CROATICA CHEMICA ACTA
CCACAA **78** (4) 601–615 (2005)

ISSN-0011-1643
CCA-3054
Original Scientific Paper

Approximate solution of dense linear systems*

Željko Jeričević

Department of Molecular and Cellular Biology, Baylor College of Medicine, Houston, TX 77030, USA
(E-mail: zeljkoj@bcm.tmc.edu)

RECEIVED SEPTEMBER 28, 2004; REVISED APRIL 12, 2005; ACCEPTED APRIL 15, 2005

Key words
Fourier
linear system
approximate solution
eigenvalue problem
distance matrix
nanotube

A novel numerical approach for approximate solution of large linear systems of a dense type has been developed. The method is based on Fourier transform although any unitary, orthogonal transform which concentrates energy in a small number of coefficients can be used. The idea comes from digital signal processing where pruning off insignificant information from spectra or filtering of selected information in frequency domain is usual practice. The procedure is to transform the linear system from the time and space domain to the frequency domain, generating a transformed system. The least significant portions in the transformed system are deleted as the whole rows and columns, yielding a smaller pruned system. The pruned system is solved in the frequency domain, generating the transform of approximate solution. Inverting the transform of approximate solution yields the approximate solution of original system. Numerical experiments illustrating feasibility of the method and quality of the approximation for 1000 by 1000 eigenvalue problem in chemical graph theory are presented.

INTRODUCTION

Solving a system of linear equations and solving eigenvalue problem is of fundamental importance in scientific computing, with wide applications in chemistry, physics, biology and engineering. For dense systems of size n, the problem is generally of the order $O(n^3)$ number of operations unless a matrix is structured in a specific way. Practical limits with today's computing technology for single processor machines are matrices of the size around 10000 by 10000. In comparison, sparse matrices with dimensions of millions are not uncommon. An approximate solution for dense, large and/or ill conditioned systems which otherwise defy classical approaches like a singular value decomposition is needed. In particular, a method which can efficiently decrease the size of column and/or row dimension, while still yielding a good approximation of solution for original large system would be useful.

The problem of solving a set of linear equations, given in matrix form is:

$$y = \boldsymbol{A}x \tag{1}$$

Here y is a column vector of size m representing a set of known values, $\boldsymbol{A}$ is a known matrix of size m by n, that is, m rows by n columns, and x is a column vector of size n. The vector x represents the solution of the above system. A system of linear equations for which $m > n$ is called overdetermined and is typical for the least squares problems.

The eigenvalue problem involves finding the eigenvalues and eigenvectors of a diagonizable matrix $\boldsymbol{B}$, expressed in matrix form is:

$$\boldsymbol{B}q_i = \lambda_i q_i \tag{2}$$

Here $\boldsymbol{B}$ is a square matrix, λ_i is the i-th eigenvalue of matrix $\boldsymbol{B}$ and q_i is the associated i-th eigenvector of matrix $\boldsymbol{B}$.

* Dedicated to academician Nenad Trinajstić, professor and friend, for his 65th birthday

THEORY

The approach presented here is comprised of the steps of transforming a linear system using the Fourier or Hartley transforms, discarding the least significant portions of the transformed system and solving a reduced size problem in the transform space. This yields approximate transform of solution constructed by using a limited bandwidth in the Fourier domain. The result is then inverse transformed to give an approximate solution in the time or space domain. The idea was first developed by Jeričević[1] for analysis of geophysical signals.

Applying a Fourier transform to Eq. (1) is formally done by premultiplying both sides with the Fourier matrix $\boldsymbol{F}$, yielding

$$\boldsymbol{F}y = \boldsymbol{FA}x \tag{3}$$

The product $\boldsymbol{F}y$ is the Fourier transform of vector y and is denoted as $\{y\}$ below, while the matrix product $\boldsymbol{FA}$ is the Fourier transform along the columns of matrix $\boldsymbol{A}$ and is denoted $\{\boldsymbol{A}\}_c$ below. Thus Eq. (3) can be rewritten as

$$\{y\} = \{\boldsymbol{A}\}_c x \tag{4}$$

The Fourier transform along the rows of matrix $\boldsymbol{A}$ can be developed in a similar fashion. Starting from Eq. (1),

$$y = \boldsymbol{AF}^{-1}\boldsymbol{F}x \tag{5}$$

Here $\boldsymbol{F}^{-1}$ is the matrix inverse of the Fourier matrix $\boldsymbol{F}$, the matrix product $\boldsymbol{AF}^{-1}$ is the inverse Fourier transform along the rows of matrix $\boldsymbol{A}$ and is denoted $\{\boldsymbol{A}\}_r^{-1}$ below, and $\boldsymbol{F}x$ is the Fourier transform of vector x and is denoted $\{x\}$ below. Thus Eq. (5) can be rewritten as

$$y = \{\boldsymbol{A}\}_r^{-1}\{x\} \tag{6}$$

Applying the Fourier transform along the columns in Eq. (6) can be written as premultiplication with the Fourier matrix $\boldsymbol{F}$

$$\boldsymbol{F}y = \boldsymbol{F}\{\boldsymbol{A}\}_r^{-1}\{x\} \tag{7}$$

Here the product $\boldsymbol{F}y$ is the Fourier transform of vector y and is denoted as $\{y\}$ below, the matrix product $\boldsymbol{F}\{\boldsymbol{A}\}_r^{-1}$ is a special two dimensional Fourier transform of matrix $\boldsymbol{A}$, is denoted as $\{\boldsymbol{A}\}$ below and is taken as a Fourier transform along the columns and an inverse Fourier transform along the rows. Note that the order in which the two dimensional transform is taken is not important. Thus Eq. (7) can also be rewritten as

$$\{\boldsymbol{A}\} = \boldsymbol{F}\{\boldsymbol{A}\}_r^{-1} = \{\boldsymbol{A}\}_c\boldsymbol{F}^{-1} = \boldsymbol{FAF}^{-1} \tag{8}$$

This leads to the following equation,

$$\{y\} = \{\boldsymbol{A}\}\ \{x\} \tag{9}$$

An equally valid result can be obtained by using $\boldsymbol{FF}^{-1}$ instead of $\boldsymbol{F}^{-1}\boldsymbol{F}$ in Eq. (5), but computationally, $\boldsymbol{F}^{-1}\boldsymbol{F}$ has the advantage because normalization of the Fourier transform is simplified when the two-dimensional transform is used. The detailed numerical example for solving a small system of linear equations based on Hilbert matrix is presented in Appendix 1 (only for the purpose of illustrating the steps in presented methodology).

Using the same approach (inserting identity matrix in a multiplication of matrix by vector and premultiplying both sides with $\boldsymbol{F}$) for the eigenvalue problem given by Eq. (2) yields the following equation

$$\boldsymbol{FBF}^{-1}\boldsymbol{FQ} = \boldsymbol{FQ\Lambda} \tag{10}$$

This leads to

$$\{\boldsymbol{B}\}\{\boldsymbol{Q}\}_c = \{\boldsymbol{Q}\}_c\boldsymbol{\Lambda} \tag{11}$$

where $\{\boldsymbol{Q}\}_c$ is a matrix containing Fourier transforms of the eigenvectors (as columns), $\boldsymbol{\Lambda}$ is diagonal matrix of eigenvalues and $\{\boldsymbol{B}\}$ is the two dimensional Fourier transform of matrix $\boldsymbol{B}$, taken as the Fourier transform along the columns and the inverse Fourier transform along the rows of matrix $\boldsymbol{B}$. The detailed numerical example (only for the purpose of illustrating the steps in presented methodology) for finding eigenvalues and eigenvectors of a small Hilbert matrix is presented in Appendix 2.

RESULTS AND DISCUSSION

The approach outlined above may be considered as a preconditioning of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. That still leaves a problem of the same order of magnitude as the initial linear system problem. However, the problem is now posed in the frequency domain in which the significance of system orthogonal components, the frequencies, is obvious. Consequently, the system can now be easily manipulated.

For an overdetermined system, that is, where the m by n matrix $\boldsymbol{A}$ has $m > n$, the most appropriate approach is to use Eq. (4) with a Fourier transform along the columns only. In the transform domain, it is now straightforward, by observing the transform magnitudes, to determine which frequencies are the most significant for the reconstruction of vector y. Consequently, Eq. (4) can be solved approximately using only the most significant frequencies present in vector $\{y\}$, preferably the frequencies with the largest magnitude. Alternative measures of significance could be average magnitude or average power computed from the system matrix. After sorting the magnitudes in $\{y\}$ and determining the threshold, the

frequencies below the threshold are not involved in computing the solution of the system. This amounts to discarding the rows in matrix $\{\boldsymbol{A}\}_c$ which do not contribute to significant frequencies in $\{y\}$. Note that the least significant rows are not merely zeroed out, but deleted. Thus the size of the matrix in the linear system to be solved is smaller.

