Euclidean Distance Degrees in Macaulay2

William Huang[1][0000-0001-6287-772X] and Jose Israel Rodriguez[2][0000-0003-3140-9944]

[1] University of British Columbia, Vancouver, Canada
[2] University of Wisconsin — Madison, Madison, Wisconsin
jose@math.wisc.edu

Abstract. We introduce `EuclideanDistanceDegree`, a *Macaulay2* package that implements symbolic and numerical methods for computing Euclidean Distance (ED) degrees. The package includes symbolic methods based on minors and conormal varieties as well as numerical methods for unit and generic ED degrees using tools from numerical algebraic geometry. We illustrate the package functionality with a range of examples in the paper and in the accompanying GitHub repository.

Keywords: ED degree · algebraic optimization · Macaulay2.

1 Introduction

One of the most widely studied invariants in algebraic optimization is the Euclidean Distance (ED) degree. It measures the complexity of nearest-point problems for real algebraic varieties: for a generic data point, it is the number of critical points of the squared Euclidean distance function on the variety. As many models across science and engineering are expressed in terms of polynomial systems, tools for computing ED degrees of arbitrary varieties allow researchers to better understand the algebraic complexity of their models.

In the `EuclideanDistanceDegree` package for the computer algebra system *Macaulay2* [9], we implement methods for computing these degrees. These include symbolic methods to compute exact ED degrees for affine models as well as numerical methods using homotopy continuation. This package relies on the Bertini software package [2] and the corresponding *Macaulay2* wrapper [1]. We also utilize the `MonodromySolver` package [6] to compute ED degrees of parameterized varieties.

2 Computing ED Degrees

The Euclidean Distance degree arises from the following problem: given a variety $X \subseteq \mathbb{R}^n$ and data point $\mathbf{u} \in \mathbb{R}^n \setminus X$, find the point $\mathbf{x}^* \in X$ which minimizes the squared distance

$$\|\mathbf{x} - \mathbf{u}\|^2 = \sum_{i=1}^{n} (x_i - u_i)^2 \tag{1}$$

We also consider the weighted squared distance, which depends on a weight vector $\mathbf{w} \in \mathbb{R}^n_{>0}$ in addition to the data point:

$$\|\mathbf{x} - \mathbf{u}\|^2_{\mathbf{w}} = \sum_{i=1}^{n} w_i (x_i - u_i)^2. \tag{2}$$

By viewing X as a complex variety in $\mathbb{C}^n$, the number of nonsingular complex critical points of $\|\mathbf{x} - \mathbf{u}\|^2_{\mathbf{w}}$ is the *weighted Euclidean distance degree* for the variety X with respect to the weights $\mathbf{w}$. This number is independent of the choice of data point $\mathbf{u}$, provided it is sufficiently general [4]. We refer to the choices of arbitrary weights, unit weights, and generic weights as the weighted, unit, and generic ED degrees, respectively.

2.1 Minors Method

Given an irreducible real algebraic variety X, the ED degree can be directly computed by viewing X as a complex variety. Let $I_X = \langle f_1, \ldots, f_k \rangle$ be its prime ideal and $\mathcal{J} = (\partial f_i / \partial x_j)$ the $k \times n$ Jacobian matrix. Let c be the codimension of X. Then the singular locus X_{sing} of X is defined by the ideal

$$I_{X_{\text{sing}}} = I_X + \langle c \times c \text{ minors of } \mathcal{J} \rangle \tag{3}$$

The augmented Jacobian matrix $\widehat{\mathcal{J}}$ is the $(k+1) \times n$ matrix obtained by placing the row vector $(w_1(x_1 - u_1), \ldots, w_n(x_n - u_n))$ above the Jacobian, i.e.

$$\widehat{\mathcal{J}} = \begin{pmatrix} \mathbf{w} \cdot (\mathbf{x} - \mathbf{u}) \\ \mathcal{J} \end{pmatrix}. \tag{4}$$

The weighted critical ideal of X with respect to the data point $\mathbf{u}$ and weights $\mathbf{w}$ is the following saturation:

$$\mathcal{C}_{X,\mathbf{u},\mathbf{w}} = \left(I_X + (c+1) \times (c+1) \text{ minors of } \widehat{\mathcal{J}} \right) : \left(I_{X_{\text{sing}}} \right)^{\infty} \tag{5}$$

For general data points $\mathbf{u}$, the variety defined by this ideal is finite and it consists of the nonsingular critical points [4, Lemma 2.1] of the squared weighted distance function. Thus the degree of $\mathcal{C}_{X,\mathbf{u},\mathbf{w}}$ is exactly the weighted ED degree of X with respect to the weights $\mathbf{w}$. This method of computing ED degrees can be implemented symbolically and we call it the *minors method*.

