

SOFTWARE METAPAPER

Matlab Code for the Discrete Hankel Transform

Ugo Chouinard and Natalie Baddour

Department of Mechanical Engineering, University of Ottawa, Ottawa, K1N 6N5, Canada
Corresponding author: Natalie Baddour (nbaddour@uottawa.ca)

Previous definitions of a Discrete Hankel Transform (DHT) have focused on methods to approximate the continuous Hankel integral transform without regard for the properties of the DHT itself. Recently, the theory of a Discrete Hankel Transform was proposed that follows the same path as the Discrete Fourier/Continuous Fourier transform. This DHT possesses orthogonality properties which lead to invertibility and also possesses the standard set of discrete shift, modulation, multiplication and convolution rules. The proposed DHT can be used to approximate the continuous forward and inverse Hankel transform. This paper describes the matlab code developed for the numerical calculation of this DHT.

Keywords: Hankel Transform; discrete; DFT; DHT; discrete transform

Funding statement: This research was financially supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

(1) Overview

Introduction

There have been many attempts to define a Discrete Hankel Transform (DHT) in the literature, however prior work has focused on proposing methods to approximate the calculation of the continuous Hankel integral, for example as given in [1, 2]. This stands in stark contrast to the approach taken with the Fourier transform where the Discrete Fourier Transform (DFT) is a transform in its own right, with its own mathematical theory of the manipulated quantities. An additional feature of a carefully derived DFT is that it can be used to approximate the continuous Fourier transform, with relevant sampling and interpolation theories that can be used. Recently, a DHT was proposed as a complete and orthogonal transform that possesses its own mathematical theory, including the standard set of shift, modulation, multiplication and convolution rules [3]. In addition, this DHT can be used to approximate the continuous Hankel transform in the same manner that the Discrete Fourier transform is known to be able to approximate the continuous Fourier transform.

Overview of the Discrete Hankel Transform

The Continuous Hankel Transform

The forward Hankel transform of order n transforms a function $f(r)$ in the spatial domain to a function $F(\rho)$ in the spatial frequency domain and is given by [4, p. 5.6]

$$\mathbb{H}_n\{f(r)\} = F_n(\rho) = \int_0^\infty f(r) r J_n(\rho r) dr \quad (1)$$

where $J_n(z)$ is the n th order Bessel function of the first kind. The inverse transform is given by

$$\mathbb{H}_n^{-1}\{F_n(\rho)\} = f(r) = \int_0^\infty \rho F_n(\rho) J_n(\rho r) d\rho \quad (2)$$

More on the continuous transform can be found in [4].

Discrete Hankel Transform

The n th order discrete Hankel transform (DHT) proposed in [3] is defined as the transformation of the discrete vector \mathbf{f} to vector \mathbf{F} given by

$$\mathbf{F} = \mathbf{Y}^{nN} \mathbf{f} \quad (3)$$

This discrete transform consists of taking an $N-1$ vector \mathbf{f} and a $(N-1) \times (N-1)$ square matrix of Hankel order n , \mathbf{Y}^{nN} , to perform the matrix-vector multiplication and obtain the $N-1$ DHT vector \mathbf{F} . If the DHT as defined in (3) is used to approximate the CHT, then the vector \mathbf{f} represents the sampled function to be transformed and the vector \mathbf{F} represents the discrete function in the transformed (Hankel) domain. The \mathbf{Y}^{nN} matrix in equation (3) is defined as having the m, k th entry given by

$$Y_{m,k}^{nN} = \frac{2}{j_{nN} J_{n+1}^2(j_{nk})} J_n\left(\frac{j_{nm} j_{nk}}{j_{nN}}\right) \quad 1 \leq m, k \leq N-1 \quad (4)$$

where j_{nk} is the k th zero of the Bessel function of the first kind of order n [3]. Properties of the DHT as defined in equation (3) are shown in [3].

Since the core of the tested discrete transform is the transformation matrix \mathbf{Y}^{nN} , various properties have to be maintained. One of these properties is that the matrix \mathbf{Y}^{nN}

possesses orthogonality properties, where $Y^{nN}Y^{nN} = I$. In order to preserve the requisite properties of Y^{nN} and therefore of the DHT itself, the first Bessel zero used in computing the entries of the Y^{nN} matrix is the first non-zero value of the Bessel zero of order n . If the Y^{nN} matrix is not assembled following this rule, the matrix loses its orthogonality property and thus performing the discrete transform leads to improper results. If the DHT is used to approximate a CHT, then this restriction also applies to the discretization of the continuous function, as shall be discussed further below.

