Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Contour Plots of Analytic Functions

W. Gautschi[1] and J. Waldvogel

Research Report No. 96-15
October 1996

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

[1] Computer Science Department, Purdue University, West Lafayette, IN 47907, USA

Contour Plots of Analytic Functions

W. Gautschi[1] and J. Waldvogel

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 96-15 October 1996

Abstract

This is a tutorial on generating contour lines of an analytic function $f(z)$. The emphasis is on using mathematical software (MATLAB, to a lesser extent MAPLE) for implementing the algorithms, and efficient programs together with explanations are presented. Two different approaches are suggested: (1) generating level lines as contours of, e.g., constant modulus or constant phase of the function $f(z)$, (2) setting up and numerically integrating an appropriate differential equation for the contour under consideration. Both methods are demonstrated by means of the nth partial sum $f(z) = e_n(z)$ of the exponential series. The line of constant modulus satisfying $|e_n(z)| = 1$ has a practical significance: it delineates the region of absolute stability for an explicit Taylor integrator of order n.

[1] Computer Science Department, Purdue University, West Lafayette, IN 47907, USA. Supported, in part, by the National Science Foundation under grant DMS-9305430

1. Introduction

There are two easy ways in MATLAB to construct contour plots of analytic functions, i.e., lines of constant modulus and constant phase. One is to use the MATLAB `contour` command for functions of two variables, another to solve the differential equations satisfied by the contour lines. This is illustrated here for the function $f(z) = e_n(z)$, where

$$e_n(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^n}{n!} \tag{1}$$

is the nth partial sum of the exponential series. The lines of constant modulus 1 of e_n are of interest in the numerical solution of ordinary differential equations, where for $1 \le n \le 4$ they delineate regions of absolute stability of explicit one-step methods of order n (cf. [4, §9.3.2]). For $n \ge 1$ the 1-line of e_n delineates the region of absolute stability of the explicit Taylor method of order n.

2. Contour Plots by the `contour` Command

Let f be analytic and $f(z) = re^{i\varphi}$. We may consider the modulus r as a function of two variables x, y, where $z = x + iy$; similarly for the phase φ, $-\pi < \varphi \le \pi$. Hence, we can apply the MATLAB command `contour` to r and φ to obtain the lines of constant modulus and phase.

In the MATLAB program below, the set of all x- and y-values is collected (in true MATLAB spirit) in a matrix `a`, which is operated upon to compute the desired values of r and φ for $f = e_n$ as input matrices to the routine `contour`.

The program begins with the definitions of the mesh `h` and the number `nmax` of contour plots to be generated. The vector `bounds` contains common lower and upper bounds for the x- and y-coordinates applicable for all plots. The bounds used here have been chosen to accommodate contour plots of the first four exponential sums. Then the contour levels `vabs0` and `vang0` for the modulus and phase of $f(z)$ are defined. The last preparatory step is generating the vectors `x` and `y` containing the discrete x- and y-values to be used in the matrix `a` of grid points. In the loop over `n` the values `f` of e_n on the entire grid are generated by almost the same statements that would evaluate e_n at a single point, where `t` stands for an individual term of the series (1). The only difference is the statement `t=t.*a/n`, in which the operation symbol `.*` invokes the element-by-element product of the matrices `t` and `a`. The last line of the program (here turned off by the comment sign %) generates the encapsulated postscript file `fign.eps` of `figure(n)`, ready to be printed or incorporated into a text file.

```
% Contour plots of the first nmax exponential sums (Figure 1)
%
h=1/64; nmax=4; bounds=[-3.25 .75 -3.375 3.375];
vabs0=[0:.1:1]; vang0=[-.875:.125:1]*pi;
x=bounds(1):h:bounds(2); y=bounds(3):h:bounds(4);
a=ones(size(y'))*x+i*y'*ones(size(x));
% Next line: a shorter way of generating a (more memory!)
% [xx,yy]=meshgrid(x,y); a=xx+i*yy;
```

```
t=ones(size(a)); f=t;
for n=1:nmax
  if n<=2, vabs=vabs0; vang=vang0;
    elseif n==3, vabs=[vabs0 .47140452]; vang=vang0;
    else vabs=[vabs0 .58882534];
         vang=[vang0 1.48185376 -1.48185376]; end;
  t=t.*a/n; f=f+t;
  figure(n); clf; hold on;
  axis(bounds); axis image;
  contour(x,y,abs(f),vabs);
  contour(x,y,angle(f),vang);
end;
% figure(n); print -deps fign;
```

The results for $n = 1 : 1 : 4$[1] are shown in the plots below. Clearly visible are the n zeros of e_n from which emanate the lines of constant phase. Near these zeros, the lines of constant modulus become circle-like with radii tending to 0 as the zeros are approached. The contour lines are for $r = .1 : .1 : 1$ and $\varphi = -\frac{7}{8}\pi : \frac{1}{8}\pi : \pi$.

