

High Precision Conformal Map of the Unit Disk to Star-Shaped Domains

Martin Holzinger

Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10,
1040 Vienna, Austria; martin.holzinger@tuwien.ac.at

SNE 30(3), 2020, 85-94, DOI: 10.11128/sne.30.tn.10521
Received: July 15, 2020; Revised: August 20, 2020;
Accepted: August 25, 2020
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Having the conformal map from unit square to unit disk at hand, we ask ourselves for a way to numerically map the disk to more general domains. Once such domains are parametrized by the square, knowledge of metric quantities flows in by the derivatives of the map and simulation of PDEs on such domains can easily be achieved. One way to numerically construct the map for star-shaped regions yields over solving Theodorsen's integral equation which establishes the boundary correspondence of angles. Focusing on highly accurate solutions, we present *Mathematica* test implementations and results for maps from unit disk to inverted ellipse (a), unit square (b) and onto a more general domain (c). Finally, the metric impact of the conformal map on the PDE itself is being investigated to enlighten the process of correcting spacial Finite-Difference approximations in general.

Introduction

In [4] the CTDS-Method has been used to solve a heat conduction problem on the unit square. Moreover, we showed that using elliptic functions to conformally map the square to the unit disk provides a way to treat partial differential equations aside regular structured grids necessary for CTDS. In an alternative point of view, by conformal parametrization of the disk by the square this regularity is being kept up - the map thus is not interpreted as point transformator but rather as an introducer of curvi-linear co-ordinates. Following this path, we investigate ways to numerically construct conformal maps from unit disk to more general domains.

1 THEODORSEN'S EQUATION

Suppose you are interested in finding the conformal map $\zeta = \xi + i\eta = f(w) = f(u + iv)$ from unit disk \mathfrak{E} onto a domain \mathfrak{G} bounded by a closed, piecewise Jordan arc \mathfrak{C} with polar representation $\rho(\theta)$. For uniqueness we demand $f(0) = 0$, $f'(0) > 0$ and moreover, let \mathfrak{G} be *star-shaped* with respect to $\zeta_0 = 0$,

$$\zeta_0 + t(\hat{\zeta} - \zeta_0) \in \mathfrak{G}, \quad \forall t \in [0, 1), \quad \forall \hat{\zeta} \in \mathfrak{C}.$$

GAIER [3] presents an elegant way of construction by first deriving an equation for the (real) *function of boundary correspondence* which links the angle $\varphi \in [0, 2\pi)$ on \mathfrak{E} to the angle $\theta(\varphi)$ describing \mathfrak{C} . For f regular on \mathfrak{E} and continuous on the closure $\bar{\mathfrak{E}}$ he concludes

$$\eta(e^{i\varphi}) = \eta(0) + \frac{1}{2\pi} \int_{0(p.v.)}^{2\pi} \xi(e^{i\vartheta}) \cot \frac{\varphi - \vartheta}{2} d\vartheta.$$

Introducing the help function $F(w) := \ln \frac{f(w)}{w}$,

$$F(w) = \ln |f(w)| - \ln |w| + i \arg(f(w)) - i \arg(w)$$

can be derived and putting all together results in

$$\theta(\varphi) = \varphi + \frac{1}{2\pi} \int_{0(p.v.)}^{2\pi} \ln \rho(\theta(\vartheta)) \cot \frac{\varphi - \vartheta}{2} d\vartheta$$

known as THEODORSEN'S nonlinear and singular integral equation for establishing the boundary correspondence. For existence and uniqueness of a solution, $\rho(\theta)$ in $[0, 2\pi]$ absolute continuous and $|\sigma(\theta)| := |\rho'(\theta)/\rho(\theta)| \leq \varepsilon < 1$ almost everywhere are sufficient conditions. Latter is the so-called ε -condition.

As will be considered later, even in case of $\varepsilon > 1$ numerical algorithms are able to produce feasible results if carefully picked. On the other hand, the shape of the concrete geometry has heavy impacts on the quality of the numerical approximation, e.g. if \mathfrak{C} contains corners.

2 Approximation Methods for THEODORSEN'S EQUATION

In order to achieve a high-precision approximation for $\theta(\varphi)$, the function of boundary correspondence, for continuous and 2π -periodic real functions $h(\varphi)$ let us first introduce the operator

$$\mathcal{K}[h](\varphi) = h^*(\varphi) := \frac{1}{2\pi} \int_{0(p.v.)}^{2\pi} h(\vartheta) \cot \frac{\varphi - \vartheta}{2} d\vartheta.$$

Here GAIER summarizes in [3] that the linear operator $\mathcal{K} : h \mapsto h^*$ has some properties *très chic*, namely producing an again 2π -periodic, continuous function $h^*(\varphi)$, whose Fourier series can be gained immediately out of the series for $h(\varphi)$: If we have an expansion

$$h(\varphi) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\varphi + b_n \sin n\varphi)$$

at hand, we can deduce the series

$$h^*(\varphi) = \sum_{n=1}^{\infty} (a_n \sin n\varphi - b_n \cos n\varphi)$$

for the so-called *conjugate function* $h^*(\varphi)$. Having now a closer look at THEODORSEN'S equation, with $h = \ln \rho(\theta)$ it re-writes as $\theta(\varphi) = \varphi + \mathcal{K}[\ln \rho(\theta)](\varphi)$ and a Fourier sum for the nonlinear and singular integral can be applied. In fact, Fourier synthesis of $\ln \rho(\theta)(\varphi)$ turns out to be a very powerful method for our purpose of numerically constructing conformal maps.

Fix point iteration. One way to numerically treat THEODORSEN'S equation is by means of successive approximation. The 'fix point' in question is in our case a monotonically increasing function as a solution of the functional fix point problem

$$\theta = \iota + \mathcal{K}[\ln \rho(\theta)] = \mathcal{F}(\theta).$$

Formally, the iteration process is then defined by

$$\theta_{n+1} = \mathcal{F}(\theta_n), \quad n = 0, 1, 2, \dots,$$

and the iteration itself can by default be started with $\theta_0 = \iota$, the identity function (which trivially solves the problem for $\mathfrak{C} = \mathfrak{E}$). Of course any other more apt θ_0 can be used instead if at hand, e.g. any established boundary correspondence function $\theta(\varphi)$ for a similar shaped geometry \mathfrak{C} would be predestined for such an intention.