2.2 Left Kernel Method

For varieties of large codimension, the minors method is computationally intensive due to the saturation step along with the possibility of there being exponentially many minors. To avoid incurring these costs, we introduce the *left kernel method*, which uses Lagrange multipliers to solve the equations defining critical points. We present this method for unit ED degrees, but the same ideas will apply to the weighted case.

Let X be an irreducible real algebraic variety defined by the polynomials $f_1, \ldots, f_k$ and denote $f(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) & \cdots & f_k(\mathbf{x}) \end{pmatrix}^T$. By introducing Lagrange multipliers $\lambda = [\lambda_0 : \lambda_1 : \cdots : \lambda_k] \in \mathbb{P}^k$, we define

$$G(\mathbf{x}, \lambda) = \begin{pmatrix} f(\mathbf{x}) \\ \lambda_0(\mathbf{x} - \mathbf{u}) + \sum_{i=1}^{k} \lambda_i \nabla f_i(\mathbf{x})^T \end{pmatrix} \tag{6}$$

The system $G(\mathbf{x}, \lambda) = 0$ consists of $n + k$ equations in $n + k$ degrees of freedom. If we assume that X is a complete intersection given by the polynomials $f_1, \ldots, f_k$, then for every pair $(\mathbf{x}, \lambda) \in V(G(\mathbf{x}, \lambda))$, the vector $\mathbf{x}$ is a critical point of the squared distance function [11]. Thus the number of isolated solutions to $G(\mathbf{x}, \lambda)$ is exactly the unit ED degree of X. In our implementation, we use the Bertini software package to solve this system numerically.

2.3 Homotopy Method

The left kernel method can be extended into what we call the *homotopy method*, which is especially useful for exploring the behavior of weighted ED degrees for special choices of weights or data. Let $F = \{f_1, \ldots, f_k\}$ and $G = \{g_1, \ldots, g_m\}$ be two sets of polynomials such that $V(G)$ is a complete intersection containing $V(F)$ as an irreducible component. Denote by $\mathcal{J}_G$ the Jacobian of G and introduce Lagrange multipliers $\lambda = [\lambda_0 : \lambda_1 : \cdots : \lambda_m] \in \mathbb{P}^m$. We seek to solve the system

$$M(\mathbf{x}, \lambda) = \begin{pmatrix} \lambda_0 & \cdots & \lambda_m \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{x}} \|\mathbf{x} - \mathbf{u}\|^2_{\mathbf{w}} \\ \mathcal{J}_G \end{pmatrix} \tag{7}$$

subject to $\mathbf{x} \in V(F)$. The number of such solutions $(\lambda, \mathbf{x})$ is the ED degree of $V(F)$. Using homotopy continuation, we do this computation in two stages:

1. We construct a start system of the form (7) using generic weights. We then leverage Bertini to solve the start system to obtain an initial bound on the ED degree of $V(F)$.
2. Using the results of stage one, we perform a parameter homotopy to track the solutions from the start system to the target system. By filtering out points that correspond to infinity or do not vanish on F, we get the desired ED degree.

The two-stage approach allows one to efficiently compute ED degrees for multiple choices of weights $\mathbf{w}$ or data $\mathbf{u}$ because the first stage only needs to be executed once. Correctness of the left-kernel and homotopy formulations assumes that the relevant input polynomials define a complete intersection: F for the left-kernel method, and G for the homotopy method.

2.4 Conormal Varieties

Suppose instead that the variety X is given by a parameterization. More precisely, let $\phi_1, \ldots, \phi_n$ be polynomials in d variables. We assume $n > d$ and define the map

$$\phi : \mathbb{C}^d \to \mathbb{C}^n \qquad \mathbf{x} \mapsto (\phi_1(\mathbf{x}), \ldots, \phi_n(\mathbf{x})) \tag{8}$$

The variety X is the Zariski closure of $\operatorname{im}\phi$. Suppose $\dim X = d$ so that the Jacobian map

$$\nabla_{\mathbf{x}}(\phi_1, \ldots, \phi_n) : \mathbb{C}^d \to \mathbb{C}^{n\times d} \qquad \mathbf{y} \mapsto \left(\frac{\partial \phi_i}{\partial x_j}(\mathbf{y})\right) = \mathcal{J}(\mathbf{y}) \tag{9}$$

is generically full rank. The points for which the Jacobian is not full rank form the set of critical points for the parameterization (8). If $\mathbf{y} \in \mathbb{C}^d$ is not a critical point, then the columns of $\mathcal{J}(\mathbf{y})$ span a d-dimensional subspace of $\mathbb{C}^n$. We globally describe the orthogonal complement of this subspace by finding a spanning set for $\ker \mathcal{J}^T$.