At points z_0 where $e'_n(z_0) = e_{n-1}(z_0) = 0$, $n \geq 2$, two lines of constant modulus intersect (cf. §3.1 below). The respective r-values are $r = |e_n(z_0)|$, or $r = |z_0|^n/n!$, since

$$e_n(z) = e_{n-1}(z) + \frac{z^n}{n!}. \tag{2}$$

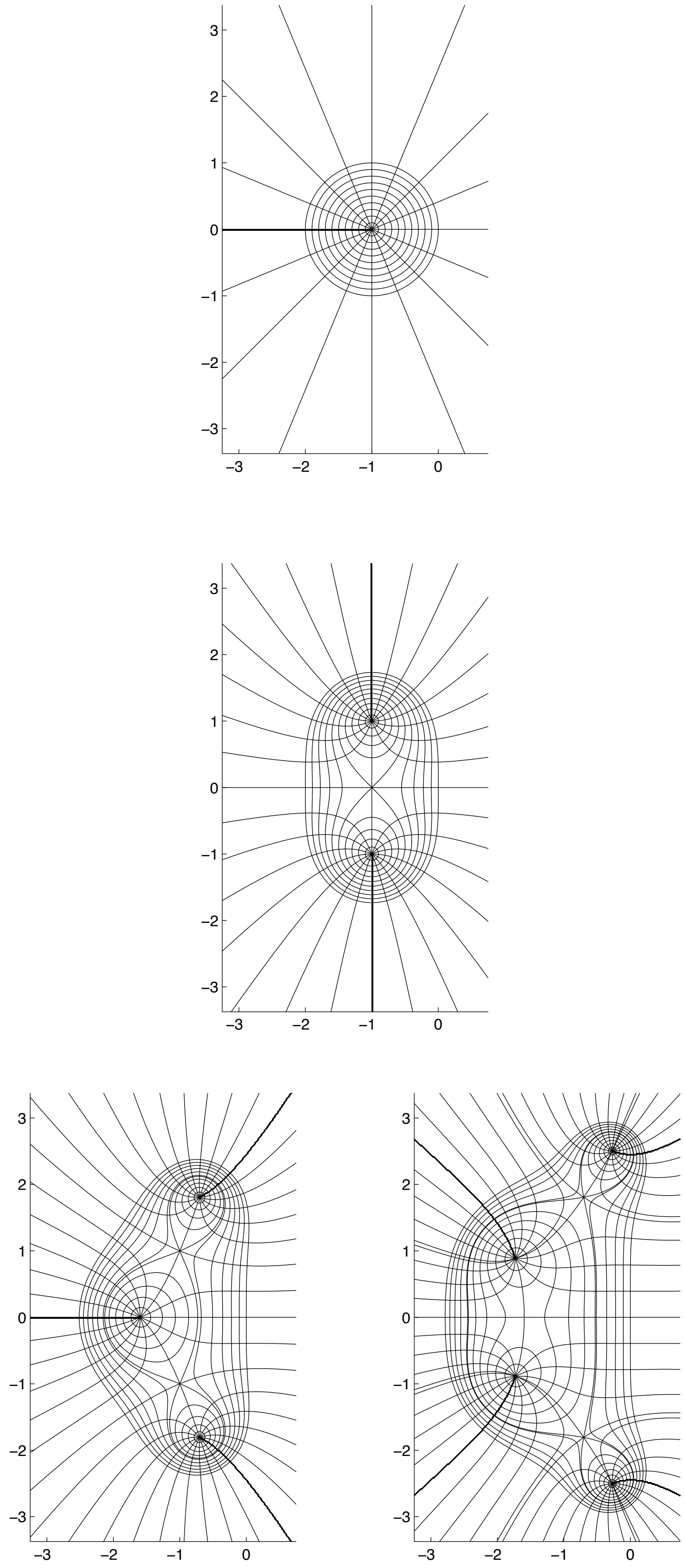
These critical lines are also included in the plots (see the `if` statement of the program). When $n = 2$, they go through $z_0 = -1$, where $r = \frac{1}{2}$, while for $n = 3$ and $n = 4$, one has to 8 decimal digits: $z_0 = -1 \pm i$, $r = \sqrt{2}/3 = .47140452$ and $z_0 = -.70196418 \pm 1.80733949i$, $r = (1.93887332)^4/24 = .58882535$, respectively.

