Computing the Newtonian Graph

(Extended abstract)

Dexter Kozen* Kjartan Stefansson*

kozen@cs.cornell.edu stefan@cs.cornell.edu

June 15, 1994

Abstract

A polynomial $f \in \mathbb{C}[z]$ defines a vector field $N_f(z) = -f(z)/f'(z)$ on $\mathbb{C}$. Certain degenerate curves of flow in N_f give the edges of the Newtonian graph, as defined by [6]. These give a relation between the roots of f and f', much similar to the linear order, when f has real roots only.

We give a purely algebraic algorithm to compute the Newtonian graph and the basins of attraction in the Newtonian field. The resulting structure can be used to query whether two points in $\mathbb{C}$ are within the same basin of attraction. This gives us an algebraic approach to root-finding using Newton's method. This method extends to rational functions and more generally to any functions on $\mathbb{C}$ whose flow is algebraic over $\mathbb{C}(e)$.

1 Introduction

We follow the definitions of Smale [6] and define the *Newtonian vector field* of a polynomial $f \in \mathbb{C}[z]$ by $N_f(z) = -\frac{f(z)}{f'(z)}$. The name is derived from the fact that $x_{k+1} \leftarrow x_k + N_f(x_k)$ is Newton's method for root approximation.

The vector field N_f defines a flow on $\mathbb{C}$ where the flow comes (almost everywhere) from a pole (in the case of f a polynomial, from infinity) and

*Computer Science Department, Cornell University, Ithaca, NY 14853

converges (almost everywhere) to a root of f. There are degenerate curves of flow connecting roots of f and f', and the basins of flow split $\mathbb{C}$ into finitely many regions. These connecting curves will be the edges of our Newtonian graph (to be defined more formally later). This graph has been studied and the possible graphs that can arise have been classified for polynomials [7].

We will give a symbolic algorithm to compute the graph from the polynomial. The output of the algorithm is a structure that can act as an oracle to answer simple questions such as

1. Given $a, b \in \mathbb{C}$, are a and b in the same basin?
2. Given $a, b \in \mathbb{C}$, are a and b on the same curve of flow?
3. Given $a \in \mathbb{C}$, is a on a basin boundary?
4. Given $a \in \mathbb{C}$, is a on a graph edge?

So not only do we get the topology of the graph, we get a method for membership testing for the interesting regions in the field. Such a structure can for instance be used in a "guaranteed" Newton's method, modifying the step size at every point to ensure that we stay within a basin.

We then sketch how to extend the definition of a Newtonian graph for rational functions. We also observe that the resulting fields on $\mathbb{C}$ satisfy certain algebraic conditions. Given such conditions we can define the graph and compute it.

2 The Newtonian Graph

We have defined the Newtonian field of a polynomial. A vector field such as N_f on $\mathbb{C}$ defines a flow on $\mathbb{C}$. Given $z \in \mathbb{C}$ the flow through z is a function $\phi_z : I \to \mathbb{C}$, where $I \subseteq \mathbb{R}$ is an interval containing zero, ϕ_z differentiable with

$$\begin{aligned} \frac{d\phi_z(t)}{dt} &= N_f(\phi_z(t)) \\ \phi_z(0) &= z. \end{aligned}$$

That is, ϕ parameterizes the flow starting at z and at every point the speed and direction agrees with the field. An example of flow of f of degree 4 is given in Figure 1.

