





# A PGD-based Method for Robot Global Path Planning: A Primer

N. Montés<sup>1</sup>, F. Chinesta<sup>2</sup>, A. Falcó<sup>1</sup>, M. C. Mora<sup>3</sup>, L. Hilario<sup>1</sup> and J. L. Duval<sup>4</sup>

<sup>1</sup>*Department of Physics, Mathematics and Technological Sciences, University CEU Cardenal Herrera, 46115, Alfara del Patriarca, Spain*

<sup>2</sup>*PIMM, ENSAM ParisTech ESI GROUP Chair on Advanced Modeling and Simulation of Manufacturing Processes, Paris, France*

<sup>3</sup>*Department of Mechanical Engineering and Construction, Universitat Jaume I, Castellón, Spain*

<sup>4</sup>*ESI Group, RUNGIS CEDEX, France*


**Keywords:** Model Order Reduction Techniques, PGD, Path Planning, Potential Field Methods, Laplace Equation.


**Abstract:** The present paper shows, for the first time, the technique known as PGD-Vademecum as a global path planner for mobile robots. The main idea of this method is to obtain a Vademecum containing all the possible paths from any start and goal positions derived from a harmonic potential field in a predefined map. The PGD is a numerical technique with three main advantages. The first one is the ability to bring together all the possible Poisson equation solutions for all start and goal combinations in a map, guaranteeing that the resulting potential field does not have deadlocks. The second one is that the PGD-Vademecum is expressed as a sum of uncoupled multiplied terms: the geometric map and the start and goal configurations. Therefore, the harmonic potential field for any start and goal positions can be reconstructed extremely fast, in a nearly negligible computational time, allowing real-time path planning. The third one is that only a few uncoupled parameters are required to reconstruct the potential field with a low discretization error. Simulation results are shown to validate the abilities of this technique.


## 1 INTRODUCTION


An essential in robotics is to guide the robot safely from a start to a goal position among a set of obstacles. For this purpose, a collision-free path must be generated, which implies a computationally hard geometric path planning unfeasible in real-time (RT) applications (Reif, 1979). This problem is known in the literature as motion planning or the piano mover's problem and its complexity has motivated a lot of research works in the field of robot path planning. Some works have studied subproblems of the general approach (Kavraki and LaValle, 2008). Other researchers have considered alternative planning paradigms under simplified but realistic assumptions such as, for instance, sampling-based planners, grid-based searches, interval-based searches, geometric algorithms, etc (Kavraki and LaValle, 2008).

One of the most used algorithm is the Artificial Potential Field method (APF), (Khatib, 1986). This technique defines an artificial potential field in the configuration space (C-space) that generates a path from a start to a goal position. This method is very fast for RT applications. However, the robot could get stuck in a local minimum of the potential function. This problem can be solved using harmonic functions in the generation of the potential field (Canny, 1998), which satisfy the Laplace equation in the C-space and completely eliminate local minima as they satisfy the Min-Max principle (Rimon and Koditschek, 1992). These functions were initially proposed in (Zhachmanoglou and Thoe, 1986) and used for robot path planning in (Connolly et al., 1990; Kim and Khosla, 1992; Akishita et al., 1993; Guldner et al., 1997; Waydo and Murray, 2003; Rosell and Iniguez, 2002; Saudi and Sulaiman, 2012). The main problem of this technique, addressed in (Waydo and Murray, 2003), is that the solution must be numerically computed in a discrete mesh and, therefore, the computational cost increases exponentially with the mesh resolution. In (Gingras et al., 2010), one of the last

<sup>a</sup>  <https://orcid.org/0000-0002-0661-3479>

<sup>b</sup>  <https://orcid.org/0000-0001-6225-0935>

<sup>c</sup>  <https://orcid.org/0000-0003-0627-6764>

<sup>d</sup>  <https://orcid.org/0000-0003-0729-6628>

studies, an scanned environment composed by 1500 triangles had a computational cost of 19.2 Sec in a laptop Dell latitude with an Intel Core 2 Processor, 2GB RAM, inadmissible for RT applications.

More recently, in the field of advanced computational methods, a novel alternative called Proper Generalized Decomposition (PGD) has appeared to compute the Laplace equation. It is a completely new approach for solving classic high-complexity problems (Chinesta et al., 2013; Chinesta et al., 2014). Many difficult problems can be efficiently converted into a multidimensional framework, which opens up novel possibilities to solve old and new problems with approaches not foreseen until now. In a PGD framework, the model can be solved only once with the aim of obtaining a general solution that includes all the solutions for every value of the parameters, that is, a Computational Vademecum.

The goal of the present paper is to present the computation of a PGD-based computational Vademecum (PGD-Vademecum for short) for using the potential flow theory based on harmonic functions in two RT path planning applications. This paper is organized as follows. Section 2 introduces the potential flow theory and obtains a harmonic function derived from the Laplace equation. In Section 3, a specific PGD-Vademecum for robot path planning is calculated and a numerical example is used to demonstrate the benefits of the method. In Section 4, some simulation examples are provided. Finally, in Section 5 conclusions and future works are presented.

