A Mathematica Package for Visualizing Objects Inmersed in $\mathbb{R}^4$

Ricardo Velezmoro[1], Robert Ipanaqué[1(✉)], and Josel A. Mechato[2]

[1] Universidad Nacional de Piura, Urb. Miraflores s/n, Castilla, Piura, Peru
{rvelezmorol,ripanaquec}@unp.edu.pe
[2] Universidad Privada Antenor Orrego,
Av. Los Tallanes Zona Los Ejidos s/n, Piura, Peru
jmechatod1@upao.edu.pe

Abstract. Visualizing objects in a space of a certain dimension is quite useful to facilitate the understanding of the mathematical concepts linked to such objects, through the geometric interpretation of them. This paper describes a new Mathematica package, `4DSketches`, for visualizing objects immersed in $\mathbb{R}^4$ (points, lines, arrows, curves, surfaces, solids and complex surfaces) using 3D models as an extension of the 2D models, already established, for visualizing objects immersed in $\mathbb{R}^3$. To simulate the change of views in $\mathbb{R}^4$ rotational matrices are used. In addition, our outputs are consistent with Mathematica's notation. To show the performance of the package, several illustrative and interesting examples are described.

Keywords: 3D models to viewing objects in $\mathbb{R}^4$ · Plot surfaces in $\mathbb{R}^4$ · Plot solids in $\mathbb{R}^4$ · Plot complex surfaces

1 Introduction

The visualization of hiper-objects in spaces of dimension greater than three was proposed by Noll [8]. Then we find the works of Hoffman and Zhou [6,15] in which the visualization of surfaces in four-dimensional space is described, as well as some applications are provided. Subsequently, several articles related to the visualization of objects immersed in four-dimensional space have been written [2,3,5,11–13].

3D models to visualize objects immersed in the four-dimensional Euclidean space can be considered as an extension of the 2D models to visualize objects immersed in the three-dimensional Euclidean space. These models can be defined by the origin of $\mathbb{R}^3$, an apocryphal basis and a submersion from $\mathbb{R}^4$ in $\mathbb{R}^3$. Mathematica can be used to perform the various calculations [14]. In addition, since we do not have a "spatial" screen, we only have a flat screen (the computer screen), we can take advantage of Mathematica's capabilities to simulate the graphs in $\mathbb{R}^3$. To simulate different points of view in $\mathbb{R}^4$ we can use rotation matrices, similar to how they are used in $\mathbb{R}^3$ [6,8,15].

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11624, pp. 479–493, 2019.
https://doi.org/10.1007/978-3-030-24311-1_35

This paper presents a new Mathematica package, `4DSketches`, which incorporates commands for visualizing points, lines, arrows, curves, surfaces and solids inmersed in $\mathbb{R}^4$. In addition, commands have been incorporated for visualizing graphs of complex functions, of complex variables, defined in explicit parametric and polar forms. The commands provides the user with a highly intuitive, mathematical-looking output consistent with Mathematica's notation and syntax [7].

The structure of this paper is as follows: Sect. 2 introduce some mathematical definitions about the submersion, apocryphal basis and an m-dimensional model for visualizing objects inmersed in $\mathbb{R}^{m+1}$. For the sake of illustration, some 2D and 3D models are also briefly described in this section. Then, Sect. 3 introduces the new Mathematica package, `4DSketches`, and describes the commands implemented within. The performance of the package is also discussed in this section by using some illustrative examples to plot points, lines, arrows, curves, surfaces, solids and complex surfaces. Finally, Sect. 4 closes with the main conclusions of this paper.

2 Mathematical Preliminaries

Definition 1. A mapping is called an *submersion* if it has the same local type as a canonical projection of Euclidean spaces [9].

Definition 2. An *apocryphal basis* $\hat{\mathcal{B}}$ of $\mathbb{R}^m$ consists of a set of $m+1$ tangent vectors to $\mathbb{R}^m$ such tha rank $\left(\hat{\mathcal{B}}\right) = m$.

Definition 3. An *m-dimensional model* for visualizing objects inmersed in $\mathbb{R}^{m+1}$ consists of a triad $\{O, \hat{\mathcal{B}}, \varphi\}$; where O is the origin of $\mathbb{R}^m$, $\hat{\mathcal{B}}$ is an apocryphal basis of $\mathbb{R}^m$, and φ is a submersion from $\mathbb{R}^{m+1}$ to $\mathbb{R}^m$.

Previous definitions will be considered only for values $m = 2$ and $m = 3$.

