

Research Article

Combining Rational Approximation with Asymptotic Wiener – Hopf Factorization Algorithm for Matrix Functions: Implementation and Testing

Rougerie T^{1*} and Kisil A²

¹ENSTA Paris Tech, Boulevard des Maréchaux, Palaiseau, France

²Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Centre for Mathematical Sciences, Wilberforce Rd, Cambridge CB3 0WA, UK

Abstract

This paper discusses how an asymptotic Wiener–Hopf factorization can be implemented for a wide class of functions. Asymptotic Wiener–Hopf factorization was discussed [19] and the convergence for matrices sufficiently close to the identity matrix is shown. We demonstrate how the algorithm can be successfully implemented with the help of the rational approximations. The idea is to simplify the matrix first by rationally approximating and then perform the approximate factorisation. There is no compromise in accuracy since the factorisation is approximate anyway and rational approximations are very precise usually. There is also a mapping of real line discussed and implemented to make the rational approximation more optimal. The code is tested against some easy examples which are calculated by hand. The use of this code is illustrated with some more complicated matrix functions motivated by applications. The method has been implemented for 2×2 and 4×4 matrices but can easy be adapted for any size matrix. The code will be made available with the publication of this paper. We note that to date there are very few implemented Wiener–Hopf factorisation available due to instabilities, so this paper will make an important contribution to this area.

Keywords: Wiener-Hopf; Asymptotic factorization; Riemann-Hilbert

Introduction

Many problems in mathematics, physics and engineering give rise to partial differential equations (PDEs); hence the solution of these problems relies on the ability to solve different types of PDEs. The study of PDEs is a vast and expanding area of research [1-15]. This paper concerns the Wiener–Hopf method, which is used for a broad collection of PDEs [16-20]. The Wiener–Hopf technique is an elegant method, which extends the separation of variables technique used to investigate PDEs. It enabled to solve PDEs even with sharp transitions in boundary conditions which arise in acoustic, finance, probability, hydrodynamic, elasticity, potential and electromagnetic theories [21-25].

The most challenging step of the method is Wiener-Hopf factorizations. There are fundamentally two different cases to consider: the scalar and the matrix Wiener-Hopf factorizations. For the scalar case the solution of Wiener-Hopf factorizations can be expressed in terms of a Cauchy type integral. Matrix Wiener-Hopf factorizations is a natural extension of the traditional scalar Wiener-Hopf factorizations. It allows to solve more advanced and realistic problems and is the field of current intensive research. Approximating matrix Wiener-Hopf equations is complex due to instabilities. This paper concerns reliable approximate matrix Wiener-Hopf factorizations and its implementation. The existence of matrix factorizations under general assumptions has been known for a long time [12,22]. Nevertheless, to date constructive matrix factorizations remains a formidable challenge. By constructive factorizations it is meant that there is an algorithmic way to obtain the factorizations. Due to its complexity, different classes of matrix functions have to be treated separately. Below we outline some of the few classes of matrix functions with known constructive factorizations and give references that an interested reader can follow. Constructive factorizations algorithms exist for classes of Veitch and Peake [25].

• Rational matrix functions. An algorithm for factorization has been known to exist for a long time. Nevertheless it has not been successfully implemented (as far as the authors know) due to instabilities.

- Upper or lower triangular matrix functions (with factorable diagonal elements). They can be reduced to a scalar equation.
- Functionally commuting matrix functions [18]. One of the reasons is that one could apply some of the techniques that have been developed in the scalar setting. For some conditions under which the commutative factorisation is possible [4].
- Piece-wise constant matrix functions [10].
- Self-adjoint matrices have been studied [23].
- Daniele-Khrapkov matrices [6,15].

What complicates the application of algorithms further is that, given a matrix, it is difficult to determine which class it belongs to, if any. Pre- and post-multiplication can transform the matrix to a right form but this is done mainly using *ad hoc* techniques. A systematic treatment of some transformations is given [8].