A similar procedure would work for an underdetermined system, that is, where the m by n matrix $\boldsymbol{A}$ has $n > m$. Here the most appropriate approach is to use Eq. (6) with an inverse Fourier transform along the rows only. Then the columns in matrix $\{\boldsymbol{A}\}_r$ which do not contribute to significant frequencies are discarded. For a system which is too large in both dimensions, the most appropriate approach is to use Eq. (9) with a Fourier transform along the columns and an inverse Fourier transform along the rows. Then both rows and columns which do not contribute to significant frequencies may be discarded.

In the case of overdetermined systems which are typical for the least squares problems, the transform along the columns and subsequent reduction in number of rows of the transformed matrix may be all that is needed to simplify the computations. In such a case, it is obvious that the wavelet transform can also be used in a similar way, because the scaling and linearity of the applied transform are the only prerequisites for validity of Eq. (4).

The three equations, Eqs. (4), (6) and (9), are based on the same idea and derived from Eq. (1) for corresponding type of problems. The different approaches allow possible simplification or approximation, arising from size reduction of the original problem. Eq. (4) is most appropriate for overdetermined problems, Eq. (6) for underdetermined problems, and Eq. (9) for problems simply too large in both dimensions. Each frequency in the Fourier domain represents the best approximation in the least squares sense to the original sequence by that particular frequency. Consequently, the approximate solution constructed in the Fourier domain is the least squares approximation of the true solution with the set of chosen frequencies. This can be used to control the accuracy of the approximation. The original form of the problem (Eq. 1), cannot be treated in the same way without perturbing it in an uncontrolled manner.

After transforming to the Fourier domain, the problem of solving the system is still of the same order of complexity. If the linear system started with a real matrix, the problem becomes more complicated because its transform will generally be complex. This complication can be addressed by using the Hartley transform.[2] The complication of dealing with a set of complex equations can be eliminated by converting our equations into the real Hartley transform domain.

The justification for the whole approach lies in constructing the approximate solution, based not on all, but only on the most significant frequencies. This is similar to filtering in digital signal processing, but applied to the linear algebra problem. The limitation of this filter is in the shape of the filter, which cannot be arbitrary, but has to include whole rows and columns in the frequency domain. By working in the Fourier domain, the size of the matrix will be reduced significantly, if it is feasible. The reduction in size pays off quickly, since the computational problem has an order the size of the matrix dimension to the third power, $O(n^3)$.

The approach outlined above discards whole rows and/or columns from transformed matrix, but it can also be used for constructing the sparse matrix in the transform space out of a dense matrix in the original domain by thresholding according to transform magnitudes. This would keep the size of the problem the same, but will make it sparse as was done previously by Beylkin[3] with the wavelet transform. In particular, diagonally dominant problems will not allow a simple reduction in size, but may allow the approximation of a full sized dense problem by a banded system in the Fourier domain.

It is clear that a unitary transform like the Fourier transform will not change the nature of the problem. If we start with a singular matrix, then its transform will be singular also. However, in the transform domain it may be easier to construct the model system which will have the most significant properties of the original system, but will not be singular. This may be a proper strategy to deal with ill-conditioned problems, namely by constructing an approximate, but stable, solution in the Fourier domain. The inverse problems are typical in that regard and are often solved by using pseudo-inverse or regularization. In this context, discarding rows and/or columns in the transformed matrix may be used to match the size and rank of the matrix.

Our general knowledge of the Fourier transform allows us to estimate feasibility of this approach by observing the overall distribution of values in a matrix. For example, it would be expected that sparse matrices will contain broad-band frequency information, meaning that successful compression in frequency domain may not be possible. In crystallography, the Fourier space is sometimes called reciprocal space, meaning that the broad peak in real space will be narrow in the Fourier space. There is a certain amount of flexibility concerning ordering of rows and columns in a matrix, as dictated by linear algebra. By reordering the rows and columns or doing any other allowed linear algebra manipulations prior to taking the transform, the frequency distribution in the Fourier transform of a matrix may be changed to suit our needs (See Figures 1, 2 and 4). This is not possible in regular signal processing, where the signal is defined as a time or space series.

A particular advantage of the present method is that computing solutions in transform space does not require construction of new linear algebra routines, but can use existing software for solving the problem in the transform domain.

As a practical example of methodology, let us assume that we want to use 20 absolutely largest eigenvalues and their associated eigenvectors of the distance matrix for a large molecule to construct a set of molecular descriptors.

Figure 1. 1000 by 1000 distance matrix shown as image. The dark region along main diagonal represents small values and the light ones the high values in the matrix, ranging from 0 to 104.

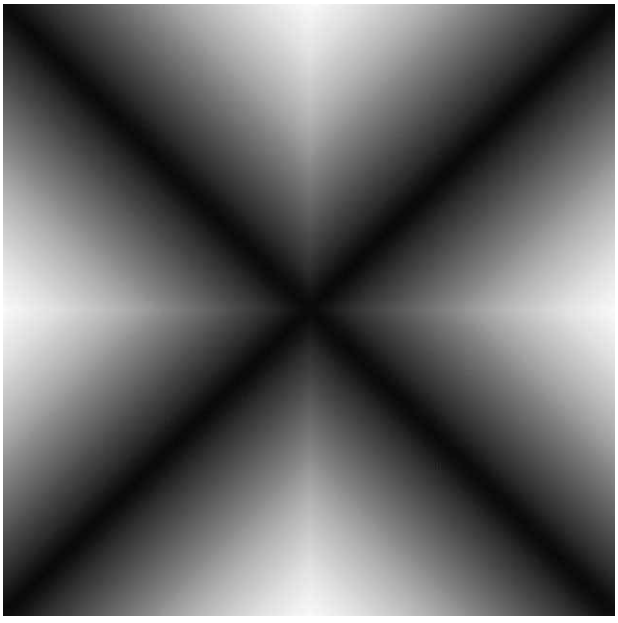

Figure 2. Image of rearranged distance matrix. Note the smooth transitions between beginning and end of each column and row, unlike the original matrix in Figure 1.

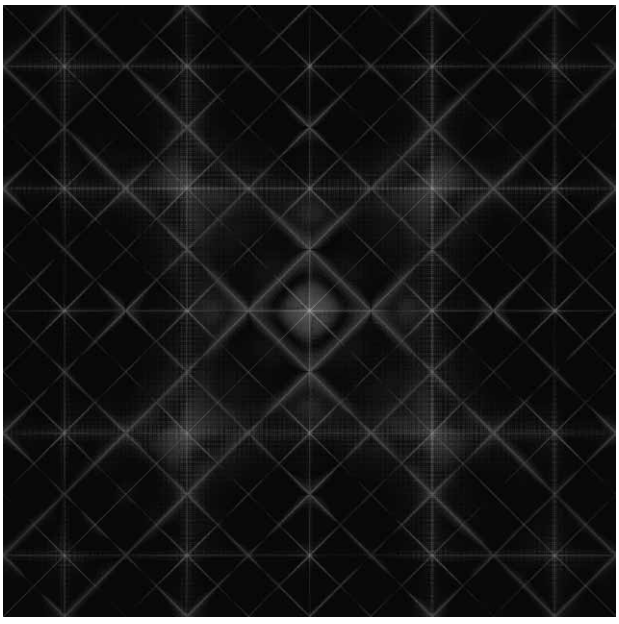

Figure 3. Magnitude of the Fourier spectrum for the distance matrix in Figure 2. The shown values are spanning five orders of magnitude and are presented on log scale in range from –4.5 to 1.5. Frequency 0,0 is in the middle of the image.

Using various levels of approximation in the Fourier domain, we can find out how good the approximation for true eigenvalues and eigenvectors are coming from truncated Fourier transform. A thousand carbon atoms molecule from the class of single wall carbon nanotubes of the type armchair (5,5) will give a 1000 by 1000 distance matrix with spectrum containing positive and negative eigenvalues. The matrix is shown as image in Figure 1. Although the matrix elements are changing smoothly, there is a sharp jump in values at the end and beginning of each column and row. Due to the periodicity of discrete Fourier transfom, a large number of frequencies would be required do describe these sharp transitions. A better arrangement of the same matrix (which will concentrate power in fewer Fourier coefficients then the original arrangement) is presented in Figure 2 and its Fourier transform magnitude in Figure 3. For the comparison sake, the Fourier transform magnitudes for both matrices are summed along the row and plotted as curves in Figure 4. The plots clearly show that rearranged matrix has three groups of the most significant frequencies around frequencies 0, 200, –200, 400 and –400 while the original matrix has slow decaying, almost monotonically decreasing frequency representation. Consequently, rearranged matrix can be better represented with few significant frequencies. The Fourier transform magnitude profiles like ones in Figure 4 are used to determine which frequencies (*i.e.* rows and columns) should be kept and which discarded in constructing approximate system in the transform domain.