What's good for the r-lines is good for the φ-lines! The singular points for them are also the zeros z_0 of e'_n (cf. §3.2), to which there correspond φ-values defined by $e_n(z_0)/|e_n(z_0)| = (z_0/|z_0|)^n = e^{i\varphi}$, i.e., $\varphi = n \arg z_0$. Thus, for $n = 2$, we have $\varphi = 0 \pmod{2\pi}$, whereas for $n = 3$ we get $\varphi = \pm\frac{\pi}{4}$ corresponding to $z_0 = -1 \pm i$, respectively. All three of these φ-values are included among the values already listed above. For $n = 4$, the two values of z_0 shown in the previous paragraph yield $\varphi = 4 \arg z_0 = \pm 1.48185376 \pmod{2\pi}$. These critical φ-lines are also shown in the plots in Figure 1.

The figure was generated by means of the step size `h=1/64` in order to obtain a good resolution, even for the "branch cuts“ corresponding to `|angle(f)|=pi`. The choice `h=1/32` is a good compromise, whereas `h=1/16` is very fast while still producing satisfactory plots.

[1] This MATLAB notation stands for $n = 1, 2, 3, 4$.

Figure 1. *Contour Plots of the First* 4 *Exponential Sums*



3. Differential Equations

For an analytic function f, let

$$w = f(z), \quad w = re^{i\varphi}, \quad z = x + iy. \tag{3}$$

3.1. *Contour lines* $r =$ const. To describe the lines $r =$ const, it is natural to take φ as independent variable. Differentiating

$$f(z(\varphi)) = re^{i\varphi}, \quad r = \text{const}, \tag{4}$$

with respect to φ then gives $f'(z)\frac{dz}{d\varphi} = ire^{i\varphi} = if(z)$, that is,

$$\frac{dz}{d\varphi} = i\, q(z)\ , \qquad \text{where} \qquad q(z) = \frac{f(z)}{f'(z)}. \tag{5}$$

With s the arc length, one has

$$\frac{ds}{d\varphi} = \sqrt{\left(\frac{dx}{d\varphi}\right)^2 + \left(\frac{dy}{d\varphi}\right)^2} = \left|\frac{dz}{d\varphi}\right| = |q(z)|,$$

so that

$$\frac{dz}{ds} = \frac{dz}{d\varphi}\frac{d\varphi}{ds} = i\ \frac{q(z)}{|q(z)|}. \tag{6}$$

Written as a system of differential equations, this is

$$\begin{aligned} \frac{dx}{ds} &= -\mathrm{Im}\left\{\frac{q(z)}{|q(z)|}\right\}, \\ \frac{dy}{ds} &= \ \ \mathrm{Re}\left\{\frac{q(z)}{|q(z)|}\right\}, \end{aligned} \qquad z = x + iy. \tag{7}$$

If we are interested in a contour line crossing the real axis, we must find an initial point $x(0) = x_r$, $y(0) = 0$ for (7) with real x_r such that $f(x_r) = r$ (assuming $f(x)$ real for real x). In the case $f(x) = e_n(x)$, this is easy if $r \geq 1$, since $e_n(0) = 1$ and $e_n(x)$ monotonically increases for $x \geq 0$. There is thus a unique $x_r \geq 0$ such that $e_n(x_r) = r$. If $0 < r < 1$, this is still possible when n is odd. Then, $e'_n(x) = e_{n-1}(x) > 0$, since all zeros of e_m, when m is even, are known to be complex [3] (cf. also [1]). Thus, e_n monotonically increases from $-\infty$ to $+\infty$ as x increases from $-\infty$ to $+\infty$, and there is a unique $x_r < 0$ such that $e_n(x_r) = r$. When n is even, we have $e_n(x) > 0$ for all real x, and $e'_n = e_{n-1}$ vanishes at exactly one point $x_0 < 0$, where e_n has a minimum (cf. [2]). Owing to (2) and $e_{n-1}(x_0) = 0$, we have $e_n(x_0) = x_0^n/n!$, and there is a solution $x_r < 0$ of $e_n(x_r) = r$ if and only if $r \geq x_0^n/n!$. For smaller positive values of r, one must find a complex initial point $x(0)$, $y(0) > 0$ near one of the complex zeros of e_n.

