EasyChair Preprint

№ 16

Leveraging compute clusters for large-scale parametric screens of reaction-diffusion systems

Md Shahriar Karim, Hans G. Othmer and David M. Umulis

March 16, 2018

Abstract—**Reaction-diffusion (RD) models are widely used to study the spatio-temporal evolution of pattern formation during development. Nonlinear RD models are usually analytically intractable, and require numerical solution methods. Interrogation of RD models for a large physiological range of parameters covers many orders of magnitude- establishing situations where solutions are stiff and solvers fail to provide accurate results to the time-dependent problem. The spatial dependence of these parameters, and the nonlinearity of the underlying dynamics, impose additional challenges. We developed an efficient approach of simulating stiff RD models of pattern formation and we used supercomputer clusters to carry out a large scale screen of spatially varying parameters for biological pattern formation. The approaches outlined herein are applicable to any systems biology problem requiring numerical approximation of RD equations with spatially non-uniform properties and stiff non-linear reactions.**

Index Terms—**Reaction-diffusion models, Systems biology, Cluster-computing, *In-silico* data.**

I. Introduction

The ever increasing use of mathematical modeling in the analysis of biological systems has facilitated our understanding in many ways. Computational models are now used to consider the interactions of components within and between cells in several ways including testing hypotheses, generating new hypotheses and designing new experiments. [1]–[21].

Although computational models (for instance, RD systems) produce insights into how system behavior is regulated by the interaction network, the complexity of systems can prevent decisive conclusions. For instance, because numerous factors (temperature change, pH variation, lack of nutrients) can change the dynamics of biochemical process, it is difficult to determine the exact kinetics of physiological processes. This demands an exhaustive screen on the physiological range of parameter values which, due to the sheer number of simulations needed, imposes a large computational burden. Parameters may vary over space and, as parameters are chosen from a large range, the parameters in any given simulation may be very small, very large, or both- conditions that create numerical stiffness. Taken together, the numerical screening requires a fast, accurate, and efficient simulation strategy for millions of realizations of RD models to generate trustworthy *in-silico* data within a stipulated time.

To efficiently conduct a large-scale screen of RD models, this work integrates a multi-step method and implemented it in CVODE [22] on super-computer clusters. The rest of the paper is organized as follows. First, we provide an example generic RD system in section II that is useful in many systems, and in many contexts. It is followed by a precise outline of the exact simulation strategy in Section III. Section IV demonstrates the efficacy of the strategy with an example system from systems biology research that attempts to identify scaling mechanisms of pattern formation. Finally, this paper concludes with a discussion along with a few potential enhancement to fine tune the performance of the proposed strategy.

II. Reaction-diffusion models

Let's consider a system S with impermeable boundaries, where N species of concentration $C = [C_1, C_2 \ldots C_N]$ that transport within S by diffusion. The spatio-temporal evolution of any component k can be represented using the following equation:

$$\begin{aligned} \tfrac{\partial C_k}{\partial t} &= \nabla(D_k(\mathbf{C}, \mathbf{x}, \mathbf{P})\nabla C_k) + R(\mathbf{C}, \mathbf{x}, \mathbf{P}) \ \text{ in } S \\ \mathbf{j}_k(\mathbf{C}, \mathbf{x}, \mathbf{P}) &= -D_k(\mathbf{C}, \mathbf{x}, \mathbf{P})\nabla C_k \ \text{ on } \partial S \\ \text{where, } \nabla &= \hat{\mathbf{e}_x}\tfrac{\partial}{\partial x} + \hat{\mathbf{e}_y}\tfrac{\partial}{\partial y} + \hat{\mathbf{e}_z}\tfrac{\partial}{\partial z} \end{aligned} \tag{1}$$

with appropriate initial conditions. Here, $R(\mathbf{C}, \mathbf{x}, \mathbf{P})$ contains the interaction terms, with $\mathbf{P}$, $\mathbf{x}$ and $\mathbf{m}$ representing the kinetic parameters, spatial coordinates, and components interacting within S respectively. For a spatially-dependent diffusivity, the 1D form of Eq.2 simplifies to:

$$\begin{aligned} \frac{\partial C_k}{\partial t} &= \frac{\partial}{\partial x}\left(D_k(\mathbf{C}, \mathbf{x}, \mathbf{P})\frac{\partial C_k}{\partial x}\right) + R(\mathbf{C}, \mathbf{x}, \mathbf{P}) \\ &= D_k(\mathbf{C}, \mathbf{x}, \mathbf{P})\frac{\partial^2 C_k}{\partial x^2} + \left(\sum_l \frac{\partial D_k}{\partial C_l}\frac{\partial C_l}{\partial x}\right)\frac{\partial C_k}{\partial x} \\ &+ R(\mathbf{C}, \mathbf{x}, \mathbf{P}) + ICs + BCs \end{aligned} \tag{2}$$