The Table I (in two parts, I.I and I.II) illustrates quality of approximation for various sizes of the Fourier trans-

TABLE I.I Eigenvalues for complete rearranged matrix and approximations of size 900 to 300 in the Fourier domain

	1000	900	800	700	600	500	400	300
1	3.70282E+04	3.70282E+04	3.70282E+04	3.70282E+04	3.70282E+04	3.70282E+04	3.70282E+04	3.70281E+04
2	2.47751E+01	2.47714E+01	2.47662E+01	2.44511E+01	2.44334E+01	2.43973E+01	2.43274E+01	2.42409E+01
3	2.37712E+01	2.37615E+01	2.37501E+01	2.36617E+01	2.36285E+01	2.36085E+01	2.33822E+01	2.32543E+01
4	1.75095E+01	1.74027E+01	1.73555E+01	1.70272E+01	1.69821E+01	1.69550E+01	1.69234E+01	1.68761E+01
5	1.72810E+01	1.72407E+01	1.72212E+01	1.69352E+01	1.68193E+01	1.66126E+01	1.62503E+01	1.57187E+01
6	1.25156E+01	1.25061E+01	1.25027E+01	1.24178E+01	1.24096E+01	1.23592E+01	1.23025E+01	1.22356E+01
7	1.17072E+01	1.17009E+01	1.16909E+01	1.16634E+01	1.16252E+01	1.16068E+01	1.14897E+01	1.13613E+01
8	1.14935E+01	1.14902E+01	1.14837E+01	1.12739E+01	1.11824E+01	1.11732E+01	1.09012E+01	1.06971E+01
9	1.13056E+01	1.13043E+01	1.13012E+01	1.11012E+01	1.08433E+01	1.08213E+01	1.07744E+01	1.02020E+01
10	1.05273E+01	1.05133E+01	1.05061E+01	1.04596E+01	1.04255E+01	1.03928E+01	1.03203E+01	9.89204E+00
11	9.19964E+00	9.19454E+00	9.19292E+00	9.14461E+00	9.14060E+00	9.13057E+00	9.11023E+00	9.06739E+00
12	7.96360E+00	7.96124E+00	7.94212E+00	7.90435E+00	7.78950E+00	7.78149E+00	7.14889E+00	7.08117E+00
13	7.94322E+00	7.93905E+00	7.84307E+00	7.57835E+00	7.22179E+00	7.20387E+00	6.38412E+00	6.31461E+00
14	6.65955E+00	6.65330E+00	6.64450E+00	6.49183E+00	6.43328E+00	6.41282E+00	6.03640E+00	5.45342E+00
15	6.55446E+00	6.52785E+00	6.51064E+00	6.41969E+00	6.28693E+00	6.15359E+00	5.53245E+00	5.32766E+00
16	6.36710E+00	6.36411E+00	6.31016E+00	6.27449E+00	6.13040E+00	5.64697E+00	5.46354E+00	4.37865E+00
17	6.32633E+00	6.31851E+00	6.26425E+00	6.02551E+00	5.62339E+00	5.57014E+00	5.35336E+00	4.08509E+00
18	5.73254E+00	5.72788E+00	5.71894E+00	5.69375E+00	5.59043E+00	5.50575E+00	4.74728E+00	3.87499E+00
19	5.72464E+00	5.71988E+00	5.71471E+00	5.53712E+00	5.39848E+00	5.37343E+00	4.60252E+00	3.66105E+00
20	5.46887E+00	5.46623E+00	5.43058E+00	5.35938E+00	5.30021E+00	4.64878E+00	3.99553E+00	3.47833E+00
21	5.43741E+00	5.43509E+00	5.37649E+00	5.17066E+00	4.84131E+00	4.23359E+00	3.98962E+00	3.28563E+00
22	4.89337E+00	4.89012E+00	4.87331E+00	4.78575E+00	4.39626E+00	4.22824E+00	3.86139E+00	3.14035E+00
23	4.88368E+00	4.88113E+00	4.81607E+00	4.64397E+00	4.38451E+00	4.21291E+00	3.85834E+00	2.77797E+00
24	4.51275E+00	4.48537E+00	4.47660E+00	4.43073E+00	4.37285E+00	4.18053E+00	3.76453E+00	2.66338E+00
25	4.49157E+00	4.47199E+00	4.43593E+00	4.42213E+00	4.36422E+00	4.12073E+00	3.73890E+00	2.57363E+00
976	–4.17100E+01	–4.14700E+01	–4.13625E+01	–4.09743E+01	–4.06575E+01	–4.02119E+01	–4.01260E+01	–4.00157E+01
977	–4.43991E+01	–4.43932E+01	–4.42625E+01	–4.42225E+01	–4.31586E+01	–4.29515E+01	–4.29037E+01	–4.28897E+01
978	–5.06179E+01	–5.06096E+01	–5.05895E+01	–5.04735E+01	–5.04554E+01	–5.03964E+01	–5.02924E+01	–4.97055E+01
979	–5.76997E+01	–5.76951E+01	–5.76021E+01	–5.75558E+01	–5.68757E+01	–5.65809E+01	–5.64918E+01	–5.64414E+01
980	–6.66628E+01	–6.66562E+01	–6.66449E+01	–6.66354E+01	–6.65165E+01	–6.64942E+01	–6.63805E+01	–6.59120E+01
981	–7.10352E+01	–7.10160E+01	–7.10060E+01	–7.09836E+01	–7.09734E+01	–7.09585E+01	–7.08475E+01	–7.05539E+01
982	–7.70090E+01	–7.70057E+01	–7.69531E+01	–7.68976E+01	–7.64629E+01	–7.61234E+01	–7.60367E+01	–7.59884E+01
983	–9.04567E+01	–9.04517E+01	–9.04436E+01	–9.04360E+01	–9.03185E+01	–9.03057E+01	–9.02135E+01	–8.98516E+01
984	–1.06270E+02	–1.06269E+02	–1.06244E+02	–1.06183E+02	–1.05863E+02	–1.05597E+02	–1.05584E+02	–1.05466E+02
985	–1.18960E+02	–1.18746E+02	–1.18622E+02	–1.18381E+02	–1.18226E+02	–1.17371E+02	–1.16466E+02	–1.15221E+02
986	–1.27592E+02	–1.27592E+02	–1.27588E+02	–1.27577E+02	–1.27534E+02	–1.27454E+02	–1.27384E+02	–1.27120E+02
987	–1.32651E+02	–1.32637E+02	–1.32600E+02	–1.32547E+02	–1.32504E+02	–1.32431E+02	–1.32310E+02	–1.32103E+02
988	–1.53320E+02	–1.53320E+02	–1.53315E+02	–1.53255E+02	–1.53024E+02	–1.52833E+02	–1.52824E+02	–1.52752E+02
989	–1.89849E+02	–1.89849E+02	–1.89846E+02	–1.89839E+02	–1.89835E+02	–1.89723E+02	–1.89672E+02	–1.89491E+02
990	–2.35573E+02	–2.35573E+02	–2.35571E+02	–2.35530E+02	–2.35324E+02	–2.35246E+02	–2.35239E+02	–2.35237E+02
991	–3.05722E+02	–3.05722E+02	–3.05720E+02	–3.05716E+02	–3.05711E+02	–3.05672E+02	–3.05572E+02	–3.05457E+02
992	–3.98579E+02	–3.98579E+02	–3.98578E+02	–3.98552E+02	–3.98390E+02	–3.98380E+02	–3.98376E+02	–3.98374E+02
993	–5.60164E+02	–5.60164E+02	–5.60163E+02	–5.60160E+02	–5.60158E+02	–5.60157E+02	–5.60035E+02	–5.59970E+02
994	–7.95248E+02	–7.95248E+02	–7.95248E+02	–7.95233E+02	–7.95150E+02	–7.95147E+02	–7.95144E+02	–7.95142E+02
995	–9.38694E+02	–9.38640E+02	–9.38620E+02	–9.38575E+02	–9.38557E+02	–9.38039E+02	–9.36924E+02	–9.34940E+02
996	–1.02734E+03	–1.02733E+03	–1.02732E+03	–1.02730E+03	–1.02728E+03	–1.02725E+03	–1.02722E+03	–1.02714E+03
997	–1.31585E+03	–1.31585E+03	–1.31585E+03	–1.31585E+03	–1.31584E+03	–1.31584E+03	–1.31584E+03	–1.31575E+03
998	–2.23616E+03	–2.23616E+03	–2.23616E+03	–2.23614E+03	–2.23612E+03	–2.23612E+03	–2.23612E+03	–2.23612E+03
999	–6.29661E+03	–6.29661E+03	–6.29661E+03	–6.29661E+03	–6.29661E+03	–6.29661E+03	–6.29660E+03	–6.29659E+03
1000	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04