## 2 BACKGROUND

### 2.1 Path Planning based on the Potential Flow Theory

During the last years, many path planning techniques have been based on the potential flow theory (Connolly et al., 1990; Kim and Khosla, 1992; Akishita et al., 1993; Guldner et al., 1997; Waydo and Murray, 2003; Rosell and Iniguez, 2002; Saudi and Sulaiman, 2012; Gingras et al., 2010), focused mainly in the resolution of the Laplace equation. First of all, let us outline the mathematical model describing the flow of an inviscid incompressible fluid. Assuming a steady state irrotational flow in the Eulerian framework, the velocity  $\mathbf{v}$  obeys the relation

$$\nabla \times \mathbf{v} = \mathbf{0}, \quad (1)$$

and hence the velocity is the gradient of a scalar potential function, i.e.  $\mathbf{v} = \nabla u$ . Then the potential  $u$  ap-

pears as a solution of the Laplace equation:

$$\Delta u = 0. \quad (2)$$

By using a 2.5D mould filling model similar to (Doménech et al., 2016) it is possible to introduce a localized fluid source (respectively, sink) modelled by a Dirac term  $\delta_S$  (respectively,  $-\delta_T$ ) added to the right hand side of (2). To this end we assume a unit amount of fluid injected at point  $S$  during a unit of time and the same unit withdrawn at point  $T$ , the velocity of the fluid is now the solution of the Poisson equation, that includes the source term  $f = \delta_S - \delta_T$ :

$$-\Delta u = \delta_S - \delta_T. \quad (3)$$

Equation (3) must be complemented by appropriate boundary conditions. In these sense, the fluid cannot flow through the boundaries, a condition expressed by  $\mathbf{v} \cdot \mathbf{n}$  ( $\mathbf{n}$  being a vector normal to the boundary  $\Gamma$ ). On  $\Gamma$  the velocity must verify:

$$-\nabla u \cdot \mathbf{n} = 0, \quad (4)$$

which is the usual the Neumann boundary condition expressed by

$$\left. \frac{\partial u}{\partial n} \right|_{\Gamma} = 0. \quad (5)$$

The resolution of the Poisson equation under these conditions produces a potential field from the Starting point  $S$  (source) to the Target point  $T$  (sink), without deadlocks (Gingras et al., 2010).

### 2.2 Resolution of the Poisson Equation using PGD

Consider the two dimensional Poisson equation

$$\Delta u(x, y) = f(x, y) \quad (6)$$

over a two-dimensional rectangular domain  $\Omega_{\underline{X}} = \Omega_X \times \Omega_Y \subset \mathbb{R}^2$  with boundary condition  $\left. \frac{\partial u}{\partial n} \right|_{\Gamma} = q$ . For all suitable test functions  $u^*$ , the weighted residual forms reads

$$\int_{\Omega_{\underline{X}}} u^* \cdot (\Delta u - f) d\Omega_{\underline{X}} \quad (7)$$

The classical way of accounting for Neuman conditions is to integrate by parts the weighted residual form and implement the flux condition as a so-called natural boundary condition:

$$\begin{aligned} \int_{\Omega_{\underline{X}}} \nabla u^* \cdot \nabla u d\Omega_{\underline{X}} &= \int_{\Omega_{\underline{X}}} u^* \cdot f d\Omega_{\underline{X}} \\ &- \int_{\Omega_{\underline{X}}} u^*(x, y = \Gamma) \cdot q d\Omega_{\underline{X}} \end{aligned} \quad (8)$$

Our goal is to obtain a PGD approximate solution to (8) in the separated form

$$u(\Omega_{\underline{X}}) = \sum_{i=1}^N X_i(x) \cdot Y_i(y) \quad (9)$$

We shall do so by computing each term of the expansion one at a time, thus enriching the PGD approximation until a suitable convergence criterion is satisfied. At each enrichment step  $n$ , ( $n \geq 1$ ), we have already computed the  $n - 1$  first terms of the PGD approximation (9):

$$u^{n-1}(\Omega_{\underline{X}}) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) \quad (10)$$

We now wish to compute the next term  $X_n(x) \cdot Y_n(y)$  to obtain the enriched PGD solution

$$\begin{aligned} u^n(x, y) &= u^{n-1}(x, y) + X_n(x) \cdot Y_n(y) = \\ &= \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) + X_n(x) \cdot Y_n(y) \end{aligned} \quad (11)$$

Both functions  $X_n(x)$  and  $Y_n(y)$  are unknown at the current enrichment step  $n$  and an alternative iterative scheme is applied. We use index  $p$  to denote a particular iteration.

$$u^{n,p}(x, y) = u^{n-1}(x, y) + X_n^p(x) \cdot Y_n^p(y) \quad (12)$$

The simplest one is an alternating direction strategy that computes  $X_n^p(x)$  from  $Y_n^{p-1}(y)$  and then  $Y_n^p(y)$  from  $X_n^p(x)$ . An arbitrary initial guess  $Y_n^0(y)$  is specified to start the iterative process. The non-linear iterations proceed until reaching a fixed point within a user-specified tolerance, see (Chinesta et al., 2013; Chinesta et al., 2014). The convergence of the above procedure to the weak solution of (6) is proved in (Falcó and Nouy., 2012).

### 3 PATH PLANNING BASED ON THE PGD-VADEMECUM

The preceding section has presented a simple example application of the resolution of the Poisson equation using PGD in a case where the 2D space is decomposed in  $X$  and  $Y$ . (Chinesta et al., 2013; Chinesta et al., 2014) demonstrate that parameters in a model can be set as additional coordinates when using the PGD approach. In the following sections, a path planning example is presented where these additional coordinates are all the possible combinations of the start and target positions and can be included in the source term of the Poisson equation (6).

### 3.1 Definition of the Source Term

First of all, it is necessary to assume that a constant source term  $f$  in equation (6) is actually a non-uniform source term  $f(\Omega_{\underline{X}}, \Omega_{\underline{S}}, \Omega_{\underline{T}})$ , where  $\Omega_{\underline{X}} = \Omega_{\underline{x}} \times \Omega_{\underline{y}}$ ,  $\Omega_{\underline{S}} = \Omega_r \times \Omega_s$  and  $\Omega_{\underline{T}} = \Omega_r \times \Omega_t$ . In this definition, the start and target points  $\underline{S}$  and  $\underline{T}$  are defined by means of Gaussian models with mean and variance:  $\underline{S} = (s, r)$  and  $\underline{T} = (t, r)$ , respectively. In these models,  $s$  and  $t$  are the mean values located in specific points  $\underline{X} = (x, y)$  in each separated space  $\Omega_{\underline{S}}, \Omega_{\underline{T}}$  and  $r$  is the variance. Gaussian models are used instead of Delta Dirac models because they provide much better results in a PGD-Vademecum than Delta Dirac model, as explained in (Chinesta et al., 2013). In order to define the source term, the next two matrices must be constructed first:

$$f(\underline{X}, \underline{S}) = \begin{bmatrix} f(x_1, s_1) & \dots & f(x_1, s_N) \\ \vdots & \ddots & \vdots \\ f(x_N, s_1) & \dots & f(x_N, s_N) \end{bmatrix} \quad (13)$$

$$f(\underline{X}, \underline{T}) = \begin{bmatrix} f(x_1, t_1) & \dots & f(x_1, t_N) \\ \vdots & \ddots & \vdots \\ f(x_N, t_1) & \dots & f(x_N, t_N) \end{bmatrix}$$

