LAB 4- PGD SEPARATED REPRESENTATIONS IN CARTESIAN DOMAINS AND PGD CODE GENERALIZATION

BY

SACHIN SRINIVASA SHETTY

Contents

List of Figures

1 Introduction

The Proper Generalized Decomposition(PGD) is implemented for the Space-Space domain. With the help of the PGD, the computational complexity of the Cartesian domain is reduced. In Part-1 of the lab, the PGD is implemented for the Poisson equation in a 2D rectangular domain with homogeneous Dirichlet boundary conditions on four edges. The PGD is also formulated for a general source term.

In Part-2 of the lab, the PGD is obtained from a generalized function,`easy_PGD`. The inputs to the code are in an appropriate tensor structure. This generalized PGD implementation is performed for the Poisson equation. The Convergence study is done for the Alternating directions stagnation criterion and the global convergence criterion.

2 Part 1:PGD separated representations in Cartesian domains

The 2D Poisson equation is considered and the PGD formulation is performed for this problem.The rank-d space time separated is represented as,

$$u^{(d)}(x,y) = \sum_{i=1}^{d} \alpha_i \phi_x^i(x)\phi_y^i(y)$$

The space-space weighted residual of the 2D Poisson equation,

$$\nabla^2 u + f(x,y) = 0$$

$$\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial y^2} = -f(x,y) \tag{1}$$

Computing the new term by introducing

$$u(x,y) = u^{(d)}(x,y) + \tilde{u}(x,y)$$

The equation (1) becomes,

$$\sum_{i=1}^{d} \alpha_i \phi_y^i \Delta\phi_x^i + \phi_y\Delta\phi_x + \sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i + \phi_x\Delta\phi_y + f(x,y) = 0$$

The test function in terms of $\phi_x(x), \phi_y(y)$ and the respective variations,$\phi_x^*(x), \phi_y^*(y)$ is

$$v = \phi_x^*(x)\phi_y(y) + \phi_x(x)\phi_y^*(y)$$

$$v = \phi_x^*\phi_y + \phi_x\phi_y^*$$

Introducing the test function into the above equation and integrating over the space-time domain,

$$\sum_{i=1}^{d} \alpha_i \phi_y^i \Delta\phi_x^i, \phi_x^*\phi_y + \phi_x\phi_y^*) + (\phi_y\Delta\phi_x, \phi_x^*\phi_y + \phi_x\phi_y^*) + (\sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i, \phi_x^*\phi_y + \phi_x\phi_y^*)$$
$$+ (\phi_x\Delta\phi_y, \phi_x^*\phi_y + \phi_x\phi_y^*) + (f(x,t), \phi_x^*\phi_y + \phi_x\phi_y^*) = 0 \tag{2}$$

2.1 Space x Problem

The Alternating Directions algorithm is applied to the above obtained equation, ϕ_y is considered known and the test function reduces to

$$v = \phi_x^* \phi_y$$

So the equation (2) reduces to

$$(\sum_{i=1}^{d} \alpha_i \phi_y^i \Delta\phi_x^i, \phi_x^* \phi_y) + (\phi_y \Delta\phi_x, \phi_x^* \phi_y) + (\sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i, \phi_x^* \phi_y) + (\phi_x \Delta\phi_y, \phi_x^* \phi_y) + (f(x,t), \phi_x^* \phi_y) = 0 \quad (3)$$

The above equation in integral form is

$$\int \sum_{i=1}^{d} \alpha_i \phi_y^i \Delta\phi_x^i \phi_x^* \phi_y dxdy + \int \phi_y \Delta\phi_x \phi_x^* \phi_y dxdy + \int \sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i \phi_x^* \phi_y dxdy + \int \phi_x \Delta\phi_y \phi_x^* \phi_y dxdy + \int f(x,t) \phi_x^* \phi_y dxdy = 0$$

Using Separation properties of the Integrals,we get

$$\int_x \phi_x^* \Delta\phi_x dx \int_y \phi_y \phi_y dy + \int_x \phi_x \phi_x^* dx \int_y \Delta\phi_y \phi_y dy = -\int_x f(x) \phi_x^* dx \int_y f(t) \phi_y dy - \int_x \sum_{i=1}^{d} \Delta\phi_x^i \phi_x^* dx \int_y \alpha_i \phi_y^i \phi_y dy - \int_x \sum_{i=1}^{d} \phi_x^i \phi_x^* dx \int_y \alpha_i \Delta\phi_y^i \phi_y dy$$