2.1 2D Models to Visualize Objects in $\mathbb{R}^3$

In practice, to chance, we use two-dimensional models for visualizing objects immersed in $\mathbb{R}^3$. In effect, in general, we have

$$O = \{0,0\}, \quad \hat{\mathcal{B}} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\} \quad \text{and} \quad \varphi(\mathbf{p}) = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{pmatrix}.$$

To plot a point in $\mathbb{R}^3$ the submersion φ must be applied and then proceed to plot the point. For example, for the point $\mathbf{p} = (p_1, p_2, p_3)$ in $\mathbb{R}^3$ we will graph

$$\mathbf{q} = \varphi(\mathbf{p}) = (e_{11}p_1 + e_{21}p_2 + e_{31}p_3,\ e_{12}p_1 + e_{22}p_2 + e_{32}p_3)$$

in $\mathbb{R}^2$. And so for any triad. In particular, we could fixing the model

$$O = \{0,0\}, \quad \hat{\mathcal{B}} = \left\{ \frac{3}{5\sqrt{2}}(-1,-1), (1,0), (0,1) \right\} \quad \text{and}$$

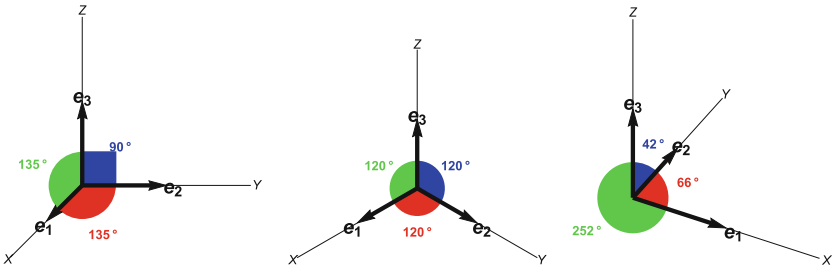

Fig. 1. 2D models to visualize objects in $\mathbb{R}^3$. Dimmetric (left), symmetric (center) and trimmetric (right).

$$\varphi(\mathbf{p}) = \left(p_2 - \frac{3p_1}{5\sqrt{2}},\ p_3 - \frac{3p_1}{5\sqrt{2}} \right) .$$

that will lead us to the preferred model by a large majority of authors of mathematical texts. This model is known as the *dimmetric model* (Fig. 1).

We can also choose $\hat{\mathcal{B}} = \left\{ \frac{1}{2}\left(-\sqrt{3}, -1\right), \frac{1}{2}\left(\sqrt{3}, -1\right), (0,1) \right\}$ (*symmetric model*) or the apocryphal basis that allows simulating the *Mathematica* default view point

$$\hat{\mathcal{B}} = \left\{ \frac{11}{10}\left(\cos(18°), -\sin(18°)\right), \frac{3}{4}\left(\sin(42°), \cos(42°)\right), (0,1) \right\},$$

this set gives rise to the *trimmetric model* (Fig. 1). For example, Fig. 2 show the graphs of some objects immersed in $\mathbb{R}^3$ obtained by means of the dimmetric model.

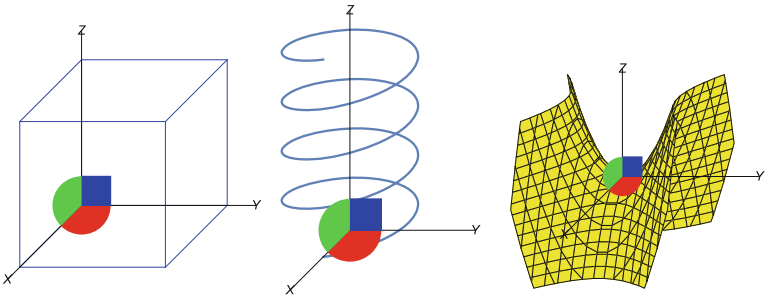

Fig. 2. Visualization of some objects immersed in $\mathbb{R}^3$ with a 2D dimmetric model. Cube (left), curve (center) and surface (right).

The graphs obtained by any model lack movements in $\mathbb{R}^3$, only admit movements in $\mathbb{R}^2$. To simulate movements in $\mathbb{R}^3$ it is necessary to use rotation matrices. For example, the product

$$R = R_z(\gamma)\, R_y(\beta)\, R_x(\alpha)\,,$$

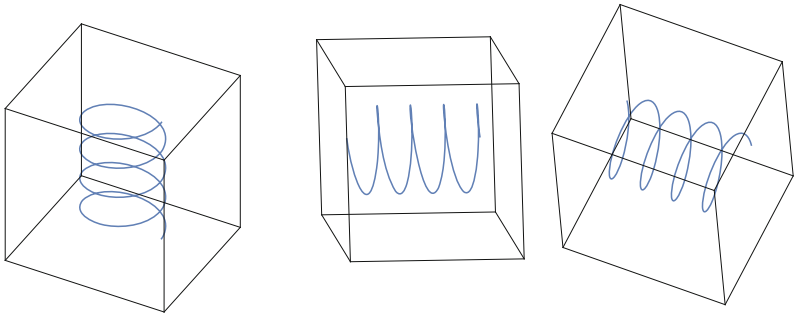

Fig. 3. Simulation of 3D rotations of a curve in $\mathbb{R}^3$.

serves to represent a rotation about axes z, y and x. In this way, starting from any model (which initially only allows movements in $\mathbb{R}^2$), we could simulate interactive movement in $\mathbb{R}^3$. Figure 3 was obtained after applying rotations to the coordinate functions of the curve starting from the trimmetric model.