Here, IC's and BC's are necessary initial and boundary conditions respectively. Nonlinearity of Eq.2 often makes the analytical solution intractable, and the underlying dynamics of the system is numerically approximated. To obtain the approximate solution, finite difference method (FDM) or finite element method (FEM) can be used. Discretization of the PDE system, as in Eq.2, translates the problem from a boundary value parabolic PDE system to an initial value problem of ordinary differential equations (ODEs) of the form:

$$C^{'}(t) = f(t, C(t)), C(t_0) = C_0 \tag{3}$$

A general form of the one-step method of Eq.3 is as follows [23]:

$$C_{i+1} = C_i + \Delta t Z(f, t_i, C_i, C_{i+1}, \Delta t) \tag{4}$$

where $\Delta t = t_{i+1} - t_i$ and Z is the aggregation of all the discretized transport and reaction terms. Any numerical method that involves update terms on the left side and the function evaluation on the right is treated as implicit. To numerically solve such an implicit method, Newton's method [24] is often applied to determine the value at t_{i+1}.

A. Numerical Challenges

The numerical approximation of RD models introduces numerical issues that often limit either the accuracy of the solution, or the ability to solve the problem at all. First, stiffness becomes a major challenge that arises from separation of space and time-scales that lead to widely varying eigenvalues of the Jacobian matrix constructed to compute the numerical solutions. Secondly, when carrying out a large-scale parametric screen, the amount of data, and the time required to generate it, limit the types of problems that can be solved. Further explanation of all these challenges are provided below:

1) Stiffness: The stiffness of Ordinary Differential Equation (ODE) System is not precisely defined. Instead, a number of different observations have been proposed to differentiate between a stiff and non-stiff system. Even a stable and well-posed system may behave stiffly when the system is approximated using a discrete approach [25]. From a computational point of view, in [26], stiffness is defined as "... are the problems for which explicit methods don't work". Mathematically, these definitions do not provide a standard measure of stiffness for the system, and a few alternative approaches for stiffness quantification are outlined below:

1) A system could be stiff because of the presence of widely varying time-scales in the system.
2) A stiff system should have no unstable component (that is, all eigenvalues have negative real parts), and one of the stable components should be dominantly stable (meaning, a very large negative real part). Also, the solution will vary slowly with respect to the most negative real part of eigenvalues [27]. The stiffness of the system can be quantified using the stiffness ratio (SR), where SR is defined as the ratio between the magnitude of largest and smallest negative real part of eigenvalues.

Let us consider a system of N differential equations as $C^{'} = AC$. Solution of the system can be written as $C(t) = C_0 e^{At}$. Now, for a diagonalizable matrix A, eigenvalue decomposition provides $A = T\Lambda T^{-1}$, where i^{th} column of T is the eigenvector of eigenvalue λ_i. The diagonal matrix Λ contains a total of N eigenvalues $(\lambda_1, \lambda_2, \ldots \lambda_N)$. With $A = T\Lambda T^{-1}$, the solution C_t takes the following form:

$$C(t) = C_0 e^{T\Lambda T^{-1}} = C_0 T e^{\Lambda} T^{-1} \tag{5}$$

The system is stiff if the exponential term has vastly different time-scales, which is determined by the eigenvalues of A. A more formal definition of stiffness, defining a stiffness ratio (SR), is as follows:

$$SR = \text{Max}|\text{Re}(\lambda_k)|/\text{Min}|\text{Re}(\lambda_k)| \tag{6}$$

Generally, RD models of system length L have two different time-scales– i) diffusion time-scale (defined as, $\tau_D = L^2/D$), ii) reaction time-scale (defined as, $\tau_R = 1/\text{decay rate}(k_\delta)$). The time-scales τ_D and τ_R may change abruptly due to the spatial dependence of both D and k_δ. As a result, RD models are prone to stiffness depending on the relative amounts of these quantities. Also, the stiffness of the system may change in time as components that regulate each other are created or destroyed. Additionally, when parameter values are selected over a wide range, it requires appropriate arrangements for a large-scale and efficient simulation.

2) Mesh size: Numerical approximation of PDEs gives rise to two different errors, namely, i) truncation error due to discretization, and ii) round-off error. The two errors behave oppositely with the mesh spacing used to discretize the spatial domain. If system length L is divided into N mesh points with $h = L/N$, truncation error approaches zero as N becomes larger. While truncation error reduces, the rounding error increases with larger N. An optimum mesh size for RD models balances the errors to keep both at an acceptable level. Additionally, if h is unusually small, it causes a drastic increase in computational cost.

