Solution of Nonlinear Equations with Uncertain Parameters

Talk Outline

- Toy example
- Transformation of a PDF
- Homotopy
- How to solve nonlinear equation via homotopy
- Stochastic differential equation
- Ito process
- Positioning by Geodetic Resection
- Non Gaussian Noise
- Promotion

Toy example

Let us consider the following polynomial, where δ is a Gaussian noise with $\delta = \mathcal{N}(0, \sigma)$

```
In[]:= q = x^2 + (8 + δ) x - 9;
```

Its algebraic solution

```
In[]:= sol = Solve[q == 0, x]
```

Out[]=

$$\left\{\left\{x \to \frac{1}{2}\left(-8-\delta-\sqrt{100+16\,\delta+\delta^2}\right)\right\}, \left\{x \to \frac{1}{2}\left(-8-\delta+\sqrt{100+16\,\delta+\delta^2}\right)\right\}\right\}$$

Let us consider the second solution

```
In[]:= s = x /. First[sol〚2〛]
```

Out[]=

$$\frac{1}{2}\left(-8-\delta+\sqrt{100+16\,\delta+\delta^2}\right)$$

To get the distribution of s we transform of the distribution of δ. Let

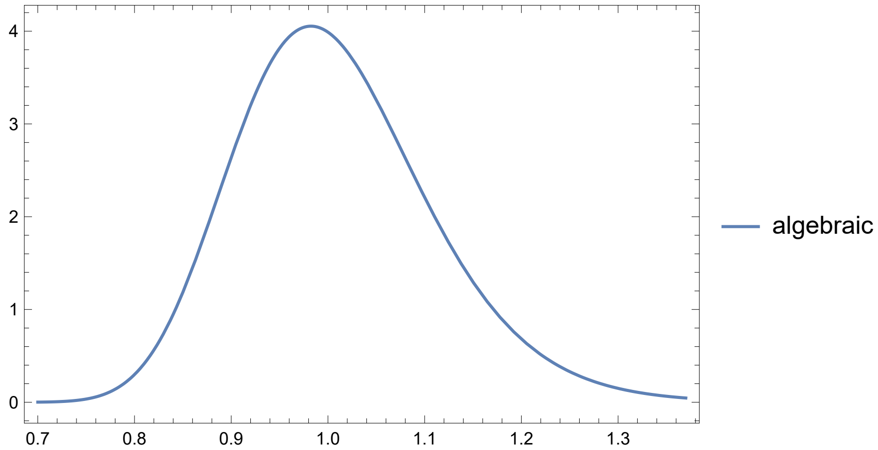
```
In[]:= σ = 1;
```

In[∘]:=
```
𝒟 = TransformedDistribution[s, δ ≈ NormalDistribution[0, σ]];
```

The PDF of $\mathcal{D}$

In[∘]:=
```
p0 = Plot[PDF[𝒟, ξ] // Evaluate,
  {ξ, 0.7, 1.37}, Frame → True, PlotLegends → {"algebraic"}]
```

Out[∘]=



The PDF of the second solution

The mean value

In[∘]:=
$$\mu_2 = \int_{-\infty}^{\infty} \xi \, \text{PDF}[\mathcal{D}, \xi] \, d\xi \; // \; N$$

Out[∘]=
```
1.00915
```

The standard deviation

In[∘]:=
$$\sigma_2 = \sqrt{\int_{-\infty}^{\infty} (\xi - \mu_2)^2 \, \text{PDF}[\mathcal{D}, \xi] \, d\xi} \; // \; N$$

Out[∘]=
```
0.103058
```

Is the result Gaussian too, $\mathcal{N}(\mu_2, \sigma_2)$?

Let us generate random samples

In[∘]:=
```
data = RandomVariate[𝒟, 10 000];
```

and find a fitted distribution

In[∘]:=
```
FindDistribution[data, TargetFunctions → {NormalDistribution}]
```

Out[∘]=
```
NormalDistribution[1.01011, 0.103581]
```

Transformation of a PDF

What does the function `TransformedDistribution[]` do?

Illustration

Assuming that our density function as

$$f_X(\xi) = e^{-\xi}$$

Let consider the following transform

$$Y = \sqrt{X} \text{ or in general } Y = \phi(\mathrm{X})$$

the inverse transform is

$$\mathrm{X} = Y^2 \text{ or } \mathrm{X} = \phi^{-1}(\mathrm{Y})$$

where $\phi^{-1}(.) = (.)^2$

then the transformed PDF can be computed as

$$f_Y(\xi) = f_X\left(\phi^{-1}(\xi)\right)\left|\frac{d\,\phi^{-1}(\xi)}{d\xi}\right| = e^{-\xi^2}\,|\,2\,\xi\,|$$

Employing the built-in function

```
In[•]:= fX = PDF[ExponentialDistribution[1], ξ]
Out[•]=
```

$$\begin{cases} e^{-\xi} & \xi \geq 0 \\ 0 & \text{True} \end{cases}$$

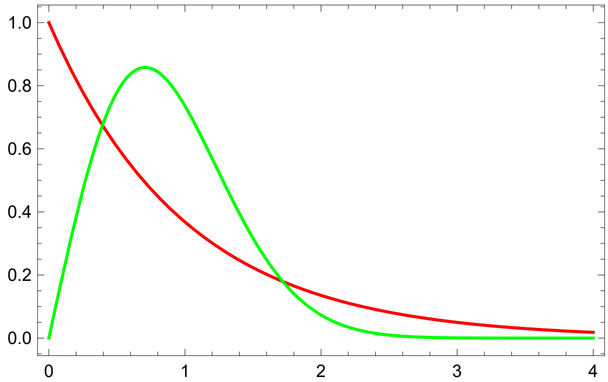
```
In[•]:= Y = TransformedDistribution[√X, X ≈ ExponentialDistribution[1]]
Out[•]=
WeibullDistribution[2, 1]
```

$$x^{\alpha-1}\,e^{-(x/\beta)^{\alpha}}$$

```
In[•]:= fY = PDF[Y, ξ]
Out[•]=
```

$$\begin{cases} 2\,e^{-\xi^2}\,\xi & \xi > 0 \\ 0 & \text{True} \end{cases}$$

Out[∘]=



The original PDF (red) and its transformed form (green)

Homotopy

Concept of homotopy

The continuous deformation of an object to an other object is known as *homotopy*. Let us consider a simple geometric example. Define the homotopy between a circle and a square.

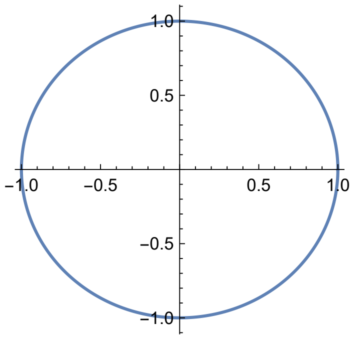
$$x = R\cos(\alpha)$$
$$y = R\sin(\alpha)$$

```
In[1]:= Clear["Global`*"]

In[2]:= circle = Table[{Cos[α], Sin[α]}, {α, 0, 2 Pi, 0.02}];

In[3]:= ListPlot[circle, Joined → True, AspectRatio → 1]
```

Out[3]=



while the parametric equations of the square

$$x = f(\alpha)\, R\cos(\alpha)$$
$$y = f(\alpha)\, R\sin(\alpha)$$

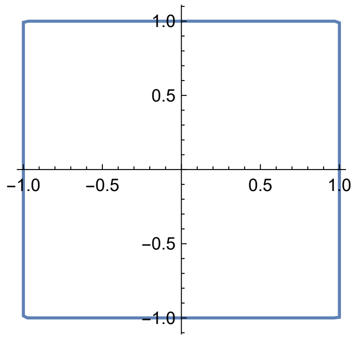
where

$$f(\alpha) = \frac{1}{\max(|\sin(\alpha)|, |\cos(\alpha)|)}$$

```
In[4]:= square = Table[{Cos[α], Sin[α]} 1/(Max[Abs[Sin[α]], Abs[Cos[α]]]), {α, 0, 2 Pi, 0.02}];

In[5]:= ListPlot[square, Joined → True, AspectRatio → 1, ImageSize → Small]
```

