# NAG Toolbox for MATLAB

# f08ww

## 1 Purpose

f08ww forms the right or left eigenvectors of the real generalized eigenvalue problem $Ax = \lambda Bx$, by backward transformation on the computed eigenvectors given by f08yx. It is necessary to call this function only if the optional balancing function f08wv was previously called to balance the matrix pair $(A, B)$.

## 2 Syntax

```
[v, info] = f08ww(job, side, ilo, ihi, lscale, rscale, v, 'n', n, 'm',
m)
```

## 3 Description

If the matrix pair has been previously balanced using the function f08wv then f08ww backtransforms the eigenvector solution given by f08yx. This is usually the sixth and last step in the solution of the generalized eigenvalue problem.

For a description of balancing, see the document for f08wv.

## 4 References

Ward R C 1981 Balancing the generalized eigenvalue problem *SIAM J. Sci. Stat. Comp.* **2** 141–152

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **job – string**

Specifies the backtransformation step required.

**job** = 'N'

No transformations are done.

**job** = 'P'

Only do backward transformations based on permutations.

**job** = 'S'

Only do backward transformations based on scaling.

**job** = 'B'

Do backward transformations for both permutations and scaling.

**Note:** this must be identical to the parameter **job** as supplied to f08wv.

*Constraint*: **job** = 'N', 'P', 'S' or 'B'.

2: **side – string**

Indicates whether left or right eigenvectors are to be transformed.

**side** = 'L'

The left eigenvectors are transformed.

**side** = 'R'

The right eigenvectors are transformed.

*Constraint*: **side** = 'L' or 'R'.

3: **ilo – int32 scalar**

4: **ihi – int32 scalar**

$i_{\mathrm{lo}}$ and $i_{\mathrm{hi}}$ as determined by a previous call to f08wv.

*Constraints*:

if $\mathbf{n} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{n}$;
if $\mathbf{n} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$.

5: **lscale**$(*)$ **– double array**

**Note**: the dimension of the array **lscale** must be at least $\max(1, \mathbf{n})$.

Details of the permutations and scaling factors applied to the left side of the matrices $A$ and $B$, as returned by a previous call to f08wv.

6: **rscale**$(*)$ **– double array**

**Note**: the dimension of the array **rscale** must be at least $\max(1, \mathbf{n})$.

Details of the permutations and scaling factors applied to the right side of the matrices $A$ and $B$, as returned by a previous call to f08wv.

7: **v**(**ldv**,$*$) **– complex array**

The first dimension of the array **v** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{m})$

The matrix of right or left eigenvectors, as returned by f08wv.

## 5.2 Optional Input Parameters

1: **n – int32 scalar**

*Default*: The first dimension of the array **v**.

$n$, the order of the matrices $A$ and $B$ of the generalized eigenvalue problem.

*Constraint*: $\mathbf{n} \geq 0$.

2: **m – int32 scalar**

*Default*: The second dimension of the array **v**.

$m$, the required number of left or right eigenvectors.

*Constraint*: $0 \leq \mathbf{m} \leq \mathbf{n}$.

## 5.3 Input Parameters Omitted from the MATLAB Interface

ldv

## 5.4 Output Parameters

1: **v**(**ldv**,$*$) **– complex array**

The first dimension of the array **v** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{m})$

The transformed right or left eigenvectors.

2: **info – int32 scalar**

**info** $= 0$ unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**info** $= -i$

If **info** $= -i$, parameter $i$ had an illegal value on entry. The parameters are numbered as follows: 1: **job**, 2: **side**, 3: **n**, 4: **ilo**, 5: **ihi**, 6: **lscale**, 7: **rscale**, 8: **m**, 9: **v**, 10: **ldv**, 11: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

## 7 Accuracy

The errors are negligible.

## 8 Further Comments

The number of operations is proportional to $n^2$.

The real analogue of this function is f08wj.

## 9 Example

```
job = 'B';
side = 'R';
ilo = int32(1);
ihi = int32(4);
lscale = [1;
     1;
     0.1;
     0.1];
rscale = [1;
     1;
     0.1;
     0.1];
v       =       [complex(0.08699119110808952,       -0.1955405777103713),
complex(0.05499605789738971, ...
              +0.03184570466928372),    complex(-0.5392491646518425,    -
0.26969107394004), ...
     complex(0.04674934655727678, -0.05969756780599979);
           complex(-0.1298236544718471,  +0.1445968878107905),  complex(-
0.106007923400722, ...
                    -0.07052699695181323),     complex(0.6026933129913244,
+0.1759518874430412), ...
     complex(-0.08013964520890554, +0.09557046409891881);
                      complex(0.4797322755076589,    -0.520267724492341),
complex(0.6391835562004952, ...
                +0.3608164437995049),    complex(-0.726493750928186,    -
0.273506249071814), ...
     complex(0.5616063942288206, -0.4383936057711793);
          complex(-0.06890678406189565,  +0.0906228238958222),  complex(-
0.1388422329463264, ...
```

```
              -0.03029648763481868),   complex(-0.04222304821712557,   -
0.00652456755391087), ...
     complex(-0.1292547111769504, +0.04201677539294552)];
[vOut, info] = f08ww(job, side, ilo, ihi, lscale, rscale, v)
```

```
vOut =
    0.0870 - 0.1955i    0.0550 + 0.0318i   -0.5392 - 0.2697i    0.0467 -
0.0597i
   -0.1298 + 0.1446i   -0.1060 - 0.0705i    0.6027 + 0.1760i   -0.0801 +
0.0956i
    0.0480 - 0.0520i    0.0639 + 0.0361i   -0.0726 - 0.0274i    0.0562 -
0.0438i
   -0.0069 + 0.0091i   -0.0139 - 0.0030i   -0.0042 - 0.0007i   -0.0129 +
0.0042i
info =
           0
```