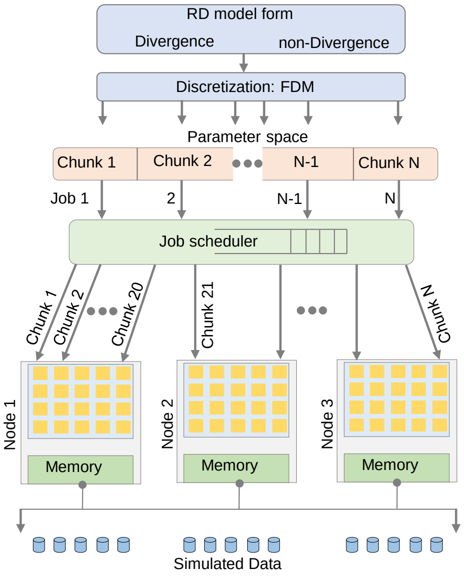

Fig. 1. Cluster-based data generation: The parameter space Ω is subdivided into N equal chunks. The discretized RD model uses these chunks, and each chunk runs on a separate core available in cluster node. Here, a square block in yellow represents individual core, and each cluster node consists of 20 different cores. For simulation, each core is accessed using single user policy to optimize the simulation time.

3) Time constraints on data generation: While the discretization itself creates computational burden because of a sufficiently large N, allowing the system dynamics to evolve for a long enough time (T_{ss}) to reach steady state increases the CPU-time needed in each realization of the RD model. A large T_{SS} is common in systems with spatially varying coefficients, as the time-scales may vary many fold for processes that occur in different regions of space. For instance, if the intrinsic diffusivity and degradation rate of a species in an RD model is $D = 0.1\mu m^2 s^{-1}$ and $k_\delta = 0.0001 s^{-1}$, the associated time-scales for a system of $L = 100\mu m$ is $\tau_d = L^2/D = 27hrs$ and $\tau_R = 1/k_\delta = 2.8hrs$. If spatial regulation reduces D by 10 fold, and the degradation rate is slowed down by 100 fold, τ_D and τ_R change to $500hrs$. This is suggestive of a large T_{SS} requirement. Also, regulation (positive, negative, and none) of biophysical properties, and a large parameter space Ω, increase the total CPU-hours necessary to generate the *in-silico* data.

III. Simulation Strategy

To efficiently simulate RD models, a few factors that cover computational challenges are outlined here, and the steps are schematically shown in Fig.1.

1) *Divergence vs. Non-divergence form:* RD models may either be in a divergence (also known as conservative form) or non-divergence form. Specifically, the derivative of parameters does not appear in the governing

equations [2] for the divergence form, whereas it does in a non-divergence form (Eq.2). Although both forms produce similar qualitative and quantitative data, they may demonstrate model specific performance for convergence failures, accuracy, and required CPU-time needed etc, for each formulation. Therefore, it is important to identify the best model-specific approach for subsequent steps.

2) Discretization: Use FDM or FEM approach to discretize the spatial domain. An optimum mesh size reduces CPU-time required for simulation while restricting errors within an acceptable range through a balance between the round-off and truncation error. This study considers both absolute error (ABS) and the Root Mean Square Error (RMSE), and the corresponding equations are as follows:

$$E_{ABS} = \max_{C_i \in C, i \in 1 \dots N} |C_i - C_i^s|$$

where C_i is the simulated value at spatial point i, and C_i^s is the reference value at i^{th} spatial point. Out of all the values, only the absolute value of the maximum difference is considered. The root mean square error definition is as follows:

$$E_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - C_i^s)^2}$$

where, N is the total number of mesh points, and is varied over a wide range. The absolute error is the absolute difference of the reference and sample data, and only the maximum difference is considered to quantify the error.

3) Subdividing the parameter space: The parameter space Ω is subdivided into M smaller disjoint chunks ($\Omega_1 \cap \Omega_2 = \varnothing$) as follows:

$$\Omega = \bigcup_{i=1}^{M} \Omega_i \tag{7}$$

The choice of M depends on the size of the parameter space, and does not affect the simulation time. However, a careful choice obviously eases the post-processing and management of the generated data.

4) Solver for the ODE system: The computation intensive step in Netwon's method of solving a nonlinear ODE system is the Jacobian J calculation and update J at the each iteration step. To circumvent the cost of J calculation, this work uses a multi-step method, as implemented in CVODE [22], to numerically simulate the set of ODEs obtained from a discretized RD model. Specifically, the method is based on the backward differentiation formula, and implements a matrix-free [28] method that does not require storage and update of the Jacobian J from the underlying linear system during every iteration step. Also, the solver option with the Krylov subspace method [29] further reduces the underlying computational cost. Together, CVODE based simulation reduces CPU-time for each simulation, and hence, is considered as the core of the proposed simulation strategy.

5) Computing-clusters: A set of computer nodes connected together through local area network to facilitate parallel and distributed computing of problems requiring intensive computational power. Clusters are generally optimized for high-performance computations, and their computational power may vary depending on specific hardware configurations being used.

