Method and program for the interpolation of experimental results of determining the mechanical properties of mineral composites for modern machine-tools

Mihai Ciupan^{1,*}, Marcel Popa¹, and Emilia Ciupan²

¹Technical University of Cluj-Napoca, Manufacturing Engineering Dept., Cluj-Napoca, Romania

²Technical University of Cluj-Napoca, Management and Economical Engineering Dept., Cluj-Napoca, Romania

Abstract. The paper presents a method of interpolating bifactorial experimental data with the aim of extending the results over the entire domain defined by the independent variables. This approach can be used in determining the optimum mineral casting mixes for a given application. Mechanical properties such as stiffness, ultimate tensile strength, yield strength and Poisson's coefficient can be improved. The paper describes the implementation of a radial basis function (RBF) algorithm in a MATLAB program for the interpolation of data, plotting of the results and the identification of the maximum dependent variable and the independent variables that correspond to it. It also provides the source code for the program along with explanations regarding its use. RBF interpolations can be used successfully in response surface methodology to generate approximations of the studied variables based on the experimental results. The program can be extended to work on any number of input and output variables.

1 Introduction

The author has developed a MATLAB program that creates interpolations using radial basis functions (RBF) in order to extend the information on mineral casting obtained from a limited set of experiments. The aim is to extrapolate this limited data to cover the entire range of the independent variables.

Mineral casting represents a subclass of polymer concrete and is used to create structural elements of modern machine-tools. The polymers commonly used are thermosetting resins, such as epoxy, polyester, polyurethane, which have varying properties that they impart to the final material [1]. The aggregates are generally crushed granite and silica sand. Mineral casting can be recycled by grinding and mixing with new resin and aggregates, minimizing its environmental impact [2]. Its main advantages for these applications are stiffness, yield strength and ultimate tensile strength. Stiffness can be

* Corresponding author: mihai.ciupan@gmail.com

defined by Young's Modulus in tensile testing, yield strength by the stress that creates a plastic deformation of 0.2%, and ultimate tensile strength by the breaking stress.

In the case of mineral casting it is important to maximize stiffness and yield strength. Stiffness is a function of the constituent materials (mineral aggregates and polymer), their shape and positioning. Because the stiffness of the aggregates is significantly higher than that of the polymer, increasing the aggregate content improves the stiffness. Thus, choosing a mineral casting mix involves the following objectives:

- maximizing the volume occupied by the mineral aggregates so as to increase the stiffness and lower the polymer consumption;
- maintaining adequate yield and ultimate tensile strength for a given application by using enough resin;
- improving the workability in order to decrease the cavities and pores of the cast parts and to simplify the manufacturing process.

In the production of mineral casting the aggregates are sieved into different sizes and the main difficulty consists in designing a mix where different percentages of the aggregates combine so as to minimize the space between them and decrease the amount of resin used. All this has to be balanced with an adequate percentage of resin so as to maintain workability and strength.

The paper is based on the response surface methodology applied in paper [3] and uses sets of generated Young's Modulus (E) values as the variable that depends on p and q. The interpolation with RBF functions will be constructed on a set of (p, q, E) points. Variable p represents the percentage of resin mass from the total mass of the mineral casting. Variable q is a parameter in Funk and Dinger's optimization curve involved in calculating the mass percentage of each of the aggregate sizes used in the mineral casting mix [4]. Variable q is a parameter that depends on the shape of the aggregate and is sometimes used based on previous knowledge about the rocks from a certain geographical area. Funk and Dinger's equation is as follows:

$$p(d) = \frac{d^q - d_{\min}^q}{d_{\max}^q - d_{\min}^q} \quad (0),$$

where d = maximum diameter of the particles from the considered size,
 p(d) = percentage of the total mineral casting mass,
 $d_{\min}$ = minimum diameter of the aggregates used,
 $d_{\max}$ = maximum diameter of the aggregates used,
 q = a parameter that accounts for the shape of the mineral particles.

2 Mathematical basis

Article [5] presents a short introduction of the mathematical basis of RBF interpolations. Equation 1 shows a definition RBF function, meaning a function which has as only variable the distance between the point (x,y,z) and another point called a center.

$$z(x, y) = w\phi(\sqrt{(x - x_c)^2 + (y - y_c)^2}) = w\phi(\|x - c\|) = w\phi(r) \quad (1)$$

An example of a radial basis function is "thin-plate spline", presented in equation 2 [5]:

$$\phi(r) = r^2 \log(r) \quad (2)$$

The most common RBF functions used for interpolation are the ones presented in table 1 [6].