Since the kernel may be over-parameterized, we evaluate its generators at a general point $\mathbf{y} \in \mathbb{C}^d$. We then find a subset of $n-d$ elements which span a $(n-d)$-dimensional subspace and select the corresponding generators of $\ker \mathcal{J}^T$. Let $M \in (\mathbb{C}[\mathbf{x}])^{n\times(n-d)}$ be the matrix whose columns are those elements and define the parametric map

$$\Psi_M : \mathbb{C}^d \times \mathbb{C}^{n-d} \to \mathbb{C}^n \times \mathbb{C}^n \qquad (\mathbf{x}, \lambda) \mapsto \left(\phi(\mathbf{x}), M(\mathbf{x}) \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{n-d} \end{pmatrix}\right) \tag{10}$$

Under the given assumptions, the image of Ψ_M is n-dimensional and we call its Zariski closure an *affine conormal variety*. This construction can be used to compute ED degrees for varieties given via parameterizations. We wish to find points $\mathbf{x}$ such that

$$\sum_{i=1}^{n} 2w_i(\phi_i(\mathbf{x}) - u_i)\frac{\partial \phi_i}{\partial x_j}(\mathbf{x}) = 0 \qquad j = 1, \ldots, d \tag{11}$$

This is equivalent to the statement

$$\mathcal{J}(\mathbf{x})^T \begin{pmatrix} 2w_1(\phi_1(\mathbf{x}) - u_1) \\ \vdots \\ 2w_n(\phi_n(\mathbf{x}) - u_n) \end{pmatrix} = 0 \tag{12}$$

Using the parameterization (10), we see that critical points are exactly the pairs $(\mathbf{x}, \lambda)$ which satisfy

$$\begin{pmatrix} 2w_1(\phi_1(\mathbf{x}) - u_1) \\ \vdots \\ 2w_n(\phi_n(\mathbf{x}) - u_n) \end{pmatrix} = M(\mathbf{x}) \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{n-d} \end{pmatrix} \tag{13}$$

The ideal generated by these n equations is the critical ideal of the parameterized variety X and its degree is the weighted ED degree of X.

3 Implementations

The `EuclideanDistanceDegree` package implements the four methods for computing ED degrees described in Section 2. Each method is implemented across three variants with differing levels of generality:

- `Weight`: user specified weights and data
- `Unit`: unit weights and random data
- `Generic`: random weights and data

All numerical methods rely on homotopy continuation and may underestimate the ED degree due to path failures, e.g. when paths diverge to infinity. The package writes the corresponding Bertini input files to a temporary directory. This directory can be changed using the `TempDirectory` option, allowing users to inspect, modify, and rerun Bertini computations. Users with Bertini experience may mitigate the possibility of path failures by modifying the configuration in these files, for example by changing tolerance or step sizes.

3.1 Minors Method: Symbolic Computation

To compute ED degrees symbolically using the minors method as described in Section 2.1, the package includes the three methods `symbolicWeightEDDegree`, `determinantalUnitEDDegree`, and `determinantalGenericEDDegree`. The latter two methods take as input a list of polynomials defining an irreducible variety and computes the corresponding ED degree. For additional control over the computation, the `symbolicWeightEDDegree` method also takes a data point and weight vector.

