Solving the Helmholtz equation using RBF-FD

Index

- Radial basis functions.
- RBF interpolation.
- RBF methods for PDEs.
  - Global methods.
  - RBF-FD.
- Helmholtz equation.
  - Numerical results.
- Summary.

Radial basis functions (RBF)

- The RBF approach is generally attributed to Rolland Hardy, who in 1971 proposed it for the purpose of interpolating scattered 2-D data.
- In 1990 Edward J. Kansa recognized the ability of RBFs to provide accurate approximations for derivatives of functions known only at scattered data.
- This opened the door for using RBFs to solve PDEs.

Radial basis functions (RBF)

Properties of RBF methods:

- It is a meshless method.
- Easy local grid refinement.
- There are two main methods, global and RBF-FD. Global methods can achieve spectral convergence.
- Coding effort and computational cost of RBFs are independent of the geometry and the dimension.

RBF interpolation

Suppose you know the values of a function at a set of given points:

$$\{x_i\} \quad \{f(x_i)\}$$

Then

$$f(x) = \sum_{i=1}^{N} \alpha_i \phi(||x - x_i||)$$

with α_i the interpolation coeficientes and the ϕs are the RBFs functions.

RBF interpolation

The interpolation coefficients α_i are calculated using collocation:

$$\begin{bmatrix} \phi(\|x_1 - x_1\|) & \phi(\|x_2 - x_1\|) & \cdots & \phi(\|x_N - x_1\|) \\ \phi(\|x_1 - x_2\|) & \phi(\|x_2 - x_2\|) & \cdots & \phi(\|x_N - x_2\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_1 - x_N\|) & \phi(\|x_2 - x_N\|) & \cdots & \phi(\|x_N - x_N\|) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix}$$

To shorten the notation we will call A to the RBF interpolation matrix:

$$A\vec{\alpha} = \vec{f}$$

RBF interpolation

It is very common to add some low order polynomials to the interpolation basis along with certain matching constraints:

$$f(x) = \sum_{i=1}^{N} \alpha_i \phi(x - x_i) + \gamma_1 + [\gamma_1, \gamma_2]x$$

$$\sum_{i=1}^{N} \alpha_i = 0$$

$$\sum_{i=1}^{N} x_i \alpha_i = 0 \qquad \sum_{i=1}^{N} y_i \alpha_i = 0$$

$$\left[\begin{array}{ccc:ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_N & y_N \\ \hdashline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_N & & 0 & \\ y_1 & \cdots & y_N & & & \end{array}\right] \left[\begin{array}{c} \alpha_1 \\ \vdots \\ \alpha_N \\ \hdashline \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array}\right] = \left[\begin{array}{c} f(x_1) \\ \vdots \\ f(x_N) \\ \hdashline 0 \\ 0 \\ 0 \end{array}\right]$$

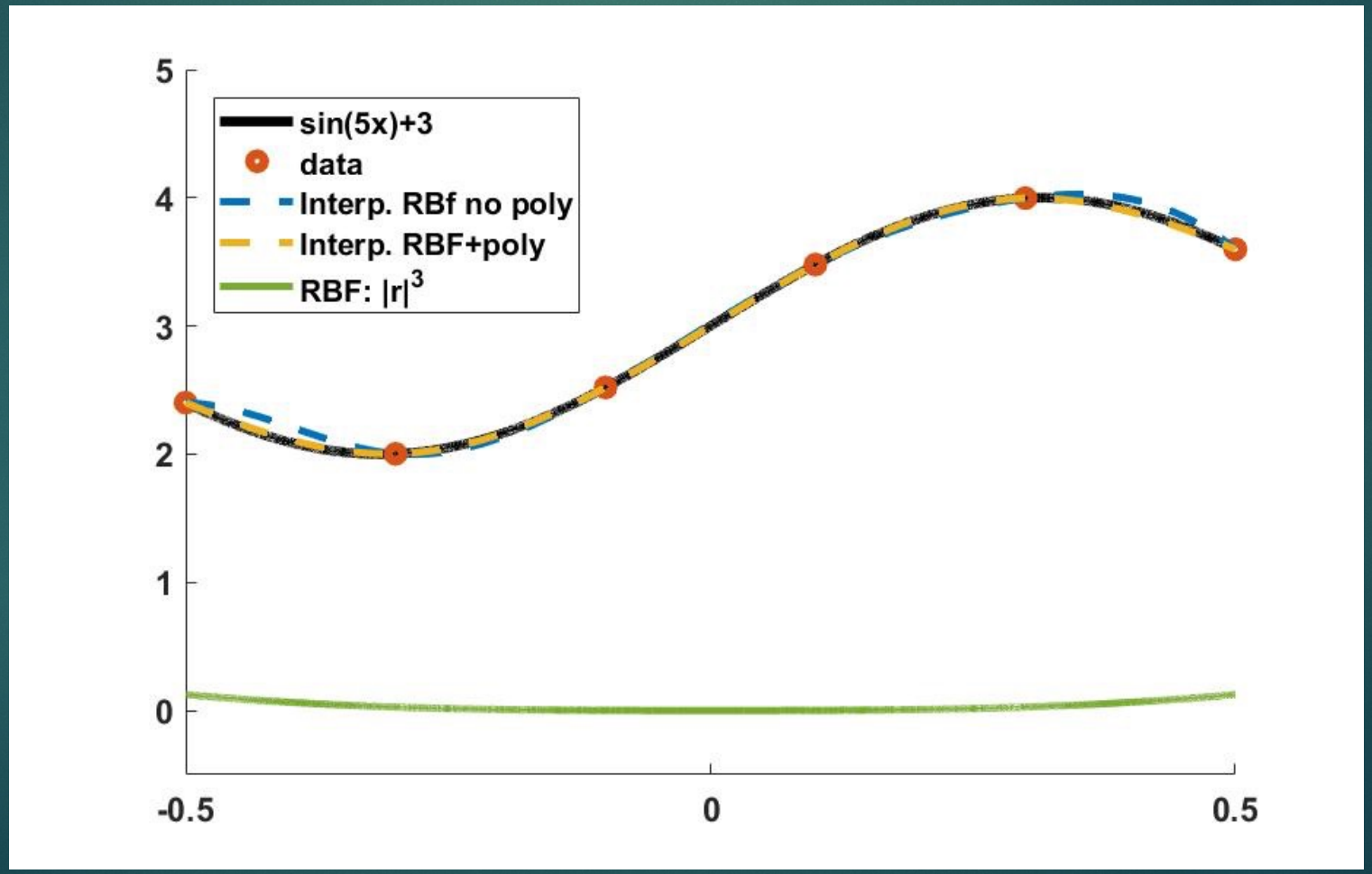
RBF interpolation

There are many kinds of RBFs:

- Polyharmonic splines (PHS): r^k if k is odd, $r^k \ln(r)$ if k is even
- Gaussian (GA): $e^{-(\varepsilon r)^2}$
- Multiquadric (MQ): $\sqrt{1+(\varepsilon r)^2}$
- Bessel (BE) ($d = 1,2,\ldots$) $\dfrac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}}$

