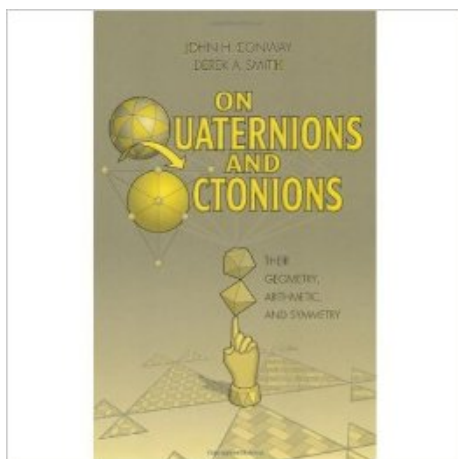


Scripta-Ingenia

The Quaternion Algebra



Conway, John H., and Derek A. Smith, On quaternions and octonions, *AMC 10 (2003)*: 12.

product on \mathbb{R}^4 is then the \mathbb{R} -bilinear product $\mathbb{R}^4 \times \mathbb{R}^4 \rightarrow \mathbb{R}^4$; $(a, b) \mapsto ab$ with unity 1, defined by the formulae $i^2 = j^2 = k^2 = -1$ and $ij = k = -ji$, $jk = i = -kj$ and $ki = j = -ik$. Although the quaternion product is associative, it is not commutative, for example $ji \neq ij$.

More interesting properties on the subject can be found in the book *Topological Geometry*, by Ian R. Porteus (1969). For example, Proposition 10.22 of that book states an important application for the quaternion product: Each rotation of \mathbb{R}^3 is of the form ρ_q for some non-zero quaternion q , and every such map is a rotation of \mathbb{R}^3 . The map ρ_q associates to every pure quaternion x the pure quaternion qxq^{-1} . This means that if the vector v_{rot} is obtained by rotating the vector v through an angle of θ around the axis defined by a unit vector u then $v_{rot} = qvq^{-1}$, with u and v considered as pure quaternions and $q = \cos \frac{\theta}{2} + u \sin \frac{\theta}{2}$.

The next step in the ladder of Cayley algebras is the octonion algebra which has 8 dimensions. A book that investigates quaternions and octonions, is the book by Conway and Derek, whose cover is displayed in the picture above. This book investigates the geometry of quaternion and octonion algebras. Following a comprehensive historical introduction, the book illuminates the special properties of 3- and 4-dimensional Euclidean spaces using quaternions, leading to enumerations of the corresponding finite groups of symmetries. The second half of the book discusses the less familiar octonion algebra, concentrating on its remarkable *triality symmetry* after an appropriate study of Moufang loops. The authors also describe the arithmetics of the quaternions and octonions. The book concludes with a new theory of octonion factorization. Topics covered include the geometry of complex numbers, quaternions and 3-dimensional groups, quaternions and 4-dimensional groups, Hurwitz integral quaternions, composition algebras, Moufang loops, octonions and 8-dimensional geometry, integral octonions, and the octonion projective plane.

Certain real and complex algebras arise naturally in the detailed study of the groups of orthogonal transformations on finite-dimensional real and complex vector spaces. These are the Clifford algebras. Examples of such algebras are \mathbb{R} itself, the real algebra of complex numbers \mathbb{C} and the real algebra of quaternions \mathbb{H} . The letter H is the initial letter of the surname of Sir William Hamilton, who first studied quaternions and gave them their name.

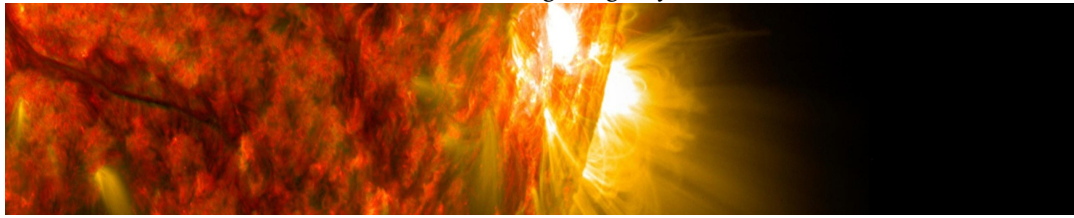
Quaternions find uses in both theoretical and applied mathematics, in particular for calculations involving three-dimensional rotations such as in three-dimensional computer graphics, computer vision and crystallographic texture analysis.

As a set \mathbb{H} is equal to \mathbb{R}^4 and it is equipped with the usual addition and scalar multiplication. Let 1, i , j and k denote the elements of the standard basis for \mathbb{R}^4 . The *quaternion*

A Scripta-Ingenia assume-se como uma revista de divulgação científica tratando temas da ciência e da tecnologia, cobrindo todas as áreas do saber no domínio das ciências exactas ou aplicadas. Interessa-se ainda por artigos de opinião, sobre tópicos científicos ou não, desde que escritos por autores na área das ciências e da engenharia, e que reflitam as suas opiniões enquanto membros dessa comunidade. Este é o seu número quatro e corresponde ao Solstício de Verão do ano 2015.

Chief Editor — Nelson Martins-Ferreira
CDRSP-ESTG, IPLeiria

NASA's breathtaking images of the sun.



The following is an excerpt from the introduction of the book *Topological Geometry*, by Ian R. Porteus (1969), where the author gives an illustration of the unit sphere in \mathbb{R}^4 , with its antipodal points identified, relating it with the group of rotations of \mathbb{R}^3 while explaining the analogy with the soup plate trick.

Almost the simplest example of all is the unit circle, S^1 , in the plane \mathbb{R}^2 . This is a smooth curve. It also has a group structure, if one interprets its points as complex numbers of absolute value 1, the group product being multiplication. This group may be identified in an obvious way with the group of rotations of \mathbb{R}^2 , or indeed of S^1 itself, about the origin.

What about \mathbb{R}^3 ? The situation is now more complicated, but the ingredients are analogous. The complex numbers are replaced, not by a three-dimensional, but by a four-dimensional algebra called the quaternion algebra and identifiable with \mathbb{R}^4 just as the complex algebra is identifiable with \mathbb{R}^2 , and the circle group replaced by the group of quaternions of absolute value 1, this being identifiable with the unit sphere, S^3 , in \mathbb{R}^4 , the set of points in \mathbb{R}^4 at unit distance from 0. As for the group of rotations of \mathbb{R}^3 , this turns out to be identifiable not with S^3 but with the space obtained by identifying each point of the sphere with the point antipodal to it.

An example of how this model of the group of rotations of \mathbb{R}^3 can be used is the following.

Suppose one rotates a solid body continuously about a point. Then the axial rotation required to get from the initial position of the body to the position at any given moment will vary continuously. The initial position may be represented on S^3 by one of the points representing the identity rotation, say the real quaternion 1 which we may think of as the North pole of S^3 . The subsequent motion of the body may then be represented by a continuous path on the sphere. What one can show is that after a rotation through an angle 2π about any axis one arrives at the South pole, the real quaternion -1 . After a further full rotation about the same axis one arrives back at 1. There are various vivid illustrations of this, one of the simplest being the soup plate trick, in which the performer rotates a soup plate lying horizontally on his hand through an angle 2π about the vertical line through its centre. His arm is then necessarily somewhat twisted. A further rotation of the plate through 2π about the vertical axis surprisingly brings the arm back to the initial position. The twisting of the arm at any point in time provides a record of a possible path of the plate from its initial to its new position, this being recorded on the sphere by a path from the initial point to the new point. As the arm twists, so the path varies continuously. The reason why it is possible for the arm to return to the initial position after a rotation of the hand through 4π is that it is possible to deform the path of the actual rotation, namely a great circle on S^3 , continuously on the sphere to a point. This is not possible in the two-dimensional analogue when the group of rotations is a circle. A great circle on S^1 is necessarily S^1 itself, and this is not deformable continuously on itself to a point.

Modelação do pêndulo caótico - Parte 1

by DIOGO NASCIMENTO BAPTISTA^{a,b,c}, ALEXANDRA NASCIMENTO BAPTISTA^{a,b,c}, HUGO COSTELHA^{d,e}

^a Departamento de Matemática,
Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Leiria, Leiria, Portugal
^b Ciêncializar,
Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Leiria, Leiria, Portugal
^c CIMA
^d INESC TEC
^e Departamento de Engenharia Eletrotécnica,
Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Leiria, Leiria, Portugal

Resumo Um pêndulo físico, sujeito a um amortecimento e a uma força externa que atua sobre o eixo de rotação deste, reproduz vários tipos de movimentos, que poderão ser periódicos ou não. Controlando a amplitude e a frequência da força externa, o objetivo é prever e caracterizar o tipo de movimentos descritos pelo pêndulo. Neste primeiro trabalho sobre o tema, serão introduzidas as equações necessárias para a modelação do movimento e analisados os dados experimentais resultantes das leituras efetuadas num pêndulo cujo eixo de rotação está sujeito a um movimento horizontal periódico.

1 Introdução

Consideremos o sistema dinâmico composto pelo pêndulo, com um único ponto-massa, supondo que não existe atrito no eixo de rotação do pêndulo e que a massa do cabo rígido do pêndulo é desprezável. Assumindo que a única força a que o pêndulo se encontra sujeito é a força da gravidade, a equação que modela este sistema dinâmico é dada por

$$\frac{d\theta^2}{dt^2} + \frac{g}{L} \sin(\theta) = 0, \quad (1)$$

onde θ representa o ângulo que o cabo do pêndulo faz com o eixo vertical, g é a aceleração devida à gravidade e L representa o comprimento do cabo do pêndulo (figura 1).

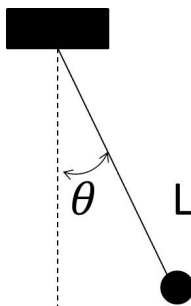


Figura 1: Esquema de um pêndulo.

Supondo que estamos nas condições do sistema descrito em (1) e considerando a resistência do ar a atuar

no ponto-massa, a equação do movimento do pêndulo amortecido pode ser dada por

$$\frac{d\theta^2}{dt^2} + \frac{g}{L} \sin(\theta) + k \frac{d\theta}{dt} = 0, \quad (2)$$

onde k é o coeficiente de viscosidade.

Se, em vez de considerarmos a resistência do ar, considerarmos uma força motora a atuar sobre o suporte do sistema, esta irá provocar um movimento horizontal oscilatório com uma determinada amplitude a e frequência ω , que iremos representar por h . Nestas condições, a equação que modela o movimento do pêndulo forçado é dada por:

$$\frac{d\theta^2}{dt^2} + \frac{g}{L} \sin(\theta) + \frac{1}{L} \cos(\theta) \frac{dh^2}{dt^2} = 0 \quad (3)$$

2 Modelação de um movimento harmónico forçado

Com o objetivo de confrontar resultados teóricos com resultados experimentais, nesta seção modelamos os dados experimentais resultantes das leituras efetuadas pelo sensor "Rotary Motion Sensor" da Vernier.



Figura 2: "Rotary Motion Sensor", da Vernier, usado para a obtenção dos dados experimentais.

O modelo matemático considerado é dado pela equação diferencial não linear de segunda ordem (3), visto que o eixo de rotação do pêndulo vai realizar um movimento horizontal oscilatório, com uma dada amplitude e frequência.

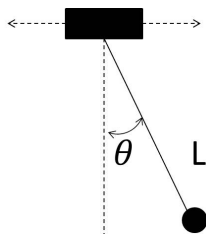


Figura 3: Esquema de um pêndulo cujo eixo de rotação realiza um movimento horizontal oscilatório.

2.1 Dados experimentais

É importante referir que o levantamento de dados foi efetuado relativamente ao ângulo θ , em função do tempo t em segundos, fixada uma determinada amplitude. Foram analisadas 3 situações distintas, sempre com um cabo de comprimento igual a 400 milímetros:

1. Considerámos, como condição inicial, o pêndulo em repouso na sua posição de equilíbrio. Do movimento que o eixo de rotação do sistema realiza apenas se conhece a amplitude, que neste caso é igual a 200 milímetros. Através da leitura dos dados, obtivemos o seguinte gráfico:

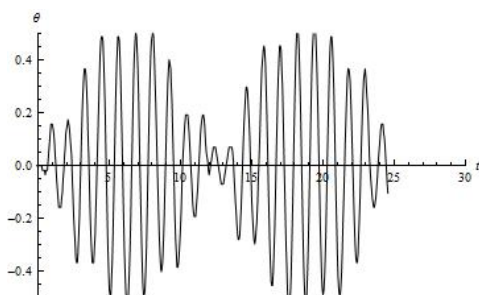


Figura 4: Movimento do pêndulo com $L = 400\text{mm}$ e $a = 200\text{mm}$.

2. Considerámos, como condição inicial, o pêndulo em repouso na sua posição de equilíbrio e a amplitude do movimento do eixo de rotação do pêndulo igual a 50 milímetros. Através da leitura dos dados, obtivemos o seguinte gráfico:

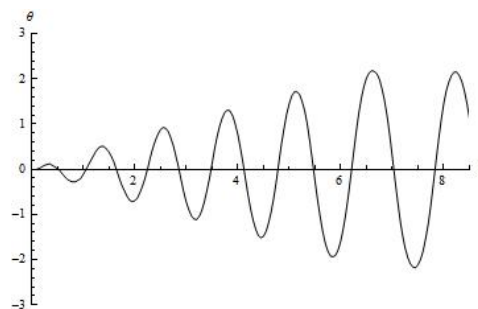


Figura 5: Movimento do pêndulo com $L = 400\text{mm}$ e $a = 50\text{mm}$.

3. Considerámos, como condição inicial, o pêndulo em repouso na sua posição de equilíbrio e a amplitude do movimento do eixo de rotação do pêndulo igual a 100 milímetros. Através da leitura dos dados, obtivemos o seguinte gráfico:

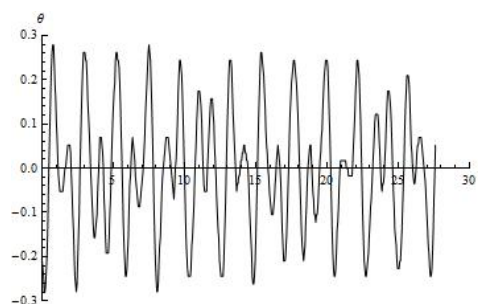


Figura 6: Movimento do pêndulo com $L = 400\text{mm}$ e $a = 100\text{mm}$.

2.2 Modelação dos dados

Analisando o comportamento do pêndulo, dadas as condições iniciais, o nosso estudo evidenciou 3 tipos distintos de comportamento, consequência da alteração da amplitude do movimento do eixo de rotação do pêndulo. No caso 1, o movimento do pêndulo é periódico; no caso 3, o movimento é caótico, isto é, não é possível prever a posição do pêndulo num instante futuro; no caso 2, identificámos o fenómeno de ressonância, dado que a frequência da força externa é igual à frequência natural do movimento do pêndulo, o que origina um crescimento linear, em função do tempo.

Para modelar a força externa que está a atuar no sistema, iremos considerar a função definida por

$$P_2(t) = \sum_{n=0}^N (-1)^n P_1(A(t - Bn)) \quad (4)$$

onde $N > 0$, $A \neq 0$ e B são constantes reais e

$$P_1(t) = \begin{cases} 0 & t < -0.5 \\ 1 & -0.5 < t < 0.5 \\ 0 & t > 0.5 \end{cases}$$

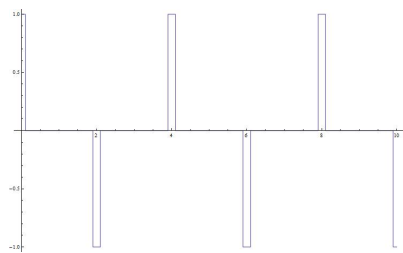


Figura 7: Gráfico de P_2 considerando $A = 5$ e $B = 2$.