The strong form of the above equation is then obtained.This is because the discretization used in this lab is Finite Differences.

$$\Delta\phi_x \int_y \phi_y \phi_y dy + \phi_x \int_y \Delta\phi_y \phi_y dy = f(x) \int_y f(t) \phi_y dy - \sum_{i=1}^{d} \alpha_i \Delta\phi_x^i \int_y \phi_y^i \phi_y dy + \sum_{i=1}^{d} \alpha_i \phi_x^i \int_y \Delta\phi_y^i \phi_y dy \quad (4)$$

2.2 Space y Problem

In the time problem ,ϕ_x is considered known and the test function reduces to

$$v = \phi_x \phi_y^*$$

So the equation (2) reduces to

$$(\sum_{i=1}^{d} \alpha_i \phi_y^i \Delta\phi_x^i, \phi_x \phi_y^*) + (\phi_y \Delta\phi_x, \phi_x \phi_y^*) + (\sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i, \phi_x \phi_y^*) + (\phi_x \Delta\phi_y, \phi_x \phi_y^*) + (f(x,t), \phi_x \phi_y^*) = 0 \quad (5)$$

The above equation in integral form is

$$\int \sum_{i=1}^{d} \alpha_i \phi_y^i \Delta B_0^i \phi_x \phi_y^* dxdy + \int \phi_y \Delta\phi_x \phi_x \phi_y^* dxdy + \int \sum_{i=1}^{d} \alpha_i \phi_x^i \Delta\phi_y^i \phi_x \phi_y^* dxdy + \int \phi_x \Delta B_1 \phi_x \phi_y^* dxdy + \int f(x,t) \phi_x \phi_y^* dxdy = 0$$

Using Separation properties of the Integrals,we get

$$\int_y \phi_y \phi_y^* dy \int_x \Delta\phi_x \phi_x dx + \int_y \Delta\phi_y \phi_y^* dy \int_x \phi_x \phi_x dx = -\int_y f(y)\phi_y^* dy \int_x f(x)\phi_x dx$$
$$- \int_y \sum_{i=1}^{d} \alpha_i B_1^i B_1^* dy \int_x \Delta\phi_x^i \phi_x dx - \int_y \sum_{i=1}^{d} \alpha_i \Delta\phi_y^i \phi_y^* dy \int_x \phi_x^i \phi_x dx$$

The strong form of the above equation is then obtained.

$$\phi_y \int_x \Delta\phi_x \phi_x dx + \Delta\phi_y \int_x \phi_x \phi_x dx = -f(y) \int_x f(x)\phi_x dx$$
$$- \sum_{i=1}^{d} \alpha_i B_1^i \int_x \Delta\phi_x^i \phi_x dx - \sum_{i=1}^{d} \alpha_i \Delta\phi_y^i \int_x \phi_x^i \phi_x dx \quad (6)$$

2.3 Finite Difference Method

The Finite Difference is used to discretize.

The second order space derivatives are approximated using the centre scheme.
For a certain time-step j,

$$(\frac{\partial^2 \phi_x}{\partial x^2})_j \approx \frac{1}{\Delta x^2}[(\phi_x)_{j+1} - 2(\phi_x)_j - (\phi_x)_{j-1}]$$

for the whole space domain x,

$$(\frac{\partial^2 \phi_x}{\partial x^2}) = K_x \phi_x$$

and similarly for the space domain derivative $(\frac{\partial^2 B_0^i}{\partial t^2})$,it is approximated at a certain time-step j as,

$$(\frac{\partial^2 \phi_x^i}{\partial x^2})_j \approx \frac{1}{\Delta x^2}[(\phi_x^i)_{j+1} - 2(\phi_x^i)_j - (\phi_x^i)_{j-1}]$$

for the whole space domain x,

$$(\frac{\partial^2 \phi_x^i}{\partial x^2}) = K_x \phi_x^i$$

Similarly, for the whole space domain y,

$$(\frac{\partial^2 \phi_y}{\partial y^2}) = K_y \phi_y \quad and \quad (\frac{\partial^2 \phi_y^i}{\partial y^2}) = K_y \phi_y^i$$

The K_x and K_y are the finite difference matrices.The I_x and I_y are the identity matrix for space and time.

