Numerical Approximation of Coefficients of Belyĭ Maps

Edray H. Goins · Luis Melara · Alejandra Alvarado

Received: date / Accepted: date

Abstract In 1984, Alexander Grothendieck, inspired by a result of Gennadii Belyĭ from 1979, constructed a finite, connected planar bipartite graph via rational functions $\beta : \mathbb{P}^1(\mathbb{C}) \to \mathbb{P}^1(\mathbb{C})$ with critical values $\{0, 1, \infty\}$ by looking at the inverse image of the triangle formed by these three points. He called such graphs Dessins d’Enfants. Conversely, Riemann’s Existence Theorem implies that every finite, connected planar graph arises in this way.

The difficulty arises in explicitly constructing such a Belyĭ map $\beta$ from any given planar graph. We may form a valency list by considering the number of edges surrounding each vertex and each face; this forces algebraic conditions on the coefficients of the desired Belyĭ map. Hence the construction of a Belyĭ map can be reduced to the computation of roots of a system of nonlinear equations. In this paper, we
reformulate the problem of finding these roots into an unconstrained optimization problem. We implement
Newton’s method and a limited memory BFGS method. Convergence results for both methods are
presented; including a second convergence result using the Kantorovich Theorem for Newton’s method.
Numerical results are discussed and shown.

**Keywords** Newton’s method · Shabat Polynomials · Belyi maps · Dessin d’Enfants

**Mathematics Subject Classification (2000)** 11G32 · 65H10 · 49M15

1 Introduction

We are motivated by the following question. Given a loopless, connected, bipartite graph $\Gamma$ on a compact
connected Riemann surface $X$, when is $\Gamma$ the Dessin d’Enfant of a Belyi map $\beta : X \rightarrow \mathbb{P}^1(\mathbb{C})$?

Over the last several years, as computers become more powerful, there has been literature on methods
for finding equations of Belyi maps, consider the excellent survey [1]. A literature search found several
papers discussing algorithms, yet, as far as we know, no detailed description of an numerical algorithm,
issues arising from its implementation and/or numerical results are presented [2]. Our paper is concerned
in finding equations of a particular class of Belyi maps using Newton’s method and limited memory
BFGS (L-BFGS), both methods that have not yet been implemented.

In [3], published in 1996, the author states, “At present no simple methods for computation of
generalized Chebyshev polynomials corresponding to a planar tree and vice versa are known.” The paper
discusses code that computes the polynomials without participation of a human; however, a description of
the algorithm which “performs a recursive transformation of the initial polynomial...” is not shown. The
author mentions a paper in which the author found generalized Chebyshev polynomials, using Maple,
for all planar trees with at most eight edges, but requiring “creative participation of a human.”

2 Background

First, we introduce some basic definitions used in this report. A finite **graph** is a pair $\Gamma = (V, E)$ where
$V$ is the vertex set and $E$ the edge set. We will consider those graphs that are loopless, connected,
planar, and bipartite. In particular, \( V = B \cup W \) is the disjoint union of, say, black and white vertices, respectively.

These graphs may be associated to a rational function, which will be discussed later. Our goal is to “draw” these graphs on a particular Riemann surface. A Riemann surface is a surface-like configuration that covers the complex plane with several, and in general, infinitely many “sheets” \([4]\).

The extended complex plane \( \mathbb{C} \cup \{ \infty \} \simeq \mathbb{P}^1(\mathbb{C}) \) is equivalent to the unit sphere via stereographic projection

\[
S^2(\mathbb{R}) = \left\{ (u, v, w) \in \mathbb{R}^3 \mid u^2 + v^2 + w^2 = 1 \right\}.
\]

We call this the Riemann sphere; it will be the compact connected Riemann surface we will be working over. We will often abuse notation and say “\( X \) is a Riemann surface”.

### 3 BelyÏ’ Maps and Dessin d’Enfants

Let \( X \) be a compact, connected Riemann surface. It can be shown that \( X \) is an algebraic variety, that is, \( X \simeq \{ P \in \mathbb{P}^n(\mathbb{C}) \mid f_1(P) = f_2(P) = \cdots = f_m(P) = 0 \} \) in terms of a collection of homogeneous polynomials \( f_i \) over \( \mathbb{C} \) in \( n+1 \) variables. In 1979, BelyÏ’ published an article explaining how these concepts are related \([5]\).

**Theorem 3.1** Let \( X \) be a compact, connected Riemann surface.

- \( X \) is a smooth, irreducible, projective variety of dimension \( d = 1 \). In particular, \( X \) is an algebraic variety.
- If \( X \) can be defined by a polynomial equation whose coefficients are not transcendental, then there exists a rational function \( \phi : X \to \mathbb{P}^1(\mathbb{C}) \) which has at most three critical values.
- Conversely, if there exists rational functions \( \phi : X \to \mathbb{P}^1(\mathbb{C}) \) which has at most three critical values, then \( X \) can be defined by a polynomial equation where the coefficients are not transcendental.

Let \( X = \mathbb{P}^1(\mathbb{C}) \). We embed \( X \hookrightarrow \mathbb{R}^3 \). A **rational function** \( \beta : X \to \mathbb{P}^1(\mathbb{C}) \) is a map given by

\[
\beta(z) = p(z)/q(z)
\]
where \( p(z) \) and \( q(z) \) are relatively prime polynomials. The degree is defined as \( \deg(\beta) = \max \{ \deg(p), \deg(q) \} \).