Out[5]=



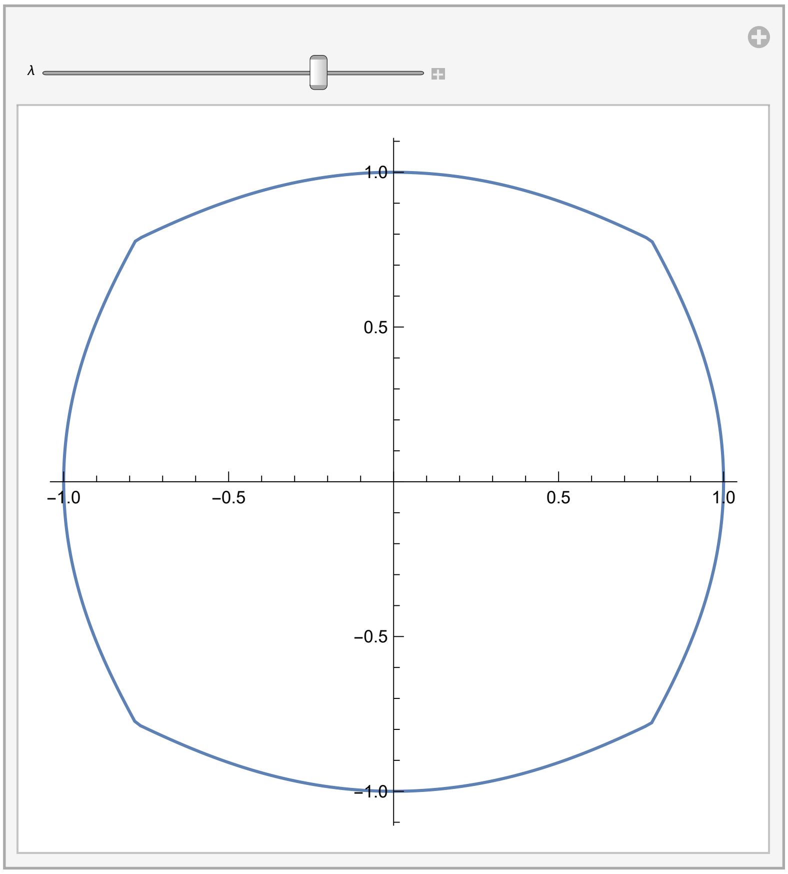
A square

Now the homotopy function

$$H(\alpha,\ \lambda) = \lambda \begin{pmatrix} R\cos(\alpha) \\ R\sin(\alpha) \end{pmatrix} + (1-\lambda) \begin{pmatrix} f(\alpha)\,R\cos(\alpha) \\ f(\alpha)\,R\sin(\alpha) \end{pmatrix}$$

```
In[6]:= Manipulate[
  ListPlot[λ circle + (1 - λ) square, Joined → True, AspectRatio → 1] // Quiet, {λ, 0, 1}]
```

Out[6]=



Deformation of square into circle vice versa

Solving nonlinear equation via homotopy

Target system, we need to solve

In[∘]:=
```
q[x_] := x^2 + 8 x - 9
```

Start system, easy to solve

In[∘]:=
```
p[x_] := x^2 - 9
```

The linear homotopy function

In[∘]:=
```
H[x_, λ_] := (1 - λ) p[x] + λ q[x]
```

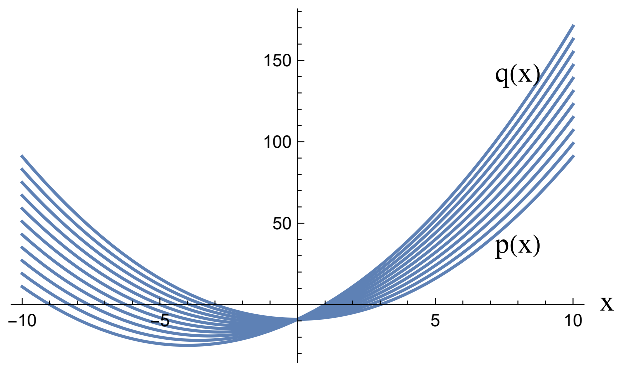
or

In[∘]:=
```
H[x, λ] // Expand
```

Out[∘]=

$-9 + x^2 + 8\,x\,\lambda$

Out[∘]=



Deformation of the function H from $p\,(\mathrm{x})$ to $q\,(\mathrm{x})$ as function of λ

Homotopy continuation method deforms $p(\mathrm{x}) = 0$, the known roots of the start system, into $q(\mathrm{x}) = 0$, the roots of the target system.

One can solve $H(\mathrm{x}, \lambda) = 0$ for different values of λ. Considering the positive root, x=3

In[∘]:=
```
x0 = 3; λ1 = 0.2; x1 = x /. FindRoot[H[x, λ1] == 0, {x, x0}]
```

Out[∘]=

2.30483

Using the result as guess value for the next solution step

In[∘]:=
```
λ2 = 0.4; x2 = x /. FindRoot[H[x, λ2] == 0, {x, x1}]
```

Out[∘]=

1.8

and so on,

In[∘]:=
```
λ3 = 0.6; x3 = x /. FindRoot[H[x, λ3] == 0, {x, x2}]
```

Out[∘]=

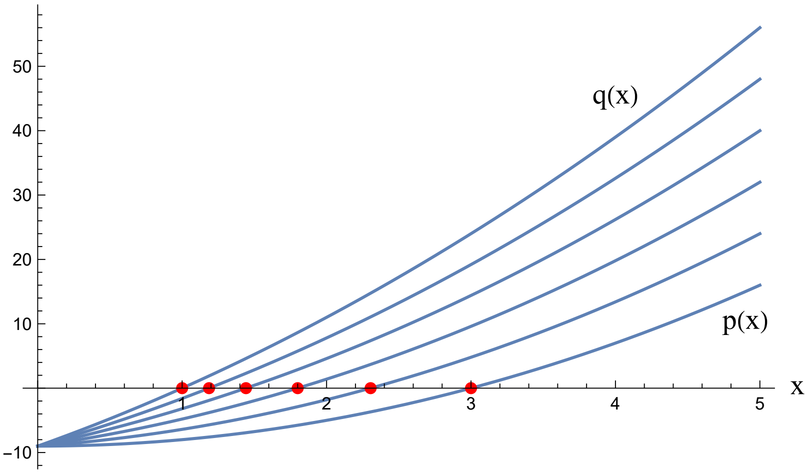
1.44187

```
In[]:= λ4 = 0.8; x4 = x /. FindRoot[H[x, λ4] == 0, {x, x3}]
Out[]=
1.18634
```

```
In[]:= λ5 = 1; x5 = x /. FindRoot[H[x, λ5] == 0, {x, x4}]
Out[]=
1.
```

Let us display the transition of a root of the polynomial $p(\mathrm{x})$ into a root of the polynomial $q(\mathrm{x})$,

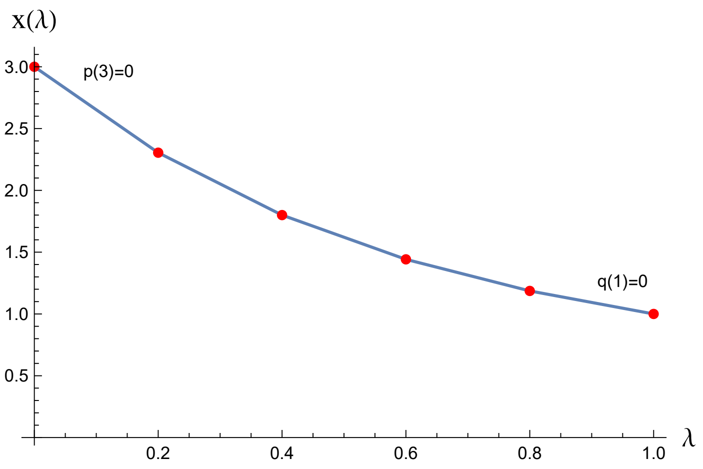
Out[]=



The transition of the root from $x = 3$ to $x = 1$ during the deformation of the function H

The homotopy path is the function $x = x\,(\lambda)$, where in every point $H\,(\mathrm{x},\lambda) = 0$.
See figure below shows the path of homotopy transition of the root of $p(\mathrm{x})$ into the root of $q(\mathrm{x})$.

