Example about the importance of good sampling

Article

Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation

February 2001 · Journal of Robotic Systems 18(2)
DOI:10.1002/1097-4563(200102)18:23.3.CO;2-F

Authors:

G. Calafiore
Politecnico di Torino

Marina Indri
Politecnico di Torino

B Bona

Download citation | Copy link

Citations (116) References (25)

Abstract

Advanced robot control schemes require an accurate knowledge of the dynamic parameters of the manipulator. This article examines various issues related to robot dynamic calibration, from generation of optimal excitation trajectories to data acquisi-tion and filtering, and experimental inertial and friction parameter estimation. In particular, a new method is developed for the determination of optimal joint trajecto-ries for the calibration experiment, which is based on evolutionary optimization techniques. A genetic algorithm is used to determine excitation trajectories that minimize either the condition number of the regression matrix or the logarithmic determinant of the Fisher information matrix. All the calibration steps have been carried out on a SCARA two-link planar manipulator, and the experimental results are discussed. 2001 John Wiley & Sons, Inc.

Notice that the experiments (here trajectories) can be planned in advance as the design matrix only depends on the experiment design (and not the experimental outcome.

Consider a motorized mechanical system (e.g. a robot arm, a drone, or some other equipment) with kinetic energy

$$\frac{1}{2}\dot{\mathbf{q}} \cdot \mathbf{M}(\mathbf{p}_I, \mathbf{q}) \cdot \dot{\mathbf{q}}$$

and potential energy $V(\mathbf{p}_I, \mathbf{q})$ where $\mathbf{p}_I$ denote the inertial parameters of the system.

The equations of motion are then

$$\mathbf{M}(\mathbf{p}_I, \mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{p}_I, \mathbf{q}, \dot{\mathbf{q}}) + \nabla_{\mathbf{q}} V(\mathbf{p}_I, \mathbf{q}) = \boldsymbol{\tau} - \mathbf{f}(\mathbf{p}_F, \mathbf{q}, \dot{\mathbf{q}})$$

where $\boldsymbol{\tau}$ is the total motor torque; $\mathbf{f}(\mathbf{p}_F, \mathbf{q}, \dot{\mathbf{q}})$ is the parametrized friction model. The vector $\mathbf{C}(\mathbf{p}_I, \mathbf{q}, \dot{\mathbf{q}})$ contains the Coriolis and centrifugal terms.

In applications, $\mathbf{p}_I$ and $\mathbf{p}_F$ must typically be calibrated. To do this, a set of experiments measuring

$$\mathbf{q}^{(k)}, \dot{\mathbf{q}}^{(k)}, \ddot{\mathbf{q}}^{(k)}, \boldsymbol{\tau}^{(k)}; \quad k = 1, \ldots m$$

are measured.

Then the equations

$$\mathbf{M}(\mathbf{p}_I, \mathbf{q}^{(k)}) \cdot \ddot{\mathbf{q}}^{(k)} + \mathbf{C}(\mathbf{p}_I, \mathbf{q}^{(k)}, \dot{\mathbf{q}}^{(k)}) + \nabla_{\mathbf{q}}^{(k)} V(\mathbf{p}_I, \mathbf{q}^{(k)}) - \boldsymbol{\tau}^{(k)} + \mathbf{f}(\mathbf{p}_F, \mathbf{q}^{(\mathbf{k})}, \dot{\mathbf{q}}^{(k)}) \simeq \mathbf{0}; \quad k = 1, \ldots, m$$

are solved with respect to $\mathbf{p}_I$ and $\mathbf{p}_F$ as a least squares problem.

However, $\mathbf{p}_I$ and $\mathbf{p}_F$ typically enters non-linearly in the equations. After the next two lectures, you will know how to handle this.

A nonlinear equation as defined here has the form

$$f(x) = 0 \tag{1}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a continuous function. Throughout these notes, we will call a (relevant) solution $\tilde{x}$.

An iterative method is initiated with a starting guess x_0. An iterative method generates a sequence $x_0, x_1, x_2, \ldots$ with some algorithm.

For a given problem and initial guess, the iterative method is called **convergent** if $x_i \to X$ for $i \to \infty$ where X is a fixed value (not necessarily a solution). If we also have $x_i \to \tilde{x}$ for $i \to \infty$, we say that the method is **convergent to a solution**.

