THE ANNALS OF “DUNAREA DE JOS” UNIVERSITY OF GALATI
FASCICLE IX. METALLURGY AND MATERIALS SCIENCE
Nº. 2 - 2022, ISSN 2668-4748; e-ISSN 2668-4756
Article DOI: https://doi.org/10.35219/mms.2022.2.04

MATLAB PROGRAM FOR DETERMINING THE INERTIA CHARACTERISTICS OF FLAT SURFACES WITH MONTE CARLO ALGORITHMS

Marius BOTIȘ, Costel PLEȘCAN*
Transilvania University of Braşov, Department of Civil Engineering, Romania
e-mail: *plescan.costel@unitbv.ro

ABSTRACT

This paper presents a Matlab program for calculating the inertia tensor for complex plane surfaces. The calculation of the moments of inertia for plane surfaces with classical methods involves decomposing the surfaces into primitive surfaces and applying Steiner's relations. The classical methods are based on the knowledge of the analytical determined moments of inertia for primitive surfaces. In the case of complex surfaces, numerical methods can be used which are based on discretizing the surface into triangles and determining the moments of inertia by applying Steiner's relations knowing the analytical moments of inertia for a triangle. Both methods are computationally intensive and are basically based on the analytical moments of inertia of an elementary surface. In the case of large and complex surfaces the Monte Carlo algorithm can be used, which is a probabilistic algorithm based on the generation of area elements within the surface for which the moments of inertia are determined and then summed over the entire area bounded by the surface. The paper presents the Matlab calculation program and application examples for the use of the probabilistic Monte Carlo algorithm.

KEYWORDS: Monte Carlo method, moments of inertia of flat surfaces, probabilistic algorithms

1. Introduction

In order to determine how to calculate moments of inertia for plane surfaces with the Monte Carlo method, we will start with the calculation of moments of inertia of a concentrated mass system. If we consider n concentrated masses into which a surface decomposes, then the moments of inertia [1] of that surface with respect to the x and y axes of a coordinate system attached to the surface are:

Axial moments of inertia:

$$J_{xx} = \int_A y^2 dm = \sum_{i=1}^{n} m_i y_i^2 \,. \tag{1}$$

$$J_{yy} = \int_A x^2 dm = \sum_{i=1}^{n} m_i x_i^2 \tag{2}$$

Centrifugal moment of inertia:

$$J_{xy} = \int_A xy dm = \sum_{i=1}^{n} m_i x_i y_i \tag{3}$$

Polar moment of inertia:

$$J_p = \sum_{i=1}^{n} m_i y_i^2 + \sum_{i=1}^{n} m_i x_i^2 \tag{4}$$

The tensor of mechanical moments of inertia in matrix form becomes:

$$[J] = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} m_i y_i^2 & -\sum_{i=1}^{n} m_i x_i y_i \\ -\sum_{i=1}^{n} m_i x_i y_i & \sum_{i=1}^{n} m_i x_i^2 \end{bmatrix} \tag{5}$$

Similarly, the tensor of geometric moments of inertia becomes:

THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE IX. METALLURGY AND MATERIALS SCIENCE
Nº. 2 - 2022, ISSN 2668-4748; e-ISSN 2668-4756
Article DOI: https://doi.org/10.35219/mms.2022.2.04

$$[J] = \begin{bmatrix} \sum_{i=1}^{n} m_i y_i^2 & -\sum_{i=1}^{n} m_i x_i y_i \\ -\sum_{i=1}^{n} m_i x_i y_i & \sum_{i=1}^{n} m_i x_i^2 \end{bmatrix} = \rho t \begin{bmatrix} \sum_{i=1}^{n} A_i y_i^2 & -\sum_{i=1}^{n} A_i x_i y_i \\ -\sum_{i=1}^{n} A_i x_i y_i & \sum_{i=1}^{n} A_i x_i^2 \end{bmatrix} \quad (6)$$

In relation (7) the connection between the tensor of geometrical moments of inertia and the tensor of the mechanical moments of inertia is presented.

$$[J] = \rho t [I] \quad (7)$$

where:

- ρ is the material density of the surface;
- t is the thickness of the surface.

