Acta Univ. Sapientiae, Mathematica, **1**, 2 (2009) 169–182

Approximating poles of complex rational functions

Levente Lócsi
Eötvös Loránd University
1117 Budapest, Pázmány Péter sétány 1/C.
email: locsi@inf.elte.hu

Abstract. In this paper we investigate the application of the Nelder–Mead simplex method to approximate poles of complex rational functions. To our knowledge, there isn't any algorithm which is able to find the poles of a function when only the values on the unit circle are given. We will show that this method can accurately approximate 1, 2 or even 3 poles without any preliminary knowledge of their locations. The work presented here has implications in the study of ECG signals.

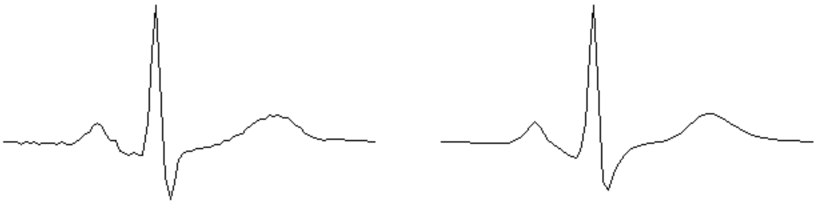
1 Introduction

The research presented in this article is motivated primarily by the fact that by combining a couple of simple complex rational functions and examining the values on the unit circle, the result can be very similar to an ECG signal (see Fig. 1). These functions can be applied for analysis, compression and denoising of ECG signals. Diagnostic applications may also be possible.

Rational functions play an important role in control theory. The Malmquist–Takenaka systems are often used to identify the transfer function of a system, see [1], [2], [9], and [10]. However automatic approximation of the poles of these functions proved not trivial when only the values on the unit circle are given and we have no preliminary knowledge about the locations of the poles.

A function, such as the one in Fig. 1, can be defined by its *poles* and the corresponding coefficients. The coefficients can be expressed by means of scalar

AMS 2000 subject classifications: 30E10, 33F05, 65E05, 65D15
Key words and phrases: Complex rational functions, Malmquist–Takenaka system, approximation of poles, Nelder–Mead simplex algorithm, ECG



(a) An ECG signal from [3]. (Apnea-ECG Database)

(b) An approximating function.

Figure 1: An ECG signal and a very similar function produced using complex rational functions.

products when using an orthonormal system, so the main problem is to find the poles generating an appropriate system.

In this paper we investigate the application of the Nelder–Mead simplex method to approximate poles of generated complex rational functions given by their values on the unit circle. The question of H^∞ approximation (see [1] and [2]) is also to be analyzed.

2 Mathematical background

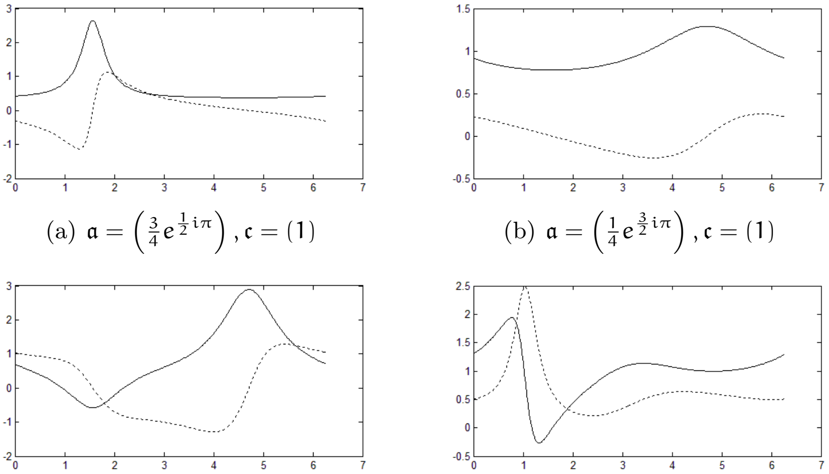
In this section we will introduce our functions of interest and recall some properties of related orthonormal systems. Then we give a summary of the Nelder–Mead simplex method, a commonly used nonlinear optimization algorithm.