IV. Example simulation: regulation of morphogen signaling

To demonstrate the efficacy of the proposed cluster-based approach, we consider a system of morphogen mediated pattern formation. In many developmental contexts, an extracellular gradient of morphogen provides positional information to a field of cells that leads to a distinct pattern. Specifically, morphogens are a group of molecules secreted from a localized source that undergo transport from the source to create a concentration gradient [9], [10], [30]–[32]. A homogenous group of cells sense their respective positional information by responding to predefined thresholds of the morphogen level [10], [30], [33] and differentiate into distinct cellular patterns (schematically shown in Fig. 2). In developmental contexts, morphogen transport may happen by diffusion, or by other alternative mechanisms [32], [34]. This work assumes that morphogen transport occurs by diffusion, and therefore, the spatio-temporal evolution of a morphogen falls under the reaction-diffusion (RD) paradigm delineated in Eq.2.

Specifically, the morphogen signaling model consists of two tightly correlated species namely, a morphogen (m) and a modulator (M), where m and M spatially regulate each-other's biophysical properties. The model considers positive (pos), negative (neg) and none for the regulation of the production, diffusion and degradation of m and M. If we consider a two species system (m and M) with spatially varying diffusion coefficients D_m and D_M respectively, 1-D representation of the system becomes:

$$\begin{aligned}\frac{\partial m}{\partial t} &= \frac{\partial}{\partial x}\left(D_m \frac{\partial m}{\partial x}\right) + K_m(m, M)\\ \frac{\partial M}{\partial t} &= \frac{\partial}{\partial x}\left(D_M \frac{\partial M}{\partial x}\right) + K_M(m, M)\end{aligned} \tag{8}$$

The model is accompanied by appropriate boundary conditions (B.C.s) and initial conditions (I.C.s). Two flux sources j_m and j_M, located either at $X = 0$ or $X = L$, produce m and M respectively, and are regulated as well. Regulation of all the three properties, each in three alternative ways, generates 729 different regulatory combinations between the properties.

A. Discretization of the example problem

1) Divergence form: The divergence form of the morphogen dynamics in the example model takes the following

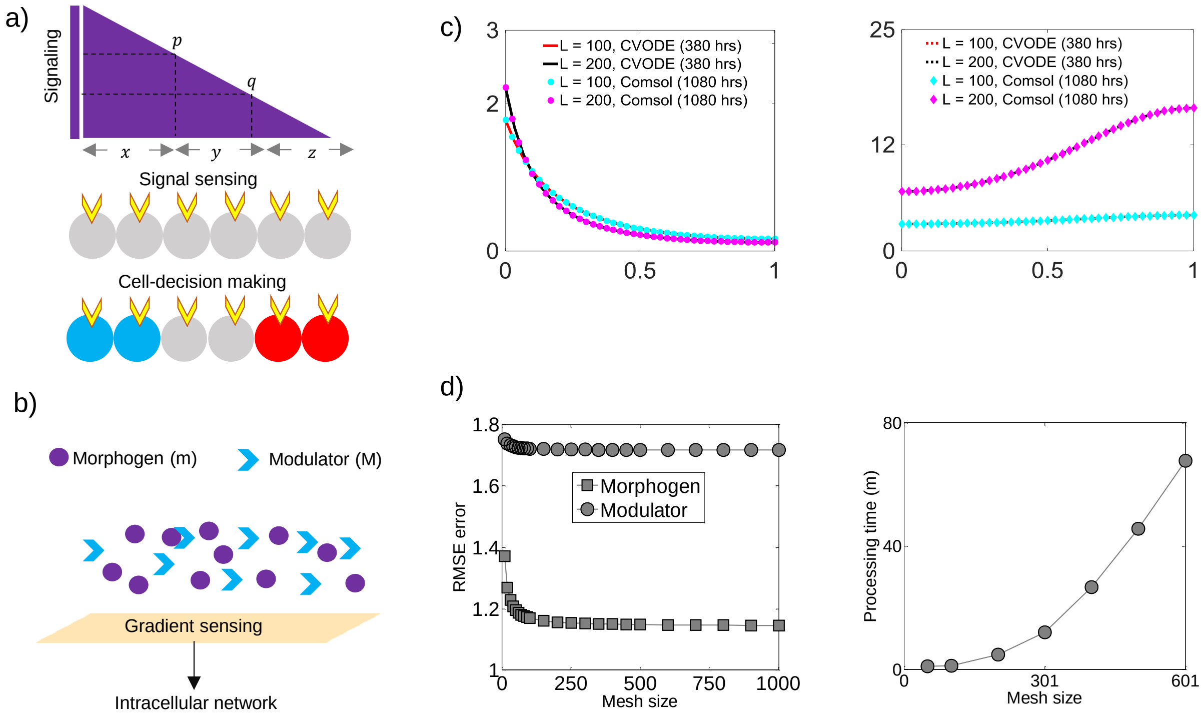

Fig. 2. Example RD problem of morphogen-modulator system, and the relevant specifications for simulation: a) Morphogen mediated pattern formation. A homogeneous field of cells sense morphogen gradient and differentiate intro patterns depending on thresholds p and q. b) Regulation of morphogen signaling by a class of secreted modulators, defined as modulators. A modulator regulates the biophysical properties of a morphogen, and vice-versa. c) Comparison of a morphogen and a modulator concentration as obtained through COMSOL and CVODE simulation. The two methods demonstrate almost equal concentration for both components. Additionally, RD system reaches steady state within the $T_{ss} = 380$ hours, a time that is calculated considering the slowest time-scale of the system. This also indicates that a sufficiently large simulation time is needed to allow the RD dynamics to equilibrate. d) Dependency of RMSE, and the simulation time, on the mesh-size. As observed, simulation time increases (right panel), but RMSE does not decrease after a minimum mesh size.