The inverse image of each \( \omega = (\omega_1 : \omega_2) \in \mathbb{P}^1(\mathbb{C}) \) is the set
\[
\beta^{-1}(w) = \left\{ z \in X \mid p(z) - wq(z) = 0 \right\}.
\]

The Fundamental Theorem of Algebra implies \( |\beta^{-1}(w)| \leq \deg \beta \). A point \( \omega \in \mathbb{P}^1(\mathbb{C}) \) is said to be a critical value if \( |\beta^{-1}(w)| < \deg(\beta) \); a point \( z \in \mathbb{P}^1(\mathbb{C}) \) is said to be ramified if \( \omega = \beta(z) \) is a critical value. We say that a rational function \( \beta(z) = p(z)/q(z) \) is a Belyî map if it has at most three critical values, \( \{\omega^{(0)}, \omega^{(1)}, \omega^{(\infty)}\} \). By applying a Môbius transformation to the map, then replace \( \beta(z) \) with the new function
\[
\frac{\omega^{(1)} - \omega^{(\infty)}}{\omega^{(1)} - \omega^{(0)}} \frac{p(z) - \omega^{(0)}q(z)}{q(z)}
\]
we may assume that \( \beta(z) \) has critical values in the set \( \{(0 : 1), (1 : 1), (1 : 0)\} \). Instead, we write \( \{0, 1, \infty\} \).

Nearly a century after Klein published his ideas, Grothendieck wrote the proposal “Esquisse d’un Programme” outlining several new directions for his research. This outline discussed the drawings by Klein and his collaborators in a new light. Following Grothendieck’s ideas, we may associate a bipartite graph \( \Gamma \) to a Belyî map \( \beta : X \to \mathbb{P}^1(\mathbb{C}) \) as follows.

The “black” vertices are \( B = \beta^{-1}(\{0\}) = \left\{ z^{(i)} \in \mathbb{P}^1(\mathbb{C}) \mid p(z^{(i)}) = 0 \right\} \).

The “white” vertices are \( W = \beta^{-1}(\{1\}) = \left\{ z^{(j)} \in \mathbb{P}^1(\mathbb{C}) \mid p(z^{(j)}) = q(z^{(j)}) \right\} \).

The edges are \( E = \beta^{-1}(\{0, 1\}) = \left\{ z^{(j)} \in \mathbb{P}^1(\mathbb{C}) \mid 0 \leq \beta(z^{(j)}) \leq 1 \right\} \).

The midpoints of the faces are \( F = \beta^{-1}(\{\infty\}) = \left\{ z^{(k)} \in \mathbb{P}^1(\mathbb{C}) \mid q(z^{(k)}) = 0 \right\} \).

We call such a graph \( \Gamma = (B \cup W, E) \) a Dessin d’Enfant for the rational function \( \beta \).

**Example 3.1** Consider the map \( \beta : \mathbb{P}^1(\mathbb{C}) \to \mathbb{P}^1(\mathbb{C}) \) such that \( \beta(z) = z^n \) for some positive integer \( n \). Then \( \beta \) is a Belyî map since the critical values are \( \{0, \infty\} \). The “white” vertices are \( W = \{1, \zeta_n, \ldots, \zeta_n^{n-1}\} \), the \( n \)th roots of unity; the “black” vertex is \( B = \{0\} \) so the Dessin d’Enfant is the star graph \( K_{1,n} \). Note that there are \( n + 1 \) vertices, \( n \) edges and by Euler’s formula, one face.

On the other hand, given a star graph \( K_{1,n} \) whose vertices are “black”, we can make a barycentric subdivision of \( K_{1,n} \). In other words, each edge is subdivided into two edges which are connected to a new “white” vertex. These two graphs are homeomorphic and the Belyî map corresponding to this graph is \( \beta(z) = -4z^n(z^n - 1) \).
First, we discuss a class of examples in some detail. Many of the ideas we will discuss here can be found in section 2.2 of [6]. Our ultimate goal is to develop a numerical method for constructing $\beta(z)$.

We say that a connected planar graph is a tree if it has only one face. The Euler Characteristic shows that it must have $|B| = N + 1$ vertices for $N$ edges. Write $B = \{z^{(0)}, z^{(1)}, z^{(2)}, \ldots z^{(N)}\}$ as the vertices and $F = \{z^{(\infty)}\}$ as the midpoint of the face. Denoting $m_i$ as the number of edges incident to each vertex $z^{(i)}$, also called valencies, the Degree Sum Formula asserts that

$$m_0 + m_1 + m_2 + \cdots + m_N = 2N. \tag{1}$$

It is assumed that the valency list $\{m_0, m_1, \ldots, m_N\}$ is a given set of positive integers. Next, we describe how to associate $\beta(z)$ to this tree.

First let the set $B$ consist of the “black” vertices. Then subdivide this graph by choosing the set $W$ to be the “white” vertices at the “midpoints” of the edges. This implies that $n_j = 2$ for $j = 1, 2, \ldots, N$ is the valency of each “white” vertex while $t_\infty = 2N$ is the valency of the one face. Write the desired Belyi map as the ratio $\beta(z) = p(z)/q(z)$, where $B = \beta^{-1}(0)$, $W = \beta^{-1}(1)$, and $F = \beta^{-1}(\infty)$ is the midpoint of the face. We seek complex numbers $z^{(i)}$, $p_0$, $q_0$, and $r_j$ such that

$$p(z) = p_0 \left[ z - z^{(0)} \right]^{m_0} \left[ z - z^{(1)} \right]^{m_1} \left[ z - z^{(2)} \right]^{m_2} \cdots \left[ z - z^{(N)} \right]^{m_N}$$

$$p(z) - q(z) = - \left[ r_N z^N + r_{N-1} z^{N-1} + \cdots + r_1 z + \cdots + r_0 \right]^2 \tag{2}$$

$$q(z) = -q_0 \left[ z - z^{(\infty)} \right]^{2N}$$

Setting each equation in (2) equal to zero, gives a polynomial in $z$ of degree $2N$ which must vanish identically. Unfortunately, we have $2N+5$ variables in just $2N$ equations, our system is underdetermined.