TABLE I.II Eigenvalues for approximations of size 200 to 10 in the Fourier domain

	200	100	50	40	30	20	10
1	3.70281E+04	3.70278E+04	3.70265E+04	3.70255E+04	3.70244E+04	3.70181E+04	3.70017E+04
2	2.41417E+01	2.36699E+01	2.15222E+01	2.10826E+01	1.77043E+01	7.04157E+00	6.26652E–01
3	2.32154E+01	1.66133E+01	1.60653E+01	1.60069E+01	7.82304E+00	5.56897E+00	5.55235E–01
4	1.68183E+01	1.18651E+01	1.02632E+01	1.01331E+01	5.38541E+00	5.12894E+00	5.46453E–01
5	1.55705E+01	9.04483E+00	7.16825E+00	6.97420E+00	2.99293E+00	5.56919E–01	
6	1.21543E+01	8.56324E+00	6.90014E+00	6.72333E+00	1.30582E+00	5.48038E–01	
7	1.13508E+01	7.47349E+00	6.29514E+00	5.49082E+00	7.03886E–01	5.34061E–01	
8	1.06115E+01	5.88978E+00	3.45221E+00	3.37665E+00	5.38009E–01		
9	1.00968E+01	4.24559E+00	3.36384E+00	1.77652E+00	5.16608E–01		
10	9.02120E+00	3.73228E+00	1.97560E+00	5.75412E–01	4.55400E–01		
11	7.30267E+00	3.25072E+00	1.25857E+00	5.42733E–01			
12	6.96669E+00	2.16351E+00	7.17728E–01	5.20016E–01			
13	6.18083E+00	2.07981E+00	5.44643E–01	4.92258E–01			
14	5.32047E+00	2.02921E+00	5.23020E–01	3.99025E–01			
15	4.04932E+00	1.39745E+00	5.05126E–01				
16	4.03045E+00	1.23375E+00	4.61758E–01				
17	3.65518E+00	1.03683E+00	4.23700E–01				
18	3.55457E+00	8.15976E–01	1.93108E–01				
19	3.24642E+00	7.14701E–01					
20	2.27873E+00	6.65298E–01					
21	2.07385E+00	5.31378E–01					
22	1.88006E+00	5.27496E–01					
23	1.57411E+00	4.89916E–01					
24	1.42516E+00	4.65900E–01					
25	1.24151E+00	4.19601E–01					
976	–3.98865E+01	–3.71418E+01	–6.97757E+00	–5.26621E–01			
977	–4.28253E+01	–3.94767E+01	–1.23129E+01	–2.17558E+00			
978	–4.87980E+01	–4.86792E+01	–1.48934E+01	–2.52923E+00			
979	–5.63957E+01	–5.62994E+01	–1.82826E+01	–3.88113E+00			
980	–6.51413E+01	–6.50903E+01	–2.27466E+01	–6.92543E+00			
981	–7.02701E+01	–6.95482E+01	–2.30836E+01	–1.37447E+01	–1.22323E–01		
982	–7.59561E+01	–7.58984E+01	–3.03383E+01	–1.81176E+01	–5.24515E–01		
983	–8.92090E+01	–8.91773E+01	–3.10536E+01	–2.26545E+01	–3.47309E+00		
984	–1.05441E+02	–1.05394E+02	–3.87326E+01	–3.00264E+01	–6.94048E+00		
985	–1.13780E+02	–1.08670E+02	–6.82688E+01	–3.08536E+01	–1.71393E+01		
986	–1.26589E+02	–1.26568E+02	–9.94385E+01	–3.86649E+01	–2.29790E+01		
987	–1.31950E+02	–1.30590E+02	–1.26538E+02	–6.72435E+01	–2.93716E+01		
988	–1.52697E+02	–1.52670E+02	–1.30225E+02	–9.91374E+01	–3.07006E+01	–3.21832E+00	
989	–1.89059E+02	–1.89045E+02	–1.89014E+02	–1.13463E+02	–6.84239E+01	–1.75484E+01	
990	–2.35122E+02	–2.35103E+02	–2.35061E+02	–1.89028E+02	–9.01241E+01	–2.51210E+01	
991	–3.05328E+02	–3.05100E+02	–3.05088E+02	–3.05072E+02	–3.05052E+02	–3.05463E+01	
992	–3.98369E+02	–3.98255E+02	–3.98242E+02	–3.98207E+02	–3.97985E+02	–6.73136E+01	
993	–5.59876E+02	–5.59682E+02	–5.59669E+02	–5.59655E+02	–5.59631E+02	–5.59469E+02	
994	–7.95138E+02	–7.95035E+02	–7.95013E+02	–7.94999E+02	–7.94843E+02	–7.87452E+02	
995	–9.31683E+02	–9.22708E+02	–8.97121E+02	–8.83992E+02	–8.56822E+02	–7.98379E+02	–6.23353E+01
996	–1.02690E+03	–1.02580E+03	–1.02519E+03	–1.02442E+03	–1.02442E+03	–1.02442E+03	–1.15972E+02
997	–1.31568E+03	–1.31546E+03	–1.31542E+03	–1.31539E+03	–1.31534E+03	–1.31508E+03	–1.31407E+03
998	–2.23612E+03	–2.23612E+03	–2.23601E+03	–2.23601E+03	–2.23601E+03	–2.23596E+03	–2.23583E+03
999	–6.29658E+03	–6.29632E+03	–6.29598E+03	–6.29580E+03	–6.29560E+03	–6.29447E+03	–6.29128E+03
1000	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02487E+04	–2.02486E+04	–2.02486E+04	–2.02485E+04

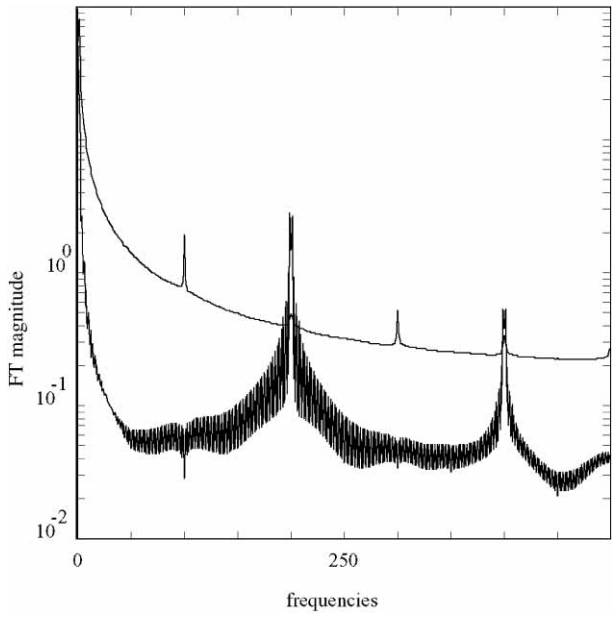

Figure 4. Along the row summation for the Fourier transform magnitude of matrices in Figure 1 and Figure 2. Upper curve coming from matrix in Figure 1 has more frequencies populated because of the sharp transitions between edges of the matrix. Lower curve is for the matrix in Figure 2 and has more compact Fourier transform representation (due to the absence of sharp transitions) with most of the energy concentrated in frequencies around 0, 200, –200, 400 and –400 (in this example magnitude is an even function, negative frequencies are not shown). The rearranged matrix (Figure 2) is better approximated with fewer Fourier coefficients.

form of rearranged distance matrix of size 1000 by 1000 for the nanotube. The numbers on the top of each column give the number of frequencies and subsequent size of the system in the Fourier domain for which eigenvalues were computed. For example, 300 means eigenvalues were found for 300 by 300 size matrix in the Fourier domain.

For systems that are smaller then 50 by 50 (added in for the sake of comparison) only the number of eigenvalues and eigenvectors up to the system size can be calculated, resulting in shorter columns at the end of the table. The results show that approximation is working remarkably well for large absolute value eigenvalues. Choosing the most significant frequencies in the Fourier spectrum will preserve them. Although there is no exactly one to one correspondence between the most significant eigenvalues/eigenvectors and the most significant frequencies for general matrix, they are related. Only in circulant matrix[4] eigenvectors are pure single frequencies of the Fourier transform. In general, the most significant frequencies contribute mostly to the significant eigenvalues/eigenvectors. This can be seen from the tables which show remarkable consistency in significant eigenvalues despite severe truncation of the Fourier transform (upper and bottom rows in Table I).

