Model Reduction

Lab-3

Space-time separated representations with Proper Generalized Decomposition

By

Sreekanth Reddy

BAKKA CHENNAIAH GARI

Under the guidance of

Kiran Sagar KOLLEPARA

Post-Doctorant, GeM

Faculty of Engineering

Ecole Centrale de Nantes

October, 2023

CONTENTS

OBJECTIVE **1**

INTRODUCTION **1**

Theory **4**

RESULTS AND DISCUSSIONS **10**

Conclusion **15**

OBJECTIVE

A non-incremental approach to transient problems

Transient problems are often solved by first discretizing the domain using classic numerical methods such as finite differences or finite elements. Then, a time stepping scheme is applied in order to integrate the equations during time. On the contrary, the **Proper Generalized Decomposition (PGD)** provides a different approach to transient problems. The use of space-time separated representations allows computing a transient solution by solving a series of space problems and time problems, only coupled through some scalar coefficients. The time problem can be solved globally, for the whole time interval of interest.

The transient heat problem is solved by **Proper Generalized Decomposition (PGD)** approach.

INTRODUCTION

Multidimensional models encountering in kinetic theory of complex materials may seem too far from everyday practice of computational science. More usual models, however, could be enriched with the addition of well chosen extra-coordinates, thus leading to brand new insights into the physics of the problem.

Imagine for instance solving the heat diffusion equation with the material thermal conductivity being unknown. This could happen because the conductivity has a stochastic nature or simply because it has not been measured prior to the solution of the problem. There are three possibilities:

(i) wait for the conductivity to be measured with the necessary accuracy (a conservative solution, but impractical in many engineering situations);

(ii) solve the equation for many different values of the conductivity to get an overall idea of the behavior of the solution (also impractical when the number of parameters increases); or

(iii) solve the heat equation only once for any value of the conductivity in a given range, thus providing a sort of general solution.

Obviously the third alternative is the most exciting one. To compute this general solution, however, it suffices to introduce the **conductivity as an extra-coordinate of the problem**, playing the same role as the standard space and time coordinates. The PGD separated approximation of the general solution would then be of the form

$$u(x,t,k) = \sum_{i=1}^{N} X_i(x) \cdot T_i(t) \cdot K_i(k) \tag{1}$$

wherein the unknown functions $K_i(k)$ are defined in the domain Ω_k of values for the thermal conductivity.

This procedure works well in practice, and it can be extended to introduce many other extra-coordinates, such as source terms, boundary conditions, initial conditions, and geometrical parameters defining the problem's spatial domain. The price to pay is the corresponding increase of the dimensionality of the resulting model that now contains the standard physical coordinates (space and time) plus all the other extra-coordinates that we decided to introduce. However, the PGD handles without difficulty this increased dimensionality.

Separating the Physical Space

Many useful mathematical models are not inherently defined in a high-dimensional space, but they can nevertheless be treated efficiently in a separated manner. Models defined in cubic domains, or hypercubic domains of moderate dimension, suggest the following separated representation

$$u(x,y,z) = \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \tag{2}$$

Thus, the PGD computation of a 3D solution involves a sequence of 1D solutions for computing the functions $X_i(x), Y_i(y) and Z_i(z)$. This is highly relevant in homogenization problems.

In the case of plate, shell or extruded geometries, one could also consider advantageously the separated approximation:

u(x, y,z) $= \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \cdot Z_i(z)$

For plate geometries, (x, y) are the in-plane coordinates and z is the thickness coordinate. In the case of extruded profiles, (x, y) represents the surface extruded in the z direction. This PGD representation makes it possible to compute fully-3D solutions at a **numerical cost characteristic of 2D solutions**, without any simplifying a priori assumption.

THEORY

The problem we are dealing with is a 1D diffusion equation. where the parameters under study are the position x and time t.