2.1 Complex rational functions

Denote by $\mathbb{C}$ the set of complex numbers and let $\mathbb{D} := \{ z \in \mathbb{C} : |z| < 1 \}$ be the open unit disk, $\mathbb{T} := \{ z \in \mathbb{C} : |z| = 1 \}$ the unit circle and $\mathbb{D}^* := \mathbb{C} \setminus (\mathbb{D} \cup \mathbb{T})$. The natural numbers will be considered as the set $\mathbb{N} := \{ 1, 2, 3, \dots \}$.

The disk algebra, i.e. the set of functions analytic on $\mathbb{D}$ and continuous on $\mathbb{D} \cup \mathbb{T}$, will be denoted by $\mathcal{A}$. The scalar product on $\mathbb{T}$ is defined by:

$$\langle f, g \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(e^{it}) \overline{g}(e^{it}) \, dt.$$



(a) $a = \left(\frac{3}{4}e^{\frac{1}{2}i\pi}\right), c = (1)$

(b) $a = \left(\frac{1}{4}e^{\frac{3}{2}i\pi}\right), c = (1)$

(c) $a = \left(\frac{1}{2}e^{\frac{1}{2}i\pi}, \frac{1}{2}e^{\frac{3}{2}i\pi}\right), c = (-1, 2)$

(d) $a = \left(\frac{1}{4}e^{i\pi}, \frac{3}{4}e^{\frac{1}{3}i\pi}\right), c = (1, i)$

Figure 2: Examples of complex rational functions.

We shall examine functions generated by the collection of

$$\varphi_n(z) := \frac{1}{(1-\overline{a_n}z)^{m_n}} \qquad (z \in \mathbb{C};\ m \in \mathbb{N};\ n = 1, \dots, m),$$

where $a_n \in \mathbb{D}$ $(n = 1, \dots, m)$ and $m_n = \sum_{i \leq n, a_i = a_n} 1$ the multiplicity of the parameter a_n. We note that φ_n has a pole in $a_n^* = 1/\overline{a_n} \in \mathbb{D}^*$ and $\Phi := (\varphi_n \colon n = 1, \dots, m) \subset \mathcal{A}$.

Fig. 2 illustrates some rational functions of the form $f = \sum_{n=1}^{m} c_n \varphi_n$ with $a = (a_1, \dots, a_m)$, $c = (c_1, \dots, c_m)$ and $m = 1, 2$. [1]

By applying the Gram–Schmidt orthogonalization procedure to Φ, we obtain an orthonormal system $\Psi := (\psi_n \colon n = 1, \dots, m)$ on $\mathbb{T}$, the so-called Malmquist–Takenaka system (introduced in [7] and [11], see also [6]), which can be expressed by the Blaschke functions:

$$B_b(z) := \frac{z-b}{1-\overline{b}z} \qquad (b \in \mathbb{D};\ z \in \mathbb{C}).$$

[1] The values on $\mathbb{T}$ are shown, i.e. for a function f we plot $f(z) = f(e^{it})$, where $t \in [0, 2\pi]$. The solid line is the real part, the dashed line is the imaginary part of $f(z)$.

Namely

$$\psi_n(z) = \frac{\sqrt{1-|a_n|^2}}{1-\overline{a_n}z} \prod_{k=1}^{n-1} B_{a_k}(z) \qquad (z \in \mathbb{C};\ n = 1, \dots, m),$$

which suggests a convenient computation method for the values of ψ_n.

The orthonormality of the Malmquist–Takenaka functions is defined by

$$\langle \psi_k, \psi_l \rangle = \delta_{kl} \qquad (k, l = 1, \dots, m),$$

where δ_{kl} is the Kronecker symbol. Note that span Φ = span Ψ, i.e. the systems Φ and Ψ generate the same m-dimensional subspace.