2.2 3D Models to Visualize Objects in $\mathbb{R}^4$

In general, a 3D model for visualizing objects immersed in the $\mathbb{R}^4$ is given by

$$O=\{0,0,0\},\quad \hat{\mathcal{B}}=\{\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3,\mathbf{e}_4\}\quad\text{and}\quad \varphi(\mathbf{p})=\begin{pmatrix}p_1\ p_2\ p_3\ p_4\end{pmatrix}\begin{pmatrix}e_{11}\ e_{12}\ e_{13}\\ e_{21}\ e_{22}\ e_{23}\\ e_{31}\ e_{32}\ e_{33}\\ e_{41}\ e_{42}\ e_{43}\end{pmatrix}.$$

To plot a point in $\mathbb{R}^4$ the submersion φ must be applied and then proceed to plot the point. For example, for the point $\mathbf{p}=(p_1,p_2,p_3,p_4)$ in $\mathbb{R}^4$ we will graph

$$\begin{aligned}\mathbf{q}=\varphi(\mathbf{p})=(&e_{11}p_1+e_{21}p_2+e_{31}p_3+e_{41}p_4,\\ &e_{12}p_1+e_{22}p_2+e_{32}p_3+e_{42}p_4,\\ &e_{13}p_1+e_{23}p_2+e_{33}p_3+e_{43}p_4)\end{aligned}$$

in $\mathbb{R}^3$. And so for any quatern.

In particular, we could fixing the model

$$O=\{0,0,0\},\quad \hat{\mathcal{B}}=\left\{\tfrac{3}{5\sqrt{3}}\left(-1,-1,-1\right),(1,0,0),(0,1,0),(0,0,1)\right\}\quad\text{and}$$

$$\varphi(\mathbf{p})=\left(p_2-\frac{3p_1}{5\sqrt{3}},\ p_3-\frac{3p_1}{5\sqrt{3}},\ p_4-\frac{3p_1}{5\sqrt{3}}\right).$$

We will call to this model the *trimmetric-trimmetric model* (Fig. 4).

We can also choose

$$\hat{\mathcal{B}}=\left\{\left(-\tfrac{\sqrt{2}}{3},-\sqrt{\tfrac{2}{3}},-\tfrac{1}{3}\right),\left(-\tfrac{\sqrt{2}}{3},\sqrt{\tfrac{2}{3}},-\tfrac{1}{3}\right),\left(\tfrac{2\sqrt{2}}{3},0,-\tfrac{1}{3}\right),(0,0,1)\right\}.$$

We will call the model obtained from these set the *symmetric model* (Fig. 5). Or, we could also choose

$$\hat{\mathcal{B}} = \left\{ \tfrac{1}{\sqrt{26}}\,(5,0,-1)\,,\, \tfrac{1}{5\sqrt{17}}\,(0,16,-4)\,,\, \tfrac{7}{10\sqrt{22}}\,(3,3,2)\,,\, (0,0,1) \right\}\,.$$

We will call the model obtained from these set the *asymmetric model* (Fig. 6). For example, Figs. 7 and 8 show the graphs of some objects immersed in $\mathbb{R}^4$ obtained using the trimmetric-trimmetric model.

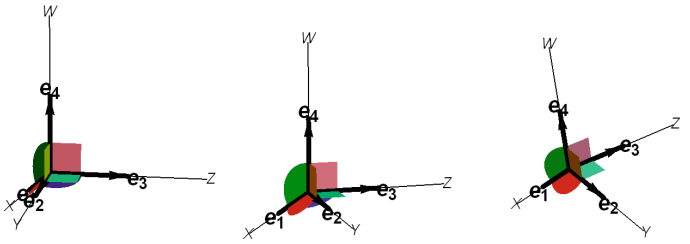

Fig. 4. 3D trimmetric-trimmetric model to visualize objects in $\mathbb{R}^4$.

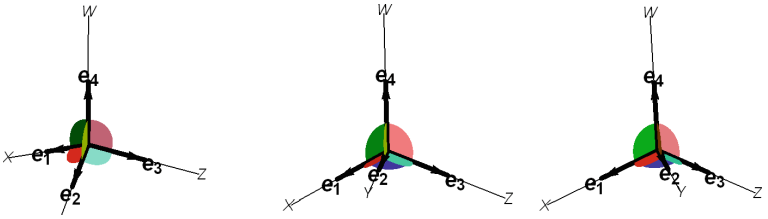

Fig. 5. 3D symmetric model to visualize objects in $\mathbb{R}^4$.

The graphs obtained by any model lack movements in $\mathbb{R}^4$, only admit movements in $\mathbb{R}^3$. To simulate movements in $\mathbb{R}^4$ it is necessary to use rotation matrices. For example, the product

$$R = R_{zw}(\psi)\, R_{yw}(\phi)\, R_{yz}(\delta)\, R_{xw}(\gamma)\, R_{xz}(\beta)\, R_{xy}(\alpha)\,,$$

serves to represent an rotation in the planes zw, yw, yz, xw, xz and xy. In this way, starting from any model (which initially only allows movements in $\mathbb{R}^3$), we could simulate interactive movement in $\mathbb{R}^4$. Figure 9 was obtained after applying rotations to the coordinate functions of the surface starting from the trimmetric-trimmetic model.