The inverse discrete Hankel transform \mathbf{f} of the vector \mathbf{F} is then given by

$$\mathbf{F} = Y^{nN} \mathbf{f} \quad (5)$$

The discrete forward and inverse Hankel transforms as given in equations (3) and (5) have been shown to possess the standard set of shift, modulation, multiplication and convolution rules. In addition, this DHT can be used to approximate the continuous Hankel transform in the same manner that the Discrete Fourier transform is known to be able to approximate the continuous Fourier transform at certain discrete points.

Discrete Hankel Transform to approximate the Continuous Hankel Transform

Given a continuous function $f(r)$ evaluated at the discrete points r_{nk} in the space domain ($1 \leq k \leq N-1$), its n th order Hankel-transform function $F(\rho)$ evaluated at the discrete points ρ_{nm} ($1 \leq m \leq N-1$), can be approximately given by [3]

$$F[m] = \alpha \sum_{k=1}^{N-1} Y_{m,k}^{nN} f[k] \Rightarrow \mathbf{F} = \alpha Y^{nN} \mathbf{f} \quad (6)$$

where α is a scaling factor to be discussed below, and $F[m] = F(\rho_{nm})$, $f[k] = f(r_{nk})$.

Conversely, given a continuous function $F(\rho)$ evaluated at the discrete points ρ_{nm} in the frequency domain ($1 \leq m \leq N-1$), its n th order inverse Hankel transform function $f(r)$ evaluated at the discrete points r_{nk} ($1 \leq k \leq N-1$), can be approximately given by

$$f[k] = \frac{1}{\alpha} \sum_{m=1}^{N-1} Y_{m,k}^{nN} F[m] \Rightarrow \mathbf{f} = \frac{1}{\alpha} Y^{nN} \mathbf{F} \quad (7)$$

For both the forward and inverse transforms, α is a scaling factor which depends on the function properties and shall be discussed further below. The choice of discretization points r_{nk} and ρ_{nm} is also discussed below. The full theory of the discrete Hankel transform is given in [3].

Discretization Points

In order to properly use the discrete transform to approximate the continuous transform, a function has to be discretized at specific sampling points. For a finite spatial range $[0, R]$ and a Hankel transform of order n , these sampling points are given in the space domain as

$$r_{nk} = \frac{j_{nk}}{j_{nN}} R \quad \text{for } 1 \leq k \leq N-1 \quad (8)$$

For the finite frequency domain range $[0, W_\rho]$ and a Hankel transform of order n , the sampling points are given by

$$\rho_{nm} = \frac{j_{nm}}{j_{nN}} W_\rho \quad \text{for } 1 \leq m \leq N-1 \quad (9)$$

It is important to note that as in the case of the computation of the transformation matrix Y^{nN} , the first Bessel zero j_{n1} used in computing the discretization points is the first non-zero value.

The relationship $W_\rho = \frac{j_{nN}}{R}$, derived in [3], holds between the ranges in space and frequency. Choosing N determines the dimension (size) of the DHT and determines j_{nN} . The determination of j_{nN} (via choosing N) determines the range in one domain once the range in the other domain is chosen. In fact, any two of R , W_ρ , j_{nN} can be chosen but the third must follow from $W_\rho R = j_{nN}$. A similar relationship applies when using the Discrete Fourier Transform, any two of the range in each domain and the size of the DFT can be chosen independently.

Scaling Factor

The scaling factor α necessary for using the DHT to approximate the CHT depends on whether the function is space-limited or band-limited. Since it might be hard to determine if a function is space or band limited, the concept of effective limit is introduced. Therefore, a function defined as being "effectively limited in space by R " means that if $r > R$, then as $r \rightarrow \infty$, $f(r) \rightarrow 0$. In other words, the function can be made as close to zero as desired by selecting an R that is large enough. The same idea can be applied to the spatial frequency domain, where the effective domain would be denoted by W_ρ . The conditions and corresponding scaling factors are listed in **Table 1**.

