A Mathematica Package for Plotting Implicitly Defined Hypersurfaces in $\mathbb{R}^4$

Luis A. Anto, Amelia M. Fiestas, Edgar J. Ojeda, Ricardo Velezmoro, and Robert Ipanaqué(✉)

Universidad Nacional de Piura, Urb. Miraflores s/n, Castilla, Piura, Peru
la86926@gmail.com, ame_18_27@hotmail.com,
{eojedam,rvelezmorol,ripanaquec}@unp.edu.pe

Abstract. Plotting implicitly defined geometric objects is a very important topic on computer graphics, computer aided design and geometry processing. In fact, the most important computer algebra systems include sophisticated tools for plotting implicitly defined curves and surfaces. This paper describes a new Mathematica package, `4DPlots`, for plotting implicitly defined hypersurfaces (solids) in $\mathbb{R}^4$ using a generalization of the bisection method that is applied to continuous functions of four variables by recursive bisection of segments contained in their domain. The output obtained is consistent with Mathematica's notation and results. The performance of the package is discussed by means of several illustrative and interesting examples.

Keywords: Implicitly defined hypersurfaces in $\mathbb{R}^4$ · Plotting implicitly defined hypersurfaces in $\mathbb{R}^4$ · Approximation of zeros of functions of four variables · The bisection method over segments in $\mathbb{R}^4$

1 Introduction

Plotting implicitly defined geometric mathematical objects is a very important topic on computer graphics, computer aided design and geometry processing [2,4–7,15,24,26,27]. There are several powerful algorithms to obtain points that satisfy the equations that express the curves and surfaces given implicitly [8,9,14,24,25,28]. In fact, the most important computer algebra systems include sophisticated tools for plotting implicitly defined curves and surfaces [13,17,23,30,31]. Currently, there are even online applications to plot these objects [11,16].

The approximation of zeros of continuous functions of various variables and real value has application to reality, specifically in nonlinear minimization problems [21]. Several works have been carried out to solve the problem of the approximation of zeros of continuous functions of various variables and real value, based on the bisection method [18,20]. In [3,22] it has been used to approximate zeros of functions of various variables and real value, thus obtaining the graph of curves and surfaces defined implicitly. Even, there is a previous result to this paper in which the wireframe plots of implicitly defined hypersurfaces is obtained [1].

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 117–129, 2020.
https://doi.org/10.1007/978-3-030-58802-1_9

This paper presents a new Mathematica package, `4DPlots`, which incorporates a command for visualizing hypersurfaces (solids) inmersed in $\mathbb{R}^4$. The encoding of the command is based on the multivariable bisection method. The command provide the user with a highly intuitive, mathematical-looking output consistent with Mathematica's notation and syntax [19].

The structure of this paper is as follows: Sect. 2 introduce the mathematical definition of the multivariate bisection method and its algorithm. For the sake of illustration, some surface plots are also briefly described in this section. Then, Sect. 3 introduces the new Mathematica package, `4DPlots`, and describes the command implemented within. The performance of the package is also discussed in this section by using some illustrative examples to plot implicit hypersurfaces. Finally, Sect. 5 closes with the main conclusions of this paper.

2 Mathematical Preliminaries

2.1 Bisection Method

Theorem 1 (Conservation of the sign). *Let $f : \mathbb{R} \to \mathbb{R}$ be continuous on c and suppose that $f(c) \neq 0$. Then there is an interval $(c-\delta, c+\delta)$ in which f has the same sign as $f(c)$.*

Theorem 2 (Bolzano). *Let $f : \mathbb{R} \to \mathbb{R}$ be continuous at each point of the closed interval $[a,b]$ and suppose that $f(a)$ and $f(b)$ have opposite signs. Then there is at least one c in the open interval (a,b) such that $f(c) = 0$.*

Theorem 3 (Intermediate value for continuous functions). *Let $f : \mathbb{R} \to \mathbb{R}$ be continuous at each point of an interval $[a,b]$. If $x_1 < x_2$ are any two points of $[a,b]$ such that $f(x_1) \neq f(x_2)$, the function f takes all the values between $f(x_1)$ and $f(x_2)$ at least once in the interval (x_1, x_2).*