Out[]=



The homotopy path is the function $x = x\,(\lambda)$, where in every point of $H = 0$.

Tracing homotopy path as initial value problem

However, one can consider this root tracing procedure as an initial value problem of an ordinary differential equation. Since $H\,(x,\,\lambda) = 0$ for every $\lambda \in [0,\,1]$, therefore

$$dH(x, \lambda) = \frac{\partial H}{\partial x}\, dx + \frac{\partial H}{\partial \lambda}\, d\lambda \equiv 0 \qquad \lambda \in [0, 1]$$

Then the initial value problem is

$$H_x \frac{dx(\lambda)}{d\lambda} + H_\lambda = 0$$

with

$$x(0) = x_0$$

In case of n variables H_x is the Jacobian of H respect to x_i, $i = 1,\ldots,n$.

In our single variable case, the two partial derivatives of the homotopy function are

```
In[•]:= dHdλ = D[H[x, λ], λ]
Out[•]=
8 x
```

```
In[•]:= dHdx = D[H[x, λ], x]
Out[•]=
2 x (1 - λ) + (8 + 2 x) λ
```

Then the right hand side of the differential equation to be solved is

```
In[•]:= deqrhs = -dHdλ/dHdx /. x → x[λ]
Out[•]=
-8 x[λ]/(2 (1 - λ) x[λ] + λ (8 + 2 x[λ]))
```

The differential equation,

```
In[•]:= deq = D[x[λ], λ] == deqrhs
Out[•]=
x'[λ] == -8 x[λ]/(2 (1 - λ) x[λ] + λ (8 + 2 x[λ]))
```

The initial value is

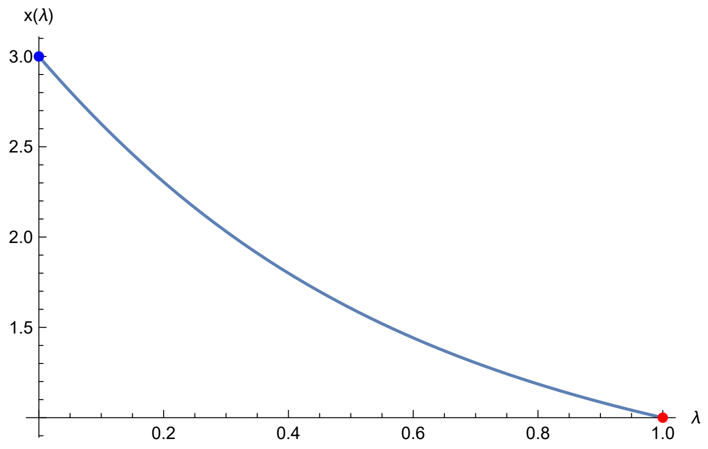
```
In[•]:= x0 = 3
Out[•]=
3
```

The numerical solution

```
In[•]:= sol = NDSolve[{deq, x[0] == x0}, {x[λ]}, {λ, 0, 1}];
```

The trajectory is the homotopy path,

Out[∘]=



The trajectory of the solution as the homotopy path

The value of the corresponding root of $q(\mathrm{x})$ is $x(\lambda)$ at $\lambda = 1$,

In[∘]:=
```
First[x[λ] /. sol /. λ → 1]
```
Out[∘]=
```
1.
```

Stochastic form of the homotopy differential equation

The target system,

In[∘]:=
```
q = x^2 + (8 + δ) x - 9;
```

The start system,

In[∘]:=
```
p = x^2 - 9;
```

The linear homotopy function is

In[∘]:=
```
H = (1 - λ) p + λ q;
```

To get the differential equation form, we compute the partial derivates of the homotopy function,

In[∘]:=
```
dHx = D[H, x]
```
Out[∘]=

$$2\,x\,(1-\lambda) + (8 + 2\,x + \delta)\,\lambda$$

and

In[∘]:=
```
dHλ = D[H, λ]
```
Out[∘]=

$$x\,(8+\delta)$$

Then the right hand side of differential equation is

In[∘]:= $d = -\frac{dH\lambda}{dHx}$

Out[∘]=

$$-\frac{x\,(8+\delta)}{2\,x\,(1-\lambda) + (8 + 2\,x + \delta)\,\lambda}$$

Stochastic form of the homotopy differential equation

Now we should linearized this equation in δ at $\delta = 0$, in order to get Ito process form of the stochastic differential equation,

In[∘]:=
```
diff = (Series[d, {δ, 0, 1}] // Normal) /. x → x[λ]
```

Out[∘]=

$$-\frac{\delta\, x[\lambda]^2}{2\,(4\,\lambda + x[\lambda])^2} - \frac{4\, x[\lambda]}{4\,\lambda + x[\lambda]}$$

The coefficient of δ

In[∘]:=
```
pw = Coefficient[diff[[1]], δ]
```

Out[∘]=

$$-\frac{x[\lambda]^2}{2\,(4\,\lambda + x[\lambda])^2}$$

The independent part on δ

In[∘]:=
```
pλ = diff[[2]]
```

Out[∘]=

$$-\frac{4\, x[\lambda]}{4\,\lambda + x[\lambda]}$$

Then our linearized differential equation is

$$\frac{dx\,(\lambda)}{d\lambda} = p_\lambda + p_W\,\delta$$

Ito process

The integral form of this equation is

$$dx\,(\lambda) = \int p_\lambda\, d\lambda + \int p_W\,\delta\, d\lambda$$

Since the Gaussian noise is derivative of the Wiener process, namely

$$\frac{dW\,(\lambda)}{d\lambda} = \delta$$

Then the Ito process of our stochastic differential equation

$$dx\,(\lambda) = \int p_\lambda\, d\lambda + \int p_W\, dW$$

Let $\sigma = 1$ then

In[∘]:=
```
σ = 1.;
```

then

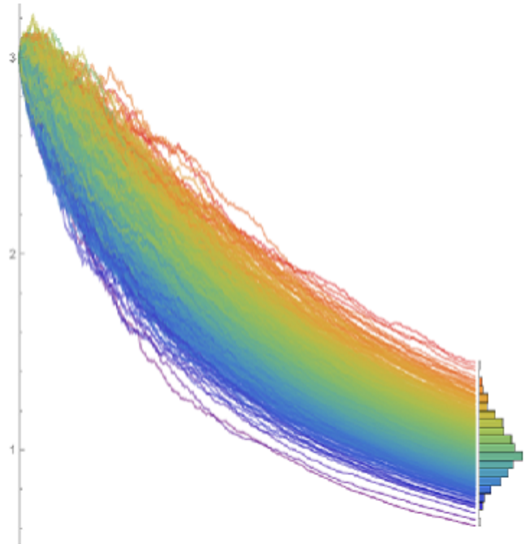
```
In[∘]:= proc = ItoProcess[ⅆx[λ] == pλ ⅆλ + pw ⅆw[λ], x[λ], {x, 3}, λ, w ≈ WienerProcess[0, σ]]
Out[∘]=
```

$$\text{ItoProcess}\left[\left\{\left\{\frac{-32\,\lambda\,x[\lambda]-8\,x[\lambda]^2}{2\,(4\,\lambda+x[\lambda])^2}\right\},\left\{\left\{-\frac{0.5\,x[\lambda]^2}{(4\,\lambda+x[\lambda])^2}\right\}\right\},x[\lambda]\right\},\{\{x\},\{3\}\},\{\lambda,0\}\right]$$