Newton's method. The well-known quadratic behavior when iterating the roots of real functions gives rise to reformulate the method to a functional zeroing problem. With $\psi(\varphi) = \theta(\varphi) - \varphi$, THEODORSEN'S equation reads $\psi = \mathcal{K}[\ln \rho(\psi + \iota)]$, hence [7, p.235]

$$\mathcal{F}(\psi) = \psi - \mathcal{K}[\ln \rho(\psi + \iota)] \equiv 0$$

has to be solved which can be formally iterated via

$$\psi_{n+1} = \psi_n - [\mathcal{F}'(\psi_n)]^{-1} \mathcal{F}(\psi_n), \quad n = 0, 1, 2, \dots,$$

whereas the statement concerning the starting function, $\psi_0 \equiv 0$ **or better if available**, turns in fact out to be essential for convergence of the iteration process.

To get the iteration method ready for implementation, one still has to resolve the role played by the operator \mathcal{F} . According to HÜBNER [6], this can be done by investigating its FRÉCHET derivative \mathcal{F}' while its necessary inversion $[\mathcal{F}']^{-1}$ is handled by solving a RIEMANN-HILBERT problem. Using these results, WEGMANN presents in [9] a formidable discrete-type algorithm which we made use of in our *Mathematica* implementations: Starting with an even (let $N = 2n$) decomposition of $[0, 2\pi)$, hence $\varphi^{(k)} = \frac{k\pi}{n}, k = 0, \dots, N-1$, and a given vector $\theta_0 = \varphi$ or better, \mathcal{K} is initially replaced by its discrete equivalent, the WITTICH operator

$$\mathcal{K}_N[h] = \sum_{j=1}^{n-1} a_j \sin j\varphi - b_j \cos j\varphi, \quad \text{where}$$

$$a_j = \frac{1}{n} \sum_{k=1}^N h^{(k)} \cos j\varphi^{(k)}, \quad b_j = \frac{1}{n} \sum_{k=1}^N h^{(k)} \sin j\varphi^{(k)}.$$

Note that all appearing vectors are set in bold type. With these preliminary remarks a WEGMANN step reads

$$\theta_{i+1} = \theta_i - \frac{\mathbf{f}}{r^2} - \frac{(\mathbf{p} + \hat{q} \cdot \tan \hat{v}) \cdot \exp(\mathbf{w})}{\mathbf{r}},$$

the remaining task is calculating the auxiliary quantities (component-by-component, code sample see page 94):

$$\mathbf{f} = \theta_i - \varphi - \mathcal{K}_N[\ln \rho(\theta_i)], \quad \mathbf{r} = \sqrt{1 + \sigma^2(\theta_i)},$$

$$\mathbf{v} = \arctan \sigma(\theta_i), \quad \mathbf{w} = \mathcal{K}_N[\mathbf{v}], \quad \hat{v} = \frac{1}{N} \sum_{k=1}^N v^{(k)},$$

$$\mathbf{q} = \frac{\mathbf{f} \cdot \sigma(\theta_i)}{\mathbf{r} \cdot \exp(\mathbf{w})}, \quad \mathbf{p} = \mathcal{K}_N[\mathbf{q}], \quad \hat{q} = \frac{1}{N} \sum_{k=1}^N q^{(k)}.$$

3 Continuation on Unit Disk

As soon as $\theta(\varphi)$, the function of boundary correspondence is numerically determined, the remaining task consists in finding a way to get hold of $f(w)$, the conformal map from \mathfrak{E} to \mathfrak{G} . However the numeric construction of $\theta(\varphi)$ was achieved in detail, let us here merely suppose that this function is provided by the implementation as a continuous function.

In case that we made use of conjugate functions within our iteration process, the discrete Fourier coefficients a_k and b_k of $\ln \rho[\theta(\varphi)]$ can even be used for a continuation to the interior of the unit disk: Solve a DIRICHLET boundary value problem [5, p.27f.] to get

$$\zeta = f(w) = w \cdot \exp \left[\frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k - ib_k) w^k \right].$$

This completes the elegant method GAIER showed us some sixty years ago. We note that Fourier synthesis of $\ln \rho[\theta(\varphi)]$ is the nucleus of the algorithm.

It is next to be stated that the underlying discrete trigonometric interpolation can only be expected to be acceptably accurate when the shape of the geometry to be mapped endorses this approach. In particular will an occurrence of vertices result in discontinuities of $\sigma(\theta)$ being fatal for this algorithm.

On the other hand, on smooth geometries even in case of $\varepsilon > 1$ WEGMANN'S algorithm can be used to produce highly accurate solutions to a certain degree, with convergence orders then falling back from quadratic to linear which is in fact a matter of computational time.

Aside trigonometric interpolation, SCHWARZ or CAUCHY formulae can be used to snatch the map, with $\theta(\varphi)$ known the former results in

$$f(w) = w \cdot \exp \left[\frac{1}{2\pi} \int_0^{2\pi} \ln \rho(\theta(\vartheta)) \frac{e^{i\vartheta} + w}{e^{i\vartheta} - w} d\vartheta \right]$$

whereas in the latter case f can be expressed as

$$f(w) = w \cdot \exp \left[\frac{1}{2\pi} \int_0^{2\pi} \frac{\ln \rho(\theta(\vartheta)) + i(\theta(\vartheta) - \vartheta)}{1 - we^{-i\vartheta}} d\vartheta \right].$$

Mathematica handles the evaluation of those complex-valued integrals with ease. Finally, from theory it is known that the overall error maxima of the numerical approximation will occur at the boundary. With this preparations, we are ready to do some numerical experiments and check the validity of our hitherto remarks.

4 Numerical Experiments

Let us first put our focus on two geometries with well-known closed-form solutions for the function of boundary correspondence as well as for the conformal map itself and then study the methods for a more general geometry where symmetry should hold with respect to the η -axis. Domain boundary points can be prescribed and its radius function can then be deduced by interpolation.

Aside the conformal map also its derivatives play a major part in a bigger context of our considerations and we shall therefore include at least numerical error examinations of first derivatives in our investigations.

4.1 Inverted ellipse

With $\rho(\theta) = \sqrt{1 - (1 - p^2) \cos^2 \theta}$ describing the boundary ($0 < p < 1$) the solution is available in closed-form, we have [3, p.264]

$$\theta(\varphi) = \arctan(p \tan \varphi)$$

for the function of boundary correspondence and

$$\zeta = f(w) = \frac{2pw}{1 + p + (1 - p)w^2}$$

for the conformal map itself. To set out to coding, we implement a basic fix point iteration code shown on the next page. Our intention is to keep things simple here to avoid programming errors and move on to more complex implementations when results are satisfying. As you can see, $\theta(\varphi)$ is implemented as a continuous function, trigonometric interpolation being carried out after evaluating $\ln \rho$ at 256 points using `LinearModelFit` throughout for the sake of higher working precision.