Thus, we use a Möbius transformation to eliminate several variables via the following substitutions:

$$Z = \gamma(z) \quad \text{in terms of} \quad \gamma(z) = \frac{z^{(1)} - z^{(\infty)}}{z^{(1)} - z^{(0)}} \frac{z - z^{(0)}}{z - z^{(\infty)}}. \tag{3}$$

$$x_i = \gamma(z^{(i+1)})$$
and we find that $\gamma(\beta^{-1}(0)) = \{0, 1, \ldots, y_{N-1}\}$ and $\gamma(\beta^{-1}(\infty)) = \{\infty\}$. This results in the ratios:

$$p(z) = R_N^2 Z^{m_0} \left[ Z - 1 \right]^{m_1} \left[ Z - y_1 \right]^{m_2} \cdots \left[ Z - y_{N-1} \right]^{m_N} / \left[ Z - \gamma(\infty) \right]^{2N}$$

$$p(z) - q(z) = \left[ R_N Z^N + R_{N-1} Z^{N-1} + \cdots + R_1 Z + \cdots + R_0 \right]^2 / \left[ Z - \gamma(\infty) \right]^{2N}$$

$$q(z) = -R_0^2 / \left[ Z - \gamma(\infty) \right]^{2N}$$

for some complex numbers $R_j$. Denoting $y_{j+N} = R_j / R_N$, the above is summarized in the following proposition.

**Proposition 4.1 (Goins)** Let $\{m_0, m_1, \ldots, m_N\}$ denote the valencies of the vertices of a tree with $N + 1$ vertices. If we can find $2N - 1$ nontrivial complex numbers $y_1, y_2, \ldots, y_{2N-1}$ such that

$$Z^{m_0} \left[ Z - 1 \right]^{m_1} \left[ Z - y_1 \right]^{m_2} \cdots \left[ Z - y_{N-1} \right]^{m_N} + y_N^2 - \left[ Z^N + \sum_{j=0}^{N-1} y_{j+N} Z^j \right]^2 = 0 \quad (5)$$

identically as a polynomial in $Z$, then the tree is the Dessin d’Enfant of the Belyǐ map

$$\beta(z) = \frac{p(z)}{q(z)} = -\frac{1}{y_N^2} \cdot Z^{m_0} \left[ Z - 1 \right]^{m_1} \left[ Z - y_1 \right]^{m_2} \cdots \left[ Z - y_{N-1} \right]^{m_N} \quad (6)$$

This Belyǐ map is a polynomial of degree $2N$. This is called a **Shabat Polynomial** after the Russian mathematician Georgii Borisovich Shabat (1952 – ), although he himself preferred to call them **Generalized Chebyshev Polynomials**.

By comparing coefficients, there are $2N - 1$ equations in $2N - 1$ unknowns, so we expect to find finitely many solutions. Since each of the $2N - 1$ coefficients $y_i$ is a complex number, we can write $y_i = x_{2i-1} \pm I x_{2i}$ in terms of the $d = 4N - 2$ real numbers $x_j$. Define the $d$ integer coefficient polynomials $F_j : A^d(\mathbb{R}) \to A^1(\mathbb{R})$ implicitly via the relation

$$z^{m_0} \left[ z - 1 \right]^{m_1} \left[ z - (x_1 \pm I x_2) \right]^{m_2} \cdots \left[ z - (x_{2N-3} \pm I x_{2N-2}) \right]^{m_N}$$

$$+ \left[ (x_{2N-1} \pm I x_{2N}) \right]^2 - \left[ z^N + \sum_{j=0}^{N-1} (x_{2j+2n-1} \pm I x_{2j+2n}) z^j \right]^2$$

$$= \sum_{i=1}^{2N-1} F_{2i-1}(x_1, x_2, \ldots, x_d) \pm I F_{2i}(x_1, x_2, \ldots, x_d) z^i$$

We can put these together to focus on the real vector-valued function $\phi : A^d(\mathbb{R}) \to A^d(\mathbb{R})$ which sends $P = (x_1, x_2, \ldots, x_d)$ to $Q = (F_1(P), F_2(P), \ldots, F_d(P))$. We wish to find all of the finitely many
real vectors $P_\infty \in \mathbb{A}^d(\mathbb{R})$ such that $\phi(P_\infty) = (0, 0, \ldots, 0)$. The idea is to start with a real vector $P_0$ and then construct a sequence $\{P_0, P_1, \ldots, P_k, \ldots\} \subseteq \mathbb{A}^d(\mathbb{R})$ such that $P_\infty = \lim_{k \to \infty} P_k$ is a solution to $\phi(P_\infty) = (0, 0, \ldots, 0)$. Consider the following two examples.

