Reduced Order Modeling of Non-Linear Radiation Diffusion Via Proper Generalized Decomposition

Anthony L. Alberti and Todd S. Palmer

Radiation Transport and Reactor Physics Research Group
Oregon State University, 3451 SW Jefferson Way, Corvallis, OR 97333
albertia@oregonstate.edu, palmerts@engr.orst.edu

INTRODUCTION

In recent years, the availability of high performance computing has lead researchers to consider the solution of radiation transport problems of greater complexity and with higher-fidelity [1, 2, 3]. However, an issue with such simulations is that they typically lie in high-parameter spaces and can suffer from what is known as the "curse of dimensionality". For example, consider the solution of a 3D transient neutron transport simulation. If a spatial mesh of one million elements is used in conjunction with 28 energy groups, S_8 level-symmetric angular quadrature, and 10^3 time steps, the problem has over $2{\times}10^{12}$ degrees of freedom. This exponential relationship with the number of the degrees of freedom (i.e. the "curse of dimensionality"), often leads to the requirement of large, costly, high performance computing resources.

Current numerical methods used to overcome this computational expense include lower-order moments-based angular treatments, few-group energy discretizations, and spatial homogenization / upscaling. The applied mathematics community has been investigating other approaches which have not undergone extensive testing in the radiation transport field. Here, we employ one promising approach (Proper Generalized Decomposition) for the solution of a non-linear time-dependent diffusion equation.

THEORY

Model order reduction is a mathematical tool that encompasses methods in which computationally cheaper representations of large-scale systems are created. A key facet of developing a useful reduced order model (ROM) is to ensure that important features of the high-fidelity model are preserved. One challenge with more traditional, well-developed reduced order modeling methods (such as proper orthogonal decomposition, active subspaces, etc. [4, 5]), is that they are data driven and require prior quantitative knowledge of a system of interest. As previously illustrated, many problems in radiation transport are of high dimensionality and obtaining high quality experimental or simulation data can be difficult and costly. For many larger scale simulations - e.g. detailed 3D test reactor models, reactor kinetics, depletion, etc - the overall impact of such *a posteriori* methods are put into question. Rather, it would be ideal to generate reduced order models *a priori* - i.e. without the requirement of pre-existing knowledge of the system behavior.

Proper Generalized Decomposition

Proper Generalized Decomposition (PGD) is an *a priori* decomposition-based method that seeks a solution representation as a series expansion whose components are products of separable functions in each dimension of interest [6]. PGD allows for the solution of a partial differential equation to be decomposed into any number of dimensions including, but not limited to, the individual spatial dimensions, material properties/model parameters (i.e. conductivity, cross sections, etc), initial conditions, and boundary conditions [7]. Figure 1 illustrates this decomposition for a three dimensional problem. If Figure 1 is contextualized to a steady state 3D solution of

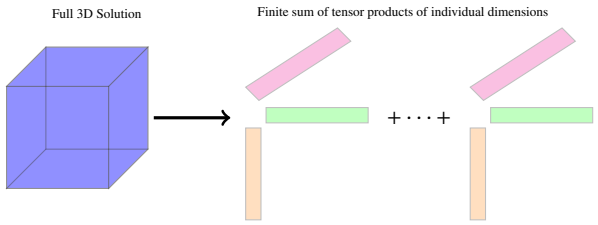

Fig. 1. 3D visual example of PGD.

some physical system in Cartesian coordinates, the number of degrees of freedom of the full matrix solution (viewed as a reference solution) is represented through an exponential relationship. For example, if each spatial dimension (x, y, and z) has 10^3 mesh nodes, the total number of degrees of freedom is $10^3 \times 10^3 \times 10^3 = 10^9$. In contrast, a PGD representation overcomes this "curse of dimensionality" by obtaining a linear relationship of the number of degrees of freedom. If the same discretization in each dimension is maintained, a PGD solution requires only $K(10^3 + 10^3 + 10^3)$ degrees of freedom; where K is a function of the regularity of the exact solution rather than the problem dimension. Furthermore, Chinesta et. al. [6] note that in many applications, K is as small as a few tens.

Pruliere et. al. [8] use PGD to calculate an unknown temperature field across a thin tape with a transient thermal source in one spatial dimension. When compared to a reference analytic solution, the error of this simulation was found to be less than 0.03% with K = 15 tensor products in the PGD expansion. PGD was first introduced by Ammar et. al. in the context of computational rheology [9]. It has since been used in a number of other applications such as 3D resin transfer molding [6], the modeling of gene regulatory networks [10], and in one instance [11] to find the dominant eigenvalue of a nuclear reactor core.

