Example of Solve

Here is a quite challenging example. Consider the triadicTagaki function

\[
T(x) = \begin{cases} 
\frac{1}{3} T(3x) + x, & 0 \leq x \leq \frac{1}{3} \\
\frac{1}{3} T(3x - 1) + \frac{1}{3}, & \frac{1}{3} \leq x \leq \frac{2}{3} \\
\frac{1}{3} T(3x - 2) - x + 1, & \frac{2}{3} \leq x \leq 1
\end{cases}
\]

In order to solve this functional equation numerically, we discretize the equation at finite amount of points.

\[
n = 5;
\]

\[
\text{eqs} = \text{Table}[
T[x] \to \text{Which}[
    x < 1/3, T[3x]/3 + x, \\
x < 2/3, T[3x - 1]/3 + 1/3, \\
    \text{True}, T[3x - 2]/3 - x + 1] = 0, (x, 0, 1, 1/n)]
\]

\[
\{0.666667 T[0] == 0, -\frac{1}{5} + T[\frac{1}{5}] - 0.333333 T[\frac{3}{5}] == 0,
-\frac{1}{3} - 0.333333 T[\frac{1}{5}] + T[\frac{2}{5}] == 0, -\frac{1}{3} + T[\frac{3}{5}] - 0.333333 T[\frac{4}{5}] == 0,
-\frac{1}{5} - 0.333333 T[\frac{2}{5}] + T[\frac{4}{5}] == 0, 0.666667 T[1] == 0\}
\]

Note, instead of integers we are using floating-point numbers. We need to solve this sparse system

\[
\text{sol} = \text{Solve}[\text{eqs}, \text{Table}[T[x], \{x, 0, 1, 1/n\}]] \quad \text{// First}
\]

\[
\{T[0] \to 0., T[\frac{1}{5}] \to 0.35, T[\frac{2}{5}] \to 0.45,
T[\frac{3}{5}] \to 0.45, T[\frac{4}{5}] \to 0.35, T[1] \to 0.\}
\]

Each solution has a form \( T[\frac{1}{5}] \to 0.35 \). We convert it to a point \((x, y)\), where \(x\) is the argument of \(T\) and \(y\) is 0.35:
Finally we use the primitive Polygon to draw the picture:

\[
\text{Show[Graphics[Polygon[({[1, 1], [[2]) & /@ sol}], \\
Frame -> True, AspectRatio -> Automatic];}
\]

WE solve a system of 1000 equations:

\[
n = 1000; \text{eqs = Table[}
T[x] - \text{Which[}x < 1 / 3, T[3 x] / 3. + x, x < 2 / 3, T[3 x - 1] / 3. + 1 / 3, \\
\text{True, T[3 x - 2] / 3. - x + 1] = 0, \{x, 0, 1, 1/n\}];} \\
\text{sol = Solve[eqs, Table[T[x], \{x, 0, 1, 1/n\}]] // First;} \\
\text{Show[Graphics[Polygon[({[1, 1], [[2]) & /@ sol}], \\
Frame -> True, AspectRatio -> Automatic];}
\]

Problem 1.
For an arbitrary 3x3 matrix A, verify the identity

\[
A^3 - \text{tr}(A) A^2 + \frac{1}{2} (\text{tr}(A)^2 - \text{tr}(A^2)) A - \text{det}(A) I = 0,
\]

where \text{tr} is the trace, \text{det} is the determinant, and I is the 3D identity matrix
A = Table[Random[], {i, 3}, {j, 3}]

{{0.477337, 0.944827, 0.140635},
  {0.659826, 0.72787, 0.827159}, {0.0760579, 0.562205, 0.960679}}

A.A.A - Tr[A] A . A +
1/2 (Tr[A]^2 - Tr[A.A]) A - Det[A] IdentityMatrix[3]

{{0., 0., 0.}, {0., 0., -4.44089×10^{-16}}, {0., 0., 0.}}

**Inner and Outer**

As it was mentioned before, the function Dot gives the scalar product:

Clear[x, y, z];
{x, y, z}.{1, 2, 3}

x + 2 y + 3 z

The function Inner is the generalization of a scalar product:

Clear[f, g];
Inner[f, {a, b, c}, {1, 2, 3}, g]
g[f[a, 1], f[b, 2], f[c, 3]]

Here f and g are any functions. If f is Times and g is Plus, we have a dot product

Inner[Times, {a, b, c}, {1, 2, 3}, Plus]
a + 2 b + 3 c

By changing f and g, we can do various manipulations with two lists:

Inner[Power, {a, b, c}, {1, 2, 3}, Plus]
a + b^2 + c^3

f and g can be pure functions as well

Inner[#1/#2 &, {a, b, c}, {1, 2, 3}, Plus]
a + b/2 + c/3

**Problem 2.**

You have two points in the 3-dimensional Euclidean space.
A) Using Inner function, find a distance between them.