Usando o software "Mathematica®", foram modelados os movimentos anteriormente descritos, considerando o seguinte código:

```
(* g - a aceleracao da gravidade *)
(* L - o comprimento do cabo*)
(* tmax - tempo decorrido*)
(* A - tempo decorrido com aceleracao maxima*)
(* B - frequencia da forza externa*)
soln[tmax_] := NDSolve[{
omega1'[t] == omega2[t],
omega2'[t] == -(1/L)*4*Sum[(-1)^(n)*
*HeavisidePi[A*(t-B*n)],{n, 0, 30}]*
*Cos[omega1[t]]-(g/L)*Sin[omega1[t]],
omega1[0] == 0, omega2[0] == 0},
{omega1, omega2}, {t, 0,tmax},
MaxSteps -> 50000];
solt = soln[tmax][[1]];
Plot[{Evaluate[{omega1[t]} /. solt]},
{t, 0, tmax}, PlotRange -> All]
```

Os resultados obtidos foram os seguintes:

1. Considerando $N = 30$, $A = 10$ e $B = 0.71$ na função dada em (4), obtemos um movimento com as mesmas características do que o obtido experimentalmente no caso 2, como podemos observar pelo seguinte gráfico.

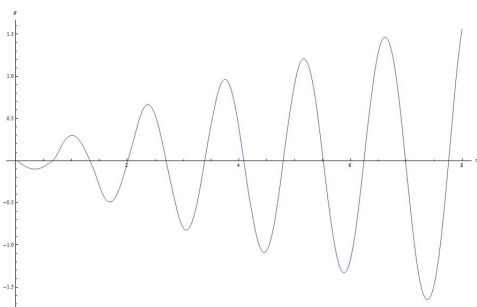


Figura 8: Solução de (3) considerando $N = 30$, $A = 10$ e $B = 0.71$ na função P_2 .

2. Se, em (4), considerarmos $N = 30$, $A = 8$ e $B = 1.7$, obtemos um movimento com as mesmas

características do que o obtido experimentalmente no caso 1, como podemos observar pelo seguinte gráfico.

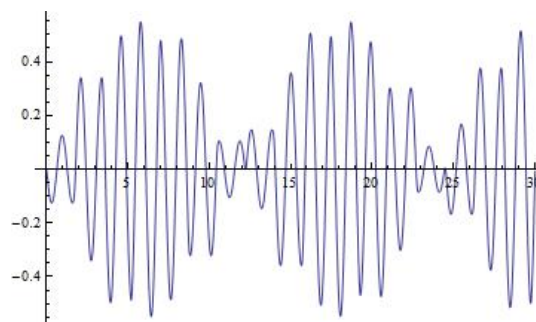


Figura 9: Solução de (3) considerando $N = 30$, $A = 8$ e $B = 1.7$ na função P_2 .

3 Conclusão

As equações diferenciais que modelam o movimento do pêndulo, forçado ou não, são bem conhecidas na literatura. A situação do eixo de rotação do pêndulo estar sob a ação de uma força motora foi também alvo de vários estudos, por exemplo [1] e [2].

Neste trabalho verificou-se que os modelos matemáticos conseguem corroborar, com alguma exatidão, os dados experimentais resultantes da análise do sistema físico. A maior dificuldade registada foi encontrar um modelo que descrevesse/incorporasse o movimento resultante da força motora, de modo que a solução obtida fosse ao encontro dos valores obtidos experimentalmente. Em trabalho futuro, é nosso objetivo modelar de forma mais exata este tipo de sistema, analisando dados experimentais associados aos movimentos resultantes da ação de uma força motora.

Nota: Os autores agradecem os contributos fornecidos por um revisor anónimo que permitiram retificar a expressão da equação (3) que se encontrava na primeira versão deste trabalho.

Bibliografia

- [1] JOHN M. SCHMITT and PHILIP V. BAYLY, Bifurcations in the Mean Angle of a Horizontally Shaken Pendulum: Analysis and Experiment, *Nonlinear Dynamics*: 15 1–14, 1998.
- [2] R. VAN DOOREN, Chaos in a Pendulum with Forced Horizontal Support Motion: a Tutorial, *Chaos, Solitons & Fractals* Vol. 7, No. 1, pp. 77–90. 1996.

Computer construction of Platonic Solids

by M. GASPAR^a, M. LEITE^b, N. MARTINS-FERREIRA^a AND B. N. PANDA^b

^a Centre for Rapid and Sustainable Product Development,
Polytechnic Institute of Leiria, Marinha Grande, Portugal

^b Departamento de Engenharia Mecânica,
Instituto Superior Tecnico, Universidade de Lisboa, Lisboa, Portugal

Abstract: In this paper we develop a novel method to generate a 3D geometrical model from 2D planer images. As an example of application, we construct 3D models of each one of the five platonic solids from their platonic graphs.

Keywords: Platonic solids, planar graphs, computer models, surfaces.

1 Introduction

Designing 3D geometries to a relatively high level of precision has become one of the challenging tasks for the design engineers. In this context, we are proposing a method for generating 3D solid models and exploring it's capability by considering the five platonic solids, namely the cube, the tetrahedron, the octahedron, the dodecahedron and the icosahedron.

A Platonic Solid is a 3D shape where each face is the same regular polygon and same number of polygons meets at each vertex. The working algorithm of our proposed method is explained in the below section.

2 Methodology

A 3D model in the space is the collection of surfaces, edges and vertices arranged in a proper order. If connection can be established in between these elements properly, then it is no longer difficult to generate solid models out of planer images. For this purpose, we have set up four design parameters such as α , g , q , β to extract the hidden information. The symbolic notations along with their meanings are explained below.

Using this topology information, 3D solid models can be generated for several application needs.

1. α is a permutation of the edges whose orbit goes to the edges of each facets
2. g is geometric realization of abstract vertices in \mathbb{R}^3
3. q identifies a surface by it's lowest vertex number
4. β is a permutation whose orbits can be identified with the vertices of the solid

The information provided by β can be used to modify the object by means of *geometrical* or *topological* transformations. An important example of geometrical transformation is the offset transformation which moves each

vertex (along with the associated incidence edges) according to a specified direction. In such cases the information encoded in the permutation β is used to track the new faces that are generated by the displacement of adjacent edges. From the point of view of topological transformations, one important aspect of it is that, the possibility to move from the graph of a surface to its dual graph by interchanging the role of the vertices and faces. With this respect, we simply have to interchange the role of the two permutations α and β .

For the method verification, the five platonic solids (mentioned in the introduction) are tested and their respective outcomes are displayed with illustrative sketches in the following section.

3 Design Examples

In this section, the four design parameters of each platonic solid are extracted and expressed in a matrix form. The syntax of the matrix is as below,

$$Matrix = [\alpha, g, q, \beta]$$

The matrices along with their respective vertices (shown in the individual section) are used as an input to our program for generating the 3D platonic solids. The solids are placed next to the matrix in the respective sections.

An example of this matrix can be seen in table 1 (see first row) where $\alpha(1)$ points to 2 and vertex g goes to 2 which corresponds to vertices $[1,-1,-1]$, as provided in section 3.1. The third parameter q equals to 1 represents one face of the cube while $\beta(1)$ is establishing a connection between vertex 1 and 14.

3.1 The Cube

Cube is the most basic platonic solid having six squares, three meeting at each corner. It is also known as hexahedron since "hexa" is a greek word for six which explains

all six squares of the cube. Fig. 1 shows the platonic graph of the cube where co-ordinates of each vertex are given below.

Cube and Tetrahedron vertices:

- [-1, -1, -1]
- [1, -1, -1]
- [1, 1, -1]
- [-1, 1, -1]
- [-1, -1, 1] % 5
- [1, -1, 1]
- [1, 1, 1]
- [-1, 1, 1]

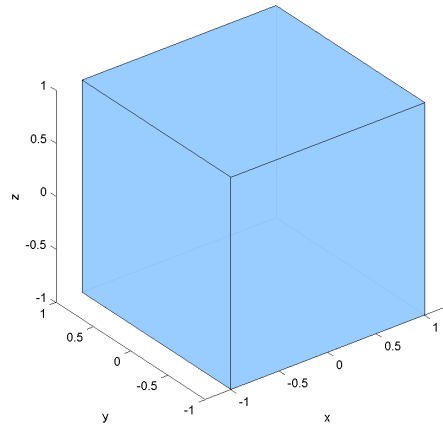


Fig. 2 - 3D Plot of a Cube.

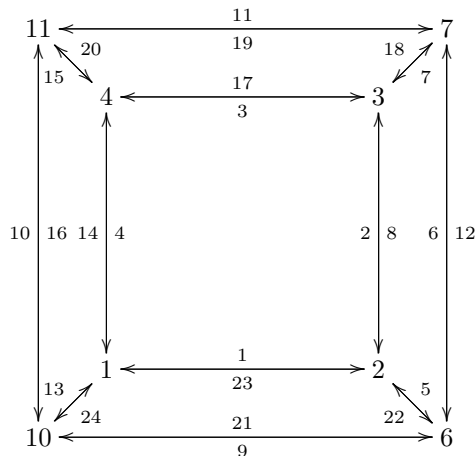


Fig. 1 - Planar graph of the cube.

Table 1 - Matrix for the Cube.

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
1	2	2	1	14
2	3	3	1	23
3	4	7	1	8
4	1	6	1	17
5	6	3	5	2
6	7	4	5	22
7	8	8	5	12
8	5	7	5	18
9	10	4	9	22
10	11	1	9	13
11	12	5	9	20
12	9	8	9	7
13	14	1	13	10
14	15	2	13	24
15	16	6	13	4
16	13	5	13	11
17	18	6	17	15
18	19	7	17	3
19	20	8	17	12
20	17	5	17	11
21	22	1	21	10
22	23	4	21	9
23	24	3	21	5
24	21	2	21	1

3.2 The Tetrahedron

The word "tetra" is a greek for four. A tetrahedron has four equilateral triangles, three meeting at each corner. Fig.3 shows the planar graph of the tetrahedron and the resulted solid is displayed in Fig.4.

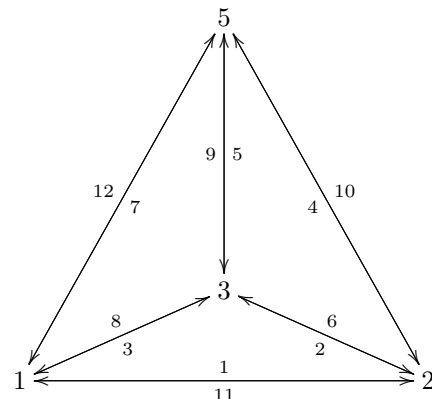


Fig. 3 - Planar graph of the Tetrahedron.

The matrix that encodes the data is the following one.

Table 2 - Matrix for the Tetrahedron.

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
1	2	8	1	8
2	3	6	1	12
3	1	3	1	6
4	5	6	4	2
5	6	1	4	10
6	4	3	4	9
7	8	1	7	5
8	9	8	7	12
9	7	3	7	3
10	11	1	10	7
11	12	6	10	1
12	10	8	10	4

From this graph we can generate the data which is displayed in the following matrix.

Table 3 - Matrix for the Octahedron.

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
1	2	14	1	11
2	3	10	1	14
3	1	13	1	6
4	5	10	4	2
5	6	9	4	23
6	4	13	4	9
7	8	9	7	5
8	9	12	7	20
9	7	13	7	12
10	11	12	10	8
11	12	14	10	17
12	10	13	10	3
13	14	11	13	16
14	15	10	13	24
15	13	14	13	1
16	17	11	16	19
17	18	14	16	15
18	16	12	16	10
19	20	11	19	22
20	21	12	19	18
21	19	9	19	7
22	23	10	22	13
23	24	11	22	21
24	22	9	22	4

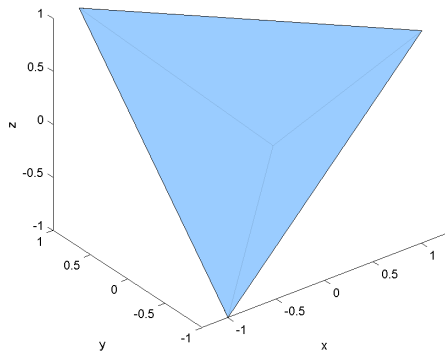


Fig. 4 - 3D plot of the Tetrahedron.

3.3 The Octahedron

Octahedron is also a triangle based platonic solid next to tetrahedron. The word "octa" is greek for eight. It consists of eight equilateral triangles, four meeting at each corner.

In order to produce the octahedron, in addition to the vertex coordinates displayed in section 3.1, we will need the following ones (note that the first line has index 9 because it is continuation of vertex in 3.1).

```
% Octahedron vertices:
[ 0, 0, -1]
[ 0, -1, 0] % 10
[-1, 0, 0]
[ 0, 1, 0]
[ 1, 0, 0]
[ 0, 0, 1]
```

The planar graph of the octahedron can be obtained as the dual of the planar graph of the cube and it is displayed below.

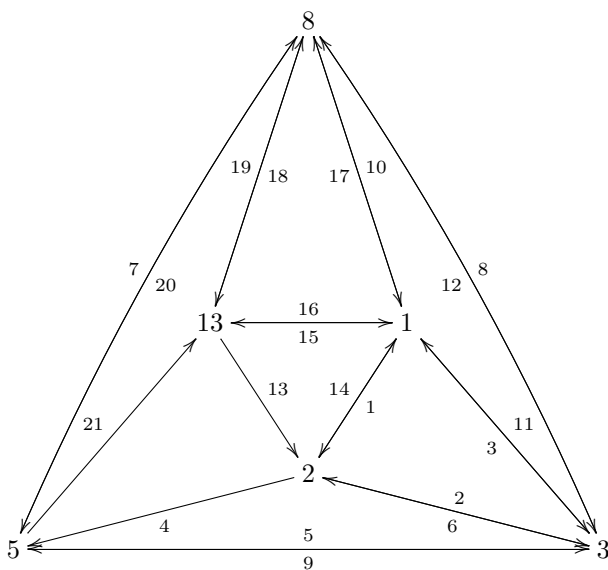


Fig. 5 - Planar graph of the Octahedron.

This matrix, along with the Vertices coordinates, contains all the needed information to generate the 3D solid (Fig. 6).

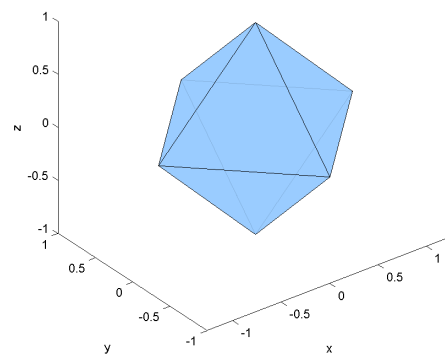


Fig. 6 - 3D plot of the Octahedron.

3.4 The Icosahedron

The last triangle based platonic solid is the icosahedron. The word "icosa" is greek for twenty. The icosahedron consists of twenty equilateral triangles, five meeting at each corner.

To build the Icosahedron we have to use the same vertices as the ones used to build the previous solids and add some others as displayed below, whose index for the first line in 15, as indicated.

```
Icosahedron vertices:
[ 1/2, 0, -1] % 15
[ -1/2, 0, -1]
[ 0, -1, 1/2]
[ 0, -1, -1/2]
[ -1, 1/2, 0]
[ -1, -1/2, 0] % 20
[ 0, 1, 1/2]
[ 0, 1, -1/2]
[ 1, 1/2, 0]
[ 1, -1/2, 0]
[ 1/2, 0, 1] % 25
[ -1/2, 0, 1]
```

The planar graph of the Icosahedron is displayed as follows (Fig. 8) from which we derive the following matrix:

Table 4 - Matrix for the Icosahedron.