Definition 1 (Zeros of a function from $\mathbb{R}$ to $\mathbb{R}$). *Let $f : \mathbb{R} \to \mathbb{R}$ be continuous on $[a,b]$, the zeros of f are the elements of the set*

$$\theta_f = \{\, x \in [a,b] \mid f(x) = 0 \,\}.$$

Definition 2 (Bisection method). *Let $f : \mathbb{R} \to \mathbb{R}$ be continuous at each point of an interval $[a,b]$, with $f(a) \cdot f(b) < 0$. The sequence $\{x_n\}_{n\in\mathbb{N}}$, such that*

$$x_n(a,b) = \begin{cases} \frac{a+b}{2} & n = 1 \vee f\left(x_1(a,b)\right) = 0\,, \\ \begin{cases} x_{n-1}\left(x_1(a,b), b\right) & f(a)\cdot f\left(x_1(a,b)\right) > 0\,, \\ x_{n-1}\left(a, x_1(a,b)\right) & \textit{otherwise}\,. \end{cases} & n > 1\,. \end{cases}$$

converges to p when $n \to \infty$, with $f(p) = 0$, as fast as $\left\{\left(\frac{1}{2}\right)^n\right\}_{n\in\mathbb{N}}$ converges to zero.

The previous definition can be coded without any problem but it is not practical when operating with tolerance, for this reason it will be necessary to resort to an algorithm associated with this method.

Program 1 (Bisection method from definition). *Mathematica code for bisection method, based on previous definition:*

```
x[n_][a_,b_]:=
 If[ n==1 || f[x[1][a,b]]==0, (a+b)/2.,
   If[ f[a] f[x[1][a,b]]>0,
         x[n-1][x[1][a,b],b], x[n-1][a,x[1][a,b]]
   ]
 ]
```

For example, consider finding the zero of $f(x) = e^{-x}(3.2\sin(x) - 0.5\cos(x))$ on the interval [3, 4]. An approximation of function's zero using the command implemented in the previous definition ($n = 20$) is:

```
f[x_] := Exp[-x] (3.2 Sin[x] - 0.5 Cos[x])
x[20][3,4]
```

3.29659

It is briefly verified that:

```
f[%]
```

$-5.464291272381732 \times 10^{-8}$

The algorithm of the bisection method used in this paper is based on the algorithm proposed by Burden et al. [10]. The modification is that the maximum number of iterations is not considered.

Algorithm 1 (Bisection method). *To find a solution to $f(x) = 0$ given the continuous function f on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs:*
INPUT *function f; endpoints a, b; tolerance TOL.*
OUTPUT *approximate solution p.*
Step 1 *Set $i = 1$;*
$FA = f(a)$.
Step 2 *While $0.5 \cdot (b - a) \geqq TOL$ do Steps 3–5.*
Step 3 *Set $p = 0.5 \cdot (a + b)$;*
$FP = f(p)$.
Step 4 *If $FP = 0$ or $|FP| < TOL$ then*
OUTPUT (p);
STOP.
Step 5 *If $FA \cdot FP > 0$ then set $a = p$;*
$FA = FP$
else set $b = p$.
Step 6 OUTPUT (p).

Program 2 (Bisection method from algorithm). *Mathematica code for bisection method, based on previous algorithm:*

```
Bisection[fun_,var_,a_,b_,TOL_]:=
 Module[{f=Function[var,fun],A=a,B=b,FA,p,FP},
  FA=f[a];
  While[0.5 (B-A)>=TOL,p=0.5 (A+B);
   FP=f[p];
   If[FP==0||Abs[FP]<TOL,Return[p];Break];
   If[FA FP>0,A=p;FA=FP,B=p];];
  p]
```

If in the previous example it is required that $TOL = 0.001$, then we obtain the next value of p:

```
Bisection[f[x],x,3,4,0.001]
```

3.29688

It is easily verified that the required precision is met, that is, $|f(p)| < 0.001$:

```
Abs[f[%]]
```

0.0000342246

2.2 Multidimensional Bisection Method

According to Gomes [14] the bisection method can be applied to functions of various variables and real value. Based on this, the definition and algorithm for the multivariate bisection method are stated as follows.

