Level Curves of Harmonic Polynomials

Closing the Circle: a Search for the Missing Arcs

Malinnikova, E. (Supervisor), Jarmund, A. H.

Norwegian University of Science and Technology, Trondheim, Norway

Abstract

It follows from the Mean Value Principle that the level set of a harmonic function cannot contain closed curves. Closed curves can be achieved by adding small arcs. Here it was found that the length of such arcs seems to decrease exponentially for the real part of complex polynomials as the degree increases.

1. Introduction

Harmonic functions are of special interest in both pure and applied mathematics. An interesting property of such (non-trivial) functions is that their level sets never form closed curves. However, by adding a small arc a closed curve can be achieved as illustrated in Figure 1. Such geometrical properties were recently investigated by Enciso and Peralta-Salas [1]. The exact nature of such nearly-closed curves is not known. This paper examines the behaviour of the level sets of the real part of complex polynomials.

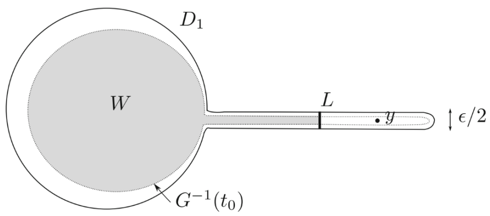


Figure 1: The level set of a harmonic function can approach a closed curve where just a small arc is missing. Figure from Enciso and Peralta-Salas [1].

First, some general theory of harmonic functions will be revisited. Then the problem to be studied is more precisely defined.

1.1. Harmonic functions

Definition 1.1. *A real-valued function u is said to be harmonic on the open, non-empty $\Omega \subseteq \mathbb{R}^n$ if u is two times continuously differentiable on Ω and u fulfils Laplace's equation, $\nabla^2 u = 0$, where ∇^2 is the well-known Laplacian operator [2].*

An important property of harmonic functions is the mean-value property, stating that if u is harmonic on the closure of a ball of radius r centred at $\mathbf{a}$, then $u(\mathbf{a})$ is equal to its own average over the open ball of radius r and centred at $\mathbf{a}$. This is summarised in Theorem 1.1.

Theorem 1.1. *If u is harmonic on the closure $\bar{B}(\mathbf{a}, r) = \{\mathbf{x} \in \mathbb{R}^n | |\mathbf{x} - \mathbf{a}| \leq r\}$ of a ball, B, then the value of $u(\mathbf{a})$ equals the average of u over the boundary of B, ∂B,*

$$u(a) = \int_{\partial B} \mathrm{d}\sigma(\delta)\, u(a + r\delta).$$

Proof. As we are only interested in the complex plane, we will restrict ourselves to $n = 2$. The general proof is stated by eg. Axler et al. [2].

Without loss of generality we consider the unit circle $B(0, 1)$. Choose $\eta = \log|\mathbf{x}| = \log\sqrt{x^2 + y^2}$. η diverges at the origin, so consider the exclusion of a small circle $B_\epsilon = B(0, \epsilon)$. Green's identity states that

$$0 = \int_{B \setminus B_\epsilon} [u\nabla^2\eta - \eta\nabla^2 u]\, \mathrm{d}V = \int_{\partial B} [uD_\mathbf{n}\eta - \eta D_\mathbf{n} u]\, \mathrm{d}S - \int_{\partial B_\epsilon} [uD_\mathbf{n}\eta - \eta D_\mathbf{n} u]\, \mathrm{d}S\,,$$

where

$$D_\mathbf{n}\eta = \hat{\mathbf{n}} \cdot \boldsymbol{\nabla}\eta = \frac{[x, y]}{\sqrt{x^2 + y^2}} \cdot \frac{[x, y]}{x^2 + y^2} = \begin{cases} 1, & [x, y] \in \partial B, \\ 1/\epsilon, & [x, y] \in \partial B_\epsilon, \end{cases}$$

$$\eta(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \partial B, \\ \log|\epsilon|, & \mathbf{x} \in \partial B_\epsilon. \end{cases}$$

From the integral above, it follows that

$$\int_{\partial B} [uD_\mathbf{n}\eta - \eta D_\mathbf{n} u]\, \mathrm{d}S = \int_{\partial B_\epsilon} [uD_\mathbf{n}\eta - \eta D_\mathbf{n} u]\, \mathrm{d}S$$

which simplifies to

$$\int_{\partial B} u\, \mathrm{d}S = \int_{\partial B_\epsilon} \left[\frac{u}{\epsilon} - \log|\epsilon| \cdot D_\mathbf{n} u\right] \mathrm{d}S\,.$$