Table 1.Radial Basis Functions [6]

Name	Definition
Thin-Plate Spline	$\phi(r) = r^2 \log(r)$
Multiquadric	$\phi(r) = \sqrt{1 + (\varepsilon r)^2}$
Inverse Multiquadric	$\phi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$
Gaussian	$\phi(r) = e^{-(\varepsilon r)^2}$
C ⁰ Matern	$\phi(r) = e^{-\varepsilon r}$
C ² Matern	$\phi(r) = e^{-\varepsilon r} (1 + \varepsilon r)$
C ⁴ Matern	$\phi(r) = e^{-\varepsilon r} (3 + 3\varepsilon r + (\varepsilon r)^2)$

As table 1 shows, most of the functions have a shape parameter, epsilon, which can be assigned a value manually or by using algorithms such as “Leave one out cross validation” (LOOCV), “General cross validation” (GCV), “Maximum Likelihood Estimator” (MLE) [6]. The shape parameter has an important role in minimizing the mean square error between the original function from which the set of interpolation points comes and the actual interpolation. The shape parameter should have a small value, but not generate numerical instability.

Equation 3 [5] describes coordinate z as a sum of RBF functions of the distance between the centers of the points using different weights and a linear polynomial term.

$$z(x,y)=\sum_{i=1}^N w_i \phi(r_i)+a_0+a_1x+a_2y \tag{3}$$

The unknowns of equation 3 are the weights w_i and a₀, a₁, a₂. These can be determined by solving the following system of linear equations [5]:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} & 1 & x_{c,1} & y_{c,1} \\ \vdots & \ddots & \vdots & & \vdots & \\ \phi_{N,1} & \cdots & \phi_{N,N} & 1 & x_{c,N} & y_{c,N} \\ 1 & & & 1 & 0 & 0 \\ x_{c,1} & \cdots & x_{c,N} & 0 & 0 & 0 \\ y_{c,1} & & y_{c,N} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_N \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

$$\text{where } \phi_{i,j} = \phi(\|c_i - c_j\|) \, .$$

Centers c₁, c₂, ..., c_n are the projections of the n points used for constructing the interpolation on the plane (X, Y): (x₁,y₁,0), (x₂,y₂,0), ..., (x_n,y_n,0).

3 The MATLAB Program

The MATLAB program gets points (p, q, E) obtained from the experimental conditions as input which are then used to build the interpolation. The user can choose between to built-in RBF functions, thin-plate spline and multiquadric, or specify a new one. These RBFs were chosen because of the general agreement that these functions give the smallest errors of interpolation. Additionally, a shape parameter must be inputted when selecting the radial basis function “multiquadric”.

The interpolation program is built as a MATLAB project and has a script “main.m” and two functions “createA.m”, respectively “rbf.m”, used to assemble the first matrix in equation 3, respectively to calculate the radial basis function between two points. The following sections show the source code implementation of the algorithm and explanations about their functioning.

3.1 Script “main.m”

Script main represents the main part of the MATLAB interpolation program. The algorithm uses general notation of variables, so it was considered that the points (p, q, E) used for the interpolation are renamed to (x, y, z). The limits of variables p, respectively q are given by l_i , u_i , respectively l_j , u_j . Variables n_i , respectively n_j define the number of nodes of the mesh generate on axes x, respectively y.

Equation 4 is rewritten as:

$$A \cdot b = zend \quad (5)$$

The algorithm assembles matrix A and matrix zend and solves the system of equations to determine the weights of the interpolation function, contained in matrix b.