Definition 3 (Closed segment). *Let $\bar{a}, \bar{b}$ be in $\mathbb{R}^d$. The line segment with extremes $\bar{a}, \bar{b}$ is the set*

$$\left[\bar{a}, \bar{b}\right] = \left\{ (1-t)\,\bar{a} + t\,\bar{b} \mid 0 \leqq t \leqq 1 \right\}.$$

Definition 4 (Multidimensional bisection method). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuous at each point of an closed set D. Let $\left[\bar{a}, \bar{b}\right] \subset D$ be with $f(\bar{a}) \cdot f\left(\bar{b}\right) < 0$. The sequence $\{x_n\}_{n\in\mathbb{N}}$, such that*

$$x_n\left(\bar{a}, \bar{b}\right) = \begin{cases} \frac{\bar{a}+\bar{b}}{2} & n = 1 \vee f\left(x_1\left(\bar{a}, \bar{b}\right)\right) = 0\,, \\ \begin{cases} x_{n-1}\left(x_1\left(\bar{a}, \bar{b}\right), \bar{b}\right) & f(\bar{a}) \cdot f\left(x_1\left(\bar{a}, \bar{b}\right)\right) > 0\,, \\ x_{n-1}\left(\bar{a}, x_1\left(\bar{a}, \bar{b}\right)\right) & \textit{otherwise}\,. \end{cases} & n > 1\,. \end{cases}$$

converges to $\bar{p}$ when $n \to \infty$, with $f(\bar{p}) = 0$, as fast as $\left\{\left(\frac{1}{2}\right)^n\right\}_{n\in\mathbb{N}}$ converges to zero.

Note that in the previous definition each approximation of $\bar{p}$ is always on the segment $\left[\bar{a}, \bar{b}\right]$.

Program 3 (Multidimensional bisection method from definition). *Mathematica code for multidimensional bisection method, based on previous definition:*

```
x[n_][a_?VectorQ,b_?VectorQ]:=
 If[ n==1 || f@@x[1][a,b]==0, (a+b)/2.,
   If[ f@@a f@@x[1][a,b]>0,
         x[n-1][x[1][a,b],b], x[n-1][a,x[1][a,b]]
   ]
 ]
```

For example, consider finding the zero of $f(x, y) = x^2 + y^2 - 1$ on the interval $[(0.5, 0.25), (1.25, 1.5)]$. An approximation of function's zero using the command implemented in the previous definition ($n = 20$) is:

```
f[x_, y_] := x^2 + y^2 - 1
x[20][{0.25,0.5},{1.25,1.5}]
```

$\{0.570971, 0.820971\}$

Here it is also briefly verified that:

```
f@@%
```

$2.6268699002685025 \times 10^{-6}$

For the same reasons explained above, we will enunciate an algorithm associated with this method.

Algorithm 2 (Multivariate bisection method). *To find a solution to* $f(\bar{x}) = 0$ *given the continuous function* f *on the closed segment* $[\bar{a}, \bar{b}]$*, where* $f(\bar{a})$ *and* $f(\bar{b})$ *have opposite signs:*
INPUT *function* f*; endpoints* $\bar{a}, \bar{b}$*; tolerance* TOL*.*
OUTPUT *approximate solution* $\bar{p}$*.*
Step 1 *Set* $i = 1$*;*
$FA = f(\bar{a})$*.*
Step 2 *While* $0.5 \cdot ||b - a|| \geqq TOL$ *do Steps 3–5.*
Step 3 *Set* $\bar{p} = 0.5 \cdot (\bar{a} + \bar{b})$*;*
$FP = f(\bar{p})$*.*
Step 4 *If* $FP = 0$ *or* $|FP| < TOL$ *then*
OUTPUT $(\bar{p})$*;*
STOP*.*
Step 5 *If* $FA \cdot FP > 0$ *then set* $\bar{a} = \bar{p}$*;*
$FA = FP$
else set $\bar{b} = \bar{p}$*.*
Step 6 OUTPUT $(\bar{p})$*.*

Program 4 (Bisection method from algorithm). *Mathematica code for multivariate bisection method, based on previous algorithm:*

```
Bisection[fun_,var?VectorQ_?VectorQ,a_?VectorQ,b_,TOL_]:=
 Module[{f=Function[var,fun],A=a,B=b,FA,p,FP},
  FA=f@@a;
  While[0.5 Norm[B-A]>=TOL,p=0.5 (A+B);
   FP=f@@p;
   If[FP==0||Abs[FP]<TOL,Return[p];Break];
   If[FA FP>0,A=p;FA=FP,B=p];];
  p]
```

If in the previous example it is required that $TOL = 0.001$, then we obtain the next value of $\bar{p}$:

```
Bisection[f[x,y],{x,y},{0.25,0.5},{1.25,1.5},0.001]
```

$\{0.571289, 0.821289\}$

It is easily verified that the required precision is met, that is, $|f(\bar{p})| < 0.001$:

```
Abs[f@@%]
```

0.000886917

Figure 1 shows the geometric interpretation of the bisection method in the univariate (left) and bivariate (right) cases. As can be seen, the zeros of the bivariate function form a curve on the xy plane and the approximation of $\bar{p}$ is made on the closed segment $[\bar{a}, \bar{b}]$.