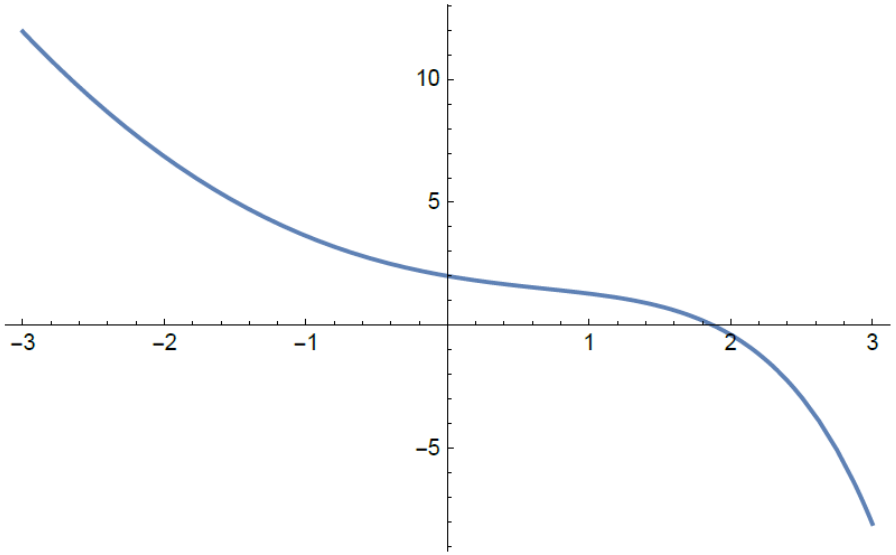
If a method is convergent (to a solution) for any problem and from any initial guess x_0, we say that the method is **globally convergent** (to a solution). Only the methods using bracketing are globally convergent to a solution.

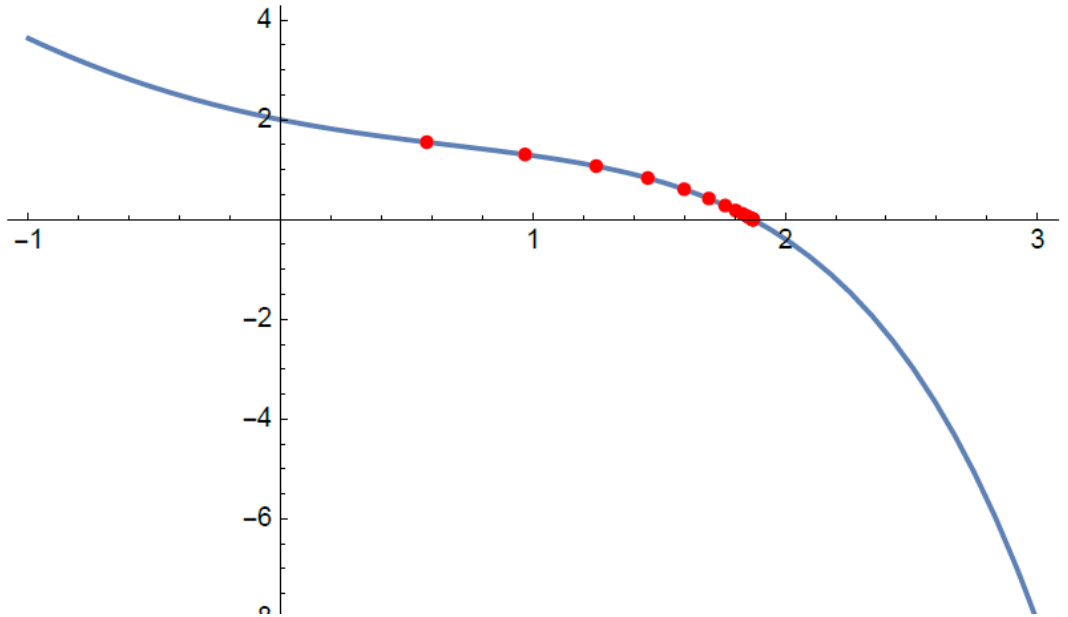
A simple example

An iterative method is initiated with a starting guess x_0. An iterative method generates a sequence $x_0, x_1, x_2, \ldots$ with some algorithm.

For a given problem and initial guess, the iterative method is called **convergent** if $x_i \to X$ for $i \to \infty$ where X is a fixed value (not necessarily a solution). If we also have $x_i \to \tilde{x}$ for $i \to \infty$, we say that the method is **convergent to a solution**.