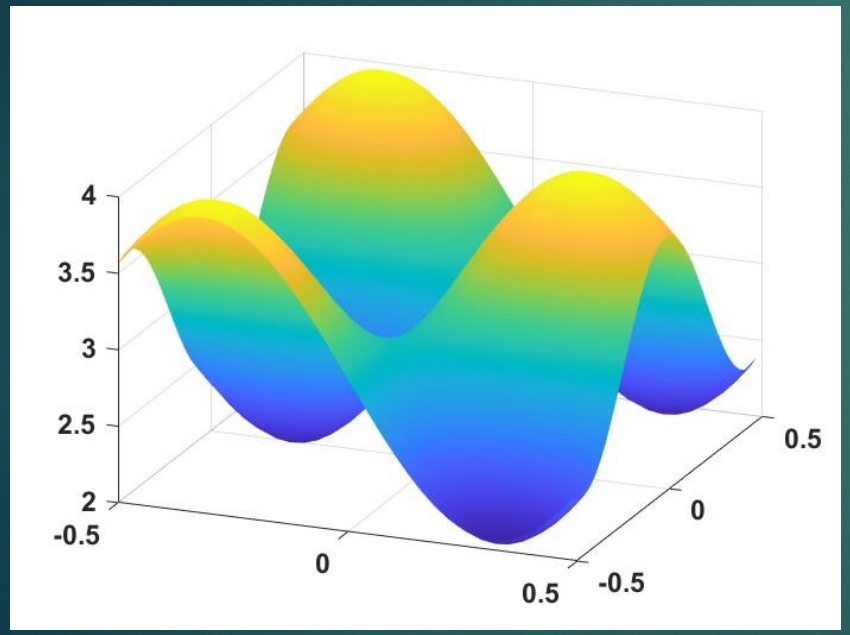
RBF interpolation

1D interpolation example

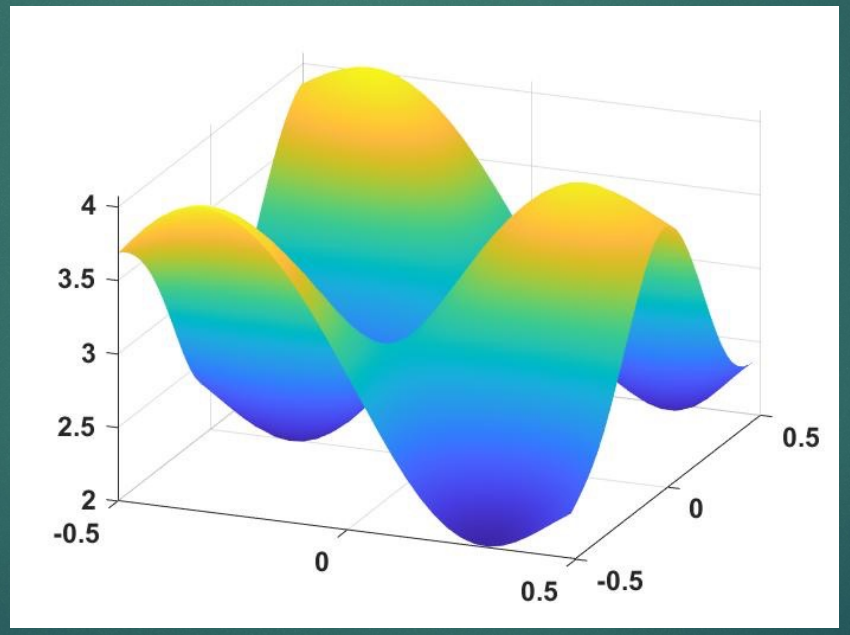


RBF interpolation

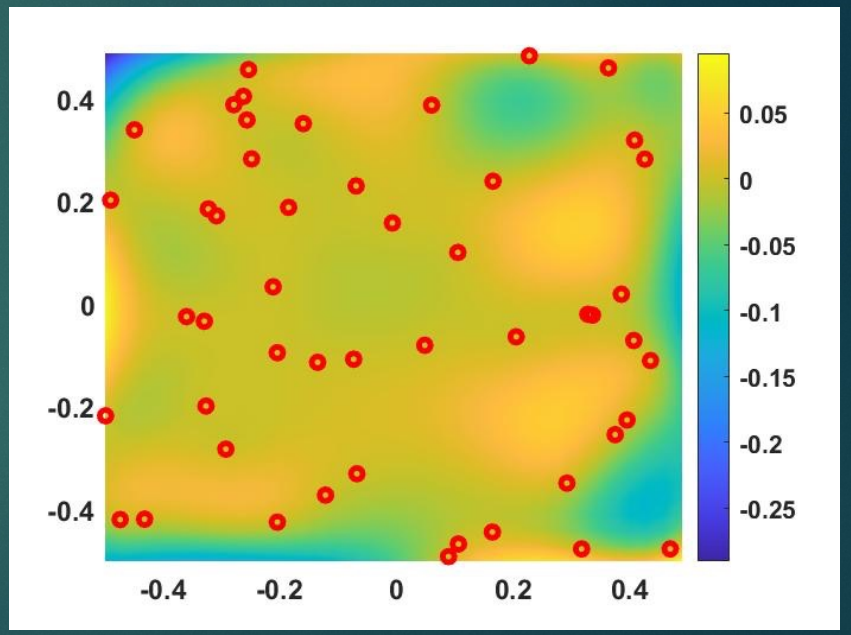
2D interpolation example: $\mathrm{f}(x, y) = \sin(5x)\cos(7y) + 3$



$\mathrm{f}(x, y)$



RBF interpolation



Interpolation error

RBF methods for PDEs

Kansa's idea is to express the solution of the equation as a linear combination of RBFs:

$$Lu = f \quad \text{in } \Omega$$
$$Bu = g \quad \text{in} \partial\Omega$$

$$u(x) = \sum_{i=1}^{N} w_i \phi\left(\|x - x_i\|\right)$$

$$\begin{bmatrix} L\phi\left(\|x_{N_b+1} - x_1\|\right) & \cdots & L\phi\left(\|x_{N_b+1} - x_N\|\right) \\ \vdots & \ddots & \vdots \\ L\phi\left(\|x_N - x_1\|\right) & \cdots & L\phi\left(\|x_N - x_N\|\right) \\ \hline B\phi\left(\|x_1 - x_1\|\right) & \cdots & B\phi\left(\|x_1 - x_N\|\right) \\ \vdots & \ddots & \vdots \\ B\phi\left(\|x_{N_b} - x_1\|\right) & \cdots & B\phi\left(\|x_{N_b} - x_N\|\right) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f\left(x_{N_b+1}\right) \\ \vdots \\ f\left(x_N\right) \\ \hline g\left(x_1\right) \\ \vdots \\ g\left(x_{N_b}\right) \end{bmatrix}$$

RBF methods for PDEs

The implementation of the boundary conditions is complicated…

- Increase the node density.
- Decrease shape parameter.
- Impose the equation over the boundary nodes, along the boundary conditions, and add ghost nodes outside the domain to balance the number of unknowns and the number of equations.

RBF-FD

In 1D finite differences formulas can be calculated using monomials as test functions:

$$Lf(x_c) = \sum_{i=1}^{n} w_i f(x_i)$$

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L1|_{x_c} \\ Lx|_{x_c} \\ \vdots \\ Lx^{n-1}|_{x_c} \end{bmatrix}$$

RBF-FD

This does not work well in higher dimensions or using scattered nodes.

Idea (A. I. Tolstykh 2000): Use RBF instead of monomials:

$$\begin{bmatrix} \phi(\|x_1 - x_1\|) & \phi(\|x_2 - x_1\|) & \cdots & \phi(\|x_n - x_1\|) \\ \phi(\|x_1 - x_2\|) & \phi(\|x_2 - x_2\|) & \cdots & \phi(\|x_n - x_2\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_1 - x_n\|) & \phi(\|x_2 - x_n\|) & \cdots & \phi(\|x_n - x_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\phi(\|x - x_1\|)\big|_{x_c} \\ L\phi(\|x - x_2\|)\big|_{x_c} \\ \vdots \\ L\phi(\|x - x_n\|)\big|_{x_c} \end{bmatrix}$$