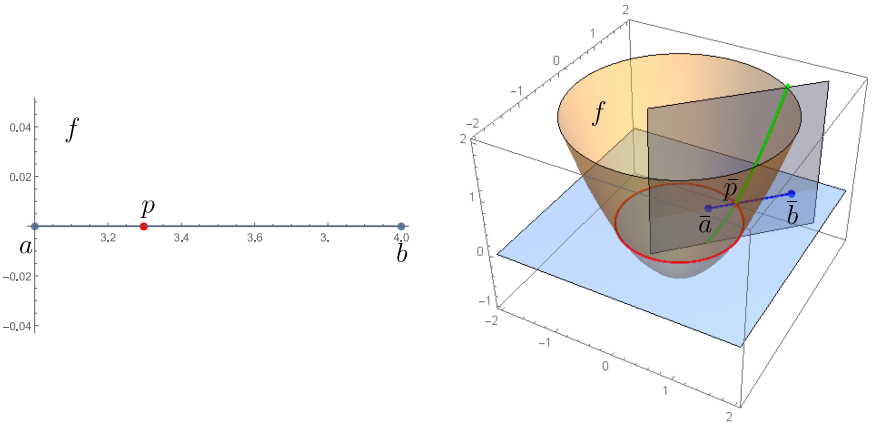

Fig. 1. Geometric interpretation of the bisection method.

If a mesh is constructed over the domain of f (Fig. 2, left), each subinterval that makes up the mesh (both horizontal and vertical) constitutes a closed segment and it is possible to apply the multivariate bisection method on such segments (Fig. 2, right). For example, the roots of a function $f : \mathbb{R}^2 \to \mathbb{R}$ make

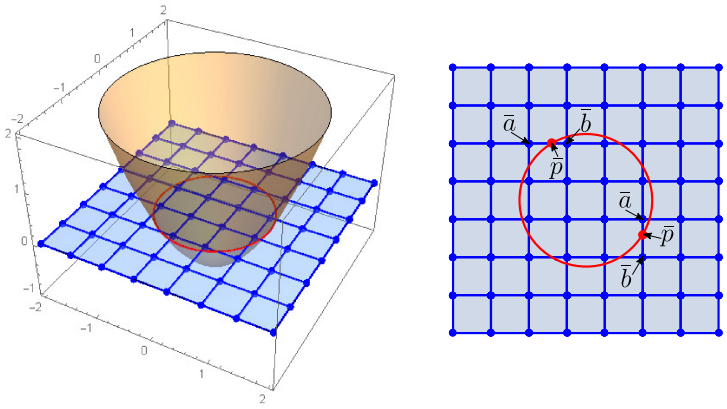

Fig. 2. Construction of a mesh on the domain of f.

up a curve (Fig. 3, left), of a function $f : \mathbb{R}^3 \to \mathbb{R}$ make up a surface (Fig. 3, right), of a function $f : \mathbb{R}^4 \to \mathbb{R}$ make up a solid, and so on.

Extending the ideas presented with the geometric interpretation of the bivariate bisection method, it is possible to have a clear idea of the geometric interpretation of the multivariate bisection method (Fig. 3).

