The Mathematica® Journal

In and Out

Edited by Paul Abbott

In and Out offers readers an opportunity to ask questions of the experts. *The Mathematica Journal* encourages readers to submit problems in care of the editor.

■ Locus of Intersection

Q: Consider the circle $(x-1)^2 + (y-1)^2 = r^2$ and the parabola $y^2 = x$. First, find the intersection points of the circle and the parabola. Second, compute the chord of the circle through the two intersection points. Third, compute the line segment through $\{1, 1\}$ perpendicular to the chord at point $P = \{x, y\}$. Then, find the locus of the point P. And, finally, animate the process as a function of r. How can I do this?

A: Computing the intersection points of the circle and the parabola as a function of r is immediate. The result is rather complicated so we suppress the output.

```
eqns[r_] = {(x – 1)^2 + (y – 1)^2 == r^2, y^2 == x};
```

```
intersections[r_] = Solve[eqns[r], {x, y}];
```

Next, we compute the chord, which is the line segment joining the two *real* intersection points, using pattern matching.

```
chord[r_] :=
  chord[r] = Line[Cases[N[intersections[r]], {x → a_ /; a ∈ ℝ, y → b_ /; b ∈ ℝ} → {a, b}, ∞]]
```

To compute the line through the point $\{1, 1\}$ perpendicular to the chord at point $P = \{x, y\}$, we use the **ExtendGraphics`Geometry`** package by Tom Wickham-Jones, available from *MathSource* (library.wolfram.com/database/Books/3753). After putting the **ExtendGraphics`** folder into the **AddOns`Applications`** directory (or, alternatively, in the directory returned by evaluating **$UserAddOnsDirectory**), we can load it in the usual fashion.

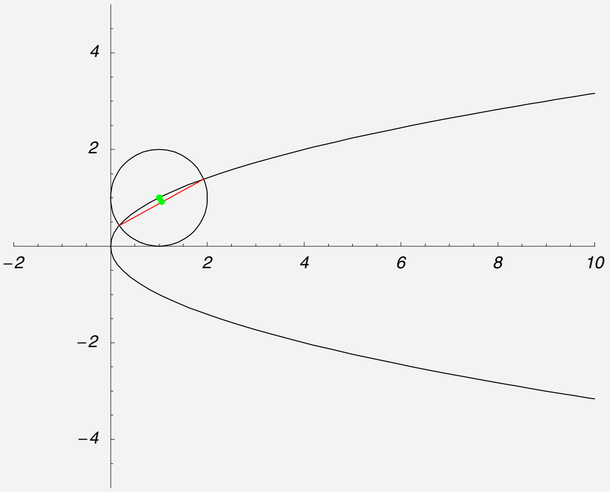
```
<< ExtendGraphics`Geometry`
```

After converting the chord to an **ImplicitLine** object (this implicit parametrization is the loci of points $\boldsymbol{p}$ that satisfy the equation $(\boldsymbol{p}-\boldsymbol{c})\cdot\vec{\boldsymbol{n}}=0$ where $\boldsymbol{c}$ is a point on the line and $\vec{\boldsymbol{n}}$ is the normal to the line), we can determine the point $P=\{x, y\}$ as a function of r using **ClosestPointOnLine**.

```
P[r_] := P[r] = ClosestPointOnLine[{1, 1}, ToImplicit[chord[r]]]
```

We are now in a position to show the chord, perpendicular, and locus of the point P as a function of r.

```
<< Graphics`;
Table[
  ImplicitPlot[eqns[r], {x, -1, 10}, PlotRange -> {{-2, 10}, {-5, 5}}, Epilog -> {Hue[1], chord[r], Hue[2/3],
      Line[{{1, 1}, P[r]}], Hue[1/3], AbsolutePointSize[3], Point[{1, 1}], Point[P[r]]}], {r, 9}];
```



Finding the *exact* locus of the point P is more complicated. Because points on the parabola have coordinates $\{y^2, y\}$, computing the locus of the point P as a function of y_1 and y_2—the two intersection points—is straightforward.

```
locus = FullSimplify[ClosestPointOnLine[{1, 1}, ToImplicit[Line[{{y1^2, y1}, {y2^2, y2}}]]]]
```

$$\left\{\frac{y_1^2+(y_2+1)\,y_1+y_2^2+y_2}{(y_1+y_2)^2+1},\ \frac{y_2+y_1\,(y_2\,(y_1+y_2)+1)+1}{(y_1+y_2)^2+1}\right\}$$

Returning to the original equations, after eliminating x, the *real* values of y at the two intersection points can be determined using **CylindricalAlgebraicDecomposition**.

```
Eliminate[eqns[r], x]
```

```
y^4 − y^2 − 2 y + 2 == r^2
```

```
Experimental`CylindricalAlgebraicDecomposition[% ∧ r > 0, {r, y}]
```

```
r > 0 ⋀ (y == Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] ⋁ y == Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2])
```