Generating 1000 trajectories with step size 0.01

```
In[∘]:= psol = RandomFunction[proc, {0, 1.0, 0.01},
          1000, Method → "StochasticRungeKuttaScalarNoise"]
Out[∘]=
TemporalData[ Time: 0. to 1.  Data points: 101 000  Paths: 1000 ]
```

```
In[∘]:= sd = psol["SliceData", 1];
```

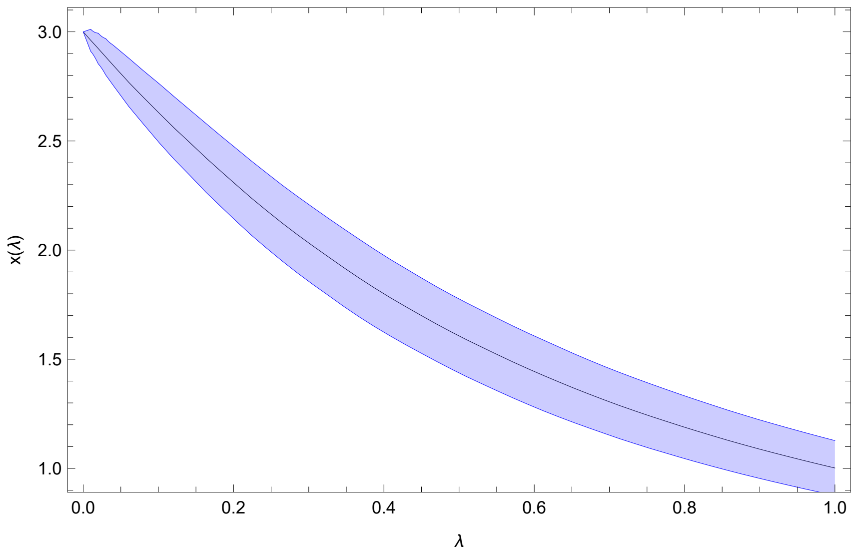


The generated trajectories and the histogram of the slice distribution at λ = 1

```
In[∘]:= p1 = Plot[Mean[psol[λ]], {λ, 0, 1},
          PlotStyle → {Black, Thin}, FrameLabel → {"λ", "x(λ)"}, Frame → True];
```

```
In[∘]:= p2 = Plot[{Mean[psol[λ]] + StandardDeviation[psol[λ]],
           Mean[psol[λ]] - StandardDeviation[psol[λ]]}, {λ, 0, 1}, Filling → {1 → {2}},
          PlotStyle → {{Blue, Thin}, {Blue, Thin}}, FrameLabel → {"λ", "x(λ)"}, Frame → True];
```

In[ ]:=
```
Show[{p1, p2}]
```
Out[ ]=



The trajectories of the Ito-solution - mean value and the standard deviation

The mean value of the solution is

In[ ]:=
```
m = Mean[psol[1]]
```
Out[ ]=
```
1.00192
```

and the standard deviation

In[ ]:=
```
s = StandardDeviation[psol[1]]
```
Out[ ]=
```
0.125574
```

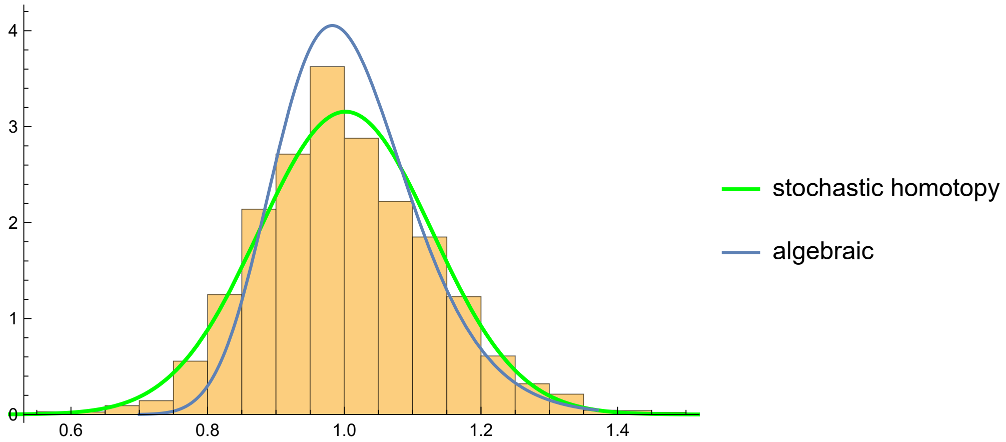
Verification of the type of distribution via simulation

In[ ]:=
```
data = RandomVariate[psol[1], 10 000];
```

In[ ]:=
```
ℋ = DistributionFitTest[data, Automatic, "HypothesisTestData"];
```

```
In[•]:= Show[Histogram[data, Automatic, "ProbabilityDensity"],
  Plot[PDF[ℋ["FittedDistribution"], x], {x, 0., 1.57},
   PlotStyle → {Green, Thick}, PlotLegends → {"stochastic homotopy"}], p0]
```

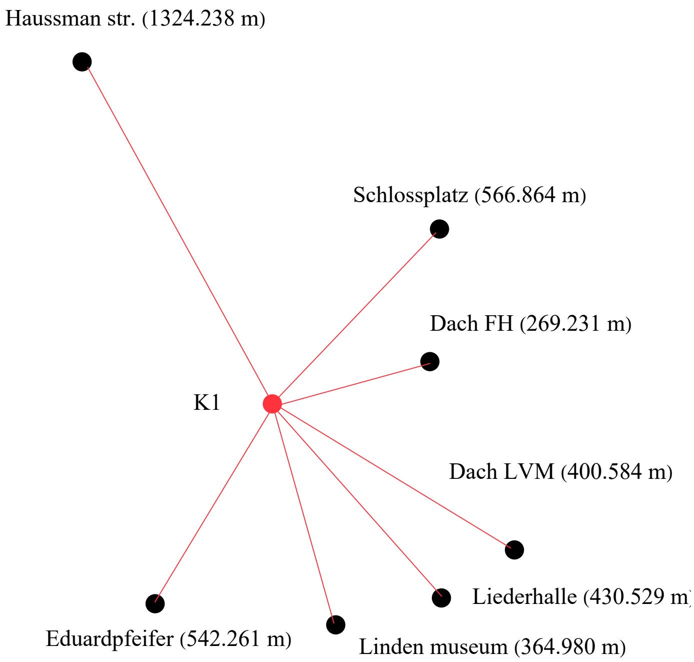
Out[•]=



The simulated pdf at $\lambda = 1$

Positioning by Geodetic Resection

Out[•]=



The Stuttgart Central Test Network

We are looking for the error distribution of the coordinates ($x0$, $y0$, $z0$) of the reference point (K1).