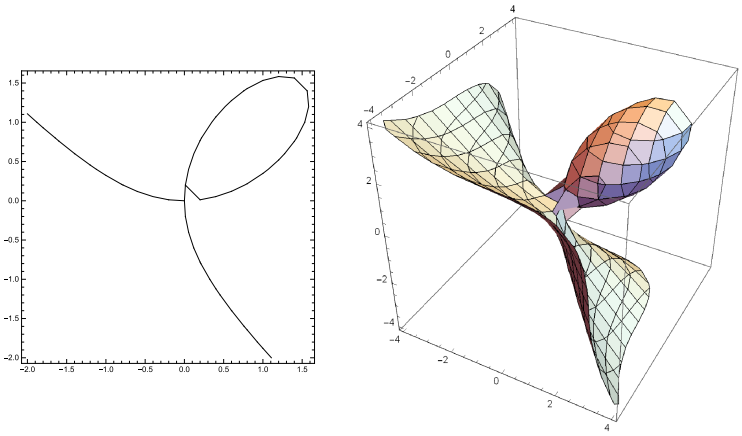

Fig. 3. Zeros of the function $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x, y, z) = x^3 + y^3 - 3xy$ (left) and $f : \mathbb{R}^3 \to \mathbb{R}$, $f(x, y, z) = x^3 + y^3 + z^3 - 3xy - 3xz - 3yz$ (right).

To trace the hypersurfaces, defined implicitly, from the following section a trimetric-trimetric model according to [29] is used. This is,

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

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

3 The Package `4DPlots`: Some Illustrative Examples

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

```
<<4DPlots.m
```

The command incorporated in this package is:

`ImplicitPlot4D`

With the `ImplicitPlot4D` command it is possible to visualize the projections of the graphs of the hyperplanes $H_1 : x = 0$ and $H_2 : y = 0$; and with the built-in `Show` command, combine both graphics.

```
H1=ImplicitPlot4D[x==0,{x,-1,1},{y,-1,1},{z,-1,1},{w,-1,1}]
```

See Fig. 4 (left)

```
H2=ImplicitPlot4D[y==0,{x,-1,1},{y,-1,1},{z,-1,1},{w,-1,1}]
```

See Fig. 4 (center)

```
Show[H1,H2]
```

See Fig. 4 (right)

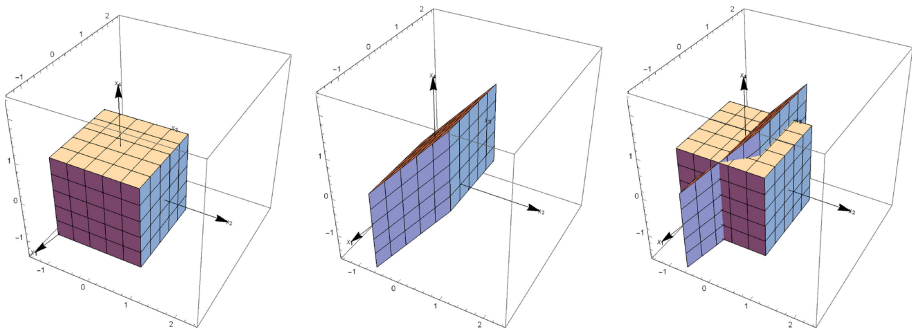

Fig. 4. Two hyperplanes.

The following sentences show the projection of the unit hypersphere $H_3 : x^2 + y^2 + z^2 + w^2 = 1$ and its normal hyperplane $H_4 : w = 1$, at the point $(0, 0, 0, 1)$.

```
H3=ImplicitPlot4D[x^2+y^2+z^2+w^2==1,{x,-1,1},{y,-1,1},{z,-1,1},
    {w,-1,1}]
```

See Fig. 5 (top-left)

```
H4=ImplicitPlot4D[w==1,{x,-1,1},{y,-1,1},{z,-1,1},{w,-1,1}]
```

See Fig. 5 (top-right)

```
Show[H3,H4]
```

See Fig. 5 (bottom-left)

```
Show[H3,H4,ViewPoint->{2.889, -0.756, 1.59}]
```

See Fig. 5 (bottom-right)

