We develop here an algorithm based on the Llewellyn Thomas algorithm for tridiagonal matrices for solving banded cyclic/periodic matrices.

Banded matrices can arise from finite difference methods, eg. :

$a_j z_{j-1} + b_j z_j + c_j z_{j+1} = d_j$

for a 3-point difference equation or

$a_j z_{j-2} + b_j z_{j-1} + c_j z_j + d_j z_{j+1} + e_j z_{j+2} = f_j$

for a 5-point formula.

A periodic version of a banded matrix can come from periodic boundary conditions. In these cases, none of the coefficients above are zero, whereas in a “regular” banded matrix, the boundary conditions make the off diagonal terms zero.

For the first example, the three (m=3) point equation $a_j z_{j-1} + b_j z_j + c_j z_{j+1} = d_j$ can be written as a cyclic or periodic version of the tridiagonal matrix with non-zero out of band elements:

$$\begin{bmatrix} b_1 & c_1 & . & .\,. & . & 0 & a_1 \\ a_2 & b_2 & c_2 & .\,. & . & . & 0 \\ . & . & . & .\,. & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & 0 & . & .\,. & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} z_1 \\ . \\ . \\ z_n \end{bmatrix} = \begin{bmatrix} d_1 \\ . \\ . \\ d_n \end{bmatrix}$$ i.e. tridiagonal except for two off-diagonal terms.

We will develop three versions of an algorithm for solving these systems. First, the system can be solved in O[n] with what we shall call the forward algorithm.

Forward algorithm:

Starting with the 1st and nth equations and working inwards to the 2nd and n-1st an so on, for the jth pair of equations we can write, for the three point equations:

for j:{2,3,4…} 1 < j < n/2

$$\begin{bmatrix} b_j & 0 \\ 0 & b_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & 0 \\ 0 & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j-1} \\ z_{n-j+2} \end{bmatrix} = \begin{bmatrix} d_j \\ d_{n-j+1} \end{bmatrix} - \begin{bmatrix} c_j & 0 \\ 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+1} \\ z_{n-j} \end{bmatrix}$$

which we rewrite as

$$\begin{bmatrix} b_j & 0 \\ 0 & b_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & 0 \\ 0 & c_{n-j+1} \end{bmatrix} \left\{ \tilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \vec{v}_{j-1} \right\} = \begin{bmatrix} d_j \\ d_{n-j+1} \end{bmatrix} - \begin{bmatrix} c_j & 0 \\ 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+1} \\ z_{n-j} \end{bmatrix}$$

for j:{1,2,3,4…} 0 < j < n/2

defining $\begin{bmatrix} z_{j-1} \\ z_{n-j+2} \end{bmatrix} = \left\{ \tilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \vec{v}_{j-1} \right\}$ for j:{2,3,4…} 1 < j < n/2

then we have

$$\begin{bmatrix} b_j & 0 \\ 0 & b_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & 0 \\ 0 & c_{n-j+1} \end{bmatrix} \left\{ \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \vec{v}_{j-1} \right\} = \begin{bmatrix} d_j \\ d_{n-j+1} \end{bmatrix} - \begin{bmatrix} c_j & 0 \\ 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+1} \\ z_{n-j} \end{bmatrix}$$

and $\widetilde{A}_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \vec{v}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ for j=1 because at j=1:

$$\begin{bmatrix} b_1 & 0 \\ 0 & b_n \end{bmatrix} \begin{bmatrix} z_1 \\ z_n \end{bmatrix} + \begin{bmatrix} a_1 & 0 \\ 0 & c_n \end{bmatrix} \begin{bmatrix} z_n \\ z_1 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_n \end{bmatrix} - \begin{bmatrix} c_1 & 0 \\ 0 & a_n \end{bmatrix} \begin{bmatrix} z_2 \\ z_{n-1} \end{bmatrix}$$

Each pair is solved for in terms of the inner next pair of values. Using the first equation with j=1 yields the result for the second equation for j=2 by solving a 2x2 system of equations. Using the matrix $\widetilde{A}_1$ and vector $\vec{v}_1$ the process is iterated repeatedly almost n/2 times, solving for all the $\widetilde{A}_j$ and $\vec{v}_j$

At the center of the matrix we have to solve either a 2x2 system of equations if n is even or a 3x3 system if n is odd. If n is even the final equation becomes, with j=n/2:

$$\begin{bmatrix} b_{n/2} & 0 \\ 0 & b_{n/2+1} \end{bmatrix} \begin{bmatrix} z_{n/2} \\ z_{n/2+1} \end{bmatrix} + \begin{bmatrix} a_{n/2} & 0 \\ 0 & c_{n/2+1} \end{bmatrix} \left\{ \widetilde{A}_{n/2-1} \begin{bmatrix} z_{n/2} \\ z_{n/2+1} \end{bmatrix} + \vec{v}_{n/2-1} \right\} = \begin{bmatrix} d_{n/2} \\ d_{n/2+1} \end{bmatrix} - \begin{bmatrix} c_{n/2} & 0 \\ 0 & a_{n/2+1} \end{bmatrix} \begin{bmatrix} z_{n/2+1} \\ z_{n/2} \end{bmatrix}$$

which is a 2x2 system solvable for the middle two values:

$$\begin{bmatrix} z_{n/2} \\ z_{n/2+1} \end{bmatrix} = \left\{ \begin{bmatrix} b_{n/2} & c_{n/2} \\ a_{n/2+1} & b_{n/2+1} \end{bmatrix} + \begin{bmatrix} a_{n/2} & 0 \\ 0 & c_{n/2+1} \end{bmatrix} \widetilde{A}_{n/2-1} \right\}^{-1} \left\{ \begin{bmatrix} d_{n/2} \\ d_{n/2+1} \end{bmatrix} - \begin{bmatrix} a_{n/2} & 0 \\ 0 & c_{n/2+1} \end{bmatrix} \vec{v}_{n/2-1} \right\}$$

If n is odd, the middle term is included:

$$\begin{bmatrix} b_j & c_j & 0 \\ a_{j+1} & b_{j+1} & c_{j+1} \\ 0 & a_{j+2} & b_{j+2} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+2} \end{bmatrix} = \begin{bmatrix} d_j \\ d_{j+1} \\ d_{j+2} \end{bmatrix} - \begin{bmatrix} a_j & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_{j+2} \end{bmatrix} \begin{bmatrix} z_{j-1} \\ 0 \\ z_{j+3} \end{bmatrix}$$

$$\begin{bmatrix} z_{j-1} \\ z_{j+3} \end{bmatrix} = \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+2} \end{bmatrix} + \vec{v}_{j-1}, j = \frac{(n-1)}{2}$$