In the limit $\epsilon \to 0$, ∂B_ϵ has a circumference of $2\pi\epsilon$ and is approaching the origin so that $u(\mathbf{x}) \to u(0)$, $\mathbf{x} \in \partial B_\epsilon$. The right hand side integrals become

$$\lim_{\epsilon \to 0} \int_{\partial B_\epsilon} \frac{u}{\epsilon}\, \mathrm{d}S = 2\pi u(0),$$

and

$$\lim_{\epsilon \to 0} \int_{\partial B_\epsilon} \log|\epsilon| \cdot D_\mathbf{n} u\, \mathrm{d}S = 0$$

as $D_\mathbf{n} u$ is bounded and $\lim_{\epsilon \to 0} \epsilon \log|\epsilon| = 0$. Hence,

$$u(0) = \int_{\partial B} u \frac{\mathrm{d}S}{2\pi}.$$

□

A result of this property, is that the level set, as defined in Definition 1.2, of a harmonic function cannot contain closed curves.

Definition 1.2. *A level set of a function f is the set $S = \{s | f(s) = c\}$ for some fixed value c.*

Theorem 1.2. *If $f : \mathbb{C} \to \mathbb{C}$ is an analytic function, then it satisfy the Cauchy-Riemann equations,*

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \qquad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}. \tag{1}$$

Proof. The proof is following the ideas of Kreyszig [3]. If $f : \mathbb{C} \to \mathbb{C}$ is an analytic function, then there exists a derivative $\frac{\mathrm{d}f}{\mathrm{d}z}$ at each point z, by definition

$$\frac{\mathrm{d}f}{\mathrm{d}z} = \lim_{\Delta z \to 0} \frac{f(z + \Delta z) - f(z)}{\Delta z}. \tag{2}$$

Write $z = x + iy$ and $\Delta z = \Delta x + i\Delta y$. Split f into its real and imaginary parts $f(z) = u(z) + iv(z)$. The limit in (2) can then be expressed as to terms, one for u and one for v,

$$\frac{\mathrm{d}f}{\mathrm{d}z} = \lim_{\Delta z \to 0} \frac{u(z + \Delta z) - u(z)}{\Delta z} + i \lim_{\Delta z \to 0} \frac{v(z + \Delta z) - v(z)}{\Delta z}. \tag{3}$$

Also, the limit can be evaluated in two ways, either by first setting $\Delta y = 0$ and then evaluating the limit $\Delta x \to 0$, or vice versa.

In the first case, $\Delta y = 0$ implies $\Delta z = \Delta x$. By writing u and v as functions of (x, y), the limits in (3) are recognised as partial derivatives with resect to x,

$$\frac{\mathrm{d}f}{\mathrm{d}z} = \lim_{\Delta x \to 0} \frac{u(x + \Delta x, y) - u(x, y)}{\Delta x} + i \lim_{\Delta x \to 0} \frac{v(x + \Delta x, y) - v(x, y)}{\Delta x} = \frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x}. \tag{4}$$

The second case where $\Delta x = 0$ so that $\Delta z = i\Delta y$ is analogue to (4), but with respect to y,

$$\frac{\mathrm{d}f}{\mathrm{d}z} = \lim_{\Delta y \to 0} \frac{u(x, y + \Delta y) - u(x, y)}{i\Delta y} + i \lim_{\Delta y \to 0} \frac{v(x, y + \Delta y) - v(x, y)}{i\Delta y} = -i\frac{\partial u}{\partial y} + \frac{\partial v}{\partial y}. \tag{5}$$

Equating (4) and (5),

$$\frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} - i\frac{\partial u}{\partial y},$$

yields two equations, one for the real part and one for the imaginary part,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \qquad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y},$$

which are the Cauchy-Riemann equations. □

The reverse of Theorem 1.2 can also be shown [3]. In this paper, the real part of complex polynomials is being studied. It is therefore useful to establish that such a function is harmonic, as in Corollary 1.2.1.

Corollary 1.2.1. *If $f : \mathbb{C} \to \mathbb{C}$ is an analytic function, then the real part $u = \text{Re}\{f\}$ is harmonic.*

Proof. By Theorem 1.2, $u = \text{Re}\{f\}$ satisfy the Cauchy-Riemann equations. These can be combined by differentiating the first equation with respect to x and substitute in the second equation,

$$\left.\begin{aligned} \tfrac{\partial u}{\partial x} &= \tfrac{\partial v}{\partial y} \\ \tfrac{\partial v}{\partial x} &= -\tfrac{\partial u}{\partial y} \end{aligned}\right\} \frac{\partial^2 u}{\partial x^2} = -\frac{\partial^2 u}{\partial y^2} \to \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \nabla^2 u = 0.$$

□

1.2. Problem

As mentioned earlier, the problem to be studied is how close to a closed curve the level set of a harmonic function can come, or how small an arc has to be added to make a closed curve. In this paper, the real part of complex polynomials, and its Cesàro summations, is investigated.

The real part $q_n(z)$ of a complex polynomial $P_n(z)$ of degree n is

$$q_n(z) = \text{Re}\{P_n(z)\} = \text{Re}\left\{\sum_{j=0}^{n} a_j z^j\right\},\ a_j \in \mathbb{R}, z \in \mathbb{C}. \tag{6}$$

Since polynomials are analytic functions, it follows from Corollary 1.2.1 that q_n is a harmonic function.

The effect of Cesàro summation is also investigated. Introduce therefore the notation

$$q_n^m = \frac{1}{n+1} \sum_{j=0}^{n} q_j^{m-1},\ m \geq 1 \tag{7}$$

and let q_n^0 be q_n defined in (6).

Two sets of coefficients were studied, namely

$$a_j = \begin{cases} 0, & j = 0 \\ 1, & j \geq 1 \end{cases} \tag{8}$$

and

$$a_j = \frac{j}{n},\ j \geq 0 \tag{9}$$

where n is the degree of the polynomial. The level sets were found for $q_n^m = c$.

By choosing the first set of coefficients and restricting the problem to the unit circle, $q_n(z)$ takes the form,

$$q_n(z) = \text{Re}\left\{\sum_{j=1}^{n} \cos j\theta + i \sin j\theta\right\} = \sum_{j=1}^{n} \cos j\theta = \frac{D_n(\theta) - 1}{2},$$

where θ is the complex argument of z and $D_n(\theta)$ the Dirichlet kernel,

$$D_n(\theta) = \sum_{k=-n}^{n} e^{ik\theta} = 1 + 2\sum_{k=1}^{n} \cos(k\theta).$$

Thus, there is a close relationship between the polynomial in question and the Dirichlet kernel, from which it follows that q_m^1 is similar to the Fejér kernel,

$$F_n(x) = \frac{1}{n+1}\sum_{k=0}^{n} D_k(x) = \frac{1}{n+1}\sum_{k=0}^{n}(1 + 2q_k).$$

However, this project will restrict itself to the study of q_n^m.

2. Method

The polynomials q_n^m were calculated for a grid $z_j = x_j + iy_j$ with x_j ranging from $x_{\min}$ to $x_{\max}$, and y_j from $y_{\min}$ to $y_{\max}$, with a distance $|x_{j+1} - x_j| = |y_{j+1} - y_j| = \mathrm{d}z$.

An algorithm was implemented in Matlab using the recursive relations

$$q_0^0 = a_0, \tag{10}$$

$$q_{n+1}^0 = \mathrm{Re}\{P_{n+1}\} = q_n^0 + \mathrm{Re}\{a_{n+1}z^{n+1}\} = q_n^0 + \mathrm{Re}\{a_{n+1}z\}\,\mathrm{Re}\{z^n\},\ n \geq 0, \tag{11}$$

and similarly

$$\begin{aligned} q_{n+1}^m &= \frac{1}{n+2}\sum_{j=0}^{n+1} q_j^{m-1} = \frac{1}{n+2}\Big(\sum_{j=0}^{n} q_j^{m-1} + q_{n+1}^{m-1}\Big) \\ &= \frac{n+1}{n+2}q_n^{m-1} + \frac{1}{n+2}q_{n+1}^{m-1},\ m \geq 1, n \geq 0. \end{aligned} \tag{12}$$

The code used is given in Appendix A.