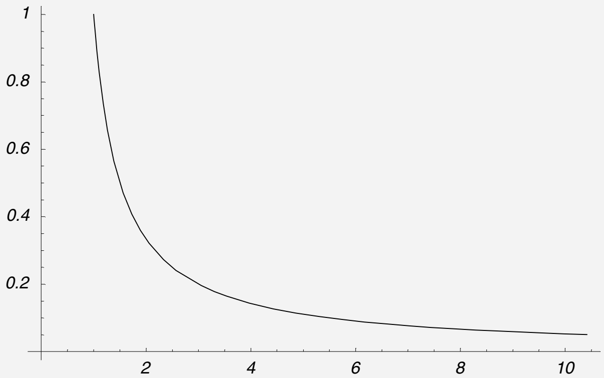
Hence, we obtain an explicit expression for the locus of P.

```
locus /. y_{i_} :→ Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, i] // FullSimplify
```

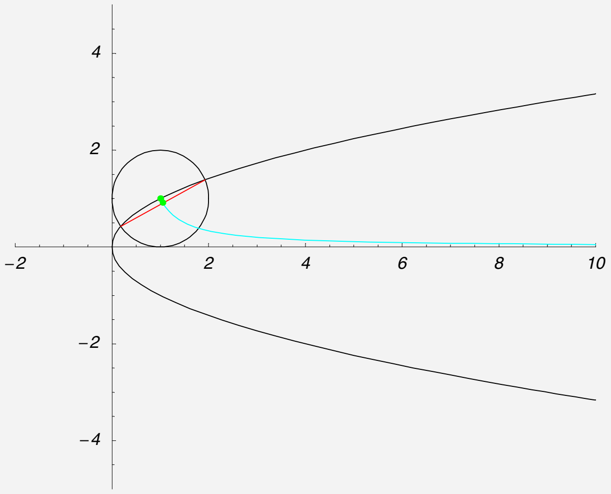
```
{(Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1]^2 +
        (Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2] + 1) Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] +
        Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2]^2 + Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2]) /
   ((Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] + Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2])^2 + 1),
  (Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2] +
        Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] (Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2]
              (Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] + Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2]) + 1) + 1) /
   ((Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 1] + Root[#1^4 − #1^2 − 2 #1 − r^2 + 2 &, 2])^2 + 1)}
```

Although this expression can be simplified further (note that **RootReduce** fails here), it is sufficient for our purpose. After plotting the locus, we can display it along with the circle, parabola, chord, and perpendicular.

```
ParametricPlot[%, {r, 0, 10}, Compiled → False];
```



```
Table[ImplicitPlot[eqns[r], {x, −1, 10}, PlotRange → {{−2, 10}, {−5, 5}},
   Epilog → {Hue[1], chord[r], Hue[1/2], First[%], Hue[2/3], Line[{{1, 1}, P[r]}],
     Hue[1/3], AbsolutePointSize[3], Point[{1, 1}], Point[P[r]]}], {r, 9}];
```



■ Pure Recursive Functions

Q: Normally, when creating a recursive function, I use the name of the function to perform the recursive call.

```
fact[n_] := If[n == 1, 1, n fact[n − 1]]
```

However, this has the disadvantage that the symbol fact is now global. Is it possible to create a pure recursive function?

A: Andrzej Kozlowski (akoz@mimuw.edu.pl) writes: **#0** is a rarely used “parameter,” which refers to the pure function itself and thus makes it possible to create pure recursive functions.

```
fact = Function[If[#1 == 0, 1, #1 #0[#1 − 1]]];
```

```
fact[5]

120
```

■ Binomial Limit

Q: How can I compute the limit of $(-1)^n \binom{-1/2}{n} \sqrt{n\,\pi}$ as $n \to \infty$ through integer values?

A: Numerically, it appears that the limit is one.

$$f(\mathrm{n\_}) = (-1)^n \binom{-1/2}{n} \sqrt{n\,\pi};$$

$$\mathrm{Table}[\{10^n,\ f(10^n)\ //\ N\},\ \{n,\ 0,\ 4\}]$$

$$\begin{pmatrix} 1 & 0.886227 \\ 10 & 0.987583 \\ 100 & 0.998751 \\ 1000 & 0.999875 \\ 10000 & 0.999988 \end{pmatrix}$$

Analytically, we can use **FunctionExpand** to replace the binomial by gamma functions.

$$\mathrm{FunctionExpand}[f(n)]$$

$$\frac{(-1)^n\,\pi}{\sqrt{n}\ \Gamma\left(\frac{1}{2}-n\right)\Gamma(n)}$$

Next, we use the reflection formula, $\Gamma(z)\,\Gamma(1-z) = \pi \csc(\pi\, z)$, here implemented via pattern matching.