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
1	2	26	1	6
2	3	17	1	34
3	1	25	1	23
4	5	25	4	3
5	6	21	4	17
6	4	26	4	9
7	8	21	7	5
8	9	19	7	10
9	7	26	7	31
10	11	19	10	50
11	12	21	10	7
12	10	22	10	13
13	14	22	13	28
14	15	21	13	11
15	13	23	13	16
16	17	23	16	19
17	18	21	16	14
18	16	25	16	4
19	20	23	19	25
20	21	25	19	18
21	19	24	19	22
22	23	24	22	38
23	24	25	22	20
24	22	17	22	2
25	26	23	25	29
26	27	24	25	21
27	25	15	25	54
28	29	22	28	60
29	30	23	28	15
30	28	15	28	7
31	32	26	31	34
32	33	19	31	8

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
33	31	20	31	46
34	35	26	34	1
35	36	20	34	33
36	34	17	34	41
37	38	18	37	40
38	39	24	37	52
39	37	17	37	24
40	41	18	40	3
41	42	17	40	39
42	40	20	40	35
43	44	18	43	40
44	45	20	43	46
45	43	16	43	56
46	47	20	46	44
47	48	19	46	32
48	46	16	46	49
49	50	16	49	59
50	51	19	49	12
51	49	22	49	12
52	53	24	52	26
53	54	18	52	37
54	52	15	52	57
55	56	18	55	53
56	57	16	55	45
57	55	15	55	58
58	59	15	58	30
59	60	16	58	56
60	58	22	58	51

And with the help of above matrix we build 3D model of the Icosahedron and it is shown in Fig.7.

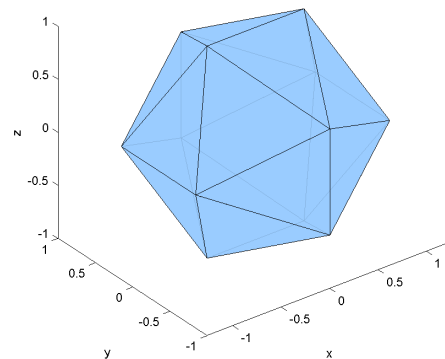


Fig. 7 - 3D plot of the Icosahedron.

3.5 The Dodecahedron

The fifth and last platonic solid is dodecahedron. The word "dodeca" is greek for twelve. The dodecahedron has twelve regular pentagons, three meeting at each corner.

The Dodecahedron is the dual of the icosahedron and hence its vertices can be obtained as the middle point of

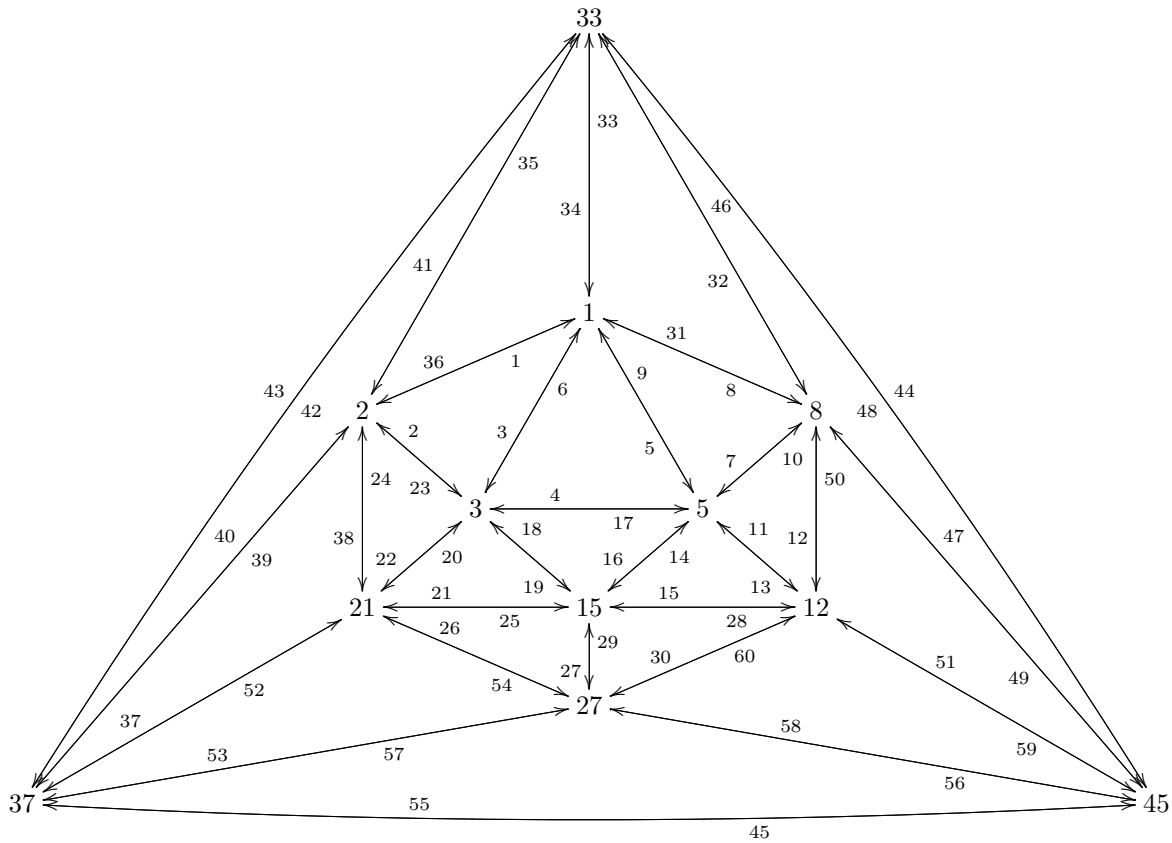


Fig. 8 - Icosahedron planar graph.

each face of the Icosahedron. It's planer image and generated 3D solid are shown in Fig. 9 and 10 respectively.

Dodecahedron vertices:

- [0, -1/3, 5/6]
- [0, 1/3, 5/6]
- [-1/2, 1/2, 1/2]
- [-1/3, 5/6, 0] % 30
- [1/3, 5/6, 0]
- [1/2, 1/2, 1/2]
- [5/6, 0, 1/3]
- [1/2, -1/2, 1/2]
- [5/6, 0, -1/3] % 35
- [1/2, 1/2, -1/2]
- [-5/6, 0, 1/3]
- [-1/2, -1/2, 1/2]
- [1/3, -5/6, 0]
- [-1/3, -5/6, 0] % 40
- [-1/2, -1/2, -1/2]
- [-5/6, 0, -1/3]
- [-1/2, 1/2, -1/2]
- [1/2, -1/2, -1/2]
- [0, -1/3, -5/6] % 45
- [0, 1/3, -5/6]

Table 5 - Matrix for the Dodecahedron.

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
1	2	32	1	31
2	3	31	1	59
3	4	23	1	7
4	5	22	1	28
5	1	21	1	37
6	7	22	6	4
7	8	23	6	58
8	9	24	6	17
9	10	25	6	12
10	6	26	6	29
11	12	26	11	10
12	13	25	11	17
13	14	30	11	49
14	15	29	11	22
15	11	27	11	30
16	17	25	16	9
17	18	24	16	57
18	19	37	16	53
19	20	40	16	50
20	16	30	16	13
21	21	27	21	15
22	23	29	21	48
23	24	38	21	43

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
24	25	33	21	39
25	21	28	21	26
26	27	28	26	38
27	28	21	26	5
28	29	22	26	6
29	30	26	26	11
30	26	27	26	21
31	32	32	31	36
32	33	34	31	41
33	34	35	31	55
34	35	36	31	60
35	31	31	31	2
36	37	32	36	1
37	38	21	36	27
38	39	28	36	25
39	40	33	36	42
40	36	34	36	32
41	42	34	41	40
42	43	33	41	24
43	44	38	41	47
44	45	39	41	33
45	41	35	41	33
46	47	39	46	51
47	48	38	46	43
48	49	29	46	22
49	50	30	46	13
50	46	40	46	19
51	52	39	51	44
52	53	40	51	50
53	54	37	51	56
54	55	36	51	34

x	$\alpha(s)$	$g(x)$	$q(x)$	$\beta(x)$
55	51	35	51	45
56	57	37	56	18
57	58	24	56	8
58	59	23	56	3
59	60	31	56	35
60	56	36	56	54

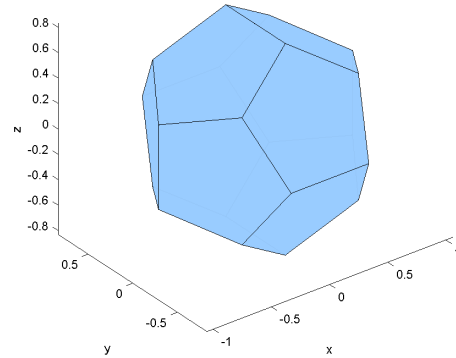


Fig. 9 - 3D plot of the Dodecahedron.

4 Conclusion

The above figures represent that our proposed method is capable of generating platonic solids without any geometrical error. In future, this approach can be extended for generating and 3D printing of complex cellular solids which is difficult to produce using the existing CAD packages.

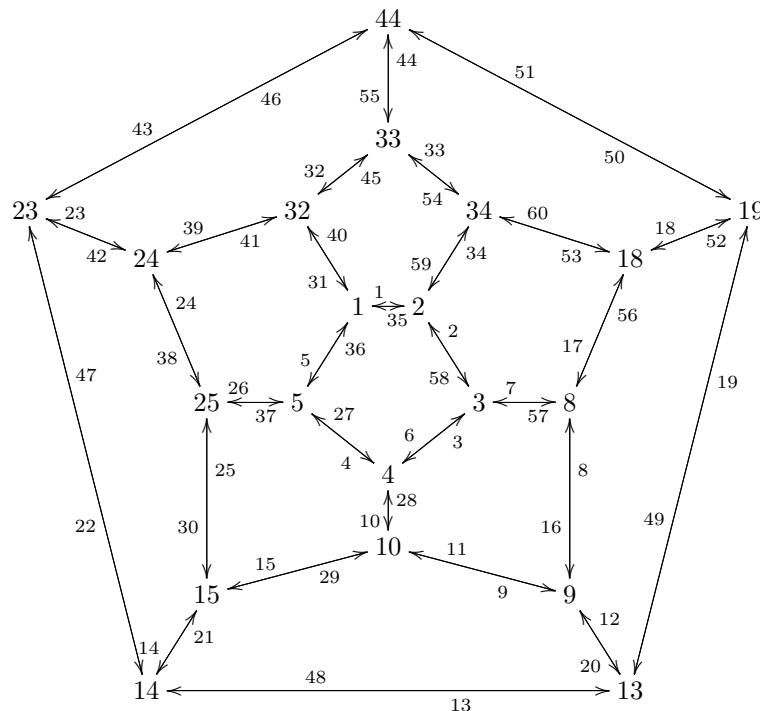


Fig. 10 - Dodecahedron planar graph.

Sistema de aquisição e registo de sinais biomédicos baseado numa plataforma de desenvolvimento *open-source* de baixo custo

by DANIEL JARDAN^a, JOÃO GIL^a, FERNANDO MARTINS^{a*}, LINO FERREIRA^{a,b}, RUI FONSECA-PINTO^{a,b}

^a Instituto Politécnico de Leiria/ESTG, Leiria, Portugal

^b Instituto de Telecomunicações, Portugal

* fernando.martins@ipleiria.pt

Resumo Este artigo apresenta um sistema de aquisição e registo de sinais biomédicos baseado na plataforma de desenvolvimento *open-source* *Arduino*. O sistema tem associado um cartão de memória para registo de dados, bem como um acelerómetro que indica a atividade/postura do paciente. É utilizado um módulo de comunicação *Bluetooth* para ligação a um dispositivo móvel (*smartphone* e *tablet*) de forma a permitir a parametrização, o comando do sistema de aquisição e a monitorização em tempo real dos dados adquiridos. Estes dados foram usados para calcular o balanço vago-simpático associado a tarefas específicas do dia-a-dia.

1 Introdução

O registo de sinais biológicos é muitas vezes vital no diagnóstico e vigilância clínica de doentes. Em determinados casos, este tipo de procedimento tem de ser efetuado em meio hospitalar em pacientes internados, seguindo determinado protocolo. Noutras situações, é mesmo necessário adquirir este tipo de informação em condições normais da vida quotidiana, e para isso são utilizados dispositivos portáteis, dedicados e normalmente dispendiosos como, por exemplo, o gravador Holter para registar continuamente o electrocardiograma (ECG). O desenvolvimento de equipamentos de diagnóstico para monitorização e análise de sinais biomédicos sofreu um forte crescimento nos últimos anos e a procura de novos sistemas de aquisição de sinais biomédicos justifica o desenvolvimento de novas soluções que respondam às exigências médicas atuais [1, 2, 3, 4].

A banalização da transmissão de dados através da Internet, do *Wi-Fi* e das redes celulares, permite que equipamentos médicos, com capacidade de ligação a dispositivos móveis, tornem possível partilhar os dados adquiridos com os técnicos de saúde em qualquer lugar e em qualquer instante, monitorização do paciente de forma remota, receber alarmes, etc. As aplicações de um dispositivo com estas capacidades são inúmeras. Por exemplo, podem evitar a necessidade do paciente se deslocar ao hospital ou à clínica para descarregar os dados, pode ser usado para monitorizar pacientes de alto risco diretamente a partir de suas próprias casas, entre outras.

O objetivo deste trabalho é o desenvolvimento de

um sistema de aquisição e registo de múltiplos sinais biomédicos baseado numa plataforma de desenvolvimento *open-source* de baixo custo e com capacidade de ligação a um dispositivo móvel. O sistema deverá, ainda, ter a capacidade de funcionar de forma autónoma com capacidade de registo dos dados recolhidos num formato que permita a posterior leitura e análise por vários tipos de software, tais como *Excel*, *Access*, *Matlab*. Face aos objetivos traçados, recorreu-se à plataforma de desenvolvimento *Arduino* para implementar o sistema, dada a sua flexibilidade, simplicidade de utilização, de ser uma plataforma aberta e especialmente devido ao facto de ter baixo custo.

Este artigo está organizado da seguinte forma. A Secção 2 faz uma descrição geral do sistema proposto. Nas Secções 3 e 4 são apresentados os detalhes do hardware e software desenvolvido, respetivamente. Na Secção 5 apresentam-se alguns resultados experimentais obtidos com o sistema proposto. Finalmente, na Secção 6 são apresentadas as conclusões e o trabalho futuro.

2 Descrição do sistema

O sistema proposto está baseado na plataforma de desenvolvimento *Arduino Uno* [5] dotada do microcontrolador ATmega328. Esta plataforma é composta por 14 entradas/saídas digitais e 6 entradas analógicas, contém também uma ligação USB (*Universal Serial Bus*) para, por exemplo, efetuar a programação e comando da placa através de um PC. Tem igualmente a possibilidade de ligar a outros dispositivos através da interface de comunicação SPI (*Serial Peripheral Interface*), ver Figura 1.

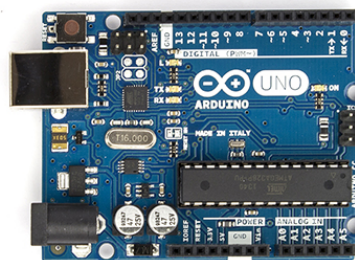


Fig 1. Plataforma de desenvolvimento *Arduino Uno* [5].