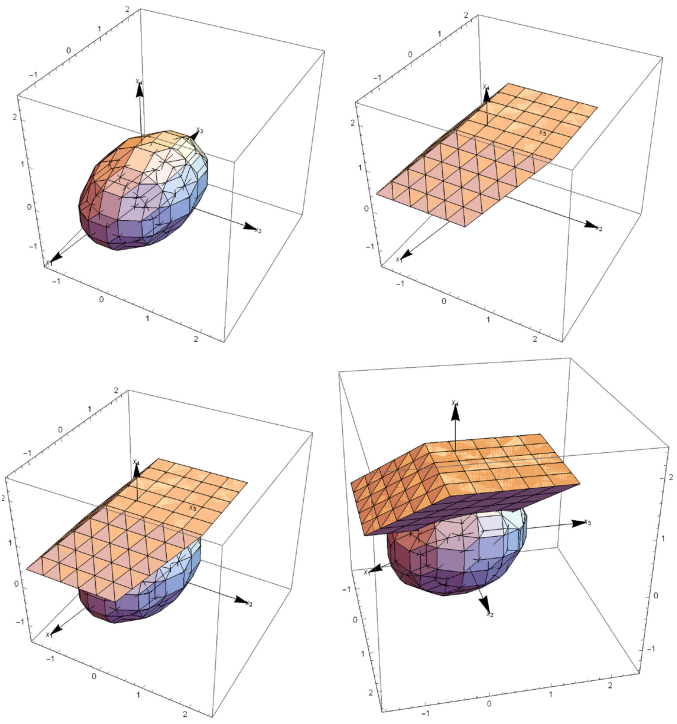

Fig. 5. Unitary hypersphere and its normal plane at $(0, 0, 0, 1)$.

The following sentences allow to obtain the projections of three hyperquadric and one equilateral hyperhyperbola.

```
ImplicitPlot4D[x^2+y^2+z^2==1,{x,-1,1},{y,-1,1},{z,-1,1},{w,-2,2}]
```

See Fig. 6 (top-left)

```
ImplicitPlot4D[x^2+y^2+z^2-w^2==1,{x,-2,2},{y,-2,2},{z,-2,2},
  {w,-2,2}]
```

See Fig. 6 (top-right)

```
ImplicitPlot4D[x^2+y^2-z^2-w^2==1,{x,-2,2},{y,-2,2},{z,-2,2},
  {w,-2,2}]
```

See Fig. 6 (bottom-left)

```
ImplicitPlot4D[x y z w==1,{x,-2,2},{y,-2,3},{z,-2,2},{w,-2,2}]
```

See Fig. 6 (bottom-right)

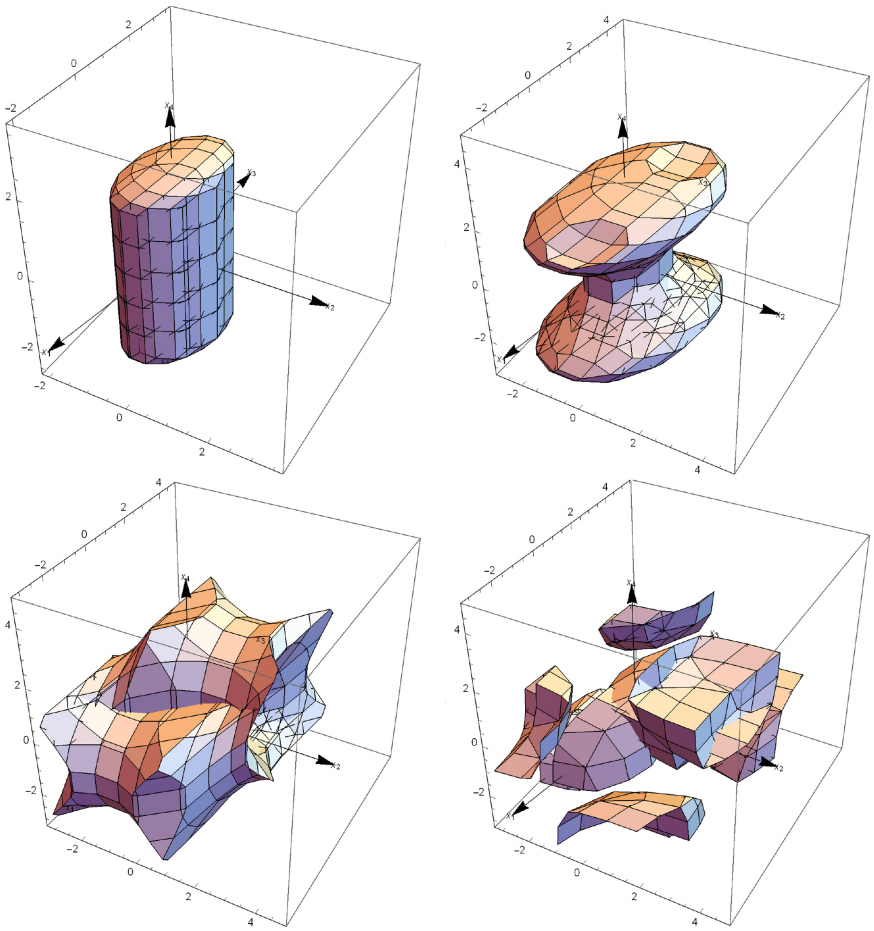

Fig. 6. Some hypersurfaces.

