

# LocalDimFinder

Jonathan D. Hauenstein

March 2, 2010

# LocalDimFinder

## 1 Introduction to LocalDimFinder

LocalDimFinder is a module of Bertini [2] that can be used to compute the  $\hat{d}$ -bounded local dimension described in [5]. That article also provides more details regarding the algorithm utilized. In summary, the algorithm is based on the Macaulay matrix reintroduced in [3] and the local dimension test of [1].

## 2 Using LocalDimFinder

LocalDimFinder has 5 input arguments:

- Complex analytic system  $F : \mathbb{C}^N \rightarrow \mathbb{C}^m$ ,
- Point  $z \in V(F)$ ,
- Depth bound (integer)  $\hat{d}$ ,
- Lower and upper dimension bounds (integer)  $0 \leq D_0 \leq D_1$ .

The main numerical output of the algorithm is the  $\hat{d}$ -bounded local dimension if it is between  $D_0$  and  $D_1$ . LocalDimFinder also displays the coranks of the Macaulay matrices for dimension  $k \in [D_0, D_1]$  up to depth  $\hat{d}$ .

The analytic system  $F$  is specified to LocalDimFinder using an `input` file with a syntax similar to `input` files for Bertini. As an example, Table 1 shows an `input` file for LocalDimFinder for the system  $\begin{bmatrix} xy^2 \\ xz \end{bmatrix}$  defined on  $\mathbb{C}[x, y, z]$ .

The other four input arguments are read into LocalDimFinder from a `test_points` file that has a similar form to a `start` file utilized by Bertini. The first line of `test_points` indicates the number of points to test. For each point to test, there is a block containing the coordinates of the point, the depth bound  $\hat{d}$ , and the dimension bounds  $D_0$  and  $D_1$ . Table 2 is a `test_points` file containing two points, namely  $(0, 1 + i, 1 - i)$  and  $(2 - 4i, 0, 0)$ , with  $\hat{d} = 3$ ,  $D_0 = 0$ , and  $D_1 = 2$  for both points.

LocalDimFinder can be launched with 0, 1, or 2 command line arguments. With no arguments, LocalDimFinder looks for the default file names “input” and “test\_points”. Otherwise, the

Table 1: Example of `input` file

```

CONFIG

% Optional configurations (see Appendix A)

END;
INPUT

variable  $x, y, z$ ;
function  $f, g$ ;

 $f = x * y \wedge 2$ ;
 $g = x * z$ ;

END;

```

Table 2: Example of `test_points` file

```

2
0 0
1 1
1 -1
3
0 2

2 -4
0 0
0 0
3
0 2

```

first command line argument is the `input` file name and the second one is the `test_points` file name. For example, the command

```
>> ./localDimFinder inputCubic12bar test_pointsCubic12bar
```

executes `LocalDimFinder` using the `input` file named “inputCubic12bar” and `test_points` file named “test\_pointsCubic12bar”.

### 3 Output of `LocalDimFinder`

`LocalDimFinder` displays output to the screen as well as in an `output` file and a `local_dim` file. For each point, the `output` file contains its coordinates, the value of the system evaluated at the

point, a summary of the coranks of the Macaulay matrices, and a conclusion statement regarding the (depth-bounded) local dimension, multiplicity, and depth.

The `local_dim` file contains the number of points tested and, for each point, its (depth-bounded) local dimension, multiplicity, and depth. If there was an error, all three values for that point will be set to  $-1$ .

Consider the example described in Tables 1 and 2, namely compute the 3-bounded local dimension of  $\begin{bmatrix} xy^2 \\ xz \end{bmatrix}$  at  $(0, 1 + i, 1 - i)$  and  $(2 - 4i, 0, 0)$ . The data contained in the `output` file generated by LocalDimFinder is summarized in Table 3. Table 4 is the `local_dim` file generated by LocalDimFinder for this example. In particular, we see that the 3-bounded local dimensions of  $(0, 1 + i, 1 - i)$  and  $(2 - 4i, 0, 0)$  are 2 and 1, respectively, with multiplicity 1 and 2, respectively, and depth 0 and 1, respectively.