Let u := u(x, t) be the temperature distribution in a bar of length L and during a time interval of interest T. Temperature is governed by:

$$\rho \cdot Cp \cdot \frac{\partial u}{\partial t} - k \cdot \frac{\partial^2 u}{\partial x^2} = f(x, , t) \tag{3}$$

Dirichlet boundary condition:

at x = xL: u(xL, t) = 0,

at x = xR: u(xR, t) = 0

Initial Condition:

u(x, 0) = 0

A rank-d space-time seperated representation is of the form:

$$u^{(d)}(x, t) = \sum_{i=1}^{d} \alpha_i \cdot B_0^i(x) \cdot B_1^i(t) \tag{4}$$

EXERCISE-1

The extended space-time weighted residual:

$$\int_\Omega \int_t (\rho \cdot Cp \cdot \frac{\partial u}{\partial t} - k \cdot \frac{\partial^2 u}{\partial x^2}) \cdot v \cdot dt \cdot d\Omega = \int_\Omega \int_t f(x, , t) \cdot v \cdot dt \cdot d\Omega \tag{5}$$

$$u(x, t) = u^d(x, t) + u^{|}(x, t) \tag{6}$$

Test function:

$$v = B_0^* B1 + B0 B_1^* \tag{7}$$

$$u^{|}(x,t) = B_0(x) \cdot B_1(t) \tag{8}$$

Equation-8 represents the rank-one separated representation. The both separated functions have to be determined.

$$\int_{\Omega}\int_{t}(\rho\cdot Cp\cdot \frac{\partial(u^d + u^{|})}{\partial t} - k\cdot \frac{\partial^2(u^d + u^{|})}{\partial x^2})\cdot(B_0^* B1 + B0 B_1^*)\cdot dt\cdot d\Omega = \int_{\Omega}\int_{t} f(x,,t)\cdot(B_0^* B1 + B0 B_1^*)\cdot dt\cdot d\Omega \tag{9}$$

Alternative Directions Algorithm:

First lets assume B1 is known, which leads to $B_1^* = 0$

$$v = B_0^* B1 \tag{10}$$

Substitute equation-4, equation-8 and equation-10 in the above equation - 9

By scalar product separability, which allows the splitting of space and time parameters. We obtain:

$$\rho \cdot Cp \cdot [\sum_{k=1}^{d} \cdot (\alpha_i B_0^i, B_0^*)(\frac{\partial B_1^i}{\partial t}, B_1) + (B_0, \cdot B_0^*)(\frac{\partial B_1}{\partial t}, B_1)]$$

$$-k \cdot [\sum_{k=1}^{d} \cdot (\alpha_i \cdot \frac{\partial^2 B_0^i}{\partial x^2}, B_0^*)(B_1^i, B_1) - (\frac{\partial^2 B_0}{\partial x^2}, B_0 *)(\cdot B_1, B_1)] - (f_0, B_0^*) \cdot (f_1, B_1) = 0$$

Now we reduce this equation to: (all the known terms are written as the constants)

$$\rho\cdot Cp\cdot[\sum_{k=1}^{d} \cdot(\alpha_i B_0^i, B_0^*)\cdot\alpha_0^i + (B_0, \cdot B_0^*)\cdot\alpha_0] - k(\frac{\partial^2 B_0^i}{\partial x^2}, B_0^*)\cdot\beta_0^i - (\frac{\partial^2 B_0}{\partial x^2}, B_0 *)\cdot\beta_0] - (f_0, B_0^*)\cdot\gamma_0 = 0 \tag{11}$$

where,

$$\alpha_0 = (\frac{\partial B_1^i}{\partial t}, B_1) \tag{12}$$

$$\beta_0 = (B1, B_1) \tag{13}$$

$$\alpha_0^i = \sum_{i=1}^{d} (\alpha_i \cdot \frac{\partial B_1^i}{\partial t}, B_1) \tag{14}$$

$$\beta_0^i = \sum_{i=1}^{d} (\alpha_i \cdot B_1^i, B_1) \tag{15}$$

$$\gamma_0 = (f_1, B_1) \tag{16}$$

Now by converting this weak form, equation-11 to strong form, we obtain:

$$[\alpha_0 I_0 - \beta_0 K_0] B_0 = \gamma_0 f_0 - \sum_{i=1}^{d} \alpha_i [\alpha_0^i \cdot I_0 - \beta_0^i \cdot K_0] \cdot B_0^i \tag{17}$$