The point z_0 where $f'(z_0) = 0$ is a singular point of (7), a point where two r-lines intersect at a right angle. This requires special care to get the integration of (7) started in all four directions. The initial point, of course, is z_0, that is, $x(0) = \mathrm{Re}\, z_0$, $y(0) = \mathrm{Im}\, z_0$. What needs some analysis is the value of the right-hand side of (7) at z_0. Let $h(z) = (z - z_0)\, q(z)$; then h is smooth near z_0 and has the Taylor expansion

$$h(z) = \frac{f_0 + \frac{1}{2}(z - z_0)^2 f''_0 + \cdots}{f''_0 + \frac{1}{2}(z - z_0) f'''_0 + \cdots}, \quad h(z_0) = \frac{f_0}{f''_0},$$

where $f_0 = f(z_0)$, etc. (we assume $f_0 \neq 0$ and $f_0'' \neq 0$). Letting

$$\frac{z - z_0}{|z - z_0|} = e^{i\theta}, \quad -\frac{1}{2}\pi < \theta \leq \frac{1}{2}\pi, \quad h(z_0)/|h(z_0)| = e^{i\omega}, \quad -\pi < \omega \leq \pi,$$

(being mindful that to each θ there is a $\theta + \pi$ corresponding to the backward continuation of the line), we then have

$$\frac{q(z)}{|q(z)|} = \frac{|z - z_0|}{z - z_0}\,\frac{h(z)}{|h(z)|} \to e^{i(\omega - \theta_0)} \quad \text{as } z \to z_0,$$

where $\theta_0 = \lim_{z \to z_0} \theta$. It remains to determine θ_0.

Along an r-line through z_0, we have

$$\begin{aligned} r^2 = |f(z)|^2 &= |f_0 + \tfrac{1}{2}(z - z_0)^2 f_0'' + \cdots\ |^2 \\ &= |f_0|^2 + \mathrm{Re}[(z - z_0)^2 f_0''\overline{f_0}] + O(|z - z_0|^3). \end{aligned}$$

Since $|f_0|^2 = r^2$, this gives

$$\mathrm{Re}\left\{\left(\frac{z - z_0}{|z - z_0|}\right)^2 f_0''\overline{f_0}\right\} = O(|z - z_0|),$$

hence, as $z \to z_0$,

$$\mathrm{Re}\,(e^{2i\theta_0} f_0''\overline{f_0}) = 0.$$

Therefore,

$$\tan 2\theta_0 = \frac{\mathrm{Re}(f_0''\overline{f_0})}{\mathrm{Im}(f_0''\overline{f_0})}. \tag{8}$$

There are exactly two solutions in $-\frac{1}{2}\pi < \theta_0 \leq \frac{1}{2}\pi$, which differ by $\frac{1}{2}\pi$, confirming the orthogonality of the two r-lines through z_0.

Note that in the case $f(z) = e_n(z)$, we have $f'(z) = e_{n-1}(z)$, so that z_0 is a zero of e_{n-1}. This is clearly visible in the plots of §2. Furthermore, $f_0 = e_n(z_0)$, $f_0'' = e_{n-2}(z_0)$ if $n \geq 2$, so that (2) with $z = z_0$, once applied as is, and once with n replaced by $n - 1$, gives $f_0 = z_0^n/n!$, $f_0'' = -z_0^{n-1}/(n-1)!$, and the equation for θ_0 reduces to

$$\tan 2\theta_0 = -\frac{\mathrm{Re} z_0}{\mathrm{Im} z_0} \qquad (f = e_n).$$

3.2. *Contour lines* $\varphi =$ const. For the lines $\varphi =$ const, we take r as the independent variable and, by differentiating