$$\%\ /.\ \Gamma\left(\frac{1}{2}-n\right) \to \frac{\mathrm{FullSimplify}\left[\Gamma\left(\frac{1}{2}-n\right)\Gamma\left(n+\frac{1}{2}\right)\right]}{\Gamma\left(n+\frac{1}{2}\right)}$$

$$\frac{(-1)^n \cos(n\,\pi)\,\Gamma\left(n+\frac{1}{2}\right)}{\sqrt{n}\ \Gamma(n)}$$

This expression simplifies, noting that $n \in \mathbb{Z}$.

```
Simplify[%, n ∈ ℤ]
```

$$\frac{\Gamma\left(n+\frac{1}{2}\right)}{\sqrt{n}\;\Gamma(n)}$$

Since we are interested in the asymptotic limit, $n \to \infty$, we use **Series** (effectively using Stirling's formula).

```
% /. Γ(n_) → Normal[Series[Γ(n), {n, ∞, 1}]]
```

$$\frac{\sqrt{\frac{1}{n}}\;n^{\frac{1}{2}-n}\sqrt{\frac{1}{n+\frac{1}{2}}}\;\left(n+\frac{1}{2}\right)^{n+\frac{1}{2}}}{\sqrt{e}}$$

Noting that $n > 0$, followed by computing the limit as $n \to \infty$, gives us the desired result.

```
Simplify[%, n > 0]
```

$$\frac{n^{-n}\left(n+\frac{1}{2}\right)^{n}}{\sqrt{e}}$$

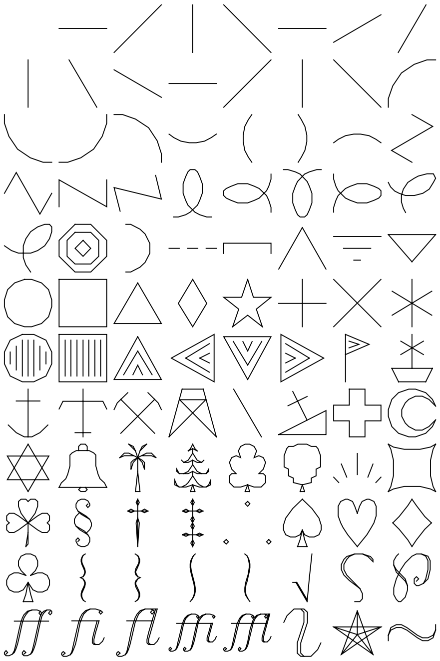
$$\lim_{n\to\infty} \%$$

$$1$$

■ 3D Text

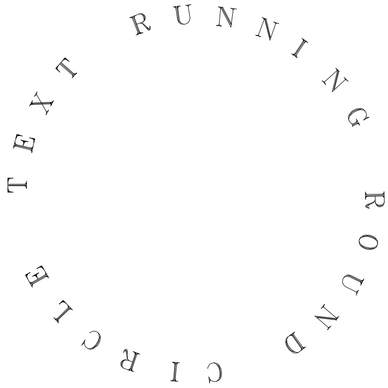
Q: Is there a way to produce 2D text for inclusion within 3D graphics?

A: See "Hershey Text in *Mathematica* Graphics" by Roland Jakschewitz at library.wolfram.com/database/Conferences/196. Using the public domain Hershey font descriptions, it is easy to produce machine-independent, scalable, and rotatable text that can then be projected onto **Graphics3D** surfaces. Here are a few examples of the more than a thousand glyphs included in the Hershey vector fonts.

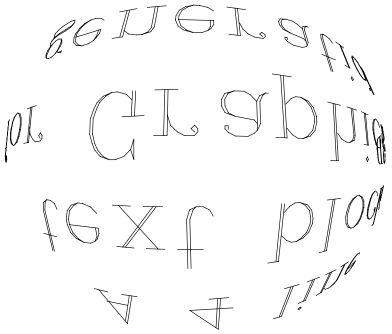


Here are two examples, taken from Jakschewitz's conference notebook.

- Text running around a circle.



- Text on the surface of a sphere.



■ Singular Values

Q: I am computing the singular values of a matrix consisting of physical data. If I have some estimate of the error in each matrix element, is there a simple way to compute the error in the singular values?

A: The singular value decomposition for a numerical matrix A consists of two matrices, U and V, and a list of singular values. Forming the diagonal matrix Σ from the singular values, the matrix decomposition reads

$$U^\dagger.\Sigma.V = A,$$

where † denotes the conjugate transpose. This operator can be implemented as follows.

```
x_^† /; MatrixQ[x] := Conjugate[x^T]
```

A can be complex and, in general, is not square. U and V are row orthonormal matrices, that is, $U.U^\dagger = I$ and $V.V^\dagger = I$. Hence

$$U^\dagger.\Sigma.V = A \Rightarrow U.A.V^\dagger = U.U^\dagger.\Sigma.V.V^\dagger = \Sigma.$$

In other words, U and V effectively “diagonalize” A.