The detailed derivation of these scaling factors was shown in [3]. It can be observed that the scaling factors for the space-limited or frequency limited cases can be derived from each other via $W_\rho R = j_{nN}$.

Implementation and architecture

The software is based on the MATLAB programming language. The implementation of the discrete Hankel transform is decomposed into distinct functions. These functions consist of the various steps that have to be performed in order to properly execute the transform. These steps are as follows:

	Condition	Scaling Factor
1	Space-limited function	$\alpha = \frac{R^2}{j_{nN}}$
2	Frequency-limited function	$\alpha = \frac{j_{nN}}{W_\rho^2}$

Table 1: Scaling factor under various conditions.

1. Calculations of N Bessel zeros of the Bessel function of order n
2. Generation of N sample points (if using the DHT to approximate the continuous transform)
3. Discretization of the function (if needed)
4. Creation of the Y^{nN} transformation matrix
5. Performing the matrix-function multiplication

The steps are the same regardless of if the function is in the space or frequency domain and are summarized in **Figure 1**.

Furthermore, the Y^{nN} transformation matrix is used for both the forward and inverse transform. Steps 2–3 only need to be performed if the function (vector) to be transformed is not already given as a set of discrete points. In the case of a continuous function in either the space or frequency domain, it is important to use the sampling points as proposed in equations (8), (9) and then to discretize the continuous function by evaluating at these points. Failing to do so results in the function not being properly transformed since there is a necessary relationship between the sampling points and the transformation matrix Y^{nN} . In order to perform the steps listed above, several Matlab functions have been developed. These functions are listed in **Table 2**.

Additionally, the matlab script **GuidetoDHT.m** is included to illustrate the execution of the necessary computational steps.

Quality control

The software was tested by using the DHT to approximate the computation of both the continuous Hankel

forward and inverse transforms and comparing the results with known (continuous) forward and inverse Hankel transform pairs. Different transform orders n were evaluated.

For the purpose of testing the accuracy of the DHT and IDHT, the dynamic error was used, defined as [6]

$$e(\nu) = 20 \log_{10} \left[\frac{|f(\nu) - f^*(\nu)|}{\max |f^*(\nu)|} \right] \tag{10}$$

This error function compares the difference between the exact function values $f(\nu)$ (evaluated from the continuous function) and the function values estimated via the discrete transform, $f^*(\nu)$, scaled with the maximum value of the discretely estimated samples. Equation (10) can be used to evaluate the computation of either forward or inverse Hankel transform via the DHT/IDHT and compared with known continuous Hankel relationships. The dynamic error uses the ratio of the absolute error to the maximum amplitude of the function on a log scale. Therefore, negative decibel errors imply an accurate discrete estimation of the true transform value. The transform was also tested for accuracy on itself. This is performed by consecutive forward and then inverse transformation in order to verify that the transforms themselves do not add errors. For this evaluation, the average absolute error $\frac{1}{N} \sum_{i=1}^N |f_i - f_i^*|$ was used.

The methodology of the testing is given in further detail in [7] and also in the theory paper, [3].

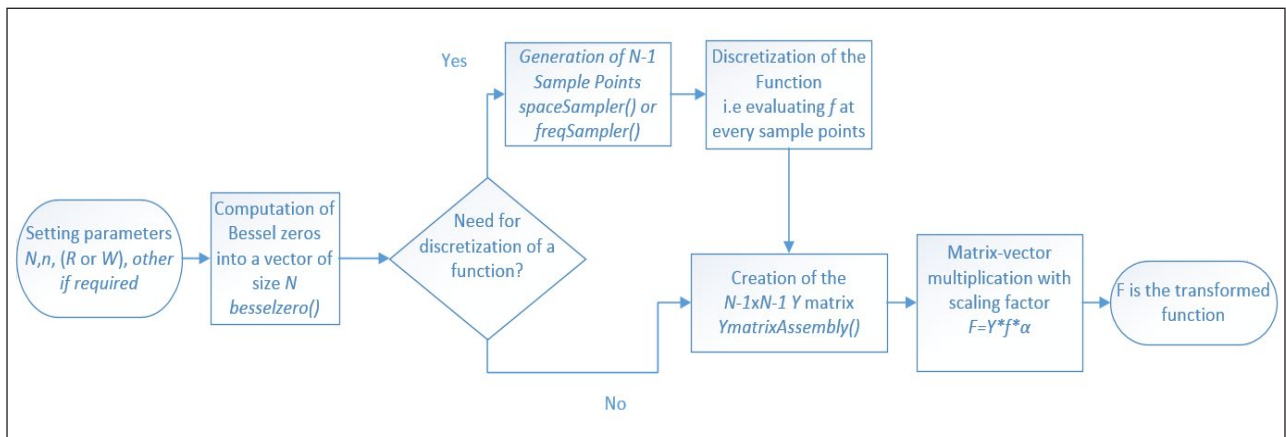


Figure 1: Steps for evaluation of DHT.