$$f(z(r)) = re^{i\varphi}, \quad \varphi = \text{const},$$

with respect to r, obtain

$$\frac{dz}{dr} = \frac{e^{i\varphi}}{f'(z)}.$$

In terms of the arc length s, we now have

$$\frac{ds}{dr} = \left|\frac{dz}{dr}\right| = \frac{1}{|f'(z)|},$$

so that

$$\frac{dz}{ds} = \frac{dz}{dr}\frac{dr}{ds} = e^{i\varphi}\ \frac{|f'(z)|}{f'(z)},$$

or, written as a system of differential equations,

$$\begin{aligned} \frac{dx}{ds} &= \mathrm{Re}\left\{e^{i\varphi}\ \frac{|f'(z)|}{f'(z)}\right\}, \\ \frac{dy}{ds} &= \mathrm{Im}\left\{e^{i\varphi}\ \frac{|f'(z)|}{f'(z)}\right\}, \end{aligned} \qquad z = x + iy. \tag{9}$$

The singular point of (9) is again z_0, a zero of f'. At this point,

$$\frac{f_0}{|f_0|} = e^{i\varphi}, \qquad -\pi < \varphi \leq \pi,$$

which determines φ. The limit of $|f'(z)|/f'(z)$ as $z \to z_0$ may be determined by a procedure similar to the one in §3.1. Instead, we directly use the Taylor series of f in z_0 in order to study the φ-lines (and the r-lines as well) near z_0 with $f'(z_0) = 0$. Let $z = z_0 + \zeta$, where ζ is a complex increment, and let

$$f_k := f^{(k)}(z_0), \quad k \geq 0, \qquad f_0 \neq 0, \quad f_1 = 0, \quad f_2 \neq 0, \tag{10}$$

be the derivatives of f at z_0. Then the Taylor series is

$$f(z_0 + \zeta) = f_0 + f_2\frac{\zeta^2}{2!} + f_3\frac{\zeta^3}{3!} + \cdots\ . \tag{11}$$

Next, we observe that by defining $w = f_0\, e^u$ in (3), i.e., by putting

$$f(z_0 + \zeta) = f(z_0)\ e^u, \tag{12}$$

the r-lines through z_0 are given by the values of ζ corresponding to purely imaginary values $u = it$, whereas the φ-lines through z_0 are given by $u \in R$. The point z_0 itself corresponds to $\zeta = u = 0$. We therefore need to solve Equ. (12), with $f(z_0 + \zeta)$ substituted from (11), for ζ, which is a typical task for MAPLE.

In the program below[2] the series (11) and the equation (12) are denoted by `s` and `eq`, respectively. The `solve` command automatically expands e^u in a Taylor series and solves the equation by means of a series progressing in appropriate powers of u (here half-integer powers). As expected, two solutions corresponding to the two possible values of the square root are found. Only the first solution tt zet0[1] is processed further: first, by substituting the abbreviations `fk` defined in Equ. (10), then by introducing the variable `v` according to

$$u = \frac{v^2 f_2}{2f_0} \qquad \text{or} \qquad v = \left(\frac{2uf_0}{f_2}\right)^{\frac{1}{2}}\ . \tag{13}$$

The symbols `D(f)` and `(D@@k)(f)` stand for the derivative of f and the kth derivative of f, respectively. The call to the function `map` causes the operation defined by its first argument, here the simplification of the radicals, to be applied to each term of the expression defined by the second argument. Finally, the call to `series` causes the O-term to be simplified.

[2] The authors are indebted to Dominik Gruntz for this program.

```
N := 5: Order := N;
s  := series(f(z0+dz), dz):
s0 := subs(D(f)(z0)=0, s);
eq := s0 = f(z0)*exp(u):
zet0 := [solve(eq, dz)];
zet1 := subs(seq( (D@@k)(f)(z0) = f.k, k=0..N-1), zet0[1]):
zet2 := map(radsimp, subs(u=v^2*f2/2/f0, zet1));
zeta := series(zet2,v);
```

The output of this program contains the following lines:

```
                   (2)          2        (3)          3
s0 := f(z0) + 1/2 D   (f)(z0) dz  + 1/6 D   (f)(z0) dz

            (4)          4       5
    + 1/24 D   (f)(z0) dz  + O(dz )

                    2                            3       2      3
                f3 v         (- 3 f2 f4 f0 + 9 f2  + 5 f3  f0) v       4
zeta := v - 1/6 ----- + 1/72 ------------------------------------ + O(v )
                  f2                             2
                                            f0 f2
```

The MAPLE program works for any $N \geq 3$, producing $N-2$ terms of the above series. However, it is fairly slow, since no ”intelligence“, such as information on the form of the resulting series, is built in. To be able to find this series, nevertheless, is a good accomplishment of a general-purpose symbolic manipulator. It can be seen that ζ may be written as a formal power series in the variable v defined in (13). If the original series (11) converges in a neighborhood of z_0, the resulting series converges in a neighborhood of $v = 0$.