Here is a random 4×5 matrix.

```
A = Table[Random[], {4}, {5}]
```

$$\begin{pmatrix} 0.332332 & 0.980599 & 0.393794 & 0.990528 & 0.703967 \\ 0.944597 & 0.268763 & 0.696153 & 0.833778 & 0.315905 \\ 0.635055 & 0.0812983 & 0.887024 & 0.3734 & 0.435773 \\ 0.817559 & 0.58996 & 0.133198 & 0.961757 & 0.350307 \end{pmatrix}$$

Compute the singular values, σ, and the matrices U, V, and Σ.

```
{U, σ, V} = SingularValues[A]; Σ = DiagonalMatrix[σ]
```

$$\begin{pmatrix} 2.70688 & 0 & 0 & 0 \\ 0 & 0.946705 & 0 & 0 \\ 0 & 0 & 0.56856 & 0 \\ 0 & 0 & 0 & 0.0956545 \end{pmatrix}$$

Since we are dealing with numerical matrices, to check whether $U^{\dagger}.\Sigma.V = A$, we use the Euclidean norm, $\|M\|$, of a matrix M, defined by $\|M\| = \sqrt{\mathrm{Tr}(M^{\dagger}.M)}$ (using DoubleBracketingBar).

```
||x_?MatrixQ|| := √Tr[x†.x]
```

```
||U†.Σ.V − A||
```

2.56255×10^{-15}

Similarly, we confirm that $U.A.V^{\dagger} = \Sigma$.

```
||U.A.V† − Σ||
```

1.33537×10^{-15}

It is easy to obtain an expression for the sensitivity of the kth singular value, σ_k, to small changes in the element $a_{i,j}$. Since

$$U.A.V^{\dagger} = \Sigma \Leftrightarrow u_k.A.v_k^{*} = \sum\nolimits_{i,j} u_{k,i}\, a_{i,j}\, v_{k,j}^{*} = \sigma_k,$$

we have that

$$\frac{\partial \sigma_k}{\partial a_{i,j}} = u_{k,i}\, v_{k,j}^{*}.$$

Hence, the total change in σ_k, due to changes in any of the elements of A, is

$$\Delta\sigma_k = \sum_{i,j} \Delta a_{i,j} \frac{\partial \sigma_k}{\partial a_{i,j}} = \sum_{i,j} u_{k,i}\, \Delta a_{i,j}\, v_{k,j}^{*} = \{U.\Delta A.V^{\dagger}\}_{k,k}.$$

This result is not surprising. We can obtain it directly as follows:

$$U.(A+\Delta A).V^{\dagger} = \Sigma + \Delta\Sigma \Rightarrow U.\Delta A.V^{\dagger} = \Delta\Sigma.$$

For example, form the matrix B by perturbing $a_{2,3}$ by 10^{-3}.

$$\boldsymbol{B = A;\ B[\![2,\,3]\!] \mathrel{+}= 10^{-3}}$$

$$0.697153$$

The change in the singular values, $\Delta\sigma_k$, is given by the diagonal entries of $U.\Delta A.V^{\dagger}$.

$$\boldsymbol{\Delta \mathrm{A} = B - A}$$

$$\begin{pmatrix} 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.001 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

$$\boldsymbol{\Delta\sigma = \mathrm{Tr}[U.\Delta\mathrm{A}.V^{\dagger},\ \mathrm{List}]}$$

$$\{0.000195748,\ 0.000252514,\ -0.000169345,\ 0.000245803\}$$

These values should be compared with those obtained by computing the difference of the singular values of B and A.

$$\boldsymbol{\mathrm{SingularValues}[B][\![2]\!] - \sigma}$$

$$\{0.000195821,\ 0.000252629,\ -0.000169175,\ 0.000245834\}$$

The agreement is very good.

Using this approach, we can also compute the effect of an arbitrary *symbolic* perturbation.

$$\boldsymbol{B = A;\ B[\![3,\,1]\!] \mathrel{+}= \epsilon;}$$

$$\boldsymbol{\Delta \mathrm{A} = B - A;}$$

$$\boldsymbol{\mathrm{Tr}[U.\Delta\mathrm{A}.V^{\dagger},\ \mathrm{List}]\ //\ \mathrm{Chop}}$$

$$\{0.191993\,\epsilon,\ 0.259727\,\epsilon,\ -0.263461\,\epsilon,\ 0.201366\,\epsilon\}$$

Paul Abbott
Department of Physics
University of Western Australia
35 Stirling Highway

Crawley WA 6009, Australia
tmj@physics.uwa.edu.au
physics.uwa.edu.au/~paul