Contour plots were made to visually examine the level sets. m and c were chosen to optimise visual appearance and kept fixed for increasing n. The gap size for the inner curve was found using Matlabs 'Data Cursor' and plotted against n. As the resulting data looked exponential, the data were fitted to an exponential model of the form $g = ae^{bn}$ where g is gap size for fixed $x = 1$, and a and b parameters to be found. Adjusted R^2s were calculated for the models to assess their goodness-of-fit.

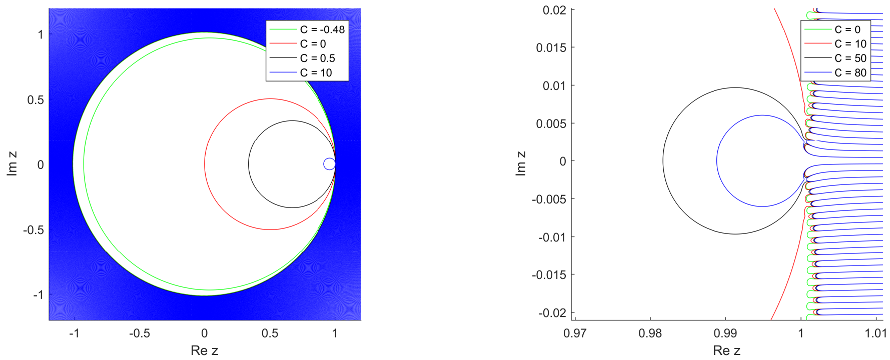
3. Results

3.1. First set of coefficients

Contour plots for polynomials with coefficients of the form

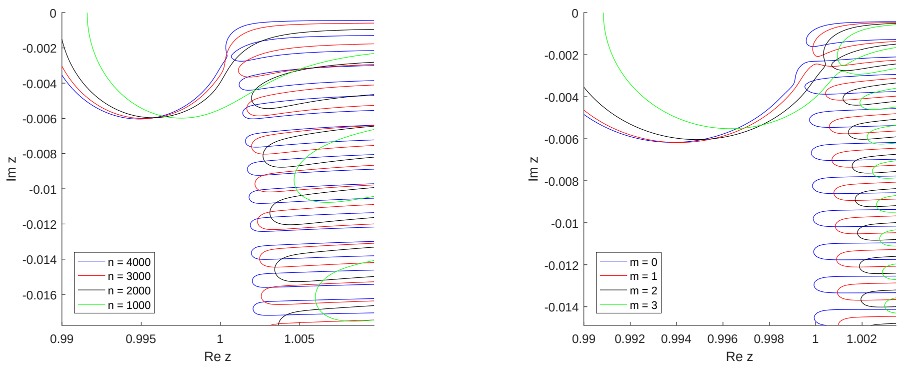
$$a_j = \begin{cases} 0, & j = 0 \\ 1, & j \geq 1 \end{cases} \tag{8 revisited}$$

are shown in Figures 2 and 3.



(a) Fixed degree of polynomial $n = 1000$. **(b)** Fixed degree of polynomial $n = 4000$.

Figure 2: Level sets for a polynomial of form (8) with fixed number of averaging $m = 2$. Note that different cs are used in the two plots.



(a) Fixed number of averages $m = 2$. **(b)** Fixed degree of polynomial $n = 4000$.

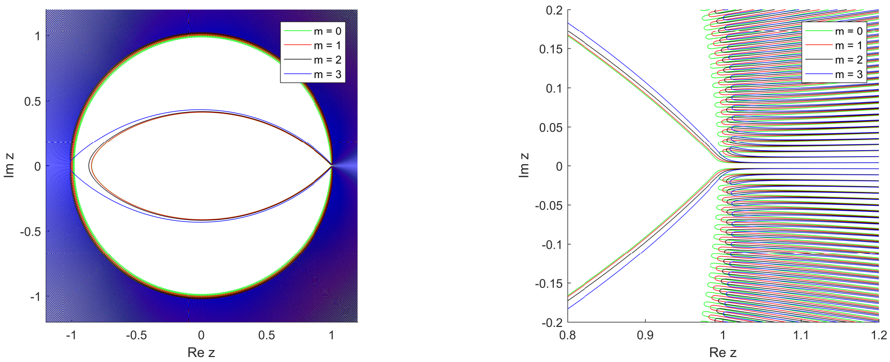
Figure 3: Level sets for a polynomial of form (8) with $c = 80$ with varying degree of polynomial n and number of averaging m.