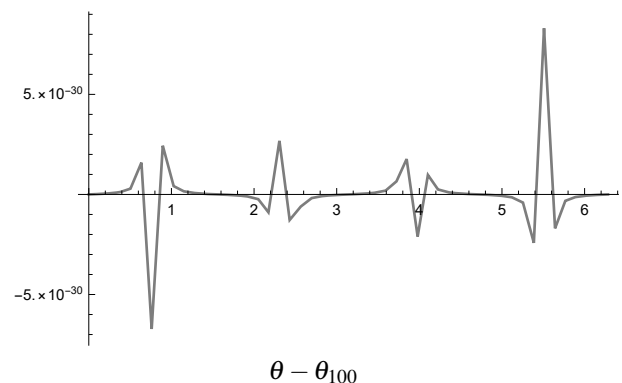


Figure 1: Verifying the base implementation: Calculating 100 fixpoint loops with $2n = 256$ discretization points.

```
(* Inverted Ellipsis: Crude Fixpoint Iteration Code as a starting point *)
p=6/10;zn=256;n=zn/2;r[phi_]=Sqrt[1-(1-p^2)Cos[phi]^2];lnR[phi_]=Log[r[phi]];
niter=100;basis=Table[Cos[2m*phi],m,0,zn/4];thetaneu[phi_]=phi;
Do[
  lnRv=Table[phi,lnR[thetaneu[phi]],phi,0,2Pi-Pi/n,Pi/n];
  lmfitlnR=LinearModelFit[lnRv,basis,phi,WorkingPrecision->80];
  lnRapprox[phi_]=lmfitlnR["BestFit",WorkingPrecision->80];
  anbeta=Coefficient[lnRapprox[phi],Table[Cos[m*phi],{m,1,n}]];
  klnRtrigapprox[phi_]=Sum[anbeta[[j]]*Sin[j*phi],{j,1,n}];
  thetaneu[phi_]=phi+klnRtrigapprox[phi],{i,1,niter}]
```

Of course we can do much better with WEGMANN'S algorithm which is predestined for smooth geometries like the inverted ellipse, with $p = 0.6$ an ϵ -condition $\max|\sigma(\theta)| = 8/15 < 1$ results. Using η -symmetry, discrete Fourier transform [5, p.40] and external calculation of multiple used structures, or code on page 94 runs fast as lightning straight down to predefined WorkingPrecision within seven iteration steps (15s in total using $2n = 1024$ points), cf. Figure 2.

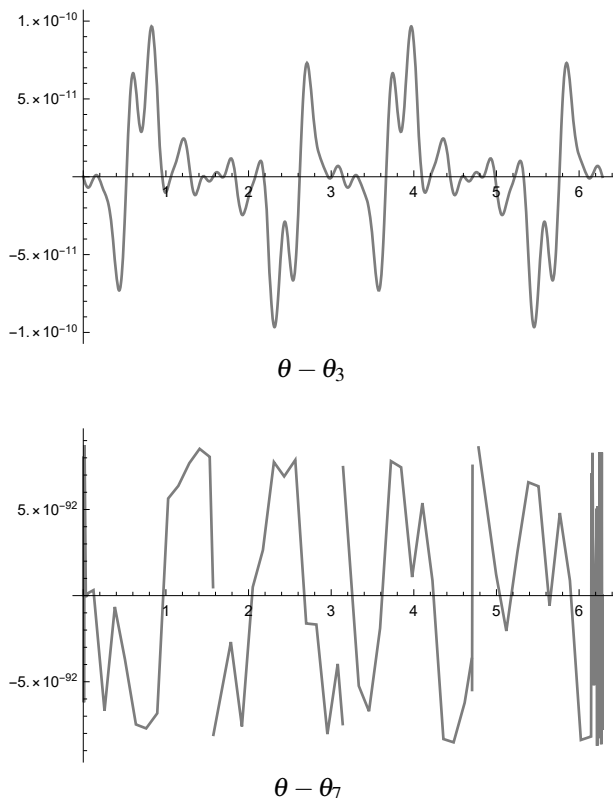


Figure 2: Approximation errors using WEGMANN'S method.

Figure 3 shows the results obtained when joining the map calculated with the trigonometric series for the unit disk and the map used in [4] transforming square to disk with elliptic functions. Note that computational errors for the derivatives show a very similar behavior.

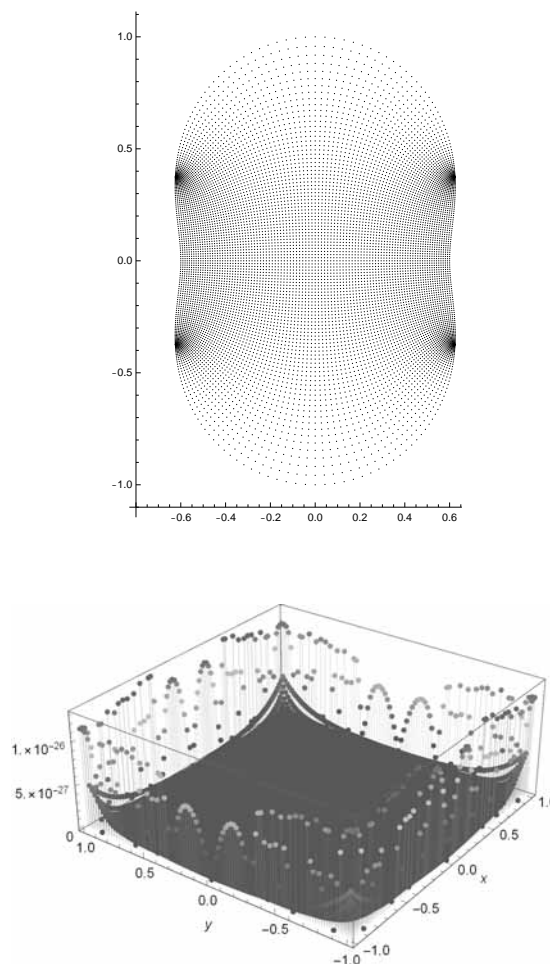


Figure 3: Mapping square to ellipse (t). Error behavior (b).

4.2 Unit square

Mapping the disk back to the square numerically is of particular interest. While all involved functions are known in closed form [3, p.265] and the square can be described via $\rho(\theta) = \sec(\theta)$, $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, trigonometric approximation will be bound to fail with respect to numerical accuracy when constructing the function of boundary correspondence. We shall use the closed-form functions only for comparison purposes here.

Main difficulties arise primarily caused by the presence of corner singularities. For a piecewise smooth closed Jordan arc BLÜMLINGER [1, p.22] showed that in a vertex point $\zeta_i = f(w_i)$ with interior turning angle α_i/π , one can extract the singularity thus the derivative has a representation $\theta'(\varphi) = |\varphi - \varphi_i|^{\alpha_i-1} \lambda(\varphi)$. Using symmetries, on $[-\frac{\pi}{4}, \frac{\pi}{4}]$ we can try an even approach

$$\theta'(\varphi) = \left(\frac{\pi^2}{16} - \varphi^2\right)^{-\frac{1}{2}} \sum_{n=0}^{\infty} a_{2n} \varphi^{2n}.$$

A similar discussion can be found in [7, p.327, Theorem (12.2.1)]. Digressing trigonometric functions and conjugate expansions, RICHARDSON presents a coding sample in [8, p.471ff.] which we were able to refine using base functions $(\frac{\pi^2}{16} - \varphi^2) \arcsin^{2n-1}(4\varphi/\pi)$ and $(\frac{\pi^2}{16} - \varphi^2)^{(2n-1)/2} \arcsin(4\varphi/\pi)$ for an approximation of the function of boundary correspondence itself. In addition, only partial integration has been applied.

Figure 4 shows the resulting error, after 60 iteration steps with RICHARDSONS method the algorithm stagnates. Taking his approximation for θ (with error 10^{-7}) as a starting function, feeding it in our procedure and doing another 40 iteration steps, an error 10^{-11} results.

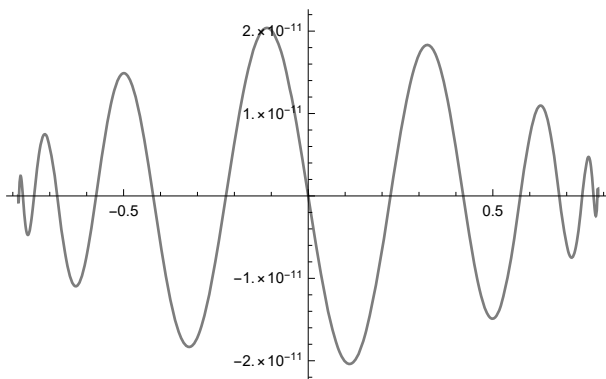


Figure 4: Approximation error $\theta - \theta_{100}$ on $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

We see that, in contrary to the smooth situation of the inverted ellipse in the section before, neither Newton's method nor the usage of trigonometric conjugation is an option to snatch the operator $\mathcal{K}[\ln \rho(\theta)]$ in this case.

Moreover, the idea behind the algorithm is to attack the integral in Theodorsen's equation directly within the fix point loop using `NIntegrate` and a handful of discretization points. Tell *Mathematica* the angular location of the singularities within the integration routine and you can expect no further complaints. As before, `LinearModelFit` is used to gain a continuous representation of θ , interpolation now being done by replacing sin-/cos-functions with the new base functions.

Figure 5 shows that we can even control the derivative error in the vicinity of the singularity while Figure 6 illustrates the overall error when numerically parametrizing the unit square by itself. The interior of the disk was in this case treated with CAUCHY formula.

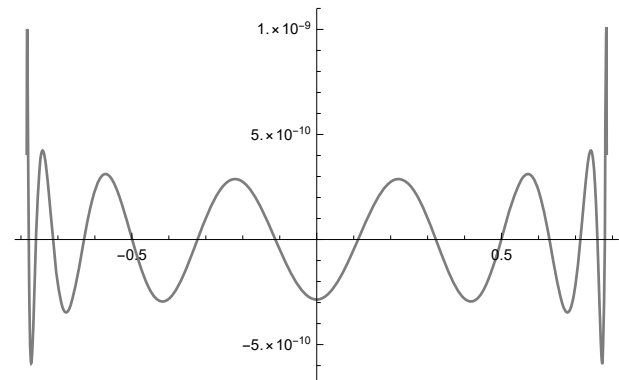


Figure 5: Approximation error $\theta' - \theta'_{100}$ on $[-\frac{\pi}{4} + 10^{-4}, \frac{\pi}{4} - 10^{-4}]$.

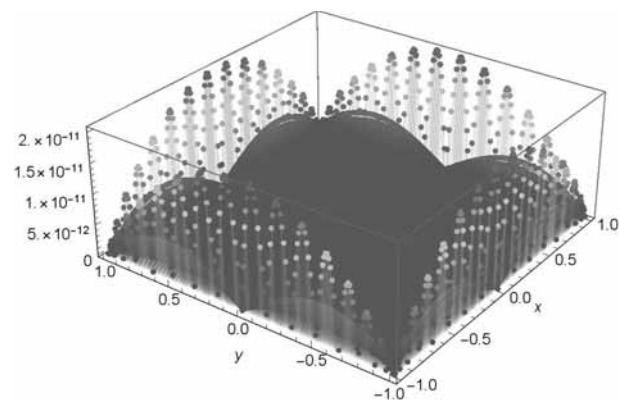


Figure 6: A previously exact mapped 201×201 -grid on the square is numerically mapped back from the unit disk. Expect the error maxima on the boundary.

4.3 General axially symmetric domains

For more general star-shaped domains we first implement a continuous boundary description function $R(\theta)$ by means of trigonometric¹ or Hermite² interpolation. In *Mathematica* this can easily be achieved by creating a list `geo` containing the prescribing boundary points (cf. Table 1 with η -symmetric shape definitions for three sample geometries). See Figure 8 for the resulting geometries and their corresponding ε -conditions. Also note that we are thus able to provide a radius value to any desired angular argument θ .

With such a provisioning of the radius function $R(\theta)$ we can now make use of one of the above mentioned iteration processes: Fix point approximation or NEWTON iteration in either its continuous or discrete form to get hold of the function of boundary correspondence.