Next it generates a mesh to cover the entire interpolation domain using the function meshgrid. For the nodes of the mesh which are saved in matrices X and Y (different from lowercase x and y) it generates matrix Z which contains the interpolated values for X and Y. Finally, these are plotted in an orthogonal axes system as a colored mesh, where blue indicates smaller values and yellow indicates higher values. The program outputs to the console the pairs of values (x, y) which give the maximum z of the interpolation. These values are in fact the pairs (p, q) which specify the mix of mineral casting for which the maximum E value is obtained.

clc

%input points used for interpolation z=f(x,y)

```
x = [7;7;8;9;11;11];
y = [34;40;36;37;34;40];
z = [12;24;22;34;15;28];
```

```
A = createA(x,y);
zend = [z;[0;0;0]];
b=A\zend; %solve system of linear equations
```

```
ni = 30;
nj = 30;
li = 7;
ui = 11;
```

```

lj = 34;
uj = 40;

[X,Y] = meshgrid (li:(ui-li)/(ni-1):ui, lj:(uj-lj)/(nj-1):uj);
Z = zeros(nj,ni);
nx = numel(x);

for i=1:nj
    for j=1:ni
        %fprintf('x%f y%f\n', X(i,j), Y(i,j));
        Z(i,j)=0+b(nx+1)+b(nx+2)*X(i,j)+b(nx+3)*Y(i,j);
        for k = 1:nx
            Z(i,j) = Z(i,j)+b(k)*rbf([X(i,j),Y(i,j)], [x(k),y(k)]);
        end
    end
end
mesh(X,Y,Z)
hold on
plot3(x,y,z,'o','Color','r')
xlabel('p[%]')
ylabel('q[*1E-2]')
zlabel('E[MPa]')
hold off

%find maximum and its arguments
maximum = max(max(Z));
[i,j]=find(Z==maximum);
ni = numel(i);

%print maximum and its arguments
fprintf('Function max value of %f is obtained for the following sets of parameters:\n',
maximum);
for k=1:ni
    fprintf('(%f, %f)\n',X(i(k),j(k)),Y(i(k),j(k)));
end

```

3.2 Function “createA.m”

Function createA.m assembles matrix A using the distances between the points and the selected radial basis function. Matrix A is then used when solving equation 5.

```

function A = createA(x,y)
%CREATEA assemble matrix A
% Detailed explanation goes here
n = numel(x);
A = zeros(n+3);
for i=1:n
    for j=1:n
        A(i,j)=rbf([x(i),y(i)], [x(j),y(j)]);
    end
    A(i,n+1)=1;

```

```

    A(i,n+2)=x(i);
    A(i,n+3)=y(i);
end
i=n+1;
for j=1:n
    A(i,j)=1;
end
i=n+2;
for j=1:n
    A(i,j)=x(j);
end
i=n+3;
for j=1:n
    A(i,j)=y(j);
end
end
end

```

3.3 Function “rbf.m”

Function rbf gets points p1 and p2 as input, which represent the centers of two points, it calculates the distance between them and based on this it returns the value of the selected rbf. The following code contains a variable eps, which is not the shape parameter, but MATLAB machine epsilon, which has the role of keeping distance “r” always different than 0.

The code as shown uses the thin-spline RBF, but when needed the multiquadric function can be selected by uncommenting it and commenting the other function. In order to use multiquadric, the shape parameter fshape has to be uncommented and given a value which correspond to the set of points being interpolated. A smaller value of fshape gives an interpolation based more on the local distribution of the initial data set and a greater value more on global distribution of the same data.

```

function rbf = rbf(p1,p2)
%rbf value of RBF between 2 points
% Detailed explanation goes here

%distance between points
r = eps+sqrt((p2(1)-p1(1))^2+(p2(2)-p1(2))^2);

%different RBFs, uncomment the one that suits your case
%comment the others
%RBF thin-plate spline
%RBF multiquadric (needs shape factor)

%thin-plate spline
rbf=r^2*log(r);

%multiquadric
%fshape =2;
%rbf = sqrt(1+(fshape*r)^2);
end

```

4 Examples of generated and interpolated experimental data

The following examples use a limited data set of 9 points, but a higher precision could be gained using a larger number at the expense of using more experimental resources. As the following examples show, interpolation has some advantages:

- There is an interpolated maximum that can be different from the original data set maximum;
- Few experimental resources are used to generate information over a wide range of the independent variables;
- There is at least one area in each interpolation that has values of the stiffness very close to the maximum.

If the interpolation reveals a new maximum value for stiffness, then a new experiment should be conducted in order to validate these approximated results.

4.1 Example 1

The input data are the following sets of (p[%], q[], E[GPa]) data: (7.000000; 0.340000; 12.000000), (7.000000; 0.370000; 31.000000), (7.000000; 0.400000; 24.000000), (9.000000; 0.340000; 15.000000), (9.000000; 0.370000; 37.000000), (9.000000; 0.400000; 16.000000), (11.000000; 0.340000; 10.000000), (11.000000; 0.370000; 22.000000), (11.000000; 0.400000; 17.000000).