2.4 PGD Formulation

Using the Finite Difference in the equations (4) and (6),we get the PGD formulation of the problem.
For the Space x Problem:

$$[\alpha_1 I_x + \beta_1 K_x]\phi_x = \gamma_1 f_x - \sum_{i=1}^{d} \alpha_i [\alpha_1^i I_x + \beta_1^i K_x]\phi_x^i$$

where,

- $\alpha_1 = \int_y K_y \phi_y \phi_y dy$ *and* $\beta_1 = \int_y \phi_y \phi_y dy$
- $\gamma_1 = \int_y f(x) \phi_y dy$
- $\alpha_1^i = \int_y K_y \phi_y^i \phi_y dy$ *and* $\beta_1^i = \int_y \phi_y^i \phi_y dy$

For the Space y problem:

$$[\alpha_2 I_y + \beta_2 K_y]\phi_y = \gamma_2 f_y - \sum_{i=1}^{d} \alpha_i [\alpha_2^i I_y + \beta_2^i K_y]\phi_y^i$$

where,

- $\alpha_2 = \int_x K_x \phi_x \phi_x dx$ *and* $\beta_2 = \int_x \phi_x \phi_x dx$
- $\gamma_2 = \int_x f(y) \phi_x dx$
- $\alpha_2^i = \int_x K_x \phi_x^i \phi_x dx$ *and* $\beta_2^i = \int_x \phi_x^i \phi_x dx$

2.5 Exact Solution

The exact solution is computed for the problem.Then the problem becomes,

$$\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial y^2} = -1 \quad in \quad (x,y) \in]0,2[x]0,1[$$

$$u = 0 \quad on \quad boundaries$$

This problem is simulated and the exact solution is obtained.

2.6 Comparison between PGD and Exact Solution

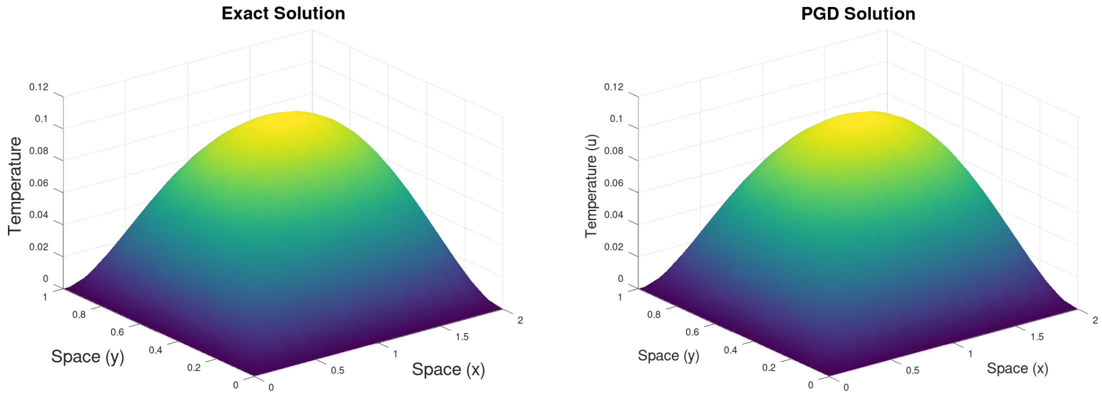
The PGD solution and the exact solution were simulated with space x domain discretized as $\Delta x = 0.05$ and space y domain discretized as $\Delta y = 0.05$.The source term is set to 1 $i.e, f(x,y) = 1$.The PGD solution was computed using the first 5 terms.The fixed-point tolerance was set to $\varepsilon = 10^{-7}$ and the tolerance for the convergence was set to $\tilde{\varepsilon} = 10^{-7}$. The below figure shows the Surface plot of the exact solution.

2.7 Error Analysis

The Error was analysed with the respect to the number of terms and the number of nodes in the mesh which is equispaced and is compared with the exact solution.The error is computed as,

$$e_N^M = \frac{||u_N^{ex} - u_N^M||_2}{||u_N^{ex}||_2}$$

The following plot was obtained for number of nodes,$N = \{11,21,41,61\}$ and for the first 10 terms.It can be observed that the relative error stabilises as the number of modes is increased.And,as we increase the number of nodes,the relative error also decreases.



(a) Surface Plot of the Exact Solution (b) Surface Plot of the PGD Solution

Figure 1: Comparison of the Surface Plots

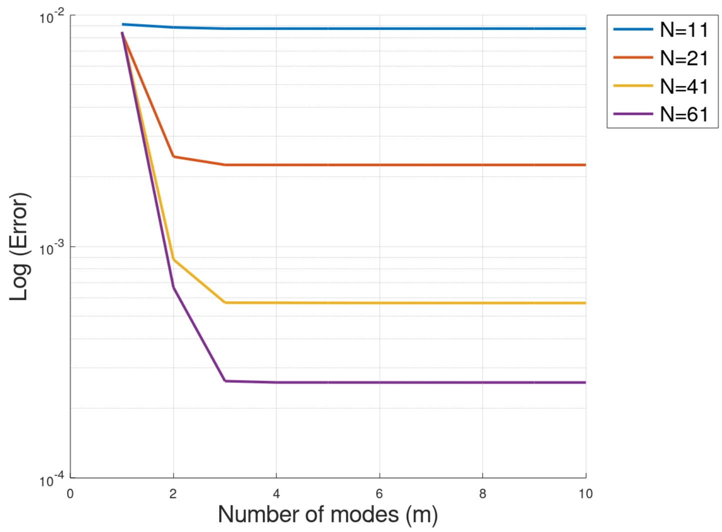


Figure 2: Evolution of Error for different modes and nodes

2.8 PGD formulation for a general source term

The PGD was formulated for a general source term .The below surface plot was obtained for the source term,$f(x,y) = (x^2 - y^2)$.

3 Part 2 : PGD Code Generalisation

A general function is created allowing to apply the PGD method to any problem with appropriate tensor structure,i.e,

$$Au = b$$

$$A = \sum_{k=1}^{N_A} A_1^k \otimes ..... \otimes A_D^k \qquad b = \sum_{k=1}^{N_b} b_1^k \otimes ...... \otimes b_D^k$$

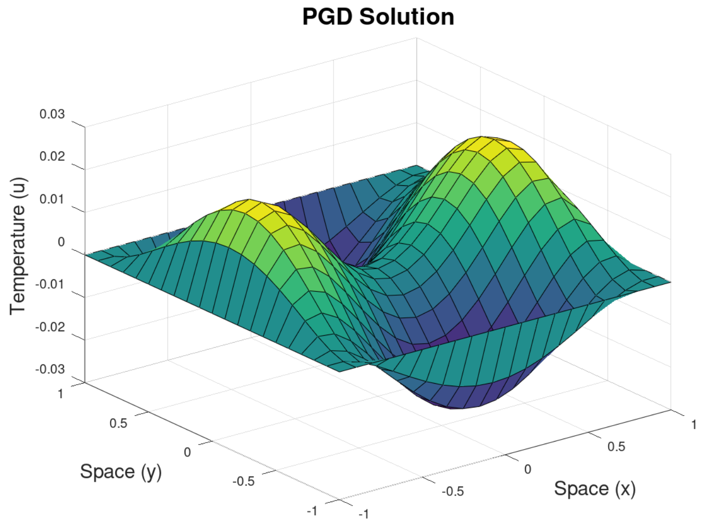


Figure 3: Surface plot of PGD solution for source term $f(x,y) = (x^2 - y^2)$

where,D is the number of dimensions , N_A is the number of terms in LHS and N_b is the number of terms in RHS. We get the solution in the following tensor structure,

$$u = \sum_{k=1}^{N_m} u_1^k \otimes .... \otimes u_D^k$$

The 2D-Poisson equation is generalised as follows,

$D = 2$,$N_A = 2$ and $N_b = 1$

AA= $\begin{bmatrix} K_x & M_x \\ M_y & K_y \end{bmatrix}$ BB= $\begin{bmatrix} b_x \\ b_y \end{bmatrix}$

3.1 Alternating Directions Stagnation Criterion

The `easy_PGD` was modified in order to implement the following Alternating directions stagnation criterion,

$$||(u_1 \otimes ...... \otimes u_D)_{current} - (u_1 \otimes ...... \otimes u_D)_{previous}||_2 < \varepsilon_1$$

The following code was implemented in the fixed-iteration loop.The graph shows the evolution of the number of iterations for stagnation for different values of ε_1.It can be observed that for mode 2,the number of iterations taken for stagnation is less than mode 1.

```
difference=((RS{1}*RS{2}')-(RS_prev{1}*RS_prev{2}'));

if(norm(difference)<epsilon)
n=iter;
break;
end
```