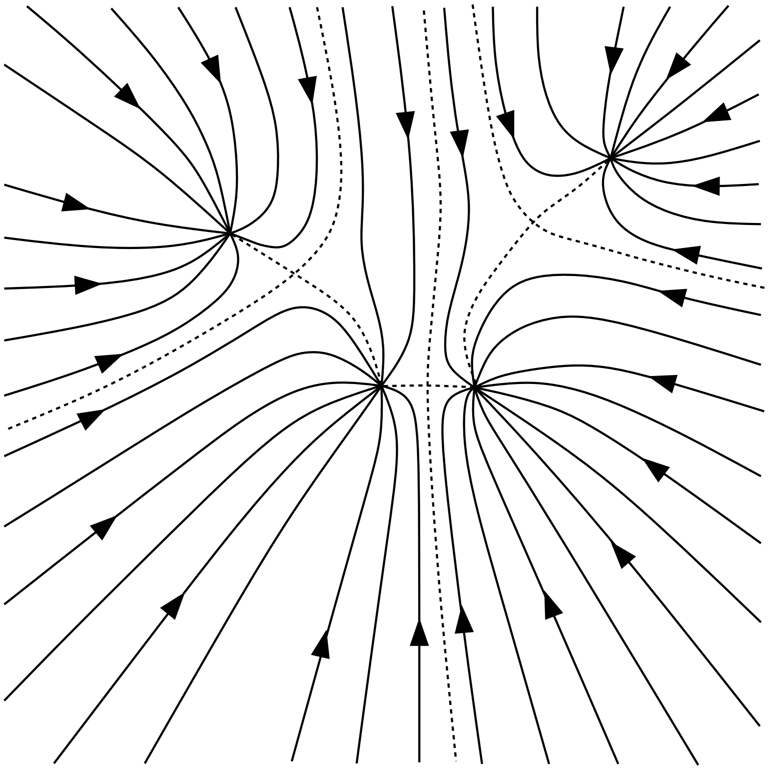

Figure 1: Flow in the Newtonian field of a degree 4 polynomial. Every curve of flow is directed to a root, except the basin boundaries (dotted lines). There is a root of f' on every basin boundary, and a curve of flow from there to "adjacent" roots (also dotted lines).

The flow exists on all of $\mathbb{C} - V_{f'}$ (where $V_{f'} = \{z \in \mathbb{C} \ : \ f'(z) = 0\}$). The existence and uniqueness follows from the theory of differential equations and the fact that N_f is a C^1 function on $\mathbb{C} - V_{f'}$ (see *e.g.* [3], §8.2 and §8.5).

The following lemma [7] gives us a very important property of the flow:

Lemma 1 *Let $f \in \mathbb{C}[z]$, and ϕ_z be flow through z in the Newtonian field N_f. Then f maps the curve $\{\phi_z(t) \ : \ t \in I\}$ to a straight line pointing to the origin. More specifically,*

$$f(\phi_z(t)) \quad = \quad e^{-t} f(z).$$

Proof. Computing $\frac{df(\phi_z(t))}{dt}$ using the chain rule gives:

$$\begin{aligned}\frac{df(\phi_z(t))}{dt} &= f'(\phi_z(t))\frac{d\phi_z(t)}{dt} \\ &= f'(\phi_z(t))N_f(\phi_z(t)) \\ &= -f(\phi_z(t)),\end{aligned}$$

which is a differential equation in t for the function $\rho(t) = f(\phi_z(t))$. Given the initial condition, $\rho(0) = f(z)$, it has the unique solution $\rho(t) = e^{-t}f(z)$, *i.e.* $f(\phi_z(t)) = e^{-t}f(z)$. □

Using the properties of ϕ one can show the following

Lemma 2 *For every $z \in \mathbb{C} - (V_f \cup V_{f'})$, ϕ_z is defined on a maximum interval containing 0, which is of the following type, for some $a, b \in \mathbb{R}$:*

1. *$(-\infty, +\infty)$, and the flow comes in from infinity and goes to a root of f;*
2. *$(-\infty, a)$ and the flow comes in from infinity and goes to a root of f';*
3. *(a, b) and the flow comes in from a root of f' and goes to another root of f';*
4. *$(a, +\infty)$ and the flow comes in from a root of f' and goes to a root of f.*