With

$$\begin{bmatrix} z_{j-1} \\ 0 \\ z_{j+3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{j-1} \\ z_{j+3} \end{bmatrix} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \vec{v}_{j-1} \right\}$$ then

$$\begin{bmatrix} b_j & c_j & 0 \\ a_{j+1} & b_{j+1} & c_{j+1} \\ 0 & a_{j+2} & b_{j+2} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+2} \end{bmatrix} = \begin{bmatrix} d_j \\ d_{j+1} \\ d_{j+2} \end{bmatrix} - \begin{bmatrix} a_j & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_{j+2} \end{bmatrix} \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \vec{v}_{j-1} \right\}$$

or, writing the components of $\widetilde{A}_{j-1}$ explicitly:

$$\begin{bmatrix} b_j + a_j A_{j-1}^{11} & c_j & a_j A_{j-1}^{12} \\ a_{j+1} & b_{j+1} & c_{j+1} \\ c_{j+2} A_{j-1}^{21} & a_{j+2} & b_{j+2} + c_{j+2} A_{j-1}^{22} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+2} \end{bmatrix} = \begin{bmatrix} d_j - a_j v_{j-1}^1 \\ d_{j+1} \\ d_{j+2} - c_{j+2} v_{j-1}^2 \end{bmatrix}$$

We can then solve for all of the $z_j$ by back substitution using $\begin{bmatrix} z_{j-1} \\ z_{n-j+2} \end{bmatrix} = \left\{ \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \vec{v}_{j-1} \right\}$

Proceeding in complexity, the m=5 point scheme $a_j z_{j-2} + b_j z_{j-1} + c_j z_j + d_j z_{j+1} + e_j z_{j+2} = f_j$ results in

$$\begin{bmatrix} c_1 & d_1 & e_1 & . & . & . & a_1 & b_1 \\ b_2 & c_2 & d_2 & e_2 & . & . & . & a_2 \\ . & . & . & . & . & . & . & . \\ e_{n-1} & . & . & . & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\ d_n & e_n & . & . & . & a_n & b_n & c_n \end{bmatrix} \begin{bmatrix} z_1 \\ . \\ . \\ . \\ z_n \end{bmatrix} = \begin{bmatrix} f_1 \\ . \\ . \\ . \\ f_n \end{bmatrix}$$ with 6 off-diagonal non-zero elements.

We can then write, for any set of 4 equations j:

$$\begin{bmatrix} c_j & d_j & 0 & 0 \\ b_{j+1} & c_{j+1} & 0 & 0 \\ 0 & 0 & c_{n-j} & d_{n-j} \\ 0 & 0 & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{n-j} \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & b_j & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 \\ 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j-2} \\ z_{j-1} \\ z_{n-j+2} \\ z_{n-j+3} \end{bmatrix} = \begin{bmatrix} f_j \\ f_{j+1} \\ f_{n-j} \\ f_{n-j+1} \end{bmatrix} - \begin{bmatrix} e_j & 0 & 0 & 0 \\ d_{j+1} & e_{j+1} & 0 & 0 \\ 0 & 0 & a_{n-j} & b_{n-j} \\ 0 & 0 & 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+2} \\ z_{j+3} \\ z_{n-j-2} \\ z_{n-j-1} \end{bmatrix}$$

where we can define

$$\begin{bmatrix} z_{j-2} \\ z_{j-1} \\ z_{n-j+2} \\ z_{n-j+3} \end{bmatrix} = \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{n-j} \\ z_{n-j+1} \end{bmatrix} + \vec{v}_{j-1}$$ as before for j:{3,5,7…} j>1 and $\widetilde{A}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \vec{v}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ for j=1.

and then

$$\begin{bmatrix} c_j & d_j & 0 & 0 \\ b_{j+1} & c_{j+1} & 0 & 0 \\ 0 & 0 & c_{n-j} & d_{n-j} \\ 0 & 0 & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{n-j} \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & b_j & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 \\ 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} \left\{ \tilde{A}_j \begin{bmatrix} z_j \\ z_{j+1} \\ z_{n-j} \\ z_{n-j+1} \end{bmatrix} + \vec{v}_j \right\}$$

$$= \begin{bmatrix} f_j \\ f_{j+1} \\ f_{n-j} \\ f_{n-j+1} \end{bmatrix} - \begin{bmatrix} e_j & 0 & 0 & 0 \\ d_{j+1} & e_{j+1} & 0 & 0 \\ 0 & 0 & a_{n-j} & b_{n-j} \\ 0 & 0 & 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+2} \\ z_{j+3} \\ z_{n-j-2} \\ z_{n-j-1} \end{bmatrix}$$

We then have either 5,6 or 7 rows left after iterating the recurrence relations: for example with 6 rows, we have:

$$\begin{bmatrix} c_j & d_j & e_j & 0 & 0 & 0 \\ b_{j+1} & c_{j+1} & d_{j+1} & e_{j+1} & 0 & 0 \\ a_{j+2} & b_{j+2} & c_{j+2} & d_{j+2} & e_{j+2} & 0 \\ 0 & a_{n-j-1} & b_{n-j-1} & c_{n-j-1} & d_{n-j-1} & e_{n-j-1} \\ 0 & 0 & a_{n-j} & b_{n-j} & c_{n-j} & d_{n-j} \\ 0 & 0 & 0 & a_{n-j+1} & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+2} \\ z_{j+3} \\ z_{j+4} \\ z_{j+5} \end{bmatrix} = \begin{bmatrix} f_j \\ f_{j+1} \\ f_{j+2} \\ f_{j+3} \\ f_{j+4} \\ f_{j+5} \end{bmatrix} - \begin{bmatrix} a_j & b_j & 0 & 0 & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j-2} \\ z_{j-1} \\ 0 \\ 0 \\ z_{n-j+2} \\ z_{n-j+3} \end{bmatrix}$$

but

$$\begin{bmatrix} z_{j-2} \\ z_{j-1} \\ 0 \\ 0 \\ z_{n-j+2} \\ z_{n-j+3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_{j-2} \\ z_{j-1} \\ z_{n-j+2} \\ z_{n-j+3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left\{ \tilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+4} \\ z_{j+5} \end{bmatrix} + \vec{v}_{j-1} \right\}$$

and so $$\begin{bmatrix} a_j & b_j & 0 & 0 & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_j & b_j & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 \\ 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and then

