm{sigm}(0) = \frac{1}{2}$ to "cheat" and generate a non-zero number without incurring a regularization cost. These observations grant valuable, non-trivial insights into ML when the "optimization space" can be treated as a function space, a characteristic of symbolic ML. In particular, we find that our function space is littered with local minima. Some of these local minima can be attributed to the very nature of function spaces, e.g., to the high number of functional approximations or equivalencies that exist for any given goal function; other local minima arise as a consequence of particular choices, made during the design of the model architecture and training, that arbitrarily and unintentionally favor certain functional forms, e.g., those using $\mathrm{sigm}(0)$ instead of $\frac{1}{2}$.

We conclude that EQL$^\div$ is a potentially capable model with some demonstrated success, but it may not scale well with increasingly complex systems, particularly those that may be encountered in multiscale modeling, because of the noted hyperparameter sensitivity and optimization challenges. These results lead us to consider modifying or moving past EQL$^\div$ as we proceed toward our multiscale modeling goal. Further analysis of the function space we are working with will help us to identify and develop the particular architectural or training modifications necessary for a more effective symbolic neural network. We also recall that EQL$^\div$ is not structurally physics-informed; in future research, the fundamental mathematical features exhibited by physical systems can be used

to design more specific, physically optimized strategies [5–7, 15] that can be used either in tandem with or independent of EQL$^{\div}$.

References

[1] Georg Martius and Christoph H. Lampert, Extrapolation and learning equations, abs/1610.02995, 2016, http://arxiv.org/abs/1610.02995

[2] Subham S. Sahoo and Christoph H. Lampert and Georg Martius, Learning Equations for Extrapolation and Control, abs/1806.07259, 2018, http://arxiv.org/abs/1806.07259

[3] Robert Tibshirani, Journal of the Royal Statistical Society. Series B (Methodological), 1, 267–288, Royal Statistical Society, Wiley, Regression Shrinkage and Selection via the Lasso, 58, 1996

[4] Adam: A method for stochastic optimization, Kingma, Diederik P and Ba, Jimmy, arXiv preprint arXiv:1412.6980, 2014

[5] Schmidt, Michael and Lipson, Hod, Distilling Free-Form Natural Laws from Experimental Data, Vol. 324, 5923, 81-85, 2009, 10.1126/science.1165893, American Association for the Advancement of Science

[6] On Learning Hamiltonian Systems from Data, Bertalan, Tom and Dietrich, Felix and Mesić, Igor and Kevrekidis, Ioannis G, arXiv preprint arXiv:1907.12715, 2019

[7] Sam Greydanus and Misko Dzamba and Jason Yosinski, Hamiltonian Neural Networks abs/1906.01563, 2019, http://arxiv.org/abs/1906.01563, arXiv, 1906.01563

[8] Durbin, Richard and Rumelhart, David E., 10.1162/neco.1989.1.1.133, Neural Computation, 1, 133-142, Product Units: A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks

[9] Y. Shin and J. Ghosh, The pi-sigma network: an efficient higher-order neural network for pattern classification and function approximation, 1991, 13-18 vol.1, 10.1109/IJCNN.1991.155142

[10] Sum-product networks: A new deep architecture, 9781467300612, 10.1109/iccvw.2011.6130310, 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Poon, Hoifung and Domingos, Pedro, 2011

[11] Koza, John R.,10.1007/BF00175355, 1573-1375, Statistics and Computing, 87-112, Genetic programming as a means for programming computers by natural selection, 1994, https://doi.org/10.1007/BF00175355

[12] Machine learning closures for model order reduction of thermal fluids, San, Omer and Maulik, Romit, Applied Mathematical Modelling, Vol. 60, 681-710, 2018

[13] Deep multiscale model learning, Wang, Yating and Cheung, Siu Wun and Chung, Eric T and Efendiev, Yalchin and Wang, Min, arXiv preprint arXiv:1806.04830, 2018

[14] Mjolsness, Eric and DeCoste, Dennis, 10.1126/science.293.5537.2051, 0036-8075, 5537, 2051-2055, American Association for the Advancement of Science, Machine Learning for Science: State of the Art and Future Prospects, Vol. 293, 2001

[15] Maziar Raissi and Paris Perdikaris and George E. Karniadakis, Physics Informed Deep Learning (Part I: Data-driven Solutions of Nonlinear Partial Differential Equations, abs/1711.10561, 2017, 1711.10561

[16] Physics-Informed Kriging: A Physics-Informed Gaussian Process Regression Method for Data-Model Convergence, Yang, Xiu and Tartakovsky, Guzel and Tartakovsky, Alexandre, arXiv preprint arXiv:1809.03461, 2018

[17] Variational Multiscale Closures for Finite Element Discretizations Using the Mori-Zwanzig Approach, Pradhan, Aniruddhe and Duraisamy, Karthik, arXiv preprint arXiv:1906.01411, 2019

[18] Heterogeneous multiscale method: A general methodology for multiscale modeling, E, Weinan and Engquist, Bjorn and Huang, Zhongyi, Phys. Rev. B, Vol. 67, Issue 9, 092101, 2003

[19] Marsland, Stephen, Machine Learning: An Algorithmic Perspective, Second Edition, 2014, 1466583282, 9781466583283

STUDENT JOURNAL OF PHYSICS

Optimizing an Electrospun Catalyst in Polyelectrolyte Membrane Fuel Cells

Surya Rajan[1]

[1]Senior, California High School, San Ramon, CA, 94583, USA

ABSTRACT

Polyelectrolyte Membrane Fuel Cells (PEMFCs) have been of great interest as a potential source of alternative energy due to their high-power output and zero-emission activity, yet their low cost efficiency relative to combustion engines has impeded commercial success. While the Pt/C catalyst is traditionally deposited on the electrodes of the PEMFC, electrospinning Pt/C has been recently used to increase electrochemically active surface area and proton conductivity in the fuel cell. This study explores optimizing the electrospinning of Pt/C onto the commercially used Nafion 117 membrane and fine tuning the deposition and composition of the nanofiber structures to maximize electrochemically active surface area and proton conductivity. By varying the electrospinning flow rate and Pt/C weight percentage, the peak power density achieved by the 32.5% wt. Pt/C nanofibers electrospun at 0.5 mL/hr indicated an optimal fiber diameter of approximately 1.25 μm. For all Pt/C wt. %, the 0.5 mL/hr nanofiber-coated membranes performed better than or equal to the 1.0 mL/hr nanofiber-coated membranes in terms of power density, supporting the agglomeration reduction picture derived from SEM imaging. Overall, tests showed a 62.5% increase in maximum power density with 32.5% wt. Pt/C nanofibers extruded at 0.5 mL/hr onto Nafion 117 membranes when compared with commercially used Nafion 117 membranes.

Keywords: *Polyelectrolyte Membrane Fuel Cells (PEMFCs), Pt/C catalyst, electrospinning, nanofiber structures*

1. INTRODUCTION

With the human population currently predicted to exceed 9.8 billion by 2050 and non-renewable resources being depleted at an alarming pace, renewable energy is increasingly being researched and implemented globally to prepare for the energy demands of future generations [1, 2]. Hydrogen Fuel Cells (HFCs), an emerging field of clean energy, have been singled out as a potential alternative to combustion engines. Like lithium-ion batteries, they produce no carbon emissions, yet hydrogen fuel cells have an energy to weight ratio ten times greater than lithium-ion batteries, much greater range, portability, and refuel times similar to combustion engines. Among fuel cells, Polyelectrolyte Membrane Fuel Cells (PEMFCs) are particularly attractive due to their high power density, low weight, low temperature operation, and low volume compared to other fuel cells [3]. However, PEMFCs have suffered commercially due to their lack of cost efficiency relative to combustion engines and other traditional sources of energy and power. Problems plaguing PEMFCs are largely the result of the sluggish and expensive platinum-based catalysis of the oxidation and reduction reactions in the membrane electrode assembly (MEA) of the cell [4]. While some studies have focused on non-platinum metal group catalysts to decrease costs, much research has concerned optimizing the deposition of the platinum catalyst in the fuel cell to maximize its catalytic activity per unit of mass deposited [5,6]. Recently, electrospinning has been taken up as a potential method to increase the maximum power density of a PEMFC. In the process of electrospinning, a high voltage is applied to a certain polymer solution volume, which then results in polymer nanofibers being deposited onto a substrate. Electrospinning is typically used to finely tune nanofiber morphology, and because of this unique characteristic, recent PEMFC research has been focused on using this technique to increase contact area between the catalyst nanofibers to increase proton conductivity and electrochemically active surface area. This paper focuses on optimizing this process by varying two parameters: flow rate and Pt/C weight percentage in the resulting nanofibers formed from a polymer solution.