4 Another Possible Command for the Package

The bisection method could extend naturally, from Definition 4, as indicated in the following definition [12].

Definition 5. *Let $F:\mathbb{R}^d\to\mathbb{R}^h$, $F=(f_1,\ldots,f_h)$, be continuous at each point of an closed set D. Let $[\bar{a},\bar{b}]\subset D$ be with $\bigwedge_{i=1}^{h} f_i(\bar{a})\cdot f_i(\bar{b})<0$. The sequence $\{\bar{x}_n\}_{n\in\mathbb{N}}=\{x_{1_n},\ldots,x_{h_n}\}_{n\in\mathbb{N}}$, such that*

$$\bar{x}_n(\bar{a},\bar{b})=\begin{cases}\frac{\bar{a}+\bar{b}}{2} & n=1\vee F\left(\bar{x}_1(\bar{a},\bar{b})\right)=\bar{0}\,,\\ \begin{cases}\bar{x}_{n-1}\left(\bar{x}_1(\bar{a},\bar{b}),\bar{b}\right) & \bigwedge_{i=1}^{h} f_i(\bar{a})\cdot f_i\left(\bar{x}_1(\bar{a},\bar{b})\right)>0\,,\\ \bar{x}_{n-1}\left(\bar{a},\bar{x}_1(\bar{a},\bar{b})\right) & \textit{otherwise}\,.\end{cases} & n>1\,.\end{cases}$$

converges to $\bar{p}$ when $n\to\infty$, with $F(\bar{p})=\bar{0}$, as fast as $\left\{h\left(\frac{1}{2}\right)^n\right\}_{n\in\mathbb{N}}$ converges to zero.

This definition can be used to approximate zeros of functions such as $F=(x^2+y^2+z^2-1,x+y+z)$ (Fig. 7, left). Even more bold and based on the isomorphism between $\mathbb{C}$ and $\mathbb{R}^2$, we could approximate the zeros of the function $F=z^2+w^2-1$, where $z=x+iy$ and $w=u+iv$. In fact, some points were obtained which are shown in Fig. 7 (center). Then in Fig. 7 (right), the points are shown along with the graph of the functions $w=\sqrt{1-z^2}$ and $w=-\sqrt{1-z^2}$ [12,29].

We emphasize here that by improving the technique or using a more effective method, a new `ImplicitComplexPlot` command can be implemented.

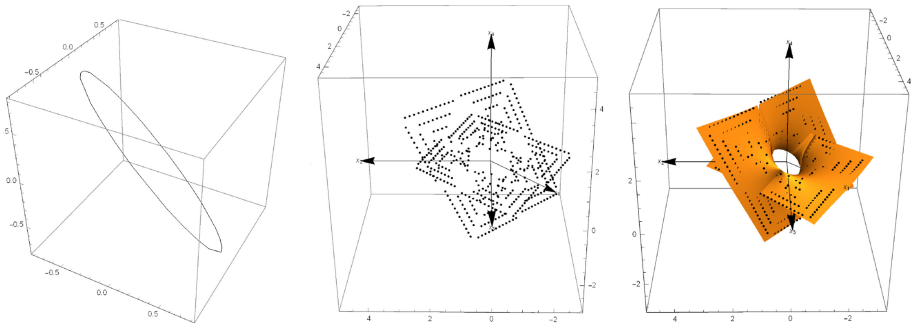

Fig. 7. Zeros of some two-variable functions and a. three real values (left) b. four real values (center and right).

5 Conclusions

In this paper, a new Mathematica package for plotting implicitly defined hypersurfaces (solids) inmersed in $\mathbb{R}^4$ is introduced. The incorporated command in

this package will help explore several aspects of various solids immersed in $\mathbb{R}^4$. The algorithm for these commands is based on multivariate bisection method. The performance of the package is discussed by means of some illustrative and interesting examples. Additionally, a compact definition is presented, as well as its respective Mathematica coding, of the bisection method.