3D models to visualize geometric objects of $\mathbb{R}^4$ require a "spatial" screen, that is, a parallelepiped box in which three perpendicular axes can be located

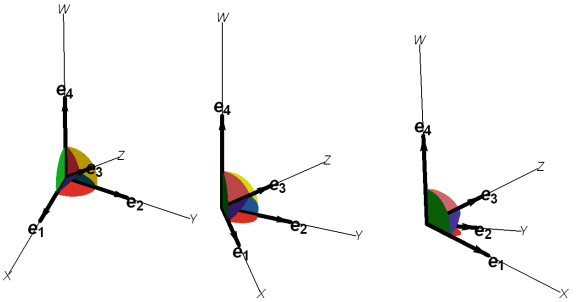

Fig. 6. 3D asymmetric model to visualize objects in $\mathbb{R}^4$.

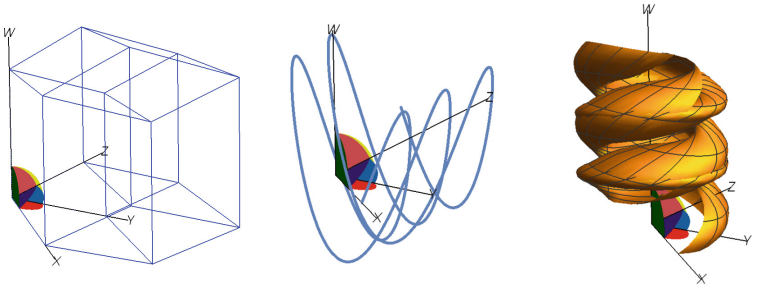

Fig. 7. Visualization of tesseract (left), curve (center) and surface (right) with a 3D asymmetric model.

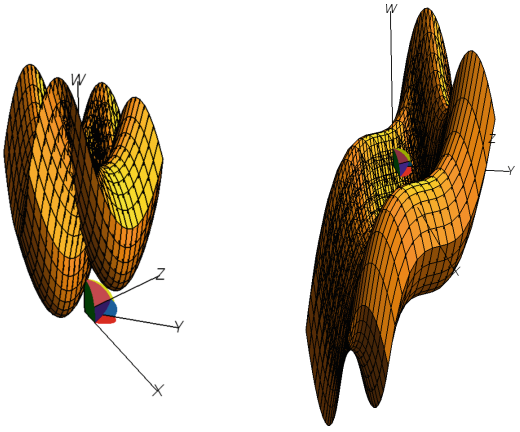

Fig. 8. Visualization of solids with a 3D asymmetric model.

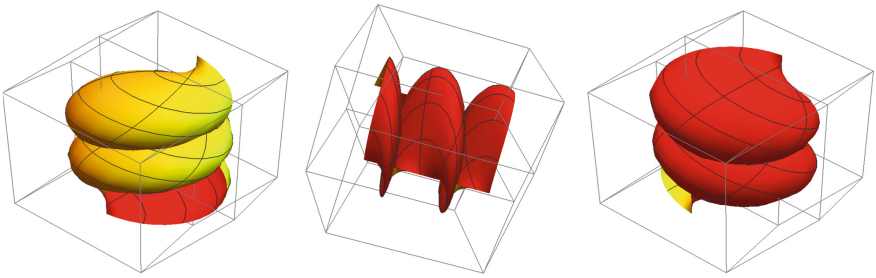

Fig. 9. Simulation of 4D rotations of the surface $S(u, v) = (\cos(u)\cos(v), \sin(u)\cos(v), \sin(v), \frac{u+v}{8})$.

[10], for example, one another and another axis on the extension of the diagonal of the cube, whose sides measures 1, located in the first octant (similar to the most common of the 2D models). Figure 10 clarifies this idea. Since we do not have these types of screens, we will use *Mathematica* since this system of computational algebra has wide graphic capabilities, so that the observer has the clear idea of seeing in $\mathbb{R}^3$ the graphics obtained the standard built-in commands, but in reality they are seen on a 2D screen (computer's flat screen).

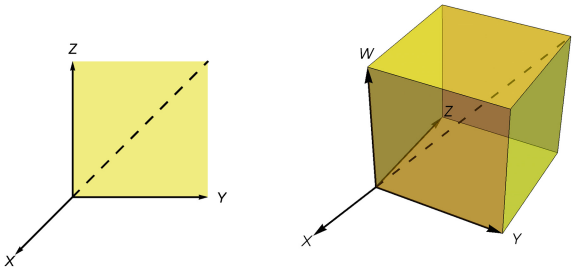

Fig. 10. The most common 2D model (left) and a simile 3D model (right).