Proof. N_f is a C^1 function $W \to \mathbb{C}$ where $W = \mathbb{C} - (V_f \cup V_{f'})$, and by Theorem in §8.5 in [3], all flow must leave any compact set of W. By Lemma 1, the (maximum) interval of ϕ_z is unbounded upwards iff the flow goes to a root of f. The same argument shows that the interval is not downward bounded iff the flow comes in from ∞. Since the flow leaves any compact set of W the only other limit points are in $V_{f'}$. □

Definition 3 The *Newtonian Graph* of $f \in \mathbb{C}[z]$ is the plane graph $G = (V, E)$ with vertices $V = V_f \cup V_{f'}$ and directed edges being the curves of flow between vertices, where these exist. □

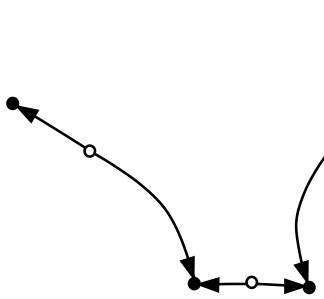

Figure 2: The graph of the field in Figure 1. The solid dots are the roots of the polynomial, the hollow ones are the roots of the derivative.

We note that the graph is not just a combinatorial structure, as the edges come with an embedding defined by ϕ.

Figure 2 shows the Newtonian graph of the field shown in Figure 1. Another example in Figure 3 shows that there can be connections between two roots of f'.

We observe that under f, every edge maps onto a straight line segment pointing to 0 in $\mathbb{C}$, with at least one endpoint in $\{f(c) \ : \ f'(c) = 0\}$. This is an immediate consequence of Lemma 1 and the fact that edges are curves of flow. We can conversely look at the pre-images (under f) of such line segments, and we get finitely many curves (at most $n(n-1)$, since f is an n to 1 mapping, and f' has at most $n-1$ roots). We will use this observation later, that the graph is contained in the pre-image $f^{-1}(\{mf(c) \ : \ f'(c) = 0, 0 \leq m \leq 1\})$. Thus the graph has finitely many edges. Furthermore [7] show that the graph is connected and go on to classify the possible types of graphs that can arise.

A basin of attraction is a connected region where the flow comes in from infinity and goes to one particular root of f. A basin boundary is the boundary of two basins. It is not hard to show that there must be a root of f' on every basin boundary, because it requires a discontinuity of N_f for the flow to "split" into two directions, and these are only at the roots of f'. Also the basin boundaries must be curves of flow themselves, so we conclude that every basin boundary is flow into a root of f'. In particular this means that basin boundaries are contained in the pre-image $f^{-1}(\{mf(c) \ : \ f'(c) = 0, 1 \leq m\})$.

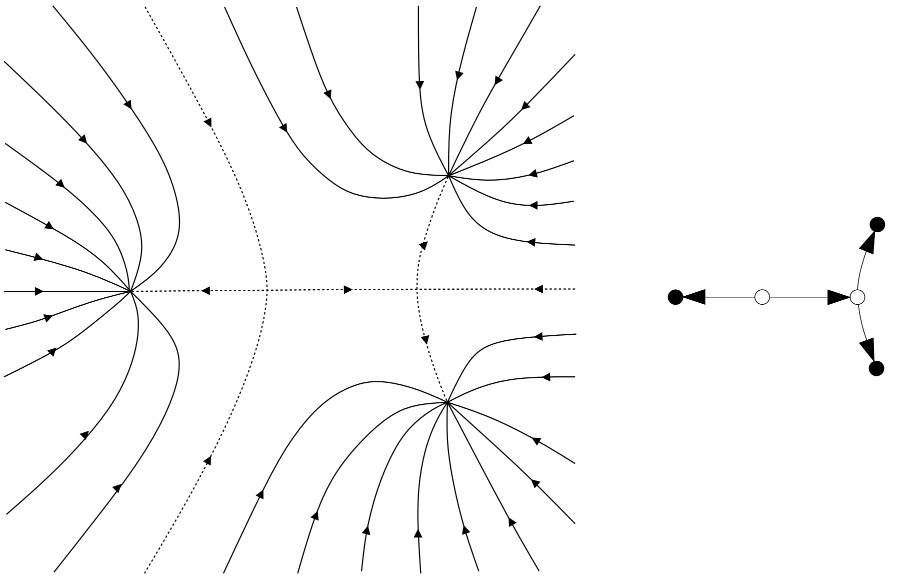

Figure 3: The field and the graph of a degree 3 (real) polynomial where the two derivative roots are linked.