2. METHODS

Four different polymer solutions consisting of deionized water, isopropanol (IPA), poly(acrylic acid), Pt/C, LIQUION™ Nafion solution with 15% wt. Nafion resins, 40% wt. deionized water, and 45% wt. IPA were prepared. The Nafion solution was used to allow for increase adherence of the catalyst electrospun layer onto the Nafion 117 membrane. PAA was used as a carrier polymer critical in any electrospinning process, while water and IPA were used as solvents. Each of these solutions contained a different Pt/C weight percentage so that once electrospinning of each of the solutions was completed, the Pt/C weight percentages of the catalyst nanofiber layer masses were 20.0%, 25.0%, 32.5%, and 40.0%. Each of these solutions were then electrospun at a voltage of 15.0 kV onto a Nafion 117 membrane substrate at two different flow rates: 0.5 mL/hr and 1.0 mL/hr. This resulted in a total of eight different samples for each Pt/C weight percentage, two per Pt/C weight percentage. Each of these samples were then tested in an H-Tech PEMFC Demo Kit at a hydrogen flow rate of 80 cubic centimeters per minute at the anode and an open-air environment at the cathode. After testing, untested samples of nanofiber catalyst layers for each of the eight samples were then analyzed through Optical Microscopy, Laser Microscopy, 3D Laser Microscopy, and Scanning Electron Microscopy (SEM).

3. RESULTS

The Nafion-PAA-Pt/C nanofiber structures were characterized using Optical Microscopy at 100x magnification. The Nafion and PAA are represented by the lighter colors in the images shown in Figure 1. The figure shows that as Pt/C weight percentage is increased, more darker colors are shown in the images, as the proportion of Nafion and PAA in the nanofibers is lesser. In addition, the images also show that the nanofibers electrospun at a lower flow rate have a more even dispersion of lighter and darker colored regions and finer arrangements, implying that electrospinning at a lower flow rate results in a more even dispersion of Pt/C, Nafion, and PAA that results in finely-tuned nanofiber structures. Fibers with 40.0% wt. Pt/C were observed to be much more interwoven and many followed circuitous paths. This correlation between Pt/C wt. % and change in fiber orientation may indicate that more viscous solutions and samples with higher Pt/C concentrations are less optimal for electrospinning as they either did not extrude from the syringe smoothly or were not caught on the rotating drum efficiently.

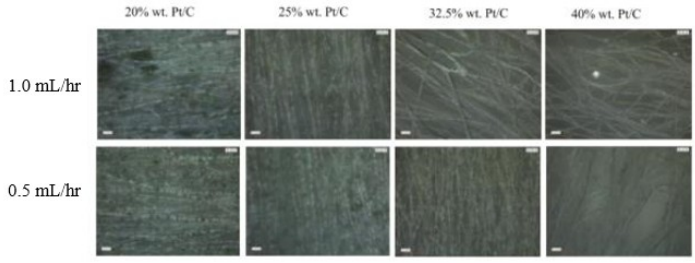

FIG. 1. Optical Microscope images at 100x magnification of 20.0% wt., 25.0% wt., 32.5% wt., and 40.0% wt. Pt/C at electrospinning flow rates of 0.5 mL/hr and 1.0 mL/hr.

The nanofiber diameters were determined through Laser Microscopy. Among the nanofibers electrospun at a flow rate of 1.0 mL/hr, the diameters of each Pt/C weight percentage nanofibers were in increasing order, respectively, 0.920 microns, 1.086 microns, 1.680 microns, and 2.024 microns. The nanofibers electrospun at a flow rate of 0.5 mL/hr had diameters in increasing order of, respectively, 0.695 microns, 1.062 microns, 1.251 microns, and 1.863 microns (Figure 2).

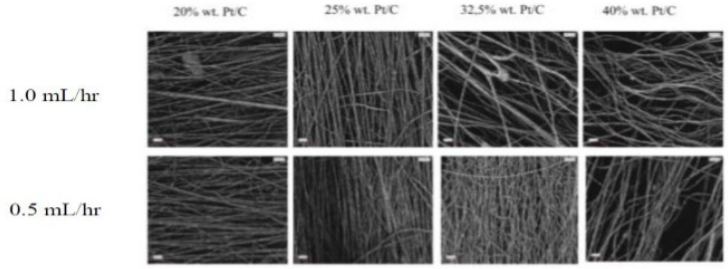

FIG. 2. Laser Microscope images at 100x magnification of 20.0% wt., 25.0% wt., 32.5% wt., and 40.0% wt. Pt/C at electrospinning flow rates of 0.5 mL/hr and 1.0 mL/hr.

Additionally, 3D Laser Microscopy was used to characterize the agglomeration of Nafion, PAA, and Pt/C across multiple layers. As Pt/C weight percentage increased, higher average nanofiber heights were observed for both flow rates. As electrospinning flow rate increased, higher average nanofiber heights were also observed (Figure 3).

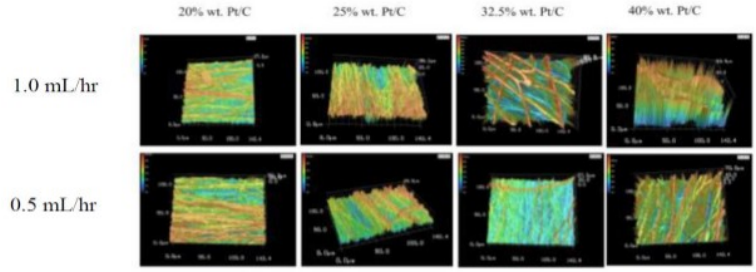

FIG. 3. 3D Laser Microscope images at 100x magnification of 20.0% wt., 25.0% wt., 32.5% wt., and 40.0% wt. Pt/C at electrospinning flow rates of 0.5 mL/hr and 1.0 mL/hr.

Scanning Electron Microscopy (SEM) and Energy-Dispersive X-Ray Spectroscopy (EDS) were conducted on nanofiber samples of varying Pt/C weight percentages: 20.0%, 25.0%, and 30.0%, and 40.0% to obtain clear evidence of platinum deposition increase on the nanofibers as platinum weight percentage increased. As shown in the following figures, with increases in Pt/C weight percentage, there is an increase in platinum content seen both in the SEM images with an electron high tension (EHT) of 2.5 kV and in the EDS spectra, as the absorption of X-rays by the platinum at a wavelength producing about 2.0 kV shows an increase in intensity (Figure 4, Figure 5, Figure 6, and Figure 7). More detailed SEM images of the nanofibers with 32.5% wt. Pt/C are shown in Figure 8 and Figure 9.

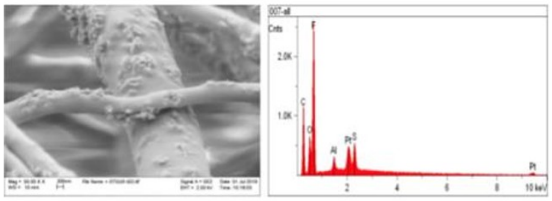

FIG. 4. SEM images and EDS spectra for 20.0% wt. Pt/C nanofibers electrospun at a flow rate of 1.0 mL/hr

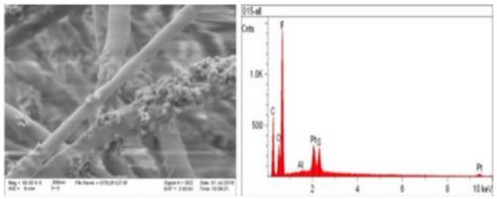

FIG. 5. SEM images and EDS spectra for 25.0% wt. Pt/C nanofibers electrospun at a flow rate of 1.0 mL/hr

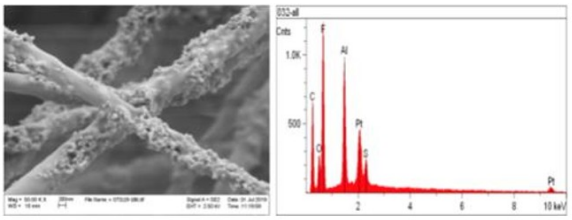

FIG. 6. SEM images and EDS spectra for 30.0% wt. Pt/C nanofibers electrospun at a flow rate of 1.0 mL/hr

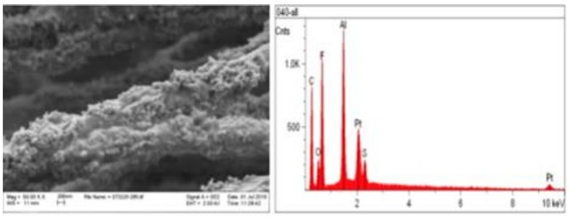

FIG. 7. SEM images and EDS spectra for 40.0% wt. Pt/C nanofibers electrospun at a flow rate of 1.0 mL/hr

