UNIVERSITY OF TRIESTE

Department of Engineering and Architecture

Master's Degree in
Computer & Electronic Engineering

A very interesting thesis

May 10, 2024

Candidate	Supervisor
Kevin Marzio	**Prof. Andrea De Lorenzo**
	Co-supervisor
	Title(?) Riccardo Zamolo

A.Y. 2023/2024

Lorem ipsum
dolor sit amet

- Cicero

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

Introduction

Spiegazione/storia mesh-based.

Perché siamo arrivati ai meshless e differenze in generale

Perché diff. automatica e aggiunto nel meshless

Chapter 1

Meshless methods

Meshless or Meshfree Methods (MMs) were developed to overcome the drawbacks of traditional mesh-based methods for the solution of Partial Differential Equations (PDE), where their true advantage is that "the approximation of unknowns in the PDE is constructed based on scattered points without mesh connectivity " [2].

They appeared for the first time in 1977 with the Smooth Particle Hydrodynamics (SPH) method [1], initially used to modeling astrophysical phenomena such as exploding stars and dust clouds was later applied in solid mechanics to overcome limitations of mesh-based methods [16]. To improve accuracy, tensile instability and spatial instability of SPH many more modern MMs have been developed: the introduction of Reproducing Kernel Particle Method (RKPM) [3] is a prime example of enhanced consistency and stability. Generalized Finite Difference (GFD) methods is another branch of numerical methods for solving PDEs that do not rely on a grid structure and many modern MMs originate from the employment of this approximation for solving PDEs.

The typical use case for MMs is the construction of an approximating field u^h for the sought solution $u\colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$ of the following boundary value problem:

$$\begin{cases} \mathcal{L}u(\boldsymbol{x}) = f(\boldsymbol{x}) & \text{in } \Omega \\ \mathcal{B}u(\boldsymbol{x}) = g(\boldsymbol{x}) & \text{on } \partial\Omega \end{cases} \tag{1.0.1}$$

where $\mathcal{L}$ is a generic linear differential operator and $\mathcal{B}$ is a different linear operator used to enforce some Boundary Condition (BC) that does not necessarily involves partial derivatives. f and g are known functions. Usually in the Computational

Fluid Mechanics (CFD) world the encountered BC are:

$$\text{Dirichlet BC:} \qquad u = g \tag{1.0.2}$$

$$\text{Neumann BC:} \qquad \frac{\partial u}{\partial \boldsymbol{n}} = g \tag{1.0.3}$$

$$\text{Robin BC:} \qquad au + b\frac{\partial u}{\partial \boldsymbol{n}} = g \tag{1.0.4}$$

where $\frac{\partial u}{\partial \boldsymbol{n}}$ indicate the normal derivative of u whereas a and b are known functions. It can also be noticed that Dirichlet and Neumann BCs are special cases of Robin BC respectively when $b(\boldsymbol{x}) = 0$ and $a(\boldsymbol{x}) = 0$.

Regardless on the Meshfree approach, the following approximation for any solution u can be written:

$$u^h(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_k B_k(\boldsymbol{x}) \tag{1.0.5}$$

where $B_k \colon \Omega \to \mathbb{R}$ are suitable basis functions and α_k are the expansions coefficients that must be determined. Different choices for the basis functions B_k leads to different formulations and in literature can be found several of these, some examples are: RKPM [3], moving least square (MLS) [4], radial basis function (RBF) [5, 6] and partition of unity (PU) [7]. Furthermore solving the PDE in (1.0.1) with the approximated solution u^h, in general, yields to a non-zero error function ϵ^h given by:

$$\epsilon^h(\boldsymbol{x}) = \mathcal{L}u^h(\boldsymbol{x}) - f(\boldsymbol{x}) \tag{1.0.6}$$