3.2. Second set of coefficients

Contour plots for polynomials with coefficients of the form

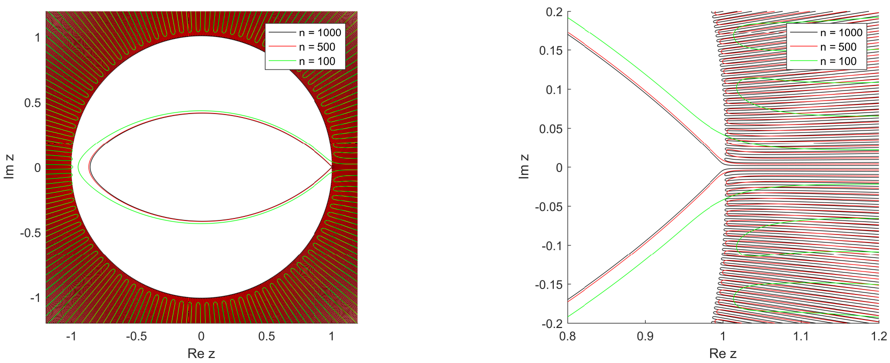
$$a_j = \frac{j}{n},\ j \geq 0 \qquad \text{(9 revisited)}$$

for $c = -0.000248$ are shown in Figure 4 for varying ms and in Figure 5 for varying n. c were chosen to maximise the inner curves and still keep them close to closed. Level sets for different cs are shown in Figure 6.



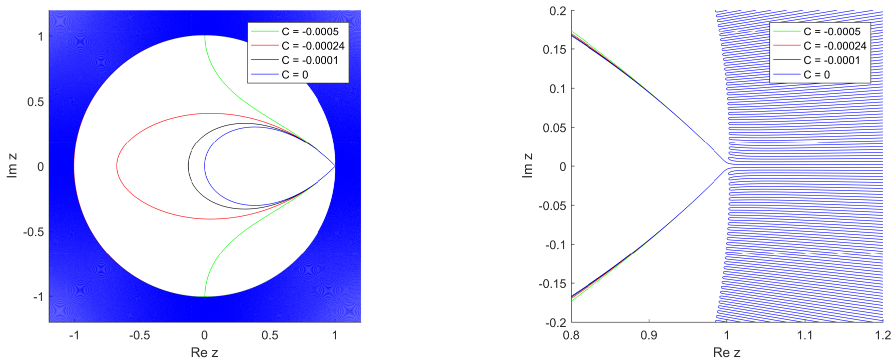
(a) (b)

Figure 4: Level sets for a polynomial of form (9) for $N = 1000$, $c = -0.000248$, and varying number of averaging m.



(a) (b)

Figure 5: Level sets for a polynomial of form (9) for $m = 2$, $c = -0.000248$, and varying degree of the polynomial n.

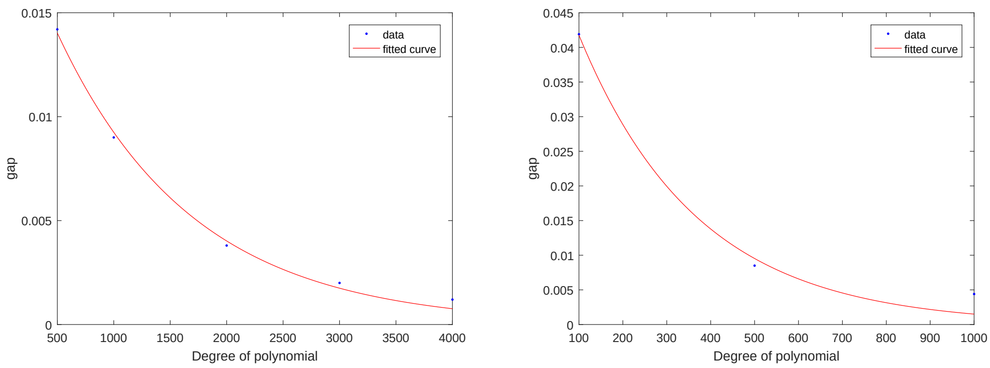


(a) **(b)**

Figure 6: Level sets for a polynomial of form (9) for $m = 2$, $n = 1000$ and varying c.