Esta plataforma de desenvolvimento adquire os sinais dos diferentes tipos de sensores através dos portos de entrada analógicos e digitais, e ativa os vários tipos de alarmes presentes através dos portos de saída. A capacidade de armazenamento é obtida através da ligação a um cartão de memória standard (*SD card*) por meio da *interface SPI*. A capacidade de comunicação com um dispositivo móvel é alcançada através de um módulo da comunicação *Bluetooth* que liga aos pinos RX e TX do porto de comunicação série disponível na plataforma. A Figura 2 dá uma perspetiva geral do sistema proposto. O sistema tem capacidade de obtenção de informação de frequência cardíaca, de temperatura corporal e ainda a posição/postura do paciente recorrendo à utilização de um acelerómetro. Considerou-se que seria uma mais-valia a inclusão deste tipo de sensor no sistema face às suas potenciais aplicações no campo médico e terapêutico. Este sensor pode ser usado na avaliação do movimento humano, na deteção de distúrbios do sono, mudanças de posição, entre outras. Os acelerómetros podem ser, igualmente, usados na medicina de reabilitação para determinar o risco de queda de pessoas (que pode resultar em fraturas ou até mesmo morte), em que as causas subjacentes estão relacionadas com uma marcha e equilíbrio deficientes. Alarmes são acionados caso sejam atingidos valores limite de ritmo cardíaco ou de temperatura corporal.

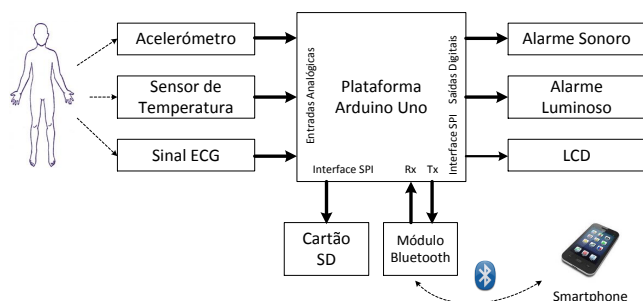


Fig 2. Diagrama de blocos do sistema proposto.

O sistema apresenta uma *interface* para um dispositivo móvel *Android* que permite ao utilizador controlar o início e a paragem da aquisição de dados, a definição do intervalo e da taxa amostragem, bem com definir e registar as atividades mais relevantes durante o período do registo dos sinais (dormir, comer, andar, correr, etc), e os instantes em que ocorrem. A *interface* permite ainda visualizar os valores instantâneos do ritmo cardíaco e da temperatura corporal. Nesta *interface* poder-se-ia, ainda, desencadear outro tipo de alarmes no próprio dispositivo móvel ou efetuar chamadas ou envio de SMS (*Short Messaging Service*) de emergência para números previamente definidos.

3 Hardware

O componente principal do sistema desenvolvido é a plataforma *Arduino Uno* (ver Figura 1), à qual são ligados os diferentes sensores e dispositivos que constituem o sistema proposto. Far-se-á de seguida uma descrição dos vários elementos de hardware usado no trabalho.

- O sinal ECG é obtido através de três elétrodos colocados no corpo do paciente, cujos sinais são devidamente acondicionados (amplificação, filtragem, isolamento). O ritmo cardíaco é obtido a partir deste sinal de ECG.
- A temperatura corporal é obtida através de um sensor de temperatura integrado (LM35) encostado ao corpo do paciente. O sinal de saída do sensor é devidamente amplificado antes chegar a uma das entradas analógicas da plataforma *Arduino*.
- Para o acelerómetro foi usado o dispositivo integrado ADXL335 [6]. É um dispositivo de pequenas dimensões, baixo consumo, que contém um acelerómetro de três eixos com saídas em tensão já acondicionadas. Este dispositivo permite, assim, a ligação direta a três das entradas analógicas da plataforma *Arduino*.
- O alarme sonoro presente no protótipo é implementado por um pequeno bescoro sendo acionado através de uma saída digital. O alarme pode ser desativado através de um botão de pressão caso corresponda a uma situação de falso alarme. Pode ser usado para diferentes fins, como por exemplo em situações em que ocorra uma possível queda, permitindo, assim, alertar as pessoas que estão nas proximidades para prestar a necessária ajuda.
- Os alarmes luminosos são implementados através de LEDs (Light-Emitting Diode) que são ativados a partir de saídas digitais. Destinam-se a sinalizar situações de temperatura corporal elevada (LED amarelo) e situações de arritmias (LED vermelho em conjunto com um "beep" sonoro).
- O sistema incorpora o LCD (Liquid Crystal Display) Nokia 5110 [7] para visualizar os valores obtidos pelos sensores em tempo real, assim como os vários parâmetros de funcionamento definidos (taxa e intervalo de amostragem, níveis de alarme, temperatura, ritmo cardíaco, etc).
- A ligação à plataforma é feita através da *interface SPI*. A capacidade do sistema para comunicar por meio de *Bluetooth* com dispositivos móveis, ou outros dispositivos, é conseguida através da utilização do módulo integrado HC06 [8]. Este módulo liga diretamente aos pinos RX e TX do porto de comunicação série da plataforma *Arduino*, necessitando apenas de ser "ativado" pelo dispositivo móvel para

que seja possível efetuar a comunicação. Este módulo permite o controlo do sistema a partir de um dispositivo externo (dispositivo móvel, PC, etc), assim como o envio de informação para esse mesmo dispositivo.

- A capacidade de armazenamento do sistema é conseguido através da ligação de um cartão de memória SD através da interface SPI. Esse cartão é usado para o registo contínuo dos dados obtidos pelo sistema através dos sensores. Os dados são gravados num ficheiro em formato CSV (Comma-Separated Values) podendo assim ser facilmente lidos e processados por diferentes tipos de software. O ficheiro poderá conter informação relativamente aos seguintes campos: hora; data; ritmo cardíaco; temperatura; aceleração em X, Y e Z; e tipo de atividade em que o paciente está envolvido. A utilização do cartão de memória permite que, uma vez configurado o sistema de aquisição de dados, este possa funcionar de forma autónoma efectuando a leitura e o registo contínuo dos dados para o cartão durante o período e com a taxa de amostragem previamente definidos. Esta opção, ao invés de guardar todos os dados no dispositivo móvel, evita a necessidade de ligação constante por Bluetooth entre os dois dispositivos. Permite, assim, um aumento de autonomia dos dispositivos e evita a perda de dados devido a uma qualquer quebra de ligação na comunicação. Todavia, em qualquer momento é possível ao paciente aceder através do dispositivo móvel aos dados que estão a ser adquiridos, assim como alterar os parâmetros de aquisição. Basta que para tal se estabeleça a comunicação Bluetooth a partir do dispositivo móvel.

O protótipo deve ser transportado pelo paciente durante o período em que decorre o processo de aquisição, pelo que se optou por uma alimentação a partir de uma pilha standard de 9 volts, sendo, por isso, facilmente substituível pelo paciente em caso de necessidade.

4 Firmware e software

O microcontrolador da plataforma de desenvolvimento foi programado usando o ambiente de programação IDE (*Integrated Development Environment*) do *Arduino* [9]. Trata-se de um software *open-source* de fácil utilização em que a programação é realizada através de um PC.

A aplicação *Android* que corre no dispositivo móvel foi desenvolvida recorrendo ao ambiente de programação *App Inventor for Android* [10]. Este ambiente de programação é uma aplicação web *open-source* disponibilizada pelo *Massachusetts Institute of Technology* (MIT). O código associado à plataforma *Arduino* tem como principais funções:

- Controlar o processo de aquisição lendo as entradas analógicas e digitais a uma taxa pré-definida (taxa de amostragem);
- Gravar os dados recolhidos no cartão de memória;
- Ativação de alarmes em situações anómalas pré-definidas (temperatura corporal elevada, braquicardia e taquicardia);
- Receber comandos do dispositivo remoto via *Bluetooth* referentes à parametrização do sistema (acerto data/hora, definição da taxa de amostragem e intervalo de amostragem) e ao seu controlo (início e paragem do processo de aquisição).

Depois da fase inicial de *setup* o programa executa um *loop* contínuo onde são executadas várias ações. A leitura das entradas a uma taxa fixa é alcançada recorrendo à capacidade de programar a plataforma *Arduino* para gerar interrupções a cada período fixo. Sempre que a interrupção é gerada o programa em execução é interrompido para correr a rotina de serviço a essa interrupção que faz a leitura e escrita dos dados em cartão. A deteção e leitura de comandos oriundos do dispositivo móvel é feita recorrendo a uma função do sistema (*SerialEvent*), que é executada em cada ciclo *loop* do código *Arduino* sempre que novos dados estejam presentes na entrada Rx do canal série.

A aplicação desenvolvida para o dispositivo móvel *Android* tem as seguintes funcionalidades:

- Estabelecer ou finalizar a ligação *Bluetooth*;
- Estabelecer os parâmetros de aquisição (taxa de aquisição e intervalo de aquisição);
- Acertar a hora/data do sistema de aquisição;
- Iniciar ou parar o processo de aquisição de dados;
- Mostrar os valores de alguns dos dados adquiridos pelo sistema de aquisição (temperatura corporal e o número de batimentos por minuto-BPM);
- Definir a atividade realizada pelo paciente no momento (escolher uma atividade a partir de uma lista previamente definida).

A comunicação com a plataforma *Arduino* é feita através de *Bluetooth* sendo usada uma *string* para enviar dados/comandos. As *strings* enviadas têm sempre o mesmo terminador e um caracter inicial diferente para cada dado/comando enviado. Os dados são enviados/recebidos apenas quando a comunicação *Bluetooth* está estabelecida. A atividade realizada pelo paciente é sempre guardada num ficheiro no dispositivo móvel. A informação deste ficheiro pode ser associada à informação presente no cartão de memória para permitir ao técnico de saúde saber quais as atividades realizadas pelo paciente e os instantes em que ocorrem. A Figura 3 mostra o aspeto da *interface* da aplicação do dispositivo móvel.

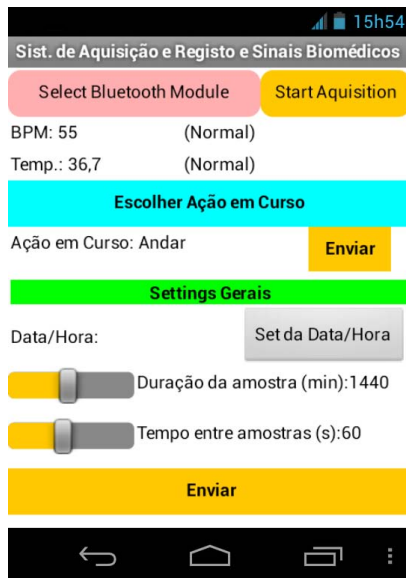


Fig 3. Interface no dispositivo móvel.

5 Resultados experimentais

O protótipo foi testado num ensaio com duração de 24 horas de registo do ritmo cardíaco, temperatura corporal e a atividade/postura do paciente com um intervalo entre amostras de um minuto. A informação guardada permitiu analisar posteriormente e de forma adequada a evolução temporal dos sinais registados. A Figura 4 ilustra um excerto dos dados obtidos e analisados a partir de uma folha em *Excel*.

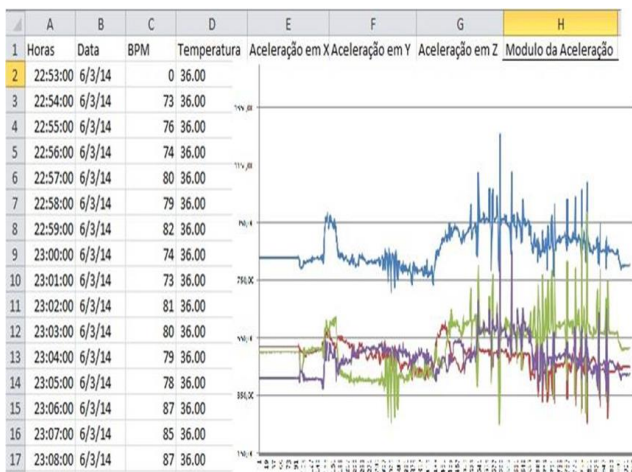


Fig 4. Dados visualizados em folha *Excel*.

Neste ensaio o módulo de comunicação *Bluetooth* e LCD foram desligados após a parametrização do sistema de aquisição e respetiva ordem de arranque. Verificou-se que a manutenção deste módulo em operação diminuía bastante a autonomia do sistema.

Para testar a aplicação em termos de avaliação clínica, os dados obtidos durante o registo foram analisados por períodos com interesse do ponto de vista da ativação fisiológica do Sistema Nervoso Autónomo (SNA). A vari-

abilidade do ritmo cardíaco obtida pela série temporal referente aos tempos entre cada onda R do ECG e a sua análise espectral através da Transformada de *Fourier*, permite, em conjunto com a informação inercial cedida pelo acelerómetro, identificar padrões de ativação das componentes simpática e parassimpática do Sistema Nervoso. Quando se estudam as variações da frequência cardíaca ao longo do dia, é sempre necessário ter em conta as necessárias variações/solicitações externas que permite manter a homeostasia. Estas adaptações posturais, emocionais e de movimento (subir escadas, sentar, correr, comer, dormir) bem como os períodos em que as mesmas acontecem, são importantes para em conjunto com a variação da frequência cardíaca avaliar as componentes do SNA e medir o balanço vago-simpático. Este tipo de metodologia em conjunto com a informação da temperatura corporal, complementada com outras informações clínicas, é mais uma ferramenta a juntar às clássicas avaliações do registo da pressão em ambulatório durante 24 horas, contribuindo para fazer uma avaliação etiológica das disautonomias (doenças do sistema nervoso autónomo).

6 Conclusões e trabalho futuro

Foi utilizada uma plataforma de desenvolvimento *Arduino* para implementar um sistema de aquisição de sinais biomédicos. A esta plataforma *open-source* e de baixo custo foi associado um cartão de memória para registo de dados e um módulo de comunicação *Bluetooth* para ligação a um dispositivo móvel. Os dados experimentais revelam que o sistema permite o registo adequado dos sinais obtidos para posterior análise por um técnico de saúde. O comando e monitorização dos dados em tempo real a partir de um dispositivo móvel torna o sistema bastante flexível. A utilização de um cartão de memória garante ao sistema a possibilidade de funcionamento de forma autónoma, evitando, assim, a necessidade de ligação contínua ao dispositivo móvel. Os dados experimentais obtidos num teste de 24 horas revelaram uma correlação fisiológica relativa ao balanço vago-simpático referente às tarefas e períodos analisados.

O trabalho futuro será direcionado para a integração e aquisição de outros sinais biomédicos (ritmo respiratório, pressão arterial, SPO2) e para um aumento da autonomia do sistema. Um outro aspeto a explorar será a possibilidade de utilização dos próprios sensores presentes nos dispositivos móveis atuais, por exemplo para obter e integrar informação de localização e tipo de atividade/postura de paciente.

Bibliografia

- [1] Mobile ECG, <http://www.alivecor.com/home>,
acedido Junho de 2015.
- [2] Guerreiro, J., Martins, R., Silva, H., Lourenço, A.,
Fred, A., BITalino: A multimodal platform for
physiological computing, In Proc. of the 10th
ICINCO Conference, Reykjavik, Iceland, Jul 2013.
- [3] Lucani, D., Cataldo, G., Cruz, J., Villegas, G.,
Wong, S., A portable ECG monitoring device with
Bluetooth and Holter capabilities for telemedicine
applications, Proceedings of the 28th IEEE EMBS
Annual International Conference, New York City,
USA, Aug 30-Sept 3 2006, pp.5244-5247.
- [4] Carvalho, G., Continuous ECG Acquisition Holter
System with mobile phone, Master Thesis, Instituto
Superior Técnico, Universidade Técnica de Lisboa,
2011.
- [5] Arduino Uno Board,
<http://www.arduino.cc/en/Main/ArduinoBoardUno>,
acedido Junho de 2015.
- [6] Analoge Devices Accelerometer ADXL335,
[http://www.analog.com/media/en/technical-
documentation/data-sheets/ADXL335.pdf](http://www.analog.com/media/en/technical-
documentation/data-sheets/ADXL335.pdf),
acedido Junho de 2015.
- [7] Nokia 5110 LCD,
[https://www.sparkfun.com/datasheets/
LCD/Monochrome/Nokia5110.pdf](https://www.sparkfun.com/datasheets/
LCD/Monochrome/Nokia5110.pdf) ,
acedido Junho de 2015.
- [8] HC-06 Serial Port Bluetooth,
[ftp://imall.iteadstudio.com/Electronic
_Brick/IM120710006/DS_IM120710006.pdf](ftp://imall.iteadstudio.com/Electronic
_Brick/IM120710006/DS_IM120710006.pdf),
acedido Junho de 2015.
- [9] Arduino Software,
<http://www.arduino.cc/en/Main/Software>,
acedido Junho de 2015.
- [10] MIT App Inventor, <http://appinventor.mit.edu>,
acedido Junho de 2015.