The GPS coordinates (X_i, Y_i, Z_i) in the global reference frame, $i = 1,...,7$. are in Table

Stations	X (m)	Y (m)	Z (m)
K1 (reference point)	(4 157 066.1116)	(671 429.6655)	(4 774 879.3704)
1	4 157 246.5346	671 877.0281	4 774 581.6314
2	4 156 749.5977	672 711.4554	4 774 981.5459
3	4 156 748.6829	671 171.9385	4 775 235.5483
4	4 157 066.8851	671 064.9381	4 774 865.8238
5	4 157 266.6181	671 099.1577	4 774 689.8536
6	4 157 307.5147	671 171.7006	4 774 690.5691
7	4 157 244.9515	671 338.5915	4 774 699.9070

The GPS coordinates of the stations

```
In[]:= Clear["Global`*"]
```

Then the input data are

```
In[]:= datan = {x1 → 4 157 246.5346, y1 → 671 877.0281, z1 → 4 774 581.6314, s1 → 566.8635 + δ,
    x2 → 4 156 749.5977, y2 → 672 711.4554, z2 → 4 774 981.5459, s2 → 1324.2380 + δ,
    x3 → 4 156 748.6829, y3 → 671 171.9385, z3 → 4 775 235.5483, s3 → 542.2609 + δ,
    x4 → 4 157 066.8851, y4 → 671 064.9381, z4 → 4 774 865.8238, s4 → 364.9797 + δ,
    x5 → 4 157 266.6181, y5 → 671 099.1577, z5 → 4 774 689.8536, s5 → 430.5286 + δ,
    x6 → 4 157 307.5147, y6 → 671 171.7006, z6 → 4 774 690.5691, s6 → 400.5837 + δ,
    x7 → 4 157 244.9515, y7 → 671 338.5915, z7 → 4 774 699.9070, s7 → 269.2309 + δ
   };
```

We have in the measured distances s_i with error δ. The number of stations is,

```
In[]:= m = 7;
```

Let us assigned the data as,

```
In[]:= X = Table[{xi, yi, zi}, {i, 1, m}] /. datan;
```

```
In[]:= Y = Table[si, {i, 1, m}] /. datan
Out[]=
{566.864 + δ, 1324.24 + δ, 542.261 + δ, 364.98 + δ, 430.529 + δ, 400.584 + δ, 269.231 + δ}
```

The prototype of the equations based on the implicit distance definition are,

```
In[]:= e = (xi - x0)^2 + (yi - y0)^2 + (zi - z0)^2 - si^2;
```

The objective function to be minimized is

```
In[]:= obj = Total[Table[e^2, {i, 1, m}]];
```

The error free solution via global minimization,

```
In[]:= sol = NMinimize[obj /. datan /. δ → 0, {x0, y0, z0}]
Out[]=
{0.00316074, {x0 → 4.15707 × 10^6, y0 → 671 430., z0 → 4.77488 × 10^6}}
```

In[•]:=
```
NumberForm[sol[[2]], 11]
```
Out[•]//NumberForm=
$\{x0 \to 4.1570661115 \times 10^6,\ y0 \to 671429.66548,\ z0 \to 4.7748793703 \times 10^6\}$

Now the objective function employing the prototype of the equations based on the explicit distance definition is,

In[•]:=
```
objr = Total[Table[(Sqrt[(x_i - x0)^2 + (y_i - y0)^2 + (z_i - z0)^2] - s_i)^2, {i, 1, m}]];
```

The solution is quiet similar,

In[•]:=
```
solr = NMinimize[objr /. datan /. δ → 0, {x0, y0, z0}]
```
Out[•]=
$\{2.26362 \times 10^{-9},\ \{x0 \to 4.15707 \times 10^6,\ y0 \to 671430.,\ z0 \to 4.77488 \times 10^6\}\}$

In[•]:=
```
NumberForm[solr[[2]], 11]
```
Out[•]//NumberForm=
$\{x0 \to 4.1570661115 \times 10^6,\ y0 \to 671429.66549,\ z0 \to 4.7748793703 \times 10^6\}$

Consequently we employ the equations based on the implicit definition, since this is a polynomial type of problem and Gröbner basis can be be used for elimination.

To use Groebner basis we need rationalize the data,

In[•]:=
```
datanR = Map[(#[[1]] → Rationalize[#[[2]], 10^-10]) &, datan];
```

Then the system of the polynomial equations representing the necessary condition of the optimum is,

In[•]:=
```
eqs = Map[(D[obj, #] /. datanR // Simplify) &, {x0, y0, z0}];
```

Algebraic solution

Let us find the solution, the probability distribution of the coordinate $x0$. Eliminating variables $y0$ and $z0$ employing reduced Gröbner basis, we get

In[•]:=
```
{grx0} = GroebnerBasis[eqs, x0, {y0, z0}, MonomialOrder → EliminationOrder];
```

Since the constant term is

In[•]:=
```
grx0[[1]] // N
```
Out[•]=
-2.90517×10^{164}

We normalize the coefficients of this polynomial,

In[•]:=
```
grx0n = grx0 / grx0[[1]] // N // Simplify;
```

This polynomial can not be solved symbolically since

In[•]:=
```
Exponent[grx0n, {x0, δ}]
```
Out[•]=
{7, 6}

Therefore we employ Taylor expansion at the error free solution of *x0*.

```
In[]:= x0P = x0 /. sol[[2]]
Out[]=
4.15707 × 10^6
```

Second order expansion is used,

```
In[]:= grx0nS = Series[grx0n, {x0, x0P, 2}] // Normal;
```

Now let us solve `grx0nS(x0,δ)=0` polynomial equation symbolically,

```
In[]:= x0δ = Solve[grx0nS == 0, x0] // Quiet;
```

Considering the first solution and neglecting higher order terms of δ than third one, we get

```
In[]:= β = Series[x0 /. x0δ[[1]], {δ, 0, 3}] // Normal
Out[]=
4.15707 × 10^6 + 0.339286 δ + 3.33517 × 10^-7 δ^2 + 4.90642 × 10^-13 δ^3
```

Considering the standard deviation of the error of the distant measurement as 1 cm,

```
In[]:= σ = 0.01;
```

Then the variable $x0$ as stochastic process is,

```
In[]:= 𝒟x0 = TransformedDistribution[β, δ ≈ NormalDistribution[0, σ]]
Out[]=
TransformedDistribution[4.15707 × 10^6 + 0.339286 x + 3.33517 × 10^-7 x^2 + 4.90642 × 10^-13 x^3,
  x ≈ NormalDistribution[0, 0.01]]
```

The mean and the standard deviation are,

```
In[]:= mx0 = Mean[𝒟x0]
Out[]=
4.15707 × 10^6

In[]:= NumberForm[mx0, 15]
Out[]//NumberForm=
4.15706611152965 × 10^6
```

and standard deviation

```
In[]:= sx0 = StandardDeviation[𝒟x0]
Out[]=
0.00339286
```

The probability density function can be computed employing random data set generated by the process,

```
In[]:= datax0 = RandomVariate[𝒟x0, 10000];

In[]:= datax00 = datax0 - x0P;
```

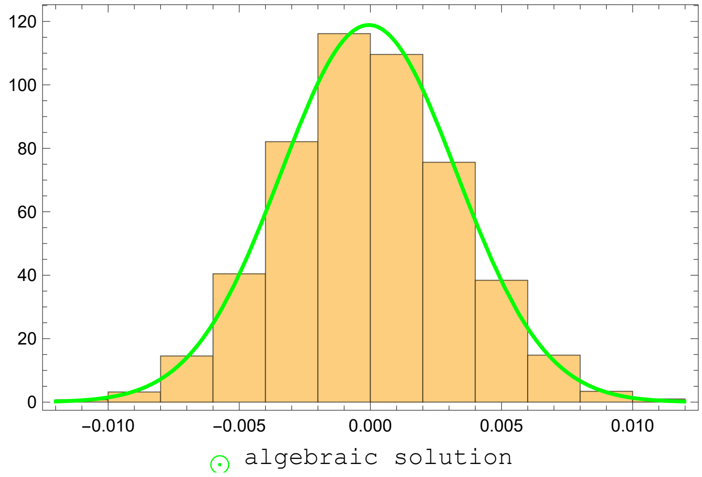
Then the displayed density function of x0 can be seen in figure below

```
In[]:= ℋ = DistributionFitTest[datax00, Automatic, "HypothesisTestData"];

In[]:= p11 = Plot[PDF[ℋ["FittedDistribution"], x], {x, Min[datax00], Max[datax00]},
    PlotStyle → {Green, Thick}, PlotLegends → Placed[" ⊙ algebraic solution", Below]];
```

```
In[]:= Show[{Histogram[datax00, Automatic, "ProbabilityDensity"], p11}, Frame → True]
```

Out[]=



The probability density function of $x0$ coordinates as stochastic variables

Stochastic solution

Now we show how to solve the problem via stochastic homotopy. The target system can be the second order system, **grx0nS** ($x0$, δ). Let us employ now x as independent variable,

```
In[]:= q = grx0nS /. x0 → x;
```

Now we employ *affine homotopy*. The homotopy function is

$$H\,(\mathrm{x},\,\lambda)\ = (1-\lambda)\,q'\,(\mathrm{x}_0)\,(\mathrm{x}-\mathrm{x}_0)+\lambda\,q\,(\mathrm{x})$$

where x_0 = x0P, see in section of the algebraic solution.

```
In[]:= x0 = x0P;
```

The start system is

```
In[]:= p = x - x0
```

Out[]=

$-4.15707\times10^{6}+x$

and derivate at x_0

```
In[]:= dqdx = D[q, x] /. x → x0
```

Out[]=

$1.48231\times10^{-21}-7.06819\times10^{-28}\,\delta-2.70834\times10^{-35}\,\delta^2+2.93084\times10^{-40}\,\delta^3+8.03014\times10^{-43}\,\delta^4+2.66134\times10^{-46}\,\delta^5-7.31937\times10^{-51}\,\delta^6$

Therefore the homotopy function can be written as,

```
In[]:= H = (1 - λ) dqdx p + λ q;
```

Then we can compute the Ito form,

```
In[•]:= dHx = D[H, x];
```

```
In[•]:= dHλ = D[H, λ];
```

The right hand side,

```
In[•]:= rhs = -dHλ/dHx // Simplify;
```

Now we should linearized this equation in δ at $\delta = 0$, in order to get Ito stochastic differential equation form,

```
In[•]:= diff = (Series[rhs, {δ, 0, 1}] // Normal) /. x → x[λ];
```

Therefore the terms of the Ito form are,

```
In[•]:= pw = Coefficient[diff[[2]], δ]
```

Out[•]=

$$\left(3.0963\times10^{-41}+8.44425\times10^{-42}\,\lambda-1.45379\times10^{-47}\,x[\lambda]-2.0313\times10^{-48}\,\lambda\,x[\lambda]+1.74858\times10^{-54}\,x[\lambda]^2\right)\Big/\left(1.48231\times10^{-21}+1.67903\times10^{-20}\,\lambda-4.03897\times10^{-27}\,\lambda\,x[\lambda]\right)^2$$

and

```
In[•]:= pλ = diff[[1]]
```

Out[•]=

$$\frac{2.01948\times10^{-27}\left(-4.15707\times10^{6}+x[\lambda]\right)^2}{1.48231\times10^{-21}+\lambda\left(1.67903\times10^{-20}-4.03897\times10^{-27}\,x[\lambda]\right)}$$

Consequently,

```
In[•]:= proc = ItoProcess[dx[λ] == pλ dλ + pw dw[λ], x[λ], {x, x0}, λ, w ≈ WienerProcess[0, σ]];
```

Generating 2000 trajectories with step size 0.001,

```
In[•]:= psol = RandomFunction[proc, {0, 1.0, 0.001},
  2000, Method → "StochasticRungeKuttaScalarNoise"]
```

Out[•]=

TemporalData[+ Time: 0. to 1. Data points: 2 002 000 Paths: 2000]

Data not in notebook. Store now

The mean value of the solution,

```
In[•]:= mxH = Mean[psol[1]]
```

Out[•]=

4.15707×10^{6}

```
In[•]:= NumberForm[mxH, 11]
```

Out[•]//NumberForm=

4.1570661116×10^{6}

and the standard deviation,

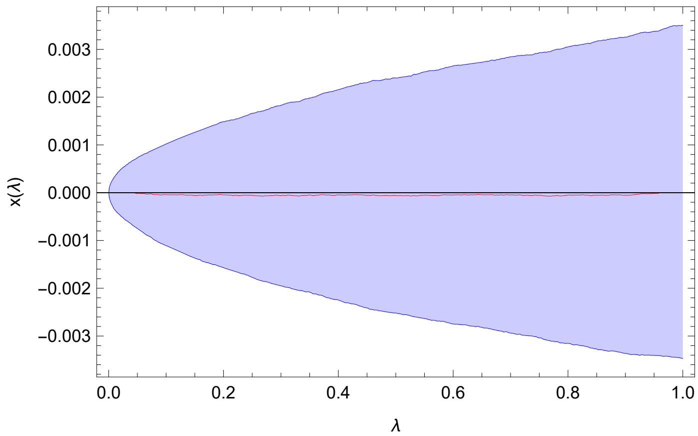
```
In[∘]:= sxH = StandardDeviation[psol[1]]
Out[∘]=
0.0033661
```

The trajectories of the solution of the Ito differential equation

```
In[∘]:= p1 = Plot[Mean[psol[λ]] - x0, {λ, 0, 1},
    PlotStyle → {Red, Thin}, FrameLabel → {"λ", "x(λ)"}, Frame → True];

In[∘]:= p2 = Plot[{Mean[psol[λ]] + StandardDeviation[psol[λ]] - x0,
     Mean[psol[λ]] - StandardDeviation[psol[λ]] - x0}, {λ, 0, 1}, Filling → {1 → {2}},
    PlotStyle → {{Blue, Thin}, {Blue, Thin}}, FrameLabel → {"λ", "x(λ)"}, Frame → True];

In[∘]:= Show[{p2, p1}]
Out[∘]=
```



The trajectories of the mean and the ± standard deviation

To find the density function of the distribution let us generate random values,

```
dataH = RandomVariate[psol[1], 10000] - x0;
```

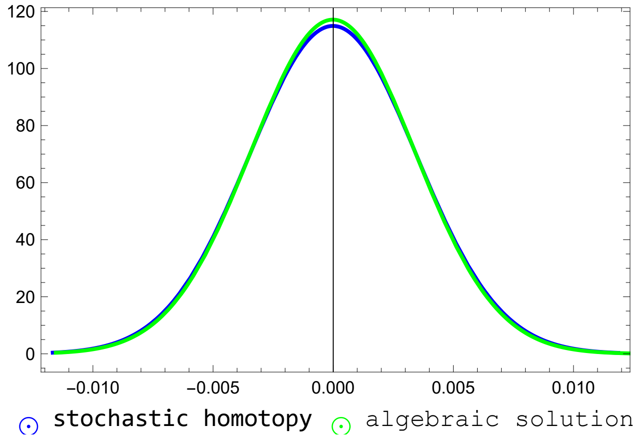
Then we fit a normal distribution,

```
In[∘]:= estimated𝒟 = FindDistribution[dataH,
   TargetFunctions → {NormalDistribution}, PerformanceGoal → "Quality"]
Out[∘]=
NormalDistribution[-9.82763 × 10^-6, 0.00347176]
```

Its density function

```
In[∘]:= p00 = Plot[PDF[estimated𝒟, x], {x, Min[dataH], Max[dataH]},
    PlotStyle → {Blue, Thick}, PlotLegends → Placed[" ⊙ stochastic homotopy", Below]];
```

In[∘]:=
```
Show[{p00, p11}, Frame → True]
```
Out[∘]=



The density functions resulted by the two different methods

The techniques can be applied to the other two ($y0$, $z0$) coordinates.