B) Same, but using Dot instead of Inner.

\[ a = \{x_1, y_1, z_1\}; \quad b = \{x_2, y_2, z_2\}; \]

\[ \text{Sqrt}\left[\text{Inner}\left[\left(\#1 - \#2\right)^2 \&, a, b, \text{Plus}\right]\right] \]

\[ \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \]

\[ \text{Inner}\left[\left(\#1 - \#2\right)^2 \&, a, b, \text{Sqrt}[\text{Plus}[\#]] \&\right] \]

\[ \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \]

\[ \text{Sqrt}[\text{Plus}@@\left((a - b)^2\right)] \]

\[ \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \]

\[ \text{Sqrt}[\left(a - b\right) \cdot \left(b - a\right)] \]

\[ \sqrt{(x_1 - x_2)
\begin{align*}
x_1 - x_2 &\quad \pm \quad x_1 + x_2 \\
y_1 - y_2 &\quad \pm \quad y_1 + y_2 \\
z_1 - z_2 &\quad \pm \quad z_1 + z_2
\end{align*} \]

The tensor product (or Kronecker product) is given by Outer, which combines each element from one list with each element of another:

\[ \text{Clear}[a, b]; \]
\[ \text{Outer}[\text{List}, \{a, b, c\}, \{d, e, f\}] \quad /\!/ \text{MatrixForm} \]

\[ \begin{pmatrix} a & a & a \\ d & e & f \\ b & b & b \\ d & e & f \\ c & c & c \\ d & e & f \end{pmatrix} \]

\[ \text{Outer}[\text{Power}, \{a, b, c\}, \{d, e, f\}] \]

\[ \{(a^d, a^e, a^f), (b^d, b^e, b^f), (c^d, c^e, c^f)\} \]

One of the "application" of Outer is to represent a card deck. We have two lists:

\[ x = \{s, h, d, c\} \]

\[ \{s, h, d, c\} \]
Clear[A];
y = Join[Ranger[2, 10], {J, Q, K, A}]

(2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A)

Now we need to create all possible combinations:

Outer[List, x, y];

Length[deck] = 52

**Problem 3.**

Implement "deal" in Mathematica: randomly picked 5 cards and distribute them to players. You will need functions Delete and Complement.

Delete[deck, Random[Integer, {1, Length[deck]}]];  
Length[%] = 51

remove[d_] := Delete[d, Random[Integer, {1, Length[d]}]]  
Complement[deck, Nest[remove, deck, 5]]  

((d, 6), {h, A}, {s, 5}, {s, 8}, {s, J})
Groebner Basis

The concept of Gröbner Bases was introduced by Bruno Buchberger in 1965 in the context of his work on performing algorithmic computations in residue classes of polynomial rings. Buchberger's algorithm for computing Gröbner Bases is a powerful tool for solving many important problems in polynomial ideal theory. The Algorithm was named after Wolfgang Gröbner who was the Ph.D. advisor to Buchberger and who stimulated the research on the subject. Gröbner basis for a set of polynomials is a set of "simpler" polynomials with the same common zeros as the original polynomial. In Mathematica GroebnerBasis is underlying tool of Solve, Eliminate and few other functions. Here is an example,

Clear[@x, y, z];
eqn = {x^2 + 3 x + y + 1, y^2 + 4 x^2 + 3 y + 1}
(1 + 3 x + x^2 + y, 1 + 4 x^2 + 3 y + y^2)

GroebnerBasis[eqn, {x, y}]
(45 + 114 y^2 - 2 y^3 + y^4, 3 + 12 x + y - y^2)

These polynomials look differently than the originals, however, they have the same zeros. The new set is simpler and demonstrates the structure of the system. Having the Groebner basis, we can solve each equation independently. This is another example

GroebnerBasis[{x^2 + y^2 + 2, x^2 + z^2 + 2, y^2 + z^2 + 2}, {x, y, z}]
{-1 - z^2, -1 - y^2, -1 - x^2}

Surely this set of polynomials is simpler.

The Groebner basis is not unique. Changing the order of variables (the second argument in GroebnerBasis), leads to a different set of polynomials:

GroebnerBasis[{x^2 - y^3 + z, x + y + z, x - y + z^3}, {x, y, z}]
(8 z + 2 z^2 + 3 z^4 - 3 z^5 - 2 z^6 + 3 z^7 - z^9, 2 y + z - z^3, 2 x + z + z^3)

GroebnerBasis[{x^2 - y^3 + z, x + y + z, x - y + z^3}, {z, y, x}]
(-50 x + 45 x^2 - 39 x^3 + 40 x^4 + 3 x^6 + x^7,
70 x + 47 x^2 - 90 x^3 - 10 x^4 - 14 x^5 - 3 x^6 + 180 x y,
-x + x^2 - y + y^3, x + y + z)

The Groebner basis is used for eliminating extra variables:

polys = {x*(s^2 + t^2 + 1) - (s^2 - 1 - t^2), y*(s^2 + t^2 + 1) - 2*s,
        z*(s^2 + t^2 + 1) - 2*s*t, w*(s^2 + t^2 + 1) - 1};
Eliminate[
{x*(s^2+t^2+1) - (s^2 - 1 - t^2) == 0, y*(s^2 + t^2 + 1) - 2*s == 0,
  z*(s^2 + t^2 + 1) - 2*s*t == 0, w*(s^2 + t^2 + 1) - 1 == 0}, {s, t, w}]

1 - y^2 - z^2 == x^2

In GroebnerBasis we use the third argument - a list of variables to be eliminated:

GroebnerBasis[polys, {x, y, z}, {s, t, w}]

(1 - x^2 - y^2 - z^2)

Problem 4.
Given an ellipse. Under which conditions are arbitrary four points of an ellipse lying on a circle.

ParametricPlot[{2.5 Cos[t], 0.8 Sin[t]},
{t, 0, 2 Pi}, AspectRatio -> Automatic];
ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 Pi}, AspectRatio -> Automatic];

Show[%, %%]

NSolve[{x^2 + y^2 == 1, x^2/2.5^2 + y^2/0.8^2 == 1}, {x, y}]

GroebnerBasis[{x^2 + y^2 - r^2, x^2 b^2 + y^2 a^2 - a^2 b^2}, {a, b, r}, {x}] // Factor

(-a^2 b^2 + b^2 r^2 + a^2 y^2 - b^2 y^2)
We rewrite the last equation:

\[(a^2 - b^2) y^2 = b^2 (a^2 - r^2)\]

From here it's easy to see when the equation has real solutions.