Visualization Updates in Maple 2025

Improved Plotting of Data using Units

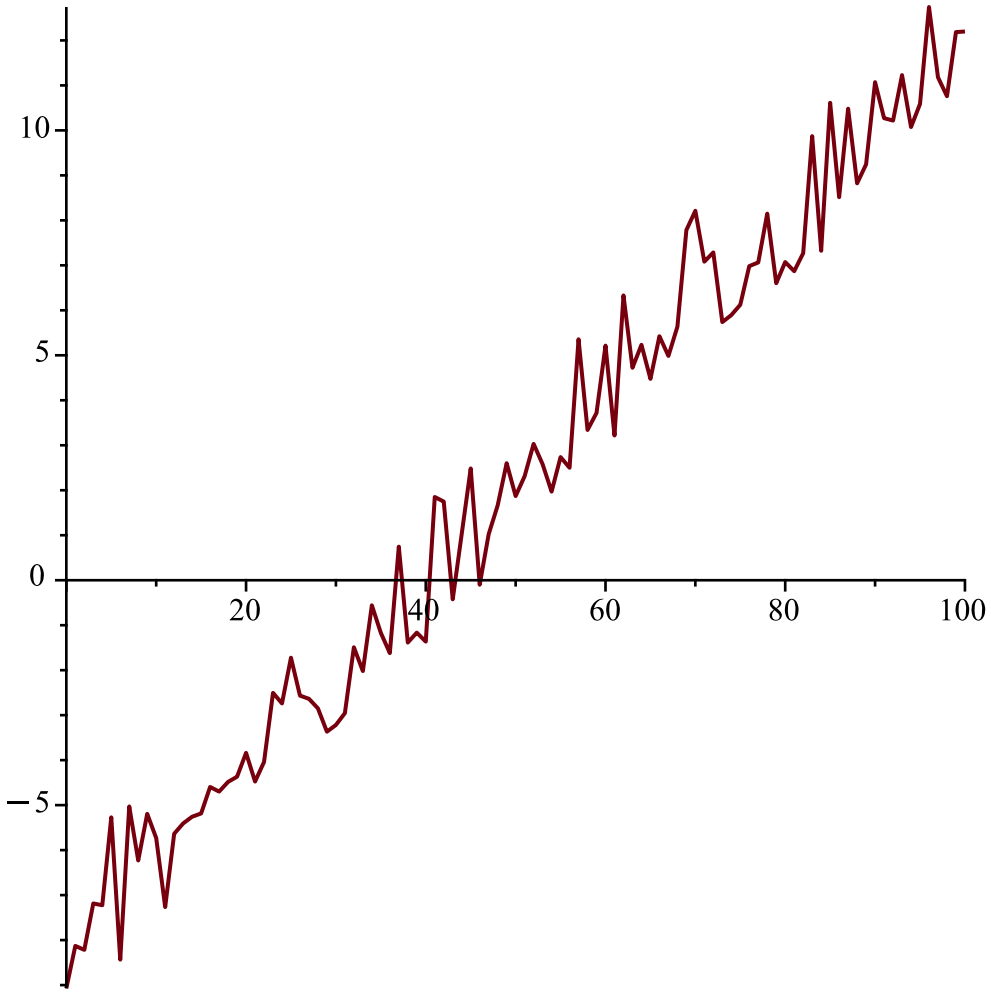
- Plotting lists of x- and y-coordinates of data points with units is seamless and efficient.
- If you have the x- and y-coordinates of some data points, you can plot them as follows.

```
> Xdata := <seq(0 .. 100)>:
> noisedata := convert(Statistics:-Sample(Normal(0, 1), 101), Vector
  [column]):
> Ydata := .2 * Xdata + noisedata -~ 8;
```

$$Ydata := \begin{bmatrix} 1 & -9.07242412799827 \\ 2 & -8.12907787054706 \\ 3 & -8.21709193690979 \\ 4 & -7.18553325475471 \\ 5 & -7.22542812804238 \\ 6 & -5.27117871582217 \\ 7 & -8.43485675434390 \\ 8 & -5.02882782824570 \\ 9 & -6.22964157858959 \\ 10 & -5.19352159124819 \\ & \vdots \end{bmatrix} \quad \text{(1)}$$

101 element Vector[column]

```
> plot(Xdata, Ydata);
```



In Maple 2025, this calling sequence accepts data with units.

```
> Xdata_units := Xdata * Unit(s);
```

$$Xdata\_units := \begin{bmatrix} 1 & 0 \\ 2 & \mathrm{s} \\ 3 & 2\ \mathrm{s} \\ 4 & 3\ \mathrm{s} \\ 5 & 4\ \mathrm{s} \\ 6 & 5\ \mathrm{s} \\ 7 & 6\ \mathrm{s} \\ 8 & 7\ \mathrm{s} \\ 9 & 8\ \mathrm{s} \\ 10 & 9\ \mathrm{s} \\ & \vdots \end{bmatrix} \quad \textbf{(2)}$$