NASA's breathtaking images of the sun.

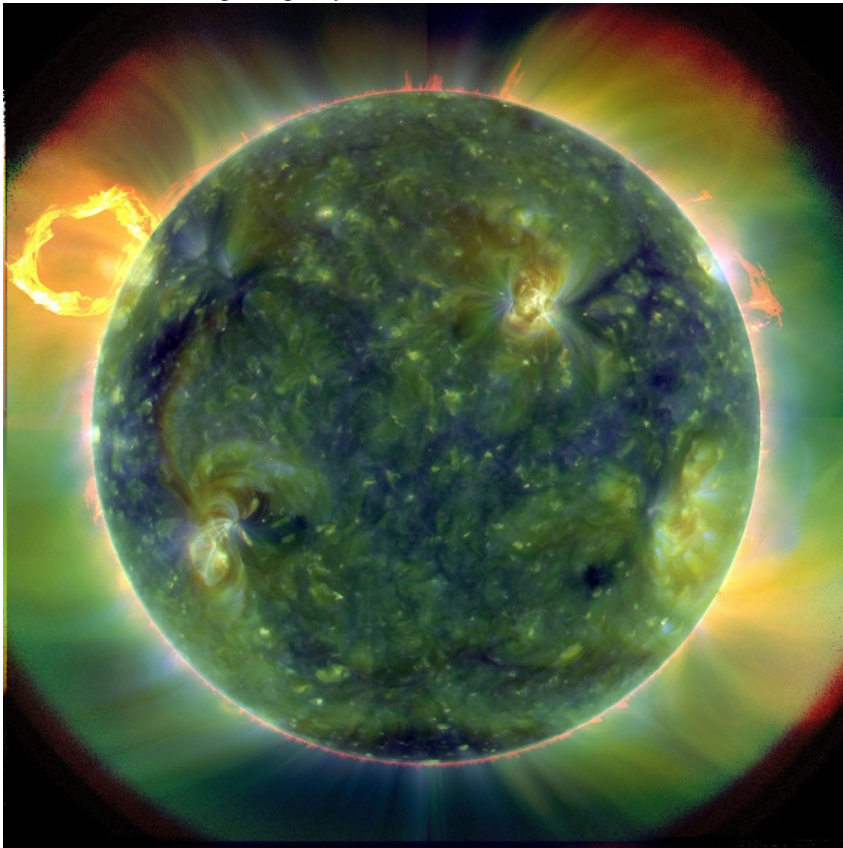


Image Credit:
NASA/Goddard/SDO
AIA Team

A full-disk
multi-wavelength
extreme
ultraviolet image of the
sun taken by SDO on
March 30, 2010. False
colors trace different
gas temperatures.
Reds are relatively cool
(about 60,000 Kelvin,
or 107,540 F); blues
and greens are hotter
(greater than 1 million
Kelvin, or 1,799,540 F).

Computer tool for maximizing the placement of congruent polyhedra

by M. GASPAR^a, N. ALVES^a, A. MATEUS^a AND N. MARTINS-FERREIRA^a

^a Centre for Rapid and Sustainable Product Development, Polytechnic Institute of Leiria, Marinha Grande, Portugal

Abstract: Given multiple identical polyhedral objects and a parallelepiped container, how should one place the objects so that the largest number fits inside the container? This simple question is important in many applications, yet the answer is elusive. In fact, we know of no published solution for this very general formulation. Still, in many circumstances, further restrictions apply, resulting in a large number of variations requiring different algorithmic strategies. This paper is the continuation of [12] and focus on the fundamental concepts and tools that are used for this kind of problem, such as the no-fit polygon. We also present some of its many variations, giving in particular one that applies to the stereolithographic rapid prototyping technology.

Keywords: Rapid Prototyping, Stereolithography, Containment, Packing, Optimization.

1 Introduction

The problem of maximizing the number of items that fit inside a container is one of a family of problems known as Cutting and Packing (C&P), most of which are notoriously hard. For an overview of this family of problems, see [4]. These problems include for instance the cutting of large sheets or rolls of paper to produce a given number of specific size sheets, with minimal waste; or the order and positions for placing different-sized boxes in a container.

The particular problem we will address is that of placing the maximum number of objects in the work area of an additive rapid manufacturing or rapid prototyping (RP) machine. This is most relevant for those RP techniques where a whole layer is processed at a time, as is the case in stereolithography. When using extrusion-based processes, we are usually interested in minimizing the total path length, and thus software for those machines will try to minimize the total area occupied by a user-specified set of parts. In stereolithography, on the contrary, since the irradiation of a layer is done by projection, it takes an approximately constant time and energy to process each layer, regardless of the fraction of the work area that is occupied by the fabricated parts. So, placing more parts in the work area directly translates to time- and energy-efficiency.

Several methods have been proposed for packing multiple distinct parts in a container, either minimizing the length of the container or maximizing the total number of parts. Most of those methods have some of the following characteristics: based on heuristic, iterative, stochastic; and most are formulated for two dimensional problems or for a restricted set of part geometries (for instance, parallelepiped boxes). For large production runs, trial-and-error, progressive optimization, or processing-intensive algorithms may be useful. But for the small

or one-time series that are typical of rapid prototyping/rapid manufacturing, a fast method for obtaining near optimal layouts is desirable.

Although this application does not require it, we will consider only identical, congruent objects placed in a regular lattice. Also, as it is the most common case, only parallelepiped working areas will be considered.

This problem involves two limit-problems, according to the relative dimensions of the container and the contained objects: if the dimensions of the container are much larger than those of the objects, the solution is mostly dependent on the density of the placement; conversely, if the dimensions are similar, the solution depends more on the intersection of the objects with the boundary of the container. In the first case we have a packing problem, whereas in the second case we have a containment problem.

Typically, the objects to fabricate cannot be stacked vertically, since they must be supported against gravity. Further, as there is a privileged direction (the direction of slicing), the orientation of the parts may be restricted in one axis, for proper surface finishing or mechanical properties. These two restrictions are important, as we shall see, in that they allow us to reduce this three-dimensional packing problem to a two-dimensional equivalent.

In this paper we will focus on the fundamental concepts and tools used for this kind of problem, such as the no-fit polygon. These concepts will be illustrated by means of an interactive software application we developed in Matlab. We briefly present two previously published methods for finding optimal or nearly-optimal solutions to the packing and containment problems in two dimensions [1, 5], the limitations and advantages of each method, and how they may be applied to this problem. Finally we discuss how the interactive application developed can be useful in exploring the application of similar techniques to a few other three-dimensional problems.

2 Description of the method

The solution to the problem stated in the introduction involves several conceptually distinct optimization problems; this separation is helpful in understanding the problem, even if not all sub-problems can be treated independently. The following subsections deal with each of the steps presented below:

1. **Reduction of the three-dimensional problem to a two-dimensional one.** In this section we will analyze the constraints under which it is possible to apply the algorithms for the planar packing for problems formulated in three-dimensional space.
2. **Densest linear packing of polygons.** In this section we will present the concept of no-fit polygon, how it is computed, and how it captures the non-overlapping constraint in one dimension.
3. **Finding feasible solutions to dense regular packing of polygons.** In this section we will again use the no-fit polygon, and show how it can be extended to find non-overlapping tiling of the plane.
4. **Determination of the densest tiling.** This section mainly describes the method proposed by Stoyan and Patsuk [1] to identify, among all non-overlapping tiling, those of maximum density.
5. **The Containment Problem.** In this section, for further development, we give some directions for determining the translation and rotation of a (densest) periodic tiling that minimizes the number of lattice cells intersecting the boundary of the container, on the one hand, and if there exists a non-optimal lattice that may produce a better overall solution, on the other.

2.1 Reduction of the three-dimensional problem to a two-dimensional one

The condition of no vertical overlap means that for some given orientation of a polyhedron P , we need only consider the packing of its projection on the horizontal plane (we also use P to refer to that polygonal projection). Thus, if we additionally restrict the rotation of P only to the vertical axis, the problem is effectively reduced to the two-dimensional case. The same reasoning can be applied to the containment problem.

Although rapid fabrication is sometimes used for the production of pre-assembled multi-part structures, we will restrict our analysis to the single part case, i.e., where the interior of the polyhedral surface is simply connected. This implies that the projection along any direction will have a simply connected interior. Note that its boundary may still not be a simple polygon. There may be ‘holes’ in the polygon, but clearly the polygon will not fit inside

its own holes; so, in fact, we may consider only the outer boundary of the polygon.

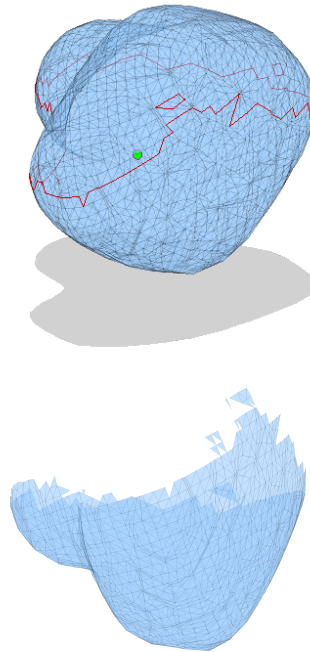


FIGURE 1: A polyhedron with a polygonal contour separating facets with upward and downward normals. The union of the contour projections is represented as the shadowed area. Downward oriented facets are displayed in the separated picture.

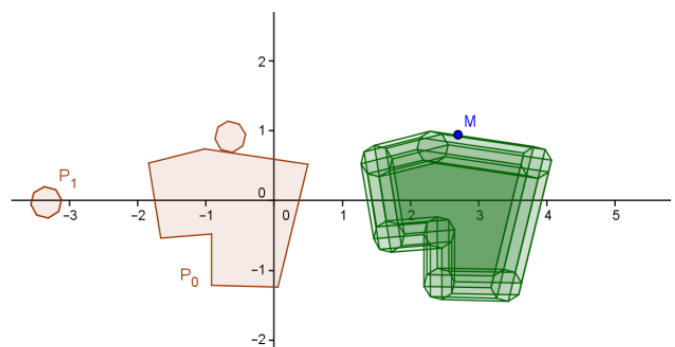


FIGURE 2: Dilation of a polygon by a regular octagon (approximation of circumference) equate to enforcing a separation between parts.

Obtaining the polygonal boundary of the projection can be done as follows: find the sets of edges that separate facets with upward and downward normal; chain those edges to form closed oriented contours; finally, obtain the union of the regions bounded by those oriented contours projected into the plane, as illustrated in the upper part of Figure 1.

In many cases it may be required to have a minimum separation between adjacent parts. This can be enforced by replacing the part with its dilation by sphere of proper

radius. But since we will only deal with the projection, an easier approach is to apply, say dilation, to the projection. This procedure is illustrated in Figure 2.

The exact shape of the dilation of a polygon by a circumference will not be polygonal, since it will contain arcs of circumference. Still, as the desired gap will usually be much smaller than the typical dimensions of the part, it can be approximated by a regular polygon. The dilation of a polygon by another polygon can be computed by using the same concept of no-fit polygon that we will use for defining the non-overlapping condition, which will be introduced in the next section.

2.2 Densest linear packing of polygons

In order to find the optimal lattice packing, we must be able to efficiently determine when two objects overlap in any given direction. This is done via the concept of no-fit polygon (NFP) or the closely related phi-function 0-level [2].

The no-fit polygon represents the set of translations of a polygon P1 such that P1 and another polygon P0 just touch; it can also be viewed as the boundary of the region defined by the points $P=(r;\theta)$ such that the translations of polygon P1 by OP does intersect polygon P0. As the name implies, this boundary is polygonal. We will denote this as $NFP(P0,P1)$.

It is important to note that the NFP may be non-convex and even non-simple – i.e., composed of multiple disjoint polygons. The meaning of a non-simple NFP is that some admissible translations might require ‘lifting’ one of the polygons (see for instance figure 3 below). For some applications this positions might not be acceptable, as the two pieces become interlocked.

The NFP may also contain zero-area components: an isolated point or loops of parallel overlapping edges. Those correspond to tight fit of the two polygons such that they can not move (isolated point) or can only move in a linear path. When we consider only the boundary of the NFP without those degenerate cases, it is called the ‘regularized NFP’ [11].

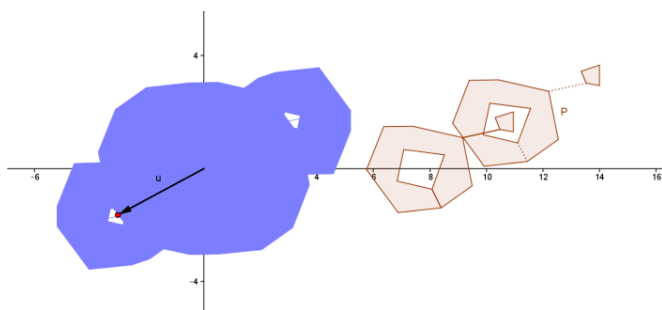


FIGURE 3: Interlocking and non-simple NFP. The holes in the NFP (on the left) represent non-overlapping translations of polygon P that can not be achieved without crossing P.

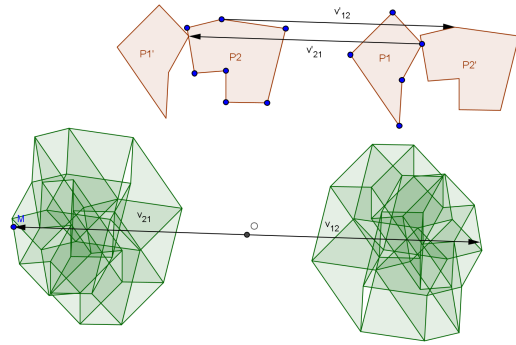


FIGURE 4: The NFP of polygons P1 with P2, and P2 with P1, and how they relate to the translations of the second polygon that do not overlap the first one. Point M on the no-fit polygon of P2 with P1 (bottom left) corresponds to the translation of P1 by $v_{21} = OM$ such that P1’ touches P2 but doesn’t intersect it. Moreover, considering a point in P2’ and sliding it around P1 will trace the outer contour of the NFP(P1,P2) (bottom right).

A mechanistic interpretation of the NFP, is that we bring the two polygonal regions to contact and slide one of them (P1) along the boundary of the other, tracing the position of a fixed point in P1 (see for instance Fig. 2 or Fig 4). This clearly shows that the boundary of the NFP is formed by the edges of the two original polygons. Indeed, if both polygons are convex, the sliding movement is along the full extent of each edge of both polygons, whereas if any of them is non-convex or non-simple, sliding traverses only part of some edges. Note that if any of the polygons is non-convex, not all edges will be necessarily present in the NFP.