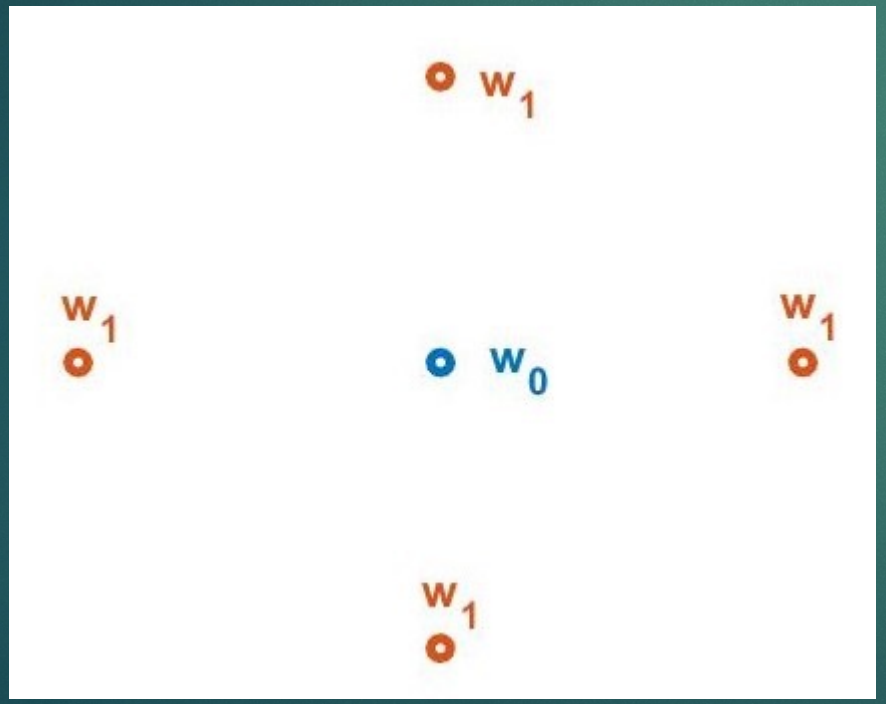
RBF-FD

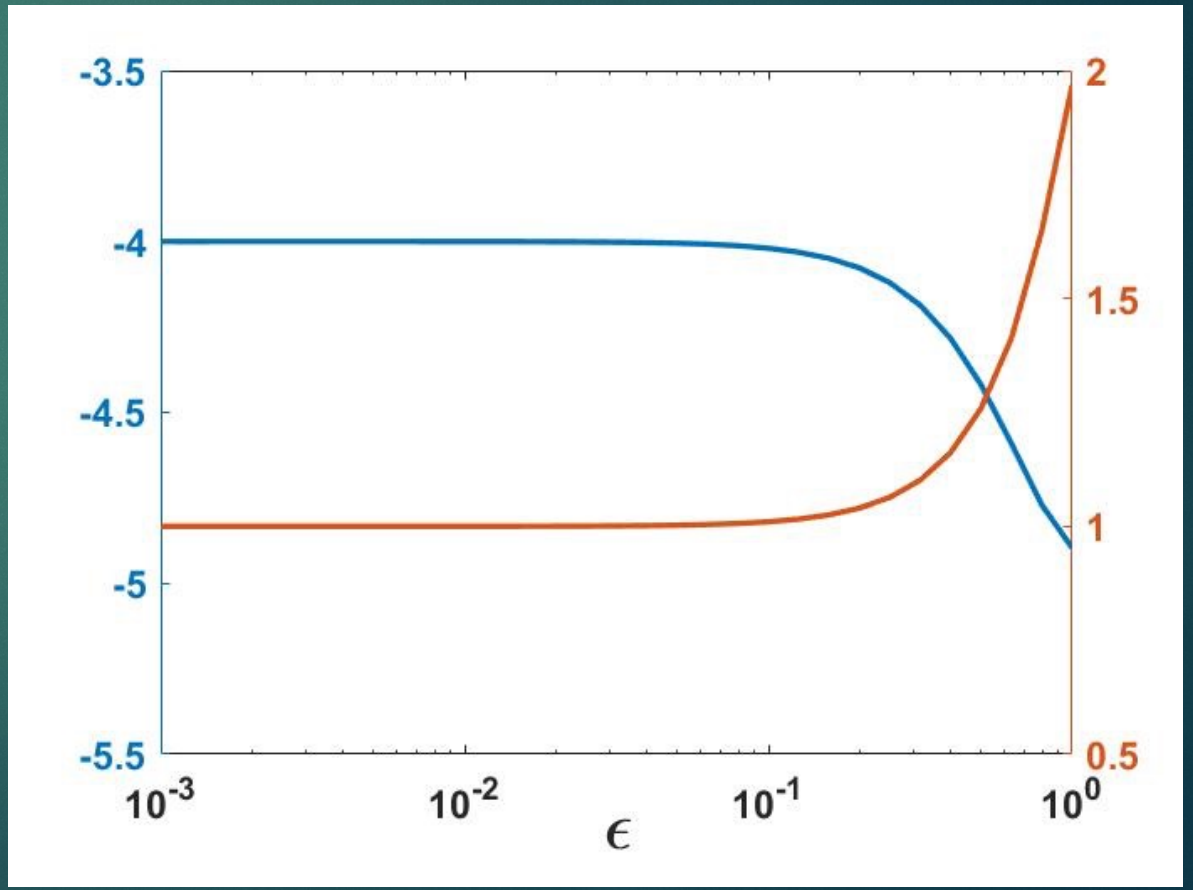
You can get better results augmenting the basis with polynomials:

$$\left[\begin{array}{ccc:ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hdashline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right]\left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hdashline w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} L\phi\left(\|x-x_1\|\right)\big|_{x_c} \\ \vdots \\ L\phi\left(\|x-x_n\|\right)\big|_{x_c} \\ \hdashline L1\big|_{x_c} \\ Lx\big|_{x_c} \\ Ly\big|_{x_c} \end{array}\right]$$

RBF-FD

Example of the calculated Laplacian weights with the Gaussian RBF.





RBF-FD

For the polyharmonic r^3 augmented with a constant you get the weights -4.2426 for the central node and 1.0607 for the rest.

This might look worse than the result obtained by using the Gaussian, but it has the advantage that you don't need to choose a value for ε.

The optimal value for ε depends on the average distance between the stencil nodes.

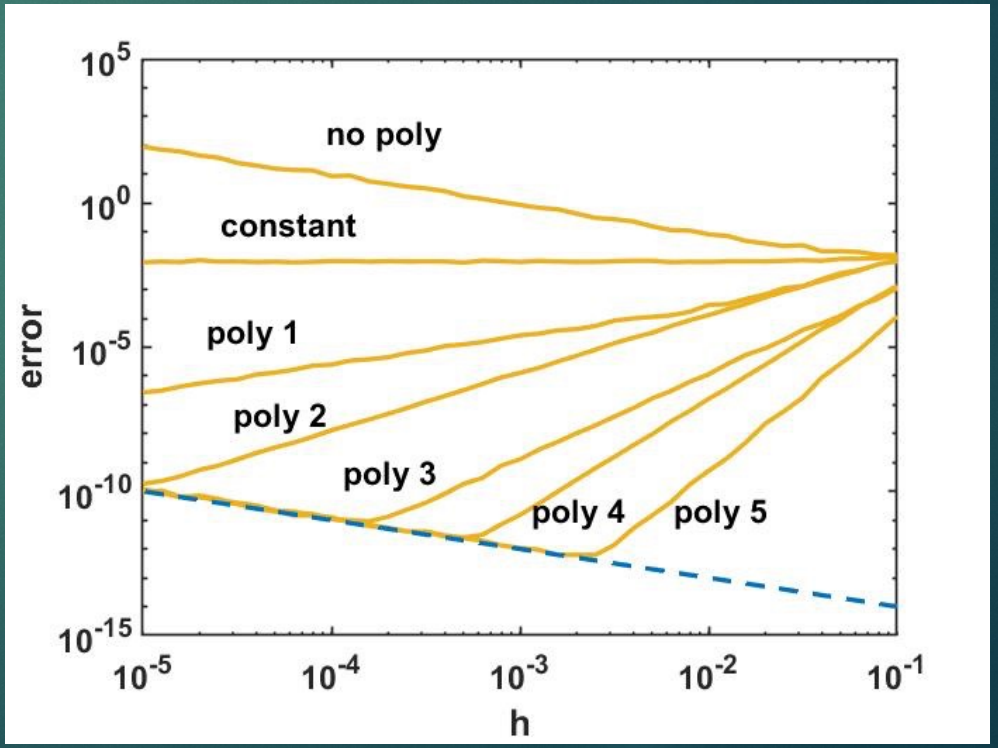
RBF-FD

Example, by Manuel Kindelan, of the effect of the polynomials in the calculation of the first derivative of the function $1 + \sin(4x) + \cos(3x) + \sin(2x)$.

RBF: r^3

Stencil size: 37

300 random node configurations.



RBF-FD

From the previous system you get the FD coefficients for the inner points.

If we use polyharmonic splines augmented with polynomials the coefficients for the boundary conditions are calculated similarly:

$$\left[\begin{array}{ccc:ccc} & & & 1 & x_1 & y_1 \\ & A & & \vdots & \vdots & \vdots \\ & & & 1 & x_n & y_n \\ \hdashline 1 & \cdots & 1 & & & \\ x_1 & \cdots & x_n & & 0 & \\ y_1 & \cdots & y_n & & & \end{array}\right]\left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hdashline w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{array}\right] = \left[\begin{array}{c} B\phi\left(\|x-x_1\|\right)\big|_{x_c} \\ \vdots \\ B\phi\left(\|x-x_n\|\right)\big|_{x_c} \\ \hdashline B1\big|_{x_c} \\ Bx\big|_{x_c} \\ By\big|_{x_c} \end{array}\right]$$

RBF-FD

With this method you don't have to do all the tricks you need for global RBF methods as long as:

- The number of nodes per stencil is at least double the number of polynomials used for the calculation of the weights.

Helmholtz equation

The Helmholtz equation:

$$\nabla(D\nabla u) + k_0^2 u = \mathrm{s} \qquad \text{in } \Omega$$

$$u + \alpha D(\vec{n} \cdot \nabla u) = 0 \qquad \text{in } \delta\Omega$$

Here $D = \dfrac{1}{3(\mu_s(1-g)+\mu_a)}$ and $k_0^2 = -\mu_a + \mathrm{i}\dfrac{\omega}{c_0}$.

Helmholtz equation

We are going to solve it using 10000 nodes, 300 of them on boundary, which means an average distance between nodes of 0.017 .

