NAG Toolbox

nag_wav_3d_init (c09ac)

1 Purpose

nag_wav_3d_init (c09ac) returns the details of the chosen three-dimensional discrete wavelet filter. For a chosen mother wavelet, discrete wavelet transform type (single-level or multi-level DWT) and end extension method, this function returns the maximum number of levels of resolution (appropriate to a multi-level transform), the filter length, the total number of coefficients and the number of wavelet coefficients in the second and third dimensions for the single-level case. This function must be called before any of the three-dimensional transform functions in this chapter.

2 Syntax

```
[nwlmax, nf, nwct, nwcn, nwcfr, icomm, ifail] = nag_wav_3d_init(wavnam, wtrans,
mode, m, n, fr)
```

```
[nwlmax, nf, nwct, nwcn, nwcfr, icomm, ifail] = c09ac(wavnam, wtrans, mode, m,
n, fr)
```

3 Description

Three-dimensional discrete wavelet transforms (DWT) are characterised by the mother wavelet, the end extension method and whether multiresolution analysis is to be performed. For the selected combination of choices for these three characteristics, and for given dimensions ($m \times n \times fr$) of data array A, nag_wav_3d_init (c09ac) returns the dimension details for the transform determined by this combination. The dimension details are: l_{max}, the maximum number of levels of resolution that would be computed were a multi-level DWT applied; n_f, the filter length; n_{ct} the total number of wavelet coefficients (over all levels in the multi-level DWT case); n_{cn}, the number of coefficients in the second dimension for a single-level DWT; and n_{cfr}, the number of coefficients in the third dimension for a single-level DWT. These values are also stored in the communication array **icomm**, as are the input choices, so that they may be conveniently communicated to the three-dimensional transform functions in this chapter.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **wavnam** – CHARACTER(*)

The name of the mother wavelet. See the C09 Chapter Introduction for details.

wavnam = 'HAAR'
Haar wavelet.

wavnam = 'DB$\boldsymbol{n}$', where $\boldsymbol{n} = 2, 3, \ldots, 10$
Daubechies wavelet with $\boldsymbol{n}$ vanishing moments ($2\boldsymbol{n}$ coefficients). For example, **wavnam** = 'DB4' is the name for the Daubechies wavelet with 4 vanishing moments (8 coefficients).

wavnam = 'BIOR***x.y***', where ***x.y*** can be one of 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5 or 3.7
: Biorthogonal wavelet of order ***x.y***. For example **wavnam** = 'BIOR3.1' is the name for the biorthogonal wavelet of order 3.1.

Constraint: **wavnam** = 'HAAR', 'DB2', 'DB3', 'DB4', 'DB5', 'DB6', 'DB7', 'DB8', 'DB9', 'DB10', 'BIOR1.1', 'BIOR1.3', 'BIOR1.5', 'BIOR2.2', 'BIOR2.4', 'BIOR2.6', 'BIOR2.8', 'BIOR3.1', 'BIOR3.3', 'BIOR3.5' or 'BIOR3.7'.

2: **wtrans** – CHARACTER(1)

The type of discrete wavelet transform that is to be applied.

wtrans = 'S'
: Single-level decomposition or reconstruction by discrete wavelet transform.

wtrans = 'M'
: Multiresolution, by a multi-level DWT or its inverse.

Constraint: **wtrans** = 'S' or 'M'.

3: **mode** – CHARACTER(1)

The end extension method.

mode = 'P'
: Periodic end extension.

mode = 'H'
: Half-point symmetric end extension.

mode = 'W'
: Whole-point symmetric end extension.

mode = 'Z'
: Zero end extension.

Constraint: **mode** = 'P', 'H', 'W' or 'Z'.

4: **m** – INTEGER

The number of elements, m, in the first dimension (number of rows of each two-dimensional frame) of the input data, A.

Constraint: **m** ≥ 2.

5: **n** – INTEGER

The number of elements, n, in the second dimension (number of columns of each two-dimensional frame) of the input data, A.

Constraint: **n** ≥ 2.

6: **fr** – INTEGER

The number of elements, fr, in the third dimension (number of frames) of the input data, A.

Constraint: **fr** ≥ 2.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **nwlmax** – INTEGER

The maximum number of levels of resolution, $l_{\max}$, that can be computed if a multi-level discrete wavelet transform is applied (**wtrans** = 'M'). It is such that $2^{l_{\max}} \leq \min(m, n, fr) < 2^{l_{\max}+1}$, for $l_{\max}$ an integer.