From relations (6) and (7) it can be noted that knowing the geometric inertia tensor of the surface allows the determination of the mechanical inertia tensor if the thickness and material density of the surface are known.

2. The algorithm for calculating the inertia tensor with the Monte Carlo method

To determine the inertia tensor with the Monte Carlo method for plane surfaces, the surface is decomposed into a system of concentrated masses for which the moments of inertia can be determined with relations (1)-(4). The mass elements dm are obtained by dividing the surface mass M by the number of probabilistically generated elements N according to relation (8).

$$dm = \frac{M}{N} \quad (8)$$

N is the number of probabilistically generated points in the surface domain for which the moments of inertia are determined; -M the total mass of the surface. The higher the number of points generated on the surface domain, the higher the probability of tending to the exact solution. Using the Monte Carlo algorithm, one can determine both the geometric and the mechanical inertia tensor with the relationships shown below. Mechanical moment of inertia tensor:

$$[J] = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{N} dm y_i^2 & -\sum_{i=1}^{N} dm x_i y_i \\ -\sum_{i=1}^{N} dm x_i y_i & \sum_{i=1}^{N} dm x_i^2 \end{bmatrix}; dm = \frac{M}{N}; (x_i\ y_i) \epsilon A \quad (9)$$

Tensor of geometric moments of inertia:

$$[I] = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} dA y_i^2 & -\sum_{i=1}^{N} dA\, y_i \\ -\sum_{i=1}^{N} dA x_i y_i & \sum_{i=1}^{N} dA x_i^2 \end{bmatrix}; dA = \frac{A}{N}; (x_i\ y_i) \epsilon A \quad (10)$$

where:

- A is the area of the surface;
- x_i, y_i are the coordinates of the concentrated masses dm.

3. Presentation of Matlab programs for the calculation of the inertia tensor on plane surfaces with the probabilistic Monte Carlo method

In order to validate the probabilistic Monte Carlo algorithm for the calculation of the inertia tensor in the case of flat surfaces, the implementation of the algorithm in Matlab code is presented below. The inertia tensor is calculated for a square and a triangle. In both cases the moments of inertia are determined analytically and probabilistically for comparison.

3.1. The implementation of the Monte Carlo algorithm in MATLAB program in the case of a square

Matlab code implementation of the Monte Carlo method [2] for determining the moments of inertia for a square of mass M, the axial moments of inertia and the centrifugal moment of inertia for a square have the analytical expressions below:

Axial moments of inertia:

$$I_{yy} = I_{xx} = \frac{1}{3} M a^2 \quad (11)$$

THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE IX. METALLURGY AND MATERIALS SCIENCE
Nº. 2 - 2022, ISSN 2668-4748; e-ISSN 2668-4756
Article DOI: https://doi.org/10.35219/mms.2022.2.04

Centrifugal moment of inertia:

$$I_{xy} = -\frac{1}{4}Ma^2 \quad (12)$$

The Matlab program [3] for determining the inertial characteristics for the square with the probabilistic Monte Carlo method is presented below:

```
clear;
%Calculation of moments of inertia for a square
M=1;% mass of square
a=1;% the square side where infinitesimal masses are randomly generated
N=3000;%number of infinitesimal masses
dm=M/N;% infinitesimal mass element
Ixx=0;
Iyy=0;
Ixy=0;
n=0;%initializations
while n<N
x=a*rand;
y=a*rand;
scatter(x,y,40,'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor',[1 0 0],...
    'LineWidth',2)
hold on;
rtemp=[x y];
%Theoretical values of moment of inertia
Ixx=Ixx+dm*(y^2);
Iyy=Iyy+dm*(x^2);
Ixy=Ixy-dm*(x*y);
n=n+1;
end
hold off;
Ixxt=1/3*M*a^2;
Iyyt=1/3*M*a^2;
Ixyt=-M*a^2/4;
disp('Theoretical value of moment of inertia Ixxt =');
disp(Ixxt);
disp('Value moment of inertia with Monte Carlo Ixx=');
disp(Ixx);
disp('Theoretical value of moment of inertia Iyyt =');
disp(Iyyt);
disp('Value moment of inertia with Monte Carlo Iyy=');
disp(Iyy);
disp('Theoretical value of moment of inertia Ixyt =');
```