Once the general form of the approximated solution in (1.0.5) has been properly defined, it can be employed for discretizing the PDE, reported in (1.0.1), over a set of N generated nodes $\{\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \,\}$ distributed in the physical domain $\Omega \cup \partial\Omega$. The Weighted Residual Method is used to do so: a set of test functions $\{\, \Gamma_1, \ldots, \Gamma_N \,\}$ orthogonal to ϵ^h are used to integrate the error to zero:

$$\begin{aligned} \int_\Omega \Gamma_i \epsilon^h \, d\Omega &= \int_\Omega \Gamma_i(\mathcal{L}u^h(\boldsymbol{x}) - f(\boldsymbol{x}))\, d\Omega \\ &= \int_\Omega \Gamma_i \Big[\mathcal{L}\Big(\sum_{k=1}^{N} \alpha_k B_k(\boldsymbol{x})\Big) - f(\boldsymbol{x})\Big]\, d\Omega = 0 \end{aligned} \tag{1.0.7}$$

where $i = 1, \ldots, N$. From the choice of functions Γ_i the following two formulations are obtained [2]:

Galerkin Meshless Methods that find a weak solution for the PDE by using as test functions the basis functions B_k. These formulations require domain integration and special techniques to enforce boundary conditions;

Collocation Meshless Methods that find a strong solution for the PDE by using Dirac delta functions centered at the discretization nodes as test functions. Basically they enforce equaitons (1.0.1) on a finite set of nodes, and by doing so, they do not require any domain integration.

Finally, once constraints (1.0.7) are enforced, the solution of problem (1.0.1), i.e. the values $\{\, u^h(\boldsymbol{x}_1), \dots, u^h(\boldsymbol{x}_N) \,\}$, can be found solving the linear system resulting from PDE's discretization. We remark that the system obtained is not linear in general, but it is in this case since the PDE is linear. Nevertheless, also non-linear PDEs can be reduced to the aforementioned case thanks to a proper linearization.

1.1 Motivation of case study

In this section we motivate the specific case that will be studied in the following chapters: the generic heat equation with internal heat generation [8] at steady state given by

$$\begin{cases} -\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}) & \text{in } \Omega \\ u(\boldsymbol{x}) = g(\boldsymbol{x}) & \text{on } \partial\Omega \end{cases} \tag{1.1.1}$$

The reason behind this choice are due to both its wide range of applications in different areas (e.g. electrostatic, chemistry, gravitation and others) and the simplicity of its analytical manipulation. Equation (1.1.1) is one of the many PDEs that can be solved using the meshless approaches.

In particular for this thesis we will:

- consider domains $\Omega \subset \mathbb{R}^d$ with $d = 1, 3$;
- use RBFs augmented with polynomial terms as basis functions B_k to approximate the solution u in a similar way to what has been done in [9];
- formulate the problem in its strong form, so we will use the collocation technique in combination with the Weighted Residual Method.

thus we will apply the RBF-generated Finite Differences (RBF-FD) [10, 11] to solve (1.1.1) in one and three dimensional physics domain.

Mono dimensional physics domain are used to gain confidence with the techniques explained while three dimensional domain are used for their application in real case problems as those faced at Esteco. 2D cases, instead, have been skipped in the analysis since they would only be used to bridge the 2 previous scenarios and therefore would have had no practical utility.

RBFs are used for scattered data interpolation both for their physical foundation[overview 20, Hardy 1990] and for their profitable use in applications like meteorology, turbulence analysis and neural network [2]

Finally, collocation technique is employed because thanks to its ability to discretize the Boundary Value problem (1.1.1) expressed in strong form leads to a real fully meshless approach [12].

Chapter 2

RBF-FD method

In this chapter we explain in more details the MM used within this work: the Radial Basis Function generated Finite Differences (RBF-FD) method. To do so we consider a physical domain $\Omega \cup \partial\Omega \subset \mathbb{R}^d$ indicating with Ω the open subset, $\partial\Omega$ its boundary and with $d \in \mathbb{N}$ its dimension; over the presented domain we define a boundary value problem with the same form of the one reported in (1.0.1) and we show how RBF-FD method can be used to solve it.

As any other MM, before starting, a set of N points distributed over the domain where to discretize the PDE is required. We indicate it with $\mathcal{X} := \{\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \mid \boldsymbol{x_i} \in \Omega \cup \partial\Omega,\, i = 1, \ldots, N \,\}$. To remark also the fact that nodes are placed both inside the domain and on its boundary we partition $\mathcal{X}$ in $\mathcal{X}_\mathcal{I}$, the set of internal nodes, and $\mathcal{X}_\mathcal{B}$, the set of boundary nodes, defined as:

$$\mathcal{X}_\mathcal{I} := \{\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_I} \mid \boldsymbol{x}_i \in \Omega,\, i = 1, \ldots, N_I \,\}$$
$$\mathcal{X}_\mathcal{B} := \{\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_B} \mid \boldsymbol{x}_i \in \partial\Omega,\, i = 1, \ldots, N_B \,\}$$

where N_I and N_B indicate respectively the number of nodes inside and on the boundary. Of course $N_I + N_B = N$ and $\mathcal{X}_\mathcal{I} \cup \mathcal{X}_\mathcal{B} = \mathcal{X}$.