Similarly, Compute B1, assuming B0 is known

$$v = B_1^* B0 \tag{18}$$

$$\int_\Omega \int_t (\rho \cdot Cp \cdot \frac{\partial u}{\partial t} - k \cdot \frac{\partial^2 u}{\partial x^2}) \cdot (B_1^* B0) \cdot dt \cdot d\Omega = \int_\Omega \int_t f(x,,t) \cdot (B_1^* B0) \cdot dt \cdot d\Omega \tag{19}$$

Substitute equation-2, equation-6 and equation-16 in the above equation - 17

By scalar product separability, which allows the splitting of space and time parameters. We obtain:

$$\rho \cdot Cp \cdot [\sum_{k=1}^{d} \cdot (\alpha_i \frac{\partial B_1^i}{\partial t}, B_1^*)(B_0^i, B_0) + (B_0, \cdot B_0)(\frac{\partial B_1}{\partial t}, B_1^*)]$$

$$-k \cdot [\sum_{k=1}^{d} \cdot (\alpha_i \cdot \frac{\partial^2 B_0^i}{\partial x^2}, B_0)(B_1^i, B_1^*) - (\frac{\partial^2 B_0}{\partial x^2}, B_0)(\cdot B_1, B_1^*)] - (f_0, B_0) \cdot (f_1, B_1^*) = 0$$

Now we reduce this equation to:

$$\rho \cdot Cp \cdot [\cdot (\alpha_i \frac{\partial B_1^i}{\partial t}, B_1^*) \alpha_1^i + \alpha_1 (\frac{\partial B_1}{\partial t}, B_1^*)] - k \cdot [\cdot \beta_1^i (B_1^i, B_1^*) - \beta_1 (\cdot B_1, B_1^*)] - \gamma_1 \cdot (f_1, B_1^*) = 0$$

where,

$$\alpha_1 = (B0, B_0) \tag{20}$$

$$\beta_1 = (\frac{\partial^2 B_0}{\partial x^2}, B_0) \tag{21}$$

$$\alpha_1^i = (\alpha_i \cdot b_0^i, B_0) \tag{22}$$

$$\beta_1^i = (\alpha_i \cdot \frac{\partial^2 B_0^i}{\partial x^2}, B_0) \tag{23}$$

$$\gamma_1 = (f_0, B_0) \tag{24}$$

Now by converting this weak form to strong form, we obtain:

$$[\alpha_1 G_1 - \beta_1 I_1] B_1 = \gamma_1 f_1 - \sum_{i=1}^{d} \alpha_i [\alpha_1^i \cdot G_1 - \beta_1^i \cdot I_1] \cdot B_1^i \tag{25}$$

The equation-17 & 25 are in the form Ax=b, to compute B0 and B1 alternatively until the convergence.

Evaluating the Convergence Criterion: Alternative directions algorithm

Alternative directions algorithm

```
Initialize: $B_0$ and $B_1$ randomly
Update coefficients $\beta_1$, $\gamma_1$, and $\delta_1$, $B_0 = (\beta_1 M_0 + \gamma_1 K_0)^{-1} \delta_1 f_0$
Update coefficients $\beta_0$, $\gamma_0$, and $\delta_0$, $B_1 = (\beta_0 W_1 + \gamma_0 M_1)^{-1} \delta_0 f_1$
Evaluate convergence criterion $\|B_0^k \cdot B_1^k - B_0^{k-1} \cdot B_1^{k-1}\|$
```

$$\epsilon = ||B_0^k \cdot B_1^k - B_0^{k-1} \cdot B_1^{k-1}|| \tag{26}$$

where,

B_0^k, B_1^k are from current iterations.

$B_1^k - B_0^{k-1}$ are from the previour ieration.