Eigenvector approximation for eigenvector 20 (associated with twentieth absolutely largest eigenvalue = –77.0090 numbered 982 in Table I) is shown in Figure 5. The severity of approximation increases from left to right. Curve 0 is a complete solution (no approximation), numbers 2–9 are approximations using 900 to 100 frequencies decreasing in step of 100 frequencies. The approximate eigenvectors look remarkably well although some details on curves are getting smoothed out as the size of matrix decreases. However, the reduction of system size from 1000 by 1000 to 100 by 100 is remarkable. This reduction in size would speed up the linear algebra part of the computations by a factor of 10^3.

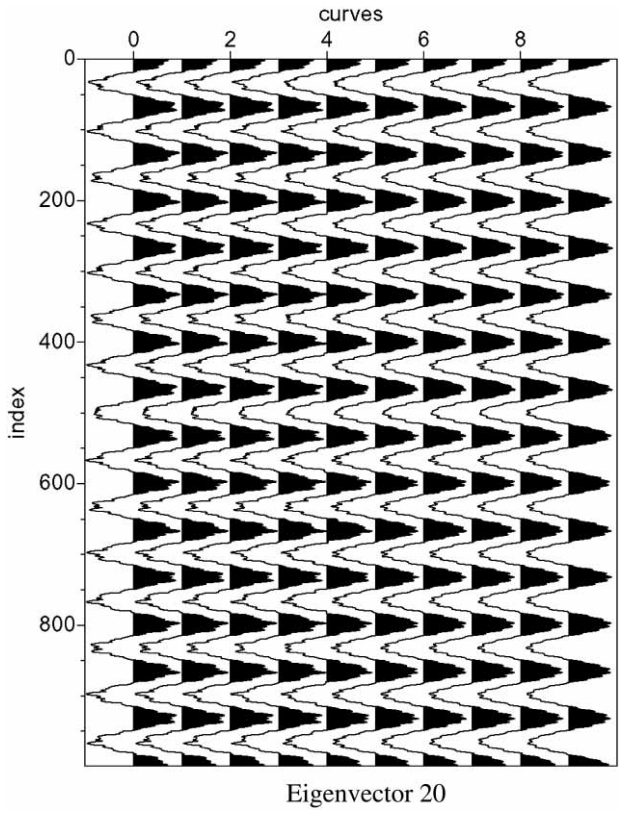

Figure 5. Plot of eigenvector associated with twentieth absolutely largest eigenvalue = –77.0090 (numbered 982 in Table I) for various levels of approximation. Positive values are filled in with black, negative are plotted as points. Leftmost curve is a true value computed from 1000 by 1000 matrix. Rightmost curve is based on 100 by 100 Fourier domain approximation of the rearranged matrix.

Curve	0	1	2	3	4	5	6	7	8	9
Size	1000	900	800	700	600	500	400	300	200	100

CONCLUSION

An approximate solution method for large, dense linear systems, inspired by signal processing has been developed. The method opens some new possibilities for solving large linear problems in scientific computing. Detailed understanding of discrete Fourier transform is needed for a successful application of the methodology.

APPENDIX 1

Detailed numerical procedure for solving a system of linear equations in the Fourier domain.

Pseudo code:

```
#   {} denotes Fourier transform
#   [] denotes Hartley transform
    input matrix A
    input known vector y
#   Fourier transform of y
    y → {y}
#   Fourier transform of A along the columns
    A → {A}_c
    if (real_system == TRUE) then
#       Hartley transform of y from the Fourier transform of y
        {y} → [y]
#       Hartley transform of A along the columns
        {A}_c → [A]_c
#       Inverse Fourier transform of [A]_c along the rows
        [A]_c → {[A]_c}_r^-1
#       2 Dimensional Hartley transform of A
        {[A]_c}_r^-1 → [{[A]_c}_r^-1]_r = [A]
    else
#       Inverse Fourier transform of {A}_c along the rows
#       yields 2 Dimensional Fourier transform of A
        {A}_c → {{A}_c}_r^-1 = {A}
    endif
    delete the least significant columns and rows in {A} or [A]
    pack and solve smaller system
    repack the solution vector of smaller system and invert the transform
```

Square Hilbert matrix of size $N = 6$ (with rearranged rows and columns) used as an example. Solution vector is $x = (1,1,1,1,1,1)$. Transpose symbols for x and y column vectors omitted for simplicity. Hilbert matrix

$$\boldsymbol{A}(i,j) = \frac{1}{i+j+1} \quad 0 \le i < N \ \text{ and } \ 0 \le j < N$$

is a well known theoretical example of ill-conditioned, but non-singular matrix.

input vector y for rearranged Hilbert matrix

2.450000e+00	1.217857e+00	8.456349e–01	7.365440e–01	9.956349e–01	1.592857e+00

input rearranged Hilbert matrix $\boldsymbol{A}$

1.000000e+00	3.333333e–01	2.000000e–01	1.666667e–01	2.500000e–01	5.000000e–01
3.333333e–01	2.000000e–01	1.428571e–01	1.250000e–01	1.666667e–01	2.500000e–01
2.000000e–01	1.428571e–01	1.111111e–01	1.000000e–01	1.250000e–01	1.666667e–01
1.666667e–01	1.250000e–01	1.000000e–01	9.090909e–02	1.111111e–01	1.428571e–01
2.500000e–01	1.666667e–01	1.250000e–01	1.111111e–01	1.428571e–01	2.000000e–01
5.000000e–01	2.500000e–01	1.666667e–01	1.428571e–01	2.000000e–01	3.333333e–01

Fourier transform of y vector (real part, not normalized)

7.838528e+00	2.198178e+00	8.605520e–01	7.440116e–01	8.605520e–01	2.198179e+00

Fourier transform of y vector (imaginary part, not normalized)

0.000000e+00	4.546634e–01	1.948558e–01	–2.549245e–08	–1.948558e–01	–4.546634e–01

Hartley transform of vector *y*

7.838528e+00	1.743515e+00	6.656961e–01	7.440116e–01	1.055408e+00	2.652842e+00

columnwise Fourier transform of input matrix (real part, not normalized)

2.450000e+00	1.217857e+00	8.456349e–01	7.365440e–01	9.956349e–01	1.592857e+00
1.025000e+00	2.785714e–01	1.367064e–01	1.041306e–01	1.882937e–01	4.654762e–01
5.250001e–01	7.857141e–02	2.718256e–02	1.809171e–02	4.384920e–02	1.678571e–01
4.500000e–01	6.785712e–02	2.658729e–02	1.901160e–02	4.007929e–02	1.404763e–01
5.250001e–01	7.857141e–02	2.718256e–02	1.809171e–02	4.384920e–02	1.678571e–01
1.025000e+00	2.785714e–01	1.367064e–01	1.041306e–01	1.882937e–01	4.654762e–01

columnwise Fourier transform of input matrix (imaginary part, not normalized)

0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
1.876389e–01	6.392099e–02	3.264786e–02	2.508720e–02	4.433220e–02	1.010363e–01
1.010363e–01	2.268155e–02	8.591566e–03	5.842202e–03	1.340275e–02	4.330128e–02
–8.586639e–09	–1.840184e–09	–1.130258e–09	–1.223206e–09	–1.885106e–09	–1.655032e–11
–1.010363e–01	–2.268155e–02	–8.591566e–03	–5.842202e–03	–1.340275e–02	–4.330128e–02
–1.876389e–01	–6.392098e–02	–3.264786e–02	–2.508720e–02	–4.433219e–02	–1.010363e–01

columnwise Hartley transform of input matrix

2.450000e+00	1.217857e+00	8.456349e–01	7.365440e–01	9.956349e–01	1.592857e+00
8.373611e–01	2.146504e–01	1.040585e–01	7.904341e–02	1.439615e–01	3.644400e–01
4.239638e–01	5.588986e–02	1.859100e–02	1.224951e–02	3.044645e–02	1.245558e–01
4.500000e–01	6.785712e–02	2.658729e–02	1.901160e–02	4.007930e–02	1.404763e–01
6.260364e–01	1.012530e–01	3.577413e–02	2.393391e–02	5.725195e–02	2.111584e–01
1.212639e+00	3.424924e–01	1.693542e–01	1.292178e–01	2.326259e–01	5.665125e–01

rowwise Fourier transform (real part)