The directions θ_0 of the r-lines at z_0 are now given by the values of ζ corresponding to $u = it$ in the limit $t \to 0$. The above series and Equ. (13) immediately yield

$$\theta_0 = \arg v = \frac{1}{2} \left(\arg f_0 - \arg f_2 \pm \frac{\pi}{2}\right) .$$

Hence there are two r-lines through z_0 intersecting at right angles, in perfect agreement with Equ. (8).

The directions of the φ-lines through z_0, on the other hand, are given by (13) for real values of u. We obtain the two directions $\theta_0 \pm \frac{\pi}{4}$, i.e., the tangents of the two φ-lines through z_0 are the bisectors of the tangents of the r-lines.

4. The Contour Lines $r = 1$ of $f = e_n$

As indicated in §3.1, we need to solve (7) with initial values $x = y = 0$. Let $\mathrm{S_n}$ be the point of intersection of the 1-line of e_n with the negative real axis. By symmetry, only the portion of each 1-line lying in the upper half of the complex plane needs to be computed. This is not quite easy in MATLAB, since the `ode` routine requires a terminal value s_f of the independent variable s and (unfortunately) does not allow termination of the integration process according to some condition (such as $y \geq 0$) being violated. One way to deal with this problem is to select s_f as an upper bound of the arc length on the 1-line between the origin and $\mathrm{S_n}$. Then, only the points satisfying the condition $y \geq 0$ need to be plotted. The point $\mathrm{S_n}$ can be approximated by linear interpolation between the two points on the 1-line closest to $\mathrm{S_n}$.

In the MATLAB program below this is done by using the `find` command with the parameter $y \geq 0$ in order to find the subset of all points satisfying the condition $y \geq 0$. Their indices are collected in the vector `indices`. The indices of the points used in linear interpolation are then `l=length(indices)` and `l1=l+1`. Finally, `w` is the normalized row vector containing the two interpolation weights, and the actual interpolation is carried out by the product `w*z(l:l1,:)`.

A good upper bound s_f for the arc length is obtained from the observation that the region $|e_n(z)| \leq 1$ approaches a semidisk of radius $\rho(n)$ as $n \to \infty$. An asymptotic analysis shows that

$$\rho(n) = \exp(-1) \cdot (n + \log\sqrt{2\pi n} + O(1)).$$

It suffices to choose $O(1) = 3$; then we obtain

$$s_f \sim (1 + \frac{\pi}{2}) \exp(-1) \cdot (n + \log\sqrt{2\pi n} + 3) \tag{14}$$

as a close upper bound for the arc length up to the point $\mathrm{S_n}$.

```
% Level curves r=1 for the first 21 exponential sums (Figure 2)
%
global n
nmin=1; nmax=21; tol=3.e-8;
axis equal; hold on; % arc=[];
for n=nmin:nmax,
  sf = 0.94574*(n+.5*log(2*pi*n)+3);   %% 0.94574=(1+pi/2)*exp(-1)
  [s,z] = ode45('level',0,sf,[0;0],tol);
  indices = find(z(:,2)>=0); l = length(indices); l1 = l+1;
  w = [-z(l1,2),z(l,2)]; w = w/sum(w); z(l1,:) = w*z(l:l1,:);
  plot(z(1:l1,1),z(1:l1,2));
% arc = [arc;w*s(l:l1),sf]; %% approximate arclengths and bounds sf
end; % arc
%
% LEVEL generates the right-hand side of the differential equ
% for the 1-lines of f(z) = 1 + z + z^2/2! + ... + z^n/n!
% Save this as the m-file level.m
%
function zdot=level(s,z)
  global n
  zc = z(1)+i*z(2); t = 1; f0 = 0; f = t;
  for k=1:n, t = t*zc/k; f0=f; f = f+t; end;
  q = f/f0; zdot = [-imag(q);real(q)]/abs(q);
end;
```

The MATLAB program begins with the definitions of the parameters `nmin, nmax` and the error tolerance `tol`. The 1-lines in the range `nmin` $\leq n \leq$ `nmax` are generated in the accuracy given by `tol` and plotted. Rerunning the program with new values of `nmin` and `nmax` adds new curves to the figure. The statements turned off by the comment marks % generate a table `arc` containing the actual arc lengths and the upper bounds s_f computed from (14).