$$\begin{bmatrix} c_j & d_j & e_j & 0 & 0 & 0 \\ b_{j+1} & c_{j+1} & d_{j+1} & e_{j+1} & 0 & 0 \\ a_{j+2} & b_{j+2} & c_{j+2} & d_{j+2} & e_{j+2} & 0 \\ 0 & a_{n-j-1} & b_{n-j-1} & c_{n-j-1} & d_{n-j-1} & e_{n-j-1} \\ 0 & 0 & a_{n-j} & b_{n-j} & c_{n-j} & d_{n-j} \\ 0 & 0 & 0 & a_{n-j+1} & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+2} \\ z_{j+3} \\ z_{j+4} \\ z_{j+5} \end{bmatrix}$$

$$= \begin{bmatrix} f_j \\ f_{j+1} \\ f_{j+2} \\ f_{j+3} \\ f_{j+4} \\ f_{j+5} \end{bmatrix} - \begin{bmatrix} a_j & b_j & 0 & 0 & 0 & 0 \\ 0 & a_{j+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{n-j} & 0 \\ 0 & 0 & 0 & 0 & d_{n-j} & e_{n-j+1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left\{ \widetilde{A}_{j-1} \begin{bmatrix} z_j \\ z_{j+1} \\ z_{j+4} \\ z_{j+5} \end{bmatrix} + \vec{v}_{j-1} \right\}$$

We can workout similar formulae for 5 and 7 rows left, with:

$$\begin{bmatrix} c_j & d_j & e_j & 0 & 0 \\ b_{j+1} & c_{j+1} & d_{j+1} & e_{j+1} & 0 \\ a_{(n+1)/2} & b_{(n+1)/2} & c_{(n+1)/2} & d_{(n+1)/2} & e_{(n+1)/2} \\ 0 & a_{n-j} & b_{n-j} & c_{n-j} & d_{n-j} \\ 0 & 0 & a_{n-j+1} & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \quad \text{with 5 rows}$$

and

$$\begin{bmatrix} c_j & d_j & e_j & 0 & 0 & 0 & 0 \\ b_{j+1} & c_{j+1} & d_{j+1} & e_{j+1} & 0 & 0 & 0 \\ a_{j+2} & b_{j+2} & c_{j+2} & d_{j+2} & e_{j+2} & 0 & 0 \\ 0 & a_{(n+1)/2} & b_{(n+1)/2} & c_{(n+1)/2} & d_{(n+1)/2} & e_{(n+1)/2} & 0 \\ 0 & 0 & a_{n-j-1} & b_{n-j-1} & c_{n-j-1} & d_{n-j-1} & e_{n-j-1} \\ 0 & 0 & 0 & a_{n-j} & b_{n-j} & c_{n-j} & d_{n-j} \\ 0 & 0 & 0 & 0 & a_{n-j+1} & b_{n-j+1} & c_{n-j+1} \end{bmatrix} \quad \text{with 7 rows}$$

It all gets rather tedious. Writing more generally, if we write the first equation corresponding to the 3 point formula

$$\begin{bmatrix} b_j & 0 \\ 0 & b_{n-j+1} \end{bmatrix} \begin{bmatrix} z_j \\ z_{n-j+1} \end{bmatrix} + \begin{bmatrix} a_j & 0 \\ 0 & c_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j-1} \\ z_{n-j+2} \end{bmatrix} = \begin{bmatrix} d_j \\ d_{n-j+1} \end{bmatrix} - \begin{bmatrix} c_j & 0 \\ 0 & a_{n-j+1} \end{bmatrix} \begin{bmatrix} z_{j+1} \\ z_{n-j} \end{bmatrix}$$

as

$\widetilde{C}_{jj}\vec{z}_{jj}+\widetilde{S}_{jj}\vec{z_{jj-1}}=\vec{b}_{jj}-\widetilde{P}_{jj}\vec{z_{jj+1}}$ and defining $KU=\frac{(m-1)}{2}$

where $\vec{z}_{jj}$ is understood to be the corresponding vector with length 2 KU (or m-1) defined as

$$(z_j, z_{j+1}\ldots. z_{j+(m-3)/2}, z_{n-j+1-(m-3)/2}\ldots z_{n-j}, z_{n-j+1})=(z_j, z_{j+1}\ldots. z_{j+KU-1}, z_{n-j+2-KU}\ldots z_{n-j}, z_{n-j+1})$$

with $\vec{b}_{jj}$ defined similarly so that $\widetilde{C}_{jj}, \widetilde{S}_{jj}, \widetilde{P}_{jj}, \vec{b}_{jj}\vec{z}_{jj}$ are defined (see Appendix) for jj every j KU apart.

Substituting $\widetilde{A}_{jj}$ and $v_{jj}$ defined as:

$$\vec{z_{jj-1}}=\widetilde{A}_{jj-1}\vec{z}_{jj}+\vec{v}_{jj-1}$$

results in the second equation as

$$\{\widetilde{C}_{jj}+\widetilde{S}_{jj}\widetilde{A_{jj-1}}\}\vec{z}_{jj}+\widetilde{S}_{jj}\vec{v_{jj-1}}=\vec{b}_{jj}-\widetilde{P}_{jj}\vec{z_{jj+1}}$$

Using the definition of $\widetilde{A}_{jj}$ and $\vec{v}_{jj}$ and the second equation we can recognize the recurrence relations:

$$\widetilde{A}_{jj}=-\{\widetilde{C}_{jj}+\widetilde{S}_{jj}\widetilde{A_{jj-1}}\}^{-1}\widetilde{P}_{jj}$$

$$\vec{v}_{jj}=\{\widetilde{C}_{jj}+\widetilde{S}_{jj}\widetilde{A_{jj-1}}\}^{-1}\{\vec{b}_{jj}-\widetilde{S}_{jj}\vec{v_{jj-1}}\}$$

which are solved every jj, reducing the number of equations by (m-1) each iteration with $\widetilde{A}_0=\begin{bmatrix}0 & \widetilde{I_{(m-1)/2}}\\ \widetilde{I_{(m-1)/2}} & 0\end{bmatrix}$ an (m-1)x(m-1) orthogonal matrix and $\widetilde{I_{(m-1)/2}}$ the (m-1)/2x(m-1)/2 identity matrix.