7.838528e+00	2.198178e+00	8.605520e–01	7.440116e–01	8.605520e–01	2.198178e+00
1.743515e+00	9.238529e–01	5.028493e–01	4.272473e–01	5.028493e–01	9.238529e–01
6.656964e–01	4.774184e–01	3.214717e–01	2.803061e–01	3.214717e–01	4.774184e–01
7.440116e–01	5.018218e–01	3.315116e–01	2.893215e–01	3.315116e–01	5.018218e–01
1.055408e+00	7.117952e–01	4.472516e–01	3.827173e–01	4.472516e–01	7.117952e–01
2.652842e+00	1.336933e+00	6.863642e–01	5.763962e–01	6.863642e–01	1.336933e+00

rowwise Fourier transform (imaginary part)

0.000000e+00	–4.546633e–01	–1.948558e–01	2.149970e–08	1.948558e–01	4.546633e–01
0.000000e+00	–1.642786e–01	–9.516452e–02	9.361096e–09	9.516452e–02	1.642786e–01
0.000000e+00	–6.973357e–02	–4.919933e–02	3.975638e–09	4.919933e–02	6.973357e–02
0.000000e+00	–7.457446e–02	–5.120563e–02	5.055223e–10	5.120563e–02	7.457446e–02
0.000000e+00	–1.137812e–01	–7.658054e–02	1.157829e–09	7.658054e–02	1.137812e–01
0.000000e+00	–2.488019e–01	–1.392122e–01	–5.206385e–09	1.392122e–01	2.488019e–01

2 dimensional Hartley transform of input matrix

7.838528e+00	1.743515e+00	6.656961e–01	7.440116e–01	1.055408e+00	2.652842e+00
1.743515e+00	7.595744e–01	4.076847e–01	4.272473e–01	5.980138e–01	1.088131e+00
6.656964e–01	4.076848e–01	2.722724e–01	2.803061e–01	3.706710e–01	5.471520e–01
7.440116e–01	4.272473e–01	2.803060e–01	2.893215e–01	3.827173e–01	5.763962e–01
1.055408e+00	5.980139e–01	3.706710e–01	3.827173e–01	5.238321e–01	8.255764e–01
2.652842e+00	1.088131e+00	5.471520e–01	5.763962e–01	8.255764e–01	1.585735e+00

Hartley transform packed using
5 most significant frequencies (rowwise): 0 1 5 2 4
5 most significant frequencies (columnwise): 0 1 5 2 4

7.838528e+00 1.743515e+00 2.652842e+00 6.656961e–01 1.055408e+00
1.743515e+00 7.595744e–01 1.088131e+00 4.076847e–01 5.980138e–01
2.652842e+00 1.088131e+00 1.585735e+00 5.471520e–01 8.255764e–01
6.656964e–01 4.076848e–01 5.471520e–01 2.722724e–01 3.706710e–01
1.055408e+00 5.980139e–01 8.255764e–01 3.706710e–01 5.238321e–01
y vector frequencies packed in the same way as columnwise frequencies
7.838528e+00 1.743515e+00 2.652842e+00 6.656961e–01 1.055408e+00

SVD of the above matrix gives the following singular values (***U*** & ***V*** matrices not shown)
s[0] = 9.596282e+00
s[1] = 1.307621e+00
s[2] = 7.313725e–02
s[3] = 2.900383e–03
s[4] = 8.033032e–07

packed Hartley transform of solution vector computed using first four singular values
1.000001e+00 7.586220e–06 –1.792940e–05 –8.987379e–05 8.150839e–05
repacked Hartley transform of solution vector (zero inserted for deleted frequency 3)
1.000001e+00 7.586220e–06 –8.987379e–05 0.000000e+00 8.150839e–05 –1.792940e–05

solution vector (from 5 by 5 reduced system using first four singular values)
9.999825e–01 9.998739e–01 1.000181e+00 1.000003e+00 9.998400e–01 1.000126e+00

Another iteration, using 3 by 3 system
Hartley transform packed using
3 most significant frequencies (rowwise): 0 1 5
3 most significant frequencies (columnwise: 0 1 5
7.838528e+00 1.743515e+00 2.652842e+00
1.743515e+00 7.595744e–01 1.088131e+00
2.652842e+00 1.088131e+00 1.585735e+00
y vector frequencies packed in the same way as columnwise frequencies
7.838528e+00 1.743515e+00 2.652842e+00

SVD of the above matrix gives the following singular values (***U*** & ***V*** matrices not shown)
s[0] = 9.288025e+00
s[1] = 8.885208e–01
s[2] = 7.292748e–03

packed HT of solution vector computed using all three singular values
9.999989e–01 –3.967757e–05 2.937355e–05
repacked HT of solution vector (zero substituted for deleted frequencies: 2, 3, 4)
9.999989e–01 –3.967757e–05 0.000000e+00 0.000000e+00 0.000000e+00 2.937355e–05

solution vector (from 3 by 3 reduced system using all three singular values)
9.999886e–01 9.999340e–01 9.999443e–01 1.000009e+00 1.000064e+00 1.000054e+00

Another iteration, using 1 by 1 system
Hartley transform packed (all frequencies except 0,0 are deleted resulting in 1 by 1 matrix)
7.838528e+00
y vector frequencies packed in the same way as columnwise frequencies
7.838528e+00

SVD of the above matrix gives the following singular value (***U*** & ***V*** matrices not shown)

$s[0]$ = 7.838528e+00

packed Hartley transform of solution vector

1.000000e+00

repacked Hartley transform of solution vector

1.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

solution vector (from 1 by 1 reduced system using 1 singular value)

1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00

Comments on the above solutions:

1. The above example is presented only to illustrate the details of numerical procedure in traceable, easily verifiable steps. It does not pay off to apply the procedure on a matrix of such a small size which can be easily handled directly. However, some issues of ill–conditioning may be more easily addressed in frequency domain.

2. Rearranging of Hilbert matrix was done to improve concentration of power in smaller number of Fourier transform coefficients. Because of the periodicity of discrete Fourier transform, the sharp transition between the first and the last element in each column and row of the matrix ***A*** introduces more power into high frequencies. Rearranging the rows and columns in input matrix reduces this undesirable effect.

3. Reducing the size of a system in frequency domain by discarding the least significant frequencies improves the condition number of the matrix. Value for ratio of highest/lowest singular values is decreasing.

4. Because the transform is not normalized, the singular values in the transfer domain are inflated by the factor N (size of matrix ***A*** column), but that does not affect solution of the system because the transform of vector y on the other side of equation is also not normalized. This saves $N(N+1)$ divisions in the computation of the transforms.

5. If the solution vector is smooth (in the above case it contained only frequency 0), the reduction of the system can be severe and still give good results.

APPENDIX 2

Detailed numerical procedure for solving an eigenvalue problem in the Fourier domain.

Pseudo code:

```
#   {} denotes Fourier transform
#   [] denotes Hartley transform
    input matrix A
#   Inverse Fourier transform of A along the rows
    A → {A}_r^-1
    if (real_system == TRUE) then
#       Hartley transform of A along the rows
        {A}_r^-1 → [A]_r
#       Fourier transform of [A]_r along the columns
        [A]_r → {[A]_r}_c
#       2 Dimensional Hartley transform of A
        {[A]_r}_c → [{[A]_r}_c]_c = [A]
    else
#       Fourier transform of {A}_r^-1 along the columns
#       yields 2 Dimensional Fourier transform of A
        {A}_r^-1 → {{A}_r^-1}_c = {A}
    endif
    delete the least significant columns and rows in {A} or [A]
    pack and solve smaller system
    repack the transform of eigenvectors for smaller system and invert the transform
```

The same square Hilbert matrix of order 6 (with rearranged rows and columns) as in Appendix 1 is used as an example.

input matrix

1.000000e+00	3.333333e–01	2.000000e–01	1.666667e–01	2.500000e–01	5.000000e–01
3.333333e–01	2.000000e–01	1.428571e–01	1.250000e–01	1.666667e–01	2.500000e–01
2.000000e–01	1.428571e–01	1.111111e–01	1.000000e–01	1.250000e–01	1.666667e–01
1.666667e–01	1.250000e–01	1.000000e–01	9.090909e–02	1.111111e–01	1.428571e–01
2.500000e–01	1.666667e–01	1.250000e–01	1.111111e–01	1.428571e–01	2.000000e–01
5.000000e–01	2.500000e–01	1.666667e–01	1.428571e–01	2.000000e–01	3.333333e–01

rowwise inverse Fourier transform (real part) of input matrix

2.450000e+00	1.025000e+00	5.250001e–01	4.500000e–01	5.250001e–01	1.025000e+00
1.217857e+00	2.785714e–01	7.857141e–02	6.785712e–02	7.857141e–02	2.785714e–01
8.456349e–01	1.367064e–01	2.718256e–02	2.658729e–02	2.718256e–02	1.367064e–01
7.365440e–01	1.041306e–01	1.809171e–02	1.901160e–02	1.809171e–02	1.041306e–01
9.956349e–01	1.882937e–01	4.384920e–02	4.007929e–02	4.384920e–02	1.882937e–01
1.592857e+00	4.654762e–01	1.678571e–01	1.404763e–01	1.678571e–01	4.654762e–01

rowwise inverse Fourier transform (imaginary part) of input matrix

0.000000e+00	–1.876389e–01	–1.010363e–01	8.586639e–09	1.010363e–01	1.876389e–01
0.000000e+00	–6.392099e–02	–2.268155e–02	1.840184e–09	2.268155e–02	6.392098e–02
0.000000e+00	–3.264786e–02	–8.591566e–03	1.130258e–09	8.591566e–03	3.264786e–02
0.000000e+00	–2.508720e–02	–5.842202e–03	1.223206e–09	5.842202e–03	2.508720e–02
0.000000e+00	–4.433220e–02	–1.340275e–02	1.885106e–09	1.340275e–02	4.433219e–02
0.000000e+00	–1.010363e–01	–4.330128e–02	1.655032e–11	4.330128e–02	1.010363e–01

rowwise Hartley transform of input matrix

2.450000e+00	8.373611e–01	4.239638e–01	4.500000e–01	6.260364e–01	1.212639e+00
1.217857e+00	2.146504e–01	5.588986e–02	6.785712e–02	1.012530e–01	3.424924e–01
8.456349e–01	1.040585e–01	1.859100e–02	2.658729e–02	3.577413e–02	1.693542e–01
7.365440e–01	7.904341e–02	1.224951e–02	1.901160e–02	2.393391e–02	1.292178e–01
9.956349e–01	1.439615e–01	3.044645e–02	4.007930e–02	5.725195e–02	2.326259e–01
1.592857e+00	3.644400e–01	1.245558e–01	1.404763e–01	2.111584e–01	5.665125e–01

columnwise Fourier transform (real part) of the above Hartley transform gives 2 dimensional Fourier transform of the input matrix ***A***

1.306421e+00	2.905858e–01	1.109494e–01	1.240019e–01	1.759013e–01	4.421403e–01
3.663630e–01	1.539755e–01	7.956973e–02	8.363696e–02	1.186325e–01	2.228222e–01
1.434253e–01	8.380821e–02	5.357862e–02	5.525194e–02	7.454193e–02	1.143940e–01
1.240019e–01	7.120789e–02	4.671768e–02	4.822026e–02	6.378622e–02	9.606604e–02
1.434253e–01	8.380821e–02	5.357862e–02	5.525194e–02	7.454193e–02	1.143940e–01
3.663630e–01	1.539755e–01	7.956973e–02	8.363696e–02	1.186325e–01	2.228222e–01

columnwise Fourier transform (imaginary part) of the above Hartley transform

0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
7.577723e–02	2.737976e–02	1.162226e–02	1.242908e–02	1.896354e–02	4.146699e–02
3.247597e–02	1.586075e–02	8.199888e–03	8.534271e–03	1.276342e–02	2.320203e–02
–3.583283e–09	–1.560183e–09	–6.626064e–10	–8.425372e–11	–1.929715e–10	8.677308e–10
–3.247597e–02	–1.586075e–02	–8.199888e–03	–8.534271e–03	–1.276342e–02	–2.320203e–02
–7.577723e–02	–2.737976e–02	–1.162226e–02	–1.242908e–02	–1.896354e–02	–4.146699e–02

2 dimensional Hartley transform of input matrix

1.306421e+00	2.905858e–01	1.109494e–01	1.240019e–01	1.759013e–01	4.421403e–01
2.905858e–01	1.265957e–01	6.794748e–02	7.120788e–02	9.966899e–02	1.813552e–01
1.109494e–01	6.794746e–02	4.537873e–02	4.671767e–02	6.177851e–02	9.119201e–02
1.240019e–01	7.120789e–02	4.671768e–02	4.822026e–02	6.378622e–02	9.606604e–02
1.759013e–01	9.966896e–02	6.177851e–02	6.378621e–02	8.730536e–02	1.375961e–01
4.421403e–01	1.813552e–01	9.119199e–02	9.606604e–02	1.375961e–01	2.642892e–01

packed Hartley transform using all frequencies in order of significance: 0 1 5 4 2 3

1.306421e+00	2.905858e–01	4.421403e–01	1.759013e–01	1.109494e–01	1.240019e–01
2.905858e–01	1.265957e–01	1.813552e–01	9.966899e–02	6.794748e–02	7.120788e–02
4.421403e–01	1.813552e–01	2.642892e–01	1.375961e–01	9.119199e–02	9.606604e–02
1.759013e–01	9.966896e–02	1.375961e–01	8.730536e–02	6.177851e–02	6.378621e–02
1.109494e–01	6.794746e–02	9.119201e–02	6.177851e–02	4.537873e–02	4.671767e–02
1.240019e–01	7.120789e–02	9.606604e–02	6.378622e–02	4.671768e–02	4.822026e–02

eigenvalues for packed Hartley transform

wr[0] = 1.618900e+00
wr[1] = 2.423609e–01
wr[2] = 1.632154e–02
wr[3] = 6.157392e–04
wr[4] = 1.257138e–05
wr[5] = 6.424014e–08

eigenvectors for packed Hartley transform (eigenvectors written as rows)

–8.807625e–01	–2.343641e–01	–3.494819e–01	–1.565329e–01	–1.019159e–01	–1.109227e–01
–4.556150e–01	3.918261e–01	4.966863e–01	4.341718e–01	3.200961e–01	3.181559e–01
–1.264042e–01	1.518260e–01	6.440523e–01	–3.082627e–01	–4.754283e–01	–4.744571e–01
–2.605136e–02	3.784722e–01	–1.904360e–02	–8.069104e–01	2.470228e–01	3.789365e–01
3.932562e–03	3.606842e–01	–1.924776e–01	9.148835e–03	5.636602e–01	–7.176680e–01
3.694864e–04	–7.041647e–01	4.232936e–01	–2.019650e–01	5.306142e–01	–5.131362e–02

repacked Hartley transform eigenvectors as rows

–8.807625e–01	–2.343641e–01	–1.019159e–01	–1.109227e–01	–1.565329e–01	–3.494819e–01
–4.556150e–01	3.918261e–01	3.200961e–01	3.181559e–01	4.341719e–01	4.966863e–01
–1.264042e–01	1.518260e–01	–4.754283e–01	–4.744571e–01	–3.082627e–01	6.440523e–01
–2.605136e–02	3.784721e–01	2.470228e–01	3.789365e–01	–8.069104e–01	–1.904360e–02
3.932562e–03	3.606842e–01	5.636602e–01	–7.176679e–01	9.148835e–03	–1.924776e–01
3.694863e–04	–7.041647e–01	5.306143e–01	–5.131362e–02	–2.019650e–01	4.232936e–01

Fourier transform of eigenvectors as rows (real part)

–8.807625e–01	–2.919230e–01	–1.292244e–01	–1.109227e–01	–1.292244e–01	–2.919230e–01
–4.556150e–01	4.442562e–01	3.771340e–01	3.181559e–01	3.771340e–01	4.442562e–01
–1.264042e–01	3.979391e–01	–3.918455e–01	–4.744571e–01	–3.918455e–01	3.979391e–01
–2.605136e–02	1.797143e–01	–2.799438e–01	3.789365e–01	–2.799438e–01	1.797143e–01
3.932562e–03	8.410331e–02	2.864045e–01	–7.176679e–01	2.864045e–01	8.410331e–02
3.694863e–04	–1.404356e–01	1.643246e–01	–5.131362e–02	1.643246e–01	–1.404356e–01

Fourier transform of eigenvectors as rows (imaginary part)

0.000000e+00	–5.755892e–02	–2.730851e–02	0.000000e+00	2.730851e–02	5.755892e–02
0.000000e+00	5.243011e–02	5.703786e–02	0.000000e+00	–5.703786e–02	–5.243011e–02
0.000000e+00	2.461131e–01	8.358283e–02	0.000000e+00	–8.358283e–02	–2.461131e–01
0.000000e+00	–1.987579e–01	–5.269666e–01	0.000000e+00	5.269666e–01	1.987579e–01
0.000000e+00	–2.765809e–01	–2.772557e–01	0.000000e+00	2.772557e–01	2.765809e–01
0.000000e+00	5.637292e–01	–3.662896e–01	0.000000e+00	3.662896e–01	–5.637292e–01

eigenvectors as rows

–1.833980e+00	–7.855436e–01	–5.181426e–01	–4.444427e–01	–6.229331e–01	–1.079533e+00
1.505321e+00	–8.962528e–01	–9.508685e–01	–9.080154e–01	–9.668303e–01	–5.170446e–01
–5.886741e–01	5.667874e–01	–8.884656e–01	–1.231516e+00	–3.254442e–01	1.708888e+00
1.524261e–01	1.311662e+00	–1.153595e–01	–1.324304e+00	1.021589e+00	–1.202321e+00
2.728027e–02	1.478572e+00	–1.085412e+00	1.126203e+00	–1.083074e+00	–4.399739e–01
–3.166005e–03	–5.950524e–01	–1.685673e+00	6.612036e–01	1.536007e+00	8.889821e–02

normalized eigenvectors as rows (length = 1)