The next best option is approximate constructive factorisation which is what is mainly used in applications [6,17]. There are a number of approaches but they are usually tailored to a particular application and hard to generalize and make algorithmic. In other cases such as rational matrix functions there does exist an algorithm for Wiener– Hopf factorizations but it is not implemented due to instabilities. The

*Corresponding author: Rougerie T, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Centre for Mathematical Sciences, Wilberforce Rd, Cambridge CB3 0WA, UK, Tel: 33181871740; E-mail: a.kisil@maths.cam.ac.uk

Received October 31, 2017; Accepted November 30, 2017; Published December 01, 2017

Citation: Rougerie T, Kisil A (2017) Combining Rational Approximation with Asymptotic Wiener–Hopf Factorization Algorithm for Matrix Functions: Implementation and Testing. J Appl Computat Math 6: 374. doi: 10.4172/2168-9679.1000374

Copyright: © 2017 Rougerie T, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

I

authors are not aware of any implemented, freely available algorithms for Wiener–Hopf factorisation of matrix functions. This paper will be concerned with implementation of one type of approximate constructive factorisation called asymptotic factorisation [19], which is outlined below.

The structure of the paper is as follows. We outline the mathematical ideas in the proposed procedure. The next section comments on the implementation of the algorithm in MATLAB. The use of the code to compute examples starting with easy cases which can be verified by hand then by examples which arise in acoustics. We end with a few conclusive remarks.

Procedure for Approximate Wiener-Hopf Factorizations

There are a few steps in the implemented approximate Wiener-Hopf factorizations, which are described below.

Asymptotic Wiener-Hopf factorization

Let *M* be a matrix of the following form:

$$M(x) = (m_{ij}(x))_{i, j \in \{1, 2\}},$$
(1)

where the $m_{ij}(x)$ are functions analytic around the real axis R. Let *I* be the identity matrix and $\varepsilon \in [0,1]$. We can write

$$M(x) = I + \varepsilon G(x), \tag{2}$$

the asymptotic algorithm produces a good approximate Wiener-Hopf factorization if $\varepsilon G(x)$ is small compared to *I*. This will be discussed in numerical examples.

We are going to find step by step approximate Wiener-Hopf multiplicative factorizations (we will only consider the case of zero partial indices), we aim to find $M^-(x) M^+(x)$ where:

 $M(x)=M^{-}(x)M^{+}(x),$

on the real line and in a superscript+(or -) indicates that the function is *analytic in the upper (or lower) half-plane*. We will look for an approximate factorization of the form [19]:

$$I + \in G \approx \left(I + \sum_{i \in \mathbb{N}^*} \epsilon^i \ G_{i-1}^-\right) \left(I + \sum_{i \in \mathbb{N}^*} \epsilon^i \ G_{i-1}^+\right) \tag{3}$$

We define the error Δ_j at a step *j* as the term neglected in the approximation:

$$\Delta j = I + \varepsilon G - S_j^- S_j^+ \text{ where } S_j^- = I + \sum_{i=1}^{j} \epsilon^i G_{i-1}^- \text{ and } S_j^+ = I + \sum_{i=1}^{j} \epsilon^i G_{i-1}^+$$
(4)

We will describe here the first four steps that will be used in the code. It would be trivial to include more steps in the code if there was a need for it.

For the first step, we are looking for G_0^- and G_0^+ respectively analytic on the lower half-plane and the upper half-plane, so that

$$\begin{split} I + \varepsilon G &\approx \left(I + \varepsilon G_0^- \right) \left(I + \varepsilon G_0^+ \right) \\ &= I + \varepsilon \left(G_0^- + G_0^+ \right) + \varepsilon^2 \left(G_0^- G_0^+ \right) \end{split}$$

Hence, at the first order of ε , we have the Wiener-Hopf additive splitting:

$$G = G_0^- + G_0^+ . (5)$$

The error is

$$\Delta_{1} = (I + \in G) - (I + \in G_{0}^{-})(I + G_{0}^{+}) = \in^{2} G_{0}^{-} G_{0}^{+}$$
(6)

