

[https://github.com/t-o-k/Maxima-bezier/rational\\_bezier\\_surface\\_3d.wmx](https://github.com/t-o-k/Maxima-bezier/rational_bezier_surface_3d.wmx)

Copyright (c) 2020 Tor Olav Kristensen, <http://subcube.com>

Use of this source code is governed by the GNU Lesser General Public License version 3, which can be found in the LICENSE file.

```
(%i1) kill(all)$
```

```
(%i1) load("draw")$
```

```
(%i2) load("bezier")$
```

```
(%i3) tau: 2*%pi$
```

```
(%i4) angle: tau/4;
```

```
(angle)  $\frac{\pi}{2}$ 
```

```
(%i5) w: matrix([ 1.0, cos(angle/2), 1.0 ]);
```

```
(w)  $\begin{pmatrix} 1.0 & \frac{1}{\sqrt{2}} & 1.0 \end{pmatrix}$ 
```

```
(%i6) weights: transpose(w).w;
```

```
(weights)  $\begin{pmatrix} 1.0 & \frac{1.0}{\sqrt{2}} & 1.0 \\ \frac{1.0}{\sqrt{2}} & \frac{1}{2} & \frac{1.0}{\sqrt{2}} \\ 1.0 & \frac{1.0}{\sqrt{2}} & 1.0 \end{pmatrix}$ 
```

The 9 points; <x, y, z> in the control grid:

```
[ <3, 2, 1>, <5, 2, 1>, <5, 4, 1> ]
```

```
[ <3, 2, 2>, <5, 2, 2>, <5, 4, 2> ]
```

```
[ <3, 3, 2>, <4, 3, 2>, <4, 4, 2> ]
```

This will create a surface that is 1/8 of the surface of a torus with major radius 2 and minor radius 1.

```
(%i7) points_x:
      matrix(
        [ 3, 5, 5 ],
        [ 3, 5, 5 ],
        [ 3, 4, 4 ]
      )
$
```

```
(%i8) points_y:
      matrix(
        [ 2, 2, 4 ],
        [ 2, 2, 4 ],
        [ 3, 3, 4 ]
      )
$
```

```
(%i9) points_z:
      matrix(
        [ 1, 1, 1 ],
        [ 2, 2, 2 ],
        [ 2, 2, 2 ]
      )
$
```

```
(%i10) define(f_x(u, v), rational_bezier_function_2a(points_x, weights, u, v));
```

```
(%o10) 
$$f_x(u, v) := \frac{(4.0 u^2 v^2 + 4.0 \sqrt{2} (1-u) u v^2 + 3.0 (1-u)^2 v^2 + 5.0 \sqrt{2} u^2 (1-v) v + 10 (1-u) u (1-v) v + 3.0 \sqrt{2} (1-u)^2 (1-v) v + 5.0 u^2 (1-v)^2 + 5.0 \sqrt{2} (1-u) u (1-v)^2 + 3.0 (1-u)^2 (1-v)^2) / (1.0 u^2 v^2 + 1.0 \sqrt{2} (1-u) u v^2 + 1.0 (1-u)^2 v^2 + 1.0 \sqrt{2} u^2 (1-v) v + 2 (1-u) u (1-v) v + 1.0 \sqrt{2} (1-u)^2 (1-v) v + 1.0 u^2 (1-v)^2 + 1.0 \sqrt{2} (1-u) u (1-v)^2 + 1.0 (1-u)^2 (1-v)^2)}$$