If **wtrans** = 'S', **nwlmax** is not set.

2: **nf** – INTEGER

The filter length, n_f, for the supplied mother wavelet. This is used to determine the number of coefficients to be generated by the chosen transform.

3: **nwct** – INTEGER

The total number of wavelet coefficients, n_{ct}, that will be generated. When **wtrans** = 'S' the number of rows required (i.e., the first dimension of each two-dimensional frame) in each of the output coefficient arrays can be calculated as $n_{\mathrm{cm}} = n_{\mathrm{ct}}/(8 \times n_{\mathrm{cn}} \times n_{\mathrm{cfr}})$. When **wtrans** = 'M' the length of the array used to store all of the coefficient matrices must be at least n_{ct}.

4: **nwcn** – INTEGER

For a single-level transform (**wtrans** = 'S'), the number of coefficients that would be generated in the second dimension, n_{cn}, for each coefficient type. For a multi-level transform (**wtrans** = 'M') this is set to 1.

5: **nwcfr** – INTEGER

For a single-level transform (**wtrans** = 'S'), the number of coefficients that would be generated in the third dimension, n_{cfr}, for each coefficient type. For a multi-level transform (**wtrans** = 'M') this is set to 1.

6: **icomm**(**260**) – INTEGER array

Contains details of the wavelet transform and the problem dimension which is to be communicated to the two-dimensional discrete transform functions in this chapter.

7: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **wavnam** = ⟨*value*⟩ was an illegal value.

ifail = 2

On entry, **wtrans** = ⟨*value*⟩ was an illegal value.

ifail = 3

On entry, **mode** = ⟨*value*⟩ was an illegal value.

ifail = 4

Constraint: **fr** ≥ 2.

Constraint: **m** ≥ 2.

Constraint: $\mathbf{n} \geq 2$.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example computes the three-dimensional multi-level resolution for $8 \times 8 \times 8$ input data by a discrete wavelet transform using the Daubechies wavelet with four vanishing moments (see **wavnam** = 'DB4' in nag_wav_3d_init (c09ac)) and zero end extension. The number of levels of transformation actually performed is one less than the maximum possible. This number of levels, the length of the wavelet filter, the total number of coefficients and the number of coefficients in each dimension for each level are printed along with the approximation coefficients before a reconstruction is performed. This example also demonstrates in general how to access any set of coefficients at any level following a multi-level transform.