At the center of the system matrix, if p=modulo(n,m-1) =0, then $\vec{z_{jj+1}}=\widetilde{A}_0\vec{z}_{jj}$ (recognizing that the final $\widetilde{A_{j-1}}=\{\widetilde{A}_0\}^{-1}$ )

so that the final inner values are given by

$$\vec{z}_{jj}=\{\widetilde{C}_{jj}+\widetilde{P}_{jj}\widetilde{A}_0+\widetilde{S}_{jj}\widetilde{A_{jj-1}}\}^{-1}\{\vec{b}_{jj}-\widetilde{S}_j\vec{v_{jj-1}}\}$$ for the last 2*(m-1) values at jj=n/2

More generally, if p= modulo(n,m-1) and $p\neq 0$ , we have m-1+p rows left. The central rows can be written

$$\vec{z}_L=\{\ z_j, z_{j+1}, \ldots. z_{m-1+p}\}$$

We can then state that these central values are given by:

$$\vec{z}_L=[\widetilde{C}_L+\widetilde{P}_L\widetilde{A}_0+\widetilde{I}_0^*S_{jj}\widetilde{A_{jj-1}}\widetilde{I}_0^{*T}]^{-1}\{\vec{b}_L-\widetilde{I}_0^*\widetilde{S}_{jj}\vec{v_{jj-1}}\}$$ or

$$\vec{z}_L = [\tilde{C}_{jjL} + \tilde{I}_0^* S_{jj} A_{\widetilde{jj-1}} \tilde{I}_0^{*T}]^{-1} \{ \vec{b}_L - \tilde{I}_0^* \tilde{S}_{jj} v_{\overrightarrow{jj-1}} \}$$

where $\tilde{C}_{jjL} = \tilde{C}_L + \tilde{P}_L \tilde{A}_0$ and $\vec{b}_L$ consists of the last 2*KU+p inner values.

$\tilde{I}_0^*$ is an (m-1) x (m-1+p) orthonormal matrix composed of the (m-1) x (m-1) identity matrix with p=modulo(n,m-1) extra zero rows . Note we use the fact that $\tilde{I}_0^* \tilde{I}_0^{*T} = I_{\widetilde{m-1+p}}$ that for the central rows where : $\tilde{I}_0^* \vec{z}_L = \tilde{I}_0^* \{ z_j, z_{j+1}, .... z_{j+m-1+p} \}^T = \tilde{I}_0^* \vec{z}_j =$
$\tilde{I}_0^* \{ z_j, z_{j+1} .... z_{j+(m-3)/2}, 0, 0, ..0, z_{n-j+1-(m-3)/2} ... z_{n-j}, z_{n-j+1} \}^T$

In summary, the forward algorithm consists of computing the matrix and vector every (m-1) sets of rows with

$$\tilde{A}_{jj} = -\{ \tilde{C}_{jj} + \tilde{S}_{jj} A_{\widetilde{jj-1}} \}^{-1} \tilde{P}_{jj}$$

$$\vec{v}_{jj} = \{ \tilde{C}_{jj} + \tilde{S}_{jj} A_{\widetilde{jj-1}} \}^{-1} \{ \vec{b}_{jj} - \tilde{S}_{jj} v_{\overrightarrow{jj-1}} \}$$

for n/(m-1) number of times. (integer division)

then computing the m-1+p central values needing only $A_{\widetilde{jj-1}}$ and $v_{\overrightarrow{jj-1}}$

$$\vec{z}_L = [\tilde{C}_L + \tilde{P}_L \tilde{A}_0 + \tilde{I}_0^* S_{jj} A_{\widetilde{jj-1}} \tilde{I}_0^{*T}]^{-1} \{ \vec{b}_L - \tilde{I}_0^* \tilde{S}_{jj} v_{\overrightarrow{jj-1}} \}$$

or

$$\vec{z}_L = [\tilde{C}_{jjL} + \tilde{I}_0^* S_{jj} A_{\widetilde{jj-1}} \tilde{I}_0^{*T}]^{-1} \{ \vec{b}_L - \tilde{I}_0^* \tilde{S}_{jj} v_{\overrightarrow{jj-1}} \}$$

where $\tilde{C}_{jjL} = \tilde{C}_L + \tilde{P}_L \tilde{A}_0$ and $\vec{b}_L$ consists of the last 2*KU+p inner values.

where $\tilde{I}_0^*$ is an (m-1) x (m-1+p) matrix composed of the (m-1) x (m-1) identity matrix with p=modulo(n,m-1) extra central zero rows , and $\tilde{A}_0$ is the (m-1) x (m-1) circular permutation matrix composed of the identity matrix from above.

Once the center values are known the stored matrices $\tilde{A}_{jj}$ and vectors $\vec{v}_{jj}$ are used for back substitution using $z_{\overrightarrow{jj-1}} = \tilde{A}_{jj-1} \vec{z}_{jj} + \vec{v}_{jj-1}$ .

Reverse algorithm:

We can also solve the equations in reverse. Defining $z_{\overrightarrow{jj+1}} = \bar{A}_{jj+1} \vec{z}_{jj} + \bar{v}_{jj+1}$ then our equations are

$$\{ \tilde{C}_{jj} + \tilde{P}_{jj} A_{\overline{jj+1}} \} \vec{z}_{jj} + \tilde{P}_{jj} v_{\overrightarrow{jj+1}} = \vec{b}_{jj} - \tilde{S}_{jj} z_{\overrightarrow{jj-1}}$$ leading to

$$\bar{A}_{jj} = -\left\{\widetilde{C}_{jj} + \widetilde{P}_{jj} \bar{A}_{jj+1}\right\}^{-1} \widetilde{S}_{jj}$$

$$\bar{v}_{jj} = \left\{\widetilde{C}_{jj} + \widetilde{P}_{jj} \bar{A}_{jj+1}\right\}^{-1} \left\{\vec{b}_{jj} - \widetilde{P}_{jj} \bar{v}_{jj+1}\right\}$$

Instead of starting at jj=0, we start at jj=(n-p)/2, and calculate each decreasing jj starting with $\bar{A}_{(n-p)/2+1}$ and $\bar{v}_{(n-p)/2+1}$, which we will define shortly.

At the beginning and end, or edges, of the system matrix, $\vec{z}_{jj-1} = \bar{A}_0 \vec{z}_{jj}$ (recognizing that the final $\bar{A}_j = \{\bar{A}_0\}^{-1}$ ) so that the last 2*(m-1) values at jj=1 are solved for in terms of jj=2 by

$$\vec{z}_1 = \left\{\widetilde{C}_1 + \widetilde{S}_1 \bar{A}_0 + \widetilde{P}_1 \bar{A}_2\right\}^{-1} \left\{\vec{b}_1 - \widetilde{P}_1 \bar{v}_2\right\}$$

The 2 x 2 version is given below as an example:

$$\begin{bmatrix} z_1 \\ z_n \end{bmatrix} = \begin{bmatrix} b_1 & a_1 \\ c_n & b_n \end{bmatrix}^{-1} \left\{ \begin{bmatrix} d_1 \\ d_n \end{bmatrix} - \begin{bmatrix} c_1 & 0 \\ 0 & a_n \end{bmatrix} \begin{bmatrix} z_2 \\ z_{n-1} \end{bmatrix} \right\}$$

The rest of the values are then solved by back substitution with $\vec{z}_{jj+1} = \bar{A}_{jj+1} \vec{z}_{jj} + \bar{v}_{jj+1}$ .