Case of Non - Gaussian Noise

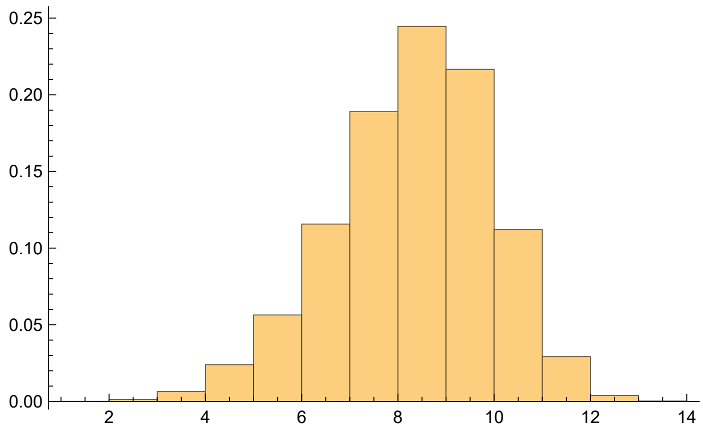
Much more realistic situation, when we measure the parameter and create a histogram of these values which may represents Non-Gaussian Noise!

Generating “Measured Data”

In[∘]:=
```
𝒟ℳ = WeibullDistribution[6, 9]
```
Out[∘]=
```
WeibullDistribution[6, 9]
```

In[∘]:=
```
data = RandomVariate[𝒟ℳ, 10 000];
```

In[∘]:=
```
Histogram[data, Automatic, "ProbabilityDensity"]
```
Out[∘]=



Measured Data of the parameter

Approximating the “measured data”

In[◦]:=
```
δW = FindDistribution[data, PerformanceGoal → "Quality"]
```
Out[◦]=
```
WeibullDistribution[5.66163, 9.07302]
```

In[◦]:=
```
Mean[data]
```
Out[◦]=
```
8.32007
```

In[◦]:=
```
StandardDeviation[data]
```
Out[◦]=
```
1.62266
```

Algebraic Method

Now δ is the parameter distribution

In[◦]:=
```
q = x^2 + δ x - 9;
```

In[◦]:=
```
sol = Solve[q == 0, x]
```
Out[◦]=

$$\left\{\left\{x \to \frac{1}{2}\left(-\delta - \sqrt{36 + \delta^2}\right)\right\}, \left\{x \to \frac{1}{2}\left(-\delta + \sqrt{36 + \delta^2}\right)\right\}\right\}$$

Let us consider the second solution

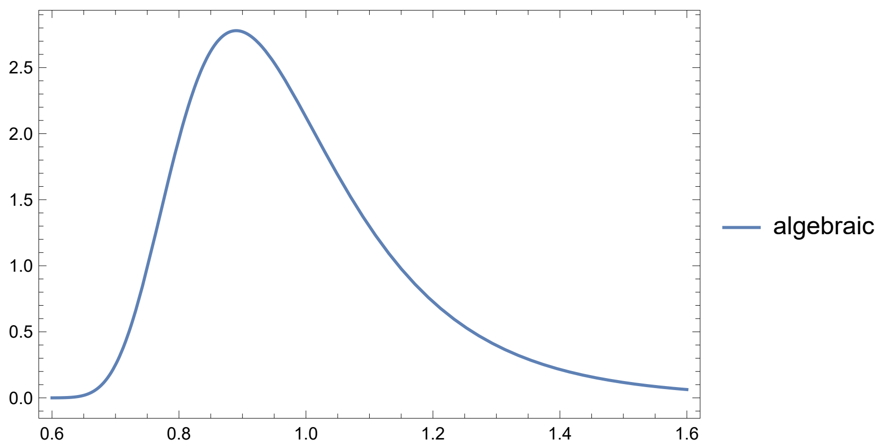
In[◦]:=
```
s = x /. First[sol〚2〛]
```
Out[◦]=

$$\frac{1}{2}\left(-\delta + \sqrt{36 + \delta^2}\right)$$

Then the distribution of this solution using distribution transform,

In[◦]:=
```
𝒟 = TransformedDistribution[s, δ ≈ δW];
```

In[◦]:=
```
p0 = Plot[PDF[𝒟, u], {u, 0.6, 1.6}, PlotLegends → {"algebraic"}, Frame → True]
```
Out[◦]=



Density function of the 2nd solution

It is clear that the result is a non-Gaussian.

In[◦]:=
```
data𝒟 = RandomVariate[𝒟, 10000];
```

```
In[•]:= Df = FindDistribution[data𝒟]
Out[•]=
ExtremeValueDistribution[0.894464, 0.135274]
```

```
In[•]:= μW = Mean[Df]
Out[•]=
0.972547
```

```
In[•]:= σW = StandardDeviation[Df]
Out[•]=
0.173496
```

Stochastic Homotopy Method

Since the Ito-form generates Gaussian trajectories, let us approximate the “measured data” via Gaussian Mixture!

```
In[•]:= δT = FindDistribution[data, TargetFunctions → {NormalDistribution}]
Out[•]=
MixtureDistribution[{0.687603, 0.312397},
 {NormalDistribution[7.81487, 1.72845], NormalDistribution[9.33793, 0.98187]}]
```

```
In[•]:= Clear[ρ1, ρ2]
```

```
In[•]:= w1 = δT[[1, 1]]
Out[•]=
0.687603
```

```
In[•]:= w2 = δT[[1, 2]]
Out[•]=
0.312397
```

```
In[•]:= μ1 = Mean[δT[[2, 1]]]
Out[•]=
7.81487
```

```
In[•]:= μ2 = Mean[δT[[2, 2]]]
Out[•]=
9.33793
```

The target system,

```
In[•]:= q = x^2 + (w1 (μ1 + ρ1) + w2 (μ2 + ρ2)) x - 9;
```

where $\rho 1 = \mathcal{N}(0, \sigma 1)$ and $\rho 2 = \mathcal{N}(0, \sigma 2)$

The start system,

```
In[•]:= p = x^2 - 9;
```

The linear homotopy function is

```
In[•]:= H = (1 - λ) p + λ q;
```

To get the differential equation form, we compute the partial derivates of the homotopy function,

In[•]:=
```
dHx = D[H, x]
```
Out[•]=

$$2\,x\,(1-\lambda)+\lambda\,(2\,x+0.687603\,(7.81487+\rho 1)+0.312397\,(9.33793+\rho 2))$$

and

In[•]:=
```
dHλ = D[H, λ]
```
Out[•]=

$$x\,(0.687603\,(7.81487+\rho 1)+0.312397\,(9.33793+\rho 2))$$

Then the right hand side of differential equation is

In[•]:=
```
d = - dHλ/dHx
```
Out[•]=

$$-\frac{x\,(0.687603\,(7.81487+\rho 1)+0.312397\,(9.33793+\rho 2))}{2\,x\,(1-\lambda)+\lambda\,(2\,x+0.687603\,(7.81487+\rho 1)+0.312397\,(9.33793+\rho 2))}$$

This is a nonlinear function of $\rho 1$ and $\rho 2$, in order to get a linear form, we use Taylor series at $\rho 1=0$ and $\rho 2 = 0$

Stochastic form of the homotopy differential equation

In[•]:=
```
Clear[GG, ρ1, ρ2, x, λ]
```
In[•]:=
```
GG[u_, v_] := d /. {ρ1 → u, ρ2 → v}
```
In[•]:=
```
GL = TaylorPolynomial[GG[ρ1, ρ2], {ρ1, ρ2}, {0, 0}, 1]
```
Out[•]=

$$-\frac{4.14533\,x}{x+4.14533\,\lambda}-\frac{0.343802\,x^2\,(1.\,\rho 1+0.454327\,\rho 2)}{(x+4.14533\,\lambda)^2}$$