```

```
(%i11) define(f_y(u, v), rational_bezier_function_2a(points_y, weights, u, v));
```

```
(%o11) 
$$f_y(u, v) := \frac{(4.0 u^2 v^2 + 3.0 \sqrt{2} (1-u) u v^2 + 3.0 (1-u)^2 v^2 + 4.0 \sqrt{2} u^2 (1-v) v + 4 (1-u) u (1-v) v + 2.0 \sqrt{2} (1-u)^2 (1-v) v + 4.0 u^2 (1-v)^2 + 2.0 \sqrt{2} (1-u) u (1-v)^2 + 2.0 (1-u)^2 (1-v)^2) / (1.0 u^2 v^2 + 1.0 \sqrt{2} (1-u) u v^2 + 1.0 (1-u)^2 v^2 + 1.0 \sqrt{2} u^2 (1-v) v + 2 (1-u) u (1-v) v + 1.0 \sqrt{2} (1-u)^2 (1-v) v + 1.0 u^2 (1-v)^2 + 1.0 \sqrt{2} (1-u) u (1-v)^2 + 1.0 (1-u)^2 (1-v)^2)}$$

```

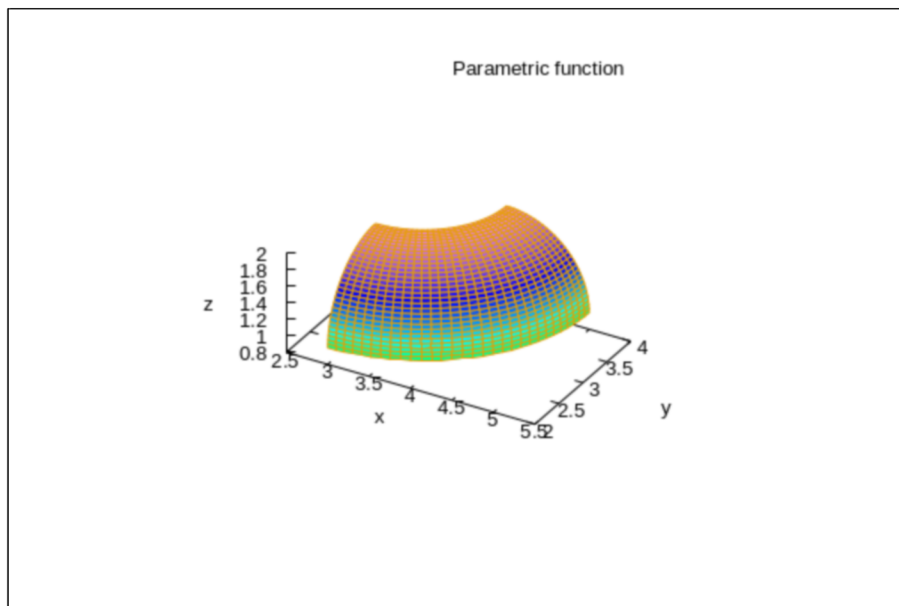
```
(%i12) define(f_z(u, v), rational_bezier_function_2a(points_z, weights, u, v));
```

```
(%o12) 
$$f_z(u, v) := \frac{(2.0 u^2 v^2 + 2.0 \sqrt{2} (1-u) u v^2 + 2.0 (1-u)^2 v^2 + 2.0 \sqrt{2} u^2 (1-v) v + 4 (1-u) u (1-v) v + 2.0 \sqrt{2} (1-u)^2 (1-v) v + 1.0 u^2 (1-v)^2 + 1.0 \sqrt{2} (1-u) u (1-v)^2 + 1.0 (1-u)^2 (1-v)^2)}{(1.0 u^2 v^2 + 1.0 \sqrt{2} (1-u) u v^2 + 1.0 (1-u)^2 v^2 + 1.0 \sqrt{2} u^2 (1-v) v + 2 (1-u) u (1-v) v + 1.0 \sqrt{2} (1-u)^2 (1-v) v + 1.0 u^2 (1-v)^2 + 1.0 \sqrt{2} (1-u) u (1-v)^2 + 1.0 (1-u)^2 (1-v)^2)}$$

```

```
(%i13) wxplot3d(
  [ f_x(u, v), f_y(u, v), f_z(u, v) ],
  [ u, 0, 1 ],
  [ v, 0, 1 ],
  same_xyz
);
```

```
(%t13)
```



```
(%o13)
```