We use

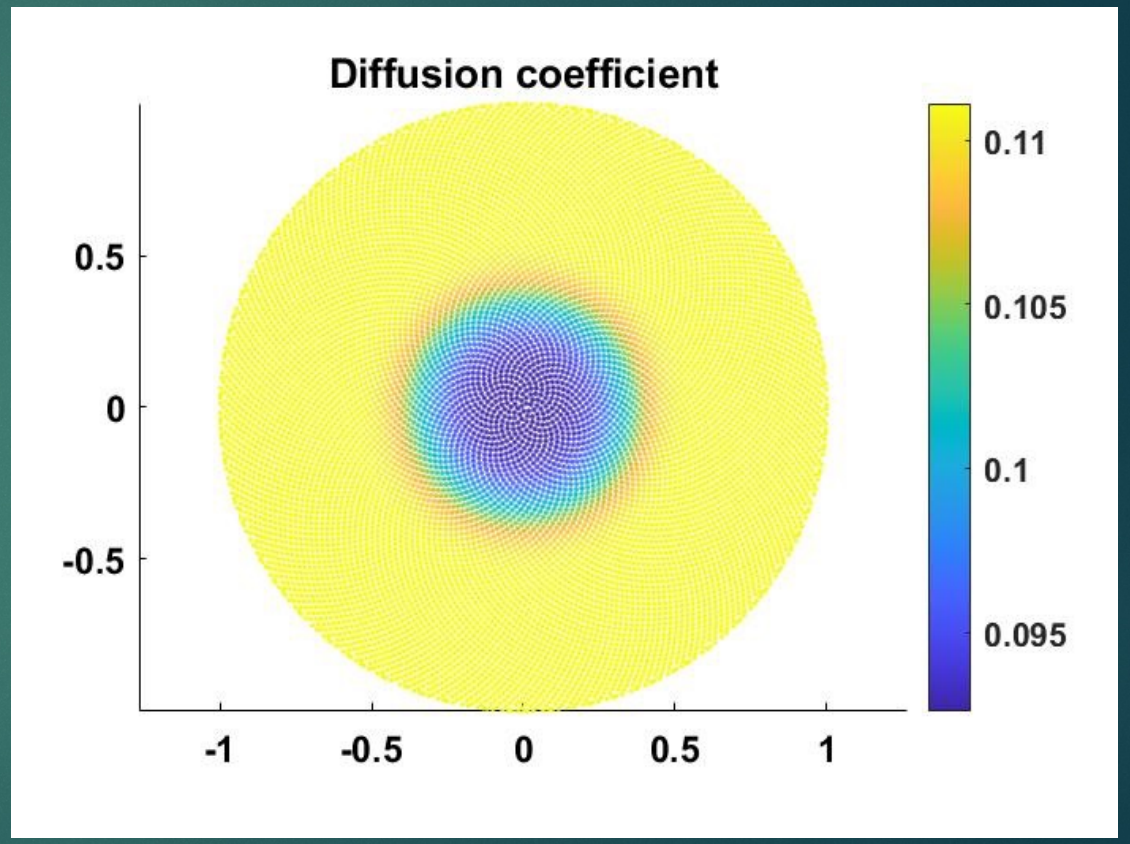
- 31 nodes stencils.
- polyharmonic splines of order 3 ($\phi(r) = r^3$).
- Polynomials up to order four.

Helmholtz equation

$g = 0.8$
$\mu_a \in [1,1.2]$
$\mu_s \in [10,12]$

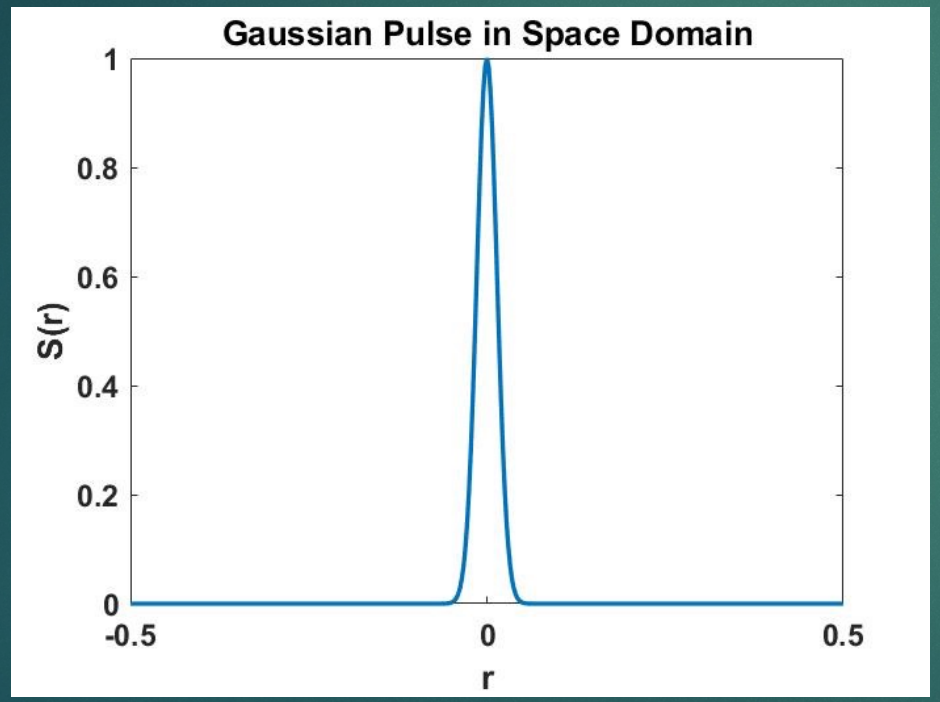
$\omega = (1/750) \cdot 10^9$ (Near infrared)
$c0 = 3 \cdot 10^9$

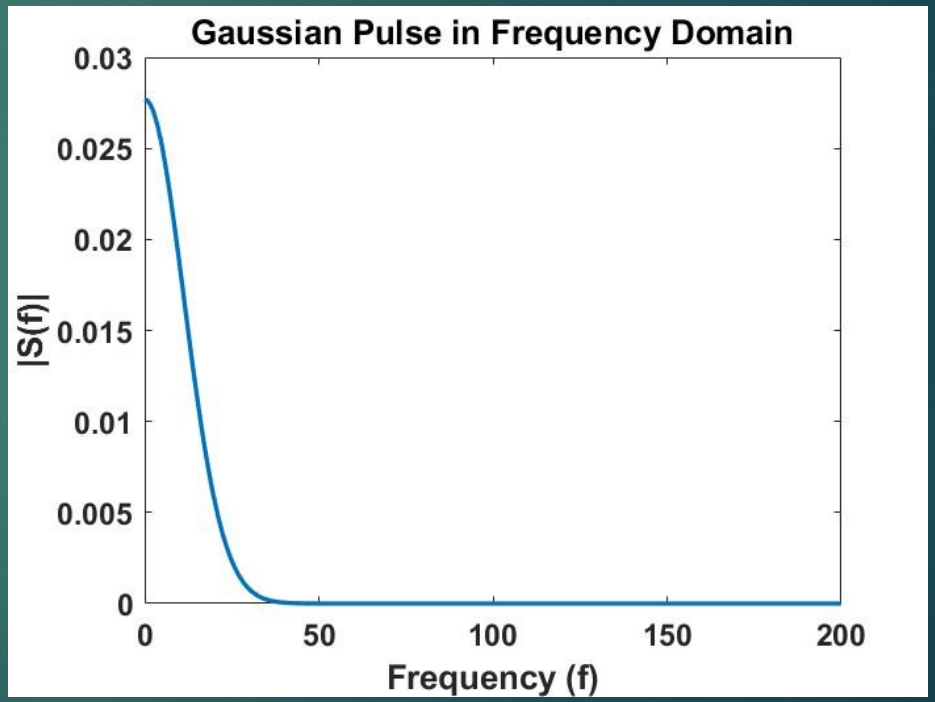
Source: $e^{-(50r)^2}$



Helmholtz equation

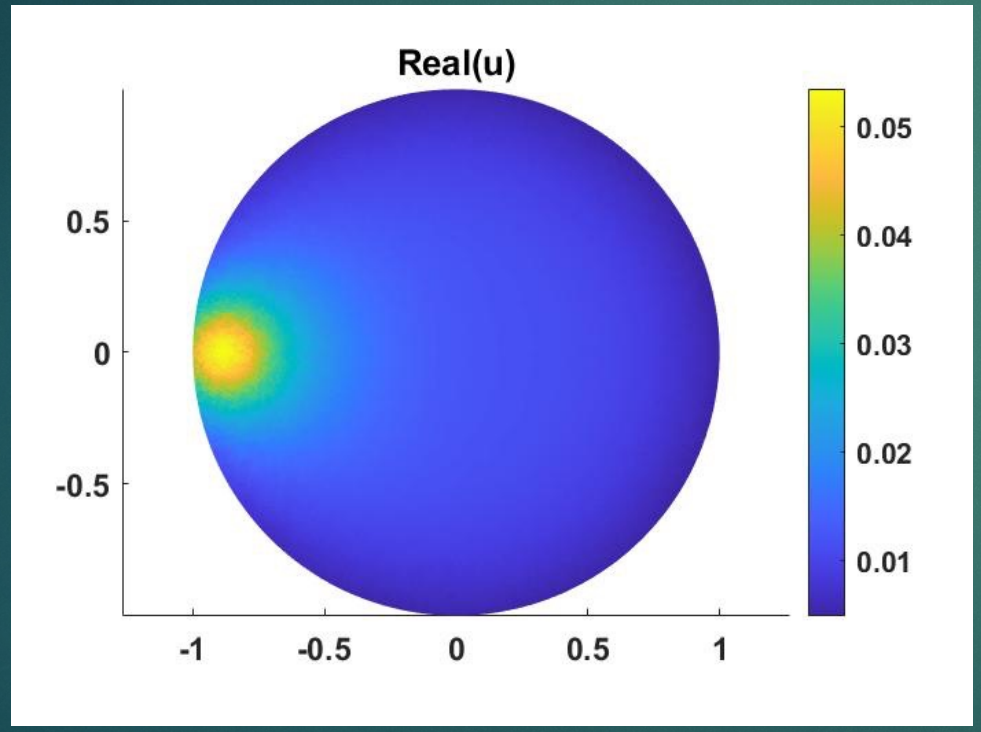
Source: I used the first 58 frequencies.