3.3. Exponential model

The inner gap distance was measured for the conditions shown in Figures 8 and 5b. The results were fitted to an exponential model as described in the Method section. For the first set, a was found to be 0.02128 (95% CI: [0.01856, 0.02399]) and b to be -0.0008331 (95% CI: [-0.0009774, -0.0006888]). For the second set, a was found to be 0.06044 (95% CI: [-0.03058, 0.1515]) and b to be -0.003694 (95% CI: [-0.01361, 0.00622]). Adjusted R^2 for the models were $R^2 = 0.9956$ and $R^2 = 0.9776$ respectively.



(a) For the first set of coefficients **(b)** For the second set of coefficients

Figure 7: Missing arc length (gap size g) shown for increasing polynomial degree n. The gap sizes were fitted to an exponential curve of the form $g = ae^{bn}$.

4. Discussion

The choice of coefficients clearly affects the shape of the level curves. Coefficients of the form of (8) creates quite circular level curves whereas the form of (9) creates more elliptic level curves. The two sets of coefficients also behave different as the parameters c, m and n are changed.

For the coefficients (8), circular inner curves are produced for $c > -0.5$ as illustrated in Figure 2a. However, if Figure 2b is studied closely, it seems that c has to approach 80 before this inner curve takes the form of Figure 1, that is, without interfering, intermediate oscillations. This is unfortunate as we wanted the inner curve to be as large as possible.

The latter effect is not observed for the coefficients of form (9). Even if the shape of the inner curve varies with c, and there is a lower limit to keep the curve close to closed as shown in Figure 6a, is this a macroscopic property, and the level sets overlap locally close to $(x, y) = (1, 0)$ as shown in Figure 6b.

The same behaviour is observed for varying number of averaging m. For the first set of coefficients, varying m affects both macroscopic shape and the shape close to $(x, y) = (1, 0)$. For the second set, only the macroscopic shape is affected. According to Figure 3b, it seems that $m = 2$ is optimal as both higher and lower ms yield interfering oscillations.

By choosing m and c as described above, the effect of increasing n can be studied. This is illustrated by Figures 3a and 8 for the first set of coefficients, and in Figure 5 for the second set. The increasing n does affect the gap size while the diameter of the curve seems constant. This shows that the effect of increasing n is in fact related to the length of the missing arc, and not the size of the curve as such.

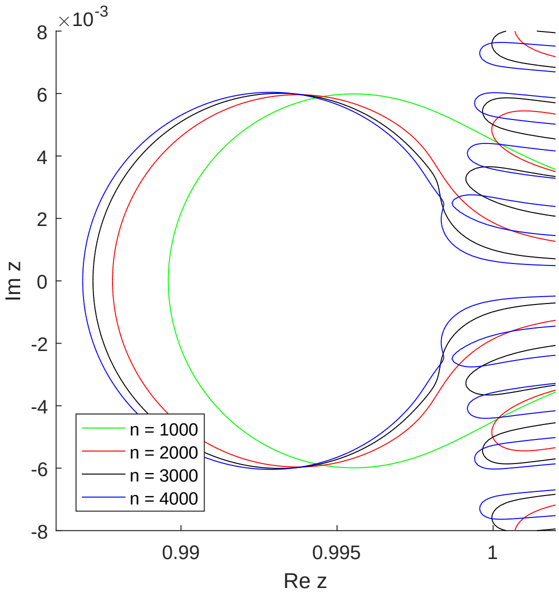


Figure 8: Level sets for coefficients given in (8) for increasing polynomial degree.

q_n^m was calculated for discrete values in a grid as described in the Method section. The contour plots produced by MATLAB seems, however, to be continuous. This suggests that some form of interpolation takes place. Also, due to the nature of numerical calculations, the level sets are likely to be chosen as $q_n^m(x, y) = c \pm \delta$, including all values for x and y that result in a function output in the interval $[c - \delta, c + \delta]$, where $\delta > 0$. The exact value of δ is not known, but is believed to be small.

Later projects could benefit of a more objective and less visual way of estimating values. Also, the curves studied in this project deviate vastly from the unit circle. More attempts should be made to decrease this deviation.

5. Conclusions

The level sets of the real part of complex polynomials can, due to the mean value property of harmonic functions, not make closed curves. The arc length needed to close a level curve decreases as the degree of the polynomial increases for at least two choices of coefficients.

The choice of coefficients influence the shape (circularity) of the level curve, but to a minor degree how it behaves. Averaging the polynomials m times seems also to alter the shape, and $m = 2$ minimises the needed arc length.