3 Computing Basins and Graph Edges

We will give an algorithm to compute the basin boundaries and the edges of the Newtonian graph. But first we need a few preliminaries on cylindric algebraic decomposition.

3.1 Cell Decomposition

We describe cylindric algebraic cell decomposition briefly. For more detailed description, see [2] or [1].

Definition 4 A *decomposition* of R^k is a finite partition $\{C_i\}_{i\in I}$ such that each C_i is connected, $C_i \cap C_j = \emptyset$ if $i \neq j$ and $\bigcup_{i\in I} C_i = \mathbb{R}^k$. For $k = 1$ such a decomposition is *cylindric* if the each C_i is either a point or an interval. For $k > 1$, the decomposition is cylindric if for all $r, 1 \leq r \leq k$, $\{\pi_{1,..,r}(C_i) \ : \ i \in I\}$ is a cylindric decomposition of $\mathbb{R}^r$. $\square$

Definition 5 Given polynomial equations, $f_i(x_1, ..., x_m) = 0$, $i = 1, ..., n$, with $f_i \in \mathbb{R}[x_1, ..., x_m]$, a *Cylindric Algebraic Decomposition* (CAD) of $\mathbb{R}^m$ is a data structure $\mathcal{D}$ with the following properties.

- $\mathcal{D}$ contains a graph, where the nodes correspond to subsets (*cells*) of $\mathbb{R}^m$, each cell being homeomorphic to $\mathbb{R}^d$ for some d, and the cells are a decomposition of $\mathbb{R}^m$.
- For all $i = 1, .., n$, $\text{sign}(f_i)$ is constant on every cell. Each cell is labelled with the signs that the f_i take on that cell.
- Every node contains an oracle such that given any $c \in \mathbb{R}^m$, the oracle can answer if c is in the subset corresponding to the node.
- Every node contains dimension information, corresponding to the dimension of the cell.
- The edges of the graph correspond to adjacency of the cells in $\mathbb{R}^m$, *i.e.* there is an edge (u, v) if the subsets that u and v represent are adjacent.
- The decomposition is cylindric.

□

Algorithms have been developed to compute (parts of) such a cell decomposition dating back to Tarski in 1948 [8]. Collins [2] has a double exponential algorithm, although it lacks some of the adjacency information. Ben-Or *et al.* [1] developed a parallel algorithm giving the same kind of decomposition (the BKR algorithm), and Kozen and Yap [4] extended that algorithm to obtain the adjacency information as well (here after named the extended BKR algorithm).

We note that due to the cylindric condition and adjacency information, an algorithm computing such a decomposition can be used on a set of polynomials with quantifiers, projecting down the result. If the input is a system of polynomials of the form

$$\begin{aligned} \exists y_1, ...y_k \ : \quad & f_1(x_1, ..., x_m, y_1, .., y_k) = 0 \\ & \vdots \\ & f_n(x_1, ..., x_m, y_1, .., y_k) = 0 \end{aligned}$$

then using CAD on $\mathbb{R}^{m+k}$ we can project the solution down to $\mathbb{R}^m$, by treating the partitions according to $y_1, ..., y_k$ as insignificant. The resulting structure can be used for answering questions of the form: Given $c \in \mathbb{R}^m$, does there exist $y_1, .., y_k \in \mathbb{R}^k$ such that $y_1, ..., y_k, c$ is a solution to the system?

We note that the order of variables is important with respect to the cylindric condition.