One distinctive feature of these methods is the ability to return the equations defining the critical ideal (5). This is done by setting the optional argument `ReturnCriticalIdeal=>true`. This functionality is particularly important for replication with other software packages like `HomotopyContinuation.jl` [3]. As an example, suppose we want to compute the ED degree of the Dingdong surface from Herwig Hauser's algebraic surfaces gallery [12]. We can compute the unit and generic ED degrees and recover the critical ideal using the following code:

```
i1: R = QQ[x, y, z];
i2: F = {x^2 + y^2 + z^3 - z^2};
i3: determinantalUnitEDDegree F
o3: 5
i4: determinantalGenericEDDegree F
o4: 9
i5: (U, W) = ({1,12,2}, {1,1,2});
i6: symbolicWeightEDDegree(F, U, W, ReturnCriticalIdeal => true)
o6 = ideal (12x - y, 145y^2 - 435y*z + 288z^2 - 870y + 4548z - 3096, 48z^3
    + 145y*z - 144z^2 + 290y - 1516z + 1032, 3y*z^2 - 6y*z - 36z^2 + 8y +
    24z)
o6: Ideal of R
```

3.2 Left Kernel Method: Complete Intersections

For varieties that are complete intersections, the left kernel method may be used as described in Section 2.2. The package implements the left kernel method in the `leftKernelWeightEDDegree` method along with its unit and generic variants. For example, to compute the weighted ED degree of the Daisy surface using specific data and weights, we use the following code:

```
i1: R = QQ[x, y, z];
i2: F = {(x^2 - y^3)^2 - (z^2 - y^2)^3};
i3: (U, W) = ({.12, .23, .25}, {.15, .331, .727});
```

```
i4: dir = temporaryFileName();
i5: leftKernelWeightEDDegree(F, U, W, TempDirectory => dir)
o5: 22
```

3.3 Homotopy Method: Beyond Complete Intersections

To compute the ED degree using the homotopy method as described in Section 2.3, the package includes the method `numericWeightEDDegree` along with its unit and generic variants. These methods differ from those described in the previous section in that they take two lists of polynomials: a list F which describes the variety of interest and another list G such that the variety $V(F)$ is an irreducible component of $V(G)$. For example:

```
i1: R = QQ[x, y];
i2: F = G = {x^2 + y^2 - 1};
i3: dir = temporaryFileName();
i4: numericUnitEDDegree(F, G, TempDirectory => dir)
o4: 2
```

These methods are all special cases of the `homotopyEDDegree` method, which is a configurable implementation of the homotopy method. The homotopy is done in the two stages described in Section 2.3: the first stage solves the start system while the second stage verifies solutions on the target system via path tracking. Configuration options are stored in a `NumericalComputationOptions` object and keys may be modified to customize the homotopy. Currently we support performing homotopy continuation on weights and data. As an example, we present a weight homotopy computation of a unit and generic ED degree. Note how after the unit degree is computed, only the second stage needs to be executed to compute the generic ED degree:

```
i1: R = QQ[x1,x2,x3,x4,x5,x6];
i2: F = (minors(2, genericMatrix(R,3,2)))_*;
i3: G = drop(F, -1);
i4: NCO = newNumericalComputationOptions(F, G);
i5: NCO#"TargetWeight" = apply(#gens R, i->1);
i6: homotopyEDDegree(NCO, "Weight", true, true)
o6: 2
i7: NCO#"TargetWeight" = apply(#gens R, i->random RR);
i8: homotopyEDDegree(NCO, "Weight", false, true)
o8: 10
```

The `NumericalComputationOptions` object allows for custom Bertini options to be specified in the `BertiniStartFiberSolveConfiguration` key. Thus path failures from the `homotopyEDDegree` method can be addressed directly in *Macaulay2* rather than through the Bertini input files, unlike the methods introduced previously.

Because solutions can be tracked to different data points without recomputing the initial stage, `homotopyEDDegree` enables efficient ED degree experiments. For convenience, we include the `averageNumericEDDegree` method to streamline the process. This method performs a stage-one solve at random initial data, generates n random data points using a user-provided sample generator, performs a stage-two solve for each sample, counts the number of real critical points, and returns the average. In [4, Example 4.5], the average real ED degree of an ellipse was computed to approach 3.05 using 10^5 samples. As a demonstration, we show how this experiment can be run for 100 samples using our package:

```
i1: loadPackage "Probability";
i2: R = QQ[x,y];
i3: F = G = {x^2 + 4*y^2 - 4};
i4: Z = normalDistribution();
i5: sampleGen = () -> apply(#gens R, i -> random Z)
i6: averageNumericEDDegree(F, G, sampleGen, 100);
o6 = 3.41
o6: RR (of precision 53)
```

3.4 Conormal Method: Parameterizations

For parameterized varieties, the conormal-variety approach from Section 2.4 is implemented in the method `parametrizedWeightEDDegree` along with its unit and generic variants. These methods take as input a list of polynomials F which parameterize a variety X and then returns the ED degree of X. For example:

```
i1: R = QQ[x1, x2, x3, x4, x5];
i2: F = {x1^2+x4^2, x2^2+x5^2, x3^2+1, x1*x2+x4*x5, x1*x3+x4, x2*x3+x5};
i3: parameterizedGenericEDDegree F
o3: 52
```

By default this is done symbolically. Setting the option `UseMonodromy=>true` will instead leverage the `MonodromySolver` [6] package to solve the critical equations.