This method, like other MMs, looks for an approximated solution of problem (1.0.1) in the form of (1.0.5).

2.1 RBF Interpolation

Every MM, as we have already seen, on its core, is just a way to approximate the solution of a PDE and RBF-FD method is no exception. The tool used to do so is called scattered data interpolation and in this subsection, after explaining what it is, we see how it is related to the solution of problem (1.0.1). During the explanation we start by seeing how it is used in general by MMs so to avoid becoming fixated on a single implementation and losing generality, furthermore this way of proceeding establishes all the steps that are also used in case of RBF basis functions.

In general, given the set of nodes $\mathcal{X}$, and a set of known real values $u(\boldsymbol{x}_1), \dots, u(\boldsymbol{x}_N)$, the interpolation problem results on finding a continuous function $u^h \colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$ that satisfy:

$$u^h(\boldsymbol{x_i}) = u(\boldsymbol{x}_i) \qquad \forall \boldsymbol{x}_i \in \mathcal{X} \tag{2.1.1}$$

MMs also aim to provide an approximation for an unknown function, but in addition give an hint on its form as reported in equation (1.0.5), that we report here for convenience, without further specifying the coefficients values:

$$u^h(x) = \sum_{k=1}^{N} \alpha_k B_k(\boldsymbol{x}) \tag{2.1.2}$$

Here is where the theory of scattered data interpolation comes in: it tells us that we are able to find the numerical values for $\alpha_1, \dots \alpha_N$ if we impose a number of conditions equal to the number of coefficients that we are looking for. These conditions must have a form like that shown in equation (2.1.1). Therefore if we replace the generic meshless approximation within each of the N interpolation conditions we can write the following system of linear equation:

$$\underbrace{\begin{bmatrix} B_1(\boldsymbol{x}_1) & \dots & B_N(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ B_1(\boldsymbol{x}_N) & \dots & B_N(\boldsymbol{x}_N) \end{bmatrix}}_{\boldsymbol{B}} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} u(\boldsymbol{x}_1) \\ \vdots \\ u(\boldsymbol{x}_N) \end{bmatrix} \tag{2.1.3}$$

that, once solved, provides us with the desired coefficients that make up u^h. We would like to stress the fact that the right hand side vector is made up of values of the unknown exact solution of problem (1.0.1), that we only *assume* to know.

During the solution of problem (1.0.1) the following constraints are enforced instead:

$$\begin{aligned} \mathcal{L}u^h(\boldsymbol{x}_j) &= f(\boldsymbol{x}_j) \quad \text{if } \boldsymbol{x}_j \in \Omega \\ u^h(\boldsymbol{x}_j) &= g(\boldsymbol{x}_j) \quad \text{if } \boldsymbol{x}_j \in \partial\Omega \end{aligned} \tag{2.1.4}$$

and granting that first N_I nodes belongs to Ω and the last N_B to $\partial\Omega$, we can find the values of the approximated solution at the points in $\mathcal{X}$ by solving:

$$\begin{bmatrix} c_{1,1} & \dots & c_{1,N_I} \\ \vdots & \ddots & \vdots \\ c_{N_I,1} & \dots & c_{N_I,N_I} \end{bmatrix} \begin{bmatrix} u^h(\boldsymbol{x}_1) \\ \vdots \\ u^h(\boldsymbol{x}_{N_I}) \end{bmatrix} = \boldsymbol{f} - \begin{bmatrix} c_{1,N_I+1} & \dots & c_{1,N_B} \\ \vdots & \ddots & \vdots \\ c_{N_I,N_I+1} & \dots & c_{N_I,N_B} \end{bmatrix} \boldsymbol{g} \tag{2.1.5}$$