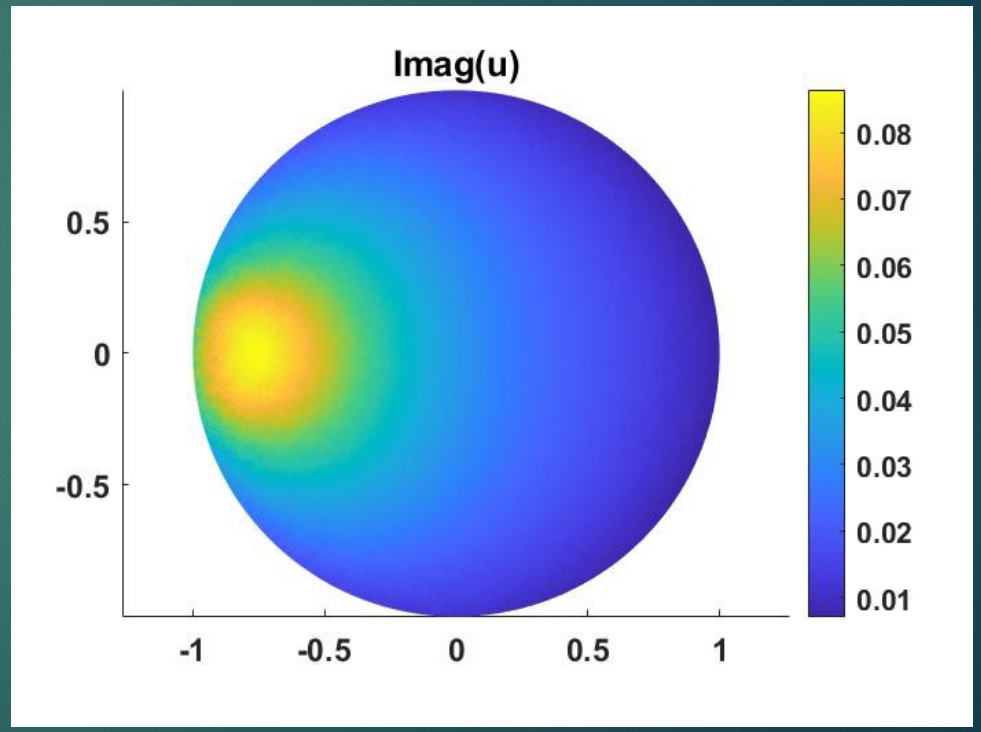




Helmholtz equation

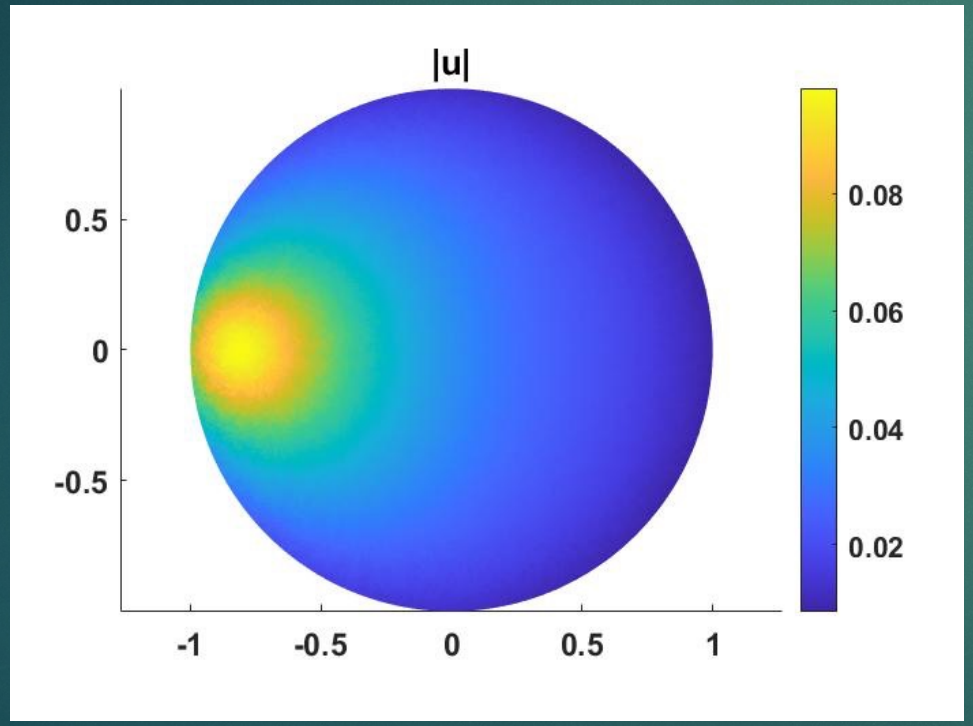
Solution:

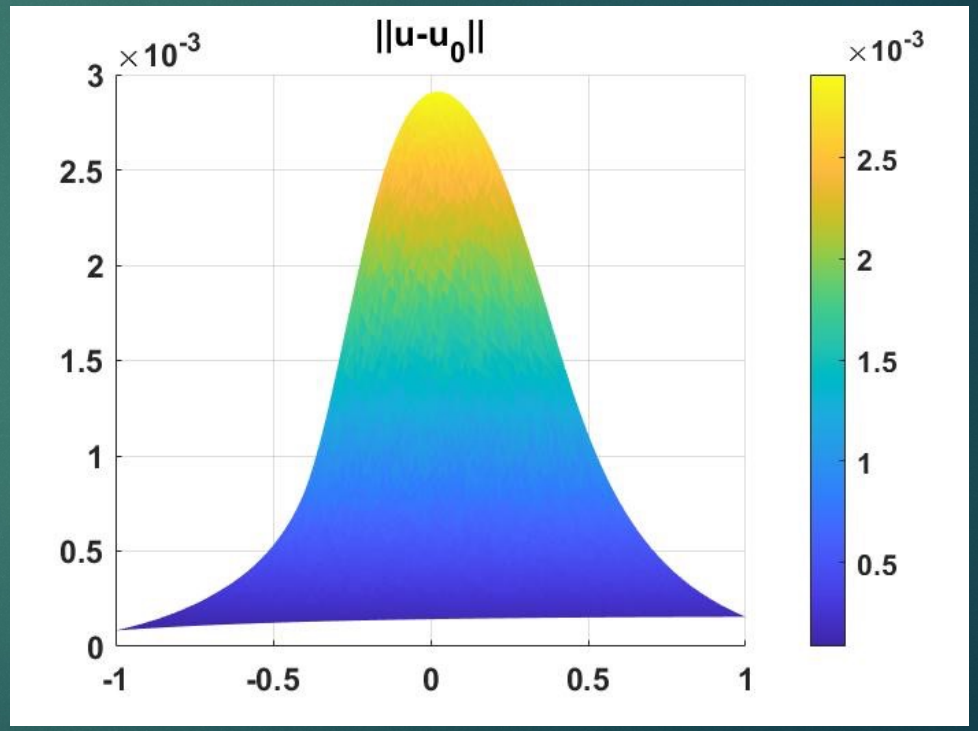




Helmholtz equation

Solution:





Helmholtz equation

The Helmholtz equation:

$$\nabla(D\nabla u) + k1_0^2 u = \mathrm{s} \quad \text{in } \Omega\backslash\mathrm{D}$$

$$\nabla(D\nabla u) + k2_0^2 u = \mathrm{s} \quad \text{in D}$$

$$u_D = u \quad in \ \ \delta D$$

$$\vec{n}_D \cdot \nabla u_D = \vec{n}_D \cdot \nabla u \quad in\ \delta D$$

$$u + \alpha D(\vec{n} \cdot \nabla u) = 0 \quad in\ \delta\Omega$$

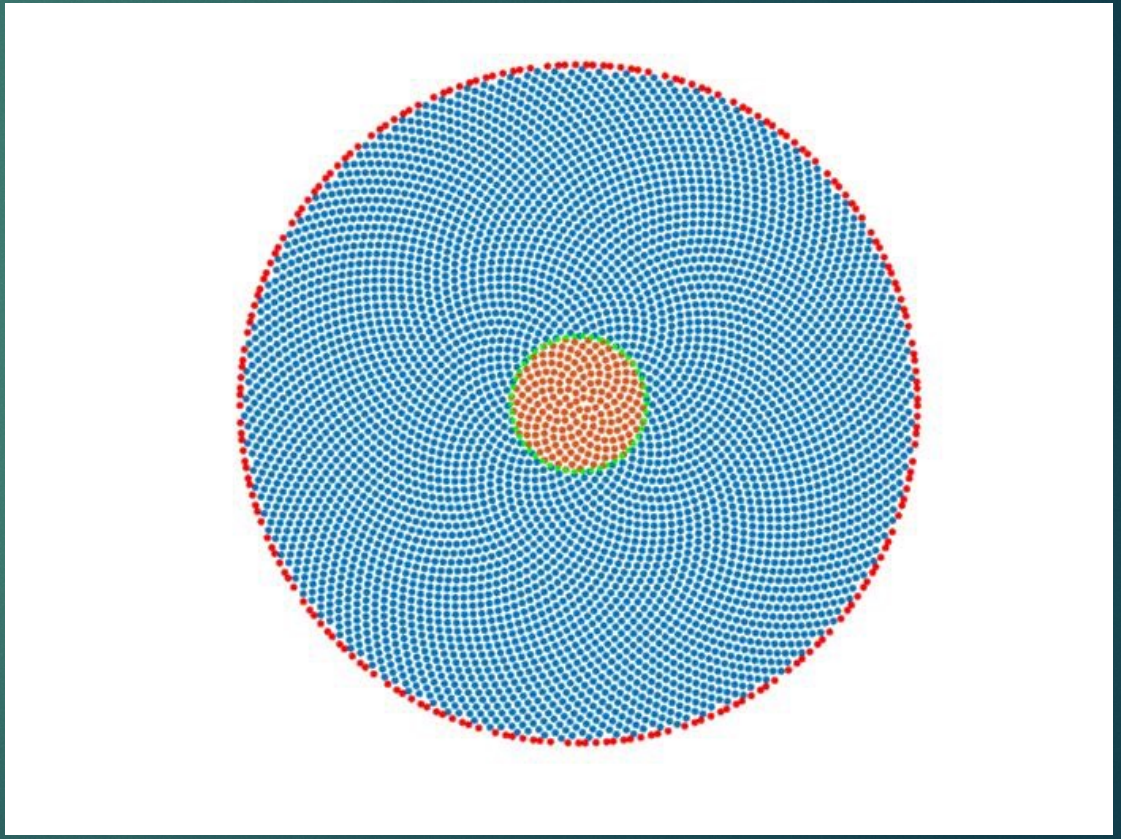
Here $k1_0^2 = -4 + i\frac{\omega}{c0}$ and $k2_0^2 = -12 + i\frac{\omega}{c0}$.