This data is interpolated using the thin-plate spline RBF and has a maximum Young’s Modulus of 37.106548 vs. the initial data set maximum of 37.0 GPa.

Fig. 1. Mesh generated from the interpolation of the data set in example 1

Fig. 2. Alternate view of the mesh in figure 1

Console output prints the maximum value of E and its corresponding parameters (p , q):
Function max value of 37.106548 is obtained for the following sets of parameters:
(8.793103, 37.103448).

4.2 Example 2

The input data are the following sets of (p [%], q [], E [GPa]) data: (7.000000; 0.340000; 12.000000), (7.000000; 0.400000; 24.000000), (8.000000; 0.360000; 22.000000), (9.000000; 0.370000; 34.000000), (11.000000; 0.340000; 15.000000), (11.000000; 0.400000; 28.000000).

This data is interpolated using the thin-plate spline RBF and has a maximum Young's Modulus of 35.250716 vs. the initial data set maximum of 34.0 GPa. This is a 3.67% increase in stiffness compared to the maximum obtained from the experiments. We can also see a large area where the values of E differ little from the maximum, but have other p and q parameters which might be more effective in terms of cost.

Fig. 3. Mesh generated from the interpolation of the data set in example 2

Fig. 4. Alternate view of the mesh in figure 3

Console output prints the maximum value of E and its corresponding parameters (p, q):
Function max value of 35.250716 is obtained for the following sets of parameters:
(9.344828, 37.517241)

5 Conclusions

The program developed in MATLAB for the interpolation of experimental data offers numerous advantages:

- Estimation of the mechanical properties of mineral casting mixes over the entire domain of study;
- The maximum points on the plot leads to identifying areas with the greatest probability of containing mineral casting mixes that have superior mechanical properties.

The benefit of using an RBF interpolation of the experimental data is the possibility of identifying new composite mix designs that have superior stiffness. In comparison a linear interpolation of the results wouldn't be able to offer any new mixes that have better properties than the ones already obtained experimentally. Also an approximation using a polynomial function might yield inaccurate results.

The application can be used for other mechanical properties of mineral casting such as ultimate tensile strength, compressive strength, bending strength, yield strength and Poisson's coefficient.

The same program can be applied for the interpolation and plotting of results for any bifactorial experiments by simply inputting other data. An example related to composite materials is the study of the effect of temperature on the mechanical properties of extrusion glass fiber reinforced polyamide composites where the ultimate tensile strength of the extruded part depends on the temperature of the metering zone and the temperature of the forming zone [7]. Paper [7] uses a polynomial approximation function to approximate the values, but it is possible that it could obtain a smaller error using RBF interpolation.

The application can be extended to work with experiments with any number of independent and dependent variables, because the RBF algorithm can operate with any number of input and output variables. If more than two independent variables are used, the

results can be plotted using multivariate data visualization functions offered by MATLAB [8].

References

1. Sabău, E., Bere, P., Hancu, L., Popescu, A., *Materiale Compozite Cu Matrice Polimerică – Lucrări De Laborator* –, Editura U.T.PRESS (2015)
2. Sabău, E., *Recycling of Polymeric Composite Materials*, doi 10.5772/intechopen.81281, Intech Open (2018)
3. Lăzărescu, L., *FE Simulation and Response Surface Methodology for Optimization of Tube Bending Process*, Annals of Dunărea de Jos University of Galați ISSN 1221-4566. 2. 98. (2010)
4. Fennis, S., Walraven, J. C., *Using particle packing technology for sustainable concrete mixture design* (2012)
5. Frei, W., *Using Radial Basis Functions for Surface Interpolation*, Comsol Blog, (2016)
6. Mongillo, Michael A., *Choosing Basis Functions and Shape Parameters for Radial Basis Function Methods*, SIAM Undergraduate Research Online. 4. 10.1137/11S010840 (2011)
7. Popescu, A, Hancu, L., Sabău, E., *Effect Of Temperature On The Mechanical Properties Of Extrusion Glass Fiber Reinforced Polyamide 6.6 Composites (GFRPA 6.6)*, Acta Technica Napocensis (2018)
8. *Visualizing Multivariate Data*, <https://www.mathworks.com/help/stats/examples/visualizing-multivariate-data.html>