form:

$$\frac{\partial m}{\partial t} = \frac{\partial}{\partial x}\left(D_m(M)\frac{\partial m}{\partial x}\right) + K_m(M,m) \tag{9}$$

As seen here, the coefficient $D_m(M)$ is spatially varying and its derivative does not appear in Eq.9. In the divergence form, a central differencing scheme [23] is applied as follows:

$$\frac{\partial m}{\partial t} = \frac{1}{\Delta x^2}\{\left(D_m(M)_{i+1/2}\right)(m_{i+1} - m_i) - \left(D_m(M)_{i-1/2}\right)$$
$$(m_i - m_{i-1})\} + K_m(M_i, m_i)$$

where $D(M)_{i+1/2}$ and $D(M)_{i-1/2}$ are the diffusion coefficients approximated at the mid point of $(i+1, i)$ and the midpoint of $(i, i-1)$, respectively, according to the formula given below:

$$D(M)_{i+1/2} = \frac{D(M_{i+1}) + D(M_i)}{2}$$
$$D(M)_{i-1/2} = \frac{D(M_i) + D(M_{i-1})}{2}$$

All the subsequent comparison uses an alias DIV for divergence form of the model discretization.

2) Non-divergence form: In a non-divergence form of PDE, we apply the chain rule to expand (alias, EC) the derivative of the spatially dependent parameters [2] as in Eq. 10.

$$\begin{aligned}\frac{\partial m}{\partial t} &= D_m(M)\frac{1}{L^2}\frac{\partial^2 m}{\partial \xi^2} + \frac{1}{L^2}\frac{\partial D_m}{\partial M}\frac{\partial M}{\partial \xi}\frac{\partial m}{\partial \xi} \\ &\quad \frac{1}{L^2}\frac{\partial m}{\partial \xi}\frac{\partial M}{\partial \xi} + K_m(M,m)\end{aligned} \tag{10}$$

After discretization of Eq. 10 using a central difference scheme along the X direction on a mesh of N grid points, the equation for a morphogen dynamics becomes as follows:

$$\begin{aligned}f_i &= \frac{\partial m_i}{\partial t} = \frac{D_m(M_i)}{L^2}\left(\frac{m_{i+1} - 2m_i + m_{i-1}}{(\Delta\xi)^2}\right) \\ &+ \frac{1}{L^2}\frac{\partial}{\partial M}\left(D_m(M_i)\right)\left(\frac{m_{i+1} - m_{i-1}}{2\Delta\xi}\right) \\ &\quad \left(\frac{M_{i+1} - M_{i-1}}{2\Delta\xi}\right) + K_m(M_i, m_i)\end{aligned} \tag{11}$$

where $0 \leq i \leq N$, $\Delta\xi = 1/N$, and the dimensionless variable ξ is defined as $\xi = x/L$. Similarly, we can obtain the discretized version of a modulator equation by applying a central difference scheme as elaborated here in Eq.11 for a morphogen. Similar equations apply for the discretization of modulator dynamics, and the PDEs of the RD system take an ODE form similar to $\frac{dm}{dt} = f(t, m)$. The numerical approximation is later obtained using CVODE [22].

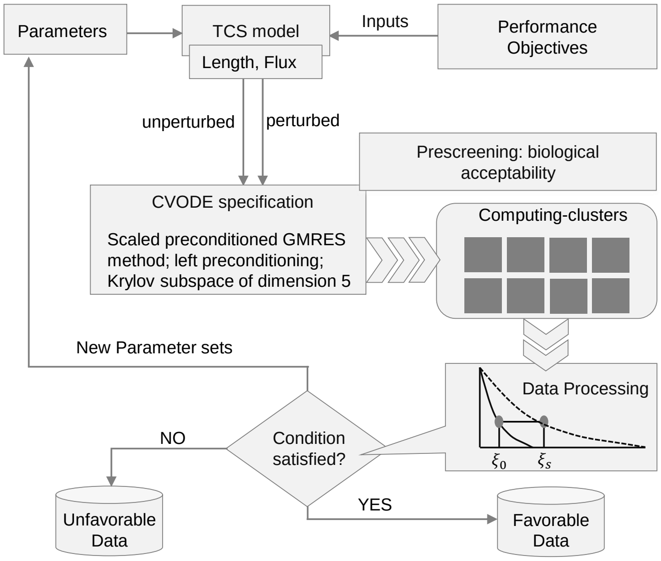

Fig. 3. Data generation and post-processing in the example model: All the steps to generate concentration data for a morphogen and a modulator, and subsequent steps to differentiate between favorable and unfavorable networks are shown here. The unperturbed and perturbed lengths of the system relate to L_1 and L_2 respectively, and for the flux these are ϕ_1 and ϕ_2 respectively.