It was not possible to achieve a unit circle-like curve without interfering oscillations in the area of the missing arc. The choice of constant for the level set depends on coefficients.

The length of the missing arc decreased as the degree increased. The decline seemed to fit an exponential model with parameters dependent on the coefficient set.

References

[1] Alberto Enciso and Daniel Peralta-Salas. Some geometric conjectures in harmonic function theory. *Annali di Matematica Pura ed Applicata*, 192(1):49–59, Feb 2013. ISSN 1618-1891. doi: 10.1007/s10231-011-0211-4. URL https://doi.org/10.1007/s10231-011-0211-4.

[2] Sheldon Axler, Paul Bourdon, and Wade Ramey. *Harmonic Function Theory*. Springer-Verlag, 2. edition, 2001.

[3] Erwin Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 3. edition, 2011.

Appendix A. Source code

Appendix A.1. Memory usage

Note that even as the relation is iterative, the amount of calculated data needed increases quickly, the memory required is approximately

$$\frac{x_{\max} - x_{\min}}{\mathrm{d}z} \frac{y_{\max} - y_{\min}}{\mathrm{d}z} (M+1)(N+1) 64\,\text{bits}, \ \ M = 0, 1$$

an example is $[1, 1] \times [-1, -1]$ and $\mathrm{d}z = 10^{-3}$, yielding a memory requirement of $\sim 32(M+1)(N+1)$ MB (making a $N = 4000$ polynomial demanding ~ 256 GB free disk space). The implemented code overwrites old data for averaging so that only m and $m-1$ is kept, removing the memory dependence of M so that the requirement is

$$\frac{x_{\max} - x_{\min}}{\mathrm{d}z} \frac{y_{\max} - y_{\min}}{\mathrm{d}z} (N+1) 128\,\text{bits}, \ \ M \geq 1$$