Function Name	Calling Sequence	Description
besszero	besszero(n,k,kind)	Calculation of k Bessel zeros of the nth order Bessel function of kind – developed in [5]
freqSampler	freqSampler(R,zeros)	Creation of sample points in the frequency domain (eq. (9))
spaceSampler	spaceSampler(R,zeros)	Creation of sample points in the space domain (eq. (8))
YmatrixAssembly	YmatrixAssembly(n,N,zeros)	Creation of Y^{nN} matrix (eq. (4)) from the zeros

Table 2: Set of available functions.

(2) Availability**Operating system**

Windows XP and higher.

Programming language

Matlab

Additional system requirements

If using Matlab 7, minimum system requirements are 512MB of RAM (1024 MB recommended) and 460MB of hard disk space. System requirements for Matlab R2014 require 1024 MB (At least 2048 MB recommended) or RAM and 1 GB for MATLAB only, 3–4 GB for a typical installation. This code should run on older versions of Matlab although it was tested using Matlab R2014.

Dependencies

Matlab software by mathworks.

List of contributors

Ugo Chouinard, Natalie Baddour, Greg von Winckel

Software location

Archive (e.g. institutional repository, general repository) (required)

Name: Figshare

Persistent identifier: <http://dx.doi.org/10.6084/m9.figshare.1453205>

Licence: BSD

Publisher: Natalie Baddour

Date published: 18/06/15

Code repository (e.g. SourceForge, GitHub etc.) (required)

Name: GitHub

Persistent identifier: <https://github.com/uchouinard/DiscreteHankelTransform>

Licence: BSD

Publisher: Natalie Baddour

Date published: 07/12/16

Language

English

(3) Reuse potential

The Discrete Hankel Transform is applicable to many areas of scientific computation and potential reuse could be very high. Further details on applications can be found in [3].

Competing Interests

The authors have no competing interests to declare.

References

1. **Leutenegger, M** *Hankel transform – File Exchange – MATLAB Central*.
2. **Wyatt, A** *Hankel Transform – File Exchange – MATLAB Central*.
3. **Baddour, N** and **Chouinard, U** 2015 (Apr.) Theory and operational rules for the discrete Hankel transform. *J. Opt. Soc. Am. A*, 32(4): 611–622. DOI: <https://doi.org/10.1364/JOSAA.32.000611>
4. **Debnath, L** and **Bhatta, D** 2007 *Integral transforms and their applications*, 2nd ed. Boca Raton: Chapman & Hall/CRC.
5. **von Winckel, G** Bessel Function Zeros – File Exchange – MATLAB Central. [Online]. Available at: <http://www.mathworks.com/matlabcentral/fileexchange/6794-bessel-function-zeros>. [Accessed: 06-Jun-2015].
6. **Guizar-Sicairos, M** and **Gutiérrez-Vega, J C** 2004 (Jan.) Computation of quasi-discrete Hankel transforms of integer order for propagating optical wave fields, *J. Opt. Soc. Am. A*, 21(1): pp. 53–58. DOI: <https://doi.org/10.1364/JOSAA.21.000053>
7. **Chouinard, U** 2015 Numerical Simulations for the Discrete Hankel Transform, B.A.Sc. thesis, University of Ottawa, Ottawa, Canada. DOI: <http://dx.doi.org/10.13140/RG.2.1.1208.3364>


How to cite this article: Chouinard, U and Baddour, N 2017 Matlab Code for the Discrete Hankel Transform. *Journal of Open Research Software*, 5: 4, DOI: <http://doi.org/10.5334/jors.82>

Submitted: 19 June 2015

Accepted: 08 December 2016

Published: 11 January 2017

Copyright: © 2017 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS 