```
> with(linalg);
```

[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian] (1)

```
>  C_cl := matrix([[15/8, -5/4, 3/8], [3/8, 3/4, -1/8], [-1/8, 3/4, 3/8]]);
```

$$C\_cl := \begin{bmatrix} \frac{15}{8} & -\frac{5}{4} & \frac{3}{8} \\ \frac{3}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \end{bmatrix} \quad (2)$$

```
> C_ac :=  matrix([[23/24, 1/12, -1/24],[-1/24, 13/12, -1/24], [-1/24, 1/12, 23/24]]);
```

$$C\_ac := \begin{bmatrix} \frac{23}{24} & \frac{1}{12} & -\frac{1}{24} \\ -\frac{1}{24} & \frac{13}{12} & -\frac{1}{24} \\ -\frac{1}{24} & \frac{1}{12} & \frac{23}{24} \end{bmatrix} \quad (3)$$

```
> C_al := multiply(C_cl, C_ac);
```

$$C\_al := \begin{bmatrix} \frac{11}{6} & -\frac{7}{6} & \frac{1}{3} \\ \frac{1}{3} & \frac{5}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{5}{6} & \frac{1}{3} \end{bmatrix} \quad (4)$$

```
> # equ. (18)
  vs := r -> sum(C_al[r,j] * v[i-r+j], j=1..3);
```

$$vs := r \rightarrow \sum_{j=1}^{3} C\_al_{r,j}\, v_{i-r+j} \quad (5)$$

```
> # equ. (19), this computes Aminus(2) which is the reconstructed
  result. omega[r] are the beta_shu
  v12 := collect(sum(omega[r] * vs(r), r=1..3), [v[i-2], v[i-1], v
  [i], v[i+1], v[i+1]]);
```

$$v12 := -\frac{1}{6}\,\omega_3\,v_{i-2} + \left(\frac{1}{3}\,\omega_2 + \frac{5}{6}\,\omega_3\right) v_{i-1} + \left(\frac{11}{6}\,\omega_1 + \frac{5}{6}\,\omega_2 + \frac{1}{3}\,\omega_3\right) v_i + \left(-\frac{7}{6}\,\omega_1 - \frac{1}{6}\,\omega_2\right) v_{i+1} + \frac{1}{3}\,\omega_1\,v_{i+2} \tag{6}$$