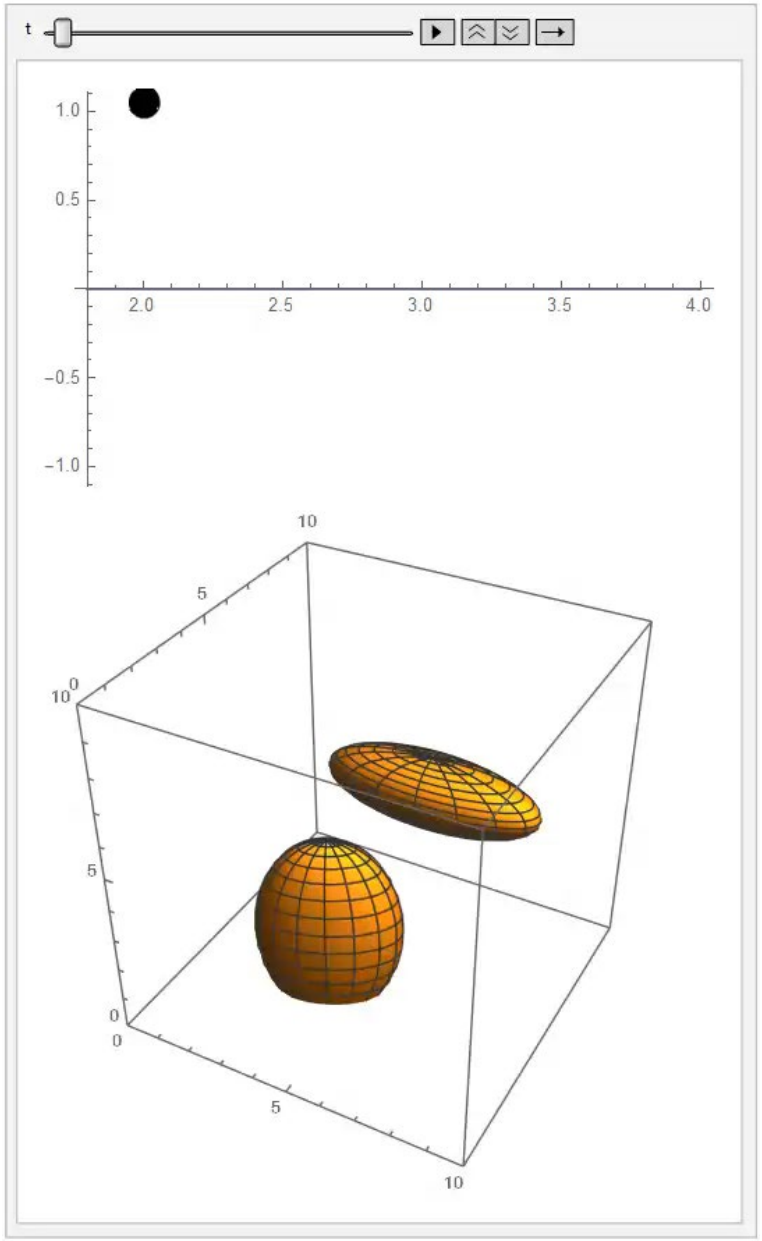
```
f[x_] = x^2 - Exp[x] + 3;

Plot[f[x], {x, -3, 3}]
```






```
exact = x /. FindRoot[f[x] == 0, {x, 0.5}]
```

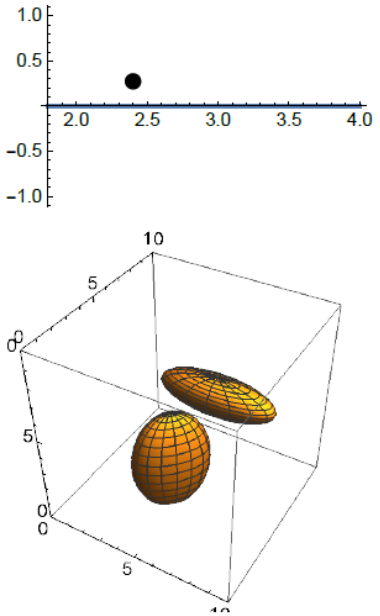
```
1.87312
```



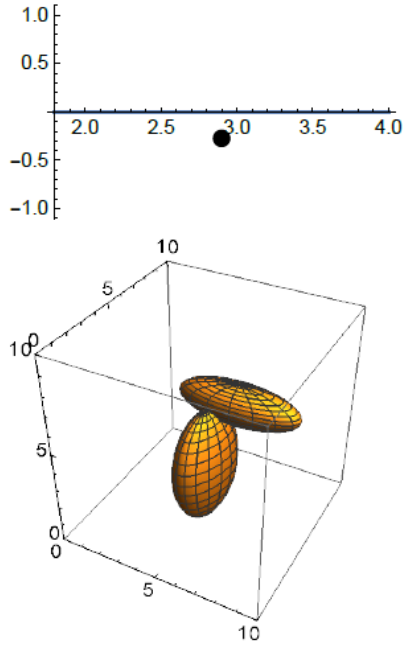
A more interesting example:

Computing time of collision for dynamic simulation.