3 The Package `4DSketches`: Some Illustrative Examples

This section describes some examples of the application of this package. Firstly, we load the package:

```
<<4DSketches.m
```

The commands incorporated in this package are:

```
Graphics4D, Plot4D, ParametricPlot4D, ComplexPlot,
ComplexParametricPlot, ComplexPolarPlot
```

Here we show a mini program to calculate the vector resulting from the sum of a set of vectors in $\mathbb{R}^4$.

```
Addition4D[v : {{_, _, _, _} ..}, p_: {0, 0, 0, 0}] :=
 Module[{n = Length[v], aux, vs, res},
  aux = FoldList[Plus, p, v];
  vs = Partition[aux, 2, 1];
  res = {aux[[1]], aux[[n + 1]]};
  {{Red, Tooltip[Arrow[Tube[#]], Last[#] - First[#]] & /@ vs},
   {Blue, Tooltip[Arrow[Tube[res]], Last[res] - First[res]]},
   {AbsolutePointSize[5], Tooltip[Point[p], p]}}]
```

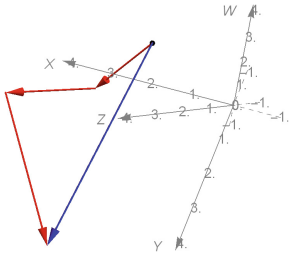

Fig. 11. Graphical visualization of the sum of vectors in $\mathbb{R}^4$ with `Graphics4D`.

In the following way we test the performance of the mini program and with the `Graphics4D` command we obtain the geometric interpretation of the sum of vectors in $\mathbb{R}^4$.

```
p = {-1, -1, 0, 0};
u = {1, 1, 0, -1}; v = {1, 1, 1, 1}; w = {-2, 3, 0, -1};
Graphics4D[Addition4D[{u, v, w}, p], Axes -> True,
 AxesLength -> {-1, 4}]
```

See Fig. 11.

With the `Plot4D` command it is possible to visualize the graph of the solid $M : w = x^2 + y^2 + z^2$ and the two solids together $M : w = x^2 + y^2 + z^2$ and $N : w = x^2 + y^2 - z^2$.

```
Plot4D[x^2 + y^2 + z^2, {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
 AxesLength -> {-1, 2}]
```

See Fig. 12 *(left).*

```
Plot4D[{x^2 + y^2 + z^2, x^2 + y^2 - z^2}, {x, -1, 1}, {y, -1, 1},
 {z, -1, 1}, AxesLength -> {-2, 2.5}, PlotStyle -> {Directive[Red,
  Opacity[0.6]], Directive[Blue, Opacity[0.6]]}]
```

See Fig. 12 *(right).*

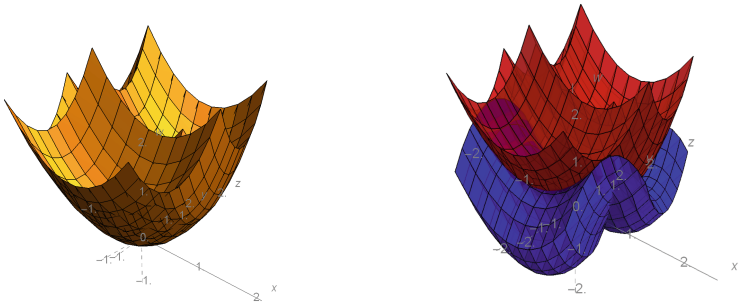

Fig. 12. Graphical visualization of solids in $\mathbb{R}^4$ with `Plot4D`.

Below are the sentences to obtain the graphs of a single curve and two curves together in $\mathbb{R}^4$ using the `ParametricPlot4D` command.

```
ParametricPlot4D[{Cos[u/Sqrt[5]], Sin[u/Sqrt[5]],
   Cos[2 u/Sqrt[5]], Sin[2 u/Sqrt[5]]}, {u, 0, 5 Pi},
  AxesLength -> {-1, 2}]
```

See Fig. 13 *(left).*

```
ParametricPlot4D[{{Cos[u/Sqrt[5]], Sin[u/Sqrt[5]],
   Cos[2 u/Sqrt[5]], Sin[2 u/Sqrt[5]]},
  {Cos[u/Sqrt[5]], Sin[u/Sqrt[5]],
   Cos[2 u/Sqrt[5]], -Sin[2 u/Sqrt[5]]}}, {u, 0, 5 Pi},
 AxesLength -> {-1, 2}]
```

See Fig. 13 *(right).*

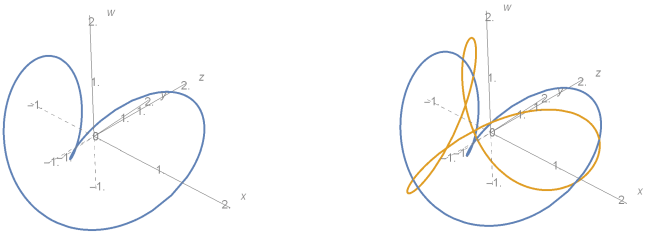

Fig. 13. Graphical visualization of curves in $\mathbb{R}^4$ with `ParametricPlot4D`.