Appendix A.2. Source code

```
clear all, clc
% This script calculates the level sets of polynomials and
    ↪ of averages of
% the polynomials using Casero summation. Level curves are
    ↪ saved as figures
% and a latex file (lat.txt) for showing them is created.
%
% The data is saved to a h5 file. Temporary data under /m(m
    ↪ #)n(n#) for m# =
% 0,1 and n# equal to the degree of the corresponding q. The
    ↪  data of
% interest, for q^m_n, is saved under /M(m#)N(n#) where m#
    ↪ equals m+1 and n# is the degree n.
% Written by Anders H. Jarmund, 2018 (contact: anderhja@stud
    ↪ .ntnu.no)


%% Configurations %%
% Define the domain, z = x + iy
dz = 0.0001;  % distance between points
x = [0.988 1.004];  % range of x: [min max]
y = [-0.008 0.008];  % range of y: [min max]

% Set degree of polynomial and number of times to average
N = 4000;  % degree of highest polynomial
N = N + 1; % to include the zeroth
M = 3;     % number of times to average
M = M + 1; % as MATLAB is not zero indexed
```

```
signN = [10 100 500 1000 2000 3000 4000]; % list of interesting degrees (for comparing), from 0 to N (not N+1)
signM = [0 1 2 3]; % list of interesting ms (for comparing), from 0-M (not M+1)

% Set the level sets, q^m_n = c
c = [-0.5 -0.004 0 0.5];

% File name
saveToFiles = 1; % Do calculations and make files? 1 for yes, 0 for no.
readFromFiles = 0; % Make plots, save figures and make latex file? 1 for yes, 0 for no.
makeLatexFile = 0;
filename = 'studforsk6'; % Filename of the kept results
filename = [filename '.h5']; %add file ending

% Define the coefficients
a{1} = [0 ones(1,N-1)]; % a_j = 0, 1
%a{1} = (0:N-1)/N; % a_j = j / N+1

%For each averaging, set the n+1 coefficients
a{2} = 1./((0:N-1)+1); %Using the same coefficients for all averages
for i = 3:M
    a{i} = a{2};
end

%% You do not have to (but are welcome to) change anything below this %%

% Make variables for the domain
x = x(1):dz:x(2);
y = y(2):-dz:y(1);
[xc, yc] = meshgrid(x, y);
z = xc + 1i*yc;

q = zeros(size(z)); %preallocate space for q

if(saveToFiles)
    while( exist(filename,'file') == 2 )
        deleteOlder = input('Do you want to overwrite older versions of the file (y for yes)? ','s');
        if strcmp(deleteOlder,'y')
            delete(filename) %deletes older versions
        else
```

```
        filename = input('\nEnter a new file name: ','s'
            ↪ );
        filename = [filename '.h5'];
    end
end

delete temp.h5

h = waitbar(0,'Please wait...'); % make a waitbar
steps = M * N;
step = 1;

m = 1; % this corresponds to the polynomial, i. e. q_0
zn = ones(size(z));
   for n = 0:N-1
       q = q + real(a{m}(n+1)*zn); % calculate q^0_n
       h5create('temp.h5', ['/m0' 'n' num2str(n)], size(
          ↪ q)) % need two files to contain all
          ↪ temporary data
       h5create('temp.h5', ['/m1' 'n' num2str(n)], size(
          ↪ q))
       h5write('temp.h5', ['/m' num2str(mod(m,2)) 'n'
          ↪ num2str(n)], q)
       waitbar(step / steps, h, ['Please wait... ('
          ↪ num2str(floor(step*100/steps)) '%)']) %
          ↪ update the waitbar
       step = step + 1;
       zn = zn.*z; % calculate z^(n+1) used in next
          ↪ iteration
       if(ismember(n, signN)) % if this n is of interest
          ↪ , save it for itself
           h5create(filename, ['/M1' 'N' num2str(n)],
              ↪ size(q))
           h5write(filename, ['/M1' 'N' num2str(n)], q)
       end
   end
clear zn;
for m = 2:M % calculate averages q^m
    q = zeros(size(z));
    for n = 0:N-1
        z = h5read('temp.h5',['/m' num2str(mod(m+1,2)) '
            ↪ n' num2str(n)]); %z is q^(m-1)
        q = n*q/(n+1) + a{m}(n+1)*z; % calculate q^m_n
        h5write('temp.h5', ['/m' num2str(mod(m,2)) 'n'
            ↪ num2str(n)], q)
        waitbar(step / steps, h, ['Please wait... ('
            ↪ num2str(floor(step*100/steps)) '%)']) %
```

```
                ↪ update waitbar
            step = step + 1;
            if(ismember(n,signN)) % if this n is of interest
                ↪ , save it for itself
                h5create(filename, ['/M' num2str(m) 'N'
                    ↪ num2str(n)], size(q))
                h5write(filename, ['/M' num2str(m) 'N'
                    ↪ num2str(n)], q)
            end
        end
    end
    delete temp.h5
end

if(readFromFiles)
    for C = c
        for n = signN
            for m = (signM+1)
                contour(x,y,real(h5read(filename,['/M'
                    ↪ num2str(m) 'N' num2str(n)])), [C C], 'b
                    ↪ ');
                axis square;
                xlabel('Re z')
                ylabel('Im z')
                fnam = ['m=' num2str(m-1) ',c=' num2str(C) '
                    ↪ ,n=' num2str(n)];
                save2pdf([fnam '.pdf'])
                close all
            end
        end
    end
end

if(makeLatexFile)
    fID = fopen('lat.txt','w');
    for C = c
        fprintf(fID,'\\begin{figure*}[t!]\n')
        fprintf(fID,'\\centering\n')
        for n = signN
            for m = (signM+1)
                fnam = ['m=' num2str(m-1) ',c=' num2str(C) '
                    ↪ ,n=' num2str(n)];
                fprintf(fID,'\t\\begin{subfigure}[b]{0.24\\
                    ↪ textwidth}\n')
                fprintf(fID,'\t\t\\centering\n')
                fprintf(fID,'\t\t\\includegraphics[width=\\
                    ↪ linewidth]{{fig/%s}\n',[fnam '}.pdf'])
```

```
                latnam = ['$m=' num2str(m-1) ', n=' num2str(
                   ↪ n) '$'];
                fprintf(fID,'\t\t\\caption{%s}\n',latnam)
                if(mod(m,4))
                    fprintf(fID,'\t\\end{subfigure}%%\n~\n')
                else
                    fprintf(fID,'\t\\end{subfigure}%%\n\n')
                end
            end
        end
        fprintf(fID,'\\caption{Caption place holder}\n')
        fprintf(fID,'\\end{figure*}\n\n\n')
    end
    fclose(fID);
end
```