Here we have implicitly decided to start the iterative solution at jj=(n-p)/2. If p=modulo(n,m-1) =0, then we have the 2KU x 2KU matrix:

$$\bar{A}_{(n/2+1)} = \begin{bmatrix} 0 & \widetilde{I}_{KU} \\ \widetilde{I}_{KU} & 0 \end{bmatrix}, \bar{v}_{(n/2+1)} = \vec{0} \quad \text{because} \quad \vec{z}_{n/2+1} = \widetilde{A}_0 \vec{z}_{n/2}$$

If $p \neq 0$ , we have p additional values of $\vec{z}_{jj}$ to solve for, and the definitions of $\bar{A}_{(n-p)/2+1}$ and $\bar{v}_{(n-p)/2+1}$ are more complex. There are two cases, p > KU or p < KU, with both reducing to the same solution at p=KU. Consider the central rows:

$\vec{z}_p = \{ z_{(n-p)/2+1}, z_{(n-p)/2+2}, \ldots. z_{(n-p)/2+p} \}$ a vector p in length vs the definition of $\vec{z}_j$ at j=(n-p)/2+1:

$$\vec{z}_{(n-p)/2+1} = \left( z_{(n-p)/2+1}, z_{(n-p)/2+2} \ldots. z_{(n-p)/2+KU}, z_{(n+p)/2+3-KU} \ldots z_{(n+p)/2+1}, z_{(n+p)/2+2} \right)$$

If p > KU, then vector $\vec{z}_p$ is longer than each half of vector $\vec{z}_{(n-p)/2+1}$ so that there is a duplication of the middle 2KU-p rows.

If p < KU, vector $\vec{z}_p$ is contained within each half of vector $\vec{z}_{(n-p)/2+1}$ and an additional 2KU-2p rows are required.

If p=0, the 2KU rows (and columns) are generated by $\bar{A}_{(n)/2+1} = \bar{A}_0$ given above.

The p rows are generated by the central p equations given by $\tilde{C}_p \vec{z}_p + \tilde{S}_p \vec{z_{(n-p)/2}} = \vec{b}_p$ where $\vec{z}_p, \vec{b}_p$ consists of the middle p inner values and $\tilde{C}_p, \tilde{S}_p$ are p x p and p x 2KU matrices.

Using the relationship between vector $\vec{z}_p$ and vector $\vec{z_{(n-p)/2+1}}$, and $\vec{z}_{(n-p)/2+1} = \bar{A}_{(n-p)/2+1} \vec{z}_{(n-p)/2} + \bar{v}_{(n-p)/2+1}$, we can define $\bar{A}_{(n-p)/2+1}$ and $\bar{v}_{(n-p)/2+1}$.

If p < KU, the first p rows of $\bar{A}_{(n-p)/2+1}$ are given by the first p rows of $-C_p^{-1} S_p$ and the last p rows of $\bar{A}_{(n-p)/2+1}$ are given by the last p rows of $-C_p^{-1} S_p$. The center of the matrix is a 2KU-2p version of $\bar{A}_0$.

If p > KU, then first KU rows of $\bar{A}_{(n-p)/2+1}$ are given by the first KU rows of $-C_p^{-1} S_p$ and the last KU rows of $\bar{A}_{(n-p)/2+1}$ are given by the last KU rows of $-C_p^{-1} S_p$, resulting in a duplication of the middle 2KU-p rows. $\bar{v}_{(n-p)/2+1}$ is similarly defined, using $C_p^{-1} b_p$ and $\bar{v}_{(n)/2}$.

Parallel algorithm:

Although the algorithms developed are recursive and inherently serial, we can solve both forward and back algorithms in parallel, cutting down the computation time.

Note that by definition $\vec{z}_{jj+1} = \bar{A}_{jj+1} \vec{z}_{jj} + \bar{v}_{jj+1}$

Now, recalling that $\vec{z}_{jj-1} = \tilde{A}_{jj-1} \vec{z}_{jj} + \vec{v}_{jj-1}$ which is also $\vec{z}_{jj} = \tilde{A}_{jj} \vec{z}_{jj+1} + \vec{v}_{jj}$, then inverting leads to $\vec{z}_{jj+1} = \tilde{A}_{jj}^{-1} \vec{z}_{jj} - \tilde{A}_{jj}^{-1} \vec{v}_{jj}$

Note however despite that:
$\tilde{A}_{jj}^{-1} \neq \bar{A}_{jj+1}$ and $\tilde{A}_{jj}^{-1} \vec{v}_{jj} \neq -\bar{v}_{jj+1}$.

We can use this lack of equality to solve the iteration equations simultaneously forward and backward, starting from the first and last set of z at the edges, jj=1 and also starting from the middle of the matrix at kk=(n-p)/2. Proceeding iteratively from either direction eventually we will be at some value jj=kk+1, and we can solve for the values of z there, then back substitute simultaneously in both directions to solve for the all the values. It is straightforward to show that at any value jj:

$$\vec{z}_{jj} = \left( \tilde{I} - \bar{A}_{jj} A_{\widetilde{jj-1}} \right)^{-1} \left( \bar{A}_{jj} v_{\vec{jj-1}} + \bar{v}_{jj} \right)$$

and

$$z_{\vec{jj-1}} = \left( \tilde{I} - A_{\widetilde{jj-1}} \bar{A}_{jj} \right)^{-1} \left( A_{\widetilde{jj-1}} \bar{v}_{jj} + v_{\vec{jj-1}} \right)$$

We can pick jj=(n/4) so that each iterative series takes half as many steps as either the usual forward or backward algorithm. As a bonus, we only have to do the $p \neq 0$ equations for the reverse algorithm. Since each iterative series is independent, the computation can be done in parallel. Each step requires a solution of a KU x KU system, (for overall O(n x KU x KU) time) but the price of splitting the calculations in two parallel segments is solving one 4KU x 4KU system at the center.