Given a function $f \in \mathcal{A}$ we can compute $\mathcal{P}_\Psi f = \mathcal{P}_{a_1,\dots,a_m} f$, the orthogonal projection of f on the subspace span Ψ by the formula

$$\mathcal{P}_\Psi f = \sum_{n=1}^{m} \langle f, \psi_n \rangle \, \psi_n.$$

Let $\mathcal{E}_\Psi f = \mathcal{E}_{a_1,\dots,a_m} f$ denote the best approximation of f in $||.||_2$, in span Ψ:

$$\mathcal{E}_\Psi f := ||f - \mathcal{P}_\Psi f||_2 = \sqrt{\langle f - \mathcal{P}_f, f - \mathcal{P}_f \rangle}.$$

Our aim is to minimize $\mathcal{E}_\Psi f$ for a given function $f \in \mathcal{A}$ (f is given by its values on $\mathbb{T}$) and $m \in \mathbb{N}$ dimension by choosing the parameters $a_1, a_2, \dots, a_m$ of the Ψ (or Φ) system 'well'.

Naturally, in our computations we use the discrete approximation of the scalar product:

$$[f, g] := [f, g]_N = \frac{1}{2\pi N} \sum_{k=0}^{N-1} f(e^{2\pi i k/N}) \, \overline{g}(e^{2\pi i k/N})$$

for a sufficiently large N. Let us choose e.g. $N = 256$. Furthermore a function is given by its values on the set

$$\mathbb{T}_N := \left\{ z \in \mathbb{T} : z = e^{2\pi i k/N};\ k = 0, \dots, N-1 \right\}.$$

2.2 The Nelder–Mead algorithm

The method introduced by Nelder and Mead in [8] is for the minimalization of a function of n variables, which depends only on the comparison of function values at the $n+1$ vertices of a general simplex, followed by the replacement of the vertex with the highest value by another point. The simplex adapts itself to the local landscape and contracts on to the final minimum. The method has been shown to be effective and computationally compact. Though there are very few proofs concerning its convergence properties (see [4] and [5]), it is widely used in practice in natural sciences and engineering for function optimization.

The method is described as follows. Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an arbitrary function and $x_1, x_2, \dots, x_{n+1} \in \mathbb{R}^n$ the vertices of the current (nondegenerate) simplex in n-dimensions. Usually the vertices are defined by an x_s starting point and $w > 0$:

$$x_1 := x_s, \quad x_i := x_s + w \cdot e_{i-1} \qquad (i = 2, \dots, n+1),$$

where e_i is the ith element of the canonical basis in $\mathbb{R}^n$.

Let $y_i := f(x_i)$ $(i = 1, \dots, n+1)$ and define the indices h and l such that

$$y_h = \max\{\, y_i \,:\, i = 1, \dots, n+1\,\}, \quad y_l = \min\{\, y_i \,:\, i = 1, \dots, n+1\,\},$$

the *highest* and *lowest* values. Further define $\overline{x}$ the *centroid* of the points x_i with $i \neq h$.

At each stage in the process x_h is replaced by a new point; three operations are used: *reflection, expansion* and *contraction* with the following parameters: $\alpha = 1, \beta = \frac{1}{2}$ and $\gamma = 2$, since the natural ('standard') strategy given by these values proved to be the best, see [8]. These are defined as follows:

- The reflection of x_h is defined by the relation

  $$x_r := (1+\alpha)\overline{x} - \alpha x_h, \quad y_r := f(x_r).$$

  If $y_l \leq y_r < y_h$, then x_h is replaced by x_r and we start again with the new simplex.

- If $y_r < y_l$, i.e. the reflection has produced a new minimum, then we expand x_r to x_e by the relation:

  $$x_e := \gamma x_r + (1-\gamma)\overline{x}, \quad y_e := f(x_e).$$

  If $y_e < y_r$, we replace x_h by x_e and restart the process; but if $y_e \geq y_l$, then we have a failed expansion, and we replace x_h by x_r before restarting.