METHODS

As a first examination of PGD in the solution of non-linear diffusion problems, we restrict our attention to a transient, 1-D diffusion equation with an analytic solution.

Model Problem

Consider the following nonlinear reaction-diffusion equation with an inhomogeneous initial condition defined on the following spatial and time domains $\Omega_x \in [-10, 10]$ and $\Omega_t \in [0, 1.024]$,

$$\frac{\partial T}{\partial t} - \frac{\partial T^2}{\partial x^2} = \frac{8}{\delta^2} T(x,t)^2 \left(1 - T(x,t)\right). \tag{1}$$

The boundary conditions for this test problem are $T(-10, t) = 1.0$ and $T(10, t) = 0.0$ [12].

The parameter $\delta > 0$ can be arbitrarily selected, however, by choosing $\delta = 1$ the following analytic solution can be obtained,

$$T(x,t) = \frac{1}{2}\left(1 - \tanh\left[x - 2t\right]\right). \tag{2}$$

In the interest of maintaining an equation structure similar to that of 1-group neutron diffusion, we rearrange the right hand side of Equation 1 and introduce a "pseudo-reaction" term to the left hand side,

$$\frac{\partial T}{\partial t} - \frac{\partial T^2}{\partial x^2} + \left(8T(x,t)^2\right) T(x,t) = 8T(x,t)^2. \tag{3}$$

Here, $8T(x,t)^2$ will serve as both a pseudo "cross section" and as a source.

PGD Mechanics

In PGD, the solution is assumed to be series expansion of products of lower-dimensional basis functions. Because the model problem is two-dimensional, this becomes

$$T(x,t) \approx \hat{T}(x,t) = \sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t) \tag{4}$$

Here, $X_i(x)$ and $T_i(t)$ are the i'th computed basis functions in space and time, respectively.

The PGD solution, $\hat{T}(x,t)$, is built progressively by a process known as "enrichment" [6, 7]. Here we assume that the K terms in the expansion have been previously computed and we develop equations for the computation of the K + 1 term,

$$\hat{T}(x,t) = \sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t) + \mathcal{X}(x)\mathcal{T}(t). \tag{5}$$

If we substitute $\hat{T}(x,t)$ for $T(x,t)$ in our model problem and multiply Equation 3 by an appropriate weight function, we can obtain a non-linear problem that is solved via an alternating direction algorithm [7, 6, 13]. In the present context, this algorithm will consist of two coupled subproblems; an initial value problem (IVP) in time, and a boundary value problem (BVP) in space. Note, the following notation is used for the definitions of the IVP and BVP,

$$\int_\Omega u(x)\, v(x)\, dx = \langle u, v \rangle_{\Omega_x}, \tag{6}$$

and the nonlinear terms of Equation 3, $T(x,t)^2$, are approximated as follows [13],

$$T(x,t)^2 \approx \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t)\right)^2. \tag{7}$$

Time Domain

The weight function for the time domain is $\mathcal{T}(t)^* = v(t)\mathcal{X}(x)$, where $v(t)$ is a test function of arbitrary order and $\mathcal{X}(x)$ is the most recently calculated spatial basis function (a constant if it is the first iteration). An initial value problem results:

$$\begin{aligned} &x_1 \langle v, \mathcal{T}' \rangle_{\Omega_t} + (x_2 + 8x_3) \langle v, \mathcal{T} \rangle_{\Omega_t} \\ &= \sum_{i=1}^{\mathrm{K}} \left[(8x_{6,i} - 8x_{7,i} - x_{5,i}) \langle v, T_i \rangle_{\Omega_t} - x_{4,i} \langle v, T_i' \rangle_{\Omega_t} \right], \end{aligned} \tag{8}$$

where

$$x_1 = \int_{\Omega_x} \mathcal{X}(x)^2 dx, \qquad x_2 = \int_{\Omega_x} \frac{d\mathcal{X}}{dx}\frac{d\mathcal{X}}{dx} dx,$$

$$x_{4,i} = \int_{\Omega_x} \mathcal{X}(x) X_i(x) dx, \qquad x_{5,i} = \int_{\Omega_x} \frac{d\mathcal{X}}{dx}\frac{dX_i}{dx} dx,$$

$$x_3 = \int_{\Omega_x} \mathcal{X}(x)^2 \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t)\right)^2 dx,$$

$$x_{6,i} = \int_{\Omega_x} \mathcal{X}(x) X_i(x) \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t)\right) dx,$$

$$x_{7,i} = \int_{\Omega_x} \mathcal{X}(x) X_i(x) \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t)\right)^2 dx.$$