Numerical model

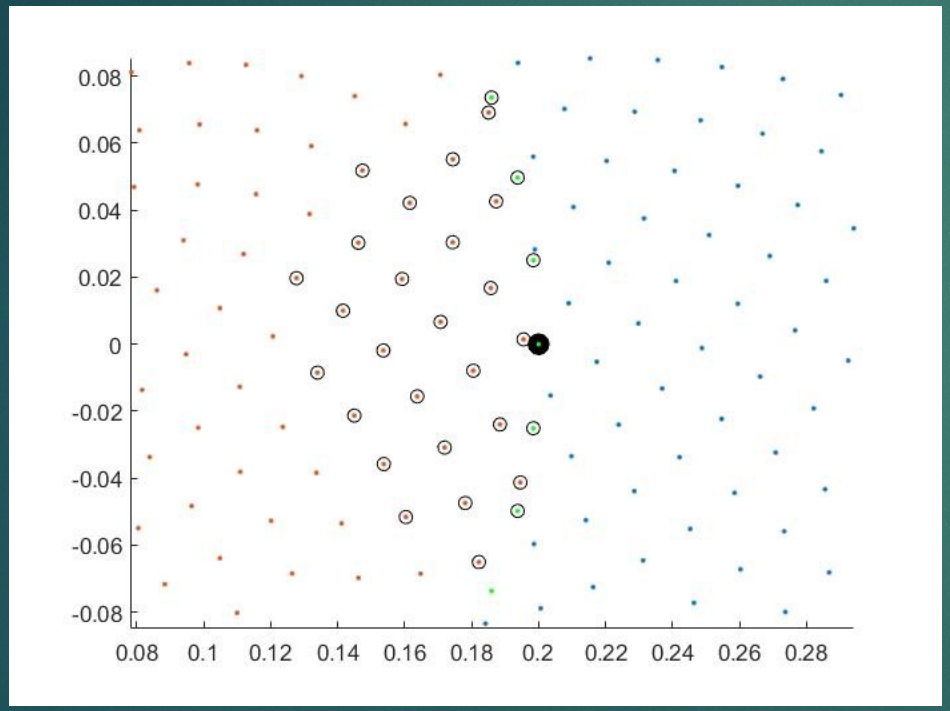
The model consists on 10000 nodes, 300 of them on the outer boundary.

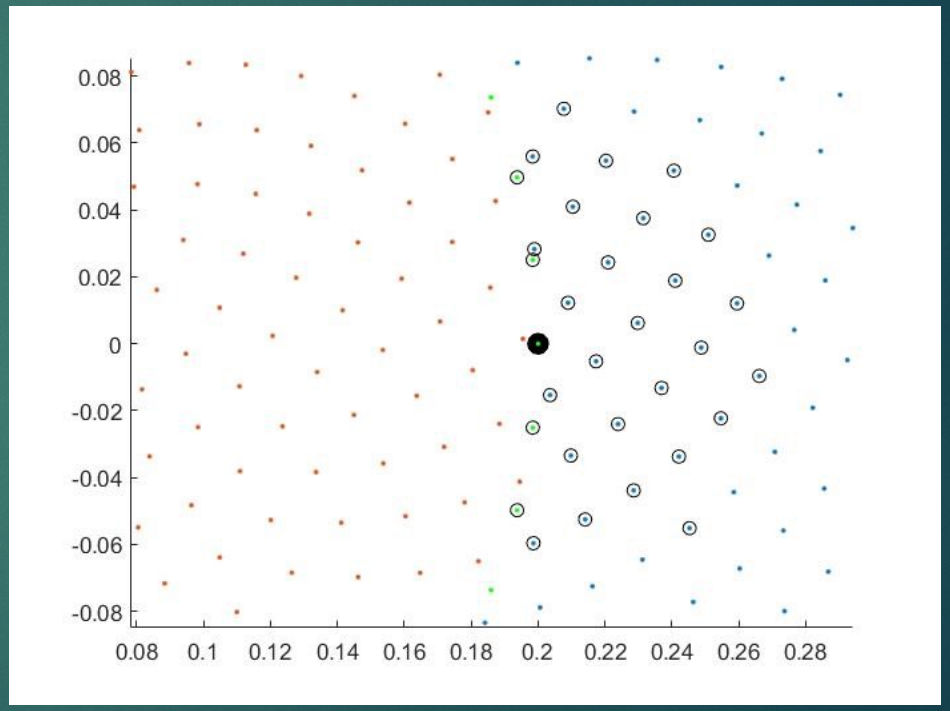
The average distance between nodes is 0.017 .

The boundary of the object has 50 extra nodes which are used twice to have as many equations as unknowns.