9.1 Program Text

```
function c09ac_example

fprintf('c09ac example results\n\n');

m  = nag_int(8);
n  = nag_int(8);
fr = nag_int(8);
wavnam = 'DB4';
mode = 'zero';
wtrans = 'Multilevel';
a = zeros(m, n, fr);
a(:, :, 1) = [10, 31, 04, 10, 13, 15, 04, 06;
              26, 24, 03, 18, 17, 22, 20, 05;
              06, 05, 06, 11, 22, 23, 23, 01;
              09, 15, 18, 01, 30, 24, 08, 01;
              18, 04, 26, 20, 31, 21, 04, 06;
              25, 23, 25, 14, 13, 03, 03, 29;
              22, 29, 07, 29, 13, 31, 03, 12;
              22, 03, 30, 05, 10, 04, 01, 19];
a(:, :, 2) = [01, 02, 14, 31, 19, 28, 06, 15;
              26, 25, 25, 04, 05, 15, 24, 05;
              01, 29, 08, 18, 22, 18, 31, 23;
              08, 04, 16, 21, 14, 02, 02, 21;
              10, 03, 14, 03, 25, 10, 24, 15;
              03, 16, 26, 21, 16, 19, 25, 27;
              28, 29, 01, 20, 03, 24, 31, 28;
              31, 28, 14, 30, 13, 29, 20, 04];
a(:, :, 3) = [31, 26, 23, 05, 22, 01, 16, 08;
              21, 01, 29, 10, 23, 14, 09, 03;
              20, 10, 11, 22, 26, 31, 03, 21;
```

```
              09, 24, 19, 03, 04, 01, 13, 29;
              18, 16, 05, 06, 09, 16, 08, 16;
              32, 19, 32, 01, 06, 04, 01, 17;
              29, 29, 02, 29, 27, 25, 31, 06;
              28, 15, 15, 22, 18, 01, 18, 14];
a(:, :, 4) = [15, 09, 04, 14, 26, 10, 03, 28;
              21, 24, 32, 27, 01, 27, 08, 16;
              10, 27, 29, 15, 13, 01, 05, 16;
              04, 01, 08, 31, 14, 06, 05, 27;
              01, 19, 11, 31, 12, 31, 17, 26;
              27, 01, 16, 06, 18, 02, 17, 17;
              30, 09, 15, 32, 32, 29, 16, 02;
              03, 11, 26, 02, 23, 08, 10, 31];
a(:, :, 5) = [12, 07, 06, 12, 01, 13, 30, 26;
              27, 27, 20, 16, 30, 28, 13, 30;
              29, 15, 15, 05, 01, 13, 31, 02;
              31, 21, 27, 30, 08, 07, 11, 03;
              17, 04, 06, 01, 09, 25, 03, 15;
              12, 18, 16, 05, 09, 16, 06, 13;
              03, 05, 26, 30, 19, 11, 32, 24;
              06, 16, 07, 15, 31, 10, 20, 14];
a(:, :, 6) = [20, 07, 17, 11, 04, 21, 25, 17;
              18, 22, 22, 06, 01, 05, 15, 17;
              25, 24, 16, 13, 19, 16, 23, 10;
              01, 31, 05, 13, 11, 12, 01, 18;
              01, 27, 09, 05, 29, 26, 23, 13;
              02, 17, 17, 14, 31, 21, 16, 05;
              26, 21, 10, 21, 09, 11, 01, 15;
              08, 15, 18, 04, 16, 09, 03, 29];
a(:, :, 7) = [26, 02, 30, 26, 07, 04, 09, 01;
              15, 02, 10, 22, 16, 15, 04, 03;
              04, 07, 32, 27, 07, 05, 17, 04;
              22, 30, 06, 18, 32, 02, 01, 31;
              15, 19, 20, 12, 10, 28, 27, 03;
              26, 31, 21, 02, 27, 10, 22, 13;
              32, 03, 27, 23, 01, 11, 04, 26;
              03, 01, 31, 21, 27, 21, 14, 09];
a(:, :, 8) = [02, 16, 16, 23, 23, 09, 27, 12;
