

PARALLEL PARTIAL EMULATION IN APPLICATIONS

*Yingjie Gao & E. Bruce Pitman**

¹*Materials Design and Innovation, University at Buffalo Buffalo, NY 14260*

*Address all correspondence to: E. Bruce Pitman, E-mail: pitman@buffalo.edu

Original Manuscript Submitted: ; Final Draft Received:

Inference, optimization, and inverse problems are but three examples of mathematical operations that require the repeated solution of a complex system of mathematical equations. To this end, surrogates are often used to approximate the output of these large computer simulations, providing fast and cheap approximation solutions. Statistical emulators are surrogates that, in addition to predicting the mean behavior of the system, provide an estimate of the error in that prediction. Classical Gaussian Stochastic Process emulators predict scalar outputs based on a modest number of input parameters. Making predictions across a space-time field of input variables is not feasible using classical Gaussian process methods. Parallel Partial Emulation is a new statistical emulator methodology that predicts a field of outputs, based on the input parameters. Parallel partial emulation is constructed as a Gaussian process in parameter space, but no correlation among space or time points is assumed. Thus the computational work of parallel partial emulation scales as the cube of the number of input parameters (as traditional Gaussian Process emulation) and linearly with space-time grid.

The numerical methods used in numerical simulations are often designed to exploit properties of the equations to be solved. For example, modern solvers for hyperbolic conservation laws satisfy conservation at each timestep, insuring overall conservation of the physical variables. Similarly, symplectic methods are used to solve Hamiltonian problems in physics. It is of interest, then, to study whether Parallel Partial Emulation predictions inherit properties possessed by the simulation outputs. Does an emulated solution of a conservation law preserve the conserved quantities? Does an emulator of a Hamiltonian system preserve the energy? This paper investigates the properties of emulator predictions, in the context of systems of partial differential equation. We study conservation properties for three different kinds of equations – conservation laws, reaction-diffusion systems, and a Hamiltonian system. We also investigate the effective convergence, in parameter space, of the predicted solution of a highly nonlinear system modeling shape memory alloys.

KEY WORDS: *surrogate models, Gaussian Process emulation, high-dimensional outputs, conservation, Parallel Partial Emulation*

1. INTRODUCTION TO SURROGATE MODELS

Surrogate models are often used to quickly approximate the results of time-consuming computational or physical experiments. In this way, a scientist can explore the range of parameters and settings that serve as experimental inputs, experiments that could not be performed because of time and/or budget constraints. For example, Bayesian inference for a system of partial differential equations may require tens or hundreds of thousands of evaluations of a large computer simulation as part of a Markov Chain Monte Carlo study. Design, optimization, and inverse problems likewise may require many solutions of a computational model. Surrogate models allow detailed, if approximate, scientific exploration of inputs and parameters, on a timescale not attainable through direct simulation or experimentation.

There are several kinds of surrogates employed in different settings. Reduced order models (ROMs) replace a complex mathematical model with a set of easier-to-solve equations that approximate outputs. For example, a ROM might replace a large system of partial differential equations with a small set of ordinary differential equations that can be solved rapidly. An experiment that includes stochastic elements may be modeled by a truncated version of a Karhunen-Loeve equation, again expecting that this equation can be solved (relatively) rapidly [1]. Physics-informed neural networks (PINNs) are function approximators that can embed physical laws defined by, for example, a system of differential equations with sufficiently smooth solutions [2].

Statistical emulators are another kind of surrogate model, often used as an approximation to the output of complex computer models. In addition to predicting the output of a simulation, statistical emulators provide an

estimate of the possible error in that prediction [3]. One common statistical emulator is the Gaussian Stochastic Process (GaSP) [4]. The classical theory of GaSP emulators predicts a scalar output measure of the solution of a system of equations, often ordinary or partial differential equations. In order to predict a field of outputs, such as the entire space-time solution of a system of differential equations, this classical approach requires that every gridpoint in space-time to be treated as an input parameter. The output is the emulated solution for every space-time point. The work in computing a GaSP emulator scales as the cube of the number of input parameters, rendering this approach to emulating a field impractical.

Recently Gu and Berger developed Parallel Partial Emulation (PPE) [5], which emulated the output of a simulation of differential equations across an entire region of space and/or time, with computational work that scales linearly with the number of space and time gridpoints.

Numerical methods used in simulations are often designed to exploit properties of the equations to be solved. For example, modern methods for hyperbolic conservation laws are designed to preserve the L_1 norm of the solution. A different kind of conservation is present in Hamiltonian systems, and symplectic methods are often used as numerical solvers in order to preserve the total energy of the system. In a similar fashion, some reduced order methods are explicitly designed to enforce properties such as conservation, as in [6]. Standard PINNs require sufficient smoothness of solutions of the underlying PDE to guarantee an accurate approximation, and PINNs may fail to approximate solutions of nonlinear hyperbolic equations; current research is exploring additional constraints that must be included within the PINN framework to guarantee conservation of norms, or of other physically important measures for conservation laws such as entropy [7-9].

Thus, it is natural to ask whether PPE predictions, which can be thought of as very fast, approximate numerical solvers, conserve norms. Does an emulated solution of a conservation law, trained using a conservative numerical method, preserve the total mass of the system? If a symplectic method is used to train a PPE for a Hamiltonian system, does the prediction preserve the total energy? To be clear, PPE does not explicitly enforce conservation nor any other property of the differential equations being studied. PPE does assume some smoothness in the covariance structure in parameter space, but does not make any assumption about the space-time structure of solutions. Thus, our conclusion that PPE preserves conservation properties of the simulator should be seen a property of the Parallel Partial Gaussian process emulator itself, and not as a consequence of enforcing norms to be preserved.

This paper studies conservation properties of the emulated solution for three different kinds of equations – conservation laws, a reaction-diffusion system, and a 2-body Hamiltonian system. We also investigate the effective convergence – in parameter space – of the predicted solution of a highly nonlinear system; this convergence study provides a quick estimate for the error in the solution or functionals of the solution that can be expected in a complex system. In the following section, we review the classical construction of a scalar Gaussian process emulator, and then explain the extension to PPE. In subsequent sections, we apply PPE to four different model systems of differential equations.

2. GASP

We begin by briefly reviewing the construction of a GaSP emulator.

2.1 Basic Gaussian Process Theory

The GaSP is designed to be a BLUP - a Best, Linear, Unbiased Predictor, based on available data. Let us explain. We assume data consisting of M d -dimensional input parameters q , and scalar outputs $Y = (Y_1, \dots, Y_M)^T$. At an untested input q^* , the predicted output is $\hat{Y}(q^*)$. As a BLUP, then, the GaSP satisfies the following:

- The GaSP predictor is the best predictor in the L_2 sense. That is, for the true output Y , the integral $\int (\hat{Y}(q) - Y(q))^2 dq$ is minimized among all predictors with finite mean and variance
- The GaSP predictor is linear in the data, in that $\hat{Y} = a_0 + \sum_{j=1}^M a_j Y_j$
- The GaSP predictor is unbiased, meaning $E[\hat{Y}(q)] = E[Y(q)]$.

The construction of a GaSP begins by assuming a Gaussian Process prior on the space of functions with finite mean and second moment. A Gaussian Process is a function $f(q)$ such that for any finite collection $q_1, q_2, \dots, q_M$ of inputs, the outputs $\{f(q_1), f(q_2), \dots, f(q_M)\} \sim MVN(\mu(q); \hat{R})$, a multivariate normal with covariance matrix $\hat{R}$ and mean μ . A common assumption is that $\hat{R}$ is separable in the d -dimensions of q , and can be written as $\hat{R} = \sigma^2 R$ for a correlation matrix R , defined by a correlation kernel, and a variance parameter σ^2 . Common correlation kernels include the power exponential $\exp(-(|q_i - q_j|)^p / (2\gamma))$ and the Matern function $\frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \frac{d}{\rho})^\nu K_\nu(\sqrt{2\nu} \frac{d}{\rho})$ with $d = |q_i - q_j|$. Hyper-parameters, and in particular the correlation lengths (γ and ρ) must be estimated from the data.