- If on reflecting x_h to x_r we find that $y_r \geq y_i$ for all $i \neq h$, i.e. that replacing x_h by x_r leaves y_r the maximum, then we define a new x_h to be either the old x_h or x_r, whichever has the lower y value (when $y_h = y_r$ then choose x_h), and form

  $$x_c := \beta x_h + (1-\beta)\overline{x}, \quad y_c := f(x_c).$$

  We then accept x_c for x_h and restart, unless $x_c > \min\{x_h, x_r\}$, i.e. the contracted point is worse than the better of x_h and x_r. For such a failed contraction we replace all x_i points by $\frac{1}{2}(x_i + x_l)$ and restart the process.[2]

We stop the iteration when the standard deviation is less then ε, a small preset value:

$$\left(\frac{1}{n}\sum_{i=1}^{n+1}(y_i - \overline{y})^2\right)^{\frac{1}{2}} < \varepsilon,$$

where

$$\overline{y} = \frac{1}{n+1}\sum_{i=1}^{n+1} y_i.$$

Let us choose e.g. $\varepsilon = 10^{-6}$.

The Nelder–Mead method is an effective and robust algorithm, but it often stops near local minima ignoring better global solutions. In these cases a reinitialization of the simplex at another starting point may prove helpful.

In Fig. 3 we illustrate the steps of a 2-dimensional simplex with starting point $x_s = (4, 6)$ and $w = 1$ optimizing the quadratic function:

$$f(x) = f(x', x'') = \frac{1}{16}(x' - 2)^2 + (x'' - 3)^2 \qquad (x', x'' \in \mathbb{R}).$$

The effects of reflection, expansion and contraction can be observed, as defined above. It is clear that in this simple case the simplex contracts on the minimum $x_{min} = (2, 3)$.[3]

[2] This operation is called a *shrink* and a shrinking parameter δ can also be defined. The standard choice is $\delta = \frac{1}{2}$.

[3] This algorithm is also known as the *amoeba method* for the similarity of the simplex's moves to the named unicellular creature.

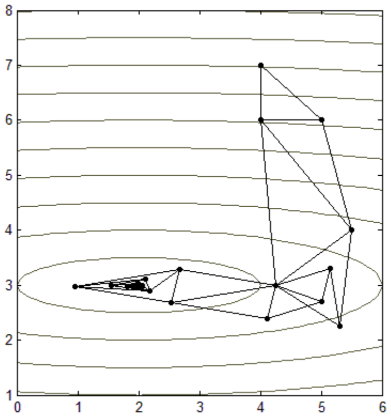


Figure 3: Moves of a Nelder–Mead simplex optimizing a quadratic function.

3 Methodology

In this section we explain how the Nelder–Mead method can be applied to find suitable parameters for the approximation. Furthermore, we describe our experiments and measurements.

Our goal is to minimize the function $\mathcal{E}_\Psi f = \mathcal{E}_{a_1,\dots,a_m} f$ introduced in Section 2.1 for a given $f \in \mathcal{A}$ function (f is given by its values on $\mathbb{T}$) and $m \in \mathbb{N}$ dimension by choosing the parameters $a_1, a_2, \dots, a_m$ of the Ψ system. For solving this minimization problem, we shall use the Nelder–Mead simplex algorithm described in Section 2.2.

The parameters of the Ψ system are to be chosen from $\mathbb{D}$. The simplex method requires vertices from $\mathbb{R}^n$. So in order to allow the simplex to move freely in $\mathbb{R}^n$ i.e. without any constraints to its steps, we set $n = 2m$ and use the map

$$\mathbb{R}^2 \ni (u, v) \quad \longmapsto \quad z = \frac{u}{\sqrt{1+u^2+v^2}} + \frac{v}{\sqrt{1+u^2+v^2}} i \in \mathbb{D}.$$

This map is a bijection between $\mathbb{R}^2$ and $\mathbb{D}$. Then a map from $\mathbb{R}^{2m}$ to $\mathbb{D}^m$ can be easily given by considering pairs of coordinates in $\mathbb{R}^{2m}$.