              15, 17, 20, 27, 05, 04, 18, 16;
              29, 32, 20, 08, 14, 32, 11, 04;
              28, 01, 15, 19, 14, 09, 30, 18;
              20, 02, 08, 11, 20, 24, 14, 03;
              18, 15, 16, 03, 23, 01, 19, 31;
              32, 27, 28, 09, 15, 23, 09, 13;
              01, 24, 30, 04, 18, 11, 01, 22];

% Query wavelet filter dimensions
[lmax, nf, nwct, nwcn, nwcfr, icomm, ifail] = ...
c09ac(...
      wavnam, wtrans, mode, m, n, fr);

% Transform one less than the max possible number of levels.
nwl = lmax - 1;

% Perform Discrete Wavelet transform
[c, dwtlvm, dwtlvn, dwtlvfr, icomm, ifail] = ...
c09fc(...
       n, fr, a, nwct, nwl, icomm);

% c09ac returns nwct based on max levels, so recalculate.
nwct = sum(7*dwtlvm(1:nwl).*dwtlvn(1:nwl).*dwtlvfr(1:nwl)) + ...
       dwtlvm(1)*dwtlvn(1)*dwtlvfr(1);

fprintf(' Number of Levels :                     %10d\n\n', nwl);
fprintf(' Length of wavelet filter :             %10d\n', nf);
fprintf(' Total number of wavelet coefficients : %10d\n\n', nwct);
fprintf(' Number of coefficients in 1st dimension for each level:\n');
fprintf(' %8d\n', dwtlvm(1:nwl));
fprintf(' Number of coefficients in 2nd dimension for each level:\n');
fprintf(' %8d\n', dwtlvn(1:nwl));
fprintf(' Number of coefficients in 3rd dimension for each level:\n');
```

```
fprintf(' %8d\n', dwtlvfr(1:nwl));

% Select the deepest level and approximation coefficients.
want_level = nag_int(nwl);
want_coeffs = nag_int(0);

% Dimensions for this set of coefficients.
nwcm = dwtlvm(1);
nwcn = dwtlvn(1);
nwcfr = dwtlvfr(1);

fprintf('\n--------------------------------\n');
fprintf(' Level %d output is %d by %d by %d.\n', nwl, nwcm, nwcn, nwcfr);
fprintf('--------------------------------\n\n');
fprintf('Approximation coefficients (LLL)\n');

%  Extract the required coefficients
[d, icomm, ifail] = c09fy(...
                          want_level, want_coeffs, c, icomm);

% Print out the selected set of coefficients
fprintf('Level %d, Coefficients %d:\n', want_level, want_coeffs);
matrix = 'General'; diag   = 'Non-unit'; fmt = 'F9.4';
labrow = 'Integer'; labcol = labrow;
rlabs  = {' '};     clabs  = rlabs;
ncols  = nag_int(80); indent = nag_int(0);

for k = 1:nwcfr
  fprintf('\n');
  title = sprintf('Frame: %3d',k);
  [ifail] =  x04cb(...
                   matrix, diag, d(:,:,k), fmt, title, labrow, ...
                   rlabs, labcol, clabs, ncols, indent);
end

% Reconstruct original data
[b, ifail] = c09fd(nwl, c, m, n, fr, icomm);

fprintf('\n Reconstruction       b : \n');
% Result should be integers so use more compact output
fmt = 'F6.1';
for k = 1:fr
  fprintf('\n');
  title = sprintf('Frame: %3d',k);
  [ifail] =  x04cb(...
                   matrix, diag, b(:,:,k), fmt, title, labrow, ...
                   rlabs, labcol, clabs, ncols, indent);
end
```