4 Examples

We conclude with two application areas, polynomial neural networks and multiview varieties, that illustrate the package's utility for parameterized models arising from applied algebraic geometry.

4.1 Polynomial Neural Networks

As a first application, we compute ED degrees for shallow polynomial neural networks (PNNs). This follows the work done in [14], which we shall briefly describe here. An L layer PNN architecture is given by a vector $\mathbf{d} = (d_0, \ldots, d_L)$ which describes layer widths and a positive integer r specifying the activation degree. The PNN corresponding to the parameters $(\mathbf{d}, r)$ is the map

$$p_{(\mathbf{d},r)} = W_L \circ \sigma_{L-1} \circ \cdots \circ \sigma_1 \circ W_1 : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L} \tag{14}$$

where $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ are linear maps and the activation functions σ_i are applied coordinate-wise, i.e.

$$\sigma_i(\mathbf{x}) = (x_1^r, \ldots, x_n^r)$$

The resulting map $p_{(\mathbf{d},r)}$ can be thought of as an element of $\mathrm{Sym}_{r^{L-1}}(\mathbb{R}^{d_0})^{d_L}$, the space of tuples of degree r^{L-1} homogeneous polynomials in d_0 variables. The closure of the image of the parameterization map in this space is the *neurovariety* corresponding to the architecture $\mathbf{d}$. Using our package, we compute that the $\mathbf{d} = (3, 1, 1)$ network with square activation has generic ED degree 13. The architecture $\mathbf{d} = (3, 2, 1)$ also has generic ED degree 13 for square activations.

4.2 Multiview Varieties

Multiview varieties arise in the study of computer vision through algebraic geometry. In this setting, the process of taking a picture can be modeled as a rational map $\mathbb{P}^3 \dashrightarrow \mathbb{P}^2$. Here we follow the definition put forth in [7,8]: let $\mathbf{C} = (C_1, \dots C_n)$ be a tuple of full rank $(h+1)\times(N+1)$ matrices. Define the map

$$\Phi_{\mathbf{C}} : \mathbb{P}^N \dashrightarrow (\mathbb{P}^h)^n \qquad X \mapsto (C_1X, \dots, C_nX) \tag{15}$$

We call the tuple $\mathbf{C}$ a *camera arrangement*. Indeed for $N = 3$ and $h = 2$, $\Phi_{\mathbf{C}}$ is just the process of taking n pictures with n cameras. The *(point) multiview variety* of $\mathbb{P}^N$ with respect to $\mathbf{C}$ is the Zariski closure of the map $\Phi_{\mathbf{C}}$, denoted $\mathbf{C}\square\mathbb{P}^N := \overline{\operatorname{im}\Phi_{\mathbf{C}}} \subset (\mathbb{P}^h)^n$.

More generally, for a subvariety $Y \subseteq \mathbb{P}^N$, the *(point) multiview variety anchored at* Y with respect to $\mathbf{C}$ is the closure

$$\mathbf{C}\square Y := \overline{\Phi_{\mathbf{C}}(Y)}.$$

The restriction of a multiview variety to the standard affine chart is denoted $(\mathbf{C}\square Y)_{\text{Aff}}$ and is called the *affine multiview variety* of $\mathbf{C}\square Y$. We are interested in the case where Y is a degree E rational surface parameterized by

$$f : \mathbb{P}^2 \to \mathbb{P}^N \qquad [s : t : u] \mapsto [f_0(s,t,u) : \cdots : f_N(s,t,u)]. \tag{16}$$

Using our package, we compute for $E = 1, N = 2, h = 2, 3, 4$ that the ED degree of the affine multiview variety for a rational surface is 8 and when $E = 2, N = 2, h = 2$, the ED degree is 12. The case for a rational curve was worked out in [8, Theorem 2.3] where it is proved that the ED degree is $3En - 2$, notably independent of h. Our computations here provide evidence that a similar result may hold in the case of a rational surface.

