ComputerGraphics.mw -- Applications to Computer Graphics (Lay, Section 2.7)

Getting Started

The following commands are needed at the start of any Maple session for this course.

```
> restart;
> with( LinearAlgebra );
```

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip] **(1.1)**

```
> with( plots );
```

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, **(1.2)** *logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

This command will help us to produce more uniform plots.

```
> opts := style=LINE, view=[0..10,0..10]:
```

Transformations of the Letter "N"

The Original Letter (Matrix)

The letter "N" can be represented by the following 2x9 matrix. (Note the matrix in the text does NOT produce an "N" -- or any letter!)

```
> N := < <0,0> | <.5,0> | <.5,6.42> | <6,0> | <6,8> | <5.5,8>
  | <5.5,1.58> | <0,8> | <0,0> >;
```

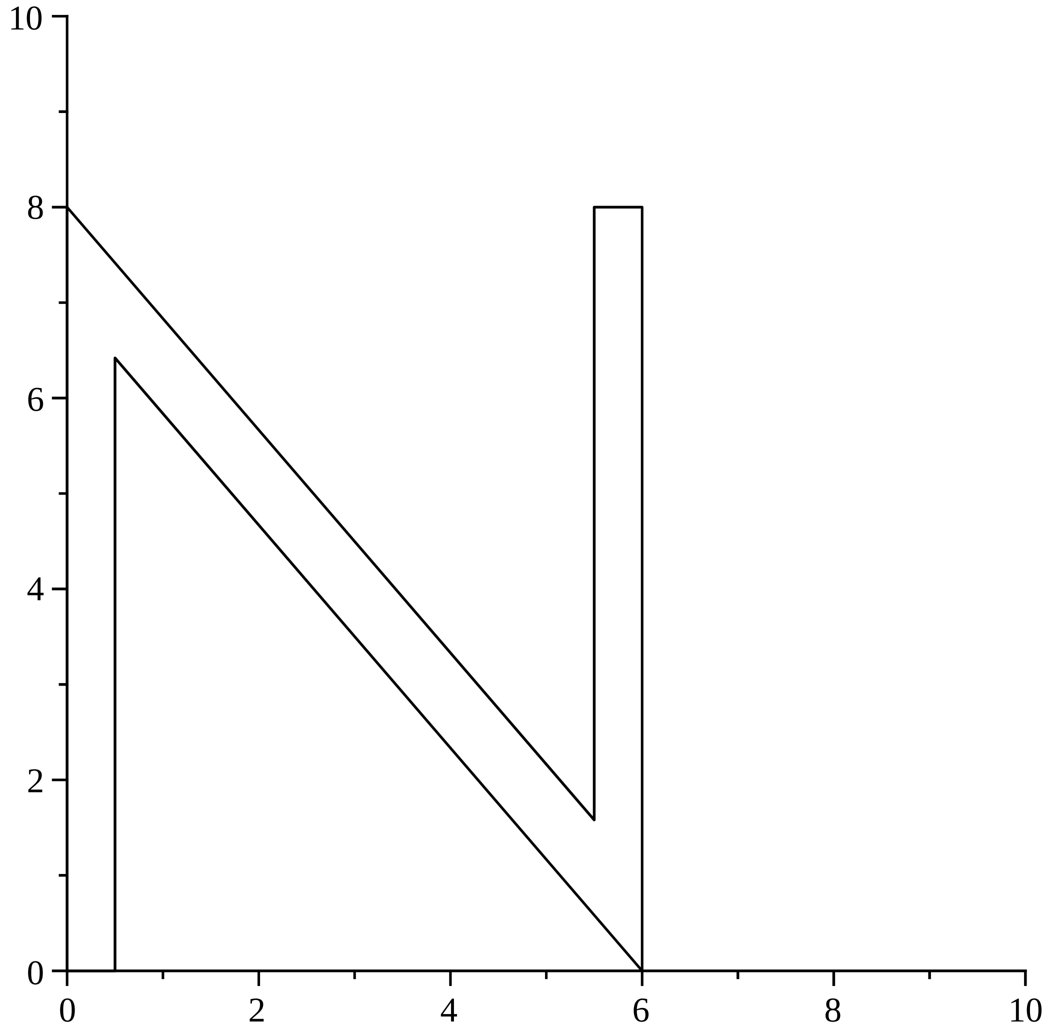
$$N := \begin{bmatrix} 0 & 0.5 & 0.5 & 6 & 6 & 5.5 & 5.5 & 0 & 0 \\ 0 & 0 & 6.42 & 0 & 8 & 8 & 1.58 & 8 & 0 \end{bmatrix} \quad \textbf{(2.1.1)}$$

The following command plots the columns of the matrix as points in the plane.

```
> plotN := pointplot( N, opts );
```

$$plotN := PLOT(\ldots) \quad \textbf{(2.1.2)}$$

```
> plotN;
```



Italics -- A Horizontal Shear

```
> A := < <1,0> | <0.25,1> >;
```

$$A := \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix} \quad \textbf{(2.2.1)}$$

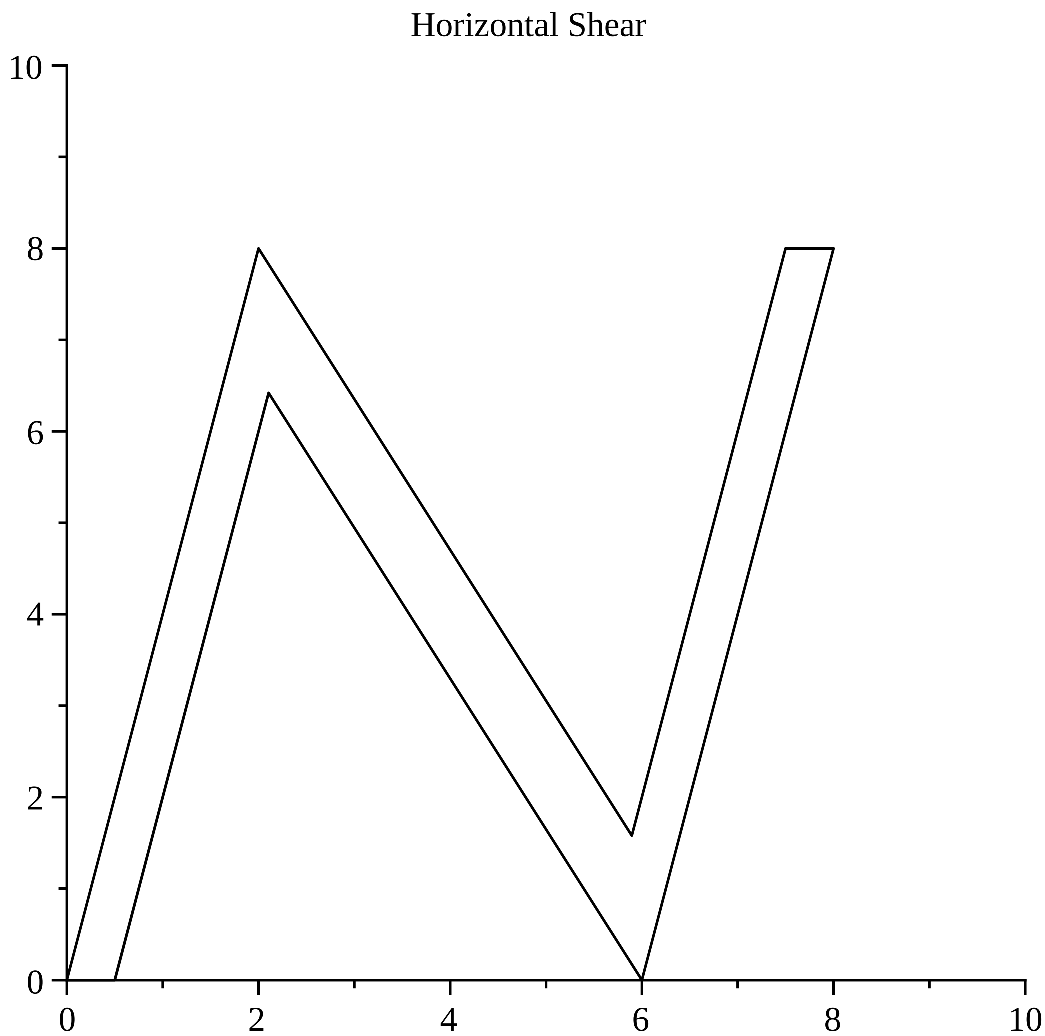
```
> Ns := A . N;
```

Ns := [[0., 0.500000000000000000, 2.10499999999999998, 6., 8.,
7.50000000000000000, 5.89499999999999956, 2., 0.],
[0., 0., 6.41999999999999993, 0., 8., 8., 1.58000000000000008, 8., 0.]] **(2.2.2)**

```
> plotNs := pointplot( Ns, opts, title="Horizontal Shear" );
```

plotNs := *PLOT*(...) **(2.2.3)**

```
> plotNs;
```



Thin -- A Horizontal Compression

```
> S := < <0.75,0> | <0,1> >;
```

$$S := \begin{bmatrix} 0.75 & 0 \\ 0 & 1 \end{bmatrix} \quad \textbf{(2.3.1)}$$

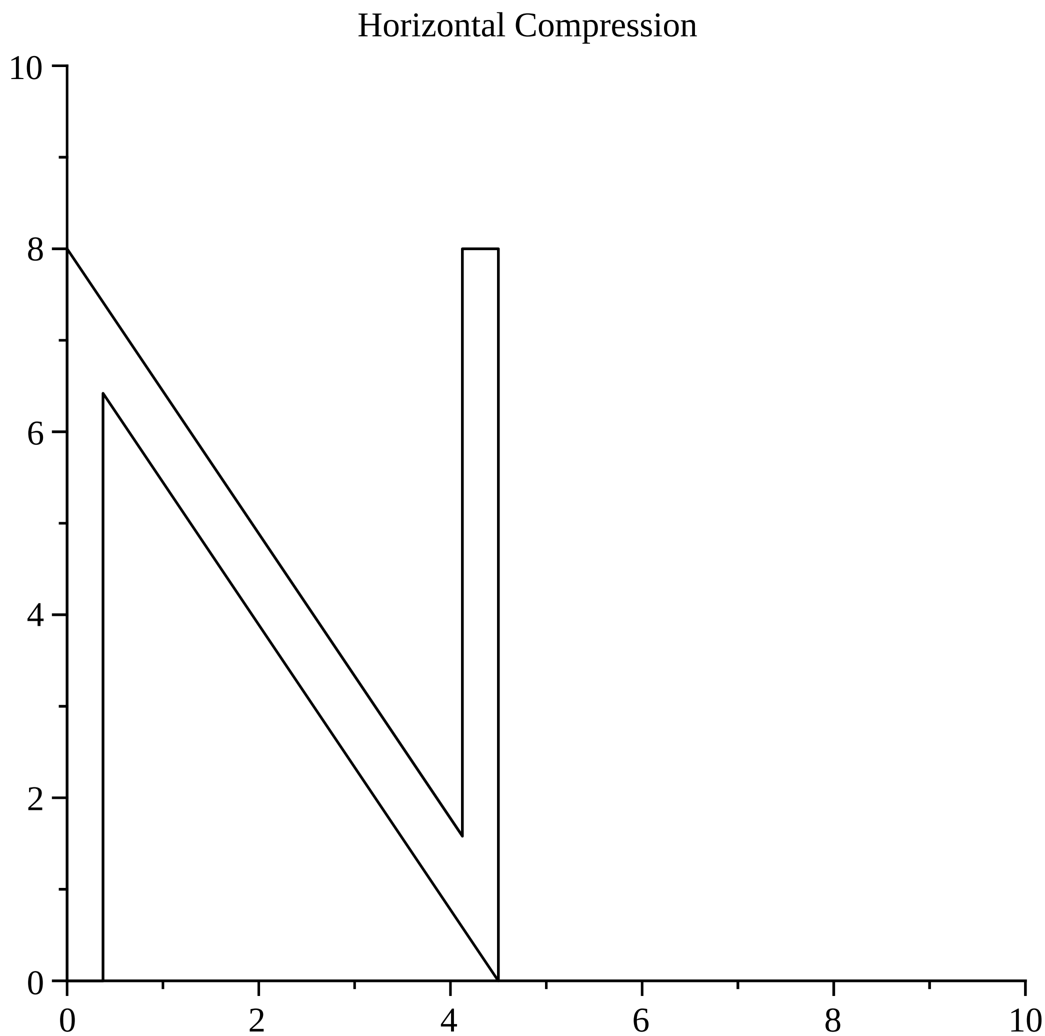
```
> Nt := S . N;
```

Nt := [[0., 0.375000000000000000, 0.375000000000000000, 4.50000000000000000, 4.50000000000000000, 4.12500000000000000, 4.12500000000000000, 0., 0.], [0., 0., 6.41999999999999993, 0., 8., 8., 1.58000000000000008, 8., 0.]] **(2.3.2)**

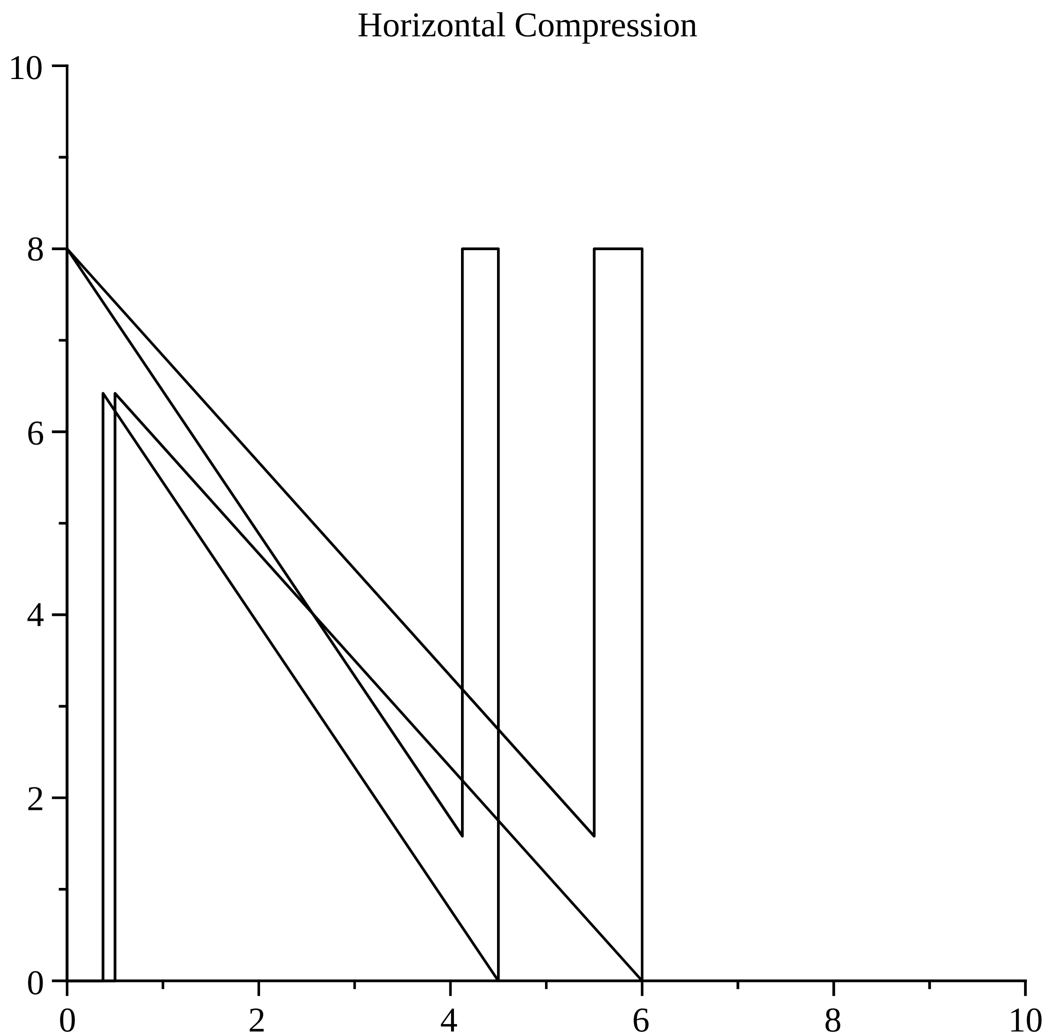
```
> plotNt := pointplot( Nt, opts, title="Horizontal
  Compression" );
```

plotNt := *PLOT*(...) **(2.3.3)**

```
> plotNt;
```




```
> display( plotN, plotNt );
```



Composite Transformations

```
> SA := S . A;    # shear, then compress
```

$$SA := \begin{bmatrix} 0.750000000000000000 & 0.187500000000000000 \\ 0. & 1. \end{bmatrix} \quad \textbf{(2.4.1)}$$

```
> AS := A . S;    # compress, then shear
```

$$AS := \begin{bmatrix} 0.750000000000000000 & 0.250000000000000000 \\ 0. & 1. \end{bmatrix} \quad \textbf{(2.4.2)}$$

```
> Nst := SA . N;
```

Nst := [[0., 0.375000000000000000, 1.57874999999999988, 4.50000000000000000, 6., 5.62500000000000000, 4.42124999999999968, 1.50000000000000000, 0.], [0., 0., 6.41999999999999993, 0., 8., 8., 1.58000000000000008, 8., 0.]] **(2.4.3)**

```
> Nts := AS . N;
```

(2.4.4)

Nts := [[0., 0.375000000000000000, 1.97999999999999998, 4.50000000000000000, 6.50000000000000000, 6.12500000000000000, 4.51999999999999957, 2., 0.], [0., 0., 6.41999999999999993, 0., 8., 8., 1.58000000000000008, 8., 0.]] **(2.4.4)**

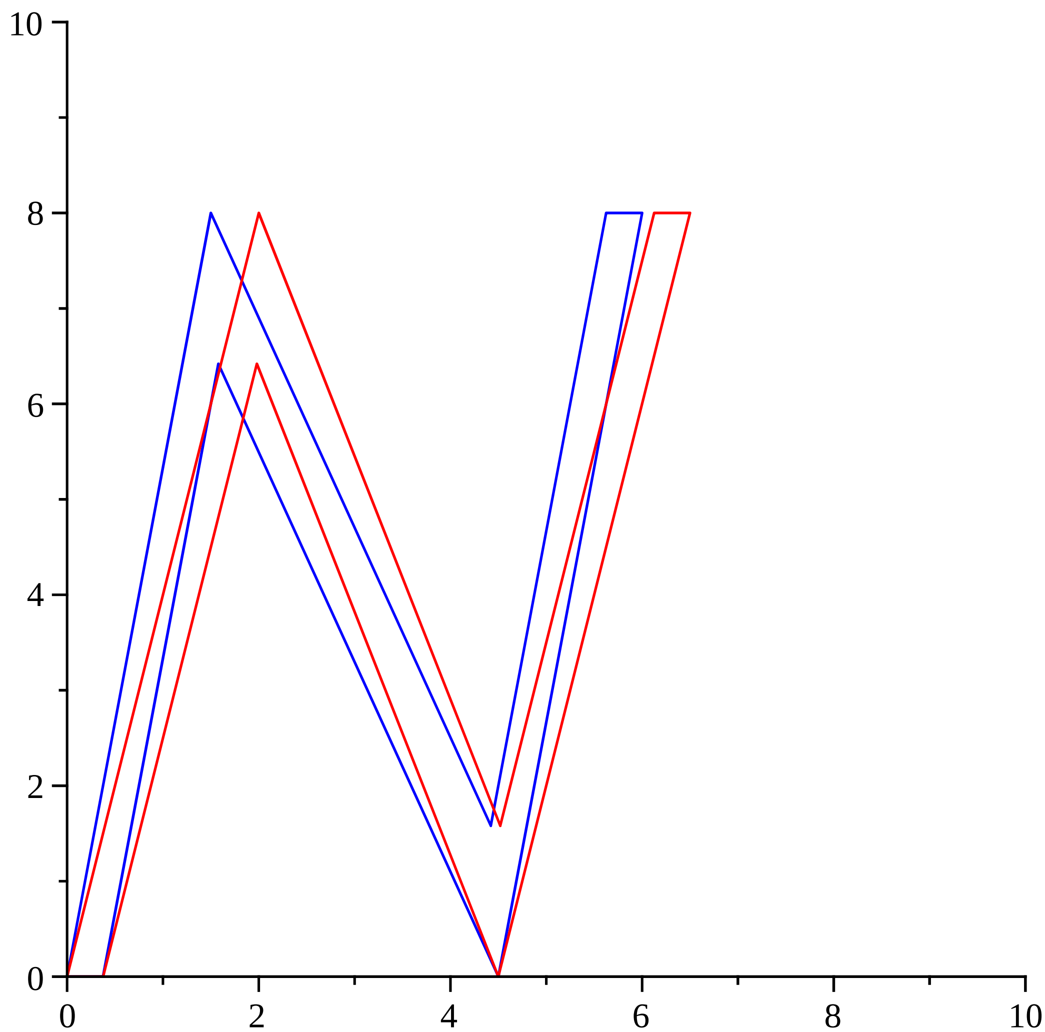
```
> plotNst := pointplot( Nst, opts, color=BLUE );
```

plotNst := *PLOT*(...) **(2.4.5)**

```
> plotNts := pointplot( Nts, opts, color=RED );
```

plotNts := *PLOT*(...) **(2.4.6)**

```
> display( [ plotNst, plotNts ] );
```



Rotation

```
> R := < < cos(Pi/6), sin(Pi/6) > | < -sin(Pi/6),  cos(Pi/6)
  > >;
```

$$R := \begin{bmatrix} \frac{1}{2}\sqrt{3} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2}\sqrt{3} \end{bmatrix} \quad (2.5.1)$$

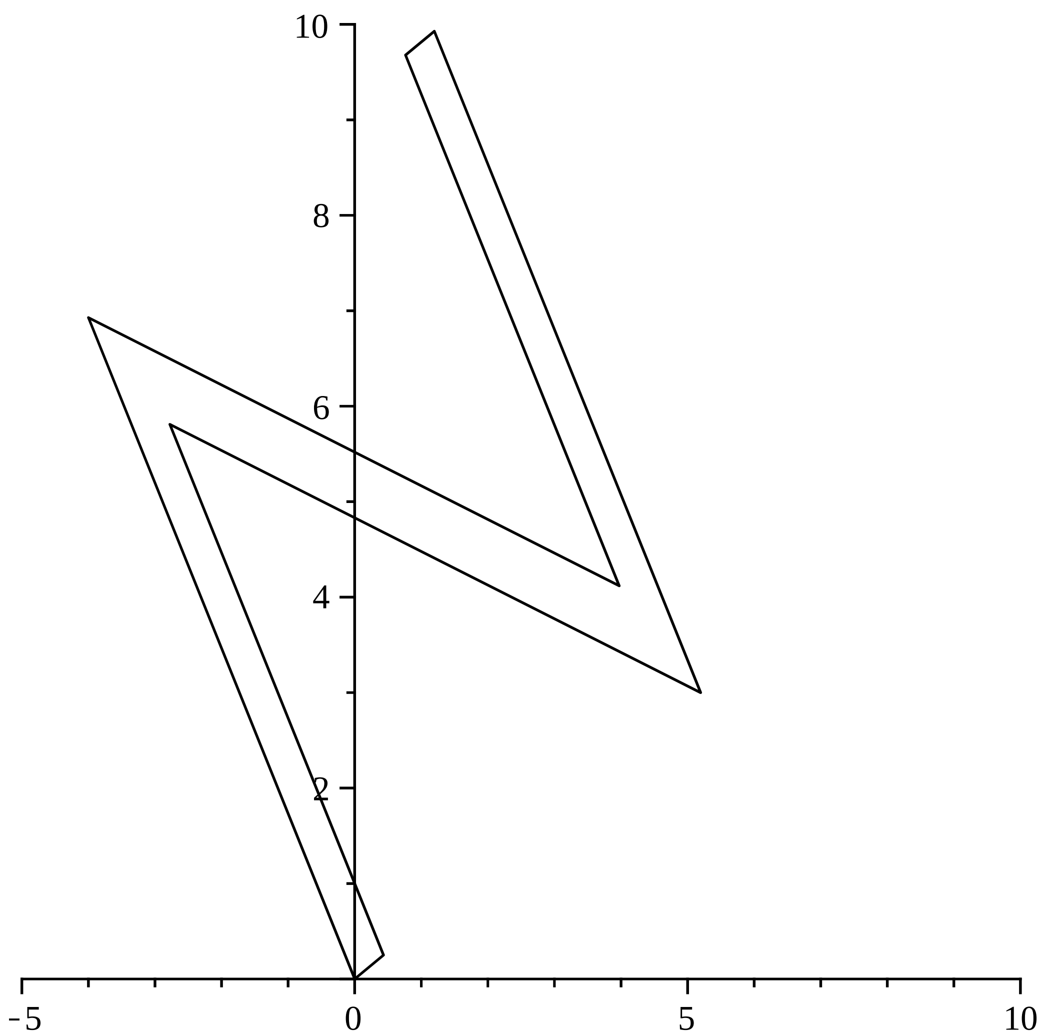
```
> Nr := R . N;
```

$$Nr := [[0, 0.2500000000\sqrt{3}, 0.2500000000\sqrt{3} - 3.210000000, 3\sqrt{3}, 3\sqrt{3} - 4, 2.750000000\sqrt{3} - 4, 2.750000000\sqrt{3} - 0.7900000000, -4, 0], [0, 0.2500000000, 0.2500000000 + 3.210000000\sqrt{3}, 3, 3 + 4\sqrt{3}, 2.750000000 + 4\sqrt{3}, 2.750000000 + 0.7900000000\sqrt{3}, 4\sqrt{3}, 0]] \quad (2.5.2)$$

```
> plotNr := pointplot( Nr, opts, view=[-5..10,0..10] );
```

$$plotNr := PLOT(...) \quad (2.5.3)$$

```
> plotNr;
```



Translation -- Homogeneous Coordinates

Recall that a translation cannot be implemented as a linear transformation in R^2. In order to create homogeneous coordinates, it is necessary to add a third row (all 1's) to the coordinate matrix.

```
> one := < 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 >;
```

$$one := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.6.1)$$

```
> Nh := < N, one >;
```

$$Nh := \begin{bmatrix} 0 & 0.5 & 0.5 & 6 & 6 & 5.5 & 5.5 & 0 & 0 \\ 0 & 0 & 6.42 & 0 & 8 & 8 & 1.58 & 8 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.6.2)$$

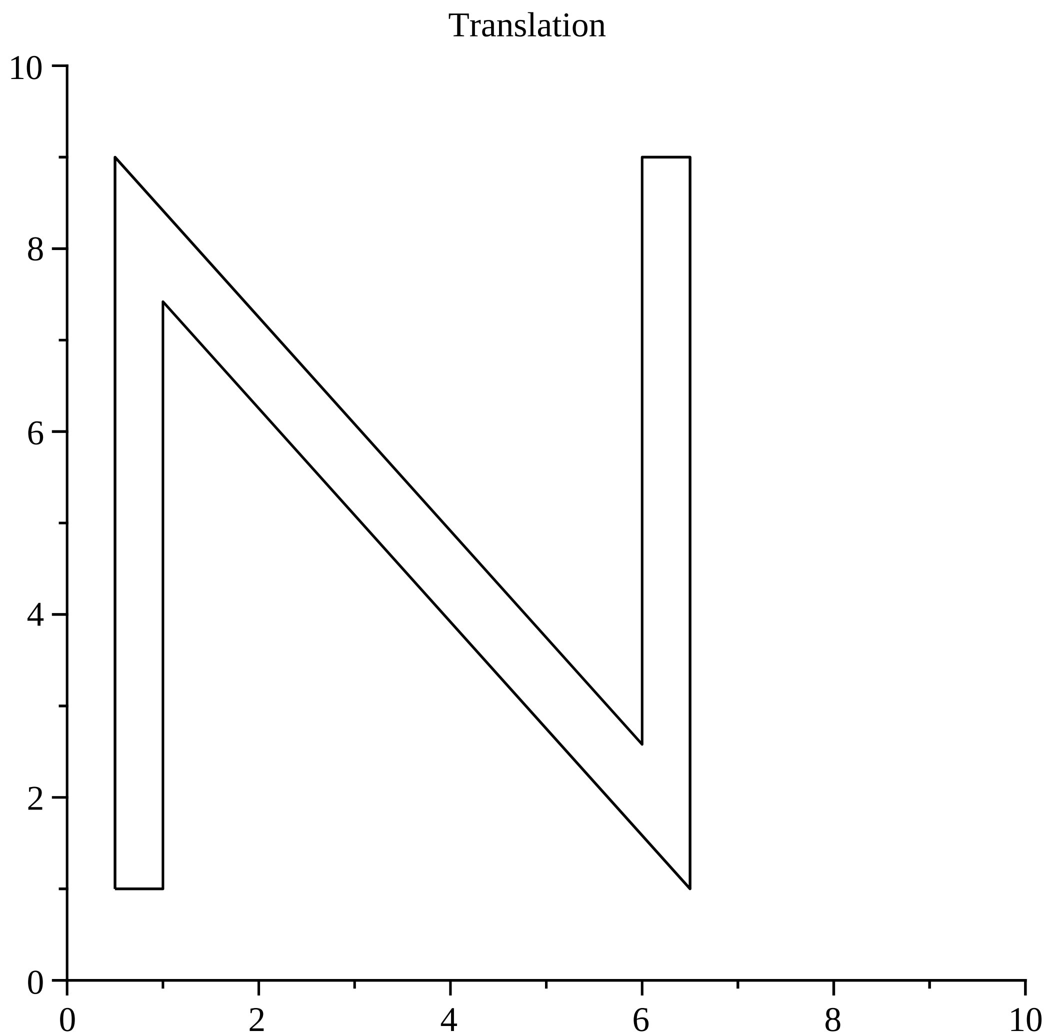
```
> T := < < 1,0,0 > | < 0,1,0> | < 0.5,1,1 > >;
```

$$T := \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6.3)$$

```
> Ntr := T . Nh;
```

Ntr := [[0.500000000000000000, 1., 1., 6.50000000000000000, 6.50000000000000000, 6., 6., 0.500000000000000000, 0.500000000000000000], [1., 1., 7.41999999999999993, 1., 9., 9., 2.58000000000000008, 9., 1.], [1., 1., 1., 1., 1., 1., 1., 1., 1.]] **(2.6.4)**

```
> pointplot( Ntr[1..2,1..-1], opts, title="Translation" );
```



Rotation about a point other than the origin

Here, we implement the rotation by 45 degrees about the point (3,7).

```
> T1 := < < 1,0,0 > | < 0,1,0 > | < -3,-7,1 > >;
```

$$T1 := \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -7 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7.1}$$

```
> T2 := < < 1,0,0 > | < 0,1,0 > | < 3,7,1 > >;
```

$$T2 := \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7.2}$$

```
> R := < < cos(Pi/4),sin(Pi/4),0 > | < -sin(Pi/4),cos(Pi/4),0
  > | < 0,0,1 > >;
```

$$R := \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 0 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7.3)$$

```
> A := T2 . R . T1;   # the composite transformation
```

$$A := \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 2\sqrt{2}+3 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & -5\sqrt{2}+7 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7.4)$$

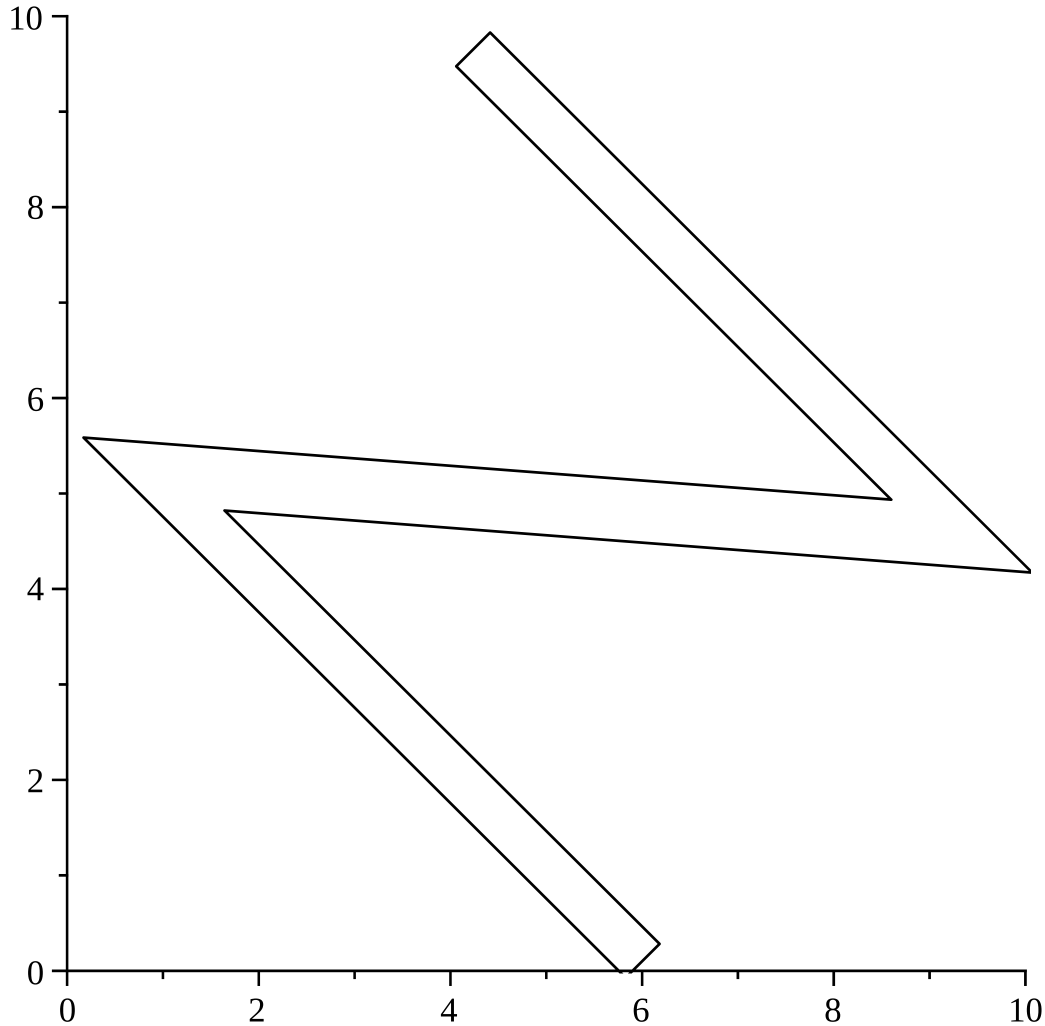
```
> Nr2 := A . Nh;
```

$$Nr2 := [[2\sqrt{2}+3, 2.250000000\sqrt{2}+3, -0.960000000\sqrt{2}+3, 5\sqrt{2}+3, \sqrt{2}+3, 0.750000000\sqrt{2}+3, 3.960000000\sqrt{2}+3, 3-2\sqrt{2}, 2\sqrt{2}+3], [-5\sqrt{2}+7, -4.750000000\sqrt{2}+7, -1.540000000\sqrt{2}+7, -2\sqrt{2}+7, 2\sqrt{2}+7, 1.750000000\sqrt{2}+7, -1.460000000\sqrt{2}+7, 7-\sqrt{2}, -5\sqrt{2}+7], [1, 1., 1., 1, 1, 1., 1., 1, 1]] \quad (2.7.5)$$

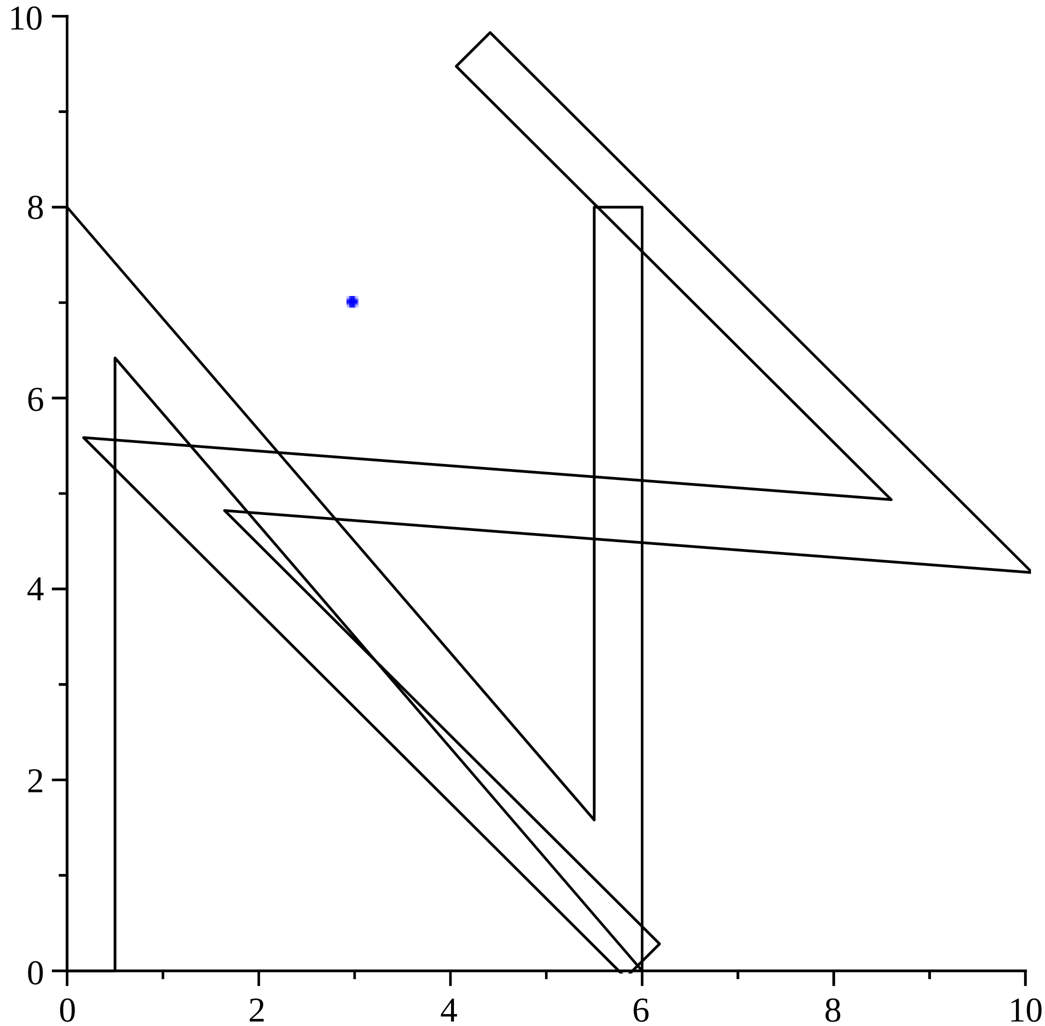
```
> plotNr2 := pointplot( Nr2[1..2,1..-1], opts );
```

$$plotNr2 := PLOT(\ldots) \quad (2.7.6)$$

```
> plotNr2;
```




```
> display( [plotN, plotNr2, pointplot( [[3,7]], symbol=
  solidcircle, color=blue )] );
```



Animated rotation about a point other than the origin

Here, we implement the animation of rotations about the point (3,7).

```
> T1 := < < 1,0,0 > | < 0,1,0 > | < -3,-7,1 > >;
```

$$T1 := \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -7 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8.1}$$

```
> T2 := < < 1,0,0 > | < 0,1,0 > | < 3,7,1 > >;
```

$$T2 := \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8.2}$$

```
> R := < < cos(Pi/4),sin(Pi/4),0 > | < -sin(Pi/4),cos(Pi/4),0
  > | < 0,0,1 > >;
```

$$R := \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 0 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8.3}$$

```
> A := T2 . R . T1;    # the composite transformation
```

$$A := \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 2\sqrt{2}+3 \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & -5\sqrt{2}+7 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8.4}$$

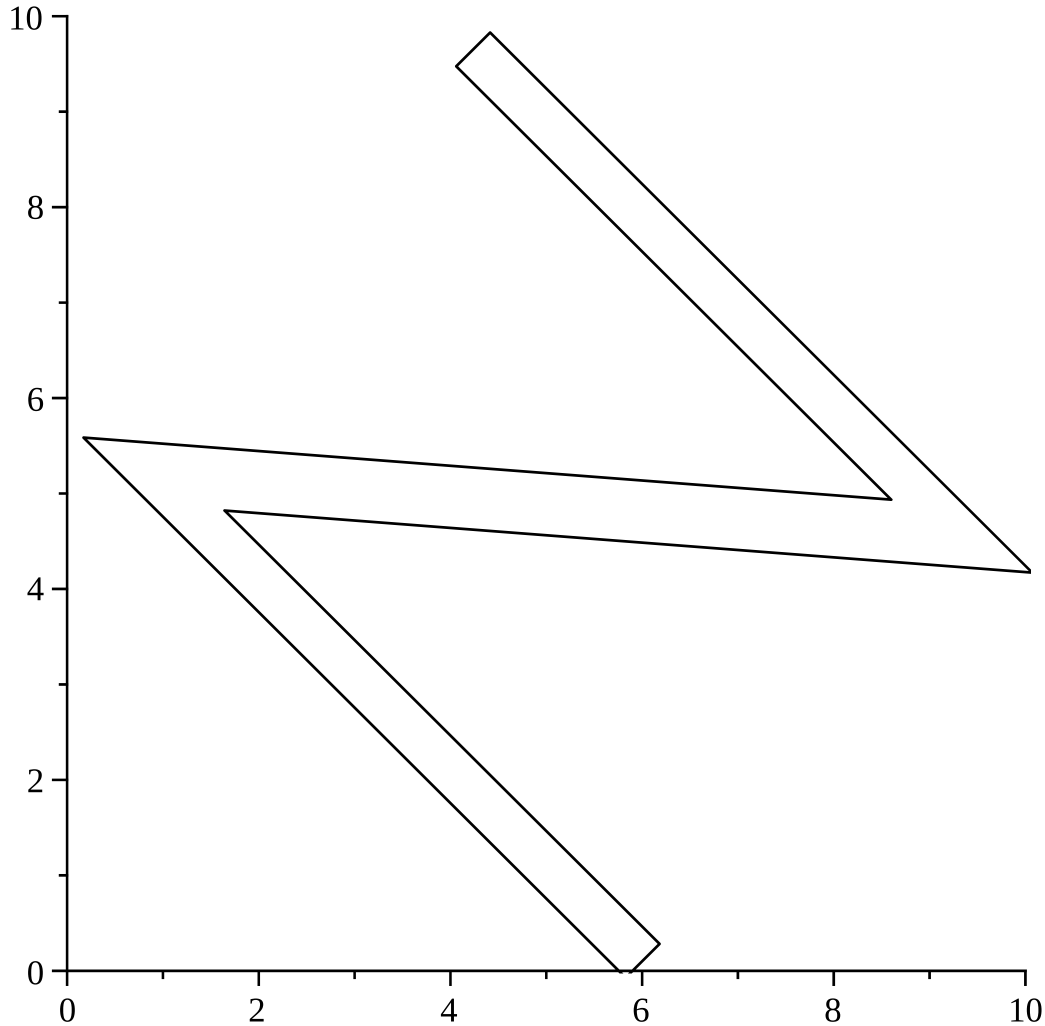
```
> Nr2 := A . Nh;
```

$$Nr2 := [[2\sqrt{2}+3, 2.250000000\sqrt{2}+3, -0.960000000\sqrt{2}+3, 5\sqrt{2}+3, \sqrt{2}+3, 0.750000000\sqrt{2}+3, 3.960000000\sqrt{2}+3, 3-2\sqrt{2}, 2\sqrt{2}+3], [-5\sqrt{2}+7, -4.750000000\sqrt{2}+7, -1.540000000\sqrt{2}+7, -2\sqrt{2}+7, 2\sqrt{2}+7, 1.750000000\sqrt{2}+7, -1.460000000\sqrt{2}+7, 7-\sqrt{2}, -5\sqrt{2}+7], [1, 1., 1., 1, 1, 1., 1., 1, 1]] \tag{2.8.5}$$

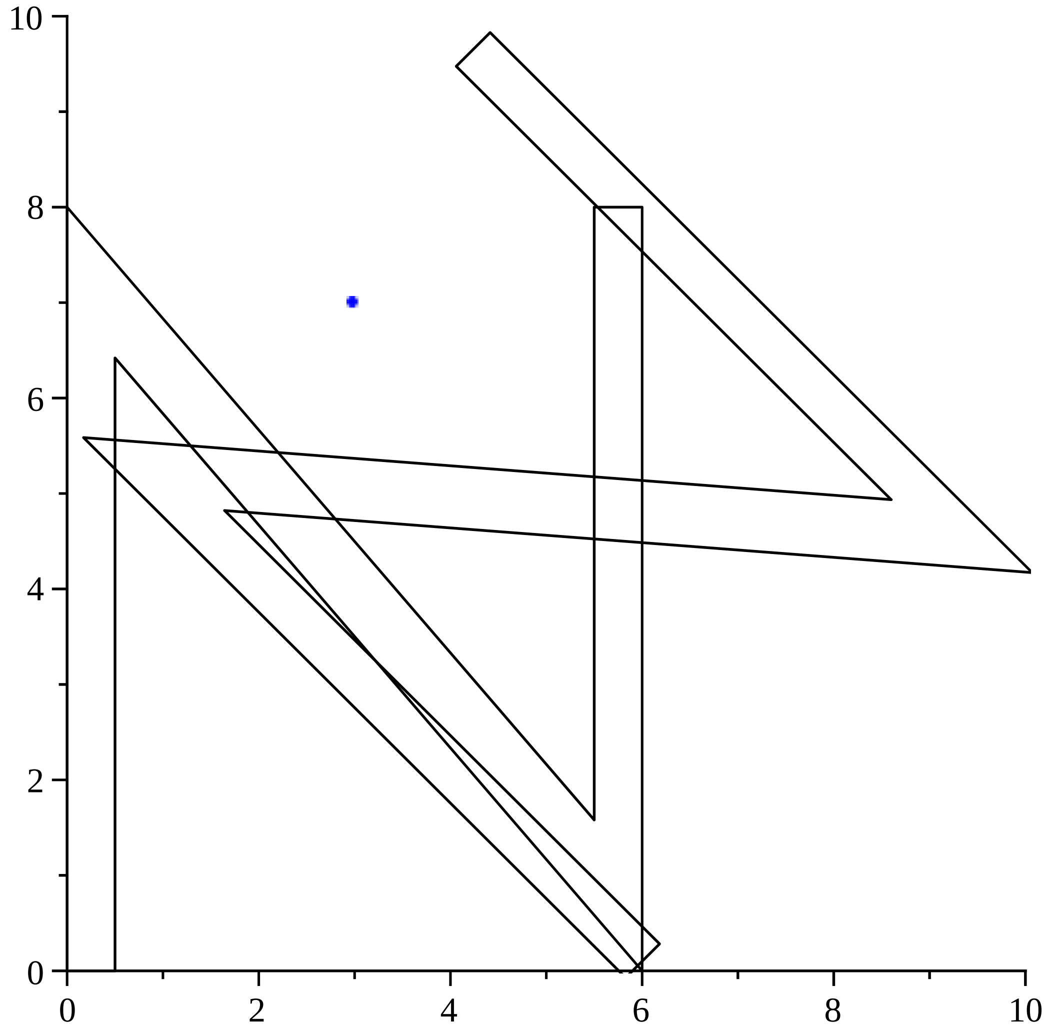
```
> plotNr2 := pointplot( Nr2[1..2,1..-1], opts );
```

$$plotNr2 := PLOT(\ldots) \tag{2.8.6}$$

```
> plotNr2;
```




```
> display( [plotN, plotNr2, pointplot( [[3,7]], symbol=
  solidcircle, color=blue )] );
```



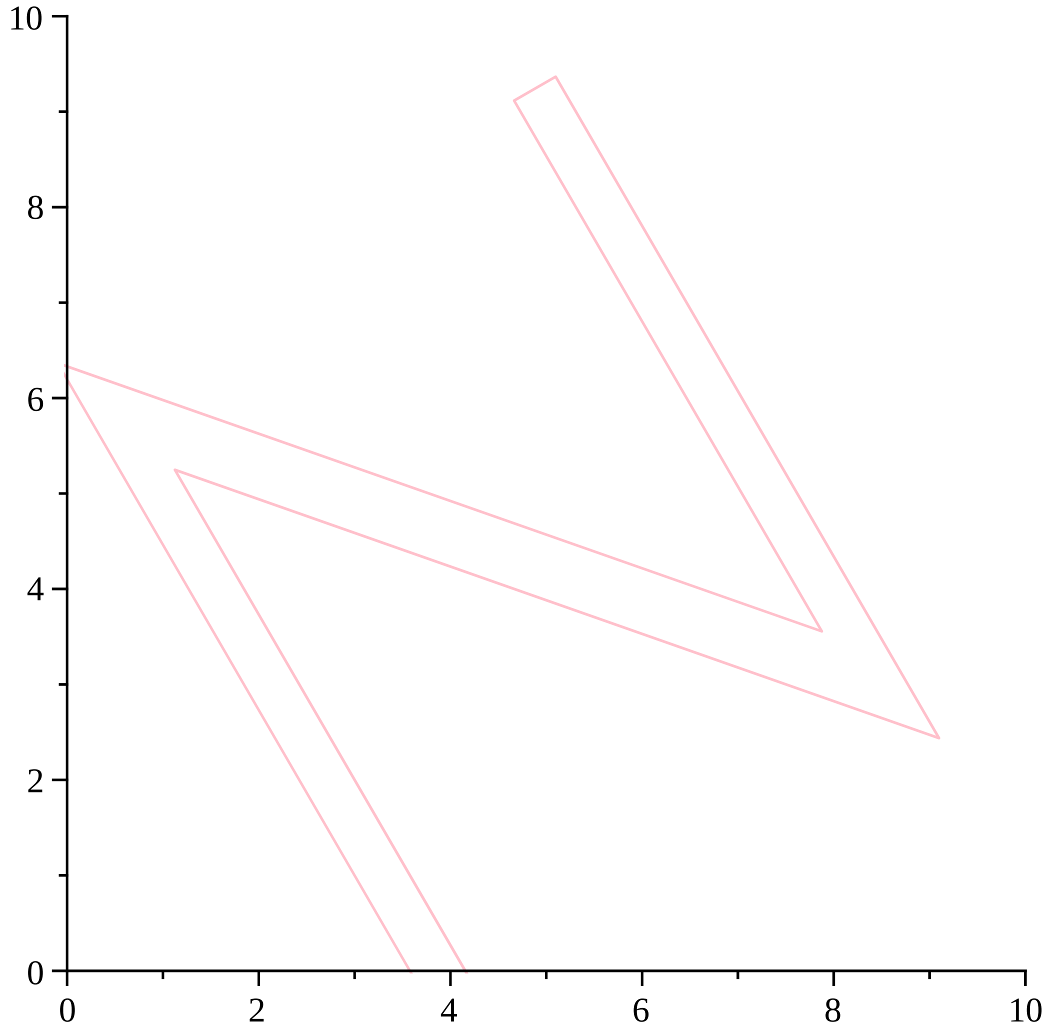

```
> plotback := display( [plotN, pointplot( [[3,7]], symbol=
  solidcircle, color=blue )] );
```

$$plotback := PLOT(\ldots) \tag{2.8.7}$$

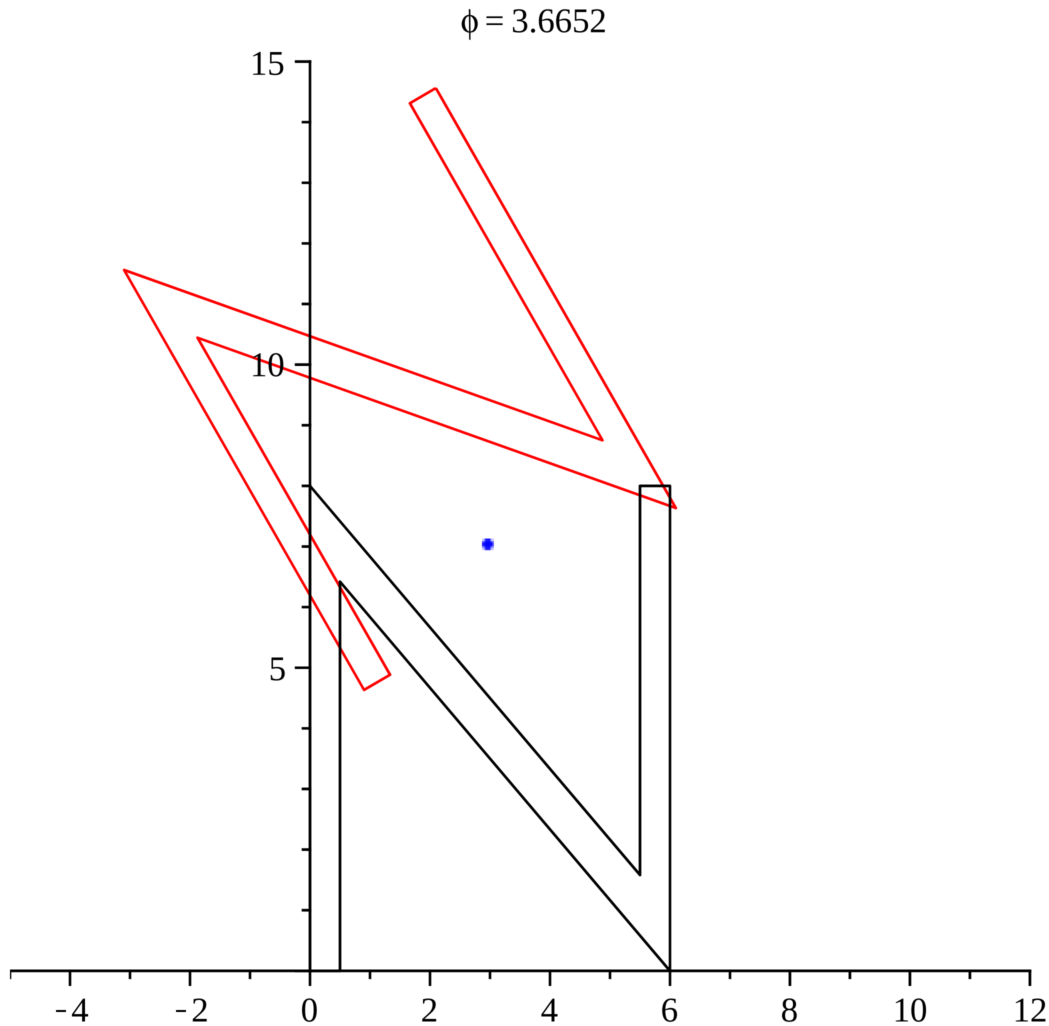
```
> R := < < cos(phi),sin(phi),0 > | < -sin(phi),cos(phi),0 > |
  < 0,0,1 > >;
```

$$R := \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8.8}$$

```
> pointplot( (T2.eval(R,phi=Pi/6).T1.Nh)[1..2,1..-1], opts,
  color=pink );
```




```
> animate( pointplot, [ (T2.R.T1.Nh)[1..2,1..-1], opts,
  color=red ], phi=0..2*Pi,
           background=plotback, view=[-5..12,0..15] );
```



ϕ = 3.6652
15
10
5
-4
-2
0
2
4
6
8
10
12