The traditional map used comes from the following idea. Imagine a half sphere on the complex unit disk $\mathbb{D}$ and then lay a plane $\mathbb{R}^2$ on the half sphere. Then the corresponding $z \in \mathbb{D}$ to an $(u, v) \in \mathbb{R}^2$ point is given by joining

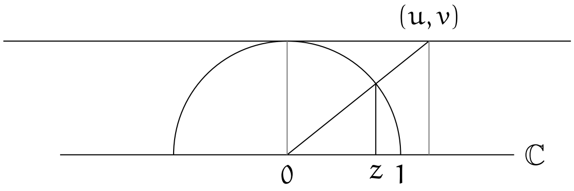


Figure 4: Mapping $(u, v) \in \mathbb{R}^2$ to $z \in \mathbb{D}$.

the complex zero with (u, v) by a straight line and projecting its intersection with the half sphere in $\mathbb{C}$ as seen in Fig. 4. The formula can be deduced from properties of the similar triangles on the figure.

So to find the $(a_1, \dots, a_m) \in \mathbb{D}^m$ parameter values minimizing $\mathcal{E}_\Psi f$, we use a simplex in $\mathbb{R}^{2m}$. Let the starting position x_s be the zero of $\mathbb{R}^{2m}$, and $w = 0.1$. The iteration stops when the standard deviation of the $\mathcal{E}_\Psi f$ values in the vertices of the simplex descend below $\varepsilon = 10^{-6}$.

For a given $m \in \mathbb{N}$ and function $f \in \mathcal{A}$ of the form

$$f(z) = \sum_{n=1}^{m} c_n \varphi_n(z) \qquad (z \in \mathbb{T};\ c_n \in \mathbb{C})$$

with parameters $a_1, \dots, a_m \in \mathbb{D}$ defining the functions φ_n $(n = 1, \dots, m)$ and the approximations $b_1, \dots, b_m \in \mathbb{D}$ of the parameters a_n define

$$\mathcal{D}f := \max\{\, |a_n - b_n| \,:\, n = 1, \dots, m \,\},$$

the error of the approximation of the poles. (For convenience, we sometimes refer to the a_n parameters of the rational system as *poles* although these are actually not poles of the functions in focus. The $1/\overline{a_n}$ values are.) Further define

$$\mathcal{H}f := \frac{\max\{\, |f(t) - (\mathcal{P}_{b_1,\dots,b_m} f)(t)| \,:\, t \in \mathbb{T} \,\}}{\max\{\, |f(t)| \,:\, t \in \mathbb{T} \,\}},$$

the relative error of the approximation in H^∞ norm. Naturally, in our computations we use the discrete approximation of $\mathcal{H}f$. And finally define

$$\mathcal{N}f \in \mathbb{N},$$

the number of calculations the simplex algorithm performs before terminating, where one calculation means evaluating $\mathcal{E}_{b_1,\dots,b_m} f$ for a given function and set of parameters.

In our first experiments we used 1024 functions with one pole a_1 randomly chosen from the uniform distribution on $\mathbb{D}$, forming $f(z) = c_1\varphi_1(z)$ with c_1 also randomly chosen from $\mathbb{D}$. We avoided extreme values of a_1 and c_1 too close to zero (less than 0.05), because these values would result in almost constant function values on $\mathbb{T}$. We also avoided a_1 values too close to $\mathbb{T}$ (greater than 0.95), because φ_1 can no longer be defined with its parameter in $\mathbb{T}$ and our discretization may prove insufficient to reflect the properties of these extreme functions. For each function f we applied the Nelder–Mead algorithm as described above to find its pole and measured the previously defined $\mathcal{D}f$, $\mathcal{H}f$ and $\mathcal{N}f$ values.

Then we generated another 1024 functions with two poles i.e. functions of the form $f(z) = c_1\varphi_1(z) + c_2\varphi_2(z)$ with a_1, a_2, c_1, c_2 chosen similarly to the previous case and measured the $\mathcal{D}f$, $\mathcal{H}f$ and $\mathcal{N}f$ values again.

This experiment has been repeated for another 1024 functions with three random poles and coefficients: $f(z) = \sum_{i=1}^{3} c_i\varphi_i(z)$.

Finally we investigated the iterated application of the simplex algorithm in the case $m = 3$. This means that if the result was not good enough (e.g. $\mathcal{D}f > 10^{-4}$), we reinitalized the simplex with $w = 0.1$ and x_s in the position reached in the previous iteration and started the optimization process again, at most 5 times.