The 2-norm (also known as the Euclidean norm or L2 norm) of a vector can be computed using the dot product.

$$||v||_2 = sqrt(v \cdot v)$$

The "L^2-norm" is reserved for application with a function phi(x),

$$|\phi|^2 = \phi \cdot \phi = \langle \phi | \phi \rangle = \int |\phi(x)|^2 \, dx$$

Similarly, by applying the above formula to equation-26,

$$\epsilon = sqrt(B_0^k \cdot B_1^k - B_0^{k-1} \cdot B_1^{k-1}, B_0^k \cdot B_1^k - B_0^{k-1} \cdot B_1^{k-1})$$

Now,

$$\epsilon^2 = (B_0^k \cdot B_1^k) \cdot (B_0^k \cdot B_1^k) - (B_0^{k-1} \cdot B_1^{k-1}) \cdot (B_0^k \cdot B_1^k), -(B_0^k \cdot B_1^k) \cdot (B_0^{k-1} \cdot B_1^{k-1}) - (B_0^{k-1} \cdot B_1^{k-1}) \cdot (B_0^{k-1} \cdot B_1^{k-1})$$

Now by the scalar product separation property,

$$\epsilon^2 = (B_0^k \cdot B_0^k) \cdot (B_1^k \cdot B_1^k) - (B_0^{k-1} \cdot B_0^k) \cdot (B_1^{k-1} \cdot B_1^k), -(B_0^k \cdot (B_0^{k-1}) \cdot B_1^k \cdot B_1^{k-1}) - (B_0^{k-1} \cdot B_0^{k-1}) \cdot B_1^{k-1} \cdot B_1^{k-1})$$

$$\epsilon^2 = (B_0^k \cdot B_0^k) \cdot (B_1^k \cdot B_1^k) - (B_0^{k-1} \cdot B_0^k) \cdot (B_1^{k-1} \cdot B_1^k), -2 * (B_0^k \cdot (B_0^{k-1}) \cdot B_1^k \cdot B_1^{k-1})$$

where the dot product of two continuous vector functions can be computed by integral,as explained earlier, In the code, it can be implemented as,

```
S_difference = np.sqrt(np.trapz(B0**2, x=x)*np.trapz(B1*B1, x=t)
                     + np.trapz(B0_Old**2,x=x)*np.trapz(B1_Old**2,x=t)
                     - 2*np.trapz(B0*B0_Old,x=x)*np.trapz(B1*B1_Old,x=t))
```

Evaluating the global Convergence Criterion: Greedy enrichment algorithm

Greedy enrichment algorithm

Initialize: B_0 and B_1 randomly

B_0, B_1 — Alternating Directions Algorithm

$u = B_0 B_1$

Evaluate global convergence:

$$\frac{\|p \cdot B_0 \cdot B_1 - B_1 \cdot \Delta B_0 + f\|}{\|f\|}$$

$\|p \cdot B_0 \cdot B_1 - B_1 \cdot \Delta B_0 + f\|$ is computing the residual of the solution, If the residual has not been sufficiently reduced, add one more term. The computation of this is very costly, therefore a simplified stopping criterion is utilised,

Compute,

$$B_0^1 = \frac{B_0}{\|B_0\|}$$

$$B_1^1 = \frac{B_1}{\|B_1\|}$$

$$\alpha_1 = \|B_0\| \|B_1\|$$

The stopping criterion is alpha[term] / alpha[0], we are normalising the alpha coefficient, When it meets the convergence criterion the enrichment of terms will stop. If not we add an enrichment at each step,

$$u = B_0 \cdot B_1 + \alpha_1 \cdot B_0^1 \cdot B_1^1 \tag{27}$$

RESULTS AND DISCUSSIONS

The solution with respect to space at time $t = 1$ second is compared and is shown in Figure 1. The PGD formulation is yielding very accurate results with an optimal number of greedy enrichments.