In this implementation, a strong formulation of Equation 8 was used,

$$\begin{aligned} &x_1 \frac{d\mathcal{T}}{dt} + (x_2 + 8x_3)\mathcal{T}(t) \\ &= \sum_{i=1}^{\mathrm{K}} \left[(8x_{6,i} - 8x_{7,i} - x_{5,i}) T_i(t) - x_{4,i} \frac{dT_i}{dt} \right], \end{aligned} \tag{9}$$

and its solution is computed using the `solve_ivp` module within *scipy* [14].

Spatial Domain

Here, the weight function is defined as $\mathcal{X}(x)^* = u(x)\mathcal{T}(t)$. As before, $u(x)$ is a test function of arbitrary order and $\mathcal{T}(t)$ is the most recently computed time basis function. A boundary value problem results:

$$\begin{aligned} &(t_2 + 8t_3) \langle u, \mathcal{X} \rangle_{\Omega_x} + t_1 \langle u', \mathcal{X}' \rangle_{\Omega_x} \\ &= \sum_{i=1}^{\mathrm{K}} \left[(8t_{6,i} - 8t_{7,i} - t_{4,i}) \langle u, X_i \rangle_{\Omega_x} - t_{3,i} \langle u', X_i' \rangle_{\Omega_x} \right], \end{aligned} \tag{10}$$

where,

$$t_1 = \int_{\Omega_t} \mathcal{T}(t)^2 dt, \qquad t_2 = \int_{\Omega_t} \mathcal{T}(t)\frac{d\mathcal{T}}{dt} dt,$$

$$t_{4,i} = \int_{\Omega_t} \mathcal{T}(t) T_i(t) dt, \qquad t_{5,i} = \int_{\Omega_t} \mathcal{T}(t)\frac{dT_i}{dt} dt,$$

$$t_3 = \int_{\Omega_t} \mathcal{T}(t)^2 \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t) \right)^2 dt,$$

$$t_{6,i} = \int_{\Omega_t} \mathcal{T}(t) T_i(t) \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t) \right)^2 dt,$$

$$t_{7,i} = \int_{\Omega_t} \mathcal{T}(t) T_i(t) \left(\sum_{i=1}^{\mathrm{K}} X_i(x) T_i(t) \right) dt.$$

The solution of this BVP is obtained via linear continuous Galerkin finite elements.

Algorithm

The IVP and BVP problems are coupled and the nonlinear system is solved via a simple fixed point alternating direction iteration, shown below. The initialization for each enrichment

Algorithm 1: Fixed Point Alternating Direction

for $m = 1$ *to* m_{max} **do**
  define $X^{(0)}, T^{(0)}$ (initialization)
  for $p = 1$ *to* p_{max} **do**
    compute $T^{(p)} = G_m(X^{(p-1)})$
    compute $X^{(p)} = F_m(T^{(p)})$
    check convergence of $X^{(p)}T^{(p)}$, ν
    set $X^{(p-1)} = X^{(p)}$
    normalize, s.t. $\|X^{(p-1)}\|^2_{\mathrm{L}_2} = 1$
  end
  normalize $X^{(p)}$ and $T^{(p)}$
  set $X^{(p)} = \mathcal{X}$ and $T^{(p)} = \mathcal{T}$
  set $\phi_{m+1} = \phi_m + \mathcal{X}\mathcal{T}$ and check convergence, ϵ
end

step, $X^{(0)}$ and $T^{(0)}$, are chosen to be a constant (unity). Other studies have initialized using random distributions [15], but we have found that the higher order enrichment steps tend to be more stable with constant initial distributions. Converged enrichment steps are normalized in the following manner,

$$\zeta = \|X^{(p)}\|^2_{\mathrm{L}_2} / (10 - -10) \qquad X^{(p)} = X^{(p)} \frac{\sqrt{\zeta\rho}}{\zeta}$$

$$\rho = \|T^{(p)}\|^2_2 / (1.024 - 0) \qquad T^{(p)} = T^{(p)} \frac{\sqrt{\zeta\rho}}{\rho}.$$

The inhomogeneous initial and boundary conditions are handled in a two step process that can be understood through superposition; the first step captures the initial condition and the second step captures the solution behavior for $t > 0$.