3.2 The Algorithm

Recall that every basin boundary and every edge is mapped by f onto a straight line. Also, the basin boundaries and edges have a root of f' as a limit. Thus, all these "interesting" curves of flow satisfy, for every z on the curve,

$$\begin{array}{rrcl} \exists c \in \mathbb{C}, m \in \mathbb{R} \ : & f(z) & = & mf(c) \\ & f'(c) & = & 0 \end{array} \tag{1}$$

Any point z on a basin boundary or an edge must satisfy these two conditions. We note that the converse is not true; $z \in \mathbb{C}$ can be a solution to (1) without being on an edge or a basin boundary.

We proceed in two steps. First we find a decomposition of $\mathbb{C}$ describing where we have solutions z to (1). Then we prune that output, because we may get solution curves which do not correspond to basin boundaries or edges.

To find the solutions to (1), we can feed the equations

$$\begin{array}{rcl} f(z) & = & mf(c) \\ f'(c) & = & 0 \end{array} \tag{2}$$

into our favorite cylindric algebraic decomposition algorithm. The resulting structure would be a decomposition of $\mathbb{R} \times \mathbb{C} \times \mathbb{C}$ describing regions where such m, c, z exist, along with the dimension of each region and adjacency information. Projecting m and c, we get curves in $\mathbb{C}$ for which there exists a solution to (1).

First let us note that algorithms such as Collins' and the extended BKR do decomposition over the reals. But we can split the equations into a real and imaginary parts, and get a decomposition of $\mathbb{R}^5 \approx \mathbb{R} \times \mathbb{C} \times \mathbb{C}$, corresponding to the equations

$$f_R(x, y) \quad = \quad mf_R(c_1, c_2)$$

$$\begin{aligned}
f_I(x,y) &= m f_I(c_1,c_2) \\
f'_R(c_1,c_2) &= 0 \\
f'_I(c_1,c_2) &= 0
\end{aligned}$$

where $f(x+iy) = f_R(x,y) + i f_I(x,y)$ with $f_R, f_I \in \mathbb{R}[x,y]$. We get a decomposition on $\mathbb{R}^5$ which corresponds to a decomposition on $\mathbb{R} \times \mathbb{C} \times \mathbb{C}$.

We then project the dimensions corresponding to $c = c_1 + ic_2$ and again project m, obtaining a decomposition of $\mathbb{C}$ corresponding to z for which there exist m and c satisfying equation (2).

This decomposition will contain the basin boundaries and graph edges. These may be partitioned into segments (bounded 1-cells) and 0-cells between such segments. There may be other 1-cells present which are not solutions to the system (introduced by the CAD algorithm to get a finer partition). However, we can always identify the segments which are a solution to this system, because all of them are labelled with the signs of the input polynomials. The curves which are actual solutions to the system (2) will all show $f(z) = mf(c)$. Hence a solution curve to the system can be reconstructed by linking such adjacent cells.

But not all solution curves are edges or basin boundaries. The following lemma classifies the types:

Lemma 6 *The output from the process above contains at most $O(n^2)$ 1-cells which are solutions curves for the input system. They are of the following types:*

1. *Adjacent to two 0-cells, one of which describes a root of f' and one which describes a root of either f or f';*
2. *Adjacent only to one 0-cell which describes a root of f';*
3. *Adjacent only to one 0-cell which describes a root of f.*

Cells of type 1 are edges of the Newtonian graph, a cell of type 2 is a basin boundary and cells of type 3 are extraneous solutions to the system.

Proof. The only 1-dimensional cells that can be solutions to the system correspond to curves of flow. Then the classification is obvious from the definition of edges and the properties of basin boundaries.

There are at most $O(n^2)$ solution curves for $f(z) = mf(c)$ with c a root of $f'(c)$ and $m \in \mathbb{R}$, because there are at most $n-1$ roots of f' and f is an n to 1 mapping. $\square$

The cells of type 1 and 2 are the ones we are interested in and we must tell these apart from the extraneous cells of type 3.