For the second order of ε :

$$+\varepsilon G \approx \left(I + \varepsilon G_0^- + \varepsilon^2 G_1^-\right) \left(I + \varepsilon G_0^+ + \varepsilon^2 G_1^+\right)$$

$$I = \left(G_0^- - G_0^+\right) = \left(G_0^- - G_0^+ - G_0^- G_0^+\right) = \left(G_0^- G_0^+ - G_0^- G_0^+ - G_0^- G_0^+\right) = \left(G_0^- G_0^+ - G_0^- G_0^+ - G_0^- G_0^+\right) = \left(G_0^- G_0^+ - G_0^- G_0^+ - G_0^- G_0^+\right) = \left(G_0^- G_0^+ - G_0^- G_0^+\right) = \left(G_0^- G_0^+ - G_0^- G_0^- - G_0^- G_0^- - G_0^- G_0^- - G_0^- - G_0^- - G_0^- G_0^- - G_0^- G_0^- - G_0^- G_0^- - G$$

Page 2 of 8

$$= I + \varepsilon \left(G_0^- + G_0^+ \right) + \varepsilon^2 \left(G_1^- + G_1^+ + G_0^- G_0^+ \right) + \varepsilon^3 \left(G_1^- G_0^+ + G_0^- G_1^+ \right) + \varepsilon^4 G_1^- G_0^- G_0^+$$

where the term in ε^2 should be equal to zero.

Therefore, we can deduce $G_1^- G_1^+$ and thus Δ_2 :

$$G_{1} = G_{1}^{-} + G_{1}^{+} = -G_{0}^{-}G_{1}^{+}, \ \Delta_{2} = \epsilon^{3} \left(G_{1}^{-}G_{0}^{+}\right) + \epsilon^{4} G_{1}^{-}G_{1}^{+}.$$
⁽⁷⁾

Recursively, for the third and fourth orders of ε , we have

 $\begin{aligned} G_{2} &= G_{2}^{-} + G_{2}^{+} = -\left(G_{0}^{-}G_{1}^{+} + G_{1}^{-}G_{0}^{+}\right), \\ \Delta_{3} &= e^{4}\left(G_{0}^{-}G_{2}^{+} + G_{2}^{-}G_{0}^{+} + G_{1}^{-}G_{1}^{+}\right) + e^{5}\left(G_{1}^{-}G_{2}^{+} + G_{2}^{-}G_{1}^{+}\right) + e^{6}\left(G_{2}^{-}G_{2}^{+}\right) \\ G_{3} &= G_{3}^{-} + G_{3}^{+} = -\left(G_{0}^{-}G_{2}^{+} + G_{2}^{-}G_{0}^{+} + G_{1}^{-}G_{1}^{+}\right), \\ \Delta_{4} &= e^{5}\left(G_{0}^{-}G_{1}^{+} + G_{3}^{-}G_{0}^{+} + G_{1}^{-}G_{2}^{+} + G_{2}^{-}G_{1}^{+}\right) + e^{6}\left(G_{1}^{-}G_{3}^{+} + G_{2}^{-}G_{2}^{+}\right) + e^{7}\left(G_{2}^{-}G_{3}^{+} + G_{1}^{-}G_{2}^{-}\right) + e^{8}\left(G_{3}^{-}G_{3}^{+}\right) \end{aligned}$

There are two issues which need to be addressed for the approximate

procedure to be successful. The first is the question of convergence i.e., if the error decreases at every step. This has been discussed [19] and the general criterion formulated, but this criteria is not easy to verify. In practice, when using this procedure this is easily verified by displaying the Δ_i . Secondly, the above procedure is designed for matrices which have zero partial indices. It has been modified for partial indices differing by at most one [20]. So in order to apply it we first want to determine the partial indices. This is in general a difficult issue. Fortunately, $(I+\varepsilon G(\alpha))$ and $\varepsilon G(\alpha)$ is small enough the partial indices are zero.

Rational matrix functions

One could find G_0^- and G_0^+ using theorem B, but the repeated evaluation of Cauchy type integral would cause computational problems. Instead we first implement the rational approximation and then split the matrices at each step. The splitting/factorization of the rational matrix function is then straightforward as is described below. When *G* is in rational form, we split the poles of the lower and the upper half-planes of each rational function in order to find G_0^- and G_0^+ . This is implemented in **split add.m.**

Let *M* be a rational matrix function, represented by

$$M(x) = \begin{pmatrix} \frac{p_{11}(x)}{q_{11}(x)} & \frac{p_{12}(x)}{q_{12}(x)} \\ \frac{p_{21}(x)}{q_{21}(x)} & \frac{p_{22}(x)}{q_{22}(x)} \end{pmatrix}$$
(9)

where p_{ij} and q_{ij} are polynomial functions, q_{ij} not identically zero on the domain, for any $i, j \in \{1, 2\}$.