The actual integration is done in the call to the integrator `ode45`. The input parameters of this procedure are: the name `'level'` (in string quotes) of the `m`-file defining the differential equations (7) to be integrated, the initial value `0` and the final value `sf` of the independent variable, the column vector `[0;0]` of the inital values, and (optionally) the error tolerance `tol`. The choice `tol=3.0e-8` yields a high-resolution plot, whereas the default `tol=1.0e-6` (when the parameter is omitted in the call) still yields a satisfactory plot. The values of the independent and dependent variables generated by the integrator are stored as the vectors `s` and `z`, respectively, ready to be plotted.

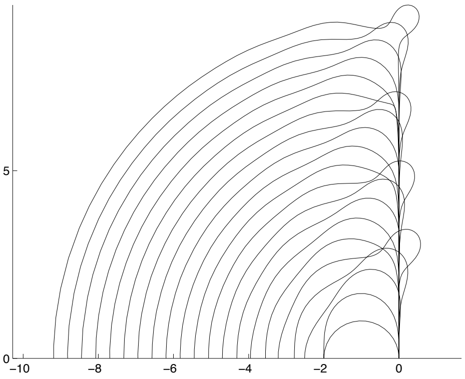
The results for $n = 1:1:21$ are shown in Figure 2 below. The features near the imaginary axis at the transition to the circular part seem to show a periodicity in n of a little over 5. For example, the curves corresponding to $n = 5, 10, 15, 21$ all show a particularly large protrusion into the right half-plane.

An investigation of this phenomenon is interesting, but exceeds the scope of this article. We limit ourselves to reporting that as $n \to \infty$, the period tends to

$$\frac{2\pi}{\frac{\pi}{2} - \exp(-1)} = 5.22329130.$$

This result was obtained by considering the function $e_\nu(z)$ for real values of ν (which leads to the incomplete gamma function) and requiring the 1-line of $e_\nu(z)$ to contain a saddle point with $e'_\nu(z) = 0$.

FIGURE 2. *Level Curves* $r = 1$ *for the First* 21 *Exponential Sums*



5. The Contour Lines $\varphi = \text{const}$ of $f = e_n$

Below is a MATLAB program that implements the method of §3.2 for any fixed $n > 0$, where the differential equations (9) must be implemented in the function `phase` and stored in the `m`-file `phase.m`. The program begins with the definitions of `n`, the error tolerance `tol`, and the desired arc length `sf` of the curve segments emanating from the zeros. Then, the vector `r` of the zeros of e_n is computed by means of the function `roots`, where the coefficients of e_n are generated by means of the gamma function. On the next line the subset of the zeros in the upper half-plane is formed by means of the `find` command with the argument `imag(r)>=0`. The statement used in the program stores all the indices defining the subset in the vector `indices`; then `r(indices)` is the vector of the zeros of e_n in the upper half-plane (which is printed for convenience).

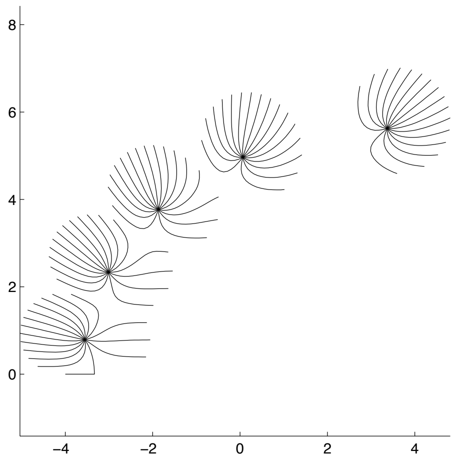
The input parameters in the call to the integrator `ode45` are: the name `'phase'` (in string quotes) of the differential equations to be integrated, the initial value `0` and the final value `sf` of the independent variable, the column vector `z0` of the inital values, and (optionally) the error tolerance `tol` (default 10^{-6} when omitted). The values of the independent and dependent variables generated by the integrator are stored as the vectors `s` and `z`, respectively, ready to be plotted.

```
% Lines of constant phase for the 10th exponential sum (Fig 3)
%
global n phi
n = 10; tol = 1e-5; sf = 1.5;
clf; axis image; hold on;
r = roots(1./gamma(n+1:-1:1));
indices = find(imag(r)>=0); zero=r(indices)
for k=1:length(zero),
  z0 = [real(zero(k));imag(zero(k))];
  for phi=-7/8*pi:pi/8:pi,
    [s,z] = ode45('phase',0,sf,z0,tol);
    plot(z(:,1),z(:,2));
  end;
end;
%
% Save this as the m-file phase.m
%
function zdot=phase(s,z)
  global n phi
  eiphi = exp(i*phi); zc = z(1)+i*z(2);
  t = 1; f = t;
  for k=1:n-1, t = t*zc/k; f = f+t; end;
  q = eiphi*abs(f)/f; zdot = [real(q);imag(q)];
end;
```