Parameters Used:

- `Max_terms` $= 5$: Maximum number of enrichments before stopping the global iteration.
- `Max_fp_iter` $= 5$: Maximum alternative directions until B_0 and B_1 are converged.
- `Epsilon` $= 1 \times 10^{-6}$: Tolerance for convergence and numerical precision.
- `Epsilon_tilde` $= 1 \times 10^{-6}$: Tolerance for convergence and numerical precision.

Smaller values of ε and $\tilde{\varepsilon}$ typically lead to higher precision but may increase computational cost and convergence time.

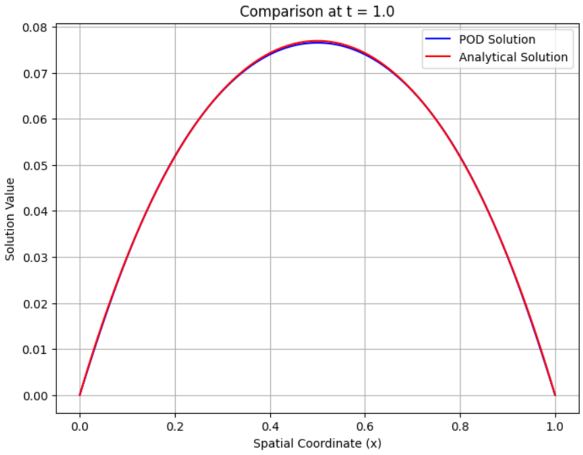

Figure 1: Enter Caption

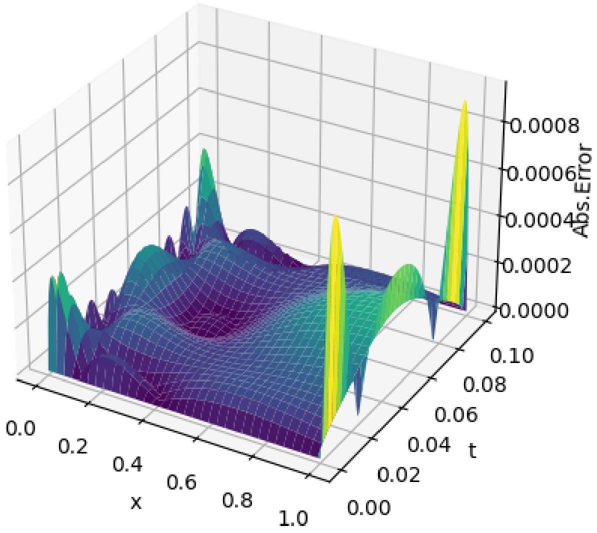

Figure 2: Surface plot of Absolute error w.r.t x ant t

Comparison of the error:

The absolute error is plotted w.r.t parameters x and t and is shown in the figure-2 . We can see how well the PGD solution is close to the analytical solution with very less number of enrichment, in our case its 5.

To Compare the errors, How they are varying w.r.t x and t, A relative error surface plot is demonstrated in the below figure-3.

To visualize how the parameters are varying w.r.t x and t, contours of PGD solution in figure-4 and Analytical solution in figure-5 is shown. We can visualize the contours are very accurate.