9.2 Program Results

```
     c09ac example results

 Number of Levels :                             2

 Length of wavelet filter :                     8
 Total number of wavelet coefficients :      5145

 Number of coefficients in 1st dimension for each level:
        7
        7
 Number of coefficients in 2nd dimension for each level:
        7
        7
 Number of coefficients in 3rd dimension for each level:
        7
        7

--------------------------------
 Level 2 output is 7 by 7 by 7.
```

```
--------------------------------

Approximation coefficients (LLL)
Level 2, Coefficients 0:

 Frame:   1
          1        2        3        4        5        6        7
 1   -0.0000  -0.0000   0.0000   0.0000   0.0001   0.0000   0.0000
 2   -0.0000  -0.0000   0.0000  -0.0001   0.0000  -0.0007  -0.0000
 3    0.0000   0.0000  -0.0001  -0.0002  -0.0020   0.0036  -0.0002
 4   -0.0000  -0.0000  -0.0002   0.0021   0.0025  -0.0124   0.0010
 5    0.0001  -0.0000  -0.0017   0.0009   0.0928   0.1155   0.0004
 6    0.0002  -0.0007   0.0013  -0.0063   0.1584   0.0931   0.0096
 7    0.0000  -0.0001   0.0003  -0.0006   0.0123   0.0061   0.0014

 Frame:   2
          1        2        3        4        5        6        7
 1   -0.0000   0.0000   0.0000  -0.0000  -0.0010  -0.0005  -0.0000
 2    0.0000  -0.0000   0.0001  -0.0006   0.0026   0.0035   0.0004
 3    0.0001  -0.0000  -0.0008   0.0027   0.0133  -0.0064  -0.0032
 4   -0.0002   0.0000   0.0032  -0.0067  -0.0708   0.0073   0.0148
 5   -0.0003   0.0035  -0.0155   0.0406  -0.3676  -0.3434  -0.0682
 6   -0.0011   0.0004   0.0241  -0.0866  -0.4993  -0.5807  -0.0674
 7   -0.0002  -0.0003   0.0048  -0.0128  -0.0800  -0.0731  -0.0045

 Frame:   3
          1        2        3        4        5        6        7
 1    0.0000   0.0000  -0.0002   0.0005   0.0006   0.0027   0.0005
 2   -0.0000   0.0002  -0.0012   0.0037  -0.0224   0.0005  -0.0006
 3   -0.0002  -0.0011   0.0067  -0.0126   0.0447  -0.0734   0.0068
 4    0.0008   0.0025  -0.0141  -0.0008   0.0872   0.3261  -0.0494
 5    0.0012  -0.0173   0.0687  -0.0681   0.5915  -0.1717   0.3943
 6    0.0016   0.0123  -0.1221   0.4190  -0.5269   1.2295   0.1617
 7    0.0003   0.0028  -0.0182   0.0396   0.1154   0.2823   0.0102

 Frame:   4
          1        2        3        4        5        6        7
 1   -0.0000  -0.0002   0.0011  -0.0030   0.0059  -0.0102  -0.0026
 2    0.0000  -0.0010   0.0042  -0.0106   0.0948  -0.0180  -0.0005
 3    0.0004   0.0061  -0.0296   0.0586  -0.3921   0.3650   0.0134
 4   -0.0018  -0.0155   0.0684  -0.0636   0.5365  -1.4566   0.0298
 5   -0.0070   0.0592  -0.1486  -0.1055  -2.9693   0.1109  -1.4193
 6   -0.0017  -0.0424   0.2595  -0.7280   2.4682  -4.1771  -0.5119
 7    0.0003  -0.0079   0.0273  -0.0205  -0.1224  -0.9982  -0.0710

 Frame:   5
          1        2        3        4        5        6        7
 1    0.0001  -0.0000  -0.0005  -0.0015   0.0804   0.1009   0.0139
 2   -0.0006   0.0033  -0.0017  -0.0019  -0.5303  -0.5712  -0.0438
 3   -0.0014  -0.0157   0.0800  -0.1856   0.4182   0.4931   0.0090
 4    0.0099   0.0522  -0.4140   1.1260   0.6111  -0.0042  -0.1288
 5    0.0831  -0.4718   0.9591  -2.9510  84.8494  91.3686  10.1751
 6    0.1599  -0.3194  -0.8962   1.8546 106.1903 117.2751  12.9904
 7    0.0213  -0.0211  -0.2179   0.4955  12.5323  12.9746   1.3422

 Frame:   6
          1        2        3        4        5        6        7
 1    0.0002  -0.0004  -0.0006   0.0005   0.0945   0.1342   0.0157
 2   -0.0008   0.0048  -0.0052   0.0013  -0.7012  -0.3668  -0.0231
 3   -0.0006  -0.0125   0.0347  -0.0396   1.3945  -0.2227  -0.1395
 4    0.0034   0.0166  -0.0246  -0.0495  -3.2417  -0.3508   0.3284
 5    0.1373  -0.4804  -0.1436   0.6068 105.5811 101.7766  10.0719
 6    0.1359  -0.6132   0.8736  -2.8616 121.1074 124.4215  13.7050
 7    0.0068  -0.0939   0.4312  -1.4152  12.9366  13.1259   1.6024

 Frame:   7
          1        2        3        4        5        6        7
 1    0.0000  -0.0001   0.0006  -0.0024   0.0134   0.0160   0.0014
 2   -0.0001   0.0006   0.0003  -0.0044  -0.0813  -0.0377  -0.0021
 3    0.0006   0.0002  -0.0206   0.0816   0.0851  -0.0274  -0.0148