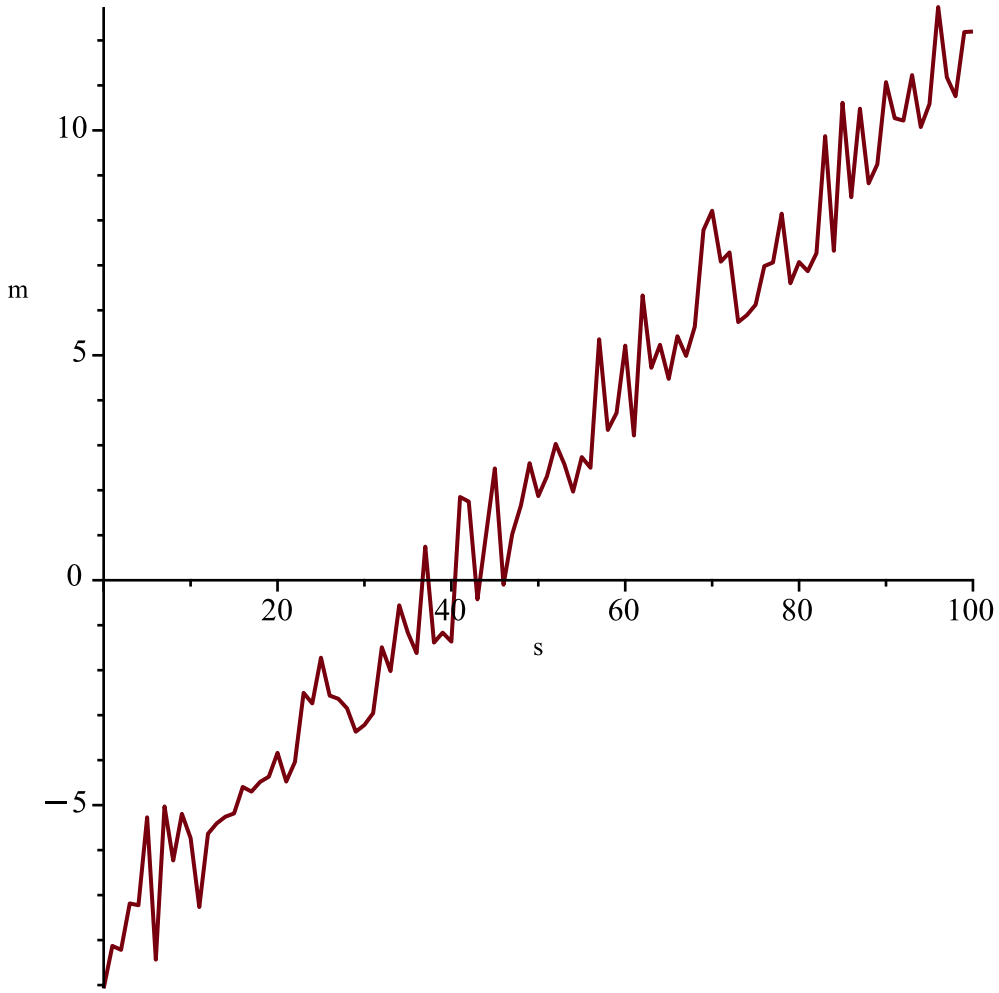
101 element Vector[column]

```
> Ydata_units := Ydata * Unit(m);
```

$$Ydata\_units := \begin{bmatrix} 1 & -9.07242412799827\ \mathrm{m} \\ 2 & -8.12907787054706\ \mathrm{m} \\ 3 & -8.21709193690979\ \mathrm{m} \\ 4 & -7.18553325475471\ \mathrm{m} \\ 5 & -7.22542812804238\ \mathrm{m} \\ 6 & -5.27117871582217\ \mathrm{m} \\ 7 & -8.43485675434390\ \mathrm{m} \\ 8 & -5.02882782824570\ \mathrm{m} \\ 9 & -6.22964157858959\ \mathrm{m} \\ 10 & -5.19352159124819\ \mathrm{m} \\ & \vdots \end{bmatrix} \quad \textbf{(3)}$$

101 element Vector[column]

```
> plot(Xdata_units, Ydata_units);
```



Custom Ranges for Contour Plots

- The command plots:-listcontplot creates a two-dimensional contour plot of a grid of values. This command now optionally accepts a pair of realcons ranges for its second and third arguments.
- Such custom ranges specify the ranges for the tickmarks for the horizontal and vertical axes.

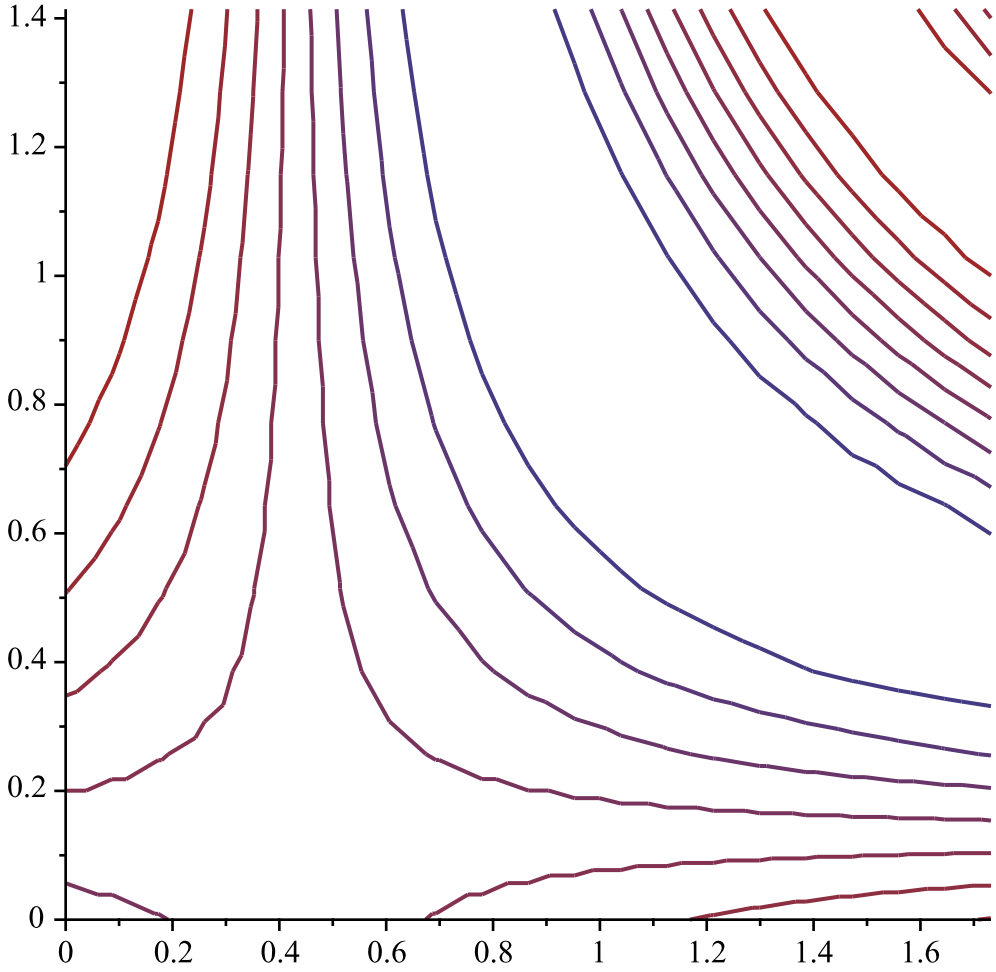
```
> xydata := Matrix([seq([seq( evalf(sin(x/5*y/5)), x=-1..10)], y=-5.
  .15)]);
```

xydata := **(4)**

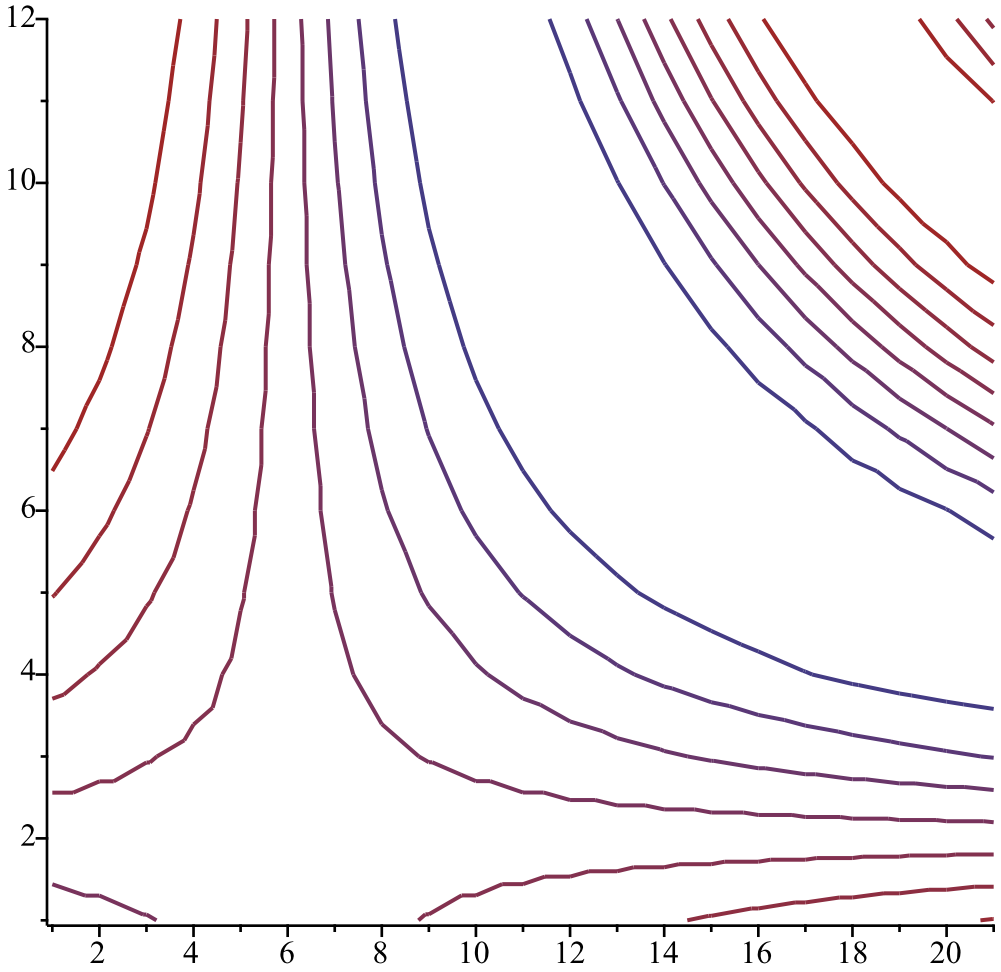
	1	2	3	4	
1	0.1986693308	0.	−0.1986693308	−0.3894183423	−0.5 ⋯
2	0.1593182066	0.	−0.1593182066	−0.3145665606	−0.4 ⋯
3	0.1197122073	0.	−0.1197122073	−0.2377026264	−0.3 ⋯
4	0.07991469397	0.	−0.07991469397	−0.1593182066	−0.2 ⋯
5	0.03998933419	0.	−0.03998933419	−0.07991469397	−0.1 ⋯
6	0.	0.	0.	0.	⋯
7	−0.03998933419	0.	0.03998933419	0.07991469397	0.11 ⋯
8	−0.07991469397	0.	0.07991469397	0.1593182066	0.23 ⋯
9	−0.1197122073	0.	0.1197122073	0.2377026264	0.35 ⋯
10	−0.1593182066	0.	0.1593182066	0.3145665606	0.46 ⋯
	⋮	⋮	⋮	⋮	

21 × 12 Matrix

```
> plots:-listcontplot(xydata, 0..sqrt(3), 0..sqrt(2));
```



```
> plots:-listcontplot(xydata);
```



New Sparse Option for Matrix Plot

- The plots:-matrixplot command has a new option sparse when rendering a matrix with discrete cells. This avoids generating plot objects for matrix entries that are equal to zero. This is more efficient for sparse matrices and makes it easier to combine plots of several sparse matrices.

```
> G := GraphTheory:-RandomGraphs:-AssignEdgeWeights(GraphTheory:-
  RandomGraphs:-RandomDigraph(25), -2.0..2.0):
```