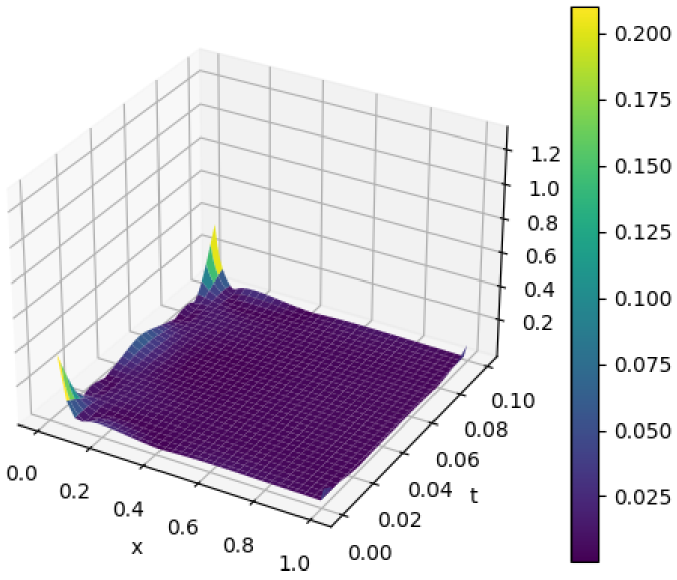

Figure 3: Surface plot of Relative error w.r.t x ant t

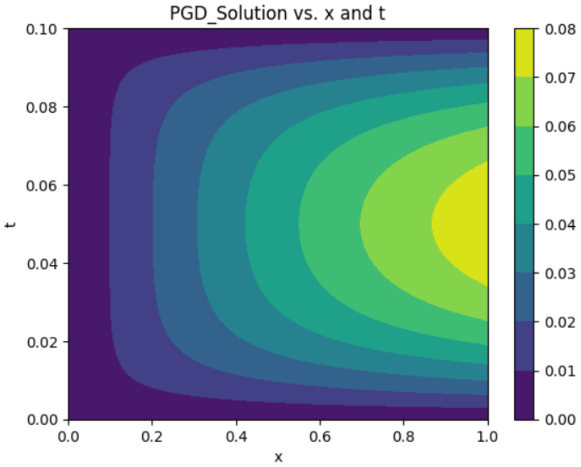

Figure 4: PGD_Contour

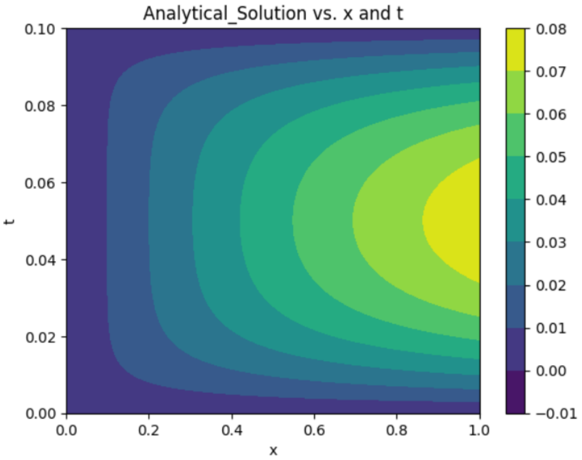

Figure 5: Analytical solution contour

Parametric Analysis:

To better understand the reliability of results, Parametric analysis is conducted.

A range of values for each parameter are defined: max_terms_values = [1, 2, 3, 5, 10, 15, 20] max_fp_iter_values = [1, 2, 3, 5, 10] epsilon_values = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1] epsilon_tilde_values = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1]

The figure.6 represents the heat map of parametric analysis w.r.t norm2. We can visualise how the Norm2 varies with change in Max. Terms and Max. Alternative Direction Iterations. The Max. FP iter should be more than 3 and max terms should be greater than 2 to have a norm2 of the error between the analytical solution and PGD solution to be minimum.

The above figure 7 represents the heat map, with Max. FP iter should be more than 5 and max terms should also be greater than 5 to have a the minimum of the maximum error i.e., infinity norm between the analytical solution and PGD solution.

The plots in this report corresponds to Max. FP iter = max terms = 5 from the parametric analysis.