 4   -0.0028  -0.0074   0.1035  -0.3488   0.0136  -0.1313   0.0288
```

```
5    0.0177  -0.0358  -0.0968   0.1416  11.4442  11.6279   0.9779
6    0.0187  -0.0759   0.0227   0.1041  13.7268  13.3069   1.5629
7    0.0002  -0.0164   0.0748  -0.2042   1.6290   1.2827   0.1547

Reconstruction       b :

Frame:   1
       1     2     3     4     5     6     7     8
1   10.0  31.0   4.0  10.0  13.0  15.0   4.0   6.0
2   26.0  24.0   3.0  18.0  17.0  22.0  20.0   5.0
3    6.0   5.0   6.0  11.0  22.0  23.0  23.0   1.0
4    9.0  15.0  18.0   1.0  30.0  24.0   8.0   1.0
5   18.0   4.0  26.0  20.0  31.0  21.0   4.0   6.0
6   25.0  23.0  25.0  14.0  13.0   3.0   3.0  29.0
7   22.0  29.0   7.0  29.0  13.0  31.0   3.0  12.0
8   22.0   3.0  30.0   5.0  10.0   4.0   1.0  19.0

Frame:   2
       1     2     3     4     5     6     7     8
1    1.0   2.0  14.0  31.0  19.0  28.0   6.0  15.0
2   26.0  25.0  25.0   4.0   5.0  15.0  24.0   5.0
3    1.0  29.0   8.0  18.0  22.0  18.0  31.0  23.0
4    8.0   4.0  16.0  21.0  14.0   2.0   2.0  21.0
5   10.0   3.0  14.0   3.0  25.0  10.0  24.0  15.0
6    3.0  16.0  26.0  21.0  16.0  19.0  25.0  27.0
7   28.0  29.0   1.0  20.0   3.0  24.0  31.0  28.0
8   31.0  28.0  14.0  30.0  13.0  29.0  20.0   4.0

Frame:   3
       1     2     3     4     5     6     7     8
1   31.0  26.0  23.0   5.0  22.0   1.0  16.0   8.0
2   21.0   1.0  29.0  10.0  23.0  14.0   9.0   3.0
3   20.0  10.0  11.0  22.0  26.0  31.0   3.0  21.0
4    9.0  24.0  19.0   3.0   4.0   1.0  13.0  29.0
5   18.0  16.0   5.0   6.0   9.0  16.0   8.0  16.0
6   32.0  19.0  32.0   1.0   6.0   4.0   1.0  17.0
7   29.0  29.0   2.0  29.0  27.0  25.0  31.0   6.0
8   28.0  15.0  15.0  22.0  18.0   1.0  18.0  14.0

Frame:   4
       1     2     3     4     5     6     7     8
1   15.0   9.0   4.0  14.0  26.0  10.0   3.0  28.0
2   21.0  24.0  32.0  27.0   1.0  27.0   8.0  16.0
3   10.0  27.0  29.0  15.0  13.0   1.0   5.0  16.0
4    4.0   1.0   8.0  31.0  14.0   6.0   5.0  27.0
5    1.0  19.0  11.0  31.0  12.0  31.0  17.0  26.0
6   27.0   1.0  16.0   6.0  18.0   2.0  17.0  17.0
7   30.0   9.0  15.0  32.0  32.0  29.0  16.0   2.0
8    3.0  11.0  26.0   2.0  23.0   8.0  10.0  31.0

Frame:   5
       1     2     3     4     5     6     7     8
1   12.0   7.0   6.0  12.0   1.0  13.0  30.0  26.0
2   27.0  27.0  20.0  16.0  30.0  28.0  13.0  30.0
3   29.0  15.0  15.0   5.0   1.0  13.0  31.0   2.0
4   31.0  21.0  27.0  30.0   8.0   7.0  11.0   3.0
5   17.0   4.0   6.0   1.0   9.0  25.0   3.0  15.0
6   12.0  18.0  16.0   5.0   9.0  16.0   6.0  13.0
7    3.0   5.0  26.0  30.0  19.0  11.0  32.0  24.0
8    6.0  16.0   7.0  15.0  31.0  10.0  20.0  14.0

Frame:   6
       1     2     3     4     5     6     7     8
1   20.0   7.0  17.0  11.0   4.0  21.0  25.0  17.0
2   18.0  22.0  22.0   6.0   1.0   5.0  15.0  17.0
3   25.0  24.0  16.0  13.0  19.0  16.0  23.0  10.0
4    1.0  31.0   5.0  13.0  11.0  12.0   1.0  18.0
5    1.0  27.0   9.0   5.0  29.0  26.0  23.0  13.0
6    2.0  17.0  17.0  14.0  31.0  21.0  16.0   5.0
7   26.0  21.0  10.0  21.0   9.0  11.0   1.0  15.0
8    8.0  15.0  18.0   4.0  16.0   9.0   3.0  29.0
```

```
 Frame:    7
        1      2      3      4      5      6      7      8
 1   26.0    2.0   30.0   26.0    7.0    4.0    9.0    1.0
 2   15.0    2.0   10.0   22.0   16.0   15.0    4.0    3.0
 3    4.0    7.0   32.0   27.0    7.0    5.0   17.0    4.0
 4   22.0   30.0    6.0   18.0   32.0    2.0    1.0   31.0
 5   15.0   19.0   20.0   12.0   10.0   28.0   27.0    3.0
 6   26.0   31.0   21.0    2.0   27.0   10.0   22.0   13.0
 7   32.0    3.0   27.0   23.0    1.0   11.0    4.0   26.0
 8    3.0    1.0   31.0   21.0   27.0   21.0   14.0    9.0

 Frame:    8
        1      2      3      4      5      6      7      8
 1    2.0   16.0   16.0   23.0   23.0    9.0   27.0   12.0
 2   15.0   17.0   20.0   27.0    5.0    4.0   18.0   16.0
 3   29.0   32.0   20.0    8.0   14.0   32.0   11.0    4.0
 4   28.0    1.0   15.0   19.0   14.0    9.0   30.0   18.0
 5   20.0    2.0    8.0   11.0   20.0   24.0   14.0    3.0
 6   18.0   15.0   16.0    3.0   23.0    1.0   19.0   31.0
 7   32.0   27.0   28.0    9.0   15.0   23.0    9.0   13.0
 8    1.0   24.0   30.0    4.0   18.0   11.0    1.0   22.0
```