Y-value of black dot indicates signed minimal distance between objects

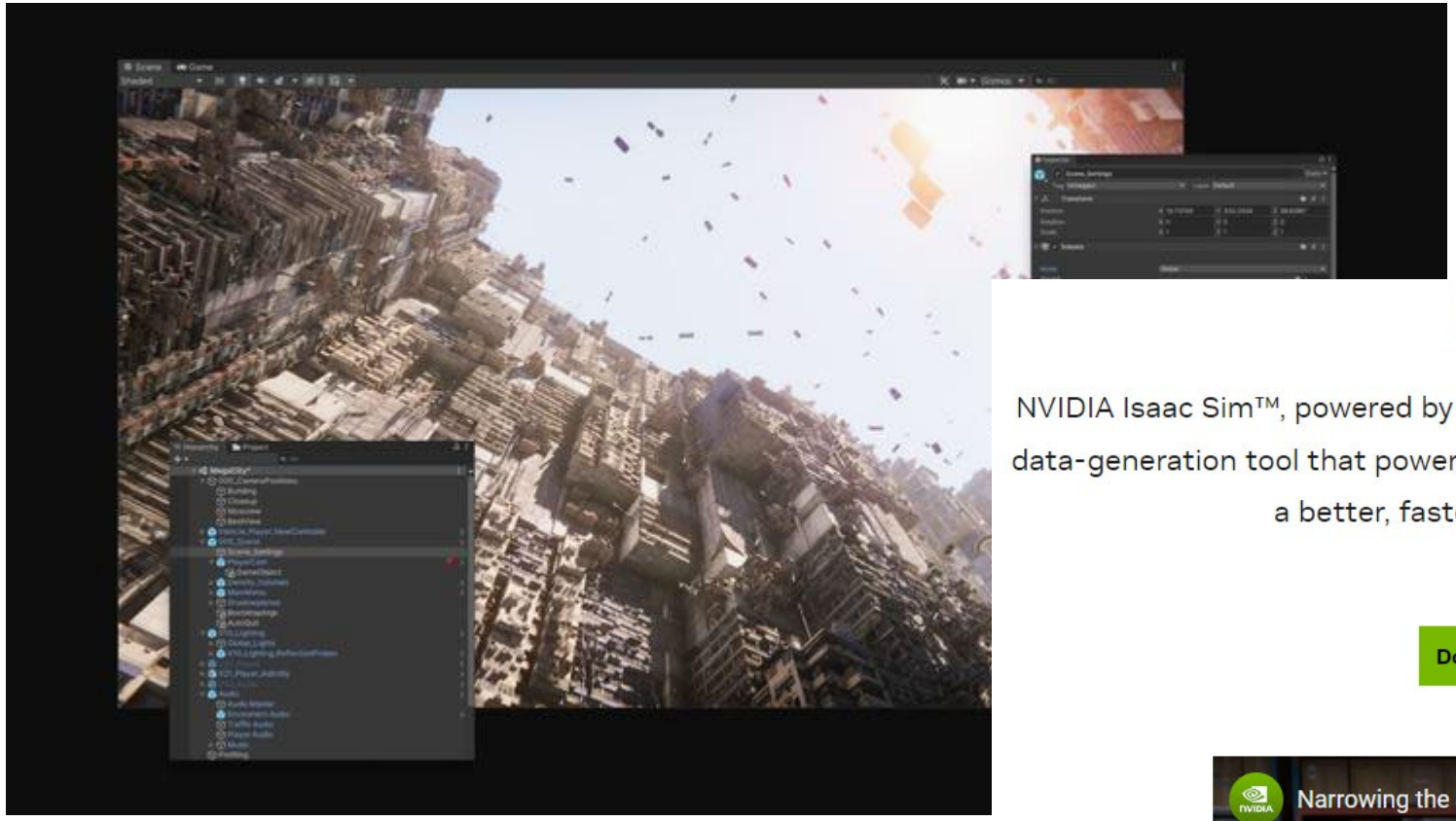


x=T



x=T+h

Computing f(x) for a single x. In this example f(x) is the signed distance at time x. Notice that f(x) is **computationally expensive** (imagine a scene with many objects and potential collisions)



Physics based game engines
Here: Unity Pro

NVIDIA Isaac Sim

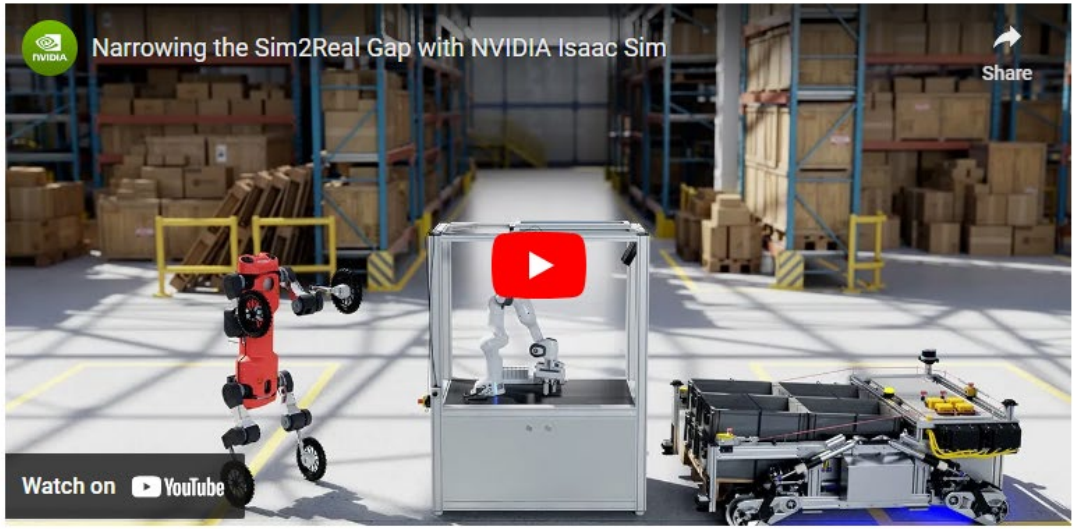
NVIDIA Isaac Sim™, powered by Omniverse™, is a scalable robotics simulation application and synthetic data-generation tool that powers photorealistic, physically accurate virtual environments. This gives you a better, faster way to develop, test, and manage AI-based robots.