The result for $n = 10$ is shown in Figure 3[3].

[3] We wrote and ran the script on August 1, 1996, while fireworks went off in celebration of the Swiss national holiday.

FIGURE 3. *Lines of Constant Phase for the* 10*th Exponential Sum*



None of the complex singular points is in evidence in Figure 3 since the φ-values chosen do not correspond to level lines passing through a complex singular point. The real singular point at $z_0 = -3.333551485$, however, is clearly visible by an abrupt right-angled turn of the line $\varphi = 0$ (near the bottom of the figure). It is curious to note how the `ode45` integrator was able to integrate right through the singularity, or so it seems.

References

[1] A.J. CARPENTER, R.S. VARGA, J. WALDVOGEL, *Asymptotics for the zeros of the partial sums of* e^z*. I*, Rocky Mountain J. Math., 21, 1991, pp. 99–120.

[2] K.E. IVERSON, *The zeros of the partial sums of* e^z, Math. Tables and Other Aids to Computation, 7, 1953, pp. 165–168.

[3] G. PÓLYA, G. SZEGÖ, *Problems and Theorems in Analysis, Vol. II, Part Five, Exercise* 74, Springer-Verlag, Berlin, 1976.

[4] H.R. SCHWARZ, *Numerical Analysis. A Comprehensive Introduction*, John Wiley & Sons, Chichester, 1989.

Research Reports

No.	Authors	Title
96-15	W. Gautschi, J. Waldvogel	Contour Plots of Analytic Functions
96-14	R. Resch, F. Stenger, J. Waldvogel	Functional Equations Related to the Iteration of Functions
96-13	H. Forrer	Second Order Accurate Boundary Treatment for Cartesian Grid Methods
96-12	K. Gerdes, C. Schwab	Hierarchic models of Helmholtz problems on thin domains
96-11	K. Gerdes	The conjugated vs. the unconjugated infinite element method for the Helmholtz equation in exterior domains
96-10	J. Waldvogel	Symplectic Integrators for Hill's Lunar Problem
96-09	A.-T. Morel, M. Fey, J. Maurer	Multidimensional High Order Method of Transport for the Shallow Water Equations
96-08	A.-T. Morel	Multidimensional Scheme for the Shallow Water Equations
96-07	M. Feistauer, C. Schwab	On coupled problems for viscous flow in exterior domains
96-06	J.M. Melenk	A note on robust exponential convergence of finite element methods for problems with boundary layers
96-05	R. Bodenmann, H.J. Schroll	Higher order discretisation of initial-boundary value problems for mixed systems
96-04	H. Forrer	Boundary Treatment for a Cartesian Grid Method
96-03	S. Hyvönen	Convergence of the Arnoldi Process when applied to the Picard-Lindelöf Iteration Operator
96-02	S.A. Sauter, C. Schwab	Quadrature for *hp*-Galerkin BEM in $\mathbb{R}^3$
96-01	J.M. Melenk, I. Babuška	The Partition of Unity Finite Element Method: Basic Theory and Applications
95-16	M.D. Buhmann, A. Pinkus	On a Recovery Problem
95-15	M. Fey	The Method of Transport for solving the Euler-equations
95-14	M. Fey	Decomposition of the multidimensional Euler equations into advection equations
95-13	M.D. Buhmann	Radial Functions on Compact Support
95-12	R. Jeltsch	Stability of time discretization, Hurwitz determinants and order stars