

The `mkpath` package

I have expanded the package of routines in the file `mkpath.inc` to include other useful procedures involving paths in two dimensions. Here is a summary of what is in the new package.

Routine: `mkpath`
Arguments: `pars /f ti tf N`
Returns:

This hasn't changed except that it is now slightly more efficient. I recall that the parametrization is the name of a procedure with two arguments. The first is the array in this call, which will usually contain parameters of the parametrization such as the radius or centre of a circle, and the second a value of t . Its output must be an array $[[x\ y][x'\ y']]$ where the points of the curve are $(x(t), y(t))$ with velocity vector $(x'(t), y'(t))$.

Routine: `mkgraph`
Arguments: Almost the same as for `mkpath`

Except that the parametrization returns an array of two numbers $[y(t)\ y'(t)]$.

This is more efficient for building graphs of functions than the full version of `mkpath`, and easier to use.

Routine: `mkpolypath`
Arguments: Exactly the same as for `mkpath`

This routine will make the path with straight lines instead of Bezier curves. It ignores the derivatives in the parametrization, and can therefore be used to check the values of (x, y) and (x', y') independently.

Routine: `parametrization-to-polygon`
Arguments: `pars /f ti tf N`
Returns: a polygon of $(N + 1)$ points, points on the path parametrized by `/f`

Routine: `parametrization-to-dpolygon`
Arguments: `pars /f ti tf N`
Returns: A *dpolygon* of $(N + 1)$ points and modified velocities on the path parametrized by `/f`

A *dpolygon* is an array of pairs $[[x\ y][(dt/3)x'\ (dt/3)y']]$ where $dt = t_f - t_i/N$.

Routine: `parametrization-to-bezier`
Arguments: `pars /f ti tf N`
Returns: An array $[[x_0\ y_0] + \text{an array of } N \text{ sets } [x_1\ y_1\ x_2\ y_2\ x_3\ y_3]]$

Routine: `bezier-to-path`
Arguments: $[p_0 [\dots] \dots]$

Builds a path of Bezier curves from the array.

Routine: `polygon-to-path`
Arguments: $[[x\ y] \dots]$

Builds a path of lines from the array.

Routine: `dpolygon-to-path`
Arguments: $[[[x\ y][x'\ y']] \dots]$

Builds a path of Bezier curves from the array.