Since f is a part of the input, the signs of f are given on every cell. In particular this allows us to verify if a curve ends at a root of f.

Depending on which algorithm we use, we may or may not have all the information needed. The extended BKR guarantees that if f is a part of the input, then the signs of f' will be provided on each cell. If we don't have this guarantee, we can always add $f'(z) = 0$ to our input equations and get the same information that way.

At this point we can determine the types of the solution curves. Now it is easy to implement the promised "pruning" step. We simply eliminate all cells of type 3. More precisely, we mark them as parts of the adjacent 2-dimensional cells (which are the basin that this cell lies in).

Now the structure can be used in answering queries. Two points are in the same basin if they are in the same 2-cell or if they are separated only by "fake" 1-cells (of type 3).

4 Improvements

Recall we did cylindric decomposition on $\mathbb{R}^5 \approx \mathbb{R} \times \mathbb{C}^2$ of the equations

$$\begin{aligned} f(z) &= mf(c) \\ f'(c) &= 0 \end{aligned}$$

and projected the solution onto $\mathbb{C}$. This can be simplified by defining

$$g(m, z) = Res_c(f(z) - mf(c), f'(c)),$$

where Res_c denotes the univariate resultant of two inputs, considered as polynomials in c. (Here we view $f(z) - mf(c)$ and $f'(c)$ as univariate polynomials in $\mathbb{C}[z, m][c]$).

Then g has the property that $g(m, z) = 0$ iff $\exists c \in \mathbb{C} \; : \; f(z) - mf(c) = 0 = f'(c)$. Hence, a decomposition of $\mathbb{R} \times \mathbb{C}$ with respect to g is the same as the projection of the decomposition of $\mathbb{R} \times \mathbb{C} \times \mathbb{C}$ with respect to the original two equations.

The only thing we must be aware of is how to obtain the necessary signs of f and f' on cells, in order to identify and link up solution curves and prune off the redundant ones. One way would be to add the equation $f(z) = 0$ (and $f'(z) = 0$, if we are not using the extended BKR), and do a decomposition with respect to f (f') and g. This is already an improvement in terms of dimensions, since we are only working with 3 real variables ($x = \mathrm{Re}(z)$,$y = \mathrm{Im}(z)$ and m) instead of 5 before.

The asymptotic complexity remains the same, but the constants are clearly much better. The extended BKR gives an NC circuit of depth $2^{O(d^2)} \log^{O(d)} n$ where d is the number of variables and n is the maximum of either the number of polynomials or their degrees. In our case the circuit will be of depth $O(\log^{O(1)} n)$ where n is the degree of the input polynomial f.

5 Applications

The Newtonian graph is of its own interest, as it describes the arrangement of the roots of f and f'. Our computation gives a complete topological information of both the graph and the basins of the Newtonian field.

The relation to Newton's method gives an interesting method of approximating all the roots of f simultaneously, guaranteeing convergence. If we start with a point z_0 in a basin, we can apply modified Newton's method, where the iteration $z_{k+1} \leftarrow z_k + N_f(z_k)$ is replaced by:

$\alpha \leftarrow 1$,
repeat
  $z_{k+1} \leftarrow z_k + \alpha N_f(z_k)$
  $\alpha \leftarrow \alpha/2$
until (z_{k+1} is in the same basin as z_k)

I.e. we scale down the step in order to ensure that we stay within the same basin. Here we use our pre-computed structure of basins to determine if two points are in the same basin. If we furthermore require that a progress is made at each step, (*i.e.* that $|f(z_{k+1})| < |f(z_k)|$, then we are guaranteed that the sequence $\{z_k\}$ will eventually converge to the root in the basin of z_0.

We remark that this does not guarantee quadratic convergence everywhere, it only ensures that the method will converge.

6 More General Newtonian Graphs

In this extended abstract, we will briefly describe how the Newtonian graph and its computation extend to more general vector fields on $\mathbb{C}$.