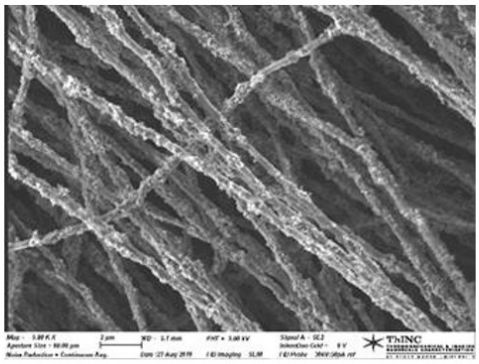

FIG. 8. SEM images of 32.5 Pt/C wt. % nanofibers electrospun at 0.5 mL/hr

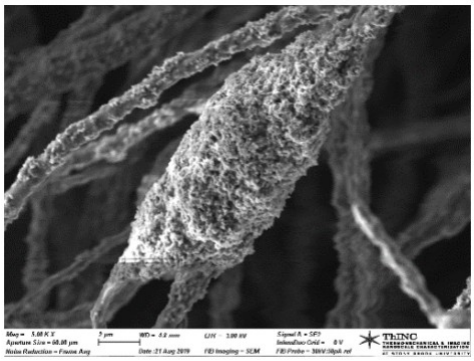

FIG. 9. SEM images of 32.5 Pt/C wt. % nanofibers electrospun at 1.0 mL/hr

An open-circuit voltage (OCV) of approximately 1.0 V was achieved, and the resulting polarization curves for 32.5% wt. Pt/C nanofibers electrospun at flow rates of 0.5 mL/hr and 1.0 mL/hr were produced and compared with the control. A maximum power density of 0.013 W/cm^2 was observed for the 32.5% wt. Pt/C nanofibers electrospun at 0.5 mL/hr while a maximum power density of 0.008 W/cm^2 was observed for a commercial PEMFC with a pure Nafion 117 membrane (Figure 10). Ultimately, a 62.5% increase in power density was observed from the commercial PEMFC to a 32.5% wt. Pt/C nanofiber-coated Nafion 117 membrane with fibers electrospun at a volumetric flow rate of 0.5 mL/hr on one side (Figure 11). Although the Nafion 117 membrane coated with 32.5% wt. Pt/C nanofibers electrospun at volumetric flow rate of 1.0 mL/hr displayed a lower maximum power density than its 0.5 mL/hr counterpart, it still showed a maximum power density of 0.010 W/cm^2, a 25% increase from the commercial PEMFC (Figure 12).

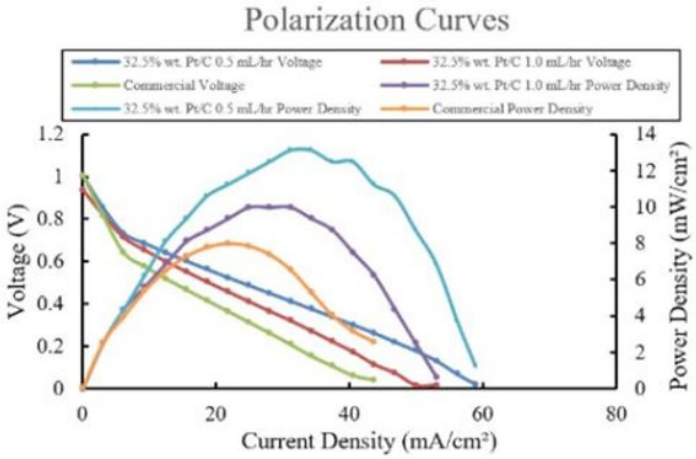

FIG. 10. Power density and polarization curves of PEMFCs with 32.5% wt. Pt/C nanofibers electrospun onto membranes at varied flow rates

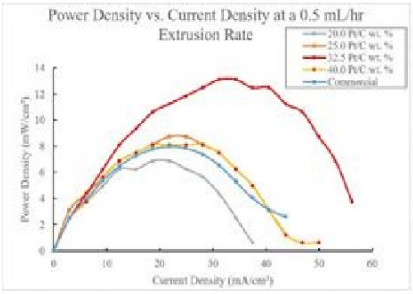

FIG. 11. Power density versus current density curves of PEMFCs with varying Pt/C wt. % nanofibers electrospun onto membranes at 0.5 mL/hr

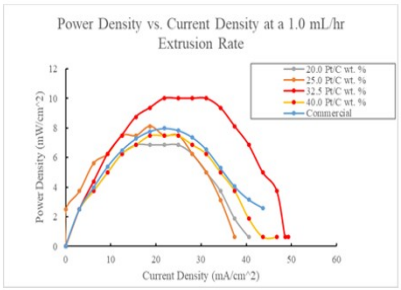

FIG. 12. Power density versus current density curves of PEMFCs with varying Pt/C wt. % nanofibers electrospun onto membranes at 1.0 mL/hr

4. CONCLUSION

Overall, PEMFC demonstration kit testing indicates that electrospinning Pt/C-Nafion PAA nanofibers onto the Nafion membrane has the capacity to increase PEMFC maximum power density by up to 62.5%, suggesting the promise of this catalyst application technique. By comparing the polarization curves of membranes coated with nanofibers of different flow rates and Pt/C weight percentages, it was found that the optimal Pt/C weight percentage was approximately 32.5%, meaning an approximate balance between platinum agglomeration, platinum content, and nanofiber diameter is achieved at that weight percentage. The optimal flow rate tested was 0.5 mL/hr, and the performance of the nanofibers electrospun at varying flow rates indicates that as flow rate decreases, PEMFC maximum power density increases. This trend can be attributed to a decrease in nanofiber diameter and platinum agglomeration, which allows for more finely-tuned nanofibers. Microscopy imaging of the nanofibers revealed key trends and characteristics which impacted their performance as a catalyst when electrospun onto the Nafion 117 membrane. As the Pt/C wt. % was increased from 20.0% to 40.0%, platinum content increased alongside platinum agglomeration and nanofiber diameter. The optimal performance of the 32.5 Pt/C wt.% nanofiber-coated membranes demonstrated an approximate balance between these performance enhancing and limiting parameters. The 0.5 mL/hr flow rate nanofibers exhibited lower levels of platinum agglomeration and

lower fiber diameters compared to the 1.0 mL/hr nanofibers, indicating that as flow rate decreases, performance of the nanofiber-coated Nafion 117 membranes would increase. However, more flow rates must be tested to confirm this trend, as there may be departures occurring at flow rates not tested in this study.

5. ACKNOWLEDGEMENTS

The author of this article gratefully acknowledges the following: The Garcia Center for Materials at Engineered Interfaces at Stony Brook University; Dr. Stoyan Bliznakov for his valuable knowledge and advice; the National Science Foundation and Morin Foundation and other sponsors for the chance to explore the fascinating power of materials science and engineering; and Garcia Alumni for their support.

REFERENCES

[1] B. R. Singh and O. Singh, "Global Trends of Fossil Fuel Reserves and Climate Change in the 21st Century," Fossil Fuel and the Environment, (2012).
[2] "World Population Projected to Reach 9.8 Billion in 2050, and 11.2 Billion in 2100 | UN DESA Department of Economic and Social Affairs," United Nations, United Nations,
(2017).
[3] "Hydrogen Fuel Cells vs Lithium-Ion Batteries in Electric Vehicles," FuroSystems, (2018).
[4] S. J. Paddison and H. A. Gasteiger, "PEM Fuel Cells, Materials and Design Development Challenges," Fuel Cells, (2012), pp. 341–367.
[5] H. M. Barkholtz, et al. "Enhanced Performance of Non-PGM Catalysts in Air Operated PEM-Fuel Cells," Int. J. Hydrogen Energ., (2016), Vol. 41, no. 47, pp. 22598–22604.
[6] B. Lim, et al. "Pd-Pt Bimetallic Nanodendrites with High Activity for Oxygen Reduction," Science, (2009), Vol. 324, no. 5932, pp. 1302–1305.

STUDENT JOURNAL OF PHYSICS

Comparative analysis of Particulate Matter

Hiral Ponda[1]

[1]M.Sc., Sem 3, Department of Physics and Electronics, St.Xavier's College, Ahmedabad -380009
hiral.ponda.7@gmail.com

ABSTRACT

This study focuses on the comparative analysis of Particulate Matters (PM) – PM_{10} and $PM_{2.5}$ between Ahmedabad (AMD) and Gandhinagar (GND) city in Gujarat for the year 2016 -2018. Seasonal and annual analysis were carried out in order to understand the trends and sources of pollution within the city. Further PM ratio analysis is also performed to identify contribution of finer and coarser mode aerosol. The results show that industrial areas contribute the most; Followed by commercial and traffic junctions in both the cities. Ratio analysis of different PM sizes is calculated as PM 2.5/10 and PM 1/2.5. Among which, PM 2.5/10 ratio is found highest during the monsoon as larger super micron particles get removed fast during monsoon than submicron particles due to washout effect. While in summer, it is the lowest due to increase in temperature and moderate wind speed. The ratio of PM 1/2.5 is highest during winter, owing to the fact that the combustion of heating appliances such as boilers, closed stoves, open fire places emit more fine particles.