The first step of the superposition process consists of obtaining a single enrichment step that satisfies the inhomogeneous initial condition. These basis functions are then stored as the first enrichment step for the PGD solution. The spatial basis function is computed by solving Equation 1 for $t = 0$ at prescribed spatial points while the time basis function is simply a vector of ones of arbitrary length. Taking the tensor product of these basis functions results in a solution that satisfies the inhomogenous initial condition as well as the constant and homogeneous boundary conditions.

The second step consists of calculating the remaining basis functions that will be used to satisfy Equation 1. These are computed using Equations 9 and Equations 10 via Algorithm 1. Since the initial and boundary conditions for the solution are already satisfied in the first step, the initial and boundary conditions for Equations 9 and Equations 10 are zero for the remaining terms in the expansion.

We have employed a relatively coarse convergence criteria on the nonlinear iterations, $\nu < 10^{-2}$. This selection agrees with previous work [15] and was found to produce suitable approximations of each new enrichment step. With this criteria, enrichment steps tend to converge in approximately 5 iterations; the max number of iterations, k_{max} is set to 10 nonlinear iterations. The maximum number of enrichment steps, m_{max}, is set to 25 and the convergence criteria for the total solution is set to $\epsilon < 10^{-3}$.

$$\nu = \|X^p\| \, \|T^p\| - \|X^{p-1}\| \, \|T^{p-1}\|, \qquad \epsilon = \frac{\|\mathcal{X}\| \, \|\mathcal{T}\|}{\sum_{i=1}^{\mathrm{K}} (\|X_i\| \, \|T_i\|)}$$

RESULTS

To assess the accuracy of the PGD solution, a relative L_2 error was used.

$$\text{Error} = \frac{\|T_{pgd} - T_{ref}\|_{\mathrm{L}_2}}{\|T_{ref}\|_{\mathrm{L}_2}} \tag{11}$$

Figure 2 shows a non-monotonic behavior as the PGD solution adds additional enrichment steps; i.e. as K increases in Equation 5. The spatial domain is discretized with 500

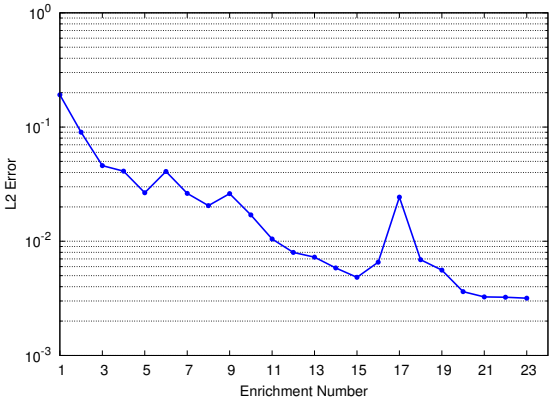

Fig. 2. L_2 error as a function of enrichment step.

elements while the time domain uses adaptive mesh refinement within `solve_ivp`.

To better understand the enrichment process, we observe the PGD solution as a function of enrichment. Figure 3 shows that the higher order enrichment steps are of decreasing magnitude and increasing frequency. This suggests that the overall PGD solution is converging to the true solution and resolving higher frequency information. At enrichment step 23, the PGD solution has converged (i.e. the criterion on ϵ is satisfied).

If the same test problem were to be solved using more traditional discretization methods with the same discretization refinement in space and 500 nodes in time, the total number

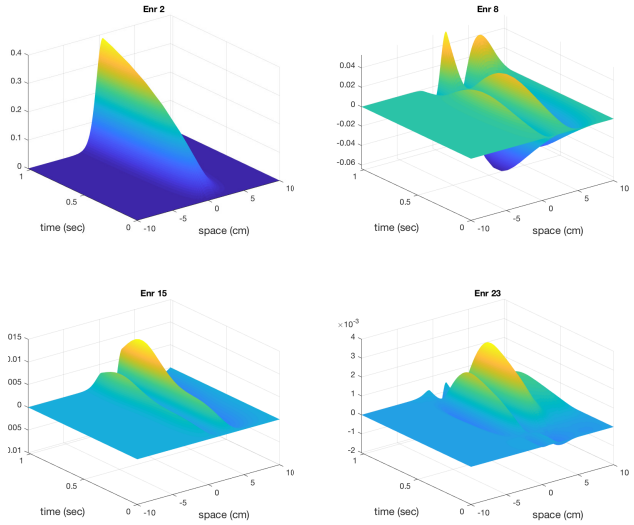

Fig. 3. PGD solution at select enrichment steps.

degrees of freedom would be 250,500. Figure 4 shows that PGD requires a fraction of this total (depending on the desired accuracy).