Applying the Single Value Decomposition (SVD) method to these matrices, the result is the decomposition of the source term in the form:

$$\begin{aligned} f(\underline{X}, \underline{S}) &= \sum_{j=1}^F \alpha_j^S \cdot F_j^S(\underline{X}) \cdot G_j^S(\underline{S}) \\ g(\underline{X}, \underline{T}) &= \sum_{j=1}^F \alpha_j^T \cdot F_j^T(\underline{X}) \cdot G_j^T(\underline{T}) \end{aligned} \quad (14)$$

Thus, the Poisson equation to be solved results in the form:

$$\Delta u(x, y) = f(\underline{X}, \underline{S}) + g(\underline{X}, \underline{T}) \quad (15)$$

### 3.2 Computation of the PGD-Vademecum

For all suitable test functions  $u^*$ , the weighted residual forms reads

$$\int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} u^* \cdot (\Delta u - f) d\Omega_{\underline{X}, \underline{S}, \underline{T}} = 0 \quad (16)$$

where  $f$  is in the form obtained in (15):

$$f = f(\underline{X}, \underline{S}) + g(\underline{X}, \underline{T}) \quad (17)$$

Now, equation (8) reads

$$\begin{aligned} \int_{\Omega_{\underline{X},\underline{S},\underline{T}}} \nabla u^* \cdot \nabla u d\Omega_{\underline{X},\underline{S},\underline{T}} &= \int_{\Omega_{\underline{X},\underline{S},\underline{T}}} u^* \cdot f d\Omega_{\underline{X},\underline{S},\underline{T}} - \\ &- \int_{\Omega_{\underline{X},\underline{S},\underline{T}}} u^*(x,y=\Gamma) \cdot q d\Omega_{\underline{X},\underline{S},\underline{T}} \end{aligned} \quad (18)$$

and the PGD-Vademecum is now

$$u(\underline{X},\underline{S},\underline{T}) = \sum_{i=1}^N R_i(\underline{X}) \cdot W_i(\underline{S}) \cdot K_i(\underline{T}) \quad (19)$$

From now on, it will be assumed that the common variance  $r$  takes a fixed value and the construction of the Vademecum will be obtained considering the solution of (15) for that value  $r$  and all the values of  $(\underline{X} = (x,y); \underline{S} = (s_1,s_2); \underline{T} = (t_1,t_2)) \in \Omega_{\underline{X}} \times \Omega_{\underline{S}} \times \Omega_{\underline{T}}$ .

Then, the PGD-Vademecum solution is constructed considering that

$$u^{n-1}(\underline{X},\underline{S},\underline{T}) = \sum_{i=1}^{n-1} R_i(\underline{X}) \cdot W_i(\underline{S}) \cdot K_i(\underline{T}) \quad (20)$$

where the enrichment step is given by

$$u^n = u^{n-1} + R(\underline{X}) \cdot W(\underline{S}) \cdot K(\underline{T}). \quad (21)$$

The key point is to find a rank-one function in the form

$$\begin{aligned} R(\underline{X}) \cdot W(\underline{S}) \cdot K(\underline{T}) = \\ R_1(x) \cdot R_2(y) \cdot W_1(s_1) \cdot W_2(s_2) \cdot K_1(t_1) \cdot K_2(t_2) \end{aligned} \quad (22)$$

which satisfies

$$\int_{\Omega_{\underline{X}} \times \Omega_{\underline{S}} \times \Omega_{\underline{T}}} u^* \cdot (\Delta u^n - f) d\Omega_{\underline{X},\underline{S},\underline{T}} = 0$$

for all  $u^*$  in the linear space of functions

$$\begin{aligned} R(\underline{X}) \cdot W(\underline{S}) \cdot K^*(\underline{T}) + R(\underline{X}) \cdot W^*(\underline{S}) \cdot K(\underline{T}) \\ + R^*(\underline{X}) \cdot W(\underline{S}) \cdot K(\underline{T}), \end{aligned}$$

where  $K^*(\underline{T})$  is orthogonal to  $K(\underline{T})$ ,  $W^*(\underline{S})$  is orthogonal to  $W(\underline{S})$  and  $R^*(\underline{X})$  is orthogonal to  $R(\underline{X})$ . In the appendix an alternating direction algorithm is provided to construct the separated representation. A prior step implies the discretization of (20) by means of the Finite-Element Method (FEM) with

$$N_x \cdot N_y + N_{s_1} \cdot N_{s_2} + N_{t_1} \cdot N_{t_2}$$

degrees of freedom and assuming that

$$N = N_x = N_y = N_{s_1} = N_{s_2} = N_{t_1} = N_{t_2}.$$

Therefore, at each iteration step, the algorithm computes the  $3N^2$  terms of the rank-one update (22) and, after  $n$  iterations, the PGD approximation of the solution  $u$  is given by (21).

### 3.3 A Simple Numerical Example

With the aim of testing the advantages of the PGD framework, a simple example is developed in this subsection. Let us consider a domain  $\Omega_{\underline{X}}$  of  $5m \times 5m$  square. Let us select a discretization of the domain using  $N_x = N_y = 50$  nodes on each side, that is, 2500 degrees of freedom and the variance  $r$  set to 1.2. In Figure 1 an example for  $\underline{S} = (1,4)$  and  $\underline{T} = (4,1)$  is shown. The left column displays the source term and the right column shows the resulting PGD reconstruction for  $n = 200$  terms.

The computational cost of the PGD reconstruction is 0.0101s in a Mac with an Intel Core 2 Duo Processor (3.06 GHz) and 4 GB RAM. It is worth noting the difference between this negligible value and the cost of a FEM approximation to solve a standard linear system, where the computational cost increases to 4.7s.

### 3.4 Residual Error of the PGD Approximation

The error of the PGD approximation versus the number of terms used can be measured by means of different techniques. A very appropriate error estimator in this case is the  $L^2(\Omega_{\underline{X}} \times \Omega_{\underline{S}} \times \Omega_{\underline{T}})$ -residual  $R(n)$ , that can be obtained inserting the PGD-Vademecum approximation in the Poisson equation, resulting in

$$R(n) = \int_{\Omega_{\underline{X}} \times \Omega_{\underline{S}} \times \Omega_{\underline{T}}} (\Delta u^n - f) \cdot (\Delta u^n - f) d\Omega_{\underline{X},\underline{S},\underline{T}} \quad (23)$$

Figure 2 shows one of the most important features of the PGD: the relevant information is stored in the first terms of the approximation.

### 3.5 Local Minima in the Approximated Solution

The use of harmonic functions solve the problem of local minima still present in APF-based techniques. The solution of an harmonic function based on flow dynamics and described by the Poisson equation is approximated in the present paper by means of the PGD-Vademecum technique. Therefore, this approximation could produce local minima in the solution if the variance value  $r$  is not adjusted properly. For the numerical example developed in the previous subsections, Figure 4 shows the maximum number of local minima of the PDG solution for all the possible Start and Target combinations ( $50^4$ ) and for different variance values ( $r = 0.1, 0.3, 0.5$  and  $0.7$ ). A specific node  $(N_1, N_2)$  is defined as a local minima if the

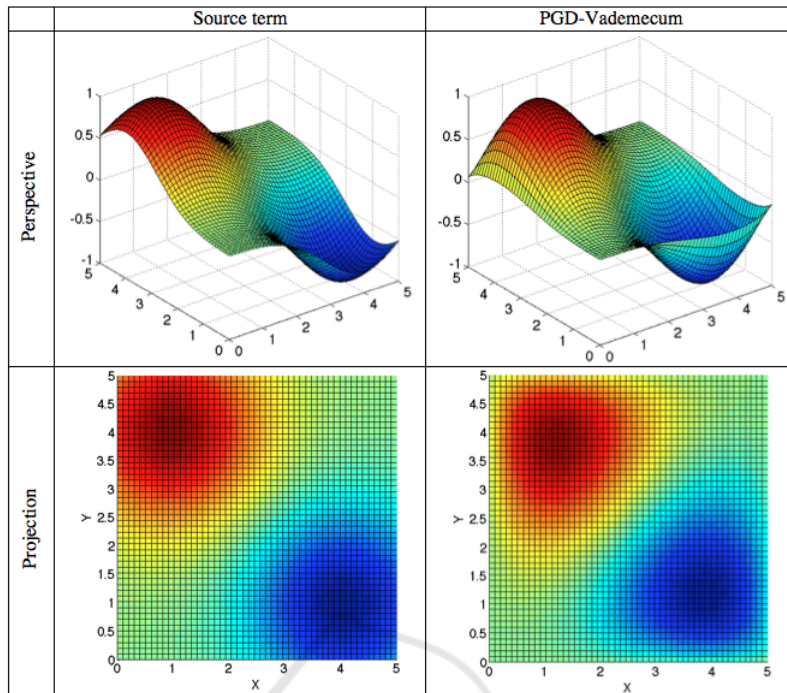


Figure 1: PGD reconstruction VS source term for  $\underline{S} = (1, 4), \underline{T} = (4, 1)$ .

eight surrounding nodes have an  $u$  value greater than its  $u$  value. As displayed in Figure 4, a free-of-local-minima PGD-Vademecum approximation can be obtained if the selected variance value is higher that 0.7.

### 3.6 Computation of Streamlines

As explained in Section 1, the use of harmonic functions solve the problem of local minima present in APF-based techniques. Harmonic functions are based on flow dynamics, described by the Poisson equation,

where the potential field is free of local minima and derives in a set of streamlines (Connolly et al., 1990; Kim and Khosla, 1992; Akishita et al., 1993; Guldner et al., 1997; Waydo and Murray, 2003; Rosell and Iniguez, 2002; Saudi and Sulaiman, 2012; Gingras et al., 2010). These flow lines are independent of time and describe the direction of a massless fluid element that travels from an initial to a final position, following a velocity field that can be obtained from the gradient of the potential field, as described in equations (25).

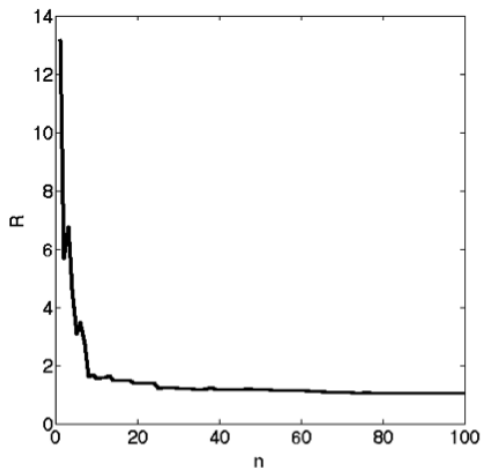


Figure 2: Residual error versus number of PGD terms.

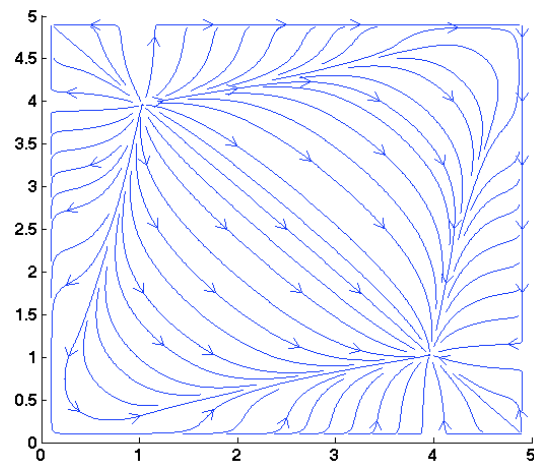


Figure 3: Streamlines.

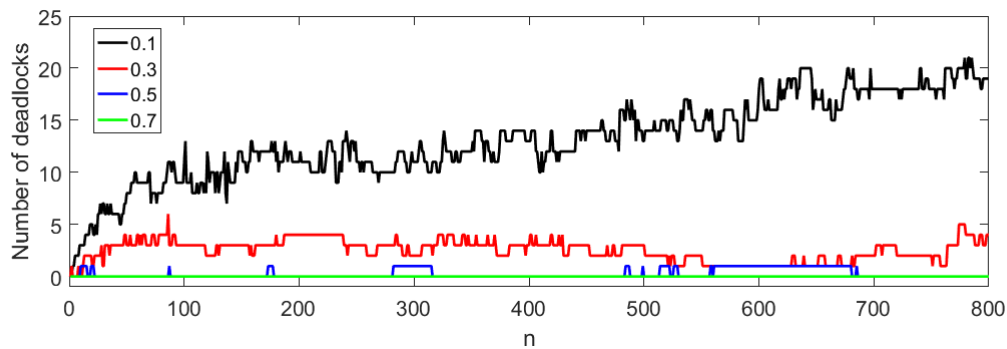


Figure 4: Local minima in the PGD approximation solution.

$$v_x = \frac{du}{dx}, v_y = -\frac{du}{dy} \quad (24)$$

Then, from the velocity field, any interpolation technique can be used to compute the streamlines as, for instance, linear, cubic, splines, etc. Fig 3 displays the streamlines that result from using linear interpolation for the PGD reconstruction showed in Figure 1.

#### 4 REAL-TIME APPLICATION (SHORTEST PATH)

Some RT simulations in Matlab have been performed with the aim of testing the advantages that the PGD-Vademecum approach offers. An omnidirectional mobile robot has been modelled that navigates in a 5x5m square environment guided by a harmonic potential field approximated by the PGD with  $50 \times 50$  nodes,  $r = 0.7$  and  $n = 200$ . For a realistic implementation, only as small Region Of Interest (ROI) is reconstructed in each algorithm execution. The ROI is composed of the surrounding nodes of the current robot position and its size depends on the maximum robot velocity. In the present example, for particular Start and Target configurations, the robot selects the shortest streamline, which is a straight line heading to the Target. Figure 5 depicts different trajectories followed by the robot beginning at the starting point  $\underline{S} = (1, 4)$  to subsequent target points  $\underline{T} = (4, 1), (3, 4), (2, 1), (4, 3)$ .

A second example is also presented in the same 5x5m square environment, potential function and parameters. This time, an static obstacle is located in the environment. For a specific Start and Target configuration, it can be seen that the robot selects the shortest streamline heading to the Target and avoiding the obstacle. Figure 6 depicts the simulation results.

#### 5 CONCLUSIONS AND FUTURE WORKS

The present paper develops, for the first time, the application of the numerical technique known as PGD-Vademecum in the global path planning problem for mobile robotics. This method generates a sort of Computational Vademecum containing all the possible robot paths for any Start and Target combinations in a predefined map. This PGD-Vademecum must be computed off-line and stored in the robot memory in order to be reconstructed on-line for any particular combination of start and target configurations. It is very fast for RT applications because its formulation results in a simple sum of products. In addition, during navigation tasks, only the surrounding nodes of the robot position need to be reconstructed in every algorithm execution. As a result, the computational costs in RT are extremely low, almost negligible. The robot paths obtained are based on the Laplace/Poisson equation and, thus, are local-minima free when the variance is properly adjusted in the PGD approximated solution. This property makes the proposed approach a promising method to solve the piano mover's problem. The only drawback noticed is the generation of a small offset in the start and target positions due to the definition of the source term, as start and target positions have a coupling effect. The solution of this problem is our immediate future work.

#### REFERENCES

- Akishita, S., Hisanobu, T., and Kawamura, S. (1993). Fast path planning available for moving obstacle avoidance by use of laplace potential. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1(1):673–678.
- Canny, J. (1998). *The complexity of robot motion planning*. MIT press, Cambridge, Massachusetts.
- Chinesta, F., Keunings, R., and Leygue, A. (2014). *The*

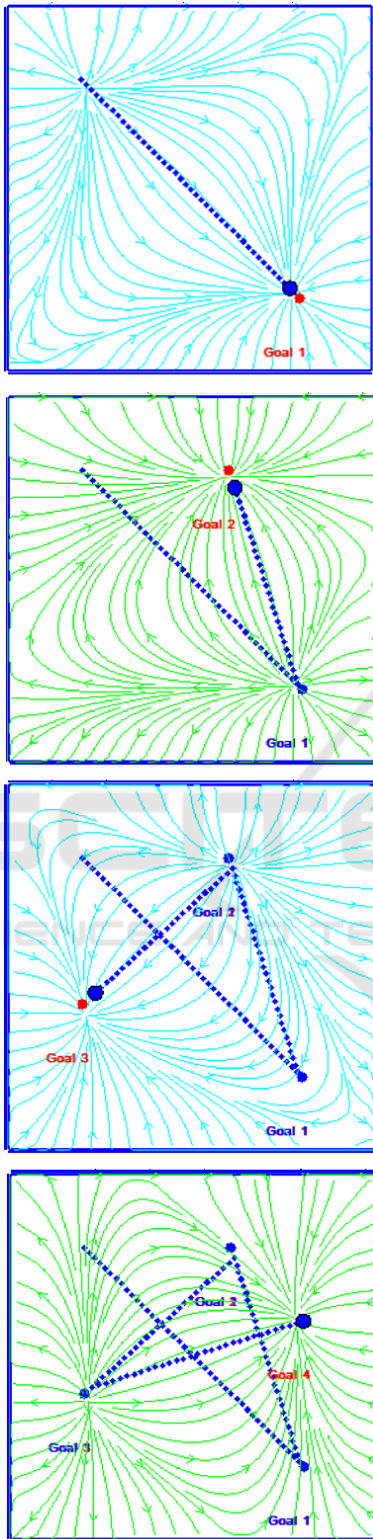


Figure 5: Simulation results: the robot visits the following targets  $\underline{T} = (4, 1), (3, 4), (2, 1), (4, 3)$ .

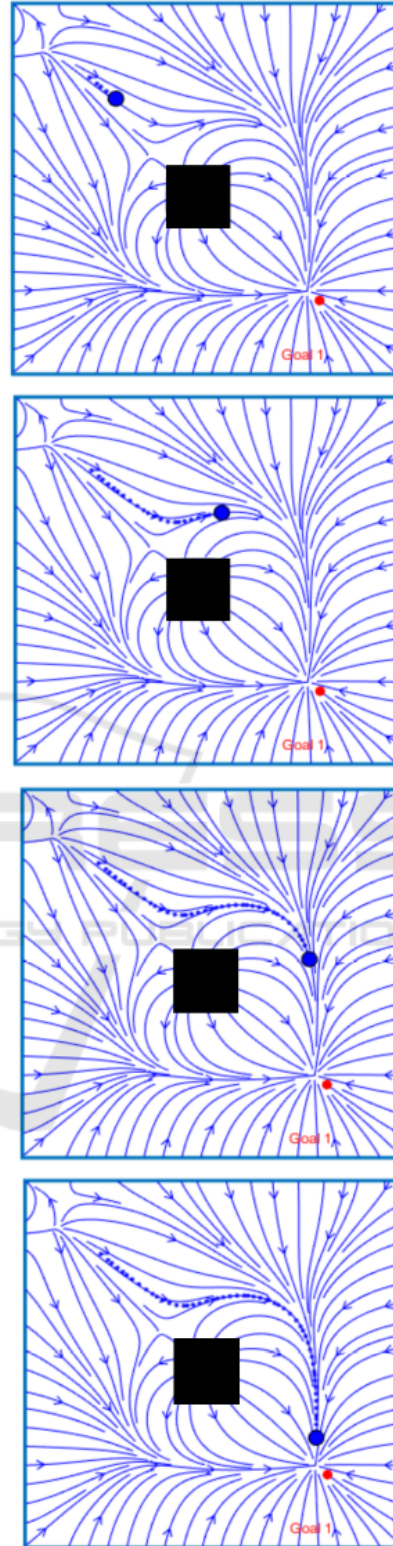


Figure 6: Simulation results: the robot navigates to the target avoiding an obstacle in the environment.

*Proper Generalized Decomposition for Advanced Numerical Simulations*. Springer Briefs in Applied Science and Technology, Berlin, 2nd edition.

- Chinesta, F., Leygue, A., Bordeu, F., Aguado, J., Cueto, E., Gonzalez, D., Alfaro, I., Ammar, A., and Huerta, A. (2013). Pgd-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1):31–59.
- Connolly, C., Burns, J., and Weiss, R. (1990). Path planning using laplace's equation. *IEEE Int. Conf. on Robotics and Automation*, 1(1):2102–2106.
- Doménech, L., Falcó, A., García, V., and Sánchez, F. (2016). Towards a 2.5d geometric model in mold filling simulation. *Journal of Computational and Applied Mathematics*, 29(1):183–196.
- Falcó, A. and Nouy, A. (2012). Proper generalized decomposition for nonlinear convex problems in tensor banach spaces. *Numerische Mathematik*, 121(3):503–530.
- Gingras, D., Dupuis, E., Payre, G., and Lafontaine, J. (2010). Path planning based on fluid mechanics for mobile robots used unstructured terrain models. *IEEE International Conference on Robotics and Automation*, 1(1):1978–1984.
- Guldner, J., Utkin, V. I., and Hashimoto, H. (1997). Robot obstacle avoidance in n-dimensional space using planar harmonic artificial fields. *Journal of Dynamic Systems, Measurement and Control*, 119(2):119–160.
- Kavraki, L. and LaValle, S. (2008). *Chapter 5. Motion Planning, Handbook of Robotics*. Siciliano, Khatib (Eds), Springer, Berlin, 2nd edition.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotic Research*, 1(1):90–98.
- Kim, J. and Khosla, P. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349.
- Reif, J. (1979). Complexity of the mover's problem and generalizations. *IEEE Symp. Found. Comput. Sci.*, pages 421–427.
- Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.
- Rosell, J. and Iniguez, P. (2002). A hierarchical and dynamic method to compute harmonic functions for constrained motion planning. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1(1):2335–2340.
- Saudi, A. and Sulaiman, J. (2012). Path planning for mobile robots using 4egsor via nine-point laplacian (4egsor9l) iterative method. *International Journal of computer applications.*, 53(16):38–42.
- Waydo, S. and Murray, S. (2003). Vehicle motion planning using stream functions. *IEEE Int. Conf. on Robotics and Automation*, 1(1):2484–2491.
- Zachmanoglou, E. and Thoe, D. (1986). *Introduction to partial differential equations with applications*. Baltimore Williams & Wilkins, Baltimore.

## APPENDIX: Alternating Directions Separated Representation Constructor

### Computing $R(\underline{X})$ from $W(\underline{S})$ , $K(\underline{T})$

$$\begin{aligned} & \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \frac{dR^*}{dR^*} \frac{dR}{dR} W^2 K^2 + R^* R \left( \frac{dW}{dW} \right)^2 K^2 + R^* R W^2. \\ & \cdot \left( \frac{dK}{dK} \right)^2 d\Omega_{\underline{X}, \underline{S}, \underline{T}} = - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \sum_{i=1}^{n-1} \frac{dR^*}{dR^*} \frac{dR_i}{dR_i} W W_i K K_i + \\ & + R^* R_i \frac{dW}{dW} \frac{dW_i}{dW_i} K K_i + R^* R_i W W_i \frac{dK}{dK} \frac{dK_i}{dK_i} d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \\ & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R^* W K (f(\underline{X}, \underline{S}) + g(\underline{X}, \underline{T})) d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \\ & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R^* (\Gamma) W K q d\Omega_{\underline{X}, \underline{S}, \underline{T}} \end{aligned} \quad (25)$$

Taking into account the following notation for the known terms

$$\begin{aligned} w_1 &= \int_{\Omega_{\underline{S}}} W^2 d\Omega_{\underline{S}} & w_5 &= \int_{\Omega_{\underline{S}}} W G_j^S d\Omega_{\underline{S}} & t_3 &= \int_{\Omega_{\underline{T}}} K K_i d\Omega_{\underline{S}} \\ w_2 &= \int_{\Omega_{\underline{S}}} \left( \frac{dW}{dW} \right)^2 d\Omega_{\underline{S}} & w_6 &= \int_{\Omega_{\underline{S}}} W d\Omega_{\underline{S}} & t_4 &= \int_{\Omega_{\underline{T}}} \frac{dK}{dK} \frac{dK_i}{dK_i} d\Omega_{\underline{T}} \\ w_3 &= \int_{\Omega_{\underline{S}}} W W_i d\Omega_{\underline{S}} & t_1 &= \int_{\Omega_{\underline{T}}} K^2 d\Omega_{\underline{S}} & t_5 &= \int_{\Omega_{\underline{T}}} K G_j^T d\Omega_{\underline{T}} \\ w_4 &= \int_{\Omega_{\underline{S}}} \frac{dW}{dW} \frac{dW_i}{dW_i} d\Omega_{\underline{S}} & t_2 &= \int_{\Omega_{\underline{T}}} \left( \frac{dK}{dK} \right)^2 d\Omega_{\underline{T}} & t_6 &= \int_{\Omega_{\underline{T}}} K d\Omega_{\underline{T}} \end{aligned} \quad (26)$$

Equation 25 is reduced to

$$\begin{aligned} & \int_{\Omega_{\underline{X}}} \frac{dR^*}{dR^*} \frac{dR}{dR} w_1 t_1 + R^* R w_2 t_1 + R^* R w_1 t_2 d\Omega_{\underline{X}} = \\ & - \int_{\Omega_{\underline{X}}} \sum_{i=1}^{n-1} \frac{dR^*}{dR^*} \frac{dR_i}{dR_i} w_3 t_3 + R^* R_i w_4 t_3 + R^* R_i w_3 t_4 d\Omega_{\underline{X}} + \\ & + \int_{\Omega_{\underline{X}}} R^* \left( \sum_{j=1}^F w_5 t_6 \alpha^S F^S(\underline{X}) + t_5 w_6 \alpha^T F^T(\underline{X}) \right) d\Omega_{\underline{X}} - \\ & - w_6 t_6 q \int_{\Omega_{\underline{X}}} R^* (\Gamma) d\Omega_{\underline{X}} \end{aligned} \quad (27)$$



### Computing $W(\underline{S})$ from $R(\underline{X}), K(\underline{T})$

$$\begin{aligned}
 & \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \left( \frac{dR}{dR} \right)^2 W^* W K^2 + R^2 \frac{dW^*}{dW^*} \frac{dW}{dW} K^2 + R^2 W^* W \cdot \\
 & \cdot \left( \frac{dK}{dK} \right)^2 d\Omega_{\underline{X}, \underline{S}, \underline{T}} = - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \sum_{i=1}^{n-1} \frac{dR}{dR} \frac{dR_i}{dR_i} W^* W K K_i + \\
 & + R R_i \frac{dW^*}{dW^*} \frac{dW_i}{dW_i} K K_i + R R_i W^* W_i \frac{dK}{dK} \frac{dK_i}{dK_i} d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \quad (28) \\
 & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R W^* K (f(\underline{X}, \underline{S}) + g(\underline{X}, \underline{T})) d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \\
 & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R(\Gamma) W^* K q d\Omega_{\underline{X}, \underline{S}, \underline{T}}
 \end{aligned}$$

Taking into account the following notation for the known terms

$$\begin{aligned}
 r_1 &= \int_{\Omega_{\underline{X}}} R^2 d\Omega_{\underline{X}} & r_6 &= \int_{\Omega_{\underline{X}}} R F_j^T d\Omega_{\underline{X}} & t_4 &= \int_{\Omega_{\underline{T}}} \frac{dK}{dK} \frac{dK_i}{dK_i} d\Omega_{\underline{T}} \\
 r_2 &= \int_{\Omega_{\underline{X}}} \left( \frac{dR}{dR} \right)^2 d\Omega_{\underline{X}} & r_7 &= \int_{\Omega_{\underline{X}}} R(\Gamma) d\Omega_{\underline{X}} & t_5 &= \int_{\Omega_{\underline{T}}} K G_j^T d\Omega_{\underline{T}} \\
 r_3 &= \int_{\Omega_{\underline{X}}} R R_i d\Omega_{\underline{X}} & t_1 &= \int_{\Omega_{\underline{T}}} K^2 d\Omega_{\underline{S}} & t_6 &= \int_{\Omega_{\underline{T}}} K d\Omega_{\underline{T}} \\
 r_4 &= \int_{\Omega_{\underline{X}}} \frac{dR}{dR} \frac{dR_i}{dR_i} d\Omega_{\underline{X}} & t_2 &= \int_{\Omega_{\underline{T}}} \left( \frac{dK}{dK} \right)^2 d\Omega_{\underline{T}} \\
 r_5 &= \int_{\Omega_{\underline{S}}} R F_j^S d\Omega_{\underline{X}} & t_3 &= \int_{\Omega_{\underline{T}}} K K_i d\Omega_{\underline{S}}
 \end{aligned} \quad (29)$$