Given input and output data $\{q_j, Y_j, j = 1, 2, \dots, M\}$, it is common to separate the overall mean trend from the rest of the process. Thus we write $Y = h(q)\psi + Z$, where h is a collection of basis functions (often a low order polynomial), ψ are coefficients for these basis functions, and Z a zero-mean Gaussian Process. In the following we continue to refer to the Gaussian Process as Y , whether or not it is zero mean.

Given a Gaussian Process prior and data $\{q, Y\}$, the posterior distribution at some untested q^* can be determined explicitly by conditioning. In particular, we can write formulae for the mean and the variance (see Sec. 2.2). The largest computational burden in constructing a GaSP is the calculation of R^{-1} , usually performed via a Cholesky decomposition. The interested reader may consult [410] for more details.

Maximum likelihood estimation (MLE) is often used to determine the hyper-parameters of the Gaussian process. MLE estimates tend to underestimate the variance term in practice [11]. A fully Bayesian approach would put priors on all hyper-parameters and solve. But these hyper-parameter priors have their own parameters, and the hierarchy continues to expand, and can quickly become prohibitively expensive to compute. A hybrid approach is to estimate the most crucial uncertainties (for example, correlation lengths) via a Bayesian analysis, and retains MLEs for other parameters. With appropriate choices of priors, this hybrid approach again results in somewhat complicated but still explicit formulae for the mean and variance; again we refer the reader to [410].

2.2 Parallel Partial Emulation

In a conventional approach, if one wished to emulate the output at many space-time locations $(\mathbf{x}, t)$, each space-time point would be treated as an input dimension, and the standard GaSP construction outlined above would be performed. Emulating over an entire space-time domain with thousands of points is, therefore, prohibitively expensive [11–13]. Nevertheless in applications there often is a need for emulation at every point of a space-time field. For example, one may be interested in performing a sensitivity analysis on the parameter inputs q of a time-dependent partial differential equation $\frac{\partial}{\partial t} \mathbf{u} = \mathcal{F}(\mathbf{u}, \nabla \mathbf{u}; q)$. Here $\mathbf{u}$ is a dependent variable depending on space points $\mathbf{x}$ and time t . The quantities of interest in such an analysis would likely include norms such as $\int \mathbf{u}^\alpha(\mathbf{x}, t; q) d\mathbf{x}$. Sensitivity analysis requires a solution of this differential equation for many parameter values, across space. Moreover the quantity of interest may change over time, so it may be important to construct an emulator for all $(\mathbf{x}, t)$ in addition to the q s.

Parallel Partial Emulation (PPE) generalizes the standard emulator construction, emulating an entire field of outputs at once [5]. Each space-time output location has its own mean μ and variance σ^2 depending on q and $(\mathbf{x}, t)$, but the correlation structure and correlation lengths depend on q only, and are shared among all space-time locations. In this way the computational burden of calculating an enormous R^{-1} is avoided. To be clear, PPE makes no assumption about the correlation among space-time points. In addition, we assume the set of basis functions h and the coefficients ψ for the overall trend are common to all points, although these are evaluated at every space-time point.

Specifically, using a hybrid approach as discussed earlier, and given M model design inputs and responses (q, Y) , the PPE predictive process mean at an untested input, q^* is given as

$$\hat{Y}(q^*) = h(q^*)\psi + r^T(q^*)R^{-1}(Y^D - h(q^D)\psi), \quad (1)$$

where

$$\psi = (h^T(q^D)R^{-1}h(q^D))^{-1}h^T(q^D)R^{-1}Y^D$$

is the generalized least squares fit to the data, with $\cdot^T$ denoting the transpose. Also, $r(q^*)$ is the correlation vector between design points q_j and the untested q^* ; that is $r(q^*) = R(q_j, q^*)$. To emphasize the construction, we employ the superscript $\cdot^D$ to indicate the input data.

We note that these formulae are the same as those for a classical (i.e. scalar) Gaussian Process construction, although the interpretation is different. Here the data Y^D and the prediction $\hat{Y}$ are space-time fields, and h, ψ are evaluated at every space-time point.

The predictive variance at each space-time point is given by

$$\begin{aligned} \text{var}(q^*) &= \sigma^2(1 - r^T(q^*)R^{-1}r(q^*)) \\ &\quad + (h(q^*) - h^{D^T}R^{-1}r(q^*))^T (h^{D^T}R^{-1}h^D)^{-1} (h(q^*) - h^{D^T}R^{-1}r(q^*)). \end{aligned} \quad (3)$$

The paper [5] applies PPE to a large computation of a geophysical mass flow, as described in [14], across a 2-dimensional spatial region (time is not an active variable in this application). However, no assumptions are made about the correlation structure of the GaSP at nearby spatial or space-time points, and the behavior of Parallel Partial Emulation predictions as a function of space and/or time is not well understood. Indeed, the construction of the Parallel Partial Emulator and the mass flow application in [5] suggest that the relationship among spatial points in the emulation should be ‘inherited’ from the spatial smoothness of a solution to the underlying partial differential equations. This is the critical issue we explore.

In particular, using example problems we examine how the predicted solution inherits conservation properties of the numerical solution methodology – that is, whether the predicted solver conserves some functional of the solution. These examples include (i) a scalar conservation law (Sec. 3), (ii) a reaction-diffusion model in which the total mass of the system is conserved although the mass of each individual species is not (Sec. 4), and (iii) the Hamiltonian system describing Kepler’s 2-body problem, and which conserves the total energy of the system (Sec. 5). We also examine a system of differential equations modeling the deformation of shape memory alloys (Sec. 6), in order to study accuracy under refinement in parameter space; we provide a summary of error estimates for PPE predictions, and this error estimate gives guidance about how accurate any solution or functional of the solution is likely to be.

In our analysis and applications we use the default settings for the *ppgasp* routine in the R package *RobustGaSP*, with a power exponential correlation function with exponent 1.9 [15].

3. PPE AND CONSERVATION LAWS

Consider the simplest nonlinear scalar conservation law, the viscous Burgers’ equation in 1 space dimension,

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \epsilon \frac{\partial^2 u}{\partial x^2} \quad (4)$$

with $f(u) = u^2/2$. The Hopf-Cole transformation provides a mechanism for calculating the exact solution of Burgers’ equation. The inviscid equation, with $\epsilon = 0$, is the starting point for understanding hyperbolic conservation laws.

Integrating this equation on $x \in [0, 1]$ with periodic boundary conditions we find

$$\begin{aligned} \int_0^1 \left[\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} \right] dx &= \int_0^1 \left[\epsilon \frac{\partial^2 u}{\partial x^2} \right] dx \\ \frac{\partial}{\partial t} \int_0^1 u(x, t) dx + f(u(1, t)) - f(u(0, t)) &= \epsilon \left[\frac{\partial u(1, t)}{\partial x} - \frac{\partial u(0, t)}{\partial x} \right] \end{aligned}$$

which yields

$$\frac{\partial}{\partial t} \int_0^1 u(x, t) dx = 0.$$

Let us write $u_j^n = u(x = j\Delta x, n\Delta t)$ as a numerical approximation of the true solution at the space point $j\Delta x$ and time point $n\Delta t$. Then a conservative numerical method can be written as

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} [F(u)_{j+\frac{1}{2}} - F(u)_{j-\frac{1}{2}}] \quad (5)$$

for some numerical flux function F . Conservation can be seen by summing Eq. 5 over all grid cells and noticing the cancellation of all fluxes except at the domain boundary. Modern methods for systems of hyperbolic conservation laws satisfy this numerical conservation property [16].

If we compute several solutions to Eq. 4 for several specified values of ϵ using a conservative numerical method, will a solution predicted by PPE for an untested value of ϵ satisfy conservation? In addressing this question it is important to recall that the construction of a standard Gaussian process emulator encodes no explicit condition enforcing properties (such as conservation) that might be of interest to disciplinary scientists.

Consider a 1-dimensional grid of points on the interval $[0, 1]$ with spacing Δx . To solve Burgers' equation we specify initial data at $t = 0$ (a square wave will be used below) and boundary conditions (periodic conditions will be used). The grid variable u_j^n is thought of as $u_j^n = \int_{(j-\frac{1}{2})\Delta x}^{(j+\frac{1}{2})\Delta x} u(x, n\Delta t) dx$, the total "mass" of $u(x, t)$ in the j^{th} grid cell at time $n\Delta t$. Because of conservation,

$$\int_0^1 u(x, n\Delta t) dx = \int_0^1 u(x, (n+1)\Delta t) dx$$

for every time step.

To study conservation, we marry a parabolic predictor-corrector scheme with Davis' approach to computing second-order accurate solutions to hyperbolic conservation laws [17]:

$$u_j^{n+\frac{1}{2}} = u_j^n - \frac{\Delta t}{2\Delta x} u_j^n \Delta_j u + \epsilon \frac{\Delta t}{\Delta x^2} (u_{j+1}^{n+\frac{1}{2}} - 2u_j^{n+\frac{1}{2}} + u_{j-1}^{n+\frac{1}{2}})$$

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} (F(u_{j+\frac{1}{2}}) - F(u_{j-\frac{1}{2}})) + \frac{\epsilon}{2} \frac{\Delta t}{\Delta x^2} [(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1})] + [(u_{j+1}^n - 2u_j^n + u_{j-1}^n)]$$

Here the numerical flux $F_{j+\frac{1}{2}}$ is composed of two parts: (i) $F(u_{j+\frac{1}{2}})$ which is readily determined as the stationary state for the inviscid Burgers equation with Riemann data $u_j^{n+\frac{1}{2}} + \frac{1}{2}\Delta_j u$ and $u_{j+1}^{n+\frac{1}{2}} - \frac{1}{2}\Delta_{j+1} u$ and limited slopes $\Delta_j u$, and (ii) a centered difference of the Laplacian term, $\epsilon(u_{j+1}^n - u_j^n)$. This computational method has a truncation error $O(\delta t^2 + \Delta x^2)$.

Consider square wave initial data

$$u(x, 0) = \begin{cases} 1 & 0.4 < x < 0.6 \\ 0 & \text{otherwise} \end{cases}$$

We remark that, for the inviscid Burgers' equation, any initial data $u^0(x)$, however smooth, which is decreasing somewhere on its domain, will form a shock in finite time. So discontinuous solutions are typical when $\epsilon = 0$. Of course viscous dissipation will smooth this shock. Tests on smooth initial data produce similar conservation results to those reported below.

For any $\epsilon \geq 0$, the numerical method conserves the mass of the solution to machine precision, at least up to a nominal time $T = 1$. Fig. 1 shows how the computed solutions changes as a function of time for fixed ϵ , and as the viscosity varies, for fixed time.

To investigate conservation we vary both the dissipation parameter ϵ and the time $t_n = n\Delta t$ at which the solutions are sampled. Specifically, consider the following: Given computed solutions u^{ϵ_k} for a set of $(\epsilon_1, t_1), (\epsilon_2, t_2), \dots, (\epsilon_K, t_K)$ pairs of parameters and times, for $0 \leq x < 1$, does the PPE solution $\hat{u}^{\epsilon_*}$ satisfy $\int_0^1 \hat{u}^{\epsilon_*}(x, t) dx = \int_0^1 u^{\epsilon_*}(x, 0) dx$? We perform this experiment on spatial grids with 100, 200, and 400 cells.

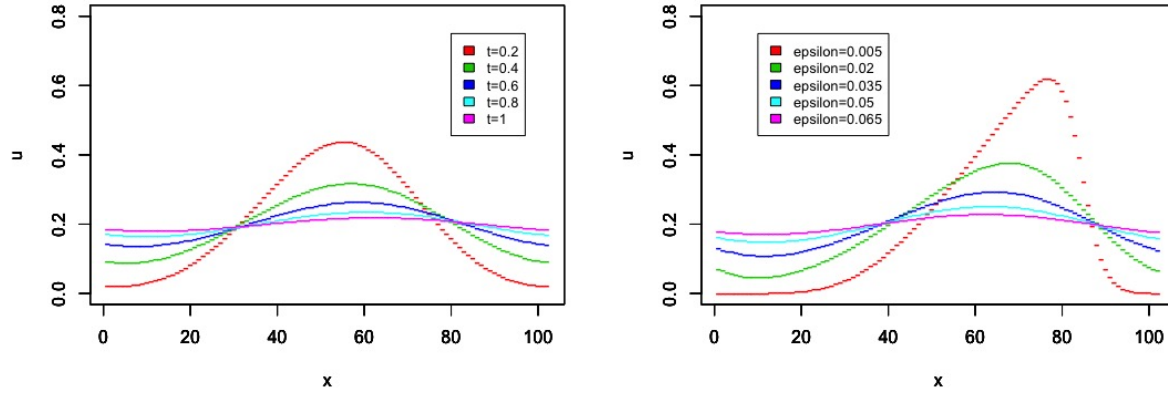


FIG. 1: The evolution of the solution to Burgers' Equation. Left: given fixed $\epsilon = 0.075$, the computed solution $u(x, t)$ flattens out as time increases. Right: at fixed time $t = 1$, the computed solution $u(x, t = 1)$ curves become smoother as ϵ increases.

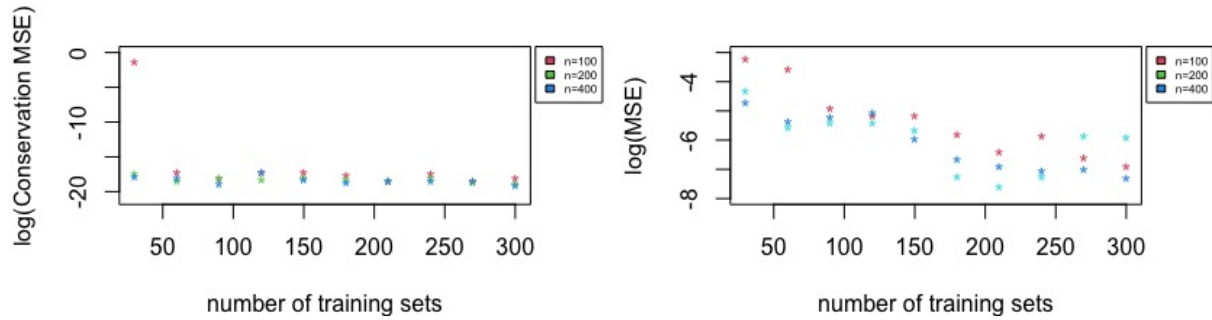


FIG. 2: Left: Conservation errors of PPE predicted solution, on three different grids, tested using 20 randomly selected (ϵ, t) data, and an increasing number of training datasets from 30 to 300. Right: Averaged MSE of PPE predicted solutions relative to computed solutions, on different grids and for varying numbers of training datasets. In both plots, n specifies the number of computational grid cells used in the construction of the emulator. Three grids, with 100, 200, and 400 cells, have been used for each emulation, denoted by red, green, and blue markers, respectively; some points in the plots overlap.

Define the conservation error as the mean squared error between the mean of the predicted PPE solutions with specific ϵ at some time t , and the initial area $\int_0^1 u(x, 0) dx$ (which is 0.2 with the specified square wave data). We consider 15 values of $\epsilon = 0.005, 0.01, \dots, 0.075$, and times $\Delta t, 2\Delta t, \dots, 501\Delta t = 1$. We select 20 random (ϵ, t) pairs as test data, and train on 30, 60, ..., 300 of the remaining pairs, again selected at random. The left plot of Fig. 2 shows the log of the conservation error of PPE predicted solutions, for the selected ϵ, t , on grids of 100, 200, and 400 cells. Recall that the computed solutions conserve mass to approximately machine precision. The right plot shows the MSE of the PPE prediction, relative to the computed solution, averaged over the 20 test predictions

$$\frac{1}{20} \sum \int_0^1 (\hat{u}^\epsilon(x, t) - u^\epsilon(x, t))^2 dx$$

computed on different grids, for 30, 60, ..., 300 training datasets.

We see that the PPE prediction conserves mass to about the same accuracy as the computed solution. In the MSE error plot, the worst error – 100 grid points, and 30 training datasets – yields a mean squared error of less than 10^{-2} , which is about the accuracy of the total computed error on that grid. It is informative to take note of the

variability owing to the size of the training data, and the averaged error on different grids.

4. A REACTION-DIFFUSION SYSTEM

For Burgers' Equation, conservation is expressed by the equation itself. Provided the numerical scheme respects conservation at the grid level, 'mass' should be conserved to high accuracy. In this section we study a system of reaction-diffusion equations (RDE), none of which individually embodies a conservation principle but which, together, conserve the total mass of the system. We again probe whether PPE predictions preserve this property.

We examine the PPE emulation of a RDE system on the domain $[-1, 1] \times [-1, 1]$, a simplification of part of a model of tumor growth (a more complete model may be found in [18], and references in that paper). Two cell species ϕ_1, ϕ_2 , diffuse and feed each other, reacting with a nutrient species v :

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} &= \nabla(\mu_1 \nabla \phi_1) + \beta \phi_2 v \\ \frac{\partial \phi_2}{\partial t} &= \nabla(\mu_2 \nabla \phi_2) + \alpha \phi_1 v \\ \frac{\partial v}{\partial t} &= -(\phi_1 + \phi_2)v\end{aligned}\tag{6}$$

The reaction terms show v being consumed at a rate proportional to the amount of ϕ_1, ϕ_2 , and the ϕ species increase synergistically. Although the diffusion tensor might vary in space or as a function of concentrations, here we prescribe a constant μ for each species. Initial and boundary conditions are also required. In the computations here we restrict ourselves to two spatial dimensions, with periodic boundary conditions on all sides. Initial data consists of constant v , and piecewise constant ϕ s, all of which are symmetric across the origin.

Note that, from the differential equations, the total amount of material $C = \int (\alpha * \phi_1 + \beta * \phi_2 + \alpha * \beta * v) dx dy$ is constant in time for periodic or no-flux boundary conditions. It is this conservation of total material that we will study. We remark that, because of symmetry considerations, we can limit ourselves to calculate a one-dimensional integral of C taken along a line through the origin.

In contrast to a study of the PPE prediction of a conservation law, the conservation property of this RDE system is not represented by a specific equation that is solved, nor a conservative numerical method, but is a consequence of the solution to all three equations. In many applications ranging from multi-phase flow to combustion, accurate tracking of material is critical to the successful simulation of solutions; this examination of RDEs is a first step in extending PPE ideas to these settings.

In this application the parameter space is 4-dimensional – $\alpha, \beta, \mu_1, \mu_2$. We work with a fixed spatial grid of 100 grid cells in x and y , with $x, y \in [-1, 1]$. The computational algorithm is an explicit second-order mid-time predictor-corrector scheme. Certainly this is not a methodology that would be used for a production code. But our intent is to study the parametric dependencies of PPE predictors. This explicit numerical scheme requires no linear algebra solver. On the other hand our study does not investigate any stiffness of the reaction terms, a phenomenon that could be important in some settings. Thus we restrict all four parameters to be positive and $O(1)$. The evidence suggests that PPE predictions trained using specialty stiff solvers for the reaction terms will exhibit conservation similar to what is reported here.

We begin, then, with initial data $v = 1$ for all x, y , and

$$\phi_j(x, 0) = \begin{cases} 1 & -0.2 < x < 0.2, \quad -0.2 < y < 0.2 \\ 0.1 & \text{otherwise} \end{cases}\tag{7}$$

The diffusion term smooths the discontinuity, but the relatively fast reaction near the origin quickly depletes v . We remark that testing with other initial data, whether smooth or discontinuous, also exhibits conservation of C as illustrated below.

As an example of the complicated behavior of the system, Fig 3 shows the weighted sum of material $\alpha * \phi_1 + \beta * \phi_2 + \alpha * \beta * v$ at four different times. In this figure all the parameters are set equal to 0.5.

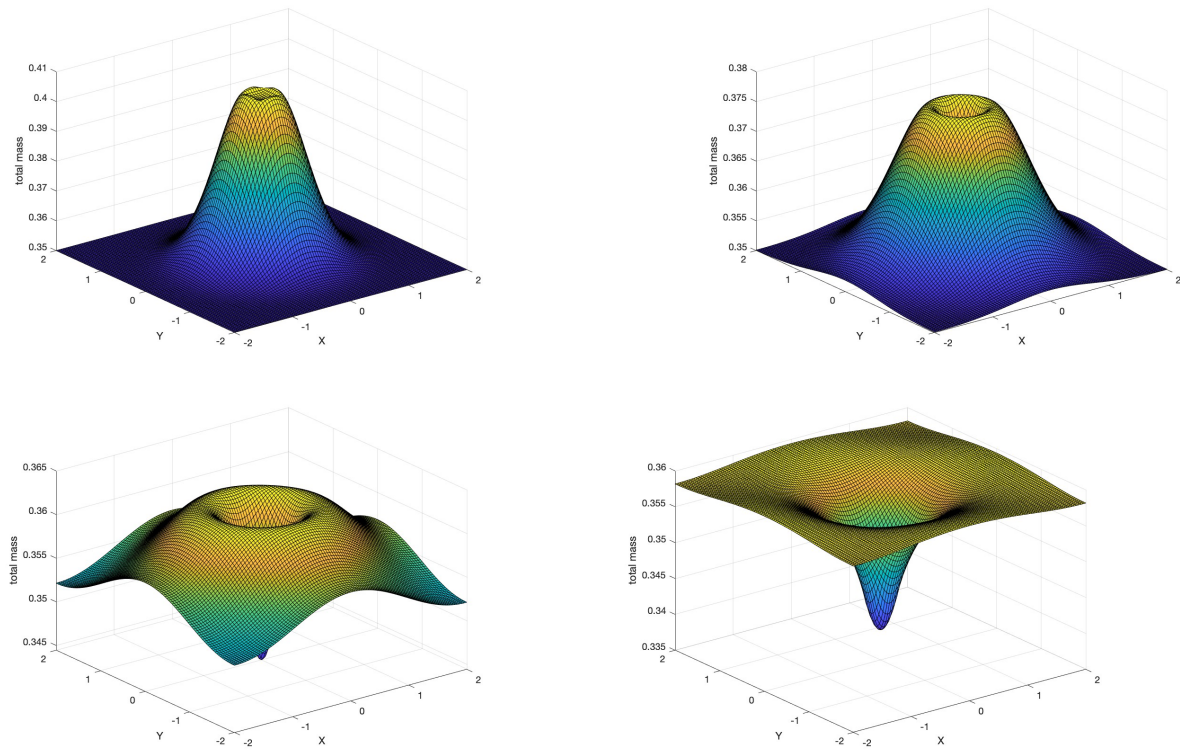


FIG. 3: The total weighted mass of material at non-dimensional times $t = 0.25$ (upper left), $t=0.5$ (upper right), $t=1.0$ (lower left), and $t=2.0$ (lower right). Parameters $\mu_1, \mu_2, \alpha, \beta$ are all equal to 0.5

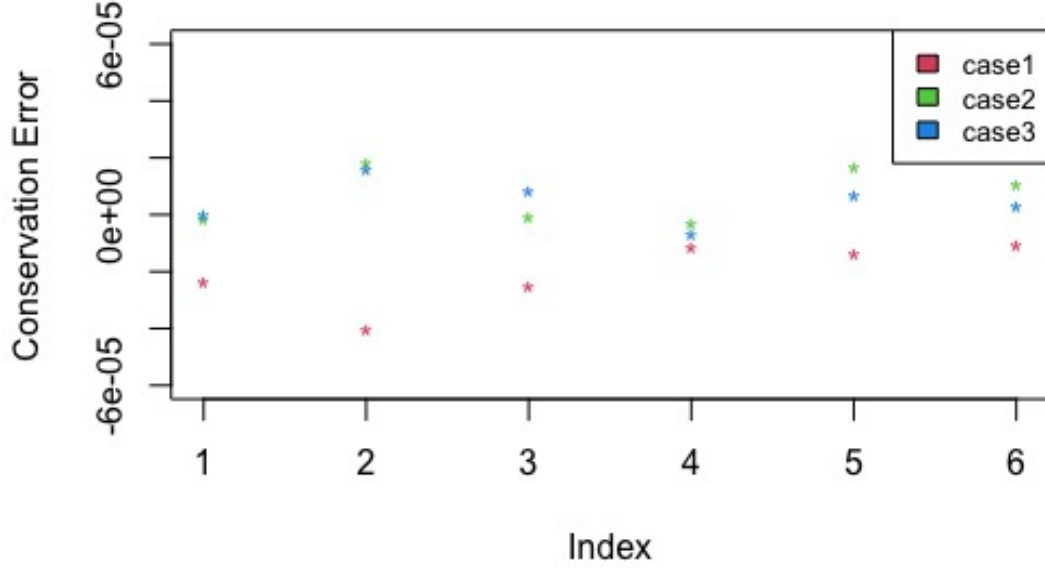


FIG. 4: Conservation error, for three separate selections of training and testing datasets represented by red, green and blue markers. Shown are each of the errors for each of the 6 collections of training data.

We will use PPE to predict the solutions ϕ_1, ϕ_2, v of the RDE system, and check that the prediction results satisfy conservation of C . To the end, we exercise the following experimental design. Consider the product parameter design space with $\mu_1, \mu_2, \alpha, \beta$ each taking on the values 0.5, 1, 1.5. Computations using these 81 outputs constitute the universe of available data. We compute numerical solutions for $(\phi_1(x, y), \phi_2(x, y), v(x, y))$ up to time $t=0.64$. From the 81 computations we extract 31 at random to be used as test data. We train our PPE emulator based on a subset of the 50 remaining computed solutions, and predict ϕ_1, ϕ_2 , and v for the 31 test inputs. Using these emulated outputs (and the symmetry noted above) we calculate

$$C = \int_{-1}^1 \alpha * \phi_1(x, y = 0, T) + \beta * \phi_2(x, y = 0, T) + \alpha * \beta * v(x, y = 0, T) dx$$

and compare this number to C at $t = 0$, to calculate the error in conservation. Specifically, we randomly selected 6 collections of training datasets, trained using either 5, 10, 20, 30, 40 or all 50 available solutions, and examine C based on the emulated output variables. The conservation error is defined as $E = \frac{1}{31} \sum (C^* - C)$ for the selected values of α, β , and C^* is the total mass of the emulated solution. Figure 4 shows this error for three different cases of training and testing data, showing each of the 6 selected collections of training data.

As should be expected, as the number of training sets increases, the conservation error decreases to a value $\sim 10^{-7}$. For reference, the conservation error of computed solution is $\sim 10^{-13}$. For context, when using 50 training datasets, the RMSE (emulated solution minus computed solution) for ϕ_1 and ϕ_2 is about 10^{-8} , and for v is around 10^{-9} .

5. HAMILTONIAN SYSTEMS

In center-of-mass coordinates, the Kepler 2-body problem, the Hamiltonian is $\mathcal{H} = \frac{p_1^2}{2} + \frac{p_2^2}{2} - \frac{1}{\sqrt{q_1^2 + q_2^2}}$ for the positions q_1, q_2 and momenta p_1, p_2 . The equations of motion are

$$\frac{dq_j}{dt} = \frac{d\mathcal{H}}{dp_j} \quad \frac{dp_j}{dt} = -\frac{d\mathcal{H}}{dq_j}$$

$\mathcal{H}$ is conserved in time, and orbits are periodic.

Consider a leap-frog numerical scheme:

$$\begin{aligned} q_j(t + \frac{\Delta t}{2}) &= q_j(t) + \frac{\Delta t}{2} p_j(t) \\ p_j(t + \Delta t) &= p_j(t) + \Delta t U_j(t + \frac{\Delta t}{2}) \\ q_j(t + \Delta t) &= q_j(t + \frac{\Delta t}{2}) + \frac{\Delta t}{2} p_j(t + \Delta t) \end{aligned}$$

with the acceleration $U_j = -q_j/(q_1^2 + q_2^2)^{3/2}$. If the initial data is $q_1(0) = 1 - a$, $q_2(0) = 0$, $p_1(0) = 0$, $p_2(0) = \sqrt{\frac{1+a}{1-a}}$, with timestep $\Delta t = 0.0005$, the leapfrog scheme preserves the Hamiltonian to 4 digits of accuracy, for 50,000 steps, which translates to 4 complete periods of the dynamics. Using the initial data a as a variable parameter, compute solutions for $a = 0, 0.1, 0.2, \dots, 0.9$. From these 10 computed solutions, choose 9 at random as training data; the objective is to emulate the 10th. We find the emulated solution conserves $\mathcal{H}$ to at least 3 digits over the time interval $0.0005 \times 50,000 = 25$, with the largest variation at the times when the orbit (in (q, p) -space) is returning to its starting location. The orbit is periodic to about 10^{-4} , and the error in the prediction – emulated solution – computed solution – is less than 10^{-3} . One example run is shown in Fig. 5.

We remark that the errors were largest if the randomly selected test run was for parameter value $a = 0.0$ or $a = 0.9$, where the PPE was extrapolating (a little) in parameter space. In spite of the larger error, however, periodicity in the dynamics was preserved.

6. PPE AND MATERIAL PHASE CHANGE

Shape memory alloys (SMAs) can recover significant strain during temperature and stress-induced transformations between low temperature martensite and high temperature austenite phases. Martensite and Austenite phases have very different mechanical, electrical, thermal and optical properties due to different crystallographic properties of the two phases, which necessitates the consideration of phase-dependent material coefficients in SMA models with phase transitions. The interested reader may consult [19].

There are a number of materials, including the alloys NiTi, CuAlNi, CuZnAl and FeMnSi, that exhibit shape memory effects. We examine a model of NiTi^{*} due to its superior structural and memory capability. However much of the description and analysis that follows can be applied to other compounds. Smith [19], Chapter 5, develops a model of temperature- and stress-induced phase transformations. His homogenized energy model smoothly joins two locally quadratic energy potentials with a Boltzmann relation for temperature, and rate laws for phase change and temperature evolution. See [19], Section 5.5 for all the governing equations of the model, and [20] for code to solve the governing equations, based on this model and calibrated with experiments.

Consider a material sample composed of an austenite A phase and two martensite variations $M^\pm$. The lattice volume is V , with total mass v , and $x_A(t), x_\pm(t)$ are the volume fractions of the three configurations. At a temperature T a piecewise quadratic Helmholtz free energy is constructed, based on a shear strain ϵ , where $\epsilon = 0$ for

^{*} More generally, one can consider an alloy Ni_xTi_{1-x} , usually with $x \sim [0.47 \text{ to } 0.51]$.

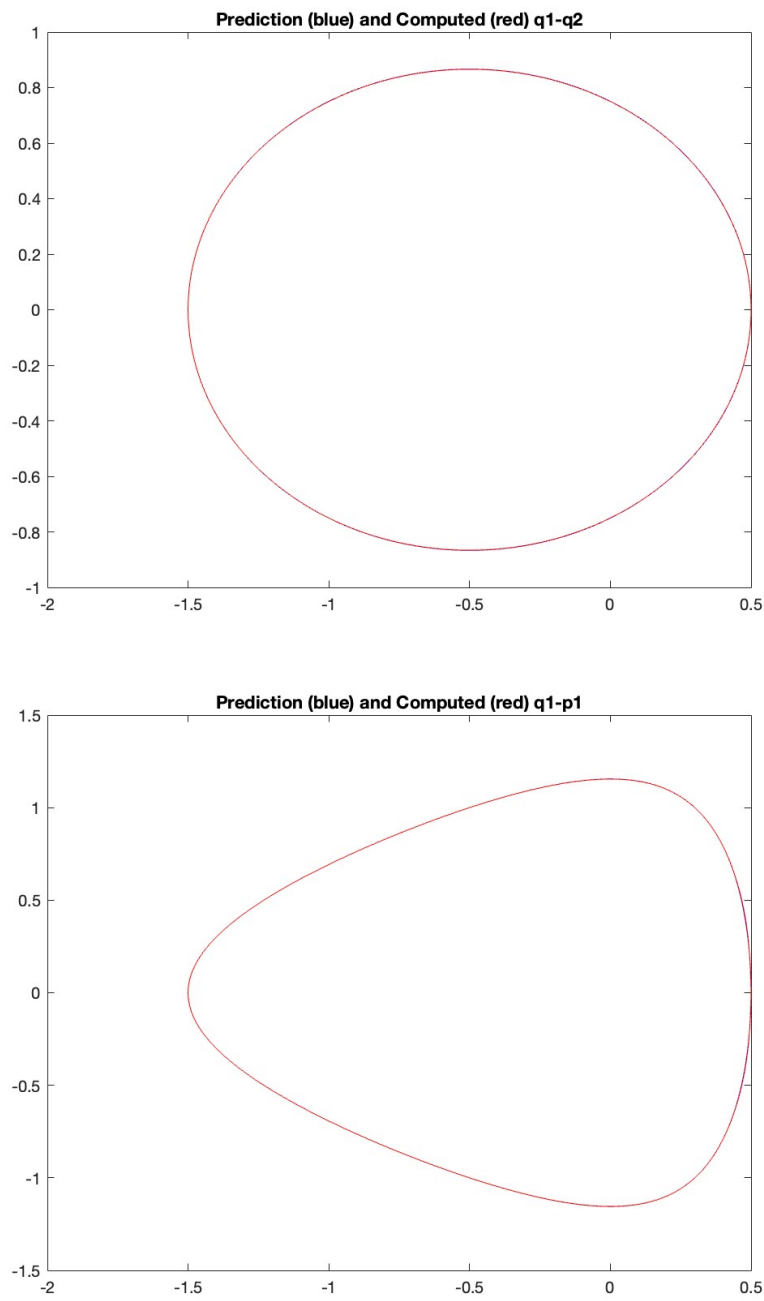


FIG. 5: Predicted (in red) and computed (in blue) plots. Computed solutions are run for 50,000 timesteps of size $\Delta t = 0.0005$, and the emulator provides 50,000 steps for the predicted solution. In the figure, the top image shows q_1 vs q_2 , and the bottom image shows q_1 vs p_1

austenite and $\epsilon = \epsilon_T$ is the equilibrium value for martensite in a stress-free configuration:

$$\Phi(\epsilon, T) = \begin{cases} \frac{Y_M}{2}(\epsilon + \epsilon_T)^2 & \epsilon \leq -\epsilon_M(T) \\ \frac{E_0(T)}{2}(\epsilon + \epsilon_0(T))^2 + \Phi_0(T) & -\epsilon_M(T) < \epsilon < -\epsilon_A(T) \\ \frac{Y_A}{2}\epsilon^2 + \Delta\beta(T) & |\epsilon| \leq \epsilon_A(T) \\ -\frac{E_0(T)}{2}(\epsilon - \epsilon_0(T))^2 + \Phi_0(T) & \epsilon_A(T) < \epsilon < \epsilon_M(T) \\ \frac{Y_M}{2}(\epsilon - \epsilon_T)^2 & \epsilon \geq \epsilon_M(T) \end{cases}$$

Here Y_M, Y_A denote the Young's moduli for the martensite and austenite phases, ϵ_M, ϵ_A are the inflection points in the energy, and $\Delta\beta$ is the energy barrier. The total Gibbs energy adds a work term to the free energy,

$$G(\sigma, \epsilon, T) = \Phi(\epsilon, T) - \sigma\epsilon.$$

Examining this energy, it may be guessed that critical values of stress occur at $\sigma_A(T) = Y_A\epsilon_A(T)$ and $\sigma_M(T) = Y_A(\epsilon_M(T) - \epsilon_T)$.

Local average strains must balance the Gibbs energy with the relative thermal energy through a Boltzmann relation $\mu(G) = Ce^{-GV/kT}$.

Probabilities for a transformation among the different phases, $p_{A\pm}(\sigma, T), p_{\pm}(\sigma, T)$ are defined by the appropriate value of μ , normalized over all possible strain configurations.

Finally the evolution of phase transformations is given by rate laws:

$$\begin{aligned} \dot{x}_-(t) &= p_{A-}x_A(t) - p_-x_-(t) \\ \dot{x}_+(t) &= p_{A+}x_A(t) - p_+x_+(t) \\ \dot{x}_A(t) &= -p_{A-}x_A(t) + p_-x_-(t) - p_{A+}x_A(t) + p_+x_+(t) \end{aligned} \quad (8)$$

Thermal evolution is also given by a rate equation

$$mc(t)\dot{T}(t) = -\Omega(T - T_e(t) + J(t) - \sum h_\alpha \dot{x}_\alpha$$

where the first term defines heat exchange with the environment, J characterizes Joule heating, and the last term characterizes the heat generated during phase transformation with specific enthalpy h_α .

At a specified temperature, a stress is applied. As the temperature is lowered we observe a change in the deformation, depending on the initial and final temperatures and the amplitude of the applied stress $-(T_0, T_1, \sigma)$, resp.

Fig 6 shows the strain vs. time, and stress vs. strain, for two sets of initial temperature and applied stress, exhibiting a hysteresis effect.

To study the accuracy of PPE for this model, we focus on the strain evolution as our quantity of interest. We select testing and training inputs among the (T_0, T_1, σ) triples. For temperatures $T_0 = 400, 401, \dots, 440K$ we select intervals of change ΔT to characterize numerical experiments. So, for example, $\Delta T = 20$ means that $T_0 = 400, 420, 440K$ are selected. Thus when ΔT is relatively large, the corresponding training sample set is small, and vice versa. We then select combinations T_0, σ as inputs. Other values of (T_0, T_1, σ) are used as test inputs. Fig 7 shows PPE predictions based on a fixed applied stress and 3 different temperature intervals: $\Delta T = 20, 10, 5$ in yellow, green and blue, respectively; the simulation result is in red. (Note: Because the starting strain values for each solution are almost identical, we ignore the first 50 time steps and only consider time steps from 51 to 401.) Not surprisingly, reducing the temperature gap ΔT in the training data yields an emulation that is closer to a simulated result.