V. Implementation steps

A. Divergence vs. non-divergence

To compare EC and DIV approaches of model implementation, we consider a system length $L = 600\mu m$, and choose negative regulation of biophysical properties. Simulated data reveal that when a morphogen and a modulator are mostly confined near their respective source of production, DIV implementation of the model approximates the reference concentration (COMSOL Multiphysics data is treated as the reference) near the origin more accurately (comparison between green line and cyan circles) than it does by the EC form of implementation (Figure 2). However, if the species concentration is distributed over the spatial domain both approaches perform well when compared against the COMSOL Multiphysics data.

Comparison of accuracy and simulation time of different approaches (EC and DIV) reveal that they perform similarly when accuracy is compared. It obtained by comparing both absolute error and root mean square error against different mesh size. Specifically, the DIV and EC forms of implementation of morphogen-modulator systems demonstrate similar trend (data not shown). However, the approaches differ in simulation time and simulation failures. For example, EC performs better when all the regulations are positive, whereas DIV works better for negative regulation. The EC form of implementation experiences more simulation failures for mesh size 301 as compared to DIV form of implementation. Furthermore, if the two approaches are compared against the time taken to simulate about 1000 grid points it is seen that EC requires less time than DIV (Table II). DIV is chosen to generate data for the subsequent analysis. However, if a user-supplied precondition matrix is possible, the failures may be removed as supplying a preconditioning matrix in a reaction-diffusion implementation often improves CVODE's performance [35].

B. CPU-hours for clusters

1) Initial calculation: An initial comparison shows different time requirements for different regulation types. Specifically, when both diffusivity and degradation rates are positively regulated, simulations take longer for a common set of parameters. This is shown in Table II. The initial comparison considers 3 motifs out of the theoretically possible 729 motifs, indicating a large CPU time for all the motifs considered. Additionally, the example morphogen-modulator system included about 22.4 million runs for initial screening, and thus, each regulatory motif was simulated for about 30,000 different parameter sets. The initial comparison was conducted on a Intel Core i5 -2050M CPU, 2.5 GHz, RAM 4GB machine with Ubuntu 12.04 operating system.

2) Computing cluster and the CPU hours : To simulate the three alternative forms of the morphogen-modulator system, we used 3 cluster nodes with 20 processors in each. The nodes were equipped with the selected CVODE solver. Example simulations consider single-user access policy as ensured through the workload manager of the supercomputing facility. The complete technical specification of the cluster is as follows– it consists of HP compute nodes with two 10-core Intel Xeon-E5 processors. Each node contains 64 GB of memory. In the cluster, all nodes have 56 Gb FDR infiniband interconnect. Each node runs Red Hat Enterprise Linux 6 and uses Moab Workload Manager 8. For resource and job management, it uses the portable batch system (PBS). We submitted jobs in queue, which we managed based on a customized job scheduler available in the Purdue supercomputing facility. The scheduler reduced the job waiting time, and increased the CPU-hours harnessed per node as well.

Study of the example RD system required a total of about 200 million (M) simulations using the computing cluster. This required about 1681 CPU-days. A detailed description of the problem is shown in Table I. Employing more cluster nodes can reduce the number of actual working days needed for the simulation. Additionally, the processing speed of a cluster is another factor that contributes to reducing the working days required for a simulation– we conducted a comparison between the two clusters (of different hardware facility) available at Purdue, and found about a two-fold performance improvement. That is, in theory, such large scale data generation may be obtainable within a few working days. Based on the availability of the cluster nodes, and other pre-arrangement for job submission, the total CPU-hours harvested during different job submission over a period is distributed as in Fig.4.

VI. Discussion

The proposed strategy efficiently generated *in-silico* data for millions of realizations of a reaction-diffusion system. However, it would require a larger time if the supercomputing

TABLE I
CPU HOURS REQUIRED DURING THE INITIAL DATA GENERATION FOR THE STUDY OF TWO PERFORMANCE OBJECTIVES

Total sim.	Chunks	Jobs	Cluster	Time (hrs)
200 M	34992	34992	3X20	40345
	15552	15552		
	139968	139968		

TABLE II
COMPARISON BETWEEN THE DIVERGENCE AND NON-DIVERGENCE FORM OF THE TCS FOR 301 MESH POINTS: $L_1 = 300\mu m$ AND $L_2 = 600\mu m$