**Example 4.1** Now suppose $N = 2$. Then $d = 4N - 2 = 6$, and we have the cases $\{m_0, m_1, m_2\} = \{2, 1, 1\}, \{1, 2, 1\}$ or $\{1, 1, 2\}$. It can be shown that $\phi : \mathbb{A}^6(\mathbb{R}) \to \mathbb{A}^6(\mathbb{R})$ sends

$$
P = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \mapsto Q = \begin{bmatrix} -2x_3^2 + 2x_4^2 \\ -4x_3x_4 \\ x_1 - 2x_3 - x_3^2 + x_4^2 \\ x_2 - 2x_4 - 2x_3x_4 \\ -1 - x_1 - 2x_3 \\ -x_2 - 2x_4 \end{bmatrix}
$$

for $\{m_0, m_1, m_2\} = \{2, 1, 1\}$;

$$
P = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \mapsto Q = \begin{bmatrix} -x_1 - 2x_3^2 + 2x_4^2 \\ -x_2 - 4x_3x_4 \\ 1 + 2x_1 - 2x_3 - x_3^2 + x_4^2 \\ 2x_2 - 2x_4 - 2x_3x_4 \\ -2 - x_1 - 2x_3 \\ -x_2 - 2x_4 \end{bmatrix}
$$

for $\{m_0, m_1, m_2\} = \{1, 2, 1\}$;

$$
P = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \mapsto Q = \begin{bmatrix} -x_1^2 + x_2^2 - 2x_3^2 + 2x_4^2 \\ -2x_1x_2 - 4x_3x_4 \\ 2x_1 + x_2^2 - x_3^2 - 2x_3 - x_3^2 + x_4^2 \\ 2x_2 + 2x_1x_2 - 2x_4 - 2x_3x_4 \\ -1 - 2x_1 - 2x_3 \\ -x_2 - 2x_4 \end{bmatrix}
$$

for $\{m_0, m_1, m_2\} = \{1, 1, 2\}$. 

7
We conclude that the solutions $P_\infty$ to $\phi(P_\infty) = (0, 0, 0, 0, 0)$ are

$$P_\infty = \begin{cases} \left(-1, 0, -\frac{1}{2}, 0, 0, 0\right) & \text{for } \{m_0, m_1, m_2\} = \{2, 1, 1\}; \\ (2, 0, \frac{1}{2}, 0, -2, 0) & \text{for } \{m_0, m_1, m_2\} = \{1, 2, 1\}; \\ \left(\frac{1}{2}, 0, \frac{1}{8}, 0, -1, 0\right) & \text{for } \{m_0, m_1, m_2\} = \{1, 1, 2\}; \end{cases}$$

and their corresponding Belyi maps are $\beta(z) = -4z^2(z-1)(z+1)$, $\beta(z) = -4z(z-1)^2(z-2)$, and $\beta(z) = -16(z-1)(z-1/2)^2$, respectively.

**5 Problem Formulation**

We seek the roots $P_\infty \in A^d(\mathbb{R})$ of the system of nonlinear equations

$$\phi(P) = 0$$

where $0 = (0, 0, \ldots, 0) \in A^d(\mathbb{R})$.

Finding analytical solutions to the system of equations (8) may not be possible through conventional means. The authors also considered symbolic mathematical software packages. However, solutions could not be obtained as the number of equations in (8) increased. An alternative approach is to approximate solutions to (8) computationally. Many computational root-finding methods exist: Newton’s method, L-BFGS, and other quasi-Newton methods, to name some. In this work, we will focus on two: Newton’s method and L-BFGS. First, we describe Newton’s method:

$$P_{k+1} = P_k - D\phi(P_k)^{-1}\phi(P_k), \quad k = 0, 1, 2, \ldots$$

where the initial iterate $P_0$ is given,
and

\[
D\phi(P) = \begin{bmatrix}
\frac{\partial F_1}{\partial x_1}(P) & \frac{\partial F_1}{\partial x_2}(P) & \cdots & \frac{\partial F_1}{\partial x_d}(P) \\
\frac{\partial F_2}{\partial x_1}(P) & \frac{\partial F_2}{\partial x_2}(P) & \cdots & \frac{\partial F_2}{\partial x_d}(P) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_d}{\partial x_1}(P) & \frac{\partial F_d}{\partial x_2}(P) & \cdots & \frac{\partial F_d}{\partial x_d}(P)
\end{bmatrix}.
\]

The \(d \times d\) matrix \(D\phi(P)\) is the Jacobian of \(\phi(P)\) and \(\text{rank}(D\phi(P)) = d\). The Jacobian matrix has full rank since there should be only finitely many solutions to the simultaneous polynomial equations, so that they cut out a zero dimensional (non-singular) variety. Such a solution exists because of the 1965 paper [7].

Newton’s method has local quadratic convergence and therefore, choosing an initial iterate \(P_0\) may be difficult. As a result, a direct implementation of (9) may generate a sequence \(P_k \to \infty\) as \(k \to \infty\). Further, \(D\phi(B) \neq D\phi(P)^T\). Our reformulation constructs a symmetric Jacobian. Thus, we write (8) as the following unconstrained optimization problem

\[
\min_{P \in A^d(\mathbb{R})} L(P) \quad \text{(10)}
\]

where the real quadratic polynomial \(L : A^d(\mathbb{R}) \to A^1(\mathbb{R})\) is given by

\[
L(P) = \frac{1}{2} \|\phi(P_c)\|_2^2 + D\phi(P_c)^T \phi(P) (P - P_c) + \frac{1}{2} (P - P_c)^T H(P_c)(P - P_c) \quad \text{(11)}
\]

for fixed \(P_c \in A^d(\mathbb{R})\), where the gradient of \(L(P)\) is

\[
DL(P) = D\phi(P_c)^T \phi(P) + H(P_c)(P - P_c) \quad \text{(12)}
\]