–7.487192e–01	–3.206968e–01	–2.115308e–01	–1.814430e–01	–2.543114e–01	–4.407175e–01
6.145449e–01	–3.658936e–01	–3.881904e–01	–3.706957e–01	–3.947068e–01	–2.110826e–01
–2.403252e–01	2.313900e–01	–3.627146e–01	–5.027645e–01	–1.328620e–01	6.976505e–01
6.222768e–02	5.354837e–01	–4.709531e–02	–5.406449e–01	4.170618e–01	–4.908457e–01
1.113712e–02	6.036246e–01	–4.431176e–01	4.597704e–01	–4.421633e–01	–1.796186e–01
–1.292516e–03	–2.429291e–01	–6.881731e–01	2.699352e–01	6.270720e–01	3.629254e–02

The result is identical to the one obtained directly (without going into frequency domain) as it is expected because all the frequencies were used.

packed Hartley transform using frequencies: 0 1 5 4 2

1.306421e+00	2.905858e–01	4.421403e–01	1.759013e–01	1.109494e–01
2.905858e–01	1.265957e–01	1.813552e–01	9.966899e–02	6.794748e–02
4.421403e–01	1.813552e–01	2.642892e–01	1.375961e–01	9.119199e–02
1.759013e–01	9.966896e–02	1.375961e–01	8.730536e–02	6.177851e–02
1.109494e–01	6.794746e–02	9.119201e–02	6.177851e–02	4.537873e–02

eigenvalues for packed Hartley transform

wr[0] = 1.599380e+00
wr[1] = 2.179368e–01
wr[2] = 1.218954e–02
wr[3] = 4.833942e–04
wr[4] = 1.287049e–07

eigenvectors for packed Hartley transform (eigenvectors written as rows)

8.887588e–01	2.333973e–01	3.488169e–01	1.546133e–01	1.002747e–01
4.438171e–01	–4.245215e–01	–5.463285e–01	–4.603176e–01	–3.353218e–01
–1.128242e–01	8.278545e–02	6.165784e–01	–4.660886e–01	–6.188778e–01
–2.012190e–02	4.785723e–01	–9.882530e–02	–7.112558e–01	5.048886e–01
7.389028e–05	–7.276109e–01	4.357949e–01	–2.026841e–01	4.894735e–01

repacked Hartley transform of approx eigenvectors as rows (zeros inserted in place of deleted frequency 3)

8.887588e–01	2.333973e–01	1.002747e–01	0.000000e+00	1.546133e–01	3.488169e–01
4.438171e–01	–4.245215e–01	–3.353218e–01	0.000000e+00	–4.603176e–01	–5.463285e–01
–1.128242e–01	8.278545e–02	–6.188778e–01	0.000000e+00	–4.660886e–01	6.165784e–01
–2.012190e–02	4.785723e–01	5.048886e–01	0.000000e+00	–7.112558e–01	–9.882530e–02
7.389027e–05	–7.276109e–01	4.894735e–01	0.000000e+00	–2.026841e–01	4.357949e–01

Fourier transform of approx eigenvectors as rows (real part)

8.887588e–01	2.911071e–01	1.274440e–01	0.000000e+00	1.274440e–01	2.911071e–01
4.438171e–01	–4.854251e–01	–3.978197e–01	0.000000e+00	–3.978197e–01	–4.854251e–01
–1.128242e–01	3.496819e–01	–5.424832e–01	0.000000e+00	–5.424832e–01	3.496819e–01
–2.012190e–02	1.898735e–01	–1.031836e–01	0.000000e+00	–1.031836e–01	1.898735e–01
7.389027e–05	–1.459080e–01	1.433947e–01	0.000000e+00	1.433947e–01	–1.459080e–01

Fourier transform of approx eigenvectors as rows (imaginary part)

0.000000e+00	5.770977e–02	2.716931e–02	0.000000e+00	–2.716931e–02	–5.770977e–02
0.000000e+00	–6.090350e–02	–6.249791e–02	0.000000e+00	6.249791e–02	6.090350e–02
0.000000e+00	2.668965e–01	7.639460e–02	0.000000e+00	–7.639460e–02	–2.668965e–01
0.000000e+00	–2.886988e–01	–6.080722e–01	0.000000e+00	6.080722e–01	2.886988e–01
0.000000e+00	5.817029e–01	–3.460788e–01	0.000000e+00	3.460788e–01	–5.817029e–01

approximate eigenvectors (eigenvectors as rows)

1.725861e+00	9.054070e–01	4.173100e–01	5.614325e–01	5.231053e–01	1.199437e+00
–1.322672e+00	5.699493e–01	1.324300e+00	6.190278e–01	1.329823e+00	1.424742e–01
–4.984268e–01	1.847433e–01	–2.499819e–01	–1.897154e+00	4.099360e–01	1.373939e+00
1.532579e–01	1.826188e+00	–6.599827e–01	–6.062362e–01	4.463591e–01	–1.280318e+00
–4.952719e–03	–6.973417e–01	–1.604378e+00	5.786792e–01	1.609552e+00	1.188842e–01

normalized approximate eigenvectors as rows (length = 1).

7.045798e–01	3.696309e–01	1.703661e–01	2.292039e–01	2.135568e–01	4.896680e–01
–5.399787e–01	2.326808e–01	5.406433e–01	2.527170e–01	5.428981e–01	5.816484e–02
–2.034819e–01	7.542112e–02	–1.020547e–01	–7.745101e–01	1.673557e–01	5.609081e–01
6.256726e–02	7.455382e–01	–2.694368e–01	–2.474949e–01	1.822253e–01	–5.226875e–01
–2.021939e–03	–2.846886e–01	–6.549845e–01	2.362448e–01	6.570970e–01	4.853428e–02

Comment:

Again, the above example is presented only to illustrate the details of numerical procedure in traceable, easy verifiable steps. It does not pay off to apply the procedure on a matrix of such a small size which can be easily handled directly. In particular, the concentration of power in small number of frequencies does not work well for such a small system. This can be seen from the eigenvectors coefficients in the above example. Even deletion of single frequency perturbed the coefficients of remaining five eigenvectors considerably. On the other hand, the rearranged distance matrix (of size 1000 by 1000) presented in the paper gives much better results due to strong concentration of power in three groups of frequencies (See Figure 4 for grouping of frequencies and Figure 5 for illustrations of changes in eigenvector 20).

Acknowledgments. – I am grateful for the interest and support of Gary A. Sitton, Damir S. Skerl and Sven Treitel.

REFERENCES

1. Ž. Jeričević, *U. S. Pat.* 5864786, 1999.
2. R. N. Bracewell, *The Hartley Transform*, Oxford University Press, New York, 1986, p. 160.
3. G. Beylkin, G. Coifman, and V. Rokhlin, *Com. Pure App. Math.* **44** (1991) 141–183.
4. P. J. Davis, *Circulant Matrices*, Chelsea Publishing, New York, 1994, p. 250.

SAŽETAK

Aproksimativno rješenje punih linearnih sustava

Željko Jeričević

Razvijena je nova numerička metoda za približno rješenje velikih linearnih sustava s punim matricama. Metoda se bazira na Fourierovoj transformaciji, ali može rabiti i druge ortogonalne i unitarne transformacije koje koncentriraju energiju u mali broj koeficijenata. Ideja je inspirirana digitalnim procesiranjem signala, gdje je filtriranje i uklanjanje nebitnih i/ili selektivnih komponenti iz spektra Fourierovih frekvencija uobičajena praksa. Procedura se sastoji u transformaciji linearnoga sustava u domenu frekvencija, gdje je moguće ukloniti najmanje značajne komponente kao čitave retke i/ili stupce što rezultira u smanjenome sustavu. Rješenje za smanjeni sustav se pronađe u domeni frekvencija, dajući transformat približnog rješenja. Inverzija transformata približnoga rješenja daje približno rješenje originalnoga sustava. Prikazani su numerički eksperimenti koji ilustriraju uporabu metode i kvalitetu približnoga rješenja za računanje vlastitih vrijednosti i vlastitih vektora na matrici 1000 puta 1000 iz kemijske teorije grafova.