Keywords: *Particulate matter, Air quality, Ratio analysis, Different Season.*

1. INTRODUCTION

As we all know that the earth's atmosphere is primarily made up of gases. Out of which, around 78.09% is nitrogen, 21% oxygen, and the rest of it is made up of other gases. Also, our atmosphere contains liquid droplets and solid particles which are very small in size, known as particulate matter (PM). Aerosol is the particulate matter which is suspended in a gas. With naked eye we can see some particles, such as dust, dirt, soot or smoke but others are so small that only an electron microscope can detect them.

According to the European committee for standardization (CEN), the different categories of particulate matter are: Inhalable, Thoracic, Alveolic and Respirable fraction. Airborne particles which can be aspirated into the air nose or mouth are known as inhalable particles. With increasing particle diameter, inhalable capacity decreases gradually with reaching a level of about 50% at 100 microns.

A dust mass fraction is classified into emission PM10, PM2.5 and PM1. Coarse particles having a diameter between 2.5 and 10 microns are known as Particulate matter 10(PM10). Fine particles with a diameter of 2.5μm or less are classified as Particulate matter 2.5. Ultrafine particles generally up to 0.1μm, if exposed to the condensation or coagulation, can inflate into the size of 1μm. Such particles are defined as PM1.

It has been observed that health is highly associated with the exposure to particle pollution. It can even lead to premature death. These health issues may include cardiovascular effects such as cardiac arrhythmias and heart attacks as well as respiratory effects such as asthma attacks and bronchitis. As a result, it can result in absences from school or work, emergency room visits, high hospital admissions and restricted activity days, importantly for those with pre-existing heart or lung disease, children and older people.

The size of particles are is directly connected to their prospect for causing health problems. Out of which, Fine particles (PM 2.5) have the greatest health risk. Once inhaled, such fine particles can go deep into the lungs. They can even get mixed into the bloodstream. If these particles are exposed to a person, they can affect his lungs and heart. Although coarse particles (PM 10-2.5) have less effect, they can affect a person's eyes, nose and throat. The EPA (Environmental Protection Agency) has standardized about who is at risk from the exposure to fine and coarse particles and includes simple measures that can be taken to reduce health risk.

2. STUDY AREA

We have divided Ahmedabad into 33 locations as shown below:

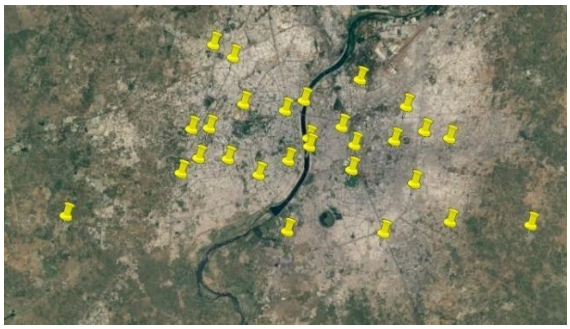

Figure 1: Study Area of Ahmedabad. The Yellow markers shows the selected location for data acquisition. Total 33 locations.

We have divided Gandhinagar into 21 locations as shown below:

Figure 2: Study Area of Gandhinagar. The blue markers shows the selected location for data acquisition. Total 21 locations.

3. INSTRUMENTATION

In this study, we have used GRIMM Portable Laser Aerosol Spectrometer and Dust Monitor Model 11-A (GRIMM Aerosol Technik GmbH & Co. KG, Germany), to analyze particle size distribution [Figure 3]. For continuous measurement of airborne particles and for measuring the particle count distribution GRIMM Aerosol Spectrometer is very useful. This instrument has four operational modes: 1) Environmental, 2) Occupational Health, 3) Mass Distribution and 4) Count Distribution. This instrument measures the concentration of particles in an optical sensor for the size from 0.25μm to 32μm in 32 different channels in different sizes, with concentration range of mass from 1-100000μg/m^3 (for mass distribution, environmental and occupational health modes) or a 1-2000000 particles/L (for count distribution mode). The sensitivity of these instruments is 1 particle/L for count mode and 1μg/m^3 for mass mode. Reproducibility of an instrument is 2%. In this study, Aerosol Spectrometer was operated in count distribution mode to produce count distribution versus time with a temporal resolution of 6 sec. The size of the particle and the wavelength of the incident light determine the strength of the light-matter interaction. The matter particles can be either a solid or a droplet or a gas molecule. To show this dependency, the parameter α is used.

$$\alpha = \pi \, d_p/\lambda \qquad (1)$$

d_p= Particle diameter, λ=incident wavelength,

π=Particle circumference for spherical particles.

This value α defines the type of scattering. For Particles much smaller than the incident wavelength (α<<1) Rayleigh scattering occurs. For particle size greater than the incident wavelength (α>>1), rays of light hitting the particle go through reflection, refraction and absorption, while the rays passing around the particles edge give rise to diffraction.

4. METHODOLOGY

The aerosol inlet or custom designed air inlets are used to lead the sample air into the measuring cell, e.g. for high wind speeds or overpressure.

Later, particles in the sample air were detected by light scattering inside the measuring cell. The scattering light pulse of every single particle gets counted and the intensity of its scattering light signal is then classified to a certain particle size.

The measuring principle is schematically shown in figure 3. The general light source of Grimm laser aerosol spectrometers and dust monitors is a laser diode. The wavelength is in visible range at 660 nm for model 11-A. The laser diode can operate in a so called Multiplex Mode in which the intensity of the laser beam is being modulated. Thus, particles can be detected over a very wide size range from 0.25μm up to 32μm.

On a flat elliptical strip, by means of illumination optics the laser beam is focused. The laser beam lights a small measuring volume evenly and subsequently inside the focus, then led into a light trap. After that, the sample air is focused aerodynamically and passed as particle flow through the inner area of the measuring volume. The particle concentration of the sample air is normally so low during the environmental measurement, that statistically seen only one particle is in the measuring volume. Very high particle concentrations can appear while measuring at particle sources, technical particulate matters, or working places, which later require a previous dilution of the sample air. As the entire sampling volume of 1.2 liter/minute gets analyzed, all spectrometers of Grimm aerosol can reach a very good counting statistic. Under a scattering angle of 90°, the scattering light emitted by every particle would be detected by a second optics and then directed to a receiver diode through a wide-angle 11 mirror. The signal of the detector can then be classified into different size channels after amplification subject to its intensity.

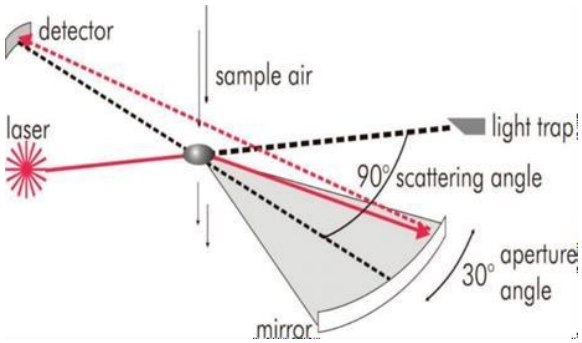

Figure 3: Measuring Principle

5. RESULT AND DISCUSSION

The table shows the standard values of PM 10 and PM 2.5 for residential and industrial area of Ahmedabad and Gandhinagar by NAAQS.

Standard values	**Other areas (Residential, open ground)**	**Industrial area**
PM2.5(µg/m³)	40µg/m³	60µg/m³
PM10(µg/m³)	80µg/m³	100µg/ m³

Table 1: Standard Values of PM

Next graphs are for PM 10 for the different seasons:

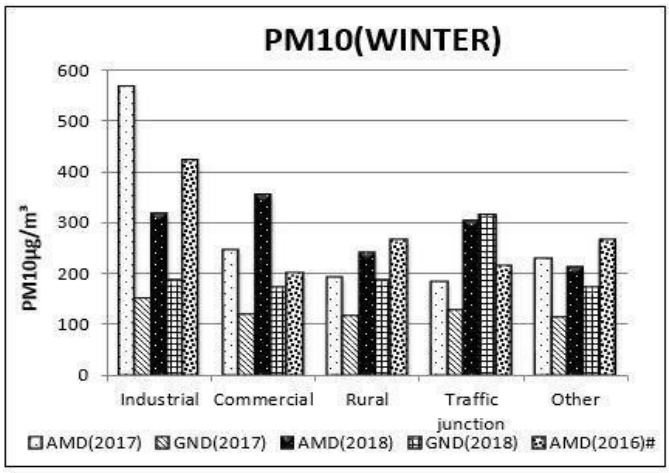

Figure4a: Concentration of PM10 for Ahmedabad and Gandhinagar

As shown in figure 4a, the concentration of PM 10 went as high as 564.55 µg/m^3 in Ahmedabad for the year 2017 (winter) at industrial area mainly due to high emission rate of factories.