The NFP is defined for two polygons, but for our application we shall consider the NFP of a polygon with itself, which we will denote by $NFP(P) = NFP(P, P)$. In this case, the NFP will obviously have central symmetry, since $NFP(P0, P1) = -NFP(P1, P0)$. Additionally, if P is convex, NFP(P) is also convex; and if P is a star polygon, the NFP(P) is also a star polygon. Further properties of the NFP can be found in [2].

The NFP can also be used to enforce the minimal gap between adjacent parts, as mentioned in previous section. Let us consider only a dilation of a polygon by a circle. If we consider the NFP of P with a circle of radius R, by the orbiting or sliding interpretation, using the center of the circle as the reference point, we can see that each point of the NFP will be at a distance at least R from P. So, if we take this NFP as the new polygon P’, the condition that $P' + v$ touches P’ implies that any point in $P + v$ is at a distance of at least 2R of a point in P.

2.3 Methods for computing the NFP

Efficient methods for computing the NFP of two polygons are mainly based on two approaches: the edge precedence and the Minkowski sum [2].

The first technique is simple and efficient, but only for the convex-convex or simple-convex cases. It is based on the sliding interpretation described above, and mainly uses the ordering of the slope of edges of both P1 and the reflection of P2 about the origin; for convex-convex case, the NFP is obtained by chaining together all edges in that order.

The second method, on the other hand, has been effectively used for arbitrary polygons, at the cost of computational complexity. The Minkowski sum, represented by \oplus , is defined for two sets of points S_a and S_b as the set of points $S = \{a + b \mid a \in S_a, b \in S_b\}$. If we consider S_a and S_b as the interior of P_a and P_b , we can define the Minkowski sum of two oriented polygons $P_a \oplus P_b$ as the boundary of $S_a \oplus S_b$. If we consider two edges $e_a = AA'$ and $e_b = BB'$, we may define their Minkowski sum as the parallelogram with edges e_{a+B} , e_{b+A} , $e_{a+B'}$ and $e_{b+A'}$. For instance, in figure 8b, the parallelogram T_{ij} is the Minkowsky sum of edges e_i and e_j .

The formal analysis of how the Minkowski sum is related to the NFP was done by Stoyan and Ponomarenko [9], who related the NFP to the boundary of $int(P_a) \oplus (int(-P_b))$. Ghosh showed that the NFP is determined only by the boundaries of both polygons, and developed methods for computing the NFP as a combination of the edges of the polygons, embodied in the edge precedence approach (see [10] and its references).

A full description of a general method for computing the NFP of two arbitrary polygons can be found in Benell and Song [2]. It is based on the fact that the NFP of a convex polygon with a simple polygon can be easily computed by the edge precedence method; thus, one of the polygons may be transformed in a convex polygon by replacing concave edges, for computing an approximate NFP; then the 'fake' edges are replaced by the original concave ones, and finally the real boundary of the forming the NFP is obtained by appropriate clipping of the edges. This is a fairly complex algorithm, and computationally expensive.

For this application, only the NFP of a polygon with itself is needed. Also, as stated previously, we will only consider the cases where the polygon is simple. As such, in our work we use a simplified version of the Minkowski sum technique, which is easier to describe and implement. The procedure is based on two operations: the union of two polygonal regions and the removal of cyclic components. For methods that can be used to perform these operations see [8].

S_1 and S_2 being the sets of vertices of each polygon, we may define the Minkowski sum of P1 and P2 as the union of all translations of P2 by \vec{OV} , for each V in S1.

$$P_1 \oplus P_2 = \bigcup_{V \in S_1} (P_2 + \vec{OV})$$

With this definition in mind, the first step of our procedure consists in computing

$$P_{1,2} = (P_1 \oplus (-P_2)) \cup ((-P_1) \oplus P_2)$$

(where $-P$ represents the reflection of P about the origin).

For a single polygon P , the procedure is further simplified, since

$$\begin{aligned} P_{12} &= (P \oplus (-P)) \cup ((-P) \oplus P) \\ &= (P \oplus (-P)) \cup -(P \oplus (-P)) \end{aligned}$$

Note that the obtained polygon $P_{1,2}$ is not the NFP, since it will possibly contain some additional edges, forming loops or cycles connected to the external boundary of $P_{1,2}$ (see figure 5b). Those cycles have a particular meaning: they represent the positions where some edges of one polygon cross an even but non-zero number of edges of the other. This implies that edge begins and ends on the outside of the other polygon, while still intersecting it; a translation vector with endpoint in the region limited by such loop will leave all vertices of P' outside of P.

In order to obtain the true NFP, regardless of whether we are computing the NFP for two or one polygon, we must remove the cycles. But if we are not interested in interlocking packings, we may as well remove all interior regions, that is, keep only the outer boundary of $P_{1,2}$.

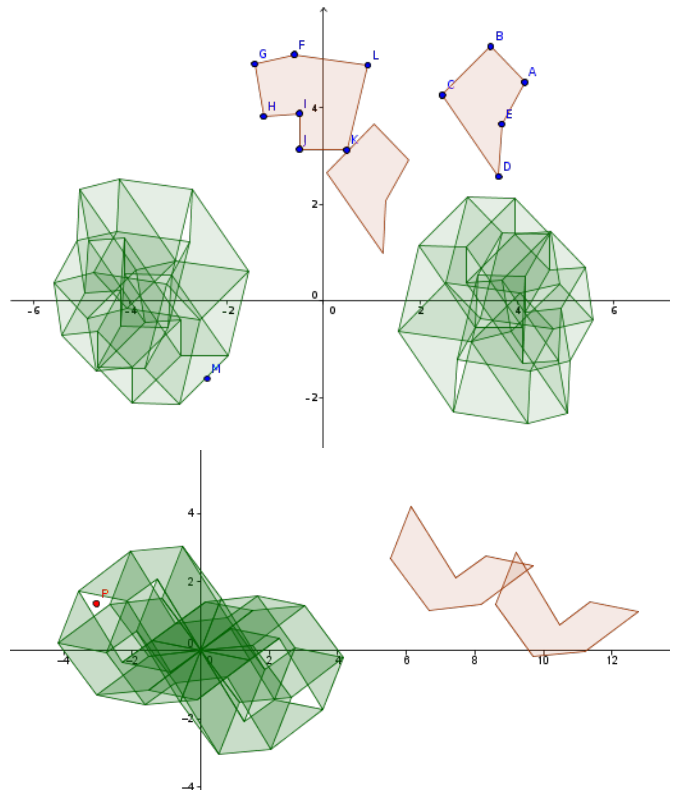


FIGURE 5: Computing the NFP of a) two polygons or b) one single polygon with itself. In b) the point P (red) lies in a 'hole' of P_{12} ; the corresponding translation, while leaving all vertices of the translated polygon outside of the original, still leads to intersection.

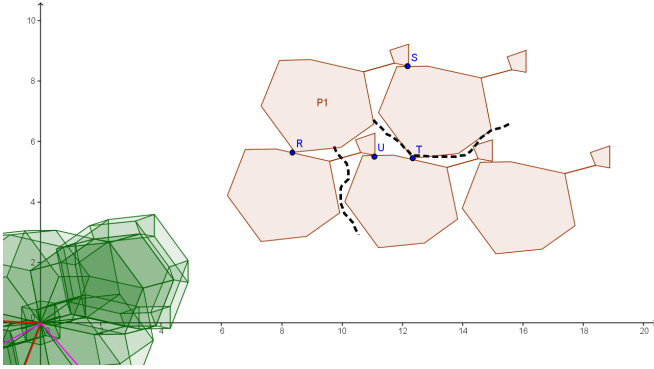


FIGURE 6: Conditions for a dense packing. If P is simply connected, and it touches $P + u$ and $P + v$, and does not intersect $P + (u \pm v)$, then it can not touch any other neighbor, and (u, v) generate a packing.

2.4 Finding feasible solutions to regular planar packing of polygons

Given the no-fit polygon (NFP) of polygon P with itself, we seek to determine all lattice bases, defined by vectors \vec{v}_1, \vec{v}_2 that produce non-intersecting tiling of P . This seems to imply that any linear combination of \vec{v}_1 and \vec{v}_2 with integer coefficients must be considered for possible intersection. We will see that for our purpose we need only consider a subset of regular packings, called dense packings. A dense packing is defined as a packing where adjacent polygons touch in at least two directions.

By considering the paths connecting two copies of P , we can see that the no-overlap conditions can be simplified for dense packings, provided that $\text{int}(P)$ is simply connected. Indeed, in two dimensions, provided that two immediate neighbor copies of P , i.e., with $(n, m) \in \{-1, 0, 1\}^2$, touch P , there is no need to consider other possible intersections. Figure 6 shows that the points $A, B, A' = A - v_1$ and $B' = B - v_2$ where P touches its immediate neighbors in a dense packing must be connected by a part of the frontier P ; since $B' + v_2 = B$, the immediate neighbors form a closed path, so that there is no possible path connecting P to higher order neighbors without entering a first-order neighbor.

The dashed lines in Figure 6 illustrate why this fails if $\text{int}(P)$ is not simply connected (the downward path), or if the packing is not dense.

Notice that the same reasoning does not apply in three dimensions. Since in that case the touch points do not ensure a closed surface, the higher order neighbors are not isolated.

From the above, we can use the NFP to restrict our search to only the lattice packings that are possibly optimal as follows: the bases formed by vectors \vec{v}_1, \vec{v}_2 with origin in the center of the NFP, endpoint on its boundary, and such that their sum and difference are not in the interior of the NFP:

$$\begin{aligned} \vec{v}_1^+ = OP, \vec{v}_2^+ = OQ, \text{ with } P, Q \in \partial NFP \\ \vec{v}_+ = \vec{v}_1 + \vec{v}_2, \vec{v}_- = \vec{v}_1 - \vec{v}_2, \text{ with } \{\vec{v}_+, \vec{v}_-\} \notin \text{int}(NFP) \end{aligned}$$

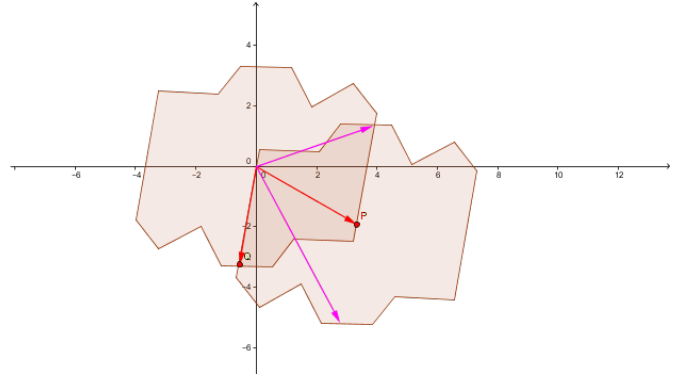


FIGURE 7: There is always a pair of vectors (u, v) such that $u + v$ lies on the NFP.

Note that given one vector ($v = OP$) lying on the boundary of the NFP, there is always at least one pair of vectors v_1 and v_2 also on the boundary of the NFP, such that their sum is $v = v_1 + v_2$ (see figure 7).

2.5 Determination of the densest packing

If the dimensions of the polygon P are much smaller than the dimensions of the container, the number of parts that can be placed on that container will depend more on maximizing the density of the packing than on minimizing the intersections with the container. In fact, the number of intersected parts will be broadly proportional to the perimeter of the container, whereas the number of non-intersected parts will be related to the ratio of the areas of the polygon and the container.

The density of a packing generated by vectors v_1 and v_2 is defined as the ratio of the area of the polygon to the area of the lattice cell (area of a parallelogram with sides v_1 and v_2); since the former is fixed, in order to maximize the density we must find the minimum of the area for the lattice cell (see Figure 8(a)).

Stoyan and Patsuk [1] prove that for any given non-dense packing, there exists always a dense packing (as defined in previous section) with no-lesser density. That means that we can restrict our search for the densest packing to only the cases where the two vectors lie on the NFP.

We present next an outline of the concepts behind the approach proposed in [1].

We start by considering the area of a parallelogram C , formed by two vectors v_1 and v_2 , with origin in the center of the NFP and endpoints on two edges, e_i and e_j (see figure 3). The area of C is given by the magnitude of the cross product of v_1 and v_2 .

The sum $v_+ = v_1 + v_2$ for all combinations of positions of both vectors define another parallelogram T_{ij} , whose sides are the edges e_i and e_j , translated by v_2 and v_1 , respectively. It can be defined also as the Minkowski sum of the two edges. The area of C decreases monotonically as each vector's endpoint moves along each edge, so it will have a minimum at a vertex of T_{ij} , that is, when both

vectors of the base are on a vertex of the corresponding edge. If all four combinations of vertices for a pair of edges are admissible, then the minimal area for bases generated from that pair is among the values obtained for each combination.

On the other hand, the region T_{ij} also represents the positions of $v_1 + v_2$. For a base (v_1, v_2) to be admissible, $v_1 + v_2$ must not be inside the NFP. Let e_k be one edge touched by $v_1 + v_2$, for some v_1 on e_i and v_2 on e_j . It should be noted that, for a simple polygon, if $v_i + v_j$ touches the NFP, then $v_i - v_j$ will always be outside the NFP; thus, these configurations are always feasible (if P is non-simple, this does not hold, and we would need to consider both the intersection of $v_1 + v_2$ and $v_1 - v_2$). So, the admissible solutions involving edges e_i and e_j correspond to a sub-region E_{ij} of T_{ij} , the boundary of which is composed from whole or parts of edges of the NFP.

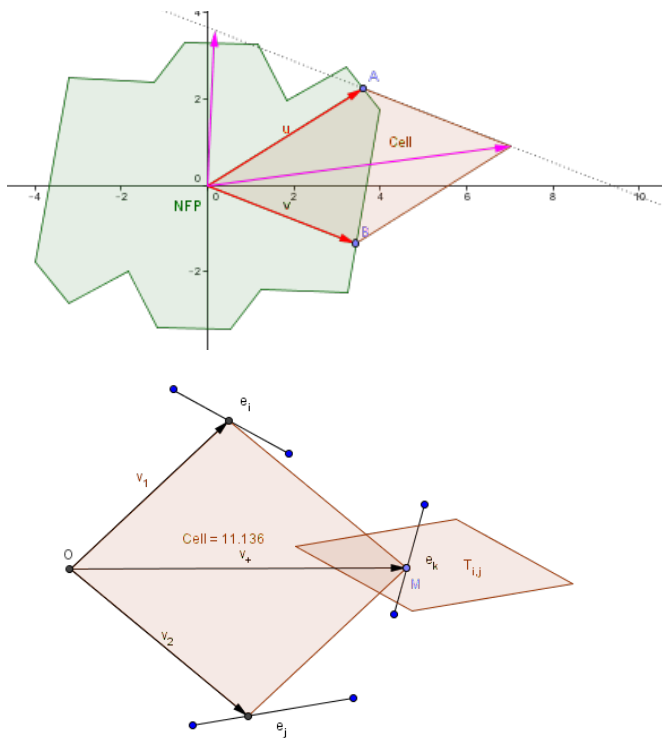


FIGURE 8: The possible lattices generated by vectors on a) a pair of edges, or b) three edges, are in correspondence with the set of points in T_{ij} or the intersection of e_k with T_{ij} , respectively.

The position of v_1 along e_i uniquely determines the positions of v_2 along e_j such that $v_1 + v_2$ touches e_k . We can thus easily find the position that minimizes the area for each combination (e_i, e_j, e_k) . If some edge of T_{ij} intersects the NFP at some edges e_k , an odd number of times, then at least one vertex of T_{ij} is not admissible. We can find the set of edges e_k that are partially or fully contained in T_{ij} . For each such edge, consider its supporting line; we want to find the minimal parallelogram touching three straight lines, at $P_i, P_j,$ and P_k , with a fixed fourth

vertex. Let t_k be the parametric coordinate of a point along edge e_k ; the corresponding t_i and t_j are uniquely determined by t_k . If the parametric representations of $P_i, P_j,$ and P_k , respectively t_i^*, t_j^* and t_k^* are all within $[0, 1]$, then t_i^* and t_j^* determine a local feasible minimum.