Extended Parallel algorithm using rotation:

Let $\widetilde{J}_n$ be the n x n exchange matrix, eg. $\widetilde{J}_4=\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}$ and the permutation/rotation matrix $\widetilde{A}_n^k=\begin{bmatrix}0&\widetilde{I}_k\\ \widetilde{I}_{n-k}&0\end{bmatrix}$ where $\widetilde{A}_n^k$ is an n x n matrix with unequal k x k and n-k x n-k identity matrices. Note that when k=0 or n $\widetilde{A}_n^k$ is the identity matrix, and with k=n/2=KU resulting in $\widetilde{A}_0$ , the (m-1) x (m-1) circular permutation matrix, or the p=0 case of $\bar{A}_{(n/2+1)}$ .

If we apply $\widetilde{J}_n\widetilde{A}_n^k$ to $\vec{b}_j$ defined as in the m-point equation, with m-odd, and n data points: $\sum_{j=1}^{n} a_{ij}z_j=b_i$ letting $i, j$ run from 1 to n. (N.B. which is not $\vec{b}_{jj}$ ), then $\sum_{j=1}^{n} a_{ij}\hat{z}_j=\widetilde{J}_n\widetilde{A}_n^k b_i$ solves the equations with $\hat{\vec{z}}$ using the rotated k indices so that $\widetilde{J}_n\widetilde{A}_n^k\hat{\vec{z}}=\vec{z}$ .

It can be seen that if we use k=(n-p)/2, and start with the p=0 case, then $\hat{z}_j$ is nearly the same as $z_j$ proceeding from the reverse direction, so that running the forward algorithm on $\hat{z}_j$ is nearly the same as running the backward algorithm on $z_j$ and vice versa, since we have reversed the direction of indices and solution as well as rotating the indices. What does “nearly the same” mean? Let us see.

Rewriting the parallel algorithm, we write the “reverse solution” $\hat{z}_{jj+1}=\bar{A}_{jj+1}\hat{z}_{jj}+\bar{v}_{jj+1}$ and the “forward solution” $\vec{z}_{jj}=\widetilde{A}_{jj}\vec{z}_{jj+1}+\vec{v}_{jj}$ . Here we have not rotated the indices of $\hat{z}_j$ i.e. $\hat{\vec{z}}_{jj}=\vec{z}_{jj}$ , leading to a 2KU solution for each $\vec{z}_{jj}$ and $\vec{z}_{jj+1}$ , written as:

$$\begin{bmatrix}\bar{A}_{jj+1}&-\widetilde{I}_{m-1}\\ \widetilde{I}_{m-1}&-\widetilde{A}_{jj}\end{bmatrix}\begin{bmatrix}\vec{z}_{jj}\\ \vec{z}_{jj+1}\end{bmatrix}=\begin{bmatrix}-\bar{v}_{jj+1}\\ \vec{v}_{jj}\end{bmatrix}$$ the solution given in the previous section.

If we make a rotated version of the system with k=(n-p)/2, and considering that there are $\Omega=(n-p)/(2KU)$ blocks of $z_{jj}$ to solve, we can write the a new rotated coordinate system as $\hat{\vec{z}}_{\Omega-jj+1}=\widetilde{J}_{2KU}\widetilde{A}_{2KU}^{KU}\vec{z}_{jj}$ with $\vec{z}_{jj}=\widetilde{A}_{jj}\vec{z}_{jj+1}+\vec{v}_{jj}$ as before, but defining $\hat{\vec{z}}_{jj}=\widetilde{B}_{jj}\hat{\vec{z}}_{jj+1}+\vec{u}_{jj}$ as the forward algorithm for the rotated system. ( $\widetilde{C}_{jj}^r, \widetilde{S}_{jj}^r, \widetilde{P}_{jj}^r, \vec{b}_{jj}^r$ have to be redefined for the rotated and reversed system in order to compute $\widetilde{B}_{jj}, \vec{u}_{jj}$ , see the Appendix)

We can then write a parallel system with two forward algorithms as

$$\begin{bmatrix}-\widetilde{B}_{jj}\widetilde{J}_{2KU}\widetilde{A}_{2KU}^{KU}&\widetilde{J}_{2KU}\widetilde{A}_{2KU}^{KU}\\ \widetilde{I}_{2KU}&-\widetilde{A}_{jj}\end{bmatrix}\begin{bmatrix}\vec{z}_{jj}\\ \vec{z}_{jj+1}\end{bmatrix}=\begin{bmatrix}\vec{u}_{jj}\\ \vec{v}_{jj}\end{bmatrix}$$ which we then solve, thereafter using back substitution to retrieve the rest of the $\vec{z}_{jj}$ and $\hat{z}_{jj}$ . Or better yet, we can use two backward algorithms, which should allow for the $p\neq 0$ case.

If we make a rotation, for example at k=(n-p)/4 with a "forward and "backward" algorithm, we can notate the rotated and reversed system as follows:

$$\vec{w}_{jj+1} = \bar{B}_{jj+1}\vec{w}_{jj} + \bar{u}_{jj+1} \quad \text{and} \quad \hat{\vec{w}}_{jj} = \tilde{B}_{jj}\hat{\vec{w}}_{jj+1} + \vec{u}_{jj} \quad \text{where} \quad \vec{c}_j = \tilde{J}_n \tilde{A}_n^k \vec{b}_j$$
$$c_{jj} = (c_j, c_{j+1} \ldots . c_{j+(m-3)/2}, c_{n-j+1-(m-3)/2} \ldots c_{n-j}, c_{n-j+1})$$
$$\bar{B}_{jj} = -\{\tilde{C}^r_{jj} + \tilde{P}^r_{jj}\bar{B}_{jj+1}\}^{-1}\tilde{S}^r_{jj} \qquad \bar{u}_{jj} = \{\tilde{C}^r_{jj} + \tilde{P}^r_{jj}\bar{B}_{jj+1}\}^{-1}\{\vec{c}_{jj} - \tilde{P}^r_{jj}\bar{u}_{jj+1}\}$$
$$\tilde{B}_{jj} = -\{\tilde{C}^r_{jj} + \tilde{S}^r_{jj}\tilde{B}_{jj-1}\}^{-1}\tilde{P}^r_{jj} \qquad \vec{u}_{jj} = \{\tilde{C}^r_{jj} + \tilde{S}^r_{jj}\tilde{B}_{jj-1}\}^{-1}\{\vec{c}_{jj} - \tilde{S}^r_{jj}\vec{u}_{jj-1}\}$$

Consider the two KU block of numbers which the first and second piece of $\vec{z}_{jj}$ :
$(z_j, z_{j+1} \ldots . z_{j+(m-3)/2}), (z_{n-j+1-(m-3)/2} \ldots z_{n-j}, z_{n-j+1})$