where $\boldsymbol{f} = [f(\boldsymbol{x}_1)\dots f(\boldsymbol{x}_{N_I})]^T$ and $\boldsymbol{g} = [g(\boldsymbol{x}_{N_I+1})\dots g(\boldsymbol{x}_{N_B})]^T$, and the coefficient matrix $\boldsymbol{C}$ is found as the solution of:

$$\begin{bmatrix} c_{1,1} & \dots & c_{N_I,1} \\ \vdots & \ddots & \vdots \\ c_{1,N} & \dots & c_{N_I,N} \end{bmatrix} = \boldsymbol{B}^{-T} \begin{bmatrix} \mathcal{L}B_1(\boldsymbol{x}_1) & \dots & \mathcal{L}B_1(\boldsymbol{x}_{N_I}) \\ \vdots & \ddots & \vdots \\ \mathcal{L}B_N(\boldsymbol{x}_1) & \dots & \mathcal{L}B_N(\boldsymbol{x}_{N_I}) \end{bmatrix} \tag{2.1.6}$$

By carefully analyzing each row $\boldsymbol{c}_i = [c_{i,1},\dots c_{i,N}]$ of matrix $\boldsymbol{C}$ we can notice that are computed solving the following linear systems:

$$\boldsymbol{B}^T c_i = \begin{bmatrix} \mathcal{L}B_1(\boldsymbol{x_i}) \\ \vdots \\ \mathcal{L}B_N(\boldsymbol{x_i}) \end{bmatrix} \qquad i = 1,\dots,N_I \tag{2.1.7}$$

which are closely related to the one obtained from scattered data interpolation reported in (2.1.3) due to the presence of the same matrix $\boldsymbol{B}$. We conclude by commenting that equation (2.1.6) with matrix $\boldsymbol{B}$ defined as in (2.1.3) holds true only in case of Dirichlet boundary conditions, otherwise $\boldsymbol{B}$ would take on a different form.

In the case of RBF-FD method the implementation of scattered data interpolation and the solution of governing equation remain the same and leads to a matrix $\boldsymbol{B}$ defined as:

$$\boldsymbol{B} = \begin{bmatrix} \Phi(\boldsymbol{x}_1,\boldsymbol{x}_1) & \dots & \Phi(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \Phi(\boldsymbol{x}_N,\boldsymbol{x}_1) & \dots & \Phi(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{bmatrix} \tag{2.1.8}$$

2.2 Radial Basis Functions

Radial Basis Functions (RBFs) are the basis functions used in (1.0.5) by the RBF-FD method to approximate the solutions of PDEs and are defined as:

$$\Phi_k(\boldsymbol{x}) = \varphi(\|\boldsymbol{x} - \boldsymbol{x}_k\|_2) \tag{2.2.1}$$

where $\boldsymbol{x}_k \in X$ is a given point , $\|\cdot\|_2$ is the euclidean distance and $\varphi\colon \Omega \cup \partial\Omega \to \mathbb{R}$, named basic function, is a scalar field which takes $\boldsymbol{x}$ as input and it is used as generator for all the basis functions. In general different basic functions φ can be used, some of them are reported in table 2.1.

To further clarify the RBFs name we can notice that they are called:

Radial since the value of each $\phi_k(\boldsymbol{x})$ at each point $\boldsymbol{x}$ depends only on the distance between that point and $\boldsymbol{x}_k$ through $\|\cdot\|_2$, they satisfy radial symmetry;

Table 2.1: Examples of basic functions where $r_k = \|\boldsymbol{x} - \boldsymbol{x}_k\|_2$ and ϵ, called shape factor, is a suitable parameter

Name	$\varphi(r_k)$
Multiquadratic	$\sqrt{1+(1+\epsilon r_k)^2}$
Inverse multiquadratic	$\left(\sqrt{1+(1+\epsilon r_k)^2}\right)^{-1}$
Thin plate splines	$r_k^l \log r_k$, l even
Gaussian	$e^{-(\epsilon r_k)^2}$
Polyharmonics	r_k^l, l odd

Basis since the set of radial functions $\phi_k(\boldsymbol{x})$ with $k = 1, \dots, N$ form a basis for the space of functions:

$$F_\Phi := \left\{ \sum_{k=1}^{N} \alpha_k \Phi_k(\boldsymbol{x}), \quad \alpha_k \in \mathbb{R} \right\}$$