Object's boundary conditions



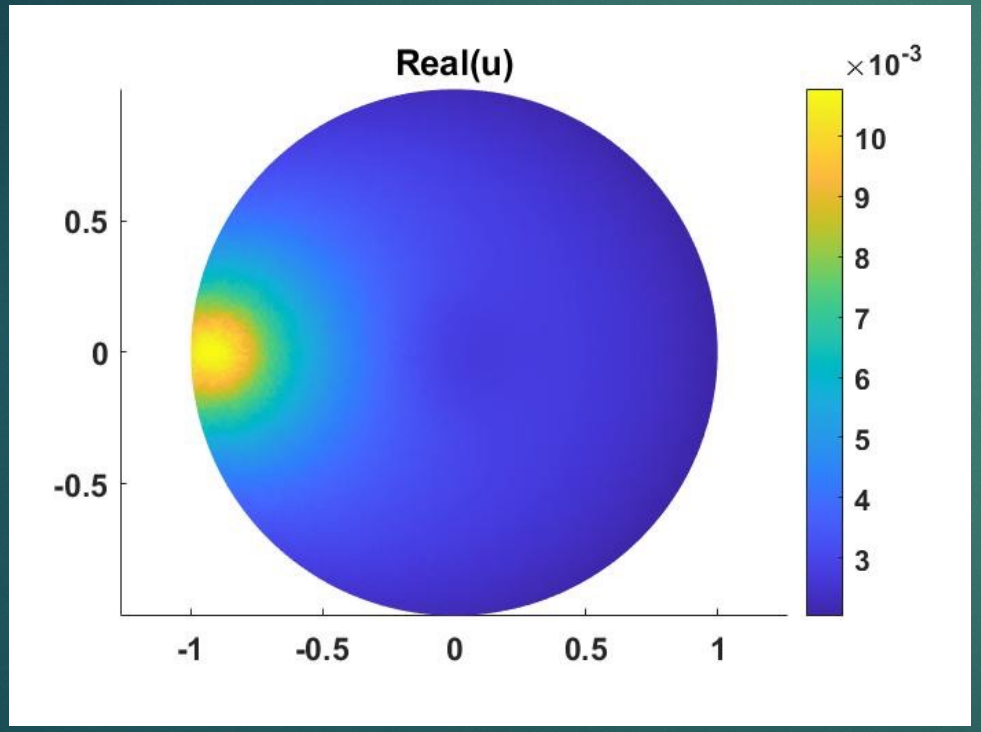


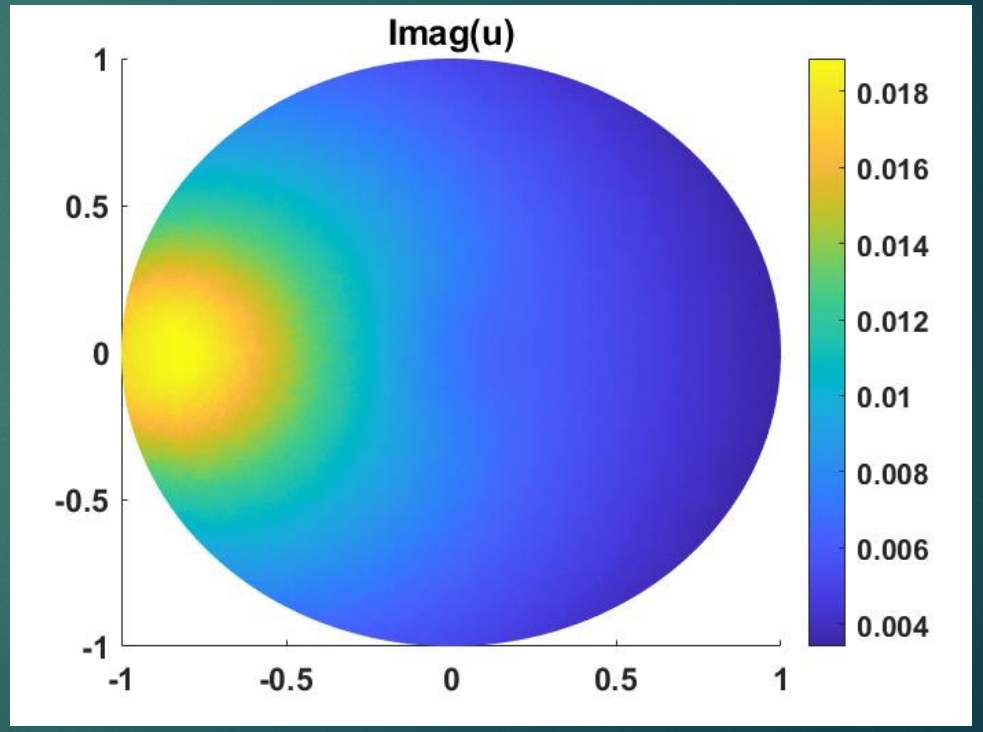
$$\vec{n} \cdot \nabla u_D = \vec{n} \cdot \nabla u$$

$$u_D = u$$

Helmholtz equation

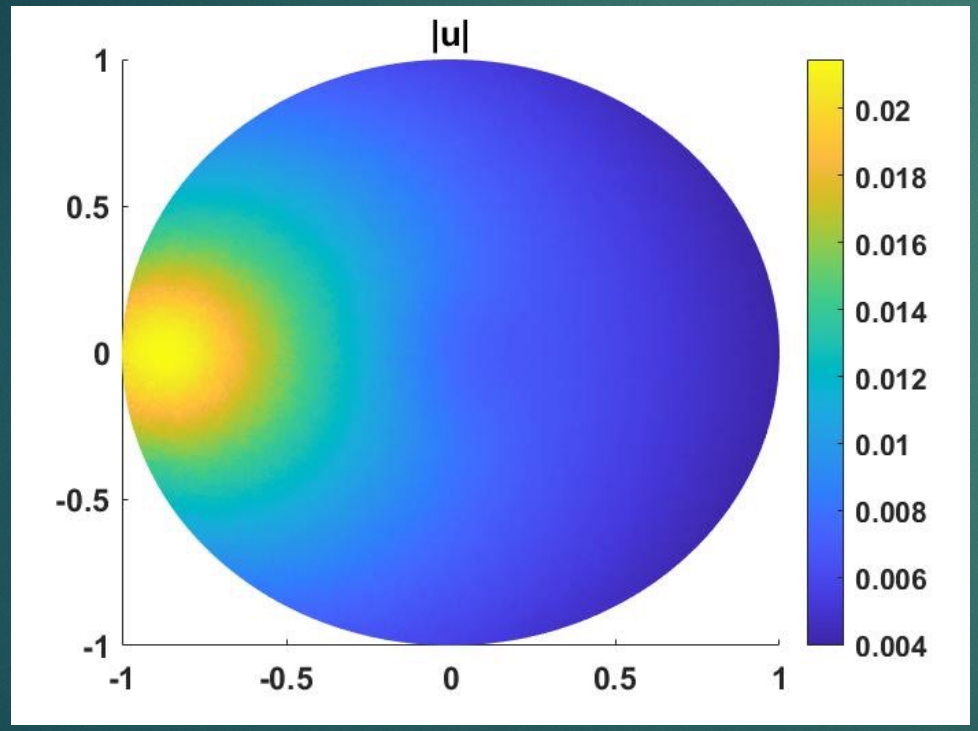
Solution:

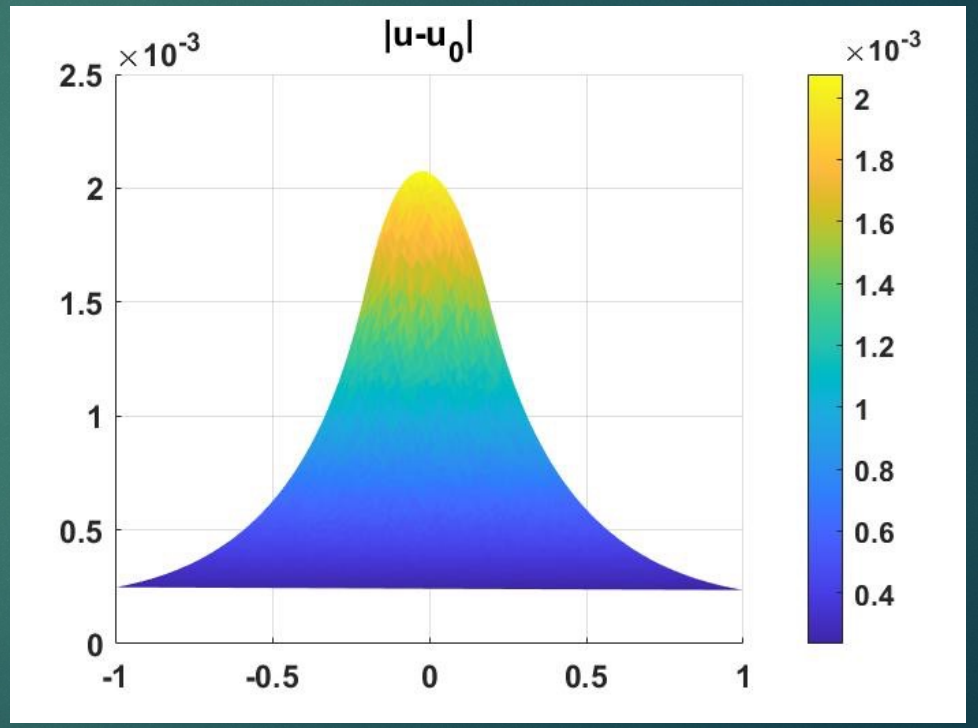




Helmholtz equation

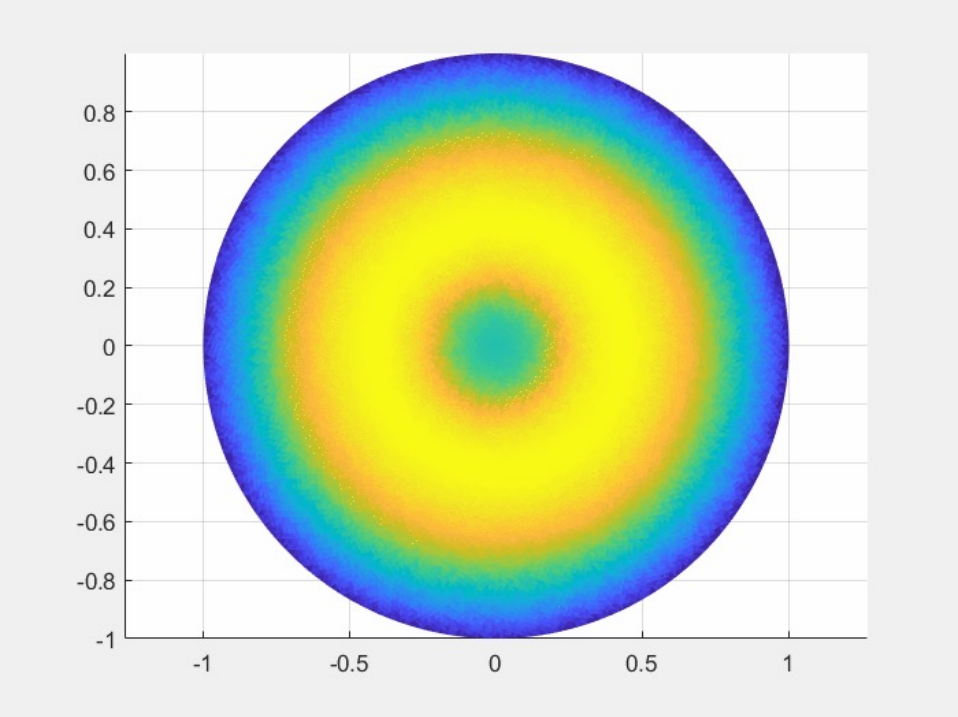
Solution:

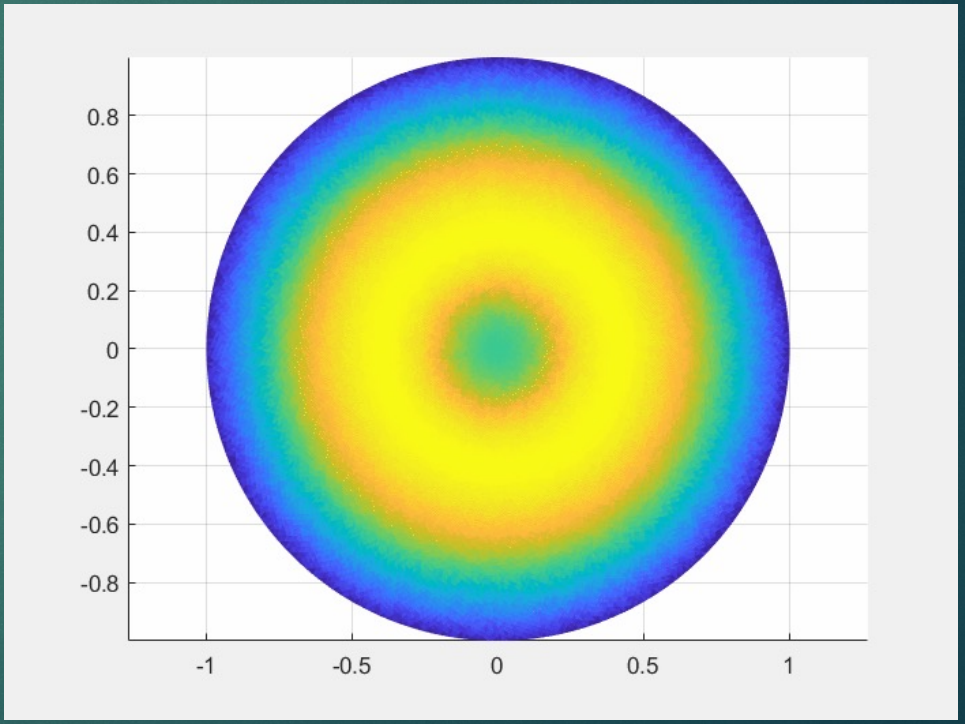




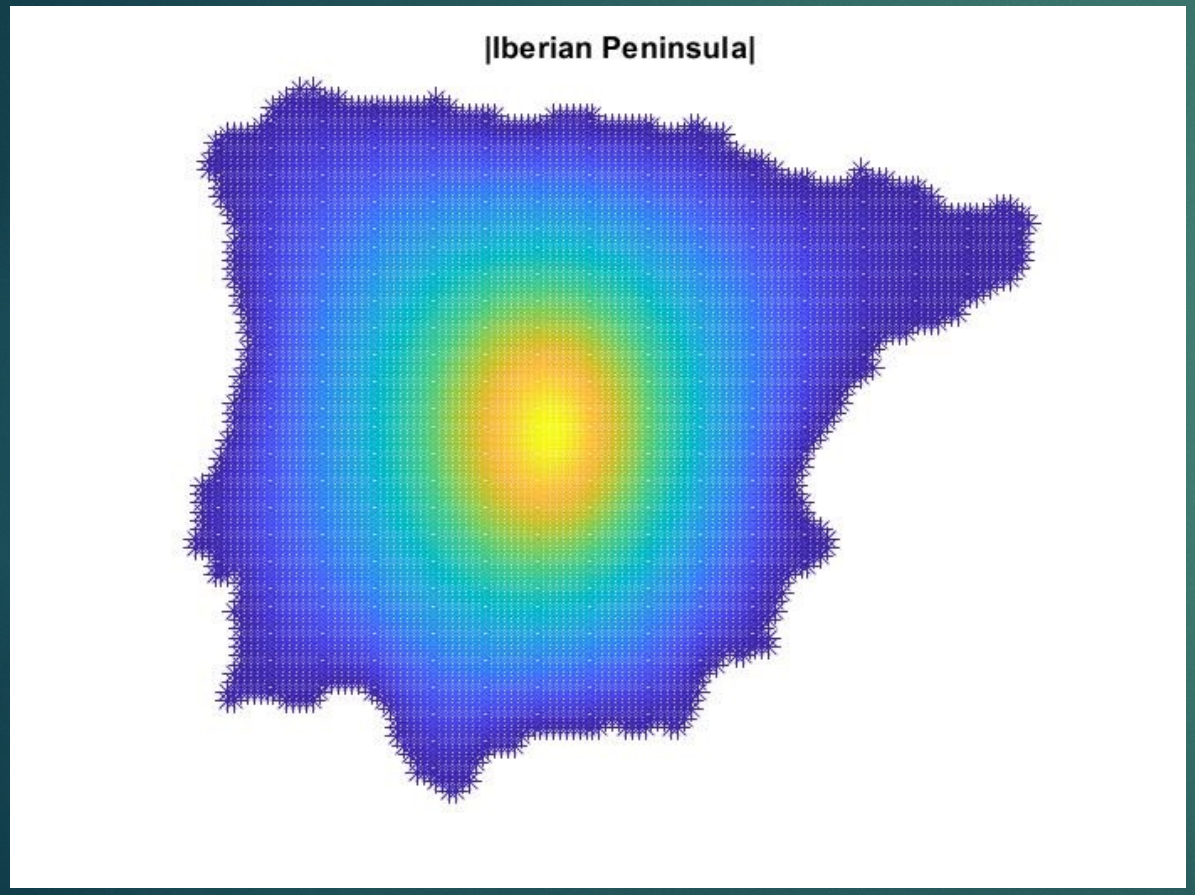
Helmholtz equation

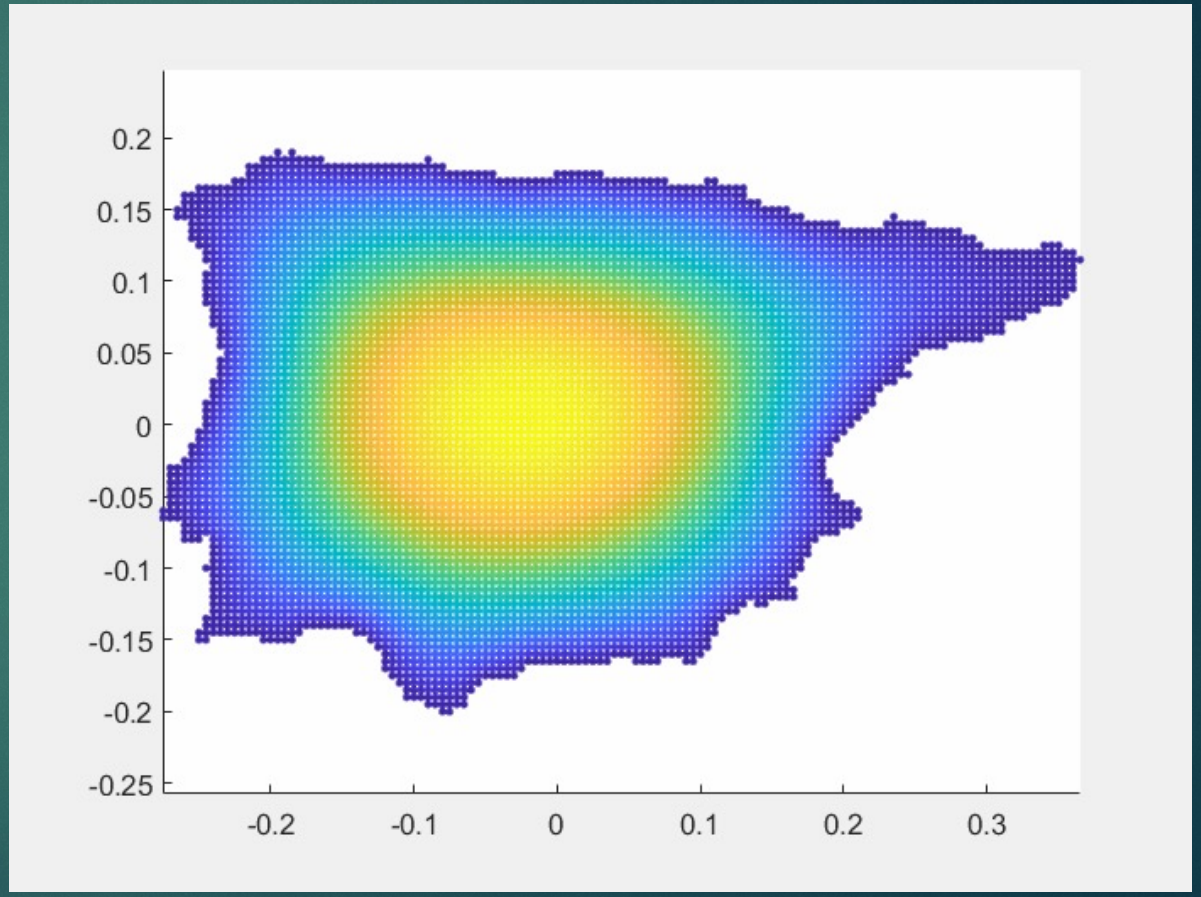
Solution:





Irregular domain





Summary

- We have seen that RBFs can be used as very efficient interpolator.
- They can be used in different ways to solve PDEs.
  - Global
  - RBF-FD
- RBF-FD is an efficient and flexible method to solve PDEs in all kind of domains.

Thank you!