By aligning the components of the vectors , we can simultaneously solve for all four vector components corresponding to the same two KU block of numbers $z_{jj}$ . Given a rotation $\theta$ (in blocks jj, not indices j) e.g. $\theta = (n-p)/4\,KU$ the corresponding sets of blocks for any jj are then $z_{jj}, \hat{z}_{\Omega-jj+1}, w_{\theta-jj}, \hat{w}_{\theta+jj+1}$

We can then recognize that because $\vec{w}_{\theta-jj}$ and $\hat{\vec{w}}_{\theta+jj+1}$ are made of the same four blocks, though in reverse order, and transposed, so that if

$$\vec{z}_{jj} = (z_j, z_{j+1} \ldots . z_{j+(m-3)/2}), (z_{n-j+1-(m-3)/2} \ldots z_{n-j}, z_{n-j+1})$$
$$\hat{\vec{z}}_{\Omega-jj+1} = \hat{\vec{z}}_{ii} = (z_i, z_{i+1} \ldots . z_{i+(m-3)/2}), (z_{n-i+1-(m-3)/2} \ldots z_{n-i}, z_{n-i+1})$$

where i then corresponds to ii just as j corresponds to jj then

$$\vec{w}_{\theta-jj} = \begin{bmatrix} \tilde{J}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} \vec{z}_j + \begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{J}_{KU} & \tilde{0}_{KU} \end{bmatrix} \hat{\vec{z}}_i$$

results in

$$\hat{\vec{w}}_{\theta+jj+1} = \begin{bmatrix} \tilde{0}_{KU} & \tilde{J}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} \vec{z}_j + \begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{J}_{KU} \end{bmatrix} \hat{\vec{z}}_i$$

$$\vec{w}_{\theta-jj} = (z_{j+(m-3)/2} \ldots z_{j+1}, z_j), (\hat{z}_{i+(m-3)/2} \ldots \hat{z}_{i+1}, \hat{z}_i)$$
$$\hat{\vec{w}}_{\theta+jj-1} = (z_{j+(m-3)/2} \ldots z_{j+1}, z_j), (\hat{z}_{i+(m-3)/2} \ldots \hat{z}_{i+1}, \hat{z}_i)(\hat{z}_{n-j+1}, \hat{z}_{n-j} \ldots \hat{z}_{n-j+1-(m-3)/2}), (z_{n-i+1}, z_{n-i} \ldots z_{n-i+1-(m-3)/2})$$

We can then substitute into the above equations and solve, at any jj. Keeping as to close to (n-p)/8KU to minimize each branch's number of steps, the 8KU x 8KU system of equations:

$$\begin{bmatrix} \bar{A}_{jj+1} & -\tilde{I}_{2KU} & \tilde{0}_{2KU} & \tilde{0}_{2KU} \\ \bar{B}_{jj+1}\begin{bmatrix} \tilde{J}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} & -\begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{J}_{KU} & \tilde{0}_{KU} \end{bmatrix} & \bar{B}_{ii+1}\begin{bmatrix} \tilde{0}_{KU} & \tilde{J}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} & -\begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{J}_{KU} \end{bmatrix} \\ \begin{bmatrix} \tilde{J}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} & -\tilde{B}_{jj}\begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{J}_{KU} & \tilde{0}_{KU} \end{bmatrix} & \begin{bmatrix} \tilde{0}_{KU} & \tilde{J}_{KU} \\ \tilde{0}_{KU} & \tilde{0}_{KU} \end{bmatrix} & -\tilde{B}_{ii}\begin{bmatrix} \tilde{0}_{KU} & \tilde{0}_{KU} \\ \tilde{0}_{KU} & \tilde{J}_{KU} \end{bmatrix} \\ \tilde{0}_{2KU} & \tilde{0}_{2KU} & \tilde{I}_{2KU} & -\tilde{A}_{ii} \end{bmatrix} \begin{bmatrix} \vec{z}_{jj} \\ \vec{z}_{jj+1} \\ \hat{\vec{z}}_{ii} \\ \hat{\vec{z}}_{ii+1} \end{bmatrix} = \begin{bmatrix} -\bar{v}_{jj+1} \\ -\bar{u}_{jj+1} \\ \vec{u}_{ii} \\ \vec{v}_{ii} \end{bmatrix}$$

We back substitute $\vec{z}_{jj}$ , $\vec{z_{jj+1}}$ , $\hat{\vec{z}}_{ii}$ and $\hat{\vec{z}}_{ii+1}$ to get the remaining values of $\vec{z}_{jj}$ and $\hat{\vec{z}}_{ii}$ , which provides roughly half of the $\vec{z}$ . Simultaneously we substitute $\vec{z}_{jj}$ , $\vec{z_{jj+1}}$ , $\hat{\vec{z}}_{jj}$ and $\hat{\vec{z}}_{jj+1}$ to get $\vec{w}_{jj}$ , $\vec{w}_{jj+1}$ , $\hat{\vec{w}}_{jj}$ and $\hat{\vec{w}}_{jj+1}$ and back substitute to compute the remaining $\vec{w}_{jj}$ and $\hat{\vec{w}}_{jj}$ Finally, we can take the $w_j$ and rotate their indices to recover the other half of the $z_j$ using $\vec{w}_j = \widetilde{J}_n \widetilde{A}_n^k \vec{z}_j$ . Since $\left[\widetilde{A}_n^k\right]^{-1} = \widetilde{A}_n^{n-k}$ and $\widetilde{J}_n^{-1} = \widetilde{J}_n$ , $\vec{z}_j = \widetilde{J}_n \widetilde{A}_n^{n-k} \vec{w}_j$ .

Each step (N/KU of them) requires a solution of a KU x KU system, (for overall O(N x KU) time) but the price of splitting the calculations in four parallel segments is solving one 8KU x 8KU system at the center.

One can imagine that rather than starting at 1, 2 or 4 locations, we could have any number of evenly spaced starts with rotation, though clearly keeping with powers of two allows for using the reverse and forward equations with fewer rotations. The number of starts is limited by the size of n versus the block size KU, with the central simultaneous equations becoming increasing more complicated.

For example, if KU = 1 (the tridiagonal system) and n=64, and if we wanted to keep p=0 for simplicity, we could consider only powers of 2:

With one thread there are 31 2x2 systems to solve, 1 end equation, and 31 back steps.
With two threads there are 15 2x2 systems, 2 simultaneous end equations (2x2 each) and 15 back steps.
With four threads there are 7 2x2 systems, 4 simultaneous end equations (2x2 each) and 7 back steps.
With eight threads there are 3 2x2 systems, 8 simultaneous end equations (2x2) and 3 back steps.
With sixteen threads there is 1 2x2 systems, 16 simultaneous end equations (2x2) and 1 back step.