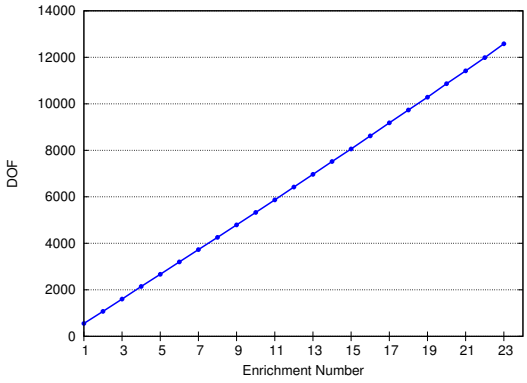

Fig. 4. Degrees of freedom of PGD solution as a function of enrichment step.

CONCLUSIONS

Many radiation transport simulations lie in high-parameter spaces and suffer from the "curse of dimensionality". Developing *a priori* reduced order models for such applications is a developing field. We show here that by using PGD on a PDE with similar structure to that of a neutron diffusion equation, reasonable accuracy and significant data storage gains are achievable. Future work will involve completing mesh refinement studies and applying lessons learned to time and energy dependent neutron diffusion benchmark problems with delayed neutrons.

REFERENCES

1. M. PUSA, "Higher-order Chebyshev Rational Approximation Method and Application to Burnup Equations," *Nuclear Science and Engineering*, **182**, 297–318 (2016).
2. Z. MAUSOLFF, M. DEHART, and S. GOLUOGLU, "Enhances geometric capabilities for the transient analysis code T-Rex and its application to simulating TREAT experiments," *Progress in Nuclear Energy*, **105**, 236–246 (2018).
3. A. SRIVASTAVA, K. P. SINGH, and S. DEGWEKER, "Monte Carlo Methods for Reactor Kinetic Simulations," *Nuclear Science and Engineering*, **189**, 152–170 (2018).
4. P. BENNER, S. GUGERCIN, and K. WILCOX, "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Review*, **57**, *4*, 483–531 (2015).
5. P. G. CONSTANTINE, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM (2015).
6. F. CHINESTA, A. AMMAR, A. LEYGUE, and R. KEUNINGS, "An overview of the proper generatlized decomposition with applications in computation rheology," *Journal of Non-Newtonian Fluid Mechanics*, **166**, 578–592 (2011).
7. F. CHINESTA, R. KEUNINGS, and A. LEYGUE, *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer*, Springer (2014).
8. E. PRULIÈRE, F. CHINESTA, A. AMMAR, A. LEYGUE, and A. POITOU, "On the solution of the heat equation in very thin tapes," *International Journal of Thermal Sciences*, **65**, 148–157 (2013).
9. A. AMMAR, B. MOKDAD, F. CHINESTA, and R. KEUNINGS, "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids," *Journal of Non-Newtonian Fluid Mechanics*, **139**, 153–176 (2006).
10. F. CHINESTA, A. AMMAR, and E. CUETO, "On the use of Proper Generalized Decomposition for solving the multidimensional chemical master equation," *European Journal of Computational Mechanics*, **19**, 53–64 (2010).
11. S. GONZÁLEZ-PINTOR, D. GINESTAR, and G. VERDÚ, "Using proper generalized decomposition to compute the dominant mode of a nuclear reactor," *Mathematical and Computer Modelling*, **57**, 1807–1815 (2013).
12. D. L. ROPP, J. N. SHADID, and C. C. OBER, "Studies of the accuracy of time integration methods for reaction-diffusion equations," *Journal of Computational Physics*, **194**, 544–574 (2004).
13. E. PRULIERE, F. CHINESTA, and A. AMMAR, "On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition," *Mathematics and Computers in Simulation*, **81**, 791–810 (2010).
14. E. JONES, T. OLIPHANT, P. PETERSON, ET AL., "SciPy: Open source scientific tools for Python," (2001–).
15. A. NOUY, "A prior model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, **199**, 1603–1626 (2010).