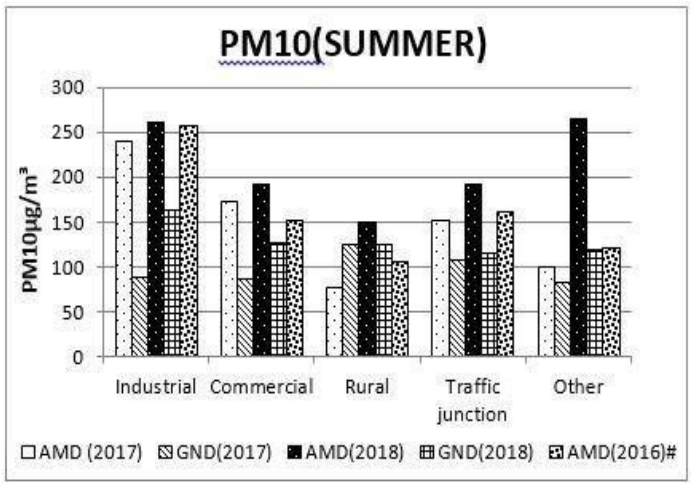

Figure4b: Concentration of PM10 for Ahmedabad and Gandhinagar

In figure 4b, the concentration of PM10 is hardly crossing the value 250 µg/m^3 for summer, possibly due to high temperature, high wind speed and low humidity concentration in the air.

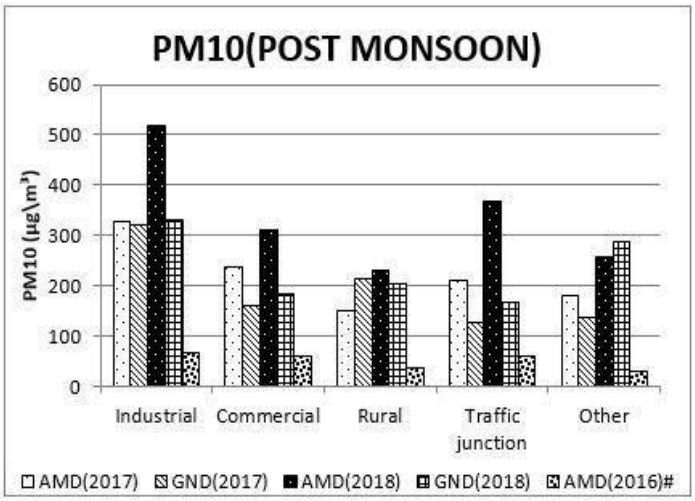

Figure4c: Concentration of PM10 for Ahmedabad and Gandhinagar

Figure 4c shows that the maximum concentration of PM 10 measured in Ahmedabad for the year 2018 (Post Monsoon) is 517.79µg/m^3 at industrial area due to high emission rate of factories.

The same figure shows the Minimum concentration of PM10 of 30.83 µg/m^3 in Ahmedabad for the year of 2016(Monsoon) at other areas due to washout of PM10 particles in the rain.

Below graphs show the data for PM2.5: From the data analysis, we can see that the concentration of PM2.5 was highest in industrial area in Ahmedabad 2018 in the season of post monsoon (Fig. 5c) which is 224.05 µg/m^3 for the same reason as for the PM10. Lowest concentration of PM2.5 was measured in other area in monsoon (Fig. 5c) at Ahmedabad during the year 2016 because of the washout effect.

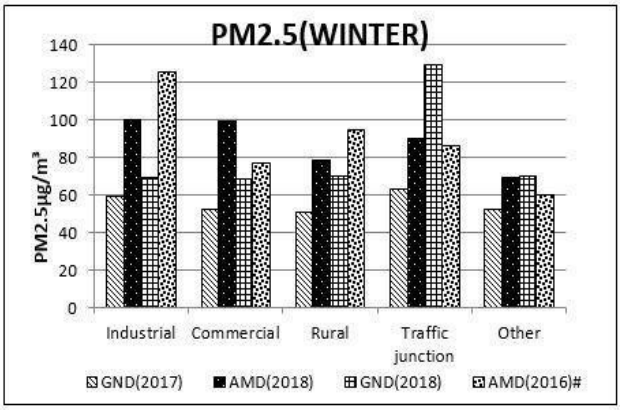

Figure 5a: Concentration of PM2.5 for Ahmedabad and Gandhinagar

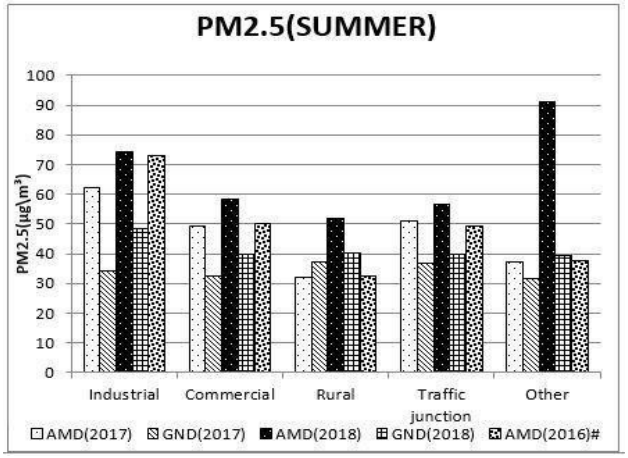

Figure 5b: Concentration of PM2.5 for Ahmedabad and Gandhinagar

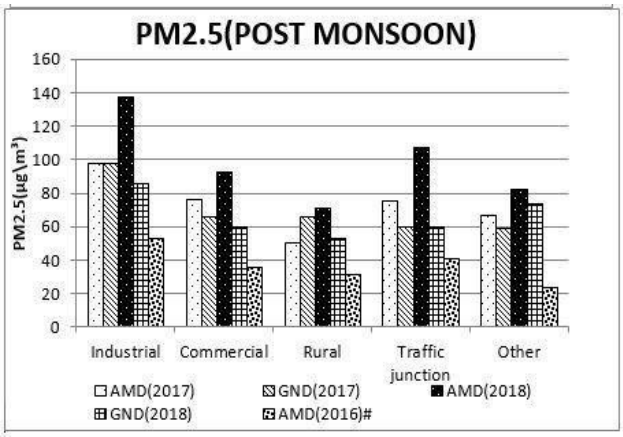

Figure 5c: Concentration of PM2.5 for Ahmedabad and Gandhinagar

6. RATIO ANALYSIS WITH CONCLUSIONS:

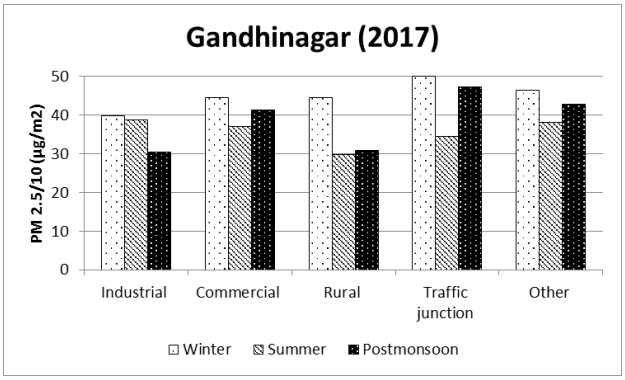

Figure 6a: Ratio of PM2.5/10

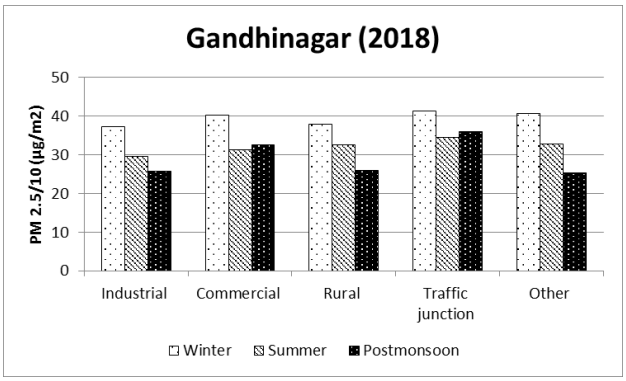

Figure 6b: Ratio of PM2.5/10

For Gandhinagar, Fig. 6a and 6b show that the PM 2.5/10 is highest during the winter season compared to the summer and post monsoon, possibly owing to the fact that the combustion of heating appliances such as boilers, stoves, and open fire places which emit more PM2.5 compared to PM10.

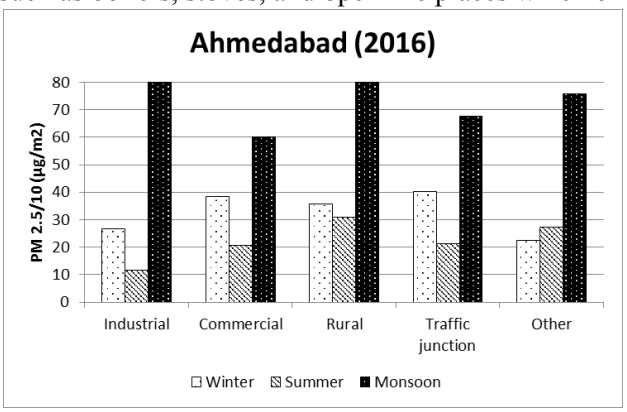

Figure 7a: Ratio of PM2.5/10

While for Ahmedabad, Fig. 7a show that, PM 2.5/10 ratio is highest in 2016 during the monsoon as larger super micron particles (PM10) get removed faster than the submicron particles (PM2.5) due to the washout effect.

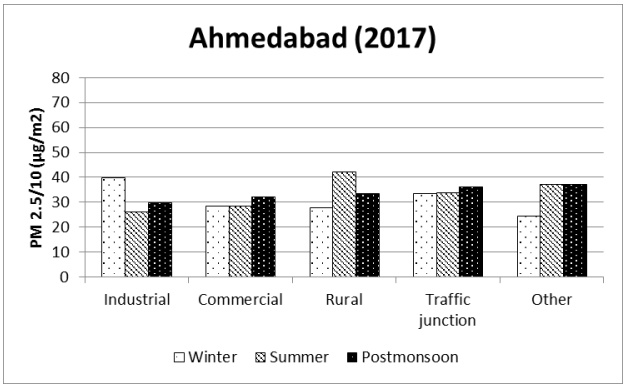

Figure 7b: Ratio of PM2.5/10

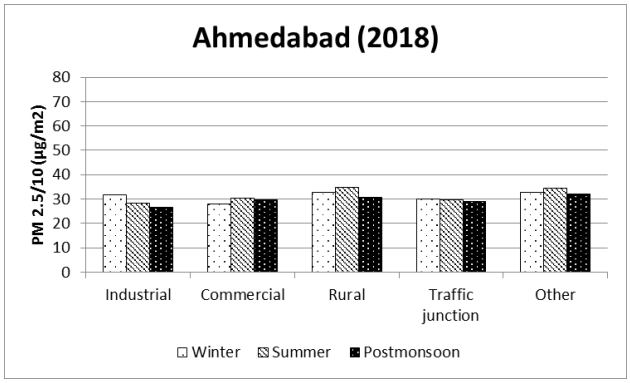

Figure 7c: Ratio of PM2.5/10

In 2017 and 2018 charts (Fig. 6a, 6b and Fig. 7b, 7c) , the average values of PM 2.5/10 are low for Ahmedabad as compared to Gandhinagar. We had not carried campaign during monsoon 2017 & 2018 as the particles get settled down due to washout effect & thus we can't get a clear picture of the pollution level over both the cities.

This indicates that Ahmedabad has higher deposition of coarser particles compared to Gandhinagar. This may be due to the resuspension of sand. Also Gandhinagar has higher green cover than Ahmedabad which reduces the resuspension of coarser particles.

7. ACKNOWLEDGEMENT

No endeavor is complete without acknowledging those who have helped me to make this project a success. I would like to express my gratitude towards Space Application Centre-ISRO for providing me necessary data from instruments to collect the data. I am also very thankful to Dr Rajesh Iyer (HOD, Dept. of Physics and Electronics, St. Xavier's College, Ahmedabad) for providing me this opportunity. I am thankful to Dr Abha Chhabra(SAC-ISRO) for her valuable guidance & support. My gratitude towards Tejas Turakhia help me in data analysis and guiding me.

REFERENCES:

[1] Andrew R. Deacon . Richard G. Derwent , Roy M. Harrison, Doug R. Middleton, Steve Moorcroft" Analysis and interpretation of measurements of suspended particulate matter at urban background sites in the United Kingdom" , The Science of the Total Environment, Vol: 203, (1997), 17-36.

[2] A. Vinod Kumar , R.S. Patil , K.S.V Nambi , "Source apportionment of suspended particulate matter at two traffic junctions in Mumbai, India" , Atmospheric Environment, Vol: 35, (2001), 4245-4251.

[3] Srimuruganandam Bathmanabhan , Shiva Nagendra Saragpur Madanayak, " Analysis and interpretation of particulate matter – PM10, PM2.5, and PM1 emission from the heterogeneous traffic near an urban roadway", Atmospheric Pollution Research, Vol: 1, (2010), 184-194.

[4] GRIMM Aerosol technik GmbH Co.KG

[5] www.EPA.gov.in

[6] safar.tropmet.res.in

[7] www.greenfacts.org

[8] particulatematter.in

STUDENT JOURNAL OF PHYSICS

Correlation of Structural properties with Thermodynamic Properties for Multi-Element AB_5-Type Hydrogen Storage Alloy

Niyanta Garkoti[1], Kuldeep Panwar[2] and Sumita Srivastava[2]

[1] M.Sc. III Semester, Department of Physics, Pt. L.M.S. Government Post Graduate College, Rishikesh (Autonomous College) Dehradun-249201, India

[2] Department of Physics, Pt. L.M.S. Government Post Graduate College, Rishikesh (Autonomous College) Dehradun-249201, India

niyanta.garkoti@gmail.com

Abstract: Wide applications of hydrogen storage alloys have resulted in synthesis of multi-element alloys. Modifying the parent alloy by substitution gives alloys whose properties can differ from the parent alloy. Important thermodynamic properties of hydrogen storage alloys are heat of formation of hydride and plateau pressure. These properties are measured through experiment. In the present study, structural properties of a class of hydrogen storage alloys have been correlated with these thermodynamic properties. Based on this correlation one can predict of heat of formation of hydride and plateau pressure for alloys without actually synthesizing them.

Keywords: *AB_5-Type Hydrogen Storage Alloy, Heat of Formation, Plateu Pressure*

1. INTRODUCTION

Hydrogen storage alloys are materials, which absorb hydrogen at certain pressure and desorb it at a lower pressure. Hydrogen atom is stored in the solid material at interstitial position. State of the art hydrogen storage alloys are AB_5, AB_2, A_2B and AB type. Among so many hydrides, AB_5 is popular for its easy activation and operation at room temperature and at little atmospheric pressure [1-3]. Nowadays, most of the AB_5-type alloys are synthesized through substitution of other elements in the parent alloy either at 'A' or at 'B'sites. The specific requirements of hydrogenation properties are met through synthesizing actual alloy with trial and error method. In earlier studies, theoretical approach for explaining thermodynamic properties have been presented [4-7]. Such studies were mostly based on ternary hydrides. At present no such model is available, which can predict the multi-element alloy properties without actually synthesizing it. In the present investigation, structural properties like lattice parameter 'a', 'c', unit cell volume and a new parameter r_B* have been correlated with the observed thermodynamic properties like heat of formation of hydride and hydrogen plateau pressure. Whereas 'a', 'c' and unit cell volume are properties of the synthesized alloy, r_B* is an equivalent radius of B in AB_5 alloy after substitution at the 'B' site. A relation has been established between structural and thermodynamic properties, which can be used to predict thermodynamic parameters of alloys.

2. METHODOLOGY

In present investigation, three series of alloys have been studied. These alloy series are termed as alloy1, alloy2 and alloy3. The nomenclature of alloys is given in Table 1. For each alloy series, experimentally observed values of lattice parameters 'a' and 'c' along with heat of formation and hydrogen plateau pressure reported in literature have been noted. r_B* has been calculated by taking stoichiometric composition of various elements at 'B' and atomic radius of each element. The values of atomic radii for elements under present study are given in Table 2. Graphs have been plotted among structural and thermodynamic parameters for each alloy and a relation has been established.

Table 1- Nomenclature of alloys

S.N.	**Name of alloy**	**Composition of alloy**
1	Alloy 1	$LaNi_4R$ (R= Al, Mn, Fe, Cu, Co, Cr)
2	Alloy 2	$LaNi_{5-x-y-z}Al_xSn_yFe_z$
3	Alloy 3	$La_{0.78}Ce_{0.22}Ni_{3.73}Co_{0.30}Al_{0.1}7Fe_{0.5-x}Si_x$ (x = 0, 0.05, 0.075, 0.1)

Table 2- Atomic radius of elements using VWR Sargent Welch Periodic Table

S.N.	Element	Atomic Radius **(Å)**
1	La	2.74
2	Ce	2.70
3	Ni	1.62
4	Co	1.67
5	Mn	1.79
6	Fe	1.72
7	Si	1.46
8	Al	1.82
9	Cu	1.57
10	Cr	1.85

The equation for calculating r_B* used in the present investigation is given as Eq. 1. The calculation for one of the members of alloy 1 is also shown as an example.

$$r_B^* = (4\, r_{Ni} + r_{Al})/5 \qquad (1)$$

For example we consider the alloy $LaNi_4Al$, obtained by starting $LaNi_5$ and five Ni atoms in substituting one of the 5 Ni atoms by an Al atom. For this alloy $r_B^* = \{(4 \diamond 1.62) + 1.82\}/5 = 1.66$.

3. RESULTS AND DISCUSSIONS

Table 3 shows all the known and calculated parameters of alloy 1-$LaNi_4R$ hydride.

Table 3- Known and calculated parameters of alloy 1 $LaNi_4R$ hydride [7-10]

S.N.	Alloy	a (Å)	c (Å)	Unit cell Volume ($Å^3$)	r_B^*(Å)	Heat of formation (kcal/mol H)	Plateau pressure (atm)
1	$LaNi_5$	5.017	3.986	86.28	1.62	-6.33	3.0
2	$LaNi_4Co$	5.018	3.981	86.2	1.63	-7.43	1.2
3	$LaNi_4Cu$	5.033	4.007	87.29	1.61	-8.068	1.6
4	$LaNi_4Fe$	5.049	4.015	88.02	1.64	-8.87	1.05
5	$LaNi_4Cr$	5.07	4.048	90.23	1.666	-10.0	0.91
6	$LaNi_4Mn$	5.089	4.082	90.91	1.654	-11.6	0.05
7	$LaNi_4Al$	5.061	4.07	89.65	1.66	-13.1	0.01

Table 3 shows that, in most of the cases heat of formation and plateau pressure decrease with increasing value of the lattice parameters, unit cell volume and r_B^*. More negative is the value of the heat of formation, more stable is the corresponding hydride. Plateau pressure is related to the heat of formation. More negative is the value of heat of formation, smaller is the plateau pressure. The variation of heat of formation and plateau pressure with unit cell volume and r_B^* are shown in Figures 1, 2, 3 and 4 respectively.

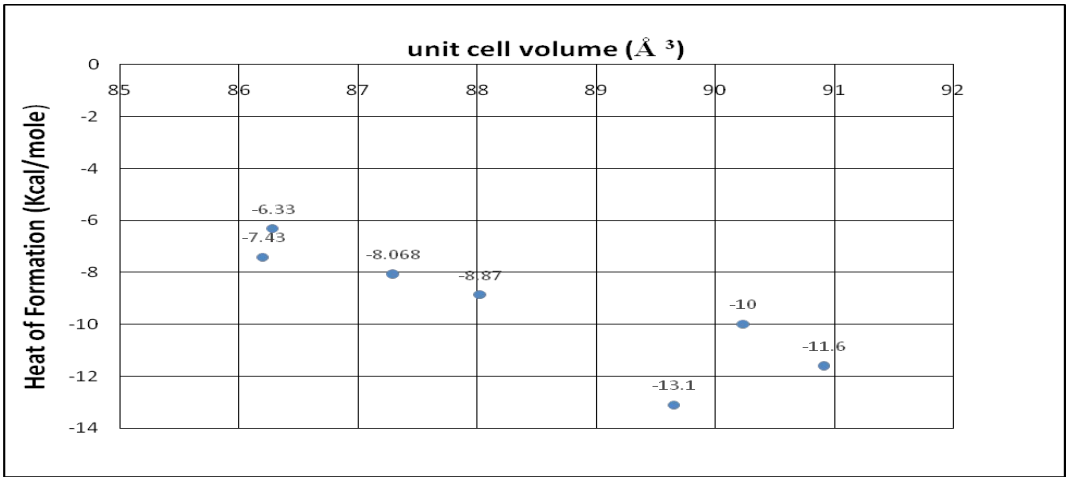

Figure 1- Variation of heat of formation with unit cell volume.

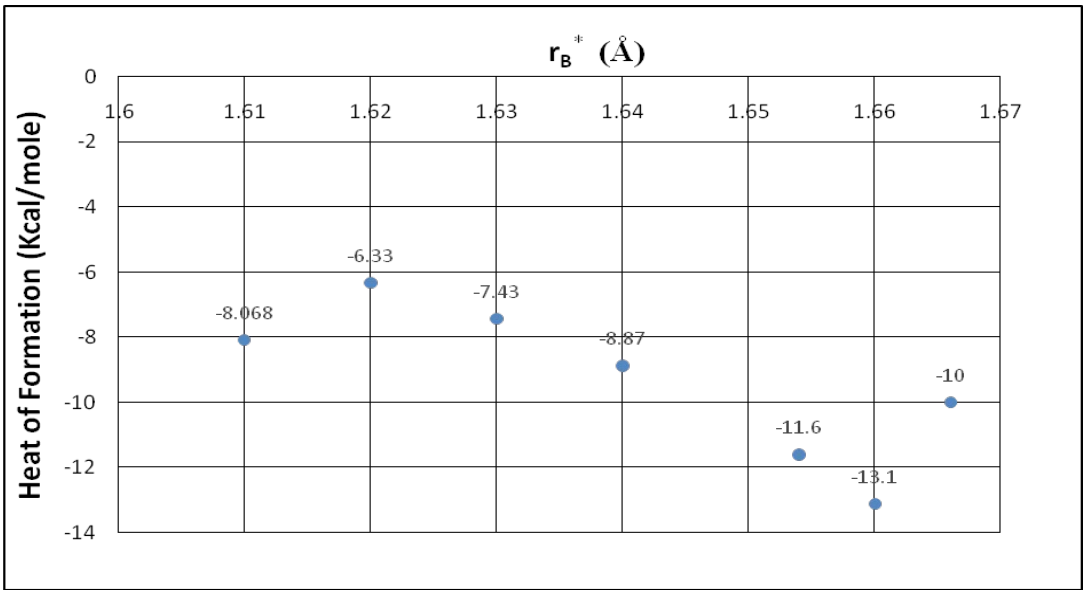

Figure 2- Variation of heat of formation with r_B^*.

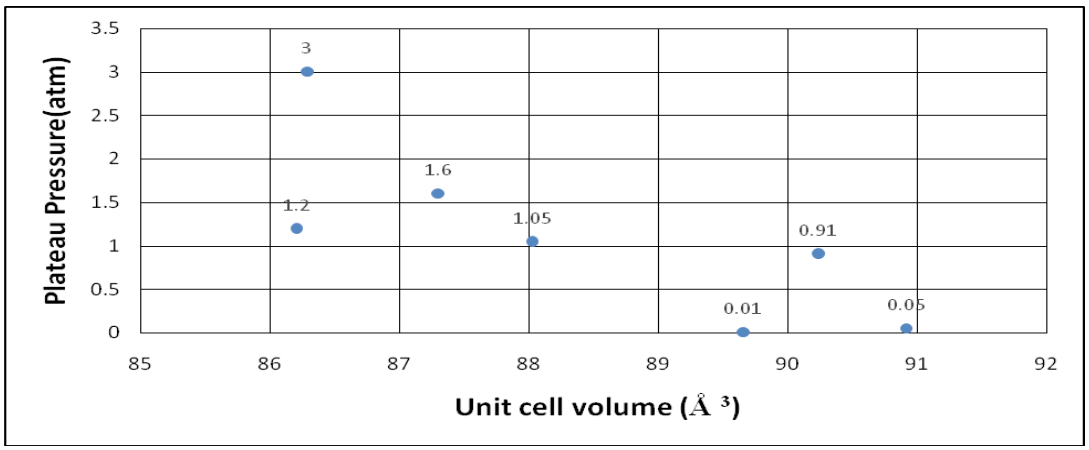

Figure 3- Variation of hydrogen plateau pressure with unit cell volume.

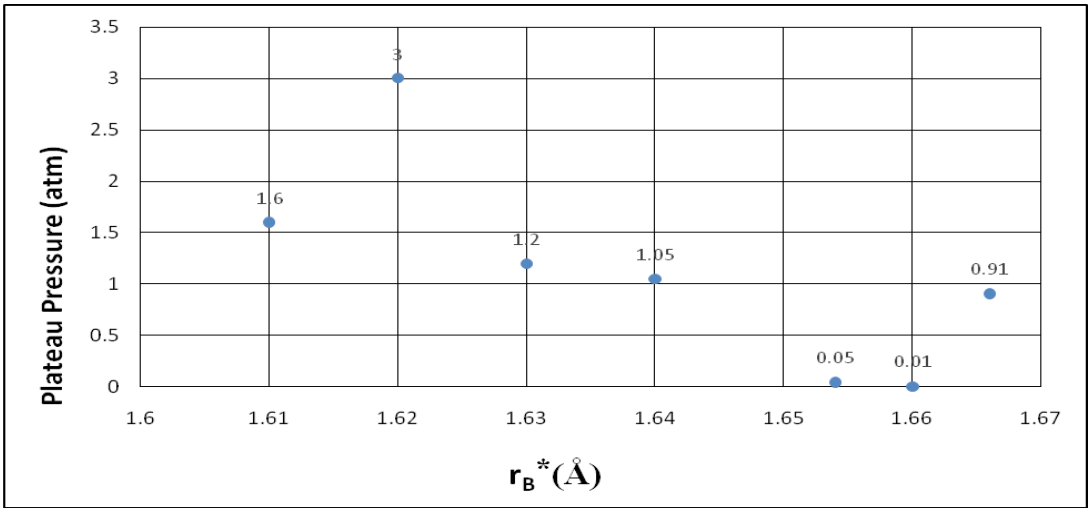

Figure 4- Variation of hydrogen plateau pressure with r_B^*.

The trend in heat of formation and plateau pressure are similar with unit cell volume and r_B^*. It may be mentioned here that unit cell volume is an experimental value obtained after the synthesis of the alloy and doing diffraction measurement, whereas r_B^* is a theoretically calculated value found without actually synthesizing the alloy. In this way, r_B^* can play a role similar to the unit cell volume. Hence by estimating the value of r_B^*, one can predict the trend of heat of formation and plateau pressure without synthesizing the actual alloy.

Similar known and calculated parameters of Alloy 2:$LaNi_{5-x-y-z}Al_xSn_yFe_z$ hydride are given in Table 4. This table also shows similar variation in heat of formation and hydrogen plateau pressure with unit cell volume and r_B^*. Both thermodynamic parameters decrease with structural parameters.

Table 4- known and calculated parameters of Alloy 2:$LaNi_{5-x-y-z}Al_xSn_yFe_z$ hydride [11]

S.N.	Alloy	a (Å)	c (Å)	Unit cell Volume ($Å^3$)	r_B^* (Å)	heat of formation kcal/mole H	Plateau Pressure (atm)
1	$LaNi_5$	5.1037	3.8936	84.1717	1.62	-6.33	0.500
2	$LaNi_{4.8}Sn_{0.1}Al_{0.1}$	5.0295	3.9987	86.9895	1.623	-8.56	0.071
3	$LaNi_{4.6}Fe_{0.2}Al_{0.2}$	5.0307	3.9997	87.0527	1.632	-8.8	0.049
4	$LaNi_{4.8}Sn_{0.2}$	5.0332	4.0107	87.3789	1.621	-8.18	0.063
5	$LaNi_{4.6}Sn_{0.2}Fe_{0.2}$	5.0403	4.0052	87.5055	1.628	-8.82	0.055

Table 5 shows same data for alloy3: $La_{0.78}Ce_{0.22}Ni_{3.73}Co_{0.30}Mn_{0.30}Al_{0.17}Fe_{0.5-x}Si_x$ multi-element hydride. In this case again, heat of formation and plateau pressure decrease with increasing value of unit cell volume. However, r_B^* does not show clear tendency of increase or decrease of heat of formation and plateau pressure. It may be due to substitution also at 'A' site.

Table 5- known and calculated parameters of Alloy 3: $La_{0.78}Ce_{0.22}Ni_{3.73}Co_{0.30}Mn_{0.30}Al_{0.17}Fe_{0.5-x}Si_x$ multi-element hydride [12]

S.N.	Alloy with x	a (Å)	c (Å)	unit cell volume ($Å^3$)	r_B^* (Å)	Heat of Formation (kcal/mole H)	Plateau pressure (atm)
1	0	5.03	4.045	88.0142	1.65	-6.584	0.5
2	0.05	5.03	4.046	88.0359	1.647	-7.514	0.45
3	0.075	5.031	4.05	88.158	1.646	-8.808	0.2
4	0.1	5.035	4.048	88.2547	1.639	-8.65	0.1

As pointed out earlier in this paper, heat of formation of hydride is a measure of the stability of the hydride. More negative is the value of the heat of formation, more stabile is the corresponding hydride. Due to stability of the hydride, hydrogen absorption may take place at lower hydrogen pressures. Further, it may be mentioned here that increased value of the unit cell volume facilitates the hydrogen absorption at a lower driving force, resulting in lower plateau pressure. These two factors are plausible explanation of the correlations observed between heat of formation of hydrides, stability of hydride, hydrogen plateau pressure and unit cell volume.

4. CONCLUSIONS

From above discussion, we conclude that the heat of formation and plateau pressure decreases with increasing value of unit cell volume in all the three class of alloys under study. Similar trends of heat of formation and plateau pressure are observed with r_B^* also. Since unit cell volume is an experimental value, structural parameter r_B^* can be used instead for predicting heat of formation and plateau pressure for an alloy without actually synthesizing it.

5. ACKNOWLEDGEMENT

The first author is grateful to her supervisor of M.Sc. Dissertation Dr. Sumita Srivastava, Professor and Head, Department of Physics, Pt. Lalit Mohan Sharma Government Post-Graduate College, Rishikesh.

REFERENCES

[1] Srivastava S, Srivastava ON. Investigations of synthesis and characterization of $MmNi_{4.3}Al_{0.3}Mn_{0.4}$ and $MmNi_{4.0}Al_{0.3}Mn_{0.4}Si_{0.3}$hydrogen storage materials through thermal and spin melting process. Int J Hydrogen Energy (1998); 23:7-13.

[2] Nagpal M, Kakkar R. An evolving energy solution: Intermediate hydrogen storage. Int J Hydrogen Energy (2018); 43:12168-88.

[3] Colbe JB et al. Application of hydrides in hydrogen storage and compression: Achievements, outlook and perspectives. Int J Hydrogen Energy (2019); 44:7780-808.

[4] Magee CB, Liu J, Lundin CE. Relationships between intermetallic compound structure and hydride formation. J less common metals (1981);78:119-38.

[5] Yoshikawa A, Matsumoto T, Yagisawa K. Relationships between crystal structure and Plateau length of pressure-composition isotherms of AB_5 hydrides. J less common met (1982);88:73-79.

[6] Van Mal HH, Buschow KHJ, Miedema AR. Hydrogen absorption in $LaNi_5$ and related compounds: experimental observations and their explanation. J less common met (1974); 35:65-76.

[7] Lundin CE, Lynch FE, Magee CB. A correlation between the interstitial hole sizes in intermetallic compounds and the thermodynamic properties of the hydrides formed from those compounds. J less common met (1977);56:19-37.

[8] Panwar K, Srivastava S. Investigations on calculation of heat of formation for multi-element AB_5-type hydrogen storage alloy. Int J Hydrogen Energy (2018);43: 11079-84.

[9] Lartigue C, PercheronGuegan A, Achard JC, Tasset F. Thermodynamic and structural properties of $LaNi_{5-x}Mn_x$ compounds and their related hydrides. J less common met (1980); 75:23-29.

[10] Diaz H, Percheron-Guegan A, Achard JC, Chatillon C, Mathieu JC. Thermodynamic and structural properties of $LaNi_{5-y}Al_y$ compounds and their related hydrides. Int J Hydrogen Energy (1979);4:445-54.

[11] Kazakov AN, Dunikov DO, Mitrokhin SV. AB_5-type intermetallic compounds for biohydrogen purification and storage. Int J Hydrogen Energy (2016);41:21774-79.

[12] Zhou W, Wang Q, Zhu D, Wu C, Huang L, Ma Z, Tang Z, Chen Y. The high-temperature performance of low-cost La-Ni-Fe based hydrogen storage alloys with Si substituting. Int J Hydrogen Energy (2016);41:14852-63.

Erratum

The effect of right-handed currents and dark side of the solar neutrino parameter space to Neutrinoless Double Beta Decay

[Student Journal of Physics,Vol. 7, No. 4, Oct-Dec. 2018]
P.K. Bishee, P. Sahu, and S. Patra

In the original paper, an important reference was missed out. Numbering that as ref.47, the new reference is

(47) K. N. Vishnudath, S. Choubey, and S. Goswami, "New sensitivity goal for neutrinoless double beta decay experiments," Phys. Rev. D99 (2019) no. 9, 095038, arXiv:1901.04313.

Including this reference, the paragraph at the end of the Introduction should be replaced by the following:

"In this paper, we study the effect of DLMA solution to the solar neutrino problem on neutrinoless double beta ($0\nu\beta\beta$) decay for both of these standard (reproduced the results presented in ref [47]) and extend our new analysis to right-handed current mechanisms and compare them with the standard LMA solution to the solar neutrino problem on $0\nu\beta\beta$ for both mechanisms. This knowledge helps the future experiment to probe in the different energy range of effective mass and find out the sensitivity on $0\nu\beta\beta$."

Similarly, the last line of the first paragraph in Section II should be replaced by the following line:

"So for standard mechanism, the inverse half- life ($T_{1/2}$) for $0\nu\beta\beta$ is given as (for earlier references, see [47,17,15,32,38])."

STUDENT JOURNAL OF PHYSICS

Volume 8 **Number 1** **2019**

CONTENTS

ARTICLES

Exploring Symbolic Neural Networks for Multiscale Applications 1
Elijah Sheridan, David Rimel and Michael S. Murillo

Optimizing an Electrospun Catalyst in Polyelectrolyte Membrane Fuel Cells 18
Surya Rajan

Comparative analysis of Particulate Matter 24
Hiral Ponda

Correlation of Structural properties with Thermodynamic Properties for Multi-Element AB_5-Type Hydrogen Storage Alloy 33
Niyanta Garkoti, Kuldeep Panwar and Sumita Srivastava

Erratum 40