Using the following sentences it is possible to obtain the graphs of a single surface and two surfaces together in $\mathbb{R}^4$ by means of the `ParametricPlot4D` command.

```
ParametricPlot4D[{u, v, u^2 + v^2, -2 u v}, {u, -1, 1},
  {v, -1, 1}, AxesLength -> {-2, 3}]
```

See Fig. 14 *(left).*

```
ParametricPlot4D[{{u, v, u^2 + v^2, -2 u v},
  {u^2 + v^2, v, -2 u v, u}},
 {u, -1, 1}, {v, -1, 1}, AxesLength -> {-2, 3}]
```

See Fig. 14 *(right).*

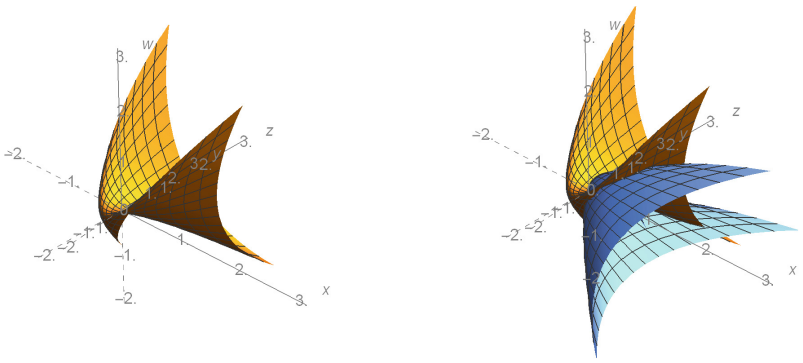

Fig. 14. Graphical visualization of surfaces in $\mathbb{R}^4$ with `ParametricPlot4D`.

The following sentences make it possible to obtain the graphs of a single solid and of two solids together in $\mathbb{R}^4$ using the `ParametricPlot4D` command.

```
ParametricPlot4D[{u^2 - v^2, 2 u v, w, u}, {u, -1, 1},
  {v, -1, 1}, {w, -1, 1}, PlotPoints -> 15,
 PlotStyle -> Directive[Opacity[0.8]], AxesLength -> {-3, 3}]
```

See Fig. 15 *(left).*

```
ParametricPlot4D[ {{u^2 - v^2, 2 u v, w, u},
 {u^2 - w^2, 2 u w, v, w}}, {u, -1, 1}, {v, -1, 1}, {w, -1, 1},
 PlotPoints -> 15, PlotStyle -> {Directive[Opacity[0.8], Red],
  Directive[Opacity[0.8], Green]}, AxesLength -> {-3, 3}]
```

See Fig. 15 *(right).*

Since the graph of a complex function $F : \mathbb{C} \to \mathbb{C}$ is a surface that supports a local parameterization by means of a Monge patch of the form

$$S(x, y) = (\Re(z), \Im(z), \Re(F(z)), \Im(F(z))) \,,$$

where $z = x + iy$, then it is possible to use the proposed models to obtain the graphs of such functions. In the case that the function is given in parametric

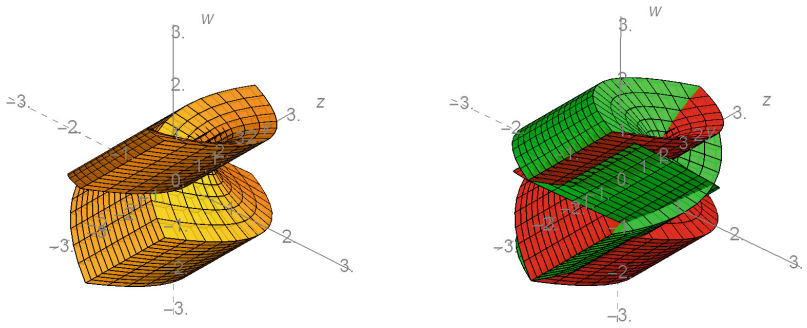

Fig. 15. Graphical visualization of solids with `ParametricPlot4D`.

form, ie $\alpha : \mathbb{C} \to \mathbb{C}^2$ such that $\alpha(z) = (\alpha_1(z), \alpha_2(z))$, then we express it in the form

$$S(x, y) = (\Re(\alpha_1(z)), \Im(\alpha_1(z)), \Re(\alpha_2(z)), \Im(\alpha_2(z)))$$

where $z = x + iy$, to be able to use the proposed models in the graphic representation of them.

Below are the sentences to obtain the graphs of a single complex surface and two complex surfaces together in $\mathbb{R}^4$ using the `ComplexPlot` command.

```
ComplexPlot[z^2, {z, -1 - I, 1 + I}, AxesLength -> {-2, 2}]
```

See Fig. 16 *(left).*

```
ComplexPlot[
 {z^2, z^3}, {z, -1 - I, 1 + I}, AxesLength -> {-2, 2}]
```

See Fig. 16 *(right).*

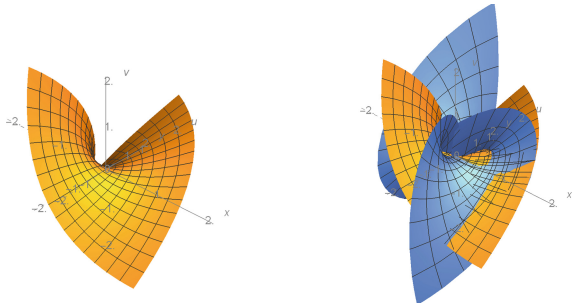

Fig. 16. Graphical visualization of complex surfaces in $\mathbb{R}^4$ with `ComplexPlot`.