Reg.	M	Time (s)		SIV Failures				RBST Failures			
				L_1		L_2		ϕ_1		ϕ_2	
		EC	D	EC	D	EC	D	EC	D	EC	D
Pos	1000	1367	1587	0	4	0	0	0	4	2	5
Neg	1000	254	234	0	0	13	0	0	0	0	0
No.	1000	168	190	0	0	0	0	0	0	0	0

cluster was not integrated with a stiff solver and other necessary considerations to discretize the underlying PDEs of an RD system. Precisely, the efficiency of this strategy depends on the identification of a suitable mesh size that reduces truncation error and balances that without making the ODE form of the system too large. A comparison shows that reducing the mesh-spacing Δx (that is, increasing mesh size) does not reduce error significantly after a certain level is attained. Instead, it increases the CPU time requirement as shown in Fig. 2. Another caveat is that, although increasing the mesh grid size reduces error (such as, RMSE and ABS), a small mesh spacing Δx may increase other error types (roundoff errors), and may generate additional convergence failures. One common attempt to improve the accuracy of CVODE solution would be reassessing the tolerance values, and preferably, using different tolerance limits for each species of the system. However, when a large number of regulatory relations of the nonlinear RD system studied here, identifying a specific tolerance values for each different stiff ODE system is not tenable. To circumvent this, the Krylov subspace-based iterative implementation and CVODE provide an alternative approach by supplying a preconditioning matrix. This approach improves the accuracy, and it also has the ability to improve convergence, reducing simulation failures.

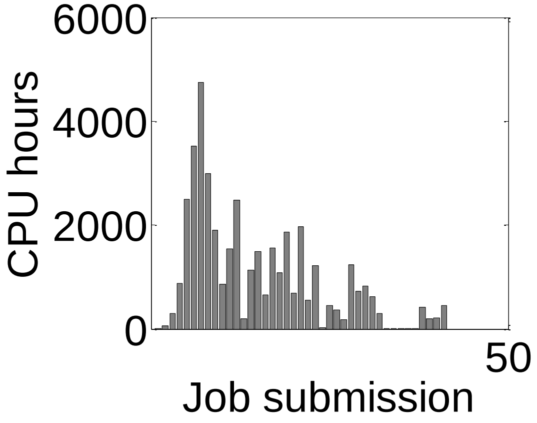

Fig. 4. Distribution of CPU hours harvested over the period of data generation: During data generation for all the models, jobs were submitted during different attempts based on the basic pre-arrangement and preparation needed for the cluster submission. The job submission duration spanned over several days.

Simulation failures were generally within 1-5%, and resulted for many different reasons. However, one crucial reason is that they arise because of improper handling of the stiffness of the system, and can be improved by supplying a customized preconditioner designed using the system equations. The example illustrated in this study uses the default preconditioning technique available in CVODE. In stiff reaction-diffusion systems, a customized preconditioner matrix for a Krylov subspace based iterative solver reduces convergence failures, and improves the performance of the solver as demonstrated in [35].

Together, the strategy demonstrated here harnesses the immense computational power of supercomputing cluster, and the ability of a reduced-storage, matrix free stiff ODE solver (CVODE) to efficiently generate *in-silico* data for reaction-diffusion systems. With adequate customization, the proposed strategy can be used to conduct similar parametric and topological screens for other systems requiring analysis of the reaction-diffusion mechanism in 1D or higher spatial dimensions.

VII. ACKNOWLEDGEMENTS

We would like to thank Aasakiran Madamanchi for his help and comments on the manuscript. We would like to thank the Rosen Center for Advanced Computing at Purdue University for technical support.

REFERENCES

[1] D. M. Umulis and H. G. Othmer, "The role of mathematical models in understanding pattern formation in developmental biology," *Bulletin of mathematical biology*, vol. 77, no. 5, pp. 817–845, 2015.

[2] R. H. Pletcher, J. C. Tannehill, and D. Anderson, *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

[3] J. Dallon, "Multiscale modeling of cellular systems in biology," *Current Opinion in Colloid & Interface Science*, vol. 15, no. 1, pp. 24–31, 2010.

[4] M. Donahue, G. Buzzard, and A. Rundell, "Experiment design through dynamical characterisation of non-linear systems biology models utilising sparse grids," *IET systems biology*, vol. 4, no. 4, pp. 249–262, 2010.

[5] M. Pargett, A. E. Rundell, G. T. Buzzard, and D. M. Umulis, "Model-based analysis for qualitative data: an application in drosophila germline stem cell regulation," *PLoS Comput Biol*, vol. 10, no. 3, p. e1003498, 2014.

[6] L. Parker, D. Stathakis, and K. Arora, "Regulation of bmp and activin signaling in drosophila," in *Invertebrate Cytokines and the Phylogeny of Immunity*. Springer, 2004, pp. 73–101.

[7] P. K. Maini, "Mathematical models in morphogenesis," in *Mathematics inspired by biology*. Springer, 1999, pp. 151–189.

[8] B. Munsky and M. Khammash, "The finite state projection algorithm for the solution of the chemical master equation," *The Journal of chemical physics*, vol. 124, no. 4, p. 044104, 2006.

[9] A. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, pp. 37–72, 1952.

[10] L. Wolpert, *Princiles of Development*. UK: Oxford University Press, 2006.