and the Hessian of \(L(P)\) is

\[
D^2L(P) = H(P_c) = D\phi(P_c)^T D\phi(P_c) = \left[ \sum_{k=1}^d \frac{\partial L_k}{\partial x_k}(P_c) \frac{\partial L_k}{\partial x_j}(P_c) \right] \quad \text{(13)}
\]

with \(D\phi(P_c)\) given in [10]. The matrix \(H(P_c)\) is symmetric positive definite (SPD) since \(D\phi(P_c)\) has full rank. Therefore, \(D^2L(P)\) is SPD and \(L(P)\) is convex. As a result, a critical point \(P_\infty\) of \(L(P)\) is a solution to [10]. A critical point \(P_\infty\) satisfying first-order necessary conditions of [10] satisfies second-order sufficiency conditions since \(L(P)\) is convex.
The first-order necessary conditions of (10) state that a critical point $P_{\infty}$ necessarily satisfies

$$DL(P) = 0.$$ 

where $0$ is the zero vector. In particular, note

$$DL(P) = 0$$

$$D\phi(P_c)^T \phi(P_c) + D\phi(P_c)^T D\phi(P_c) (P - P_c) = 0$$

$$D\phi(P_c)^T D\phi(P_c) (P - P_c) = -D\phi(P_c)^T \phi(P_c)$$

(14)

Thus, (14) is equivalent to

$$D\phi(P_c) (P - P_c) = -\phi(P_c)$$

with $D\phi(P_c)^T$ multiplied to both sides of the equation. Solving for $P$ yields $0$. As a result, finding the minimum of (10) is equivalent to finding roots of (8).

Continuing from (14) yields

$$D\phi(P_c)^T D\phi(P_c) (P - P_c) = -D\phi(P_c)^T \phi(P_c)$$

$$P = P_c - \left[ D\phi(P_c)^T D\phi(P_c) \right]^{-1} D\phi(P_c)^T \phi(P_c).$$

(15)

Equation (15) is the Newton iteration associated to (10) which we represent below

$$P_{k+1} = P_k - H(P_k)^{-1} D\phi(P_k)^T \phi(P_k) = P_k - D^2L(P_k)^{-1} DL(P_k),$$

(16)

for $k = 1, 2, \ldots$ with given $P_0$ and where $H(P)$ is given in (13).

6 Some Convergence Results

The Kantorovich theorem is a convergence result for Newton’s method. The Kantorovich theorem makes no assumptions about the existence of a solution $DL(P_{\infty}) = 0$. Instead, if

- $D^2L(P_0)$ is nonsingular,
- $D^2L$ is Lipschitz continuous in

$$N(P_0, r) = \{ P : \|P - P_0\|_2 \leq r \},$$

with $r > 0$, and
– the first step $H(P_0)^{-1} D\phi(P_0)^T \phi(P_0)$ is sufficiently small relative to the nonlinearity of $L$,

then there must be a unique root of $DL(P)$ in $N(P_0, r)$ [8].

Proposition 6.1 shows that $D^2L(P)$ is nonsingular by finding an upper bound for $\|D^2L(P)\|_2$.

**Proposition 6.1** If $D^2L(P)$ in (13) is nonsingular, then

$$\|D^2L(P)^{-1}\|_2 \leq \xi$$

for some $\xi > 0$.

**Proof** Since $D^2L(P) = H(P_0)$, then $D^2L(P)$ is constant with respect to $P$. Then, $D^2L(P)$ is nonsingular with singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_d\}$ satisfying

$$\sigma_1 > \sigma_2 > \cdots > \sigma_d > 0.$$ 

Thus, there exists a finite $\xi > 0$ such that

$$\|D^2L(P)\|_2 = \|H(P_0)^{-1}\|_2 = \frac{1}{\sigma_d} \leq \xi.$$ 

For discussion on the singular values of a matrix, see [9]. In the following Proposition 6.2 we establish the Lipschitz continuity of $D^2L$ in $N(P_0, r)$.

**Proposition 6.2** Let $D^2L(P)$ be given in (13). Then, $D^2L(P)$ is Lipschitz continuous in $N(P_0, r)$ with Lipschitz constant $\mu$.

**Proof** Let $P, \tilde{P} \in N(P_0, r)$. Then,

$$\|D^2L(P) - D^2L(\tilde{P})\|_2 = \|H(P_0) - H(\tilde{P})\|_2 < \mu$$

for some $\mu > 0$.

6.1 Convergence of Newton’s Method

Theorem 6.1 establishes convergence of Newton’s method [8]. This theorem establishes the quadratic convergence of Newton’s method.
Theorem 6.1 Let $DL : \mathbb{A}^d(\mathbb{R}) \to \mathbb{A}^d(\mathbb{R})$ be continuously differentiable in an open convex set $\Omega \subset \mathbb{A}^d(\mathbb{R})$. Assume there exists $P_\infty \in N(P_\infty, r)$ and $r, \zeta > 0$, such that $N(P_\infty, r) \subset D$, $DL(P_\infty) = 0$, $D^2L(P_\infty)^{-1}$ exists with $\|D^2L(P_\infty)^{-1}\|_2 \leq \zeta$, and $D^2L$ is Lipschitz continuous in $N(P_\infty, r)$ with Lipschitz constant $\delta$. Then there exist constants $\varepsilon > 0$ such that for all $P_0 \in N(P_\infty, r)$, the sequence $P_1, P_2, \ldots$ generated by (16) is well-defined, converges to $P_\infty$ an obeys

$$\|P_{k+1} - P_\infty\|_2 \leq \zeta \delta \|P_k - P_\infty\|_2^2, k = 0, 1, 2, \ldots$$

The proof of Theorem 6.1 follows similar steps to the one in [8]. The proof is completed by induction and uses Proposition 6.1 to establish

$$\|D^2L(P_0)^{-1}\| \leq \zeta.$$

We leave the proof of Theorem 6.1 to the reader; this proof can also be found in [8]. One can interpret Theorem 6.1 as stating that the radius of convergence of (16) is inversely proportional to the relative nonlinearity of $DL$ at $P_\infty$ [8].

The next section presents another convergence result for Newton’s method.

6.2 Kantorovich Theorem

Recall that $H(P_0)$ and $D\phi(P_0)$ are nonsingular and that $\phi(P)$ is continuous at $P_0$. Then, the first Newton step $H(P_0)^{-1} D\phi(P_0)^T \phi(P_0)$ can be shown to be bounded from above by $\eta > 0$ by using the Cauchy-Schwartz inequality. Using this fact, Propositions 6.1 and 6.2 we establish the assumptions of Theorem 6.2 resulting in the Kantorovich theorem presented below.

Theorem 6.2 (Kantorovich) Let $P_0 \in A^d(\mathbb{R})$, $DL : \mathbb{A}^d(\mathbb{R}) \to \mathbb{A}^d(\mathbb{R})$, and assume that $DL$ is continuously differentiable in $N(P_0, r)$. Assume for a vector norm and the induced operator norm that $D^2L$ is Lipschitz continuous in $N(P_0, r)$ with $D^2L(P_0)$ nonsingular, and that there exist constants $\xi, \eta \geq 0$ such that

$$\|D^2L(P_0)^{-1}\|_2 \leq \xi, \quad \|D^2L(P_0)^{-1} DL(P_0) L(P_0)\|_2 \leq \eta.$$

If $\sigma \xi \eta \leq \frac{1}{2}$ and

$$r \geq r_0 \equiv \frac{1 - \sqrt{1 - 2\mu \xi \eta}}{\mu \xi},$$

12
then the sequence \( \{P_k\} \) generated by (16) is well-defined and converges to \( P_\infty \), a unique zero of \( DL(P) \) in the closure of \( N(P_0, r_0) \).

Unlike Theorem 6.1, the Kantorovich theorem makes no assumptions about the existence of a solution to \( DL(P_\infty) = 0 \) or the nonsingularity of \( D^2L(P_\infty) \). The Kantorovich Theorem has been previously used in [11–15] to establish convergence results for Newton’s method. Proofs and further discussion of the theorem and Newton’s method can be further found in [8,11–14]. The steps of the proof of Theorem 6.1 are left to the reader and can also be found in [8].

### 7 Numerical Implementation

The numerical implementation used a globalization technique to ensure sufficient decrease of each update of Newton’s method and incorporated techniques to address the numerical singularity of \( D^2F(P) \). This section will discuss these issues.

#### 7.1 Line-Search

We implemented a line search Newton’s method to approximate a solution to (10) [8]. Considering (10), allows us to implement numerical optimization techniques that globalize the optimization problem [8]. These include, but are not limited to, a line search technique and a model-trust region approach [8]. In this work, we focus on the former.

Let the descent direction \( S_k \) be given by

\[
S_k = -D^2L(P_k) DL(P_k) = -H(P_k)^{-1} \left[ D\phi(P_k)^T \phi(P_k) \right].
\]

(17)

Given \( P_0 \), the \((k + 1)\)st iterate using a Newton line search is obtained by the iteration

\[
P_{k+1} = P_k + \alpha S_k, \quad k = 1, 2, \ldots,
\]

where \( 0 < \alpha \leq 1 \) is the line search parameter. At each iteration \( k \), the value of \( \alpha \) is chosen so that the sufficient decrease criteria

\[
L(P_k + \alpha S_k) < L(P_k) + 10^{-4} \alpha \phi(P_k)^T D\phi(P_k) S_k
\]

(18)

is met. If (18) is false, we reduce \( \alpha \) by half until (18) is satisfied [8].

13
7.2 Numerical Singularity of Hessian

The Hessian $D^2L(P_k) = H(P_k)$ is nonsingular at the $k$th iterate. However, for sufficiently small singular values, $D^2L(P)$ can be numerically singular [8,9]. To determine if $D^2L(P)$ is close to singular, we compute the condition number of $H(P_k)$,

$$\text{cond}(H(P_k)) = \frac{\sigma_1}{\sigma_d},$$

where $\sigma_1 > \sigma_d$ are the largest and smallest singular values of $D^2F$, respectively [8,9]. If $\epsilon = \text{machine epsilon}$, and

$$\text{cond}(H(P_k)) > \frac{1}{\sqrt{\epsilon}},$$

then set

$$D^2L(P_k) = H(P_k) + \gamma I$$

with $\gamma = \sqrt{n\epsilon \|D\phi(P_k)^T D\phi(P_k)\|_1}$, see [8].