The following sentences make it possible to obtain the graphs of a single complex surface and of two complex surfaces together in $\mathbb{R}^4$ using the `ParametricComplexPlot` command.

```
ComplexParametricPlot[
 {Cos[z], Sin[z]}, {z, -Pi - Pi I, Pi + Pi I},
 Mesh -> True, RegionFunction -> (#1^2 + #2^2 < 2 Pi &),
 AxesLength -> {-3.2, 3.2}, PlotPoints -> 40,
 MaxRecursion -> 8, ColorFunction -> "DarkRainbow"]
```

See Fig. 17 *(left).*

```
ComplexParametricPlot[
{{Cos[z], Sin[z]}, {z, z^2}}, {z, -Pi - Pi I, Pi + Pi I},
 AxesLength -> {-2.6, 2.6}, PlotPoints -> 40, MaxRecursion -> 8,
 RegionFunction -> (#1^2 + #2^2 + #3^2 + #4^2 + #5^2 < 7 Pi &)]
```

See Fig. 17 *(right).*

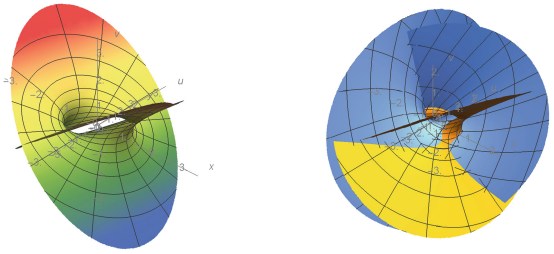

Fig. 17. Graphical visualization of complex surfaces in $\mathbb{R}^4$ with `ComplexParametricPlot`.

Finally, we can see the graphs of a single surface and two surfaces together in $\mathbb{R}^4$ using the `ComplexPolarPlot` command.

```
ComplexPolarPlot[Sin[2 T], {T, -Pi-Pi I/2, Pi+Pi I/2}, Mesh->True,
 RegionFunction -> (#1^2 + #2^2 + #3^2 + #4^2 + #5^2 < 8 &),
 AxesLength -> {-3, 3}, PlotPoints -> 40, MaxRecursion -> 8,
 ColorFunction -> "Rainbow"]
```

See Fig. 18 *(left).*

```
ComplexPolarPlot[{Sin[2 T], 1+Cos[T]}, {T, -Pi-Pi I/2, Pi+Pi I/2},
 Mesh->True, RegionFunction -> (#1^2+#2^2+#3^2+#4^2+#5^2 < 8 &),
 AxesLength -> {-3, 3}, PlotPoints -> 40, MaxRecursion -> 8]
```

See Fig. 18 *(right).*

3D models have been excellent algorithms to implement several commands in *Mathematica* to visualize geometric objects immersed in $\mathbb{R}^4$. These commands will help explore several aspects of various geometric objects immersed in $\mathbb{R}^4$. For example, given a function $f : \mathbb{R} \to \mathbb{R}$, geometrically, its derivative gives us the value of the slope of the tangent line to the graph of f at a given point in its domain. In such a way that it is easy to find the equation of the tangent line to the graph of any differentiable function in an arbitrary point of its domain.

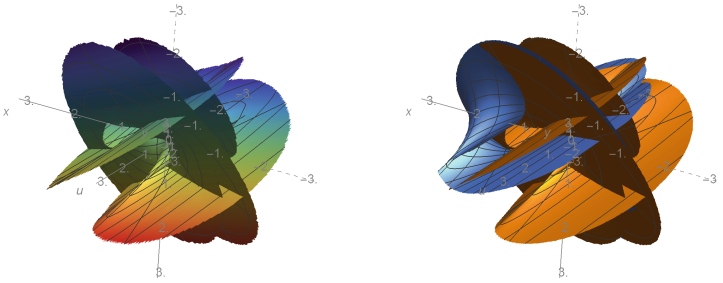

Fig. 18. Graphical visualization of complex surfaces in $\mathbb{R}^4$ with `ComplexPolarPlot`.

It is amazing to verify, geometrically, that this result is also true for complex functions $F : \mathbb{C} \to \mathbb{C}$.

Next we define the function $F(z) = z^2$ and the point $(z_0, w_0) = (0.75 + 0.5i, F(0.75 + 0.5i))$, then we plot the function, its tangent plane at the given point and the point of tangency. For this, the standard `Show` command is used together with the new `ComplexPlot` and `Graphics4D` commands.

```
F[z_] := z^2;
z0 = .75 + .35 I;
w0 = F[z0];
Show[ ComplexPlot[{F[z], w0 + F'[z0] (z - z0)}, {z, 0, 1 + I}],
 Graphics4D[{AbsolutePointSize[10], Point[{Re[z0], Im[z0],
  Re[w0], Im[w0]}]}] ]
```

See Fig. 19.