4.3 Comparisons with Existing Software

Several software packages are available for computing, or helping to compute, ED degrees of algebraic varieties symbolically. For instance the *Macaulay2* package `AlgebraicOptimization` [10] is a general-purpose symbolic package for algebraic optimization problems with dedicated methods for ED degree computations. These methods involve methods based on multidegrees, projections, as well as Fritz John formulations, the latter being similar to the equations we use in the left kernel formulation. The *Macaulay2* package `ToricInvariants` [13] computes the generic ED degree of a projective toric variety along with other invariants of the dual variety. It provides a specialized tool for ED degree computations in the toric setting.

To our knowledge, there is no dedicated package for numerically computing ED degrees. However, homotopy continuation solvers such as Bertini [2], PHCpack [15], the *Macaulay2* package `MonodromySolver` [5,6], and the Julia package HomotopyContinuation.jl [3] can compute an ED degree by forming the critical equations for the squared distance function to a generic point and counting the isolated solutions numerically. The option `ReturnCriticalIdeal=>true` (see Section 3.1) allows our package to interface with other software by returning the generating polynomials of the critical ideal (5).

Code Availability Code for the provided examples is available in the `Examples` folder of the GitHub repository https://github.com/JoseMath/EuclideanDistanceDegree.

Acknowledgments. This research was partially supported by the Alfred P. Sloan Foundation and by the National Science Foundation grant DMS-2510307. We are grateful to the anonymous reviewers for their valuable comments and suggestions, which helped improve the quality of this paper.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bates, D.J., Gross, E., Leykin, A., Rodriguez, J.I.: Bertini for Macaulay2 (2013), https://arxiv.org/abs/1310.3297
2. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu. https://doi.org/10.7274/R0H41PB5
3. Breiding, P., Timme, S.: HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) Mathematical Software – ICMS 2018. pp. 458–465. Springer International Publishing, Cham (2018)
4. Draisma, J., Horobeţ, E., Ottaviani, G., Sturmfels, B., Thomas, R.R.: The Euclidean distance degree of an algebraic variety. Foundations of Computational Mathematics **16**(1), 99–149 (Feb 2016). https://doi.org/10.1007/s10208-014-9240-x
5. Duff, T., Hill, C., Jensen, A., Lee, K., Leykin, A., Sommars, J.: Solving polynomial systems via homotopy continuation and monodromy. IMA J. Numer. Anal. **39**(3), 1421–1446 (2019). https://doi.org/10.1093/imanum/dry017
6. Duff, T., Hill, C., Jensen, A.N., Lee, K., Leykin, A., Sommars, J.: MonodromySolver: solving polynomial systems via monodromy. Version 1.16. A *Macaulay2* package available at "http://www.math.gatech.edu/ leykin"
7. Duff, T., Rydell, F.: Metric multiview geometry – a catalogue in low dimensions (2024), https://arxiv.org/abs/2402.00648
8. Finkel, B., Rodriguez, J.I.: The euclidean distance degree of one-parameter anchored multiview varieties (2026), https://arxiv.org/abs/2512.18521
9. Grayson, D.R., Stillman, M.E.: Macaulay2, a software system for research in algebraic geometry. Available at https://www.macaulay2.com
10. Härkönen, M., Hollering, B., Kashani, F.T., Rodriguez, J.I.: Algebraic optimization degree. ACM Commun. Comput. Algebra **54**(2), 44–48 (Sep 2020). https://doi.org/10.1145/3427218.3427222
11. Hauenstein, J.D.: Numerically computing real points on algebraic sets. Acta Applicandae Mathematicae **125**(1), 105–119 (Jun 2013). https://doi.org/10.1007/s10440-012-9782-3
12. Hauser, H.: Algebraic surfaces gallery, https://homepage.univie.ac.at/herwig.hauser/bildergalerie/gallery.html
13. Helmer, M., Sturmfels, B.: Nearest points on toric varieties. Math. Scand. **122**(2), 213–238 (2018). https://doi.org/10.7146/math.scand.a-101478
14. Kubjas, K., Li, J., Wiesmann, M.: Geometry of polynomial neural networks. Algebraic Statistics **15**(2), 295–328 (Dec 2024). https://doi.org/10.2140/astat.2024.15.295
15. Verschelde, J.: Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. ACM Trans. Math. Softw. **25**(2), 251–276 (Jun 1999). https://doi.org/10.1145/317275.317286