With parallel threads, if the number of threads is t then the time is O((N*KU/t + t x t x KU x KU) , with the caveat that the t*t*KU*KU system makes for a much more complicated book keeping arrangement, and that initializing threads carries overhead as well. There is potential economy in more efficiently defining the rotated systems and their equations in code.

## Appendix: Definition of matrices

In general, an m-point equation, with m-odd, can be written:

in contrast with $\sum_{j=1}^{n} a_{ij} z_j = b_i$ letting $i, j$ run from 1 to N

$$\sum_{i=1}^{i=1+2KU} AB(i,j) * z(1+mod(N+i+j-2, N)) = b_j$$

where the AB matrix is m x N, with N the number of points, similar to the general LAPACK band matrix (which we shall call CD) (https://www.netlib.org/lapack/lug/node124.html) notation, again letting $i, j$ run from 1 to N:

$CD(KU+1+i-j, j) = a_{ij} \; with \; KU = (m-1)/2$ where the **columns** of A are the columns of CD and the diagonals of A are the rows of CD, except that we allow wrapping the periodic coefficients and have the **rows** of A as the columns of AB and the diagonals of A are the rows of AB:

$$AB(2KU+2-mod(N+KU+1+i-j),i)=a_{ij}\ with\ KU=(m-1)/2$$

However, letting $i$ run from 1 to 1+2*KU (or 1 to m), with j from 1 to N, AB can efficiently be generated by

$$AB(i,j)=aij(j,1+mod(N+i+j-2-KU,N))$$

contrasted with the LAPACK "CD" form:

$$CD(i+KL,j)=aij(1+mod(N+i+j-2-KU,N),j)$$

from which we can also derive a conversion formula:

$$CD(i+KL,j)=AB(m-i+1,1+mod(N+i+j+KU-m-1,N))$$

Note that in the non-periodic banded matrix, the coefficients CD(i,j)=0 when i+j-(m+1)/2 is < 1 or > N. AB can be seen to be the periodic version of CD without the extra KL rows, and where if CD encodes $a_{ij}$ then AB encodes the transpose $a_{ij}^T$

Then the m-point equation, with m-odd, can be written:

$$\tilde{C}_{jj}\vec{z}_{jj}+\tilde{S}_{jj}\vec{z}_{jj-1}=\vec{b}_{jj}-\tilde{P}_{jj}\vec{z}_{jj+1} \quad \text{and defining} \quad KU=\frac{(m-1)}{2}$$

where $\vec{z}_{jj}$ is understood to be the corresponding vector with length 2 KU (or m-1) defined as

$$(z_j, z_{j+1} \ldots . z_{j+(m-3)/2}, z_{n-j+1-(m-3)/2} \ldots z_{n-j}, z_{n-j+1})=(z_j, z_{j+1} \ldots . z_{j+KU-1}, z_{n-j+2-KU} \ldots z_{n-j}, z_{n-j+1})$$

with $\vec{b}_{jj}$ defined similarly so that $\tilde{C}_{jj}, \tilde{S}_{jj}, \tilde{P}_{jj}, \vec{b}_{jj} \vec{z}_{jj}$ are defined below for jj every j KU apart.

With the above definition of AB[i,j] we have the square (i x k) for each j matrices:

$$\tilde{C}_{jj}=\left\{\begin{array}{ll} AB[KU+k-i+1, j+i-1] & \forall i,k\{1,2\ldots KU\} \\ AB[KU+k-i+1, n-2*KU+i-j+1] & \forall i,k\{KU+1\ldots 2*KU\} \\ else\ 0 & \end{array}\right\}$$

$$\tilde{S}_{jj}=\left\{\begin{array}{ll} AB[1+k-i, j+i-1] & \forall i,k\{1,2\ldots KU\} \\ AB[2*KU+1+k-i, n-2*KU+i-j] & \forall i,k\{KU+1\ldots 2*KU\} \\ else\ 0 & \end{array}\right\}$$

$$\widetilde{P}_{jj}=\begin{cases} AB[2*KU+1+k-i,j+i-1] & \forall i,k\{1,2...KU\} \\ AB[1+k-i,n-2*KU+i-j] & \forall i,k\{KU+1...2*KU\} \\ else\ 0 \end{cases}$$

For the forward algorithm, the last Cjj and Pjj are (m-1+p) x (m-1+p), in combination we define CjjL generally even if $p\neq 0$

$$\widetilde{C}_{jjL}=\widetilde{C}_L+\widetilde{P}_L\widetilde{A}_0=\begin{cases} AB[KU+1+k-i,j+i-1] & \forall i,k\{1,2...2*KU+p\} \\ else\ 0 \end{cases}$$

if p=0, there are no extra rows:

$\widetilde{C}_{jjL}=\widetilde{C}_L+\widetilde{P}_L\widetilde{A}_0=\widetilde{C_{N/(2KU)}}+\widetilde{P_{N/(2KU)}}\widetilde{A}_0$ using the above definitions at the last j=(n-m+2)/2=(n+1)/2-KU

For the reverse algorithm, the middle p equations can be written in the following format: $\widetilde{C}_p\vec{z}_p+\widetilde{S}_p\vec{z_{(n-p)/2}}=\vec{b}_p$ where $\vec{z}_p,\vec{b}_p$ consists of the last p inner values and $\widetilde{C}_p,\widetilde{S}_p$ are the matrices given below with dimensions p x p and p x 2KU.

$$\widetilde{C}_p=\begin{cases} AB[KU+1+k-i,j+i-1] & \forall i,k\{1,2...p\} \\ else\ 0 & j=(N-p)/2+1 \end{cases}$$

and

$$\widetilde{S}_p=\begin{cases} AB[1+k-i,j+i-1] & \forall k\geq i \quad k:\{1,2...KU\}\forall i:\{1,2...p\} \\ AB[k+p-i+1,j+i-1] & \forall k:\{KU+1...2*KU-p+i\}\forall i:\{1,2...p\} \\ else\ 0 & j=(N-p)/2+1 \end{cases}$$

For rotated and reversed systems, it is easiest to use the same definitions above and apply the rotation to the matrix AB. In order to achieve the definitions $\widetilde{C}^r_{jj},\widetilde{S}^r_{jj},\widetilde{P}^r_{jj},\vec{b}^r_{jj}$ needed to satisfy the rotated solutions given previously, it suffices to rotate the N columns of $\vec{b}$ (resulting in $\vec{c}$ given in the above main body) and to rotate the N columns of AB and then to reverse the rows and reverse the columns of AB to switch the direction of the solution after renumbering the indices.