Using a partial fraction decomposition, we can write (in the case of no repeated roots, otherwise can be easily adapted), for any $i,j \in \{1,2\}$:

$$\frac{p_{ij}(x)}{q_{ij}(x)} = k(x) + \sum_{m} \frac{r_{ij}^{m}}{x - d_{ij}^{m}}$$
(10)

where d_{ij}^m are the m poles of this fraction, $r_{ij}^m(x)$ the corresponding residues, and k(x) is the direct term.

We assume that none of the poles are on the real axis. If at least one of them is, we can rotate the x-axis using $x \mapsto \exp(i\alpha x)$, with $\alpha \in (0, \pi/4]$ and find the factorisation with respect to the new x-axis. We can split the sum between the poles of the lower and the upper half-planes:

$$\frac{p_{ij}(x)}{q_{ij}(x)} = k(x) + \sum_{m \in I_{-}} \frac{r_{ij}^{m}}{x - d_{ij}^{m}} + \sum_{m \in I_{+}} \frac{r_{ij}^{m}}{x - d_{ij}^{m}}$$

$$= \frac{p_{i}^{-}(x)}{q_{ij}^{-}(x)} + \frac{p_{ij}^{+}(x)}{q_{ij}^{+}(x)}$$
(11)

where

$$I_{-} = \left\{ m/Im \left(d_{ij}^{m} \right) < 0 \right\}, \qquad I_{+} = \left\{ m/Im \left(d_{ij}^{m} \right) > 0 \right\},$$
(12) and

$$\frac{p_{ij}^{-}(x)}{q_{ij}^{-}(x)} = \frac{1}{2}k(x) + \sum_{m \in I_{-}} \frac{r_{ij}^{m}}{x - d_{ij}^{m}} , \quad \frac{p_{ij}^{+}(x)}{q_{ij}^{+}(x)} = \frac{1}{2}k(x) + \sum_{m \in I_{+}} \frac{r_{ij}^{m}}{x - d_{ij}^{m}}.$$
(13)

We finally have

$$M_{0}^{-} = \left(\frac{p_{y}^{-}(x)}{q_{y}^{-}(x)}\right)_{i,j\in\{1,2\}} , M_{0}^{+} = \left(\frac{p_{y}^{+}(x)}{q_{y}^{+}(x)}\right)_{i,j\in\{1,2\}}$$
(14)

All the G_i , i>0 will be rational matrix functions, as sums and products of rational matrix functions G_j , j < i. Thus the additive splitting of G_i will be done as explained above.

Mapping of the real axis for even functions

The rational approximation of even function over R can be improved by employing transformations. We can map the domain by a succession of two conformal mapping in order to obtain better rational approximations. These choices of maps are based on [16]. This is implemented in **map.m** which is used in **rat approx. m**. If a non-even function was to be studied in our algorithm, the rational approximation would be performed without the mapping in **rat approx. noteven.m**.

In this section, for any even function *G* of the variable *y*, analytic on \mathbb{R} , we will write *G*(*z*) and **G**(*x*) respectively for its mapped version on the unit circle and on the interval [-1,1].

The first step is the mapping of the real line into the unit circle, using the conformal M obvious map:

$$M: y \mapsto \frac{-1+iy}{1+iy}, \quad M^{-1}: z \mapsto \frac{i(1+z)}{-1+z}$$
(15)

This map does the transformation $1 \mapsto i, -1 \mapsto -i, \infty \mapsto 1$.

We then apply a Joukowski map to project the unit circle to the interval [-1,1]:

$$J: z \mapsto \frac{1}{2} \left(z + z^{-1} \right). \tag{16}$$

Here, only half of the circle is projected on the interval [-1,1]. The function being even, the other half is the same.