6.1 PPE and errors

Without an exact solution or a conserved quantity against which to compare, we use the temperature step ΔT training structure in order to estimate an effective rate of convergence for the emulator, using a Richardson extrapolation.

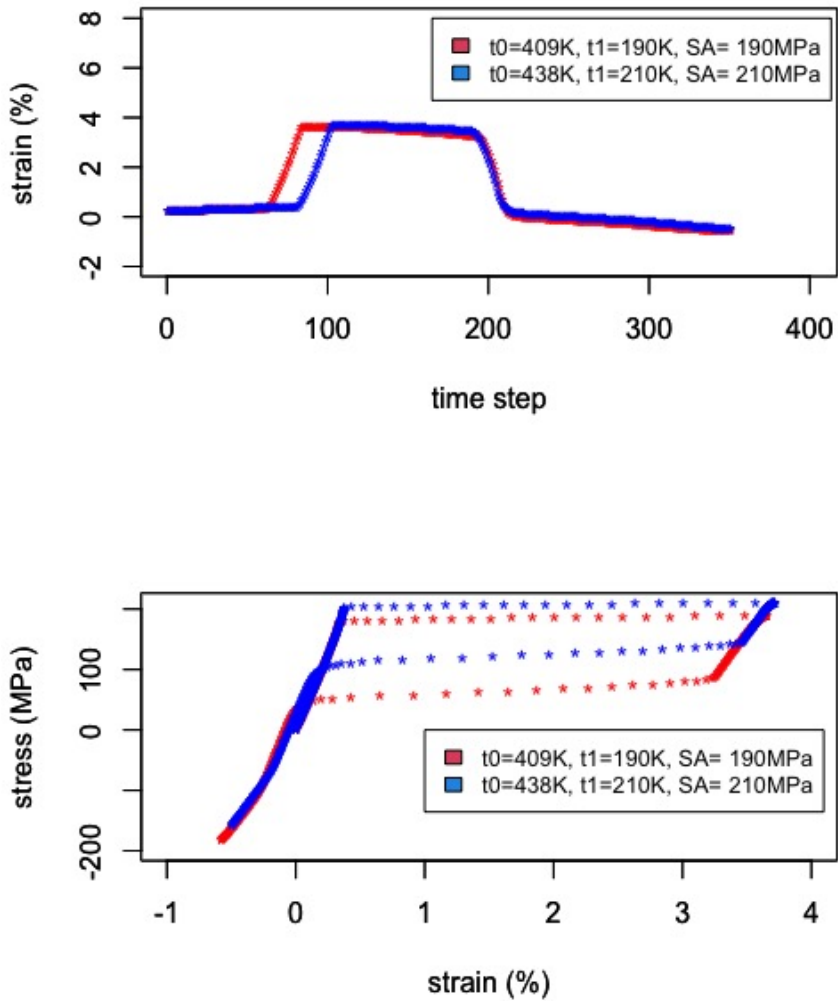


FIG. 6: Strain versus time, and stress-strain hysteresis curves, for two different values of temperature and applied stress.

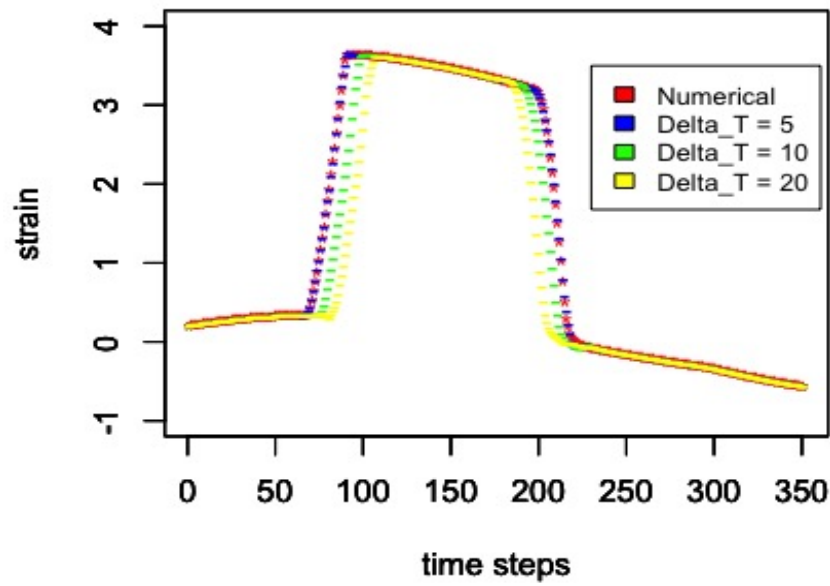


FIG. 7: PPE prediction of the strain compared with simulation results. Three sets of training data, with successively smaller temperature gaps, are shown in yellow, green, and blue, resp. The simulation result is plotted in red.

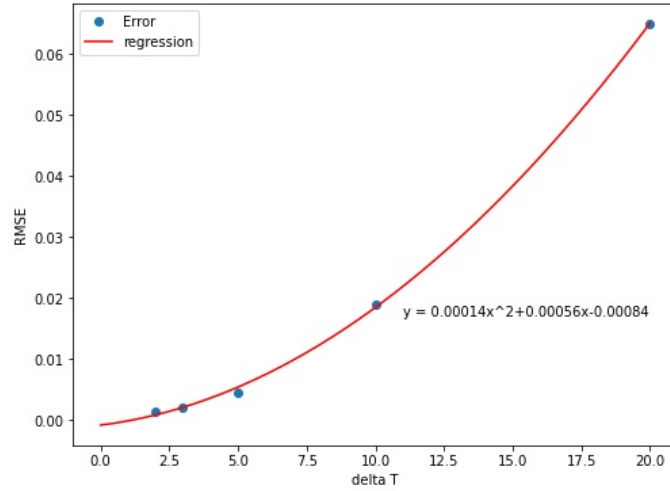


FIG. 8: A best fit polynomial approximation as a function of the temperature gap $\Delta T = 20, 10, 5, 3, 2$.

That is, we ask how rapid is convergence in a realistic range of parameter spacing, for a highly nonlinear model system? Consider the mean squared error in emulation, $(\int_t (\epsilon^* - \epsilon)^2 dt)^{\frac{1}{2}}$ where ϵ^* is the emulated strain as a function of time, across all of the spatial dimension, and ϵ is the simulated strain. We substitute a sum over time steps for the time integral. We postulate an emulation error of the form $\mathcal{E}_{\Delta T} = C(\Delta T)^P$ for constant C, P . Manipulating the ratio $D = \frac{\mathcal{E}_{\Delta T} - \mathcal{E}_{\Delta T/2}}{\mathcal{E}_{\Delta T} - \mathcal{E}_{\Delta T/2^2}} = \frac{1}{1 + (\frac{1}{2})^P}$, we can estimate the effective convergence P .

Fixing σ then, we run simulations with $\Delta T = 20, 10, 5, 2.5$ (where the $\Delta T = 2.5$ error is the average of the errors of $\Delta T = 2, 3$). Using these ΔT values yields $P \approx 2.4$. We can also make a best fit polynomial approximation of the data. Fig. 8 shows this best fit, a quadratic in ΔT .

To gain some intuition about this convergence, consider an ODE

$$\frac{du}{dt} = f(t, u; \mathbf{q}) \quad (9)$$

with initial condition $u(t = 0) = u_0$ and parameter vector $\mathbf{q}$. Solve this equation to obtain a solution at discrete times $0 = t_0, t_1, t_2, \dots, t_J = T$ and for parameters $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_K$. PPE makes a prediction of the solution for any time t not necessarily one of the t_j , and for a parameter $\mathbf{q}^*$.