The definition of a Newtonian graph (Definition 3) only uses the fact the the function $f : W \longrightarrow \mathbb{C}$ is C^2 on an open subset $W \subseteq \mathbb{C}$, which makes $N_f(z) = -f(z)/f'(z)$ a C^1 vector field on W. Lemma 1 still holds, but Lemma 6 now allows curves parameterized $(-\infty, +\infty)$ coming in from either infinity or a pole of f and going to either infinity or a root of f.

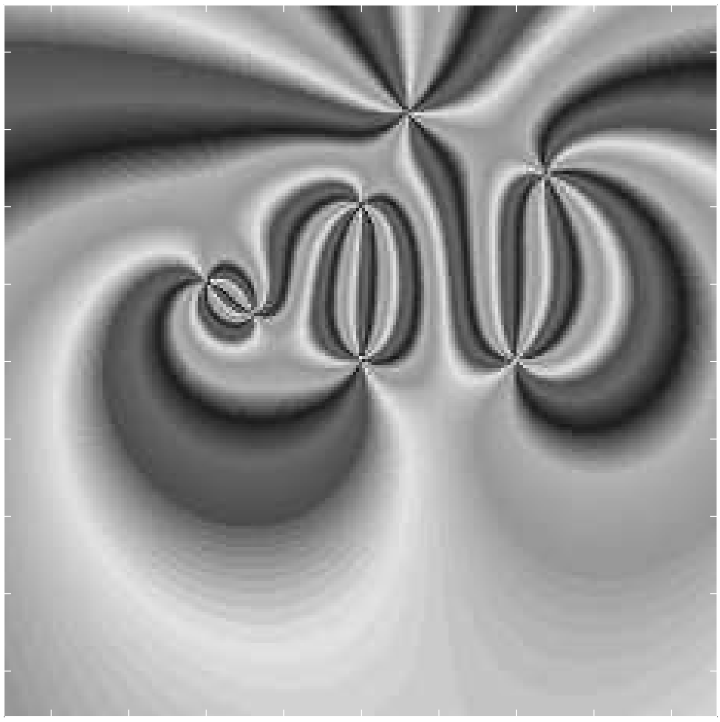

Figure 4: Flow in the Newtonian field of a rational function with three roots and four poles. Curves of fixed color indicate curves of flow.

In particular, most of the same observations hold for rational functions. The function f maps the curves of flow onto straight line segments pointing to the origin. The natural definition of a Newtonian graph for the rational function is the graph whose edges are the directed curves from poles or infinity to roots of f'; between the roots of f' and from roots of f' to roots of f or to infinity. This differs only slightly from Smale's definition [6] in that for polynomials we now count the basin boundaries as a part of the graph. A nice property of the graph is that it is symmetric in poles and roots, *i.e.* the graphs of $N_{p/q}$ and $N_{q/p}$ are identical, except the directions of the edges are reversed.

Write $f(z) = p(z)/q(z)$ with $p, q \in \mathbb{C}[z]$ with $\gcd(p, q) = 1$. If c is a root of $f'(c) = 0$ then f maps any edge into c onto a ray $\{mf(c) \ : \ 0 \leq m \leq 1\}$. Again we can consider a system of equations

$$\begin{aligned} \exists c \in \mathbb{C}, m \in \mathbb{R} \ : \ f(z) &= mf(c) \\ f'(c) &= 0 \end{aligned}$$

which now is equivalent to the system of polynomials

$$\begin{aligned} \exists c \in \mathbb{C}, m \in \mathbb{R} \ : \ p(z)q(c) &= mq(z)p(c) \\ p'(c)q(c) - p(c)q'(c) &= 0. \end{aligned}$$

This we can solve with algebraic decomposition as before and determine the actual solution curves which are edges. As before we can reduce the number of variables by using resultants. Let

$$g(m, z) \ = \ Res_c(p(z)q(c)mq(z)p(c), p'(c)q(c) - p(c)q'(c)).$$

Then the previous system is equivalent to $\exists m \ g(m, z) = 0$. As before we have devised an NC algorithm to compute the Newtonian graph.