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 [19] 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. Anto, L.A.: Aproximación de los ceros de funciones continuas de varias variables reales y valor real mediante el método de bisección con el Mathematica. Universidad Nacional de Piura (2019). http://repositorio.unp.edu.pe/handle/UNP/18923
2. Agoston, M.K.: Computer Graphics and Geometric Modeling, Implementation and Algorithms. Springer, London (2005). https://doi.org/10.1007/b138805
3. Bachrathy, D., Stépán, G.: Bisection method in higher dimensions and the efficiency number. Periodica polytechnica **56**, 81–86 (2012)
4. Barnhill, R.E.: Geometry processing: curvature analysis and surface-surface intersection. In: Mathematical Methods in Computer Aided Geometric Design, pp. 51–60 (1989)
5. Barnhill, R.E.: Geometry Processing for Design and Manufacturing. Society for Industrial and Applied Mathematics (1992)
6. Bærentzen, J., et al.: Guide to Computational Geometry Processing, Foundations, Algorithms, and Methods. Springer, London (2012). https://doi.org/10.1007/978-1-4471-4075-7
7. Bi, Z., Wang, X.: Computer Aided Design and Manufacturing. Wiley, Hoboken (2020)
8. Botsch, M., et al.: Polygon Mesh Processing. A K Peters, Ltd., Natick (2010)
9. Bloomenthal, J., et al.: Introduction to Implicit Surfaces. Elsevier, Amsterdam (1997)
10. Burden, R., et al.: Numerical Analysis, 10th edn. Cengage Learning, Boston (2015)
11. CalcPlot3D Homepage. https://www.monroecc.edu/faculty/paulseeburger/calcnsf/CalcPlot3D/. Accessed 18 Apr 2020
12. Fiestas, A.M.: Aproximación de ceros de funciones continuas de dos variables complejas y valor complejo mediante el método de bisección. Universidad Nacional de Piura (2019). http://repositorio.unp.edu.pe/handle/UNP/1963
13. Garvan, F.: The Maple Book. Chapman & Hall/CRC (2002)
14. Gomes, A.J.P.: Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms. Springer, London (2009)
15. Hartmann, E.: Geometry and Algorithms for Computer Aided Design. Darmstadt University of Technology (2003)

16. Implicit 3D Plot Homepage. http://matkcy.github.io/MA1104-implicitplot.html. Accessed 18 Apr 2020
17. Ipanaqué, R.: Breve manual de Maxima. Eumed.net (2010)
18. López, M.: The multivariate bisection algorithm. ARXIV, 1–19 (2018)
19. Maeder, R.: Programming in Mathematica, 2nd edn. Addison-Wesley, Redwood City (1991)
20. Martin, C., Rayskin, V.: An improved bisection method in two dimensions. Preprint submitted to Elsevier, pp. 1–21 (2016)
21. Morozova, E.: A multidimensional bisection method. In: Proceedings of the Fourteenth Symposium on Computing: The Australasian Theory, pp. 57–62. Australian Computer Society (1989)
22. Multi-Dimensional Bisection Method Homepage. https://www.mm.bme.hu/~bachrathy/research_EN.html#MDBM. Accessed 18 Apr 2020
23. Parekh, R.: Fundamentals of Graphics Using MatLab. CRC Press (2019)
24. Patrikalakis, N., Maekawa, T.: Shape Interrogation for Computer Aided Design and Manufacturing. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-04074-0
25. Plantinga, S.: Certified Algorithms for Implicit Surfaces (2007)
26. Sarcar, M.M.M., et al.: Computer Aided Design and Manufacturing. PHI Learning Pvt., Ltd. (2008)
27. Shirley, P.: Fundamentals of Computer Graphics, 3rd edn. Taylor & Francis Group (2009)
28. Sultanow, E.: Implizite Flächen. Mathematical Methods in Computer Aided Geometric Design, pp. 1–11
29. Velezmoro, R., Ipanaqué, R., Mechato, J.A.: A mathematica package for visualizing objects inmersed in $\mathbb{R}^4$. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11624, pp. 479–493. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24311-1_35
30. Wolfram, S.: The Mathematica Book, 5th edn. Wolfram Media Inc. (2003)
31. Zimmermann, P., et al.: Computational Mathematics with SageMath, 1st edn. SIAM- Society for Industrial and Applied Mathematics (2018)