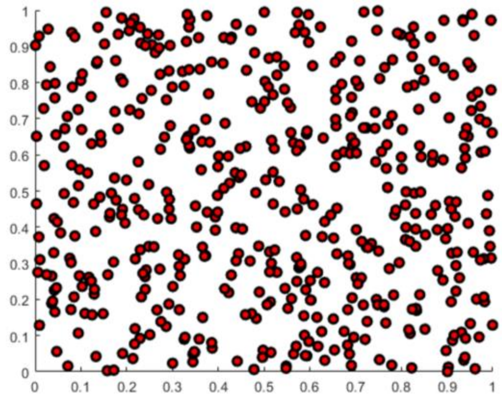

Fig. 1. *Generating 500 points randomly*

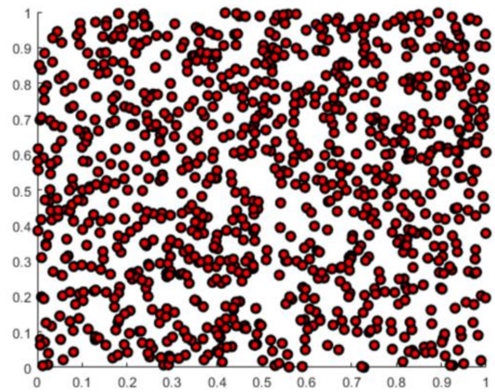

Fig. 2. *Generating 1000 points randomly*

Following the analysis, the results presented in Fig. 1 and Fig. 2 were obtained, which are compared with the analytical results obtained.

For 500 randomly generated points Fig. 1, the theoretical and probabilistic results are:

Theoretical value of moment of inertia Ixxt = 0.3333

Value moment of inertia with Monte Carlo Ixx = 0.3233

Theoretical value of moment of inertia Iyyt = 0.3333

Value moment of inertia with Monte Carlo Iyy = 0.3466

Theoretical value of the moment of inertia Ixyt = -0.2500

Value moment of inertia with Monte Carlo Ixy = -0.2480

For 1000 randomly generated points Fig. 2, the theoretical and probabilistic results are

Theoretical value of moment of inertia Ixxt = 0.3333

Value moment of inertia with Monte Carlo Ixx = 0.3283

Theoretical value of moment of inertia Iyyt = 0.3333

Nº. 2 - 2022, ISSN 2668-4748; e-ISSN 2668-4756
Article DOI: https://doi.org/10.35219/mms.2022.2.04

Value moment of inertia with Monte Carlo Iyy = 0.3247

Theoretical value of the moment of inertia Ixyt = -0.2500

Value moment of inertia with Monte Carlo Ixy = -0.2407

3.2. Implementation of Monte Carlo algorithm in MATLAB program in the case of a triangle

In the following, it is presented the MATLAB implementation of the Monte Carlo method [2] for determining the moments of inertia for an isosceles triangle of mass M. The axial moments of inertia and the centrifugal moment of inertia have the analytical expressions below:

Axial moments of inertia:

$$I_{yy} = I_{xx} = \frac{1}{6} M a^2 \tag{13}$$

Centrifugal moment of inertia:

$$I_{xy} = -\frac{1}{12} M a^2 \tag{14}$$

The Matlab program [3] for determining the inertial characteristics for the triangle with the probabilistic Monte Carlo method is presented below:

```
clear;
%Calculation of moments of inertia for a triangle
M=1;% mass of triangle
a=1;% the square side where infinitesimal masses are randomly generated
N=1000;%number of infinitesimal masses
dm=M/N;% infinitesimal mass element
Ixx=0;Iyy=0;Ixy=0;n=0;%initializations
while n<N
x=a*rand;y=a*rand;
rtemp=[x y ];
if (rtemp(:,1)+rtemp(:,2)<1)
scatter(x,y,40,'MarkerEdgeColor',[0 0 0],...
        'MarkerFaceColor',[1 0 0],...
        'LineWidth',2)
hold on;
Ixx=Ixx+dm*(y^2);Iyy=Iyy+dm*(x^2);Ixy=Ixy-dm*(x*y);
n=n+1;
end
end
hold off;
Ixxt=1/6*M*a^2;Iyyt=1/6*M*a^2;Ixyt=-M*a^2/12;
disp('Theoretical value of moment of inertia Ixxt =');
disp(Ixxt);
disp('Value moment of inertia with Monte Carlo Ixx=');
disp(Ixx);
disp('Theoretical value of moment of inertia Iyyt =');
disp(Iyyt);
disp('Value moment of inertia with Monte Carlo Iyy=');
disp(Iyy);
disp('Theoretical value of moment of inertia Ixyt =');
disp(Ixyt);
disp('Value moment of inertia with Monte Carlo Ixy=');
disp(Ixy);
```

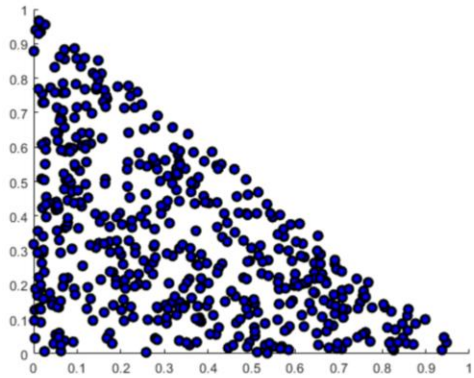

Fig. 3. *Generating 500 points randomly*

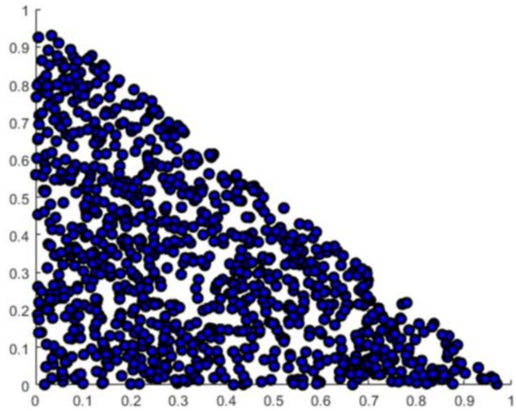

Fig. 4. *Generating 500 points randomly*

For 500 randomly generated points in Fig. 3, the theoretical and probabilistic results are:

Theoretical value of moment of inertia Ixxt = 0.1667

Value moment of inertia with Monte Carlo Ixx = 0.1635

Theoretical value of moment of inertia Iyyt = 0.1667

value moment of inertia with Monte Carlo Iyy = 0.1745

Theoretical value of moment of inertia Ixyt = -0.0833

Value moment of inertia with Monte Carlo Ixy = -0.0812

For 1000 randomly generated points Fig. 4, the theoretical and probabilistic results are:

Theoretical value of moment of inertia Ixxt = 0.1667

Value moment of inertia with Monte Carlo Ixx = 0.1619

Theoretical value of moment of inertia Iyyt = 0.1667

value moment of inertia with Monte Carlo Iyy = 0.1619

Theoretical value of moment of inertia Ixyt = -0.0833

Value moment of inertia with Monte Carlo Ixy = -0.0813

4. Results and conclusions

The results of the analysis showed:

- The Monte Carlo algorithm has a probabilistic character due to the random generation of elementary dm masses;
- The results obtained depend on the number of dm masses in the surface domain for which the inertia tensor is determined;
- The presented method can be applied in the case of large and complex surfaces where classical analytical methods or numerical methods require a very large volume of calculations.
- Results obtained with Matlab [3] have been validated for square and triangle surfaces.

References

[1]. Billingsley J., *Essentials of Dynamics and Vibration*, In: Publishing House Springer, 2018.

[2]. Barbu A., Zhu S., *Monte Carlo Methods*, In: Publishing House Springer, 2020.

[3]. ***, https://www.mathworks.com/products/matlab.html, Accessed: 22.00.2021.