Equation 28 is reduced to

$$\begin{aligned}
 & \int_{\Omega_{\underline{S}}} r_2 W^* W t_1 + r_1 \frac{dW^*}{dW^*} \frac{dW}{dW} t_1 + r_1 W^* W t_2 d\Omega_{\underline{S}} = \\
 & - \int_{\Omega_{\underline{S}}} \sum_{i=1}^{n-1} r_4 W^* W t_3 + r_3 \frac{dW^*}{dW^*} \frac{dW_i}{dW_i} t_3 + r_3 W^* W_i t_4 d\Omega_{\underline{S}} + \\
 & + \int_{\Omega_{\underline{S}}} W^* \left( \sum_{j=1}^F r_5 t_6 \alpha^S G^S(\underline{S}) + t_5 r_6 \alpha^T \right) d\Omega_{\underline{X}} - \\
 & - r_7 t_6 q \int_{\Omega_{\underline{S}}} W^* d\Omega_{\underline{S}}. \quad (30)
 \end{aligned}$$

### Computing $K(\underline{T})$ from $R(\underline{X}), W(\underline{S})$

$$\begin{aligned}
 & \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \left( \frac{dR}{dR} \right)^2 W^2 K^* K + R^2 \left( \frac{dW}{dW} \right)^2 K^* K + R^2 W^2 \cdot \\
 & \cdot \frac{dK^*}{dK^*} \frac{dK}{dK} d\Omega_{\underline{X}, \underline{S}, \underline{T}} = - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} \sum_{i=1}^{n-1} \frac{dR}{dR} \frac{dR_i}{dR_i} W W_i K^* K_i + \\
 & + R R_i \frac{dW}{dW} \frac{dW_i}{dW_i} K^* K_i + R R_i W W_i \frac{dK^*}{dK^*} \frac{dK_i}{dK_i} d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \quad (31) \\
 & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R W K^* (f(\underline{X}, \underline{S}) + g(\underline{X}, \underline{T})) d\Omega_{\underline{X}, \underline{S}, \underline{T}} - \\
 & - \int_{\Omega_{\underline{X}, \underline{S}, \underline{T}}} R(\Gamma) W K^* q d\Omega_{\underline{X}, \underline{S}, \underline{T}}
 \end{aligned}$$

Taking into account the following notation for the known terms

$$\begin{aligned}
 r_1 &= \int_{\Omega_{\underline{X}}} R^2 d\Omega_{\underline{X}} & r_6 &= \int_{\Omega_{\underline{X}}} R F_j^T d\Omega_{\underline{X}} & w_4 &= \int_{\Omega_{\underline{S}}} \frac{dW}{dW} \frac{dW_i}{dW_i} d\Omega_{\underline{S}} \\
 r_2 &= \int_{\Omega_{\underline{X}}} \left( \frac{dR}{dR} \right)^2 d\Omega_{\underline{X}} & r_7 &= \int_{\Omega_{\underline{X}}} R(\Gamma) d\Omega_{\underline{X}} & w_5 &= \int_{\Omega_{\underline{S}}} W G_j^S d\Omega_{\underline{S}} \\
 r_3 &= \int_{\Omega_{\underline{X}}} R R_i d\Omega_{\underline{X}} & w_1 &= \int_{\Omega_{\underline{S}}} W^2 d\Omega_{\underline{S}} & w_6 &= \int_{\Omega_{\underline{S}}} W d\Omega_{\underline{S}} \\
 r_4 &= \int_{\Omega_{\underline{X}}} \frac{dR}{dR} \frac{dR_i}{dR_i} d\Omega_{\underline{X}} & w_2 &= \int_{\Omega_{\underline{S}}} \left( \frac{dW}{dW} \right)^2 d\Omega_{\underline{S}} \\
 r_5 &= \int_{\Omega_{\underline{S}}} R F_j^S d\Omega_{\underline{X}} & w_3 &= \int_{\Omega_{\underline{S}}} W W_i d\Omega_{\underline{S}}
 \end{aligned} \quad (32)$$

Equation 31 is reduced to

$$\begin{aligned}
 & \int_{\Omega_{\underline{T}}} r_2 w_1 K^* K + r_1 w_2 K^* K + r_1 w_1 \frac{dK^*}{dW^*} \frac{dK}{dK} d\Omega_{\underline{T}} = \\
 & - \int_{\Omega_{\underline{T}}} \sum_{i=1}^{n-1} r_4 w_3 K^* K + r_3 w_4 K^* K + r_3 w_3 \frac{dK^*}{dK^*} \frac{dK_i}{dK_i} d\Omega_{\underline{T}} + \\
 & + \int_{\Omega_{\underline{T}}} K^* \left( \sum_{j=1}^F r_5 w_5 \alpha^S + r_6 w_6 \alpha^T G^T(\underline{S}) \right) d\Omega_{\underline{T}} - \\
 & - r_7 w_6 q \int_{\Omega_{\underline{T}}} K^* d\Omega_{\underline{T}}. \quad (33)
 \end{aligned}$$