The two maps are composed together:

$$M: y \mapsto \frac{y^2 - 1}{y^2 + 1} , \quad M^{-1} J^{-1}: x \mapsto \sqrt{\frac{1 + x}{1 - x^2}}$$
 (17)

Hence, G(y) can be mapped into $\mathbf{G}(x)$ with

$$\mathbf{G}(x) = G(M^{-1}J^{-1}(x)) \tag{18}$$

and unmapped using

$$G(y) = \mathbf{G}(JM(y)). \tag{19}$$

This transformation from **G** to *G* by unmapping [-1,1] to the realaxis keeps the rational property of the function (performed in **map roots**).

MATLAB Implementation of the Algorithm

File main.m.

We implemented an algorithm in MATLAB in order to compute the different steps of the approximation. The file **main.m** has six phases:

• Importing **data X.m**, and computing a rational approximation if the data is composed of non-rational functions,

Step 1: Computing
$$G = G_0^- + G_0^+$$

Step 2: Computing $G_1 = -(G_0^- G_0^+) = G_1^- + G_1^+$,
Step 3: Computing $G_2 = -(G_0^- G_1^+ + G_1^- G_0^+) = G_2^- + G_2^+$,
Step 4: Computing $G_3 = -(G_0^- G_2^+ + G_2^- G_0^+ + G_1^- G_1^+) = G_3^- + G_3^+$,

Plotting the results and checking that $i \mapsto \Delta_i$ is decreasing and small enough. The rational Wiener–Hopf factors are computed.

A schematics of the code is given in Figure 1. For more details see **read me.txt** file in the code.



The Chebfun library

We used the open-source *Chebfun* MATLAB library to compute a rational approximation of non-rational functions. The approximation of the $g_{ij}(x)$, $i,j \in \{1,2\}$ as rational functions is done by the *chebpade* function of *Chebfun* which consists in a Chebyshev-Pade approximation: *chebpade* $(g_{ij}(x), d_1, d_2)$ returns two polynomials $p_{ij}(x)$ and $q_{ij}(x)$ of degrees respectively d_1 and d_2 such that $g_{ij}(x) \approx \frac{p_{ij}(x)}{q_{ij}(x)}$. Choosing $d_{1a}d_2 \leq 8$ usually gives some good results. When unmapped, the half of the even function becomes entire so that the resulting polynomial functions will have a degree of ≤ 16 .

In practice, we first create a *chebfun* function on a [-1,1] interval for each $\mathbf{G}_{ij}(x)$, $i,j \in \{1,2\}$ where **G** is the mapped version of **G**. In our algorithm, we chose a [-1,0.9] interval rather than [-1,1], in order to map the function without including all the values of the function to $+\infty$.

Numerical Examples

Rational matrix function with complex poles

The first example (data A.m in the code) is a rational function given by

$$\mathbf{I} + \in \mathbf{A}(y) = \mathbf{I} + \in \begin{pmatrix} 0 & \frac{-2y^2}{y^2 + 4} \\ \frac{-8}{y^2 + 4} & 0 \end{pmatrix}$$
(20)

We derive by hand the approximate factorisation where ε =1 in order to compare it to the code output. The partial fraction decomposition gives:

$$A(y) = A_0^{-}(y) + A_0^{+}(y)$$
(21)

with

$$A_0^{-}(y) = \frac{1}{y + 2iu} \begin{pmatrix} 0 & -y \\ -2i & 0 \end{pmatrix}, \quad A_0^{+}(y) = \frac{1}{y - 2iu} \begin{pmatrix} 0 & -y \\ 2i & 0 \end{pmatrix} \quad (22)$$

At this point, the part neglected equals

$$\Delta_{1} = A_{0}^{-}(y) A_{0}^{+}(y) = \frac{2iy}{y^{2} + 4} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
(23)

Then, at the second step,

$$A_{1}(y) = -A_{0}^{-}(y)A_{0}^{+}(y) = \frac{2iy}{y^{2}+4} \begin{pmatrix} 1 & 0\\ 0 & -1 \end{pmatrix}$$
(24)