Then the coefficient of the Ito differential equation

In[•]:=
```
pw1 = Coefficient[GL, ρ1]
```
Out[•]=

$$-\frac{0.343802\,x^2}{(x+4.14533\,\lambda)^2}$$

In[•]:=
```
pw2 = Coefficient[GL, ρ2]
```
Out[•]=

$$-\frac{0.156198\,x^2}{(x+4.14533\,\lambda)^2}$$

In[•]:=
```
pλ = GL[[1]]
```
Out[•]=

$$-\frac{4.14533\,x}{x+4.14533\,\lambda}$$

Now we have the linearized form of the stochastic differential equation

$$\frac{dx(\lambda)}{d\lambda} = p_\lambda + p_{w1}\,\rho 1 + p_{w2}\,\rho 2$$

Ito process

The integral form of this equation is

$$dx(\lambda) = \int p_\lambda \, d\lambda + \int p_{w1} \, \rho 1 \, d\lambda + \int p_{w2} \, \rho 2 \, d\lambda$$

Since the Gaussian noise is derivative of the Wiener process, namely

$$\frac{dW_i(\lambda)}{d\lambda} = \rho i \quad i = 1, 2$$

Then the Ito process of our stochastic differential equation

$$dx(\lambda) = \int p_\lambda \, d\lambda + \int p_{w1} \, dW_1 + \int p_{w2} \, dW_2$$

then let us assign the values for $\rho 1$ and $\rho 2$

In[∘]:=
```
ρ1 = StandardDeviation[δT[[2, 1]]]
```
Out[∘]=
```
1.72845
```

In[∘]:=
```
ρ2 = StandardDeviation[δT[[2, 2]]]
```
Out[∘]=
```
0.98187
```

In[∘]:=
```
proc = ItoProcess[dx[λ] == pλ dλ + pw1 dw1[λ] + pw2 dw2[λ], x[λ],
  {x, 3}, λ, {w1 ≈ WienerProcess[0, ρ1], w2 ≈ WienerProcess[0, ρ2]}]
```
Out[∘]=

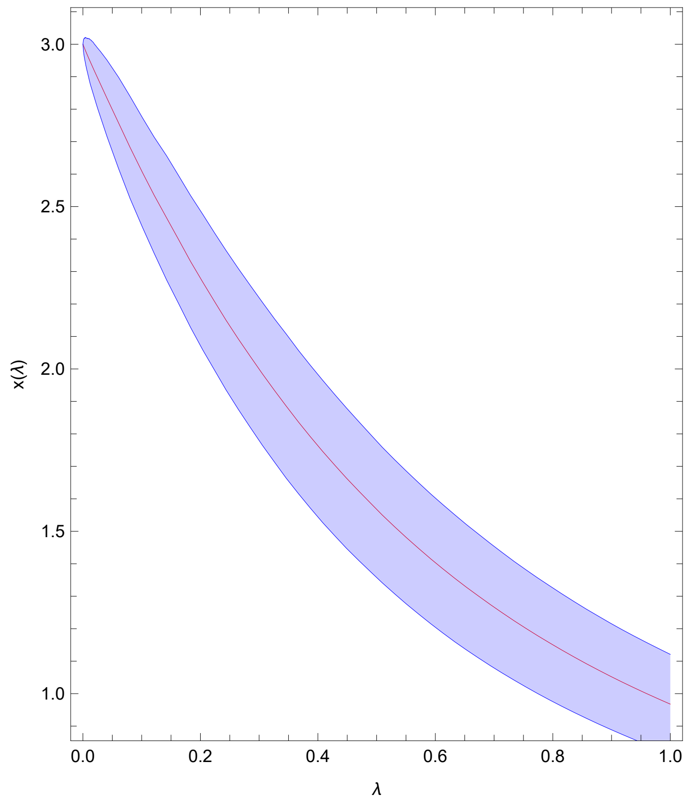
$$\text{ItoProcess}\left[\left\{\left\{0. - \frac{4.14533\, x[\lambda]}{4.14533\, \lambda + x[\lambda]}\right\}, \left\{\left\{0. - \frac{0.594244\, x[\lambda]^2}{(4.14533\, \lambda + x[\lambda])^2}, 0. - \frac{0.153366\, x[\lambda]^2}{(4.14533\, \lambda + x[\lambda])^2}\right\}\right\}, x[\lambda]\right\}, \{\{x\}, \{3\}\}, \{\lambda, 0\}\right]$$

In[∘]:=
```
psol100 = RandomFunction[proc, {0, 1.0, 0.001}, 2000, Method → "KloedenPlatenSchurz"]
```
Out[∘]=
```
TemporalData[ Time: 0. to 1.  Data points: 2 002 000  Paths: 2000 ]
Data not in notebook. Store now
```

In[∘]:=
```
p1 = Plot[Mean[psol100[λ]], {λ, 0, 1},
   PlotStyle → {Red, Thin}, FrameLabel → {"λ", "x(λ)"}, Frame → True];
```

In[∘]:=
```
p2 = Plot[{Mean[psol100[λ]] + StandardDeviation[psol100[λ]],
    Mean[psol100[λ]] - StandardDeviation[psol100[λ]]}, {λ, 0, 1}, Filling → {1 → {2}},
   PlotStyle → {{Blue, Thin}, {Blue, Thin}}, FrameLabel → {"λ", "x(λ)"}, Frame → True];
```

In[ ]:=
```
Show[{p1, p2}, AspectRatio → 1.2]
```
Out[ ]=



The trajectories of the Ito-solution - mean values and the standard deviations

The mean value of the solution is

In[ ]:=
```
m = Mean[psol100[1]]
```
Out[ ]=
```
0.967721
```

and the standard deviation

In[ ]:=
```
s = StandardDeviation[psol100[1]]
```
Out[ ]=
```
0.153295
```

The distribution of the solution

In[ ]:=
```
data100 = RandomVariate[psol100[1], 10 000];
```

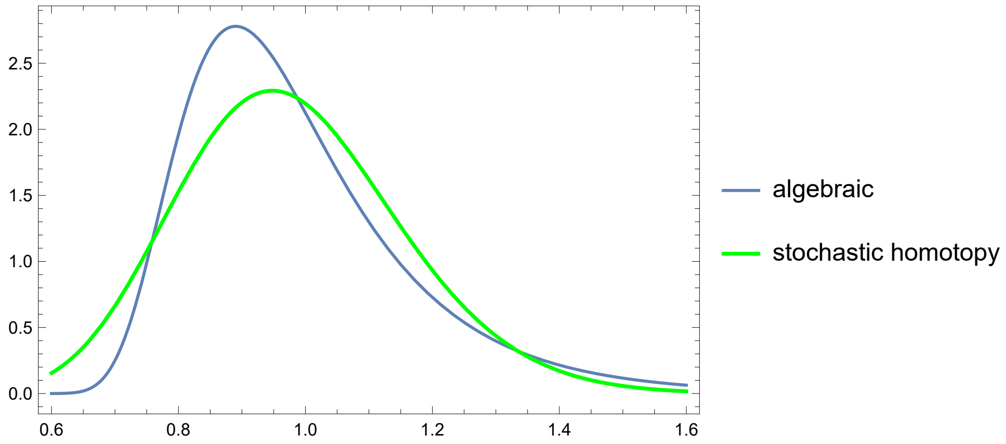
In[ ]:=
```
δT100 = FindDistribution[data100]
```
Out[ ]=
```
GammaDistribution[30.8302, 0.0317749]
```

The PDF of the solutions

In[ ]:=
```
p1 = Plot[PDF[δT100, u], {u, 0.6, 1.6},
   PlotStyle → {Green, Thick}, PlotLegends → {"stochastic homotopy"}];
```

In[∘]:= Show[{p0, p1}]

Out[∘]=



Fa-cit

The PDF transform technique provides a more practical and reliable solution!

Promotion

https://www.amazon.de/-/en/Joseph-L-Awange/dp/3030924947