The key property used in the computation is that the flow satisfies $f(\phi_z(t)) = e^{-t}f(z)$. In the case of a rational function $f(z) = p(z)/q(z)$ this translated into the polynomial equation

$$p(\phi_z(t))q(z) - e^{-t}p(z)q(\phi_z(t)) = 0.$$

This equation also implicitly defines the flow ϕ_z satisfying $\phi_z(0) = z$.

In more generality, consider any flow ϕ_z defined on $W \subseteq \mathbb{C}$, where W is almost all of $\mathbb{C}$. Assume for some polynomial g with complex coefficients and for all $z \in W$, $\phi_z(0) = z$ and $g(\phi_z(t), e^t, z) = 0$. Then we have $Dg(\phi_z(t), e^t, z) = 0$, *i.e.*

$$D_1 g(\phi_z(t), e^t, z)\phi'_z(t) + D_2 g(\phi_z(t), e^t, z) \quad = \quad 0$$

which is equivalent to

$$\phi'_z(t) \quad = \quad -D_2 g(\phi_z(t), e^t, z)/D_1 g(\phi_z(t), e^t, z). \tag{3}$$

If a basin of attraction is the area where all the flow goes from one pole to one root then as before there is a discontinuity in the field somewhere along every basin boundary. In particular, ϕ' will be undefined at such points. Equation (3) shows that $\phi'_z(t)$ extends continuously to all of $\mathbb{C}$ except the points where $D_1 g(\phi_z(t), e^t, z) = 0$ and $D_2 g(\phi_z(t), e^t, z) \neq 0$. These points can be computed as being the $x \in \mathbb{C}$ for which there exist $m, m' \in \mathbb{R}$ and $z, x' \in \mathbb{C}$ with

$$\begin{aligned} g(x, m, z) &= 0 \\ g(x', m', z) &= 0 \\ D_1 g(x', m', z) &= 0. \end{aligned}$$

The first two conditions force x and x' to be on the same curve whereas the third condition places x' at a discontinuity of the field. It is easy to verify that for rational functions we get the same equations as before.

Again we can do CAD on $\mathbb{R}^5$ and project down for the x variable to obtain the solution curves, which then are the edges of the Newtonian graph.

7 Acknowledgments

This work was supported in part by the National Science Foundation and in part by the United States Army Research Office through the Army Center of Excellence for Symbolic Methods in Algorithmic Mathematics (ACSyAM), Mathematical Sciences Institute of Cornell University under contract DAAL03-91-C-0027.

References

[1] Michael Ben-Or, Dexter Kozen and John Reif *The Complexity of Elementary Algebra and Geometry* Journal of Computer and System Sciences, Vol. 32 No. 2, April 1986

[2] G. E. Collins, *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition*, Lecture Notes in Computer Science, vol. 33, pp. 134-183, Springer-Verlag, Berlin Heidelberg New York 1975.

[3] Morris W. Hirsch and Stephen Smale *Differential Equations, Dynamical Systems, and Linear Algebra,* Academic Press, 1974

[4] Dexter Kozen, Chee-Keng Yap *Algebraic Cell Decomposition in NC,* Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS) 1985, pp. 515-521

[5] Morris Marden *The geometry of the Zeroes of a Polynomial in a Complex Variable,* American Mathematical Society, New York, 1966

[6] Stephen Smale *On the Efficiency of Algorithms of Analysis,* Bulletin of the American Mathematical Society, Vol. 13 No. 2, pp. 87-122, October 1985

[7] Michael Shub, David Tischler and Robert F. William *The Newtonian Graph of a Complex Polynomial,* SIAM J. Math. Anal., Vol. 19, No. 1, January 1988

[8] Alfred Tarski, *A Decision Method for Elementary Algebra and Geometry,* Univ. of Calif. Press Berkeley, 1948; 2nd ed. 1951.