8 Limited Memory BFGS

We implemented L-BFGS since our Hessian matrices $H(P)$ are too dense to be manipulated. L-BFGS is based on the BFGS updating formula discovered by Broyden, Fletcher, Goldfarb, and Shanno [8,10]. This formula can be derived using the model given in [10] where the inverse of the Hessian in [11], $H(P)^{-1}$, is approximated by the symmetric positive definite matrix $B$ [8,10]. The update formula takes into account curvature information from earlier iterations. However, unlike the BFGS method, the L-BFGS method uses curvature information from only the most recent updates. Thus the L-BFGS update is given by

$$P_{k+1} = P_k - \alpha B_k D\phi(P_k)^T \phi(P_k)$$

(19)

where $B_k$ is an approximation to $H(P_k)^{-1}$ and is updated by

$$B_{k+1} = V_k^T B_k V_k + \rho_k w_k w_k^T$$

(20)
with

\[ \rho_k = \frac{1}{w_k^T z_k}, \]
\[ V_k = I - \rho_k w_k^T z_k; \]
\[ w_k = P_{k+1} - P_k, \]
\[ z_k = DL(P_{k+1}) - DL(P_k), \]

and \( \alpha \) is the line search parameter. The sufficient decrease criteria in (18) was also used to obtain a value for \( \alpha \).

The initial approximation \( B_0 \) is a scalar multiple of the identity matrix. Repeated applications of (20) are used to approximate \( B_k \). Then, we compute \( B_k D\phi(P_k)^T \phi(P_k) \) to update (19).

L-BFGS converges slower than Newton’s method. In addition, the L-BFGS method was inefficient for highly ill-conditioned problems. As the number of nonlinear equations grew, so did the condition number. These weaknesses did arise in our numerical experiments. In the next section, we present numerical results obtained from implementing both algorithms.

9 Numerical Results

Newton’s method and L-BFGS were implemented in MATLAB. This section presents numerical results obtained from the implementation of both algorithms. Both numerical methods obtained the approximations close to each other. For convenience, we focus primarily on numerical results obtained using Newton’s method since those results were computed with less iterations and were more accurate than the L-BFGS results. In both instances, we solved a system of \( 4N - 2 \) nonlinear equations. All initial iterates were randomly generated from a Gaussian distribution with mean 0 and standard deviation 1. The maximum number of iterations was 4,000 for both methods. Given a tolerance \( TOL \), we used the following three different stopping criteria:

1. the norm of the nonlinear equations:

\[ \| \phi(P) \|_2 < TOL; \]
2. the norm of the gradient of the objective function $DL(P)$:

$$\|DL(P)\|_2 < TOL; \text{ and}$$

3. relative error of consecutive iterates:

$$\frac{\|P_{k+1} - P_k\|_2}{\|P_{k+1}\|_2} < TOL.$$  

Tables 1 and 2 summarize results for Newton’s method and L-BFGS method, respectively. A comparison of both tables shows that Newton’s method converged significantly faster than L-BFGS for a smaller tolerance TOL. In our implementation, we did not use the same initial iterate for both Newton’s Method and L-BFGS. Sometimes, the same initial iterate did not meet any of the stopping criteria for both methods, or the method converged faster using a different initial iterate. The initial iterates chosen for the results in Tables 1 and 2 were selected from multiple trial runs so that the number of iterations computed was as small as possible for each method. These results are only a sampling showing convergence. We do not rule out the possibility of finding initial iterates which converge in less iterations than those shown in both tables.

Table 1 Newton’s Method, TOL=$10^{-9}$

<table>
<thead>
<tr>
<th>Valency</th>
<th>Iterations</th>
<th>$|\phi(P)|$</th>
<th>$|DL(P)|$</th>
<th>$\frac{|P_{k+1} - P_k|}{|P_{k+1}|}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1, 1, 2}$</td>
<td>3</td>
<td>$4.12e-16$</td>
<td>$1.85e-15$</td>
<td>$0.49$</td>
</tr>
<tr>
<td>${1, 2, 1}$</td>
<td>6</td>
<td>$9.15e-10$</td>
<td>$5.45e-09$</td>
<td>$2.15e-05$</td>
</tr>
<tr>
<td>${2, 1, 1}$</td>
<td>6</td>
<td>$1.49e-13$</td>
<td>$2.52e-13$</td>
<td>$6.56e-07$</td>
</tr>
<tr>
<td>${2, 1, 2, 1}$</td>
<td>12</td>
<td>$6.51e-13$</td>
<td>$2.64e-12$</td>
<td>$1.04e-06$</td>
</tr>
<tr>
<td>${3, 2, 1, 1, 1}$</td>
<td>20</td>
<td>$1.37e-14$</td>
<td>$8.76e-14$</td>
<td>$3.51e-08$</td>
</tr>
</tbody>
</table>

Additionally, we include below some sample results of solutions for the following valencies: $\{2, 1, 2, 1\}$ and $\{3, 2, 1, 1, 1\}$. Without loss of generality, we chose numerical results obtained using Newton’s method and present them as complex numbers. For the valency list $\{2, 1, 2, 1\}$, the solution obtained from New-
Table 2 Limited Memory BFGS, TOL=10^{-6}

| Valency | Iterations | ||φ(P)|| | ||DL(P)|| | ||P_{k+1}−P_k|| |
|---------|------------|--------|--------|--------|----------------|
| {1,1,2} | 223        | 3.79e−07 | 8.31e−07 | 4.93e−06 |
| {1,2,1} | 262        | 1.59e−06 | 9.91e−07 | 1.72e−04 |
| {2,1,1} | 246        | 4.650e−07 | 7.62e−07 | 1.95e−06 |
| {2,1,2,1} | 227 | 4.63e−07 | 9.82e−07 | 5.71e−06 |
| {3,2,1,1,1} | 902 | 1.06e−04 | 8.39e−07 | 1.57e−06 |

ton’s method took 12 iterations to compute. The initial iterate is given by:

\[ x_0 = 0.2774574715780405154 + 0.0870459144723550599 i \]
\[ x_1 = 0.1735930514374279532 + 0.4535911533710875743 i \]
\[ x_2 = -0.6035180309882939298 − 0.7970239279958442058 i \]
\[ x_3 = 0.3606488974936966629 + 1.8850916773414958527 i \]
\[ x_4 = -0.3068218306528001649 − 1.0127239609363869466 i \]