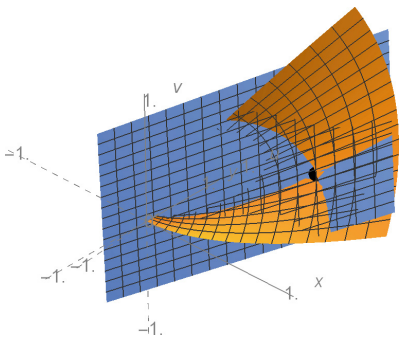

Fig. 19. Geometric interpretation of the derivative of a complex function, at a point in its domain, with `ComplexPlot` and `Graphics4D`.

The authors are aware that there is still a need to investigate hidden faces, lighting, shading, etc. in $\mathbb{R}^4$ to have a real perception of geometric objects [1,4]. An idea of this is appreciated in Fig. 20; in which two graphs are compared, one

obtained in rudimentary form (using a 2D model and *Mathematica* commands that are not for $\mathbb{R}^3$) and the other obtained with the `ContourPlot3D` command. However, we get satisfactory sketches for now.

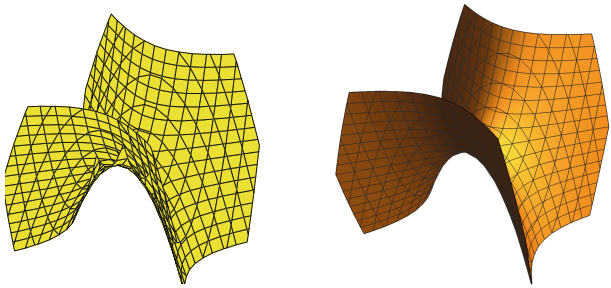

Fig. 20. The same surface generated rudimentarily (left) and generated with `ContourPlot3D` (right).

4 Conclusions

In this paper, a new Mathematica package for visualizing points, lines, arrows, curves, surfaces, solids and complex surfaces inmersed in $\mathbb{R}^4$ is introduced. The incoporated commands in this package will help explore several aspects of various geometric objects immersed in $\mathbb{R}^4$. The algorithms for these commands are based on 3D models built as an extension of the 2D models used to visualize objects immersed in $\mathbb{R}^3$. The performance of the package is discussed by means of some illustrative and interesting examples.

All the commands have been implemented in Mathematica version 11.0 and are consistent with Mathematica's notation and results. The powerful Mathematica functional programming [7] features have been extensively used to make the program shorter and more efficient. From our experience, Mathematica provides an excellent framework for this kind of developments.

Acknowledgements. The authors would like to thank to the authorities of the Universidad Nacional de Piura for the acquisition of the Mathematica 11.0 license and the reviewers for their valuable comments and suggestions.

References

1. Agoston, M.K.: Computer Graphics and Geometric Modelling. Springer, London (2005). https://doi.org/10.1007/b138805
2. Ohori, A., et al.: Visualising higher-dimensional space-time and space-scale objects as projections to $\mathbb{R}^3$. PeerJ Comput. Sci. **3**, e123 (2017). https://doi.org/10.7717/peerj-cs.123

3. Banchoff, T.F.: Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions. Scientific American Library Series. W. H. Freeman & Company, New York (1990)
4. Buss, S.R.: 3D Computer Graphics: A Mathematical Introduction with OpenGL. Cambridge University Press, Cambridge (2003)
5. Brandel, S.: 4D objects for animation: immersion on virtual reality. In: 9th Virtual Reality International Conference, Laval, France, pp. 1–5 (2007)
6. Hoffmann, C.M., Zhou, J.: Visualization of surfaces in four-dimensional space. Computer Science Technical Reports, Paper 814 (1990)
7. Maeder, R.: Programming in Mathematica, 2nd edn. Addison-Wesley, Redwood City (1991)
8. Noll, M.: A computer technique for displaying n-dimensional hyperobjects. Commun. ACM **10**(8), 469–473 (1967). https://doi.org/10.1145/363534.363544
9. Pham, F.: Some notions of local differential topology. In: Ramakrishnan, A. (ed.) Symposia on Theoretical Physics and Mathematics, pp. 65–100. Springer, Boston (1968). https://doi.org/10.1007/978-1-4684-7727-6_3
10. Sakai, Y., Hashimoto, S.: Interactive four-dimensional space visualization using five-dimensional homogeneous processing for intuitive understanding. J. Inst. Image Inf. Telev. Eng. **60**(10), 1630–1647 (2006). https://doi.org/10.3169/itej.60.1630
11. Séquin, C.H.: 3D visualization models of the regular polytopes in four and higher dimensions. In: Bridges 2002, Towson, Maryland, USA, pp. 37–48 (2002)
12. Volkert, K.: On models for visualizing four-dimensional figures. Math. Intelligencer **39**, 27–35 (2017). https://doi.org/10.1007/s00283-016-9699-1
13. Wang, W.M., et al.: Interactive exploration of 4D geometry with volumetric halos. The Eurographics Association (2013)
14. Wolfram, S.: The Mathematica Book, 4th edn. Wolfram Media, Cambridge University Press, Champaign, Cambridge (1999)
15. Zhou, J.: Visualization of four dimensional space and its applications. D. Phil. thesis, Department of Computer Science Technical Reports, Paper 922 (1991)