As we are lacking an exact solution $\theta(\varphi)$ for comparison in more general cases, the iteration differences $E_i(\varphi) := \theta_i(\varphi) - \theta_{i-1}(\varphi)$ can be observed in case that the iterations are carried out with continuous functions $\theta_i(\varphi)$ and, likewise for discrete iterations, a discrete error vector can be defined via $e_i^{(k)} := \theta_i^{(k)} - \theta_{i-1}^{(k)}$.

Moreover, it is to be expected that for the conformal map f itself the error maxima will again occur at the boundary. So once the calculation of f is successfully done, it is useful to have a look at the expression $R(\arg f(e^{i\varphi})) - |f(e^{i\varphi})|$ for an additional error estimate.

In the present cases, all boundary curves are smooth enough to allow an application of WEGMANN'S method. Here, our implementation code mentioned on page 94 is quite optimized with respect to execution time by making use of symmetries, FFT and external provisioning of multiple used quantities. As in all smooth cases, using trigonometric approximation for the conjugate operator is best choice, the result for $\theta(\varphi)$ on geometry 3 using HERMITE interpolation for $R(\theta)$ is shown in Figure 7.

When using only 1024 discretization points on the unit circle, Figure 9(t) already shows a superb discrete error behaviour but this is only half of the rent. Recall that we are primarily interested in mapping the square to the geometry and when pre-joining the map from the disk with the closed-form SCHWARZ-CHRISTOFFEL transformation, our high-accurate discrete points on the disk will not be hit by the map coming from the

square. It is therefore necessary to focus on the continuous overall errors occurring on $[0, 2\pi]$. Depending on the geometry, their maxima turn out to be in range of $10^{-11}/10^{-6}/10^{-4}$ which is quite unsatisfactory. This can be improved by using 4096 discretization points. Results of the overall errors are shown in Figure 9(b).

θ	GEO1	GEO2	GEO3 (Hermite)		
	$\rho_1(\theta)$	$\rho_2(\theta)$	$\rho_3(\theta)$	$\rho_3'(\theta)$	$\rho_3''(\theta)$
0	7	6	5	5	20
$\pi/8$	9	9	—	—	—
$2\pi/8$	9	$7\sqrt{2}$	$7\sqrt{2}$	0	-30
$3\pi/8$	$15/2$	$15/2$	—	—	—
$4\pi/8$	$13/2$	7	7	0	25
$5\pi/8$	$15/2$	$15/2$	—	—	—
$6\pi/8$	9	$7\sqrt{2}$	$7\sqrt{2}$	0	-30
$7\pi/8$	9	9	—	—	—
π	7	6	5	-5	20
$9\pi/8$	$11/2$	5	—	—	—
$10\pi/8$	6	5	$7\sqrt{2}/2$	3	5
$11\pi/8$	8	$15/2$	$10\sqrt{2}/2$	10	30
$12\pi/8$	10	10	10	0	-50
$13\pi/8$	8	$15/2$	$10\sqrt{2}/2$	-10	30
$14\pi/8$	6	5	$7\sqrt{2}/2$	-3	5
$15\pi/8$	$11/2$	5	—	—	—
2π	—	—	5	5	20

Table 1: Definition of boundary points for trigonometric interpolation and derivatives (Hermite).

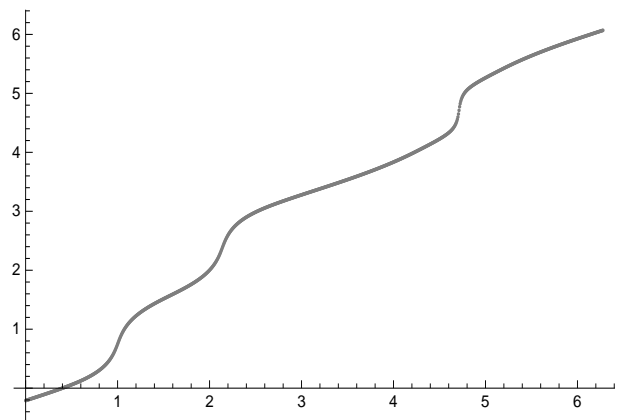


Figure 7: Function of boundary correspondence on GEO3.

¹basis=Union[Table[Cos[2m*phi], {m, 0, 4}], Table[Sin[(2m-1)*phi], {m, 1, 4}];
 lm=LinearModelFit[geo, basis, phi, WorkingPrecision->nprec, IncludeConstantBasis->False];
 R[phi_]=SetPrecision[lm["BestFit"], WorkingPrecision->nprec], Infinity];
²rfloat=Interpolation[geo, InterpolationOrder->3, PeriodicInterpolation->True, Method->"Hermite"];
 R[phi_]=SetPrecision[rfloat[phi], Infinity];