[11] F. Crick, "Diffusion in embryogenesis," 1970.

[12] H. Inomata, T. Shibata, T. Haraguchi, and Y. Sasai, “Scaling of dorsal-ventral patterning by embryo size-dependent degradation of spemanns organizer signals,” *Cell*, vol. 153, no. 6, pp. 1296–1311, 2013.

[13] T. Gregor, W. Bialek, R. R. de Ruyter van Steveninck, D. W. Tank, and E. F. Wieschaus, “Diffusion and scaling during early embryonic pattern formation,” *Proc Natl Acad Sci U S A*, vol. 102, no. 51, pp. 18 403–18 407, Dec 2005.

[14] A. Lander, Q. Nie, and F. Y. M. Wan, “Do morphogen gradients arise by diffusion?” *Dev. Cell*, vol. 2, pp. 785–96, 2002.

[15] A. D. Lander, Q. Nie, and F. Y. Wan, “Membrane-associated non-receptors and morphogen gradients,” *Bull. Math. Biol.*, vol. 69, pp. 33–54, Jan 2007.

[16] A. Lander, W. Lo, Q. Nie, and F. Wan, “The measure of success: constraints, objectives, and tradeoffs in morphogen-mediated patterning,” *Cold Spring Harbor Perspectives in Biology*, vol. 1, no. 1, 2009.

[17] D. Umulis, O. Shimmi, M. O’Connor, and H. Othmer, “Organism-scale modeling of early drosophila patterning via bone morphogenetic proteins,” *Developmental cell*, vol. 18, no. 2, pp. 260–274, 2010.

[18] D. Umulis, M. Serpe, M. O?Connor, and H. Othmer, “Robust, bistable patterning of the dorsal surface of the Drosophila embryo,” *Proceedings of the National Academy of Sciences*, vol. 103, no. 31, p. 11613, 2006.

[19] T. Kirsch, J. Nickel, and W. Sebald, “Bmp-2 antagonists emerge from alterations in the low-affinity binding epitope for receptor bmpr-ii,” *The EMBO journal*, vol. 19, no. 13, pp. 3314–3324, 2000.

[20] L. Edelstein-Keshet, *Mathematical models in biology*. Siam, 1988, vol. 46.

[21] O. Shimmi, D. Umulis, H. Othmer, and M. O?Connor, “Facilitated transport of a dpp/scw heterodimer by sog/tsg leads to robust patterning of the¡ i¿ drosophila¡/i¿ blastoderm embryo,” *Cell*, vol. 120, no. 6, pp. 873–886, 2005.

[22] S. D. Cohen and A. C. Hindmarsh, “Cvode, a stiff/nonstiff ode solver in c,” *Computers in physics*, vol. 10, no. 2, pp. 138–143, 1996.

[23] B. Bradie, *A friendly introduction to numerical analysis*. Prentice-Hall, 2006.

[24] C. T. Kelley, *Solving nonlinear equations with Newton’s method*. Siam, 2003, vol. 1.

[25] L. Brugnano and D. Trigiante, *On the characterization of stiffness for ODEs*. Watam Press, 1996.

[26] G. Wanner and E. Hairer, *Solving ordinary differential equations II*. Springer-Verlag, Berlin, 1991, vol. 1.

[27] L. F. Shampine and C. W. Gear, “A user’s view of solving stiff ordinary differential equations,” *SIAM review*, vol. 21, no. 1, pp. 1–17, 1979.

[28] P. N. Brown and A. C. Hindmarsh, “Matrix-free methods for stiff systems of ode’s,” *SIAM Journal on Numerical Analysis*, vol. 23, no. 3, pp. 610–638, 1986.

[29] Y. Saad and M. H. Schultz, “Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[30] J. Gurdon, P. Bourillot *et al.*, “Morphogen gradient interpretation,” *NATURE-LONDON-*, pp. 797–803, 2001.

[31] G. Reeves, C. Muratov, T. Sch?pbach, and S. Shvartsman, “Quantitative models of developmental pattern formation,” *Dev. Cell*, vol. 11, pp. 289–300, Sep 2006.

[32] A. Lander, Q. Nie, and F. Wan, “Do morphogen gradients arise by diffusion?” *Developmental Cell*, vol. 2, no. 6, pp. 785–796, 2002.

[33] E. L. Ferguson and K. V. Anderson, “Decapentaplegic acts as a morphogen to organize dorsal-ventral pattern in the drosophila embryo,” *Cell*, vol. 71, no. 3, pp. 451–461, 1992.

[34] S. Restrepo, J. J. Zartman, and K. Basler, “Coordination of patterning and growth by the morphogen dpp,” *Current Biology*, vol. 24, no. 6, pp. R245–R255, 2014.

[35] A. Simmons, Q. Yang, and T. Moroney, “A preconditioned numerical solver for stiff nonlinear reaction–diffusion equations with fractional laplacians that avoids dense matrices,” *Journal of Computational Physics*, vol. 287, pp. 254–268, 2015.