2.3 The Mairhuber-Curtis Theorem

From the previous discussion, in particular from equation (2.1.6), it can be noticed that matrix $\boldsymbol{B}$ has to be non singular in order to be able to solve the boundary value problem, and this must hold for each node placement $\mathcal{X}$ (to be read as every possible discretization of the problem domain) as long nodes are distinct. This property of $\boldsymbol{B}$ turns out to be dependent on the choice of the particular set of basis functions: for example if we assume $B_k(\boldsymbol{x}) \in \Pi_P^d$ and $\{ B_1(\boldsymbol{x}), \dots, B_N(\boldsymbol{x}) \}$ to be a polynomial basis of the space Π_P^d of polynomials of degree at most P in $\mathbb{R}^d$, then we are not able to guarantee that $\boldsymbol{B}$ is invertible for $d > 1$.

This issue is explained in more detail by the Mairhuber-Curtis theorem [13]; when dealing with the multidimensional case it is possible to continuously move two nodes such that they end up by interchanging their original positions without one crossing the path of the other. If these 2 are the only nodes of $\mathcal{X}$ that are moved, $\boldsymbol{B}$ ends up with 2 rows exchanged leading to a change in the sign of its determinant, and, since the determinant is a continuous function, this means that there is a moment when the latter vanishes making the matrix singular.

The inconvenience arises from the fact that the set of basis functions is independent from the node position and could be solved by simply choosing a basis that is function of nodes position. By doing so we no more fall in the case of the Mairhuber-Curtis theorem since whenever we move nodes also the base itself changes and if two nodes switches their positions not only their respective rows in

$\boldsymbol{B}$ are switched, but also their columns, forcing the determinant not to change in sign.

2.4 Polynomial augmentation

Setting aside the issue of the invertibility of matrix $\boldsymbol{B}$ in case of particular nodes arrangement discussed in previous subsection, we should also take into account the accuracy of the interpolation that we're able to achieve, which also depends upon the type of functions that we are supposed to aprroximate. Indeed RBFs approximation schemes alone are not able to interpolate constant, linear or higher degree polynomials fields and this is an issue since they are important in different engineering applications such modeling of constant strain in elastic bodies and steady temperature fields in differentially heated walls [14].

To overcome this limitation a polynomial augmentation of degree P is required, leading to the overall formulation for the RBF interpolant:

$$u^h(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \Phi_k(\boldsymbol{x}) + \sum_{k=1}^{M} \beta_k p_k(\boldsymbol{x}) \tag{2.4.1}$$

where $M = \binom{P+D}{D}$ is the number of polynomial basis functions with degree $P \leq D$, $\{\, p_1(\boldsymbol{x}) \dots p_M(\boldsymbol{x}) \,\}$ is a complete polynomial basis of Π_P^d and β_j are the corresponding coefficients to each function in the former basis. An example of polynomial basis for polynomials of degree $P = 1$ in $2D$ has the following $M = 3$ elements: $p_1(x, y) = 1$, $p_2(x, y) = x$, $p_3(x, y) = y$.

We must also note that using an interpolant with the introduced polynomial augmentation, during the approximated solution of problem (1.0.1) leads to an underdetermined system in (2.1.3). In order to obtain a square $\boldsymbol{B}$ and thus having a solvable system, the following orthogonality conditions, that guarantees polynomial reproduction, have to be imposed:

$$\sum_{i=1}^{N} \alpha_i p_k(\boldsymbol{x}_i) = 0, \qquad i = 1, \dots, M \tag{2.4.2}$$

The coefficients of u^h, which are now composed not only by $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_N]$, but also by $\boldsymbol{\beta} = [\beta_1 \dots \beta_M]$, can then be found by solving the following system:

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{0} \end{bmatrix} \tag{2.4.3}$$

where:

$$\begin{gathered}\boldsymbol{P} = \begin{bmatrix} p_0(\boldsymbol{x}_1) & \dots & p_M(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ p_0(\boldsymbol{x}_N) & \dots & p_M(\boldsymbol{x}_N) \end{bmatrix} \\ \boldsymbol{u} = [u(\boldsymbol{x}_1) . \dots u(\boldsymbol{x}_N)] \end{gathered} \tag{2.4.4}$$

From its formulation is easy to understand that the system above is simply the composition of the system in equation (2.1.3), first row, with constraints (2.4.2) written in compact form, second row.