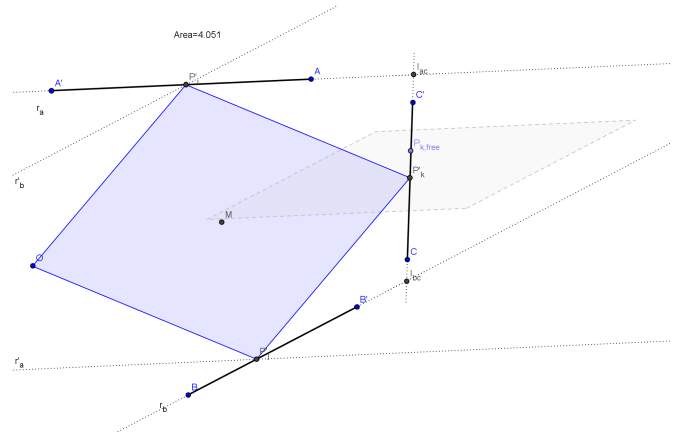


FIGURE 9: Geometric determination of the minimal parallelogram touching three edges of the NFP. P_k^* is the midpoint of segment $I_{ik}I_{jk}$. If all of P_i, P_j and P_k are inside the respective edges, the corresponding lattice generates a dense packing with maximum density for those edges.

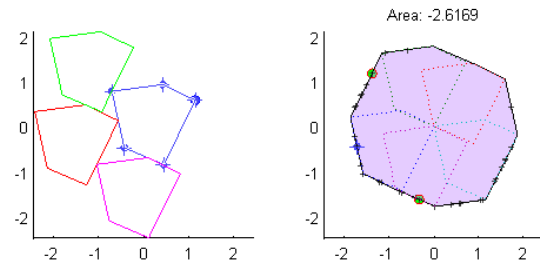


FIGURE 10: The plot on the right shows the NFP for the blue polygon on the left. On the NFP are plotted (+) the possible positions along each edge maximizing the lattice density. The maximal density lattice packing is also represented on the left plot, and its base vectors correspond to the green dots on the NFP.

Geometrically, this can be obtained by the following construction: given an origin O and three edges (e_i, e_j, e_k) , the point on edge k that gives the minimal area kissing parallelogram – parallelogram with one vertex on O and the other three on each given edge – is the midpoint of the intersections of the supporting lines for each of e_i and e_j with the supporting line of e_k (See Figure 9). Note that this is only true for edges with proper relative orientation; but it is always possible to take the proper oriented edges by replacing e_j by its symmetric $-e_j$, which also is part of the NFP, and generates the same lattice. Note also that the same procedure can be also applied to find the corresponding vertices in e_i or e_j , if we replace e_j (respectively e_i) by its symmetric; alternatively, if we take the midpoint M of OP_k , since it is the

center of symmetry of the parallelogram, the other two intersections can be obtained by intersecting the reflection of r_b about M with r_a .

The points (P_i, P_j, P_k) may or may not be all contained in the corresponding edges.

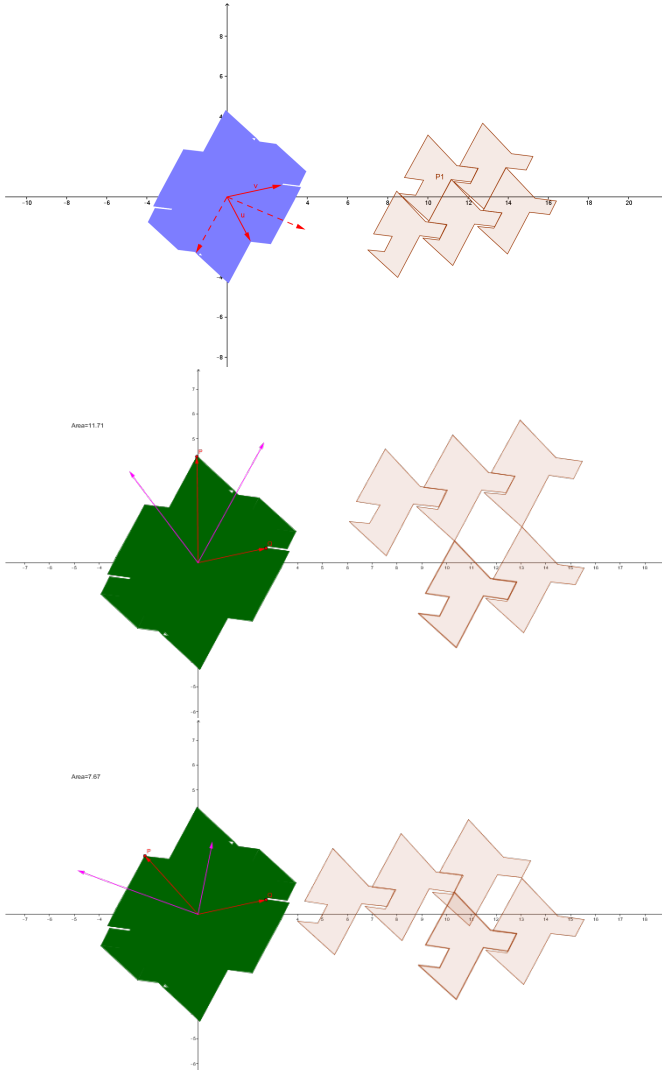


FIGURE 11: Optimal (a) and non-optimal (b,c) lattices.

The optimal solution can then be found by computing the area for all pairs of vertices of the NFP, and checking if each pair is admissible; computing the optimal position (if it is admissible) for each triple of edges, and the corresponding area; and searching for the minimum of the computed values.

2.6 The Containment Problem

Having determined a densest infinite packing of a polygon, we still have to consider if it leads to a maximal number of fully-contained copies for a given container size. In general, we have three translational degrees of freedom, plus the orientation, for the placement of the lattice in the container. In our specific application, however, the vertical translation is fixed by the requirement

that the objects be supported by the work plane. Similarly, the rotation of the lattice is restricted to the vertical axis, so that the preferred orientation is maintained.

Although this is a very important application case, there are very few published works on the packing problem which take into account the container.

If we are considering a bounded packing, we must take into account that the packing distance is in general smaller than the projection of the polygon perpendicular to the packing direction – that is, the polygon will extend beyond the 'unit' of the packing lattice. This 'protruding' part of the polygon can not however cross the bounds of the container, thus reducing the 'effective' container length available for the lattice. This is illustrated for a linear bounded packing in Figure 12.

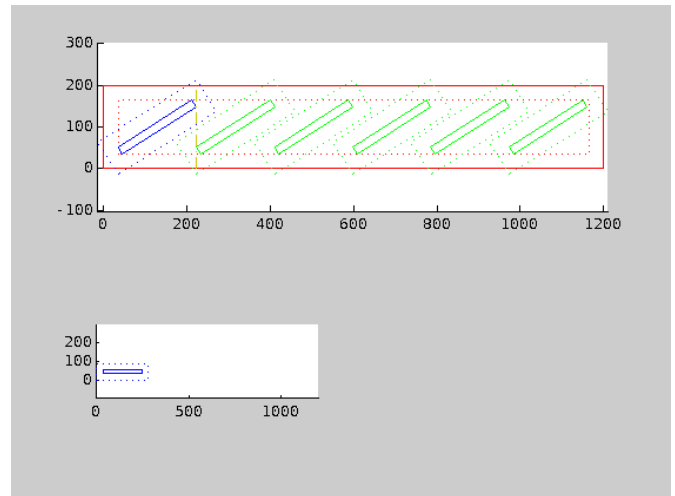


FIGURE 12: Linear bounded packing.

Although we have no closed form expression for the number of pieces contained in (or intersected by) a rectangular container as a function of the lattice position, we can establish upper and lower bounds. The upper bound is obtained by simply dividing the total area of the container by the area of the polygon. A possible lower bound reduces to the regular packing of spheres (or circles) that circumscribe the polyhedron (or polygon). This observation is useful in restricting the search-space for the optimal solution.

For a bounded packing, we will want to maximize the number of parts fully contained. Several heuristics may be applied. We may expect the densest unbounded packing to provide a near-optimal solution, and deal with both problems separately – finding a densest packing, and then finding the optimal placement of that packing relative to the container. We could also look for the densest rectangular packing, based on the assumption that a rectangular packing will be able to provide a better solution for the containment problem. Or we could consider the linear bounded packing, along one side of the container, followed by the packing of the obtained strip along the other direction.

In order to quickly produce near-optimal layouts, while retaining the ability to explore all of the problem space, including the three-dimensional cases, we developed an interactive application. The main window of this application is presented in Figure 13. After selecting an arbitrary 3D-part (described as an STL file) or parameterizing a parametric model (extruded part, swept part), the user can specify the vertical orientation and a margin around the part. The application then computes a densest lattice packing in the horizontal plane, which the user can freely modify to explore other dense or arbitrary (non-dense) packing. For each lattice selected, an optimal placement within the container is computed, which can also be manually adjusted, and the resulting number of parts is computed.

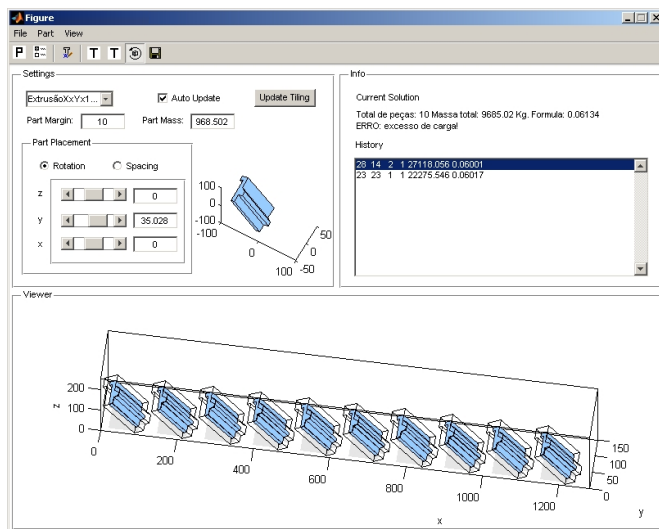


FIGURE 13: An interactive application for exploring feasible solutions to the dense regular planar packing of a polygon.

3 Conclusions

The published methods referenced above each have some limitations. The method by Stoyan and Patsuk [1] provides only a partial solution – limited to the densest lattice packing of a single polygon – but it gives an exact solution. On the other hand, the method proposed by Milenkovic [5] is very general, being able to deal with the placement of multiple different polygons. This generality comes at a cost: as he points out, his method only provides an approximation, albeit with a very tight bound, to the optimal solution.

The solution presented in this paper was specially designed for the particular case of maximizing the placement of congruent polyhedra in a parallelepiped container. It was obtained by the reduction, under suitable restrictions, to a two-dimensional case. The problem can then be solved by applying the method pre-

sented in [5]. Still, the performance and complexity of that method raises some issues. As an alternative to a fully automated solution, an interactive application was developed in which near-optimal solutions for the problem can be found with little effort. By an extensive analysis of the sub-problems we were able to combine known tools and heuristics allowing an efficient exploration in the problem-space. These tools may prove useful in finding solutions for related, less constrained, problems.

4 References

- [1] A method of optimal lattice packing of congruent oriented polygons in the plane, Y.G. Stoyan, V.N. Patsuk, *EJOR* 124 (2000) 204 – 216
- [2] A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums, J.A. Bennell, X. Song, *Computers & Operations Research* 35 (2008) 267 – 281
- [3] The geometry of nesting problems: A tutorial, J.A. Bennell, J.F. Oliveira, *EJOR* 184 (2008) 397–415
- [4] An improved typology of cutting and packing problems, G. Wascher et al., *EJOR* 183 (2007) 1109–1130
- [5] Densest translational lattice packing of non-convex polygons, V. Milenkovic, *Computational Geometry* 22 (2002) 205–222
- [6] Novas abordagens ao Posicionamento Periódico de Figuras Irregulares, M. T. Costa., PhD Thesis, Universidade do Porto, 2007.
- [7] Tools of mathematical modeling of arbitrary object packing problems, J. Bennell et al., *Ann Oper Res* (2010) 179: 343–368
- [8] A method for computing the symmetric difference of arbitrary regions in the plane, M.B. Gaspar and N. Martins-Ferreira (submitted)
- [9] Minkovsky sum and the hodograph of the dense allocation vector-function, Yu.G. Stoyan and L.D. Ponomarenko, *Reports of the Academy of Sciences of Ukrainian SSR, Ser.A(Russ)*, 1977, p.10
- [10] P. K. Ghosh, A unified computational framework for Minkowski operations, *Computers & Graphics*, Volume 17(4) (1993) 357–378.
- [11] A Simulated Annealing Based Algorithm with Collision Free Region for the Irregular Shape Packing Problem, A.K. Sat, T.C. Martins, M.S.G. Tsuzuki, preprints of the 18th IFAC World Congress, 2011.
- [12] Maximizing the placement of congruent polyhedra in a parallelepiped container, Gaspar, M. Belbut; Martins-Ferreira, N., *Numerical Analysis and Applied Mathematics ICNAAM 2012: International Conference of Numerical Analysis and Applied Mathematics. AIP Conference Proceedings, Volume 1479. AIP Conference Proceedings, Volume 1479, Issue 1, p.1588-1591*

Curvas geodésicas: um exemplo com resolução analítica.

by J. P. FATELO AND N. MARTINS-FERREIRA

ESTG, CDRSP
Instituto Politécnico de Leiria

Resumo Apresentamos um exemplo de uma superfície não trivial em \mathbb{R}^3 na qual as curvas geodésicas são encontradas analiticamente.

1 Introdução

Tal como explicado em [1], na maioria dos casos, as curvas geodésicas são obtidas com recurso a aplicações computacionais. Entre as superfícies com curvas geodésicas encontradas analiticamente e sem aproximações, destacam-se o caso da superfície esférica com a solução (*intuitiva*) correspondente aos "grandes círculos" e o caso do cilindro que, neste particular, não é muito diferente de um plano onde as geodésicas são retas. Neste artigo considera-se uma superfície de revolução que, mesmo não sendo trivial, permite uma determinação explícita, em termos de funções elementares, das suas curvas geodésicas.

Na secção 2, apresentamos a superfície considerada. Na secção 3, resolvemos as equações das geodésicas para este exemplo e explicitamos as soluções sujeitas a condições iniciais e, na secção 4, as soluções na presença de condições de fronteira.

As figuras expostas neste artigo foram geradas usando a aplicação Mathematica [2].

2 A superfície

Em geral, as superfícies de revolução em \mathbb{R}^3 podem ser parametrizadas da seguinte maneira

$$\vec{\sigma}(\alpha, \theta) = (r(\alpha) \cos \theta, r(\alpha) \sin \theta, z(\alpha)). \quad (1)$$

Neste artigo, vamos considerar o caso em que

$$r(\alpha) = \frac{1}{\sqrt{\alpha}},$$
$$z(\alpha) = \int_{\frac{1}{\sqrt[3]{4}}}^{\alpha} \sqrt{1 - \frac{1}{4x^3}} dx,$$

onde $\alpha \geq \frac{1}{\sqrt[3]{4}}$ e $\theta \in \mathbb{R}$. A função z pode ser expressa em termos da função gama Γ e da função hipergeométrica ${}_2F_1$ da seguinte maneira:

$$z(\alpha) = \frac{\sqrt{\pi} \Gamma(-\frac{1}{3})}{\sqrt[3]{4} \Gamma(\frac{1}{6})} + \alpha {}_2F_1\left(-\frac{1}{2}, -\frac{1}{3}, \frac{2}{3}, \frac{1}{4\alpha^3}\right).$$

Parte da superfície assim definida está representada na Fig. 1.