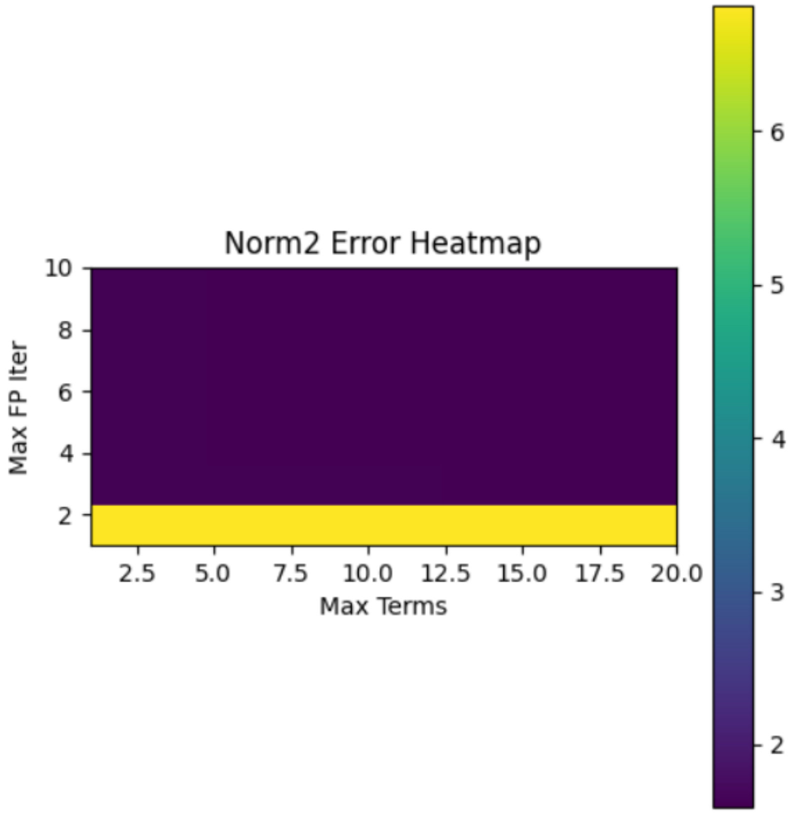

Figure 6: Norm2 Error Heat map

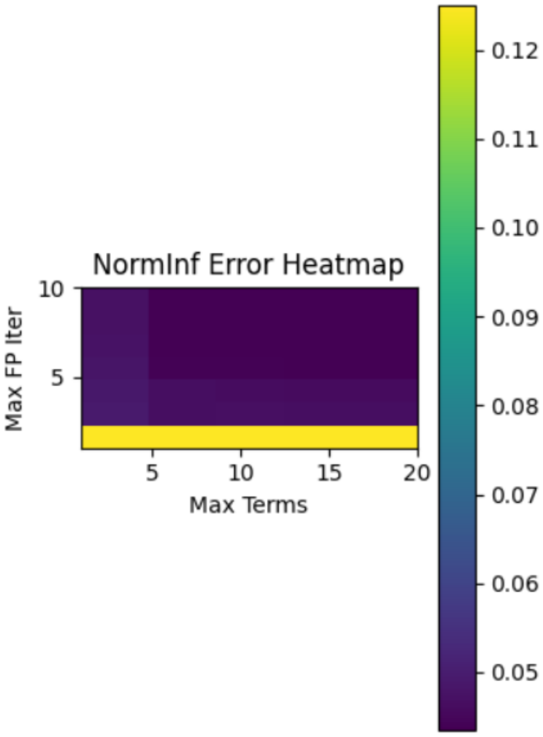

Figure 7: Infinity norm of Error Heat map

CONCLUSION:

In contrast to the Proper Orthogonal Decomposition (POD), which requires data to learn problem-specific approximation spaces, the Proper Generalized Decomposition (PGD) takes an a priori approach by computing problem-specific approximation spaces through the setup of a non-linear optimization problem.

In PGD, the parameters, encompassing both spatial and temporal dimensions, serve as coordinates. This approach can be extended to introduce additional coordinates, such as source terms, boundary conditions, initial conditions, and geometric parameters defining the problem's spatial domain. It is important to note that this extension increases the dimensionality of the resulting model. However, thanks to the scalar separability property, the complexity is effectively reduced to $O(NdD)$, where D represents the number of parameters and d represents the number of enrichments.

A parametric analysis is conducted to visualize the error by considering different convergence criteria. The results of the PGD solution closely approximate the analytical solution, and the error is found to be within the order of 10^{-3}.

Furthermore, the contours depicting how the solution varies with respect to both x and t are showcased, demonstrating a high degree of accuracy in our approach.