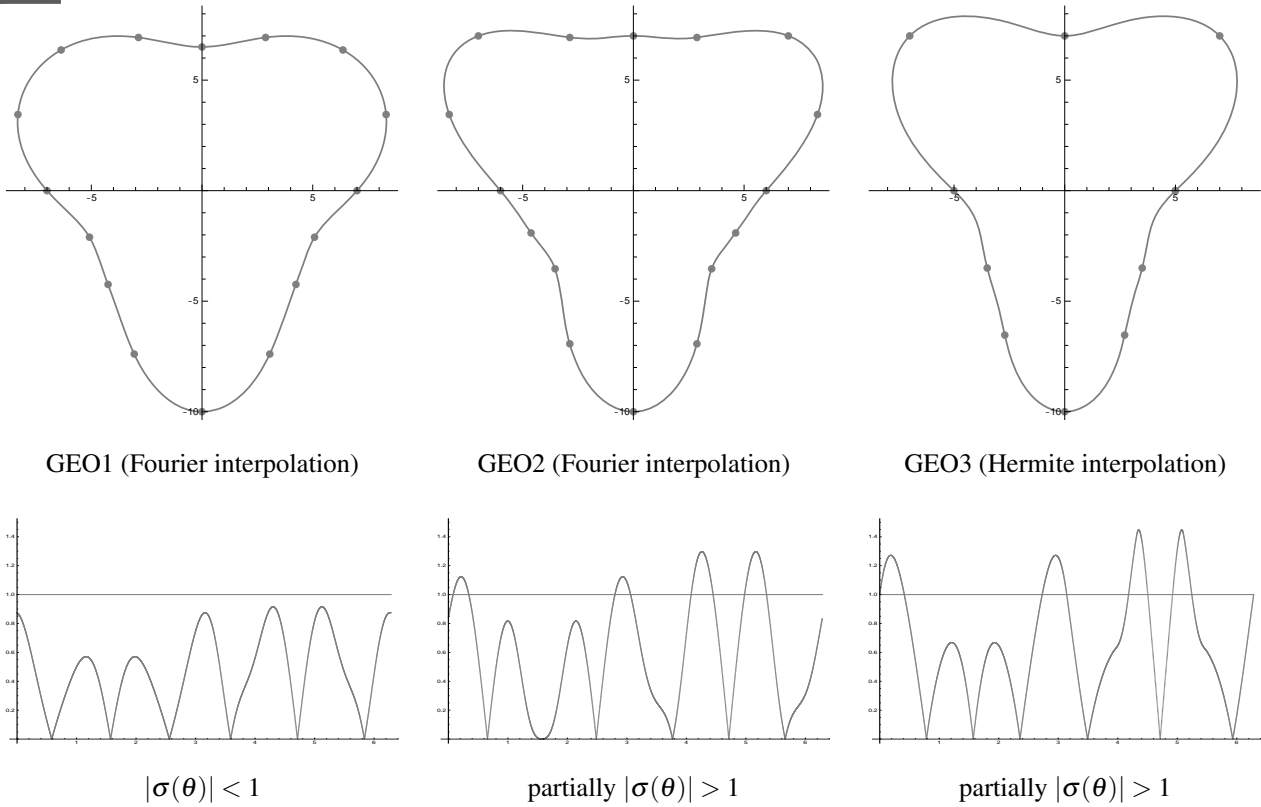


Figure 8: Points and interpolating functions $R(\theta)$ for the boundary curves (t). Symmetric $\rho(\theta)$ yields skew symmetric $\sigma(\theta)$ in $[0, \pi]$ and $[\pi, 2\pi]$ in respect of $\pi/2$ and $3\pi/2$ (b). The red line (symbolizing $\varepsilon = 1$) is used to investigate the ε -condition.

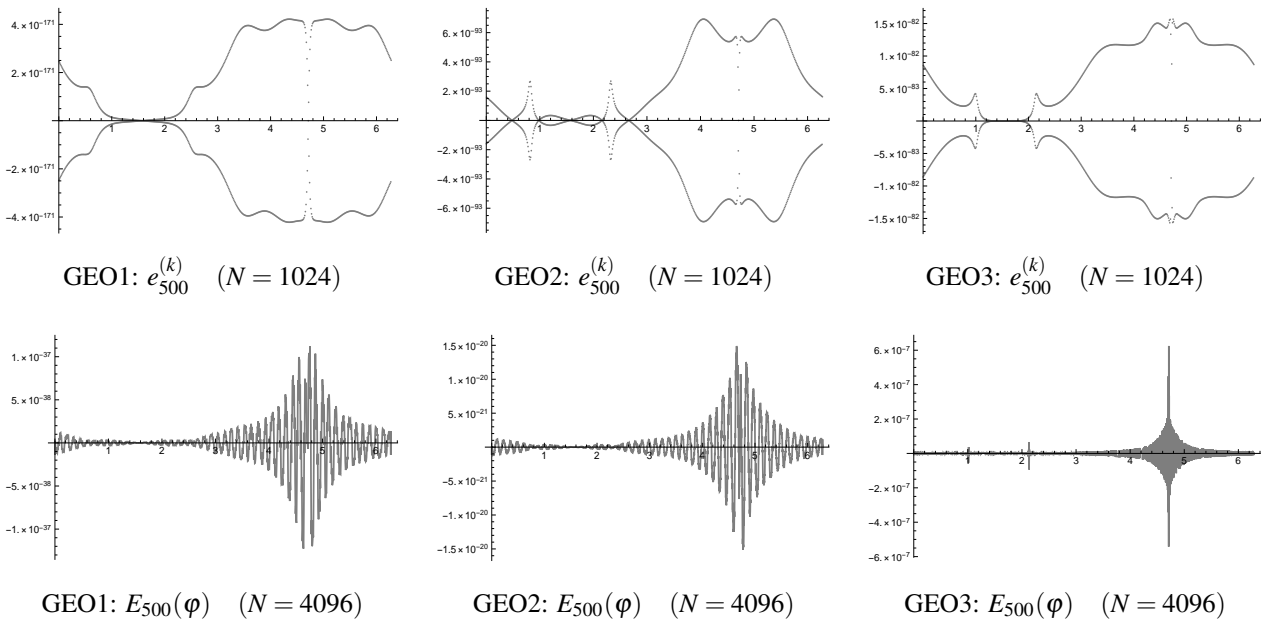


Figure 9: Discrete (t) and continuous (b) errors when iterating the function of boundary correspondence.

5 Conformal Co-ordinates

5.1 Metrics of the map

A given domain \mathfrak{G} in the complex plane can also be considered as a Riemannian manifold as a set of points. We are quite free how to address such a point $P \in \mathfrak{G}$, the most common way is the usage of Cartesian co-ordinates which can be expressed in terms of a homeomorphism $\kappa_\zeta : \mathfrak{D}_1 \subset \mathbb{R}^2 \rightarrow \mathfrak{G}$, hence $P = \kappa_\zeta(\xi, \eta)$ with the common Cartesian co-ordinates $\zeta = (\xi, \eta)$ of P . The two, here constant base vectors ∂_1, ∂_2 of the tangent space and the inner product of the manifold define

$$g_{ij}(P) := (\partial_i, \partial_j)(P) \equiv \delta_{ij}$$

as the metric (or fundamental) tensor. This can be our steppingstone to conformal parametrization.

Suppose now an other function $\kappa_z : \mathfrak{D}_2 \subset \mathbb{R}^2 \rightarrow \mathfrak{G}$ such that the point $P = \kappa_z(x, y)$ is assigned arbitrary co-ordinates. Then $z = (\kappa_z^{-1} \circ \kappa_\zeta)(\xi, \eta)$ acts as a transformation of co-ordinates (as far as a diffeomorphism). Let now $f : \mathfrak{D} \rightarrow \mathfrak{E} \rightarrow \mathfrak{G}$ be the conformal map considered before, then we can identify the way $f^{-1} = \kappa_z^{-1} \circ \kappa_\zeta$ performs the transformation of co-ordinates. The remaining thing to do is clarifying the new shape of the fundamental tensor and other metric quantities. With

$$\bar{\partial}_i = \frac{\partial x_j}{\partial \bar{x}_i} \partial_j \quad \text{and} \quad \bar{g}_{ij}(P) = \frac{\partial x_k}{\partial \bar{x}_i} \frac{\partial x_l}{\partial \bar{x}_j} g_{kl}(P)$$

as two cogredient transformation laws it is quite easy to derive $\bar{g}_{11} = \bar{g}_{22} = \xi_x^2 + \eta_x^2 = |f'(z)|^2 = 1/|f^{-1}(\zeta)'|^2$ and $\bar{g}_{12} = \bar{g}_{21} \equiv 0$. Thus our base vectors are not constant anymore but whirling around from point to point, being stretched by the same factor $\sqrt{\bar{g}_{11}}$ (we suppress the bar for convenience) but keeping orthogonality.