O parâmetro θ assume qualquer valor real e considera-se a seguinte interpretação: pontos com o mesmo valor de α mas com valores de θ que diferem por um múltiplo de 2π correspondem à mesma posição geométrica mas situam-se em *camadas* diferentes da superfície.

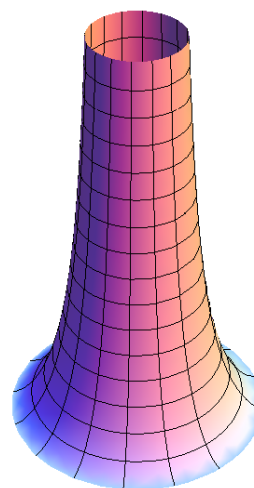


Fig. 1: Vista parcial da superfície em estudo.

Estas expressões de r e z foram escolhidas porque têm a seguinte propriedade:

$$r'^2 + z'^2 = 1.$$

Assim, o tensor métrico da superfície é uma matriz com a forma

$$g = \begin{pmatrix} r'^2 + z'^2 & 0 \\ 0 & r^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1/\alpha \end{pmatrix}.$$

A partir do tensor métrico, obtêm-se as equações das geodésicas [1]:

$$\ddot{\alpha} + \frac{1}{2\alpha^2} \dot{\theta}^2 = 0 \quad (2)$$

$$\ddot{\theta} - \frac{1}{\alpha} \dot{\alpha} \dot{\theta} = 0. \quad (3)$$

Uma geodésica na superfície é um conjunto de pontos parametrizado por $\vec{\sigma}(\alpha(t), \theta(t))$, onde t é um parâmetro real e $\alpha = \alpha(t)$ e $\theta = \theta(t)$ são soluções das equações (2) e (3). Estas equações formam um sistema autónomo uma

vez que não dependem explicitamente da variável independente t . Na secção seguinte, o sistema é resolvido com a utilização de leis de conservação, também conhecidas por *first integrals*.

3 Geodésicas a partir de um ponto dado

Nesta secção apresentam-se os cálculos necessários para obter as soluções dos caminhos geodésicos que começam num dado ponto inicial. Em primeiro lugar, observa-se que o sistema de equações (2) e (3) implica:

$$\frac{d}{dt} \left(\dot{\alpha}^2 + \frac{1}{\alpha} \dot{\theta}^2 \right) = 2\dot{\alpha} \left(\ddot{\alpha} + \frac{1}{2\alpha^2} \dot{\theta}^2 \right) = 0$$

e

$$\frac{d}{dt} \left(\frac{\dot{\theta}}{\alpha} \right) = \frac{1}{\alpha} \left(\ddot{\theta} - \frac{1}{\alpha} \dot{\alpha} \dot{\theta} \right) = 0.$$

Por conseguinte, as curvas geodésicas são soluções do seguinte sistema mais simples:

$$\dot{\alpha}^2 + \frac{1}{\alpha} \dot{\theta}^2 = v^2 \quad (4)$$

$$\dot{\theta} = \alpha l, \quad (5)$$

onde v e l são constantes arbitrárias. É possível relacionar (4) e (5), respetivamente, com a conservação da energia e do momento angular. De facto, estas quantidades são conservadas ao longo de um caminho geodésico uma vez que este é o caminho seguido por uma partícula livre, ou seja, sem forças a atuar sobre ela (para além daquelas que a mantêm na superfície).

Uma primeira classe de soluções corresponde ao caso $l = 0$, em que θ se mantém constante ao longo da geodésica, que é portanto um "meridiano" da superfície, e em que α varia linearmente com t .

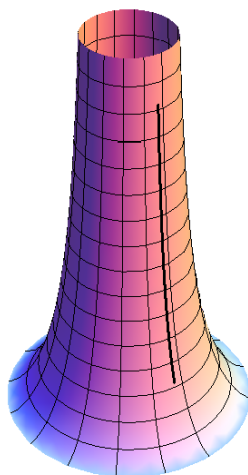


Fig. 2: Um meridiano.

Para determinar as soluções com $l \neq 0$, substitui-se (5) em (4), e obtém-se a equação

$$\dot{\alpha}^2 = v^2 - \alpha l^2,$$

donde

$$\dot{\alpha} = \pm \sqrt{v^2 - \alpha l^2}.$$

Considerando que são conhecidos os valores de α e θ em $t = 0$ (e notando $\alpha(0) = \alpha_0$ e $\theta(0) = \theta_0$), esta equação reduz-se a:

$$\sqrt{v^2 - \alpha l^2} = \sqrt{v^2 - \alpha_0 l^2} \mp \frac{l^2}{2} t.$$

Usando a notação $\beta_0 = \dot{\alpha}(0) = \pm \sqrt{v^2 - \alpha_0 l^2}$, a solução escreve-se

$$v^2 - \alpha l^2 = \left(\beta_0 - \frac{l^2}{2} t \right)^2,$$

ou seja,

$$\alpha(t) = \alpha_0 + \beta_0 t - \frac{l^2}{4} t^2. \quad (6)$$

Usando este resultado na equação (5), determina-se θ :

$$\theta(t) = \theta_0 + \alpha_0 l t + \frac{\beta_0 l}{2} t^2 - \frac{l^3}{12} t^3. \quad (7)$$

Em resumo, com as condições iniciais $\alpha(0) = \alpha_0$, $\theta(0) = \theta_0$, $\beta_0 = \dot{\alpha}(0)$ e $\omega_0 = \dot{\theta}(0) = \alpha_0 l$, as soluções de (2) e (3) são

$$\alpha(t) = \alpha_0 + \beta_0 t - \frac{\omega_0^2}{4\alpha_0^2} t^2$$

e

$$\theta(t) = \theta_0 + \omega_0 t + \frac{\beta_0 \omega_0}{2\alpha_0} t^2 - \frac{\omega_0^3}{12\alpha_0^3} t^3,$$

sempre que $\alpha_0 + \beta_0 t - \frac{\omega_0^2}{4\alpha_0^2} t^2 \geq \frac{1}{\sqrt[3]{4}}$.

Por exemplo, no caso $\alpha_0 = 1$, $\theta_0 = -2$, $\beta_0 = 4.25$ e $\omega_0 = 1.3$, a geodésica está representada na figura seguinte.

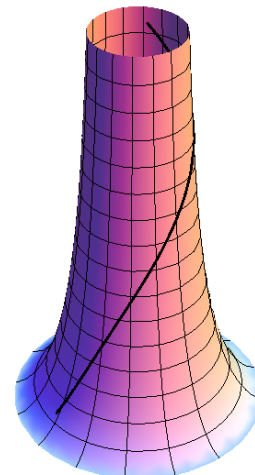


Fig. 3: Caminho geodésico 1.

Segue outro exemplo com $\alpha_0 = 2$, $\theta_0 = -1.8$, $\beta_0 = 4.5$ e $\omega_0 = 4$.

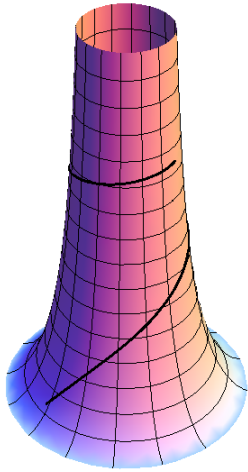


Fig. 4: Caminho geodésico 2.

Neste segundo exemplo, a curva geodésica não corresponde ao caminho mais curto entre as extremidades representadas. Mas isso não é uma contradição, conforme vamos esclarecer na próxima seção. Devido às condições iniciais, a curva não podia seguir diretamente para o ponto final, ou dito de outra forma, nas condições deste exemplo as extremidades da curva representadas estão em *camadas* diferentes da superfície o que obriga a curva a enrolar uma vez.

4 Geodésicas entre dois pontos

As soluções obtidas na seção anterior permitem obter os caminhos geodésicos a partir de um ponto inicial caracterizado pelos valores α_0 e θ_0 e conhecidos os dois parâmetros β_0 e l ou, de forma equivalente, β_0 e $\omega_0 = \alpha_0 l$. Na prática, um problema que surge com alguma frequência consiste na determinação de um caminho geodésico entre dois pontos dados. Para isso, é necessário conseguir escrever os dois parâmetros β_0 e l em função dos pontos extremos do caminho em causa. A Fig. 5 mostra um exemplo de dois pontos sobre um "paralelo" (conjunto de pontos da superfície com o mesmo valor de z) bem como o caminho geodésico que os liga.

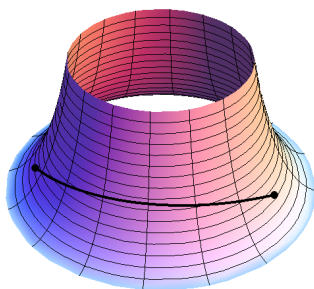


Fig. 5: Caminho geodésico entre dois pontos de um mesmo paralelo.

Os próprios paralelos não são curvas geodésicas uma vez que girar sobre paralelos de menor raio equivale a uma menor distância percorrida.

Supõe-se então que os valores de α e θ são conhecidos para dois valores de t . Sem perda de generalidade, a escolha destes valores pode ser $t = 0$ e $t = 1$:

$$\alpha(0) = \alpha_0 \quad ; \quad \theta(0) = \theta_0 \quad ; \quad \alpha(1) = \alpha_1 \quad ; \quad \theta(1) = \theta_1.$$

Uma escolha diferente de $t = 1$, correspondente ao segundo ponto dado, apenas altera a velocidade com que o caminho geodésico é percorrido mas não o *trilho* na superfície. A partir de (6) e (7), obtêm-se as relações

$$\begin{aligned} \alpha_1 &= \alpha_0 + \beta_0 - \frac{l^2}{4} \\ \theta_1 &= \theta_0 + \left(\alpha_0 + \frac{\beta_0}{2} \right) l - \frac{l^3}{12}, \end{aligned}$$

donde se conclui que:

$$\beta_0 = \alpha_1 - \alpha_0 + \frac{l^2}{4} \quad (8)$$

$$l^3 + 12(\alpha_0 + \alpha_1)l + 24(\theta_0 - \theta_1) = 0. \quad (9)$$

Esta equação de terceiro grau é incompleta e pode resolver-se através da mudança de variável

$$l = x - 4 \frac{\alpha_0 + \alpha_1}{x},$$

que produz uma equação quadrática em x^3 . A solução é:

$$\begin{aligned} l &= \sqrt[3]{12(\theta_1 - \theta_0) + 4\sqrt{9(\theta_1 - \theta_0)^2 + 4(\alpha_0 + \alpha_1)^3}} \\ &+ \sqrt[3]{12(\theta_1 - \theta_0) - 4\sqrt{9(\theta_1 - \theta_0)^2 + 4(\alpha_0 + \alpha_1)^3}}. \end{aligned} \quad (10)$$

Usando (8) e (9), as equações (6) e (7) podem agora ser escritas em termos de α_0 , α_1 , θ_0 e θ_1 . Mantendo a notação (10) por conveniência, o resultado é:

$$\alpha(t) = \alpha_0(1-t) + \alpha_1 t + \frac{l^2}{4} t(1-t) \quad (11)$$

$$\begin{aligned} \theta(t) &= \theta_0(1-t) + \theta_1 t + \frac{l}{2}(\alpha_0 - \alpha_1)t(1-t) \\ &+ \frac{l^3}{24} t(1-t)(2t-1). \end{aligned} \quad (12)$$

Estão reunidas as condições para voltar ao aparente paradoxo da Fig. 4 onde a curva representada, apesar de ser parte de uma geodésica, não é o caminho mais curto entre as suas extremidades. Para exemplificar a análise, considere os pontos $A = \vec{\sigma}(2, -1.8)$ e $B = \vec{\sigma}(5.5, 6)$. Usando as equações (11) e (12) com $\alpha_0 = 2$, $\theta_0 = -1.8$, $\alpha_1 = 5.5$ e $\theta_1 = 6$, obtêm-se a curva geodésica entre A e B representada na Fig. 6.

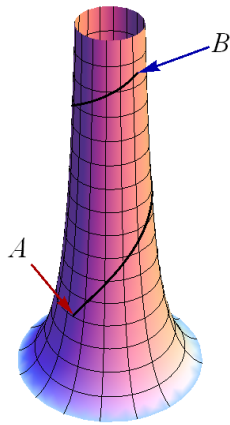
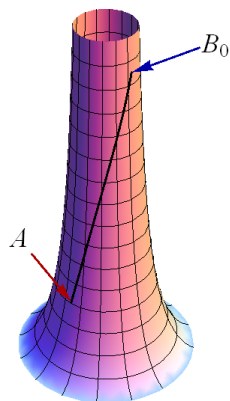
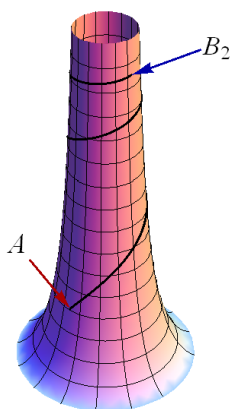


Fig. 6: Caminho geodésico entre A e B.

Fig. 7: Caminho geodésico entre A e B_0 .Fig. 8: Caminho geodésico entre A e B_2 .

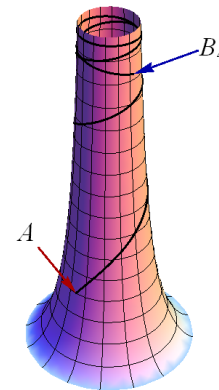
Observa-se que $\theta_1 - \theta_0 = 7.8$, ou seja, superior a 2π . O que significa que A e B não estão situados na mesma camada da superfície. O ponto $B_0 = \vec{\sigma}(5.5, 6 - 2\pi)$ está situado na mesma posição da superfície de B e na mesma camada de A. Usando as equações (11) e (12) com $\alpha_0 = 2$, $\theta_0 = -1.8$, $\alpha_1 = 5.5$ e $\theta_1 = 6 - 2\pi$, obtém-se a curva geodésica entre A e B_0 representada na Fig. 7.

Agora sim, foi encontrado o caminho mais curto entre as duas posições na superfície.

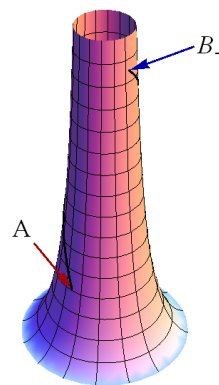
Podemos também procurar caminhos geodésicos entre A e outros pontos na mesma posição de B mas

em camadas diferentes. A curva geodésica entre A e $B_2 = \vec{\sigma}(5.5, 6 + 2\pi)$ está representada na Fig. 8.

A Fig. 9 representa a curva geodésica entre A e $B_4 = \vec{\sigma}(5.5, 6 + 6\pi)$

Fig. 9: Caminho geodésico entre A e B_4 .

enquanto a Fig. 10 representa a curva geodésica entre A e $B_{-1} = \vec{\sigma}(5.5, 6 - 4\pi)$.

Fig. 10: Caminho geodésico entre A e B_{-1} .

5 Conclusão

Este estudo resulta de um trabalho ainda em curso sobre a possibilidade de axiomatizar a noção de caminho geodésico através de uma operação binária que a cada dois pontos associa o ponto médio do percurso geodésico que os une. Neste caso a operação binária seria dada pelas equações (11) e (12) com $t = \frac{1}{2}$. O exemplo aqui ilustrado serviu o propósito de testar os axiomas de uma tal estrutura algébrica.

Bibliografia

- [1] J. P. Fatelo, Nelson Martins-Ferreira, *Curvas Geodésicas em superfícies*, Scripta-Ingenia, June 2014, No. 2, 22-25.
- [2] Wolfram Research, Inc., *Mathematica*, Version 9.0, Champaign, IL (2012).