Another partial fraction decomposition on G_1 leads to:

$$A_{1}^{-}(y) = \frac{i}{y+2i} \begin{pmatrix} 1 & 0\\ 0 & -1 \end{pmatrix} , A_{1}^{+}(y) = \frac{i}{y-2i} \begin{pmatrix} 1 & 0\\ 0 & -1 \end{pmatrix}$$
(25)

According to eqn. (7), the error is

$$\Delta_2 = \frac{1}{y^2 + 4} \begin{pmatrix} -1 & 0\\ 4 & -1 \end{pmatrix}$$
(26)

the third order of ε gives:

$$A_{2}(y) = -(A_{0}^{-}(y)A_{1}^{+}(y) + A_{1}^{-}(y)A_{0}^{+}(y)) = \frac{1}{y^{2} + 4} \begin{pmatrix} 0 & 0 \\ -4 & 0 \end{pmatrix}$$
(27)

and

$$A_{2}^{-}(y) = \frac{1}{y+2i} \begin{pmatrix} 0 & 0 \\ -i & 0 \end{pmatrix}, A_{2}^{+}(y) = \frac{1}{y-2i} \begin{pmatrix} 0 & 0 \\ i & 0 \end{pmatrix}$$
(28)

Here, the term neglected was, as in (8)

$$\Delta_3 = \frac{1}{y^2 + 4} \begin{pmatrix} -iy - 1 & 0\\ 2 & iy - 1 \end{pmatrix}.$$
 (29)

For the fourth step:

$$A_{3(y)} = \frac{1}{y^2 + 4} \begin{pmatrix} 1+i & 0\\ 0 & 1-i \end{pmatrix},$$
(30)

and

$$A_{3}^{-}(y) = \frac{1}{y+2i} \begin{pmatrix} 0.75i & 0\\ 0 & -0.25i \end{pmatrix}, A_{3}^{+}(y) = \frac{1}{y-2i} \begin{pmatrix} 0.25i & 0\\ 0 & -0.75i \end{pmatrix}$$
(31)

In the end, the error equals:

$$\Delta_4 = \frac{1}{y^2 + 4} \begin{pmatrix} -\frac{19}{16} & 0\\ 3 & -\frac{19}{16} \end{pmatrix}.$$
 (32)

The results given by the algorithm code match these derivations with an error of 10^{-16} due to MATLAB approximations.

We expect the error $\varepsilon \mapsto \Delta_i(\varepsilon)$ for a given step *i* to be decreasing when εA is small compared to I ([?art1] Mishuris & Rogosin 2014). We check this property for $\varepsilon = 1$, $\varepsilon = 0.5$, $\varepsilon = 0.05$ and $\varepsilon = 0.005$. We find that as long as εA is less than 0.8–0.9 in all elements, convergence occurs. Otherwise $\Delta_i(\varepsilon)$ does not decrease and can even increase. Figure 2 shows that the method of our algorithm works much better for εA small compared to I: the error is decreasing much faster, hence less steps are required to obtain a good approximation.

Rational matrix functions witTh poles on tThe real axis

The second example (**data B.m** in the code) is a rational function defined by:

$$\mathbf{I} + \epsilon \mathbf{B}(y) = \mathbf{I} + \epsilon \begin{pmatrix} 0 & \frac{-2y^2}{y^2 + 4} \\ \frac{-8}{y^2 + 4} & 0 \end{pmatrix}.$$
 (33)

The poles are y=2 and y=-2 so the x-axis can be rotated, using $y \rightarrow e^{ik}y$ (**rot.m**), where $k \in [0,2\pi]$ is a parameter of the algorithm. This has the effect of making sure that y=2 is included in the poles in the lower half plane and y=-2 with the poles in the upper half plane. Hence

$$\mathbf{B}(y) = \begin{pmatrix} 0 & \frac{-2y^2 e^{2ik}}{y^2 e^{2ik} - 4} \\ \frac{-8}{y^2 e^{2ik} - 4} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{-2y^2}{y^2 - 4e^{-2ik}} \\ \frac{-8e^{-2ik}}{y^2 - 4e^{-2ik}} & 0 \end{pmatrix}, (34)$$

and $y=2e^{-ik}$ and $y=-2e^{-ik}$ are the new poles. The complex plane can now be split in two halves, depending of the sign of the imaginary part of each poles. Again we calculate the decomposition by hand and compare them to the code output. The agreement is again of order of magnitude 10^{-16} .