What becomes evident now is, that with the (numeric) evaluation of $f'(z) = \xi_x + i\eta_x$ we can calculate the metric tensor in each grid point of the unit square and save it for further disposal. Same holds for the second (and higher order) derivatives which can easily be calculated and saved for a grid via $f''(z) = \xi_{xx} + i\eta_{xx}$.

One reason for focusing on second derivatives is the importance of second kind CHRISTOFFEL symbols [2]. With the setting $\chi(x, y) := \xi_x^2 + \eta_x^2$ one concludes

$$\Gamma_{11}^1 = \Gamma_{12}^2 = \Gamma_{21}^2 = \frac{\xi_x \xi_{xx} + \eta_x \eta_{xx}}{\chi},$$

and the rest of them following likewise, revealing only two of eight independent expressions [5, p.50ff.].

5.2 Differential operators

For the development of a simulation code involving PDEs it is now essential to formulate the underlying laws in co-ordinate free manner and then realize them in conformal co-ordinates, making use of the metrics. To be confusing, covariant differentiation of a tensor field can be used in such formulations: 'ordinary' partial derivatives which we treated with spatial approximation formulae applying Finite Difference schemes must be corrected by sums involving CHRISTOFFEL quantities, depending on their co- or contravariant nature we get

$$\frac{\partial \Phi_{j_1 \dots j_m}^{i_1 \dots i_n}}{\partial x_p} = \frac{\partial \Phi_{j_1 \dots j_m}^{i_1 \dots i_n}}{\partial x_p} + \sum_{h=1}^n \Gamma_{lp}^{i_h} \Phi_{j_1 \dots j_m}^{i_1 \dots i_{h-1} i_{h+1} \dots i_n} - \sum_{k=1}^m \Gamma_{jkp}^l \Phi_{j_1 \dots j_{k-1} j_{k+1} \dots j_m}^{i_1 \dots i_n}$$

and as an example, the LAPLACE operator for a scalar field in arbitrary co-ordinates will then be written³ as

$$\Delta u = g^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \left(\frac{\partial g^{ij}}{\partial x_j} + \frac{g^{ij}}{2g} \frac{\partial g}{\partial x_j} \right) \frac{\partial u}{\partial x_i}.$$

This means a quite nice generalization of the included Cartesian case, where $g_{ij} = \delta_{ij}$ yields the well-known expression involved in the heat equation. Moreover, inserting the derivative f' and doing a bit of calculation legitimates the replacement $\Delta u \rightarrow \Delta u / (\xi_x^2 + \eta_x^2)$ we used in our previous article (5, heat equation on disk).

We conclude that conformal parametrization of a given domain by the unit square results in a minimum of overhead with respect to curvilinear co-ordinates.

5.3 Boundary conditions

While DIRICHLET boundary conditions pose no further problems as the values of a state variable are prescribed, NEUMANN/NEWTON type conditions require the transformation of the outward pointing normal derivative. Here another benefit of the conformal map appears, the direction of such a vector always coincides with one of the local base vectors. Respecting orientation, we have only to correct its length and derive [5, p.53]

$$\nabla_n u = \pm \frac{1}{\sqrt{\xi_x^2 + \eta_x^2}} \frac{\partial u}{\partial x_i}.$$

With these results we are able to simulate time-dependent PDEs where state variables can even be tensor-valued. Do not forget to transform them by using the correct transformation laws in your implementation.

³Einstein summation, $g^{ik} g_{kj} = \delta_j^i$ and g denotes the determinant of g_{ij} .

Summary

In this contribution we focused on the numerical solution of THEODORSEN's nonlinear and singular integral equation to establish the function of boundary correspondence, $\theta(\varphi)$. With pre-given $\rho(\theta)$ the conformal map $f: \mathfrak{E} \rightarrow \mathfrak{G}$ can be constructed either by NEWTON iteration using Fourier expansion of $\ln \rho(\theta(\varphi))$ or - in case of increasing ε or vertex singularities - by means of fix point iteration, direct evaluation of the involved nonlinear and singular integral with successive application of SCHWARZ or CAUCHY formulae.

Neither way poses problems for an implementation in *Mathematica* and with the error estimates for $\theta(\varphi)$ on the boundary we control the overall error on the disk.

If trigonometric interpolation turns out to be too inaccurate as in case of present vertices (we considered the unit square), then the corner singularities have to be investigated and usage of more apt base functions can ship around the problem to find a more accurate approximation for the function of boundary correspondence.

We note that THEODORSEN's equation is only one of plenty approaches to numerically construct the map. If results are unsatisfactory concerning accuracy then heading out for other integral equations or approximation methods provided in literature is an idea.

In a bigger context, the obtained conformal map (and its derivatives) can be utilized in PDE-based simulations. This approach is convenient to minimize the complexity involved when introducing curvilinear coordinates. Moreover, starting with a co-ordinate free description, one is able to formulate the laws regarding metrics of the concrete conformal transformation.

From the co-ordinate point of view, the conformal metric of course alters the shape of the PDE and after all, co-ordinates are necessary for doing number crunching simulations while the underlying physical laws indeed stay invariant. We shall present such a practical simulation study in a subsequent contribution.

With this in mind, a numerical treatment of PDEs by using conformal maps combined with the CTDS method of lines gets less a matter of consistency than rather an issue of stability: let us therefore interpret the approximation error committed by the numerical determination of the function of boundary correspondence as a slight perturbation of the metric quantities. The smaller this error can be squeezed, the better will be the quality of out-coming results of our overall simulation.