The numerical solution is:

\[ x_0^* = 0.6666666666665781449 − 0.000000000000800665 i \]
\[ x_1^* = -0.333333333333341276794 − 0.000000000010265207 i \]
\[ x_2^* = 0.0740740740741482329 + 0.000000000000957750 i \]
\[ x_3^* = -0.0000000000005130163 − 0.0000000000006305043 i \]
\[ x_4^* = -0.9999999999995142774 + 0.0000000000005933269 i \]

For comparison, we also include the exact solution obtained using Mathematica:

\[ x_0^* = \frac{2}{3} \]
\[ x_1^* = -\frac{1}{3} \]
\[ x_2^* = \frac{2}{27} \]
\[ x_3^* = 0 \]
\[ x_4^* = -1. \]

The Belyi map is given by

\[ \beta(z) = 182.25(z − 1)(z − 0.666667)^2(z + 0.333333) \]  

(21)
Fig. 1  Path graph corresponding to valency \( \{2, 1, 2, 1\} \) with associated Bely˘ı map \([21]\).

and the associated graph generated by \([21]\) is given in Figure \(\square 1\).

Similarly, we include numerical results for the valency list \( \{3, 2, 1, 1, 1\} \). The solution obtained from Newton’s method took 20 iterations to compute.

The initial iterate is:

\[
x_0 = \begin{array}{c}
30.8299436246859137611 - 93.8247210627779253400i \\
\end{array}
\]

\[
x_1 = \begin{array}{c}
167.4215951779131046351 + 12.4988173845022885189i \\
\end{array}
\]

\[
x_2 = \begin{array}{c}
53.0101256550694586167 - 95.2068223401487330193i \\
\end{array}
\]

\[
x_3 = \begin{array}{c}
85.4042818220477499835 + 38.9145734517764125826i \\
\end{array}
\]

\[
x_4 = \begin{array}{c}
-115.6001084951704171999 + 3.974012698924376610i \\
\end{array}
\]

\[
x_5 = \begin{array}{c}
-45.0598603707587415101 + 10.9247941665664516364i \\
\end{array}
\]

\[
x_6 = \begin{array}{c}
-25.0552842446153825051 - 18.9901654648676405657i \\
\end{array}
\]
The numerical solution is:

\begin{align*}
    x_0 &= -0.3333333333333307058 - 0.4714045207910326774i \\
    x_1 &= -0.3333333333333348691 + 0.4714045207910326774i \\
    x_2 &= 1.33333333333199366 + 0.000000000000013371i \\
    x_3 &= 0.1666666666666651031 + 0.00000000000001160i \\
    x_4 &= 0.0000000000000000000 - 0.0000000000000000000i \\
    x_5 &= -0.0000000000000000000 - 0.0000000000000000000i \\
    x_6 &= -1.33333333333274862 - 0.0000000000000665i.
\end{align*}

For comparison, we also include the exact solution obtained using Mathematica:

\begin{align*}
    x_0^* &= -\frac{1}{3} + i\frac{\sqrt{3}}{3} \\
    x_1^* &= -\frac{1}{3} - i\frac{\sqrt{3}}{3} \\
    x_2^* &= \frac{4}{3} \\
    x_3^* &= \frac{1}{6} \\
    x_4^* &= 0 \\
    x_5^* &= 0 \\
    x_6^* &= -\frac{4}{3}.
\end{align*}

The Belyš map is

\begin{equation}
\beta(z) = -36(-1.33333 + z)(-1 + z)^2z^3((0.33333333333330706 \\
+ 0.4714045207910326774I + z)((0.33333333333334869 \\
- 0.4714045207910326774I + z),
\end{equation}

and the associated graph generated by (22) is given in Figure 2.

10 Conclusion

For both methods, computing a solution numerically was significantly faster than computing it via Mathematica. We used the Solve command in Mathematica to solve the equations. However, the MATLAB implementation of both Newton’s method and L-BFGS solved the equations faster. There were numerical
singularity issues arising as the number of coefficients \(4N - 2\) increased. The Hessian matrix \(D^2L(P_k)\) was ill-conditioned. This presented difficulties since at each iteration \(k\), we solve the following linear system

\[
D^2L(P_k)S_k = -DL(P_k)
\]

to obtain the Newton step \(S_k\). Perturbing the Hessian \(D^2L(P_k) = H(P_k) + \gamma I\) did not always help either. We also considered solving for \(S_k\) using various preconditioners but those did not yield acceptable results either.

We did not perturb the approximation to the Hessian matrix \(B_k\) in L-BFGS. However, L-BFGS does not perform good when if \(B_k\) is ill-conditioned [10]. Therefore, accurate numerical results were unable to be obtained for large systems of equations using both methods. Both Newton’s method and L-BFGS have local convergence properties. This was evident as not all initial iterates converged to a solution in 4,000 iterations.

Finally, there remains much work to be done in approximating coefficients of Belyi maps. In our work, we find one solution to \(\phi(P)\) at a time. Further, we do not know the total number of roots of \(\phi(P)\). These are some possible directions we would like to consider pursuing as future work.

**Acknowledgements** The authors would like to thank the REUF Program, SQuaREs Program, and EDGE.
References