4 Results

The statistics of our measurement results of the $\mathcal{D}f$, $\mathcal{H}f$ and $\mathcal{N}f$ values are summarized in Table 1. The histograms of these values are shown in Fig. 5, 6 and 7. Fig. 5 and 6 show the number of functions (out of 1024) with $\mathcal{D}f$ and $\mathcal{H}f$ approximation error values with an order of magnitude of 10^{-8}, 10^{-7}, etc. Fig. 7 shows the number of functions (out of 1024) with $\mathcal{N}f$ values in the intervals shown on the horizontal axis.

One can observe that in the case $m = 1$, i.e. the case of functions with one pole, the algorithm always gives a very good approximation of the pole. The order of the approximation error is better than 10^{-6} and so is the approximation error in the H^∞ norm. The algorithm is very effective and fast, it requires 90 calculations on average. We also found that the algorithm needs more steps when applied to a function with its pole closer to $\mathbb{T}$. In these cases the approximation is usually more accurate too.

In the case $m = 2$ (i.e. the case of functions with two poles) in most cases the poles can be approximated with precision at least of order 10^{-6}. The

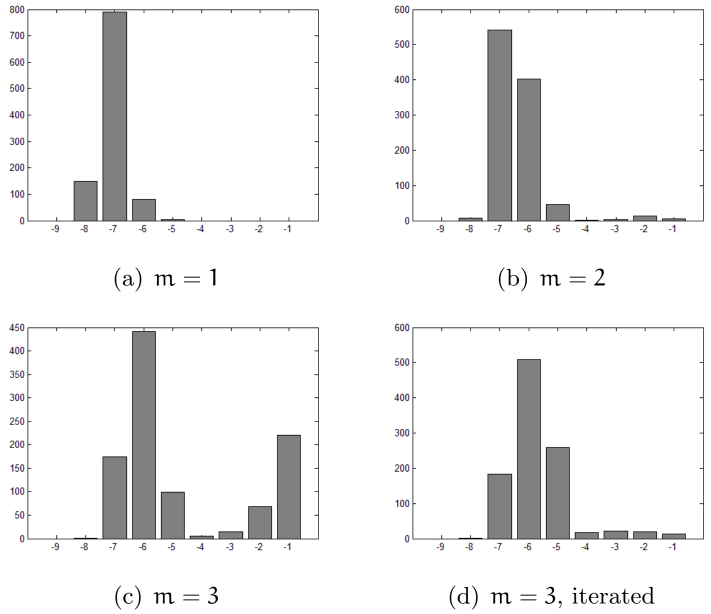
		min	max	avg	std. dev.
1 pole	$\mathcal{D}f$	$2.83 \cdot 10^{-8}$	$2.10 \cdot 10^{-5}$	$4.74 \cdot 10^{-7}$	$1.15 \cdot 10^{-6}$
	$\mathcal{H}f$	$7.20 \cdot 10^{-8}$	$2.19 \cdot 10^{-5}$	$6.52 \cdot 10^{-7}$	$1.34 \cdot 10^{-6}$
	$\mathcal{N}f$	56	116	89.92	8.30
2 poles	$\mathcal{D}f$	$5.74 \cdot 10^{-8}$	$4.27 \cdot 10^{-1}$	$1.85 \cdot 10^{-3}$	$1.95 \cdot 10^{-2}$
	$\mathcal{H}f$	$6.95 \cdot 10^{-8}$	$5.87 \cdot 10^{-3}$	$1.09 \cdot 10^{-5}$	$1.89 \cdot 10^{-4}$
	$\mathcal{N}f$	91	792	283.58	68.03
3 poles	$\mathcal{D}f$	$7.46 \cdot 10^{-8}$	$1.78 \cdot 10^{0}$	$1.13 \cdot 10^{-1}$	$2.71 \cdot 10^{-1}$
	$\mathcal{H}f$	$8.53 \cdot 10^{-8}$	$3.37 \cdot 10^{-1}$	$1.82 \cdot 10^{-3}$	$1.49 \cdot 10^{-2}$
	$\mathcal{N}f$	181	2006	712.64	272.56
3 poles iterated	$\mathcal{D}f$	$7.46 \cdot 10^{-8}$	$5.94 \cdot 10^{-1}$	$3.91 \cdot 10^{-3}$	$3.03 \cdot 10^{-2}$
	$\mathcal{H}f$	$8.53 \cdot 10^{-8}$	$2.57 \cdot 10^{-4}$	$1.91 \cdot 10^{-6}$	$1.24 \cdot 10^{-5}$
	$\mathcal{N}f$	430	2782	944.54	361.27

Table 1: The measured minimum, maximum, average and standard deviation values of $\mathcal{D}f$, $\mathcal{H}f$ and $\mathcal{N}f$ in the four investigated cases.

approximation in the H^∞ norm is also very good. The algorithm requires about 280 calculations on average. The cases when the $\mathcal{D}f$ value is in the order of 10^{-1} or 10^{-2}, are the ones when the two poles are very close to each other and there is a significant difference in the absolute values of the coefficients. In such cases the function could be almost as precisely approximated using functions with only one pole as using functions with two poles.

For functions with three poles ($m = 3$), there are lot more cases when $\mathcal{D}f$ is of the order 10^{-1}, even if the H^∞ error is small enough. We observed that in these cases the algorithm finds two poles with high precision, but the third one is far from the original. Then if we start again by initializing the simplex in the point reached (we iterate the application of the algorithm), the third pole is also find usually with an error less than 10^{-5} and the error of the H^∞ approximation also decreases. Naturally the computation cost rises with m and with the iterated application of the algorithm.

In the case of functions with even more poles, our few experiments show that this algorithm is not as powerful as in the cases detailed above (See also [4].) For instance, if the function is generated with 8 different poles, then the simplex method usually finds 4 of the poles with very small errors, but the others remain unknown.



(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

(d) $m = 3$, iterated

Figure 5: Number of functions (out of 1024) vs. order of $\mathcal{D}f$.

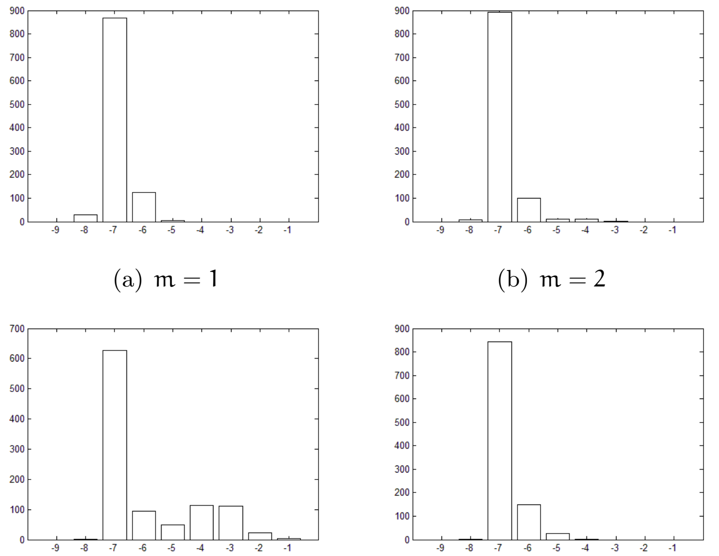
5 Conclusions

Our results show that the Nelder–Mead simplex algorithm can be applied effectively to solve the problem of approximating poles of complex rational functions with 1, 2 or even 3 poles, when the functions are given by their values on $\mathbb{T}$ and we have no preliminary knowledge about the location of the poles. We also get a satisfying approximation in H^∞ norm.

The results presented here have proven sufficient to perform promising calculations in the case of approximating ECG signals.

6 Further research

The main area of application of this research is the processing and analysis of ECG signals. The representation using complex rational functions may give an efficient way to compress and store these signals. We can gain a new method for denoising too, because of the smoothness of the functions applied. The potentials in diagnostics are also to be explored.