Figure 4: Code for Alternating Directions Stagnation Criterion

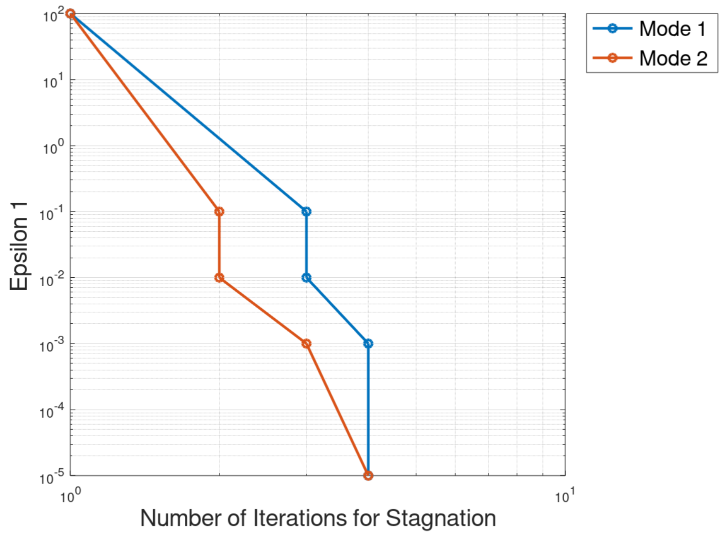


Figure 5: Convergence of First two modes

3.2 Global Convergence Criterion

The `easy_PGD` was modified in order to implement the following global convergence criterion which is based on the residual norm,

$$\frac{||b - Au||_2}{||b||_2} < \varepsilon_2$$

The following code was implemented in the PGD loop, The ε_1 was set 10^{-4} and the following

```
for m=1:mode
 AA1=AA{1,1}*FF{1}(:,m);
 AA2=AA{2,1}*FF{2}(:,m);
 AA3=AA{1,2}*FF{1}(:,m);
 AA4=AA{2,2}*FF{2}(:,m);
 AAF=AAF+(AA1*AA2')+(AA3*AA4');
endfor

 BB1=BB{1}*BB{2}';

 Denom=norm(BB1(2:end-1,2:end-1));
 Num=norm(BB1(2:end-1,2:end-1)-AAF(2:end-1,2:end-1));
 relative_err=Num/Denom;

 if(relative_err<Global_Eps)
 Glob_conv_mode=mode;
 break;
 end
```

Figure 6: Code for Global Convergence Criterion

graph shows the convergence up to the fifth mode.It can be observed that as the number of modes is increased the relative error decreases.

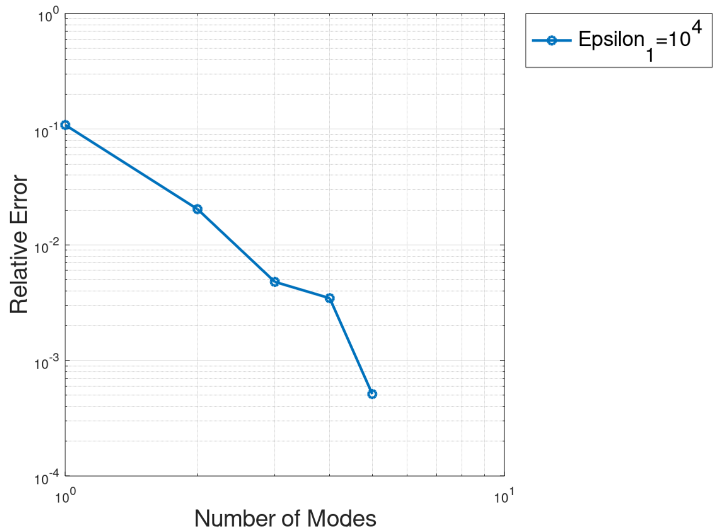


Figure 7: Global Convergence upto fifth mode

4 Conclusion

In conclusion, the PGD was formulated for the Space-Space domain problem. The results were in agreement with the exact solution for a few enrichment terms. The relative error decreases with the increase in the number of modes and nodes but it stabilizes after a certain number of enrichments. Also, the PGD was formulated for a general source term.

The PGD for the same problem was formulated with a generalized function. The results obtained from this implementation were in agreement with the exact solution. By convergence study, the relative error is observed to decrease as the number of modes is increased.