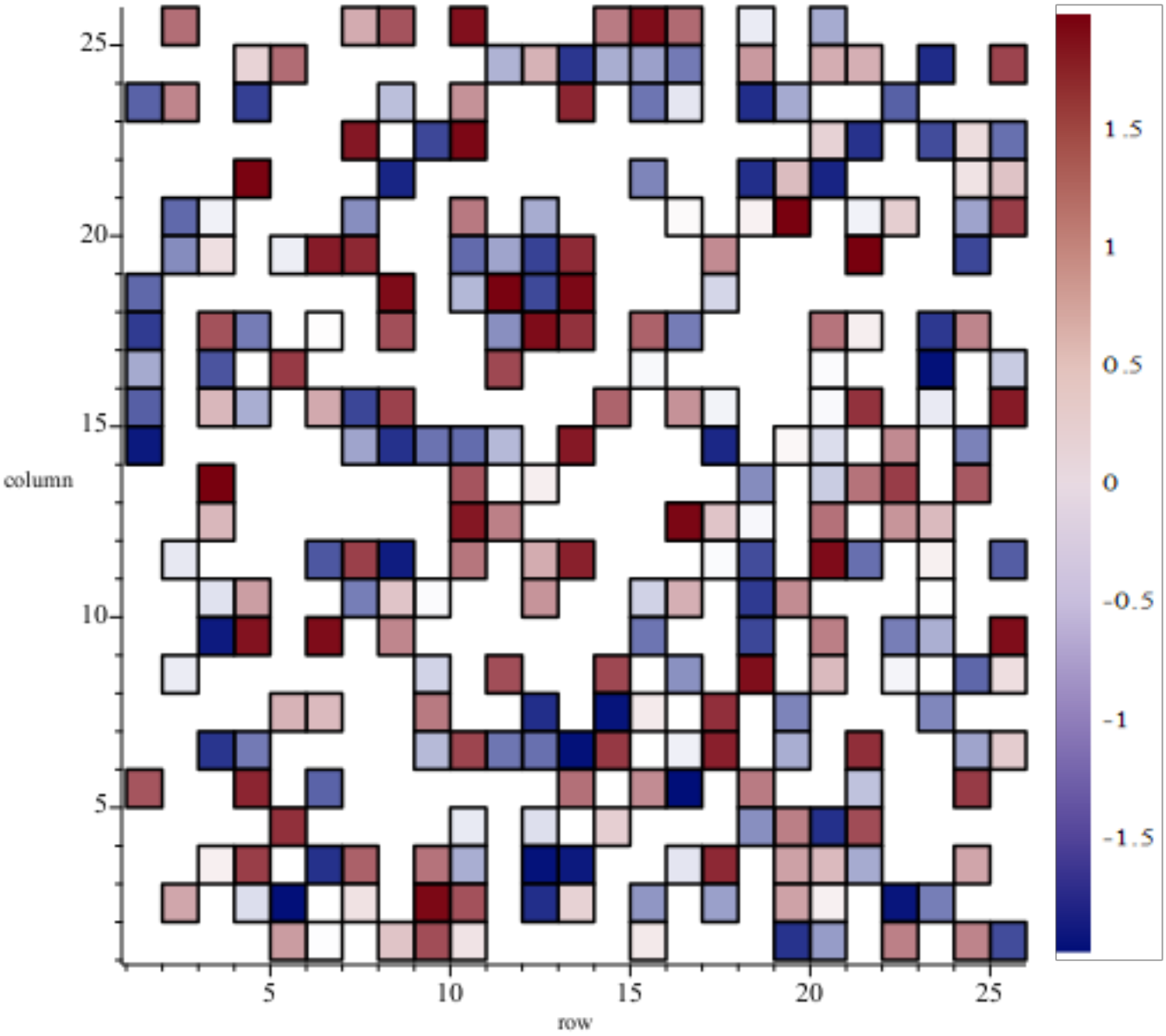
```
> M := GraphTheory:-WeightMatrix(G);
```

$M :=$ **(5)**

	1	2	3	
1	0.	0.	0.	⋯
2	0.	0.694548562020237	0.	⋯
3	0.	0.	0.12146873873663	⋯
4	0.	−0.273021877113815	1.50498476213159	⋯
5	0.774247240346537	−1.97610554829010	0.	⋯
6	−0.0201008787774621	0.	−1.7034287803852	⋯
7	0.	0.226127078634973	1.2322339831205	⋯
8	0.462111222329499	0.	0.	⋯
9	1.38464686554134	1.92616163779099	1.08714420017225	⋯
10	0.217235158855205	1.35303984428563	−0.6692646323269	⋯
	⋮	⋮	⋮	

25 × 25 Matrix

```
> plots:-matrixplot( M, 'dim'=2, 'colorscheme'=["Niagara Blue",
  "White","Niagara Red"], 'sparse', 'size'=[700,600]);
```



Updates to the Student ODEPlot Command

- The ODEPlot command in the Student:-ODEs package has been updated. You now have a more convenient way to quickly visualize the solutions for systems of two first-order ODEs.
- For details, see New ODEPlot calling sequence.

Additions to the GraphTheory Package

- The GraphTheory package in Maple offers several new special graphs as well as other improvements.

- For details, see GraphTheory Updates in Maple 2025.

Visualizations of Discrete Mathematics

- The GraphTheory package can be used to provide insight into many areas of mathematics. Some examples of using GraphTheory with the new PartiallyOrderedSets package to visualize posets, including face lattices coming from the PolyhedralSets package, are found in Advanced Math Improvements in Maple 2025.