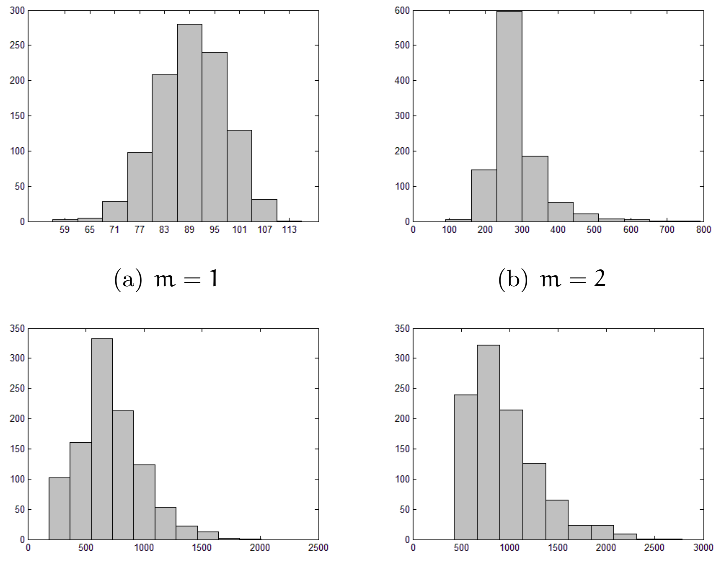
(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

(d) $m = 3$, iterated

Figure 6: Number of functions (out of 1024) vs. order of $\mathcal{H}f$.



(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

(d) $m = 3$, iterated

Figure 7: Histogram of $\mathcal{N}f$.

The effect of adding noise to the examined functions may also be investigated.

The direct use of $\mathbb{D}$ and hyperbolic coordinates instead of $\mathbb{R}^2$ in the implementation of the algorithm also seems to be an interesting field of research.

The design of new algorithms or possible improvement of the Nelder–Mead method for finding poles of functions with more singularities effectively is also to be studied.

7 Acknowledgements

I would like to express my gratitude to my supervisor Prof. Ferenc Schipp, whose stimulating suggestions and encouragement helped me all the time. I have to thank Dr. Ildikó László for her insistent help, interest and valuable hints.

Especially, I would like to thank Eszter Király, whose work has remarkably inspired this research.

References

[1] J. Bokor, F. Schipp, Approximate linear H^∞ identification in Laguerre and Kautz basis, *IFAC Automatica J.*, **34** (1998), 463–468.

[2] J. Bokor, F. Schipp, Z. Szabó, Identification of rational approximate models in H^∞ using generalized orthonormal basis, *IEEE Trans. Autom. Control*, **44** (1999), 153–158.

[3] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation*, **101** (2000), 215–220.

[4] L. Han, M. Neumann, Effect of dimensionality on the Nelder–Mead simplex method, *Optimization Methods and Software*, **21** (2006), 1–16.

[5] J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence properties of the Nelder–Mead algorithm in low dimensions. *AT&T Bell Laboratories Tech. Rep.*, Murray Hill, NJ, 1995.

[6] G. G. Lorentz, M. Golitschek, Y. Makovoz, *Constructive approximation*, Springer, Grundlehren Math. Wiss., 304, Berlin, 1996.

[7] F. Malmquist, Sur la détermination d'une classe fonctions analytiques par leurs dans un ensemble donné de points, *Compte Rendus Six. Cong. Math. Scand., Kopenhagen, Denmark*, 1925, 253–259.

[8] J. A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.*, **7** (1965), 308–313.

[9] M. Pap, F. Schipp, Malmquist–Takenaka systems and equilibrium conditions, *Mathematica Pannonica*, **12** (2001), 185–194.

[10] M. Pap, F. Schipp, Interpolation by rational functions, *Annales Univ. Sci. Budapest., Sect. Comp.*, **23** (2004), 223–237.

[11] S. Takenaka, On the orthogonal functions and a new formula of interpolations, *Japanese Journal of Mathematics*, **2** (1925), 129–145.

Received: May 25, 2009