The study of $\mathbf{I}+\varepsilon \mathbf{B}$, $\varepsilon=1$ gives some poor results because **B** is larger in magnitude than **I**. Even at the fourth step, $|\Delta_4|$ has a magnitude of more than 0.7. This huge error gives a bad approximation of $\mathbf{I}+\varepsilon \mathbf{B}$ (Figure 3).

For ε =0.005, the algorithm gives a very good approximation because $\varepsilon \mathbf{B}$ is now small compared to **I**. At the fourth step, $|\Delta_4|$ has a very small magnitude of 10⁻⁷. The approximation of *I*+ $\varepsilon \mathbf{B}$ is now very close to the actual **I**+ $\varepsilon \mathbf{B}$ (Figure 4).

Page 5 of 8





A more general rational matrix functions

We tested the algorithm on a third example (**data C.m** in the code) without any symmetry:

$$\mathbf{C}(y) = \begin{pmatrix} \frac{1}{y+i} & \frac{1}{y-2i} \\ \frac{1}{y-3i} & \frac{1}{y+2i} \end{pmatrix}.$$
 (35)

In this case again, the error $i \mapsto \Delta_i$ is not always decreasing (Figure 5), for example for $\varepsilon = 1$, but once the perturbation matrix is small enough the expected decrease is achieved.

Non rational matrix functions

This example (data G.m in the code) is motivated by acoustics, the



Wiener–Hopf problem arising when the plane wave is incident at an angle $\pi/2$ onto a periodic grating of length 2*a* which is spaced 2*b* apart [9]. The matrix to factorize is

$$\mathbf{S} = \mathbf{I} + \frac{1}{2\cosh\left((a+d)\gamma\right)} \begin{pmatrix} -\cosh\left((a-b)\gamma\right) & \frac{\gamma}{\alpha-1}\sinh\left((a-b)\gamma\right) \\ -\frac{\gamma}{\alpha+1}\sinh\left((a-b)\gamma\right) & \cosh\left((a-b)\gamma\right) \end{pmatrix}$$

where $\gamma = \sqrt{\alpha^2 - 1}$. We will find the rational approximation

$$\tilde{\mathbf{K}}(\alpha) = \mathbf{I} + \begin{pmatrix} -\frac{p_n}{q_n} & \frac{1}{\alpha - 1} \frac{l_n}{m_n} \\ -\frac{1}{\alpha + 1} \frac{l_n}{m_n} & \frac{p_n}{q_n} \end{pmatrix}$$



Considering the rational approximation directly gives poor results. Instead, we will calculate a rational approximation after a conformal mapping.

functions can be approximated and the error is less than 10⁻⁶. The poles lie where it is desired on the rotated imaginary axis and rotated real axis, Figure 6. The approximations are shown for the first step in Figure 7 and the error is large in the non-diagonal elements. The forth step Figure 8, shows that the error is very small for all entries of the matrix.

For parameters a=2.5, b=1.5 and rotate the line by exp(-0.16i), the

Page 7 of 8





Conclusions

This paper presents an implementation of asymptotic factorization for matrix functions of the form I+G. The implementation relies on rational approximation.

It is shown that this procedure converges for a number of examples as long as the elements of **G** are smaller than 0.8–1. Typically values around 0.7 or smaller give good results in all cases. In order to make this procedure applicable to a wider class it is possible to consider suitable pre- and post-multiplications [8]. For example, **S** can be far from **I** but once we transform $\mathbf{F}^-\mathbf{SK}^+$, it can be close to **I** matrix. The code is currently written for 2×2 and 4×4 matrices but can easily be extended to deal with other dimensions.

Authors Contributions

This paper resulted from a summer project of Tiphaine Rougerie under the supervision of Anastasia Kisil. The code was mostly written by Tiphaine based on the earlier code of Anastasia.

Acknowledgements

Anastasia Kisil benefited from useful discussions with Professors Abrahams,

Mishuris and Rogosin. Anastasia Kisil acknowledges support from Sultan Qaboos Research Fellowship at Corpus Christi College at University of Cambridge.

References

- Abrahams ID, Davis AMJ, Smith SGL (2008) Matrix WienerHopf approximation for a partially clamped plate. The Quarterly Journal of Mechanics and Applied Mathematics 61: 241-265.
- Antipov YA (2014) The Baker-Akhiezer function and factorization of the Chebotarev-Khrapkov matrix. Letters in Mathematical Physics 104:1365-1384.
- Doubravsky EM, Shanin AV(2012) Criteria for commutative factorization of a class of algebraic matrices.
- Ayton LJ, Peake N (2013) On high-frequency noise scattering by aero foils in flow. Journal of Fluid Mechanics 734: 144-182.
- Daniele VG, Zich RS (2014) The Wiener-Hopf method in electromagnetics. Mario Boella Series on Electromagnetism in Information and Communication Series, Institution of Engineering and Technology, SciTech Publishing.
- Daniele V, Lombardi G (2011) The Wiener-Hopf solution of the isotropic penetrable wedge problem: diffraction and total field. IEEE Transactions on Antennas and Propagation 59: 3797-3818.
- 7. Ehrhardt T, Speck FO (2002) Transformation techniques towards the

factorization of non-rational 2 \times 2 matrix functions. Linear Algebra and Its Applications 353: 53-90.

- Erbas Barıs, Abrahams ID (2007) Scattering of sound waves by an infinite grating composed of rigid plates. Wave Motion 44 :282-303.
- Erkhardt T, Spitkovskii IM (2001) Factorization of piecewise-constant matrix functions, and systems of linear differential equations. Algebra i Analiz 13: 56-123.
- Gohberg I, Kaashoek MA, Spitkovsky IM (2003) An overview of matrix factorization theory and operator applications. Factorization and integrable systems, pp. 1-102.
- Gohberg IC, Krein MG (1960) Systems of integral equations on a half line with kernels depending on the difference of arguments. American Mathematical Society Translations 2: 217-287.
- Green R, Fusai G, Abrahams ID (2010) The Wiener-Hopf technique and discretely monitored path-dependent option pricing. Mathematical Finance 20: 259-288.
- Jaworski JW, Peake N (2013) Aerodynamic noise from a poroelastic edge with implications for the silent flight of owls. Journal of Fluid Mechanics 723: 456-479.
- 14. Khrapkov AA (1971) The first basic problem for a notch at the apex of an infinite wedge, International Journal Fracture Mechanics 7: 373-382.
- 15. Kisil AV (2013) A constructive method for an approximate solution to scalar Wiener-Hopf equations. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, p: 469
- Lawrie JB, Abrahams ID (2007) A brief historical perspective of the Wiener-Hopf technique. Journal of Engineering Mathematics 59 : 351-358.

17. Litvinchuk GS, Spitkovskii IM (1987) Factorization of measurable matrix functions. Operator Theory: Advances and Applications 25.

Page 8 of 8

- Mishuris G, Rogosin S (2014) An asymptotic method of factorization of a class of matrix functions. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 470: 2166.
- Mishuris G, Rogosin S (2016) Factorization of a class of matrix-functions with stable partial indices. Mathematical Methods in the Applied Sciences 39: 3791-3807.
- Mishuris GS, Movchan AB, Slepyan LI (2008) Dynamics of a bridged crack in a discrete lattice, The Quarterly Journal of Mechanics and Applied Mathematics. 61: 151-160.
- 21. Muskhelishvili NI (1953) Singular integral equations. Boundary problems of function theory and their application to mathematical physics.
- 22. Nikolaicuk AM, Spitkovskil IM (1975) Factorization of Hermitian matrixvalued functions, and its applications to boundary value problems. Ukraine Mathematics 27 : 767-779.
- Noble B (1958) Methods based on the Wiener-Hopf technique for the solution of partial differential equations. International Series of Monographs on Pure and Applied Mathematics 7.
- Rogosin S, Mishuris G (2016) Constructive methods for factorization of matrixfunctions. IMA Journal of Applied Mathematics 81: 365-391.
- 25. Veitch B, Peake N (2008) Acoustic propagation and scattering in the exhaust flow from coaxial cylinders. Journal of Fluid Mechanics 613: 275-307.