A Taylor series approximation suggests the total error in approximation is a sum of errors, of differing order, in Δt and $\Delta \mathbf{q}$. The numerical solver is assumed to have $O(\{\Delta t\}^Q)$ accuracy. To estimate the accuracy in parameter space, first define $\Delta \mathbf{q}$ to be the diameter of the the largest ball that “fits” between parameter points in the d -dimensional space of $\mathbf{q}$. To gain some intuition, it is useful to remember that, using a power exponential with power 2, a GaSP is, essentially, a RBF with a squared exponential as the basis functions. Approximation theory suggests that RBF with squared exponential basis functions leads to an exponential convergence rate in $\Delta \mathbf{q}$, at least in the asymptotic range. The default exponent in *ppgasp* is 1.9 and not 2, but this intuition ought to hold approximately (see [21, 22] and references therein). The extrapolation results above yield an effective convergence rate of $P \approx 2.4$. Of course we can improve the numerical solver, to increase its order or convergence. Reducing the emulator approximation error depends on selection of appropriate correlation function and $\Delta \mathbf{q}$. It is typical that space-and-timesteps are small, but sampling in parameter space is usually sparse, and the total error in an emulation is dominated by $\Delta \mathbf{q}^P$ with an effective convergence rate P .

7. CONCLUSIONS

We have studied the output of Parallel Partial Emulation, and demonstrated that emulated output fields retain several properties possessed by simulation outputs. In particular we show that conservation properties – such as norms of the outputs – are preserved in the emulation process, without the need for any additional constructs to enforce this conservation. In retrospect this may not be too surprising – a GaSP, because it is a BLUP, is a linear combination of the training data. If each of the training datasets exhibits conservation, one might think the prediction likewise should. Such intuition certainly guides one’s consideration of conservation in an application such as Burgers’ equation in Sec. 3. However, this intuition breaks down when studying the reaction-diffusion system of Sec. 4. For this system, different input parameters lead to different total mass C . The PPE makes a prediction for a q^* which is not among the training data, and which possesses a mass which is not equal to the mass of any of the training data. Nevertheless, the parallel partial emulated solution accurately predicts and conserves the correct mass.

In spite of the successes of Gaussian process emulation, there is some hesitancy to applying emulated predictions in sensitive applications because of the accuracy in parameter space. Our results show that PPE outputs can accurately emulate space-time fields arising as solutions to highly nonlinear systems of differential equations. In practical applications, the error between simulations and PPE outputs depends principally on the gap between the parameter values used in training. In these sensitive applications, we suggest employing a hierarchical experimental design. That is, an initial design can provide a broad sampling of parameter space, and PPE predictions can identify a region of particular interest. Additional simulations in this region can reduce PPE error to acceptable size for the application at hand. These findings should help disciplinary scientists when considering design of experiment, and provide confidence when using PPE as a surrogate, even when high fidelity approximations are required.

ACKNOWLEDGMENTS

This manuscript includes a portion of the Ph.D. dissertation of Y. Gao in the Department of Materials Design and Innovation at the University at Buffalo. The work reported here was supported in part by NSF awards 2053874 and 1821311. The authors thank Jim Berger for sharing his insights regarding PPE, and Ralph Smith for his suggestions during the execution of this research, and for his generous sharing of data and code. The authors also thank the reviewers for their careful reading of the manuscript and for suggestions that significantly improved the manuscript.

REFERENCES

1. Xu, D. and Karniadakis, G., Modeling uncertainty in flow simulations via generalized polynomial chaos, *J. Comput. Phys.*, **187** p.137-167, 2003.
2. Raissi, M., Perdikaris, P. and Karniadakis, G.E., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.*, **378** p.686–707, 2019. doi.org/10.1016/j.jcp.2018.10.045
3. Sacks, J., Welch, W., Mitchell, T. and Wynn, H.P., Design and analysis of computer experiments *Stat. Sci.*, **4** p.409-423, 1989.
4. Santner, T.J., Williams, B.J. and Notz, W.I., *The Design and Analysis of Computer Experiments* Springer, 2003.
5. Gu, M. and Berger, J.O., Parallel partial Gaussian process emulation for computer models with massive output. *Ann. Appl. Stat.*, **10** p. 1317–1347, 2016.
6. Carlberg, K, Choi, Y. and Sargsyan, S., Conservative model reduction for finite-volume models, *J. Comput. Phys.*, **371** p.280–314 2018.
7. Mao, Z., Jagtap, A. and Karniadakis, G., Physics-informed neural networks for high-speed flows *Comp. Methods Appl. Mech. Engrg.*, **360** p.393–411, 2020. doi.org/10.1016/j.cma.2019.112789
8. De Ryck, T., Mishra, S. and Molinaro, R., wPINNs: Weak Physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws ArXiv abs/2207.08483, <https://api.semanticscholar.org/CorpusID:250626920>

9. Rodriguez-Torrado, R., Ruiz, P., Cueto-Felgueroso, L., Green, M.C., Friesen, T., Matringe, S. and Togelius, J., Physics-informed attention-based neural network for hyperbolic partial differential equations: application to the Buckley–Leverett problem *Scientific Reports*, **12** p.7557, 2022. doi.org/10.1038/s41598-022-11058-2
10. Gramacy, R., *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences* Chapman Hall/CRC, 2020.
11. Bayarri, M.J., Berger, J.O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R.J., Paulo, R., Sacks, J. and Walsh, D., Computer model validation with functional output *The Annals of Statistics*, **35** p.1874–1906, 2007.
12. Conti, A. and O’Hagan, A., Bayesian emulation of complex multi-output and dynamic computer models *J. Stat. Planning Inference*, **140** p.640–651, 2010.
13. Rougier, J., Efficient emulators for multivariate deterministic functions *J. Comput. Gr. Stat.*, **17** p.827–843, 2008.
14. Bayarri, M.J., Berger, J.O., Calder, E.S., Dalbey, K., Lunagómez, S., Patra, A.K., Pitman, E.B., Spiller, E. and Wolpert, R.L., Using Statistical and Computer Models to Quantify Volcanic Hazards, *Technometrics*, **51** p.402–413, 2009.
15. Gu, M., Palomo, J and Berger, J.O., RobustGaSP: Robust Gaussian Stochastic Process Emulation in R *The R Journal*, **11**, 2018. doi.org/10.32614/RJ-2019-011
16. LeVeque, R.J., *Finite Volume Methods for Hyperbolic Problems* Cambridge University Press 2002.
17. Davis, S.F., Simplified second-order Godunov-type methods, *SIAM J. Sci Stat. Comput.*, **9** p.445–473, 1988.
18. Everett, R., Flores, K.B., Henscheid, N., Lagergren, J., Larripa, K., Li, D., Nardini, J.T., Nguyen, P.T.T., Pitman, E.B. and Rutter, E.M., A tutorial review of mathematical techniques for quantifying tumor heterogeneity *Math. Biosci. Eng.*, **17** p.3660–3709, 2020. [doi:10.3934/mbe.2020207](https://doi.org/10.3934/mbe.2020207)
19. Smith, R., *Smart Material Systems* SIAM 2005.
20. Smith, R., Smart Materials Database https://rsmith.math.ncsu.edu/Smart_Materials_Database/index.html
21. Anjyo, K. and Lewis, J.P., RBF interpolation and Gaussian process regression through an RKHS formulation, *J. Math Industry*, **3** p.63–71, 2011.
22. Wu, Z.-M. and Schaback, R., Local error estimates for radial basis function interpolation of scattered data *IMA J. Numer. Anal.*, **13** p.13–27, 1993.