In practice the addition of polynomial basis to the RBF interpolant let us perfectly fit $u(\boldsymbol{x})$ not only on collocation points where we have $u^h(\boldsymbol{x}_i) = u(\boldsymbol{x}_i)$, but also across the rest of the domain provided that data $u(\boldsymbol{x}_1) \dots u(\boldsymbol{x}_N)$ come from a polynomial of total degree less than or equal to P. Nevertheless this procedure has as a side effect since not all the basic functions φ leads to a well-posed RBF interpolation with a non singular matrix M, but only the strictly conditionally positive definite of order $P + 1$ ones.

Chapter 3

Automatic differentiation and adjoint method

Cos'è e come funziona la differenz. automatica

Cos'è e come funziona l'aggiunto e legami con la diff. automatica

Come viene applicato il metodo dell'aggiunto nel metodo RBF-FD

Chapter 4

Results

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.1 Lorem Ipsum

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

4.1.1 Dolor sit amet

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Appendix A

Sit Amet

Bibliography

[1] T. Belytschko et al. “Meshless methods: An overview and recent developments”. In: *Computer Methods in Applied Mechanics and Engineering* (1996).

[2] Michael Hillman Jiun-Shyan Chen and Sheng-Wei Chi. “Meshfree Methods: Progress Made after 20 Years”. In: *Journal of Engineering Mechanics* (2017).

[3] S. Jun W. K. Liu and Y. F. Zhang. “Reproducing kernel particle methods”. In: *International Journal for Numerical Methods in Fluids* (1996).

[4] P. Lancaster and K. Salkauskas. “Surfaces Generated by Moving Least Squares Methods”. In: *Mathematics of Computation* (1981).

[5] E. J. Kansa. “Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates”. In: *Computers & Mathematics with Applications* (1990).

[6] E. J. Kansa. “Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations”. In: *Computers & Mathematics with Applications* (1990).

[7] M. A. Schweitzer. *Partition of Unity Method*. Lecture Notes in Computational Science and Engineering. 2003. URL: https://doi.org/10.1007/978-3-642-59325-3_2.

[8] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. 2011. URL: https://doi.org/10.1007/978-0-387-70914-7.

[9] G. R Liu and Y. T. GU. *An Introduction to Meshfree Methods and Their Programming*. Cambridge University Press, 2005.

[10] B. Fornberg and N. Flyer. *A Primer on Radial Basis Functions with Applications to the Geosciences*. 2015.

[11] B. Fornberg and N. Flyer. “Solving PDEs with radial basis functions”. In: *Acta Numerica* (2015).

[12] E. Nobile D. Miotti R. Zamolo. "A Fully Meshless Approach to the Numerical Simulation of Heat Conduction Problems over Arbitrary 3D Geometries". In: *Energies* (2021).

[13] J. C. Mairhuber. "On Haar's Theorem Concerning Chebychev Approximation Problems Having Unique Solutions". In: *Proceedings of the American Mathematical Society* (1956).

[14] R. Zamolo. "Radial Basis Function-Finite Difference Meshless Methods for CFD Problems". PhD thesis. Università degli studi di Trieste, 2018.

[15] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[16] W. Benz and E. Asphaug. "Simulations of brittle solids using smooth particle hydrodynamics". In: *Computer Physics Communications* (1995).

[17] G. Fasshauer. *Meshfree Approximation Methods with Matlab*. World Scientific Publishing Company, 2007.

[18] L. Bacer. "Fully meshless approaches for the numerical solution of partial differential equations: radial basis function finite difference method and physics-informed neural network". MA thesis. Università degli studi di Trieste, 2023.

[19] K. C. Jeyanthi. "Efficient node generation over complex 3D geometries for meshless RBF-FD method". MA thesis. Università degli studi di Trieste, 2022.

[20] Wikipedia contributors. *Finite difference method*. Online; accessed 21-February-2024. 2024. URL: https://en.wikipedia.org/wiki/Finite_difference_method#Further_reading.

Ei fu. Siccome immobile,
Dato il mortal sospiro,
Stette la spoglia immemore
Orba di tanto spiro

Alessandro Manzoni – *Il Cinque Maggio*