Table 3: Summary of data in `output`

Point	F(Point)	$k$	$c_{k,d}$	3-bounded local dim	multiplicity	depth
$(0, 1 + i, 1 - i)$	$(0,0)$	0	1,3,6,10	2	1	0
	$(0,0)$	1	1,2,3,4			
	$(0,0)$	2	1,1,1,1			
$(2 - 4i, 0, 0)$	$(0,0)$	0	1,3,5,7	1	2	1
	$(0,0)$	1	1,2,2,2			
	$(0,0)$	2	1,1,1,1			

Table 4: Example of `local_dim` file

2
2 1 0
1 2 1

## 4 A word of caution regarding LocalDimFinder

Since the input points are usually only approximate solutions to the complex analytic system, it is the user's responsibility to compute them accurately enough for LocalDimFinder to provide a meaningful answer.

## 5 Future of LocalDimFinder

In the future, LocalDimFinder will fully implement the closedness subspace of [6] to more efficiently compute the coranks of the Macaulay matrices.

# Bibliography

- [1] D.J. Bates, J.D. Hauenstein, C. Peterson, and A.J. Sommese. A numerical local dimension test for points on the solution set of a system of polynomial equations. *SIAM Journal on Numerical Analysis*, 47:3608–3623, 2009.
- [2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: Software for numerical algebraic geometry. Available at <http://www.nd.edu/~sommese/bertini>.
- [3] B. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In *Proc. of ISSAC 2005*, pages 166–123, New York, 2005. ACM.
- [4] A.J. Sommese and C.W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, Singapore, 2005.
- [5] C.W. Wampler, J.D. Hauenstein, and A.J. Sommese. Mechanism mobility and a local dimension test. Preprint, 2010.
- [6] Z. Zeng. The closedness subspace method for computing the multiplicity structures of a polynomial system. In D.J. Bates, G.-M. Besana, S. Di Rocco, and C.W. Wampler, editors, *Interactions of Classical and Numerical Algebraic Geometry*, Contemporary Mathematics 496, pages 347–362. 2009.

# Appendix A

## Configurations

LocalDimFinder, just like Bertini, is configurable. The configurations for LocalDimFinder are presented below along with a brief description.

Table A.1: Configurations for LocalDimFinder

NAME	ACCEPTABLE VALUES	DEFAULT VALUE
MPTYPE	0, 1, 2	2
PRECISION	$\geq 64$ and $\leq 3328$	96
SINGVALZEROTOL	$> 0$	1e-12
TARGETTOLMULTIPLIER	$\geq 1$	10
FINALTOL	$> 0$	1e-11

### Configurations

- MPTYPE

Indicates the level of precision to be used. Set to 0 for regular precision, 1 for higher precision, or 2 for adaptive precision. If MPTYPE is 1, the user should also set PRECISION to the desired level of precision.

- PRECISION

If MPTYPE is 1, PRECISION indicates the level of precision to use (in bits). [If MPTYPE is either 0 or 2, it has no effect.] Standard settings include 64 bits (roughly 19 decimal digits), 96 (28), 128 (38), 160 (48), 192 (57), 224 (67), and 256 (77). In general, N bits is equivalent to  $\left\lceil N \frac{\ln(2)}{\ln(10)} \right\rceil$  decimal digits. The maximum is 3328 bits (roughly 1000 digits).

- SINGVALZEROTOL

LocalDimFinder uses two methods for computing the ranks of the Macaulay matrices. The first method, which is only applied when using fixed precision (i.e., MPTYPE is either 0 or 1), is to declare singular values less than SINGVALZEROTOL to be zero. The second method, which is always applied, is discussed in the description of FINALTOL.

- TARGETTOLMULTIPLIER

If the function residual of each test point is greater than  $\text{TARGETTOLMULTIPLIER} \times \text{FINALTOL}$ , LocalDimFinder will declare that the test point does not sufficiently satisfy the system and is declared a failure.

- **FINALTOL**

LocalDimFinder uses FINALTOL in three ways:

1. to determine if each test point sufficiently satisfies the system,
2. to adjust the precision when using adaptive precision (i.e., MPTYPE is 2), and
3. to determine the rank of the Macaulay matrices.

See the description of TARGETTOLMULTIPLIER for details on item 1.

When using adaptive precision, the precision used for the computations is set to be large enough to handle  $-\log_{10}(\text{FINALTOL})$  decimal digits. For example, if MPTYPE is set to 2 and FINALTOL is set to  $10^{-20}$ , the computations will be performed using 96-bit precision (roughly 28 decimal digits).

The other method for rank determination (the first was discussed in the description of SINGVALZEROTOL) is to compare the singular values of the Macaulay matrices using different approximations of the given test point. In the setting of [1], different approximations can be easily obtained from the endgame data. LocalDimFinder uses random perturbations of the test point on the order of FINALTOL to compute different approximations.