References

- [1] Blümlinger, M. *Konforme Abbildungen von Gebieten mit Ecken*. TU Wien, Dipl.-Arb., 1984.
<http://katalog.ub.tuwien.ac.at/AC00227190>
- [2] Dirschmid, H.J. *Tensoren und Felder*. Springer-Verlag, Wien, 1996, ISBN 3-211-82754-4.
- [3] Gaier, D. *Konstruktive Methoden der konformen Abbildung*. Springer-Verlag, Berlin-Göttingen-Heidelberg, 1964.
- [4] Holzinger, M. *CTDS – Mauling Heat Equation on Unit Disk by Conformal Parametrization*. Simulation Notes Europe 2020; 30(2): 43-50. DOI: 10.11128/sne.30.tn.10511.
- [5] Holzinger, M. *Konforme Abbildungen zur Simulation von Modellen mit verteilten Parametern*. Dissertation/PhD (in German), Wien, 2020.
<http://katalog.ub.tuwien.ac.at/AC15652638>
- [6] Hübner, O. *The Newton method for solving the Theodorsen integral equation*. Journal of Computational and Applied Mathematics 14 (1986), pp. 19-30.
- [7] Kythe, P.K. *Computational Conformal Mapping*. Birkhäuser, Boston 1998, ISBN 0-8176-3996-9.
- [8] Richardson, S. *Integral Equations*. Mathematica Journal, Vol. 9, Issue 2, pp. 460-482, 2004.
<https://www.mathematica-journal.com/issue/v9i2/contents/IntegralEquations/IntegralEquations.pdf>
- [9] Wegmann, R. *Discretized versions of Newton type iterative methods for conformal mapping*. Journal of Computational and Applied Mathematics 14 (1986), pp. 207-224.

```

(* Discrete Newton Iteration According To Hübner/Wegmann;  $\eta$ -axis symmetry *)
Clear["Global`*"]; WD=FileNameTake[NotebookDirectory[],5];SetDirectory[WD];
nprec=nprecfourier=500; znp=4096; np=znp/2; niter=500;
(* load external geometry definition to make use of r[phi_] *)
Import[FileNameJoin[WD,"GLOBAL","Geo2_Definition_Fourierpoly.wl"]]
(* initialize phi_k und start-vector *)
phicore=Table[N[x,nprec],x,Pi/2,3Pi/2,Pi/np];
phia11=Table[N[x,nprec],x,0,2Pi-Pi/np,Pi/np]; svalues[i_]=phia11;
(* some terms can be calculated ex ante *)
sin2m = Chop[Table[Sin[2*m*phicore],m,1,np/2],10^-nprec]; (* rest analog *)
(* trigonometric interpolation polynomial *)
pt[t_,m_,y_]:= 1/Sqrt[Dimensions[y][[1]]] * ( Re[y[[1]]]+
  2*Sum[(Re[y[[k+1]])Cos[k*t]-Im[y[[k+1]])Sin[k*t]},{k,1,(m/2)-1}] +
  Re[y[[1 + Dimensions[y][[1]]/2]] Cos[(m/2)*t] );
(* make use of symmetries: calculate on [Pi/2,3Pi/2] extend to [0,2Pi] *)
oddeextension[vec_]:=Flatten[Append[Prepend[vec,Table[-vec[[np/2+1-k]],
  {k,0,np/2-1}]],Table[-vec[[np-k]},{k,0,np/2-2}]]];
evenextension[vec_]:=Flatten[Append[Prepend[vec,Table[vec[[np/2+1-k]],
  {k,0,np/2-1}]],Table[vec[[np-k]},{k,0,np/2-2}]]];
(* begin iteration loop *)
Do[ t0 = Timing[
lnRvalues=Log[r[svalues[i-1]]];sigmavalues=r'[svalues[i-1]]/r[svalues[i-1]];
Block[{$MinPrecision=nprecfourier,$MaxPrecision=nprecfourier},
  fklNR=Fourier[lnRvalues,FourierParameters->{0,-1}]];
lnRf[phi_] = Expand[pt[phi,Dimensions[fklNR][[1]],fklNR]];
anlnR=Append[Coefficient[lnRf[phi],Table[Cos[2m*phi],{m,1,np/2-1}],0];
bnlnR=Coefficient[lnRf[phi],Table[Sin[(2m-1)phi],{m,1,np/2}]];
(* calculate K[lnR]/K[v]/K[q] on [Pi/2,3Pi/2] for efficiency sake *)
klnRcore=Chop[Total[anlnR*sin2m-bnlnR*cos2mm1],10^-nprec];
klnRvalues=oddeextension[klnRcore]; fvalues=svalues[i-1]-phia11-klnRvalues;
vvals=ArcTan[sigmavalues]; rvalues=Sqrt[1+sigmavalues^2];
Block[{$MinPrecision=nprecfourier,$MaxPrecision=nprecfourier},
  fkv=Fourier[vvals,FourierParameters->{0,-1}]];
vfourier[phi_] = Expand[pt[phi,Dimensions[fkv][[1]],fkv]];
anv=Coefficient[vfourier[phi],Table[Cos[(2*m-1)phi],{m,1,np/2}]];
bnv=Append[Coefficient[vfourier[phi],Table[Sin[2*m*phi],{m,1,np/2-1}],0];
w=Chop[Total[anv*sin2mm1-bnv*cos2m],10^-nprec];wvalues=evenextension[w];
alpha=Chop[(1/znp)*Sum[vvals[[1]},{1,Dimensions[vvals][[1]]}],10^-nprec];
q=(fvalues*sigmavalues)/(rvalues*Exp[wvalues]);
Block[{$MinPrecision=nprecfourier,$MaxPrecision=nprecfourier},
  fkq=Fourier[qvalues,FourierParameters->{0,-1}]];
qfourier[phi_] = Expand[pt[phi,Dimensions[fkq][[1]],fkq]];
anq=Append[Coefficient[qfourier[phi],Table[Cos[2*m*phi],{m,1,np/2-1}],0];
bnq=Coefficient[qfourier[phi],Table[Sin[(2*m-1)*phi],{m,1,np/2}]];
pcore=Chop[Total[anq*sin2m-bnq*cos2mm1],10^-nprec];p=oddeextension[pcore];
qdach=Chop[(1/znp)*Sum[q[[1]},{1,Dimensions[q][[1]]}],10^-nprec];
svalues[i]=svalues[i-1]-(fvalues/(rvalues^2)) -
  ((p+qdach*Tan[alpha])*Exp[wvalues])/rvalues];];
t0 // Print, {i, 1, niter} ]; (* iteration step time, end iteration loop *)

```