Run Local

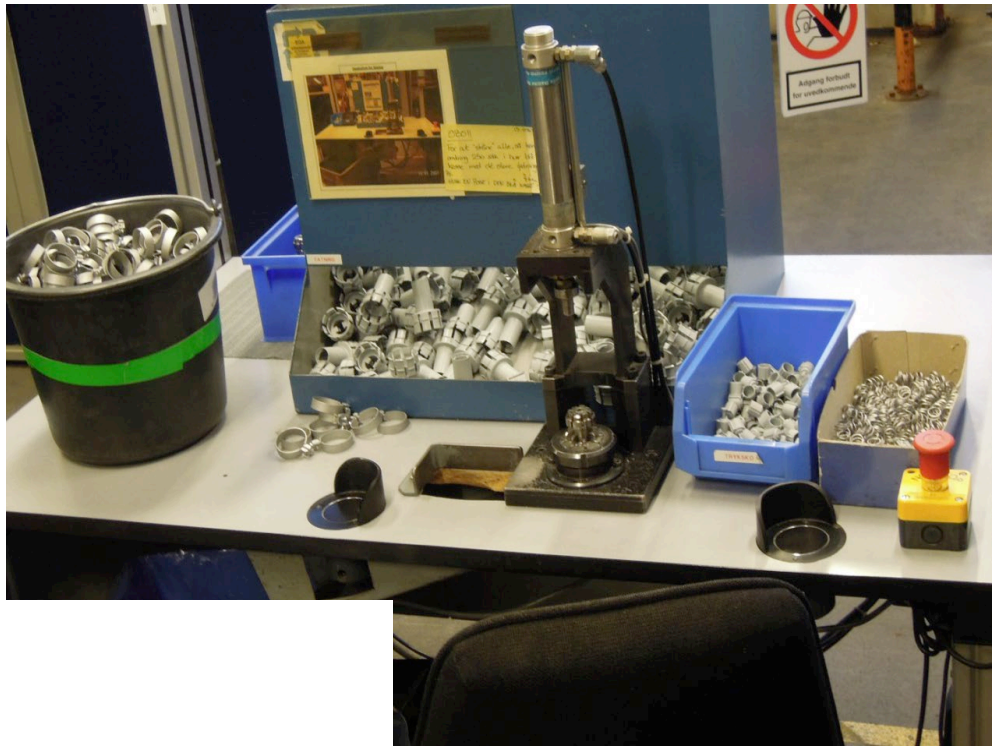
Download Omniverse

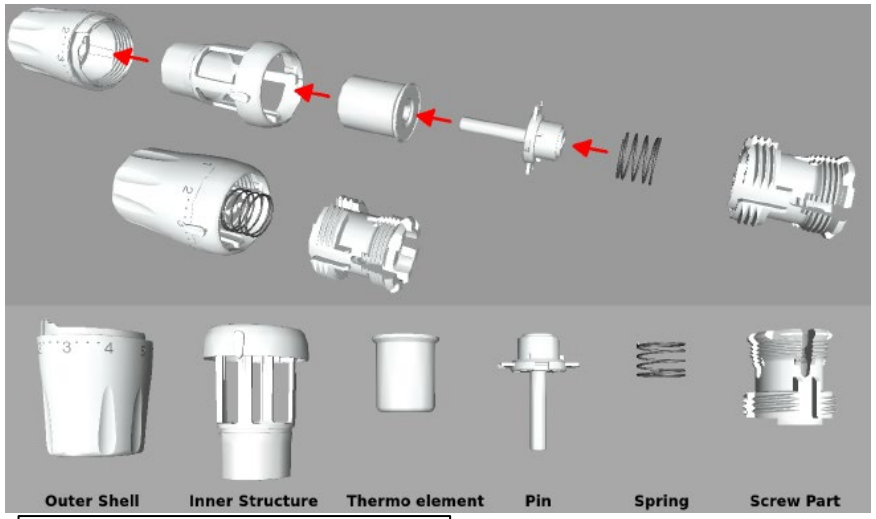
Run in the Cloud

Download Container

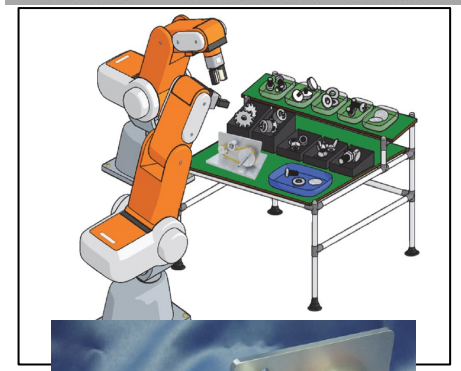


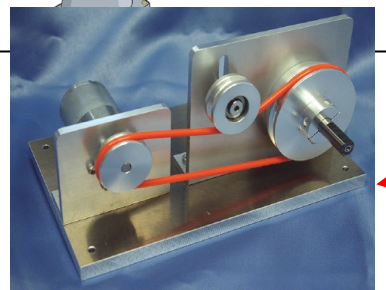
Example of usage for assembly simulation:
Finding time of collision event in dynamic simulations





Assembly of Novo insuline injection pen based on dynamic simulations





1.1 Important properties

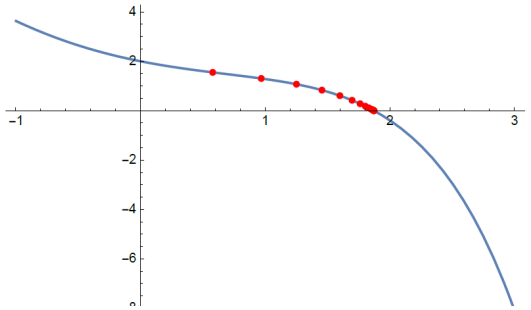
The goal of any method is to find an approximation to a solution with a desired accuracy (distance to the solution). When choosing a solution, the following properties should be considered

- **Robustness:** Is there a guaranteed convergence to the solution. Some methods are robust for all problems (when the method is globally convergent to a solution). Some method are robust for specific types of problems.
- **Efficiency:** Simply computed as the number of $f(x)$-computations needed to converge to the solution with the desired accuracy.

Convergence, order and convergence constant

If a method converges to a solution $\tilde{x}$, we write the error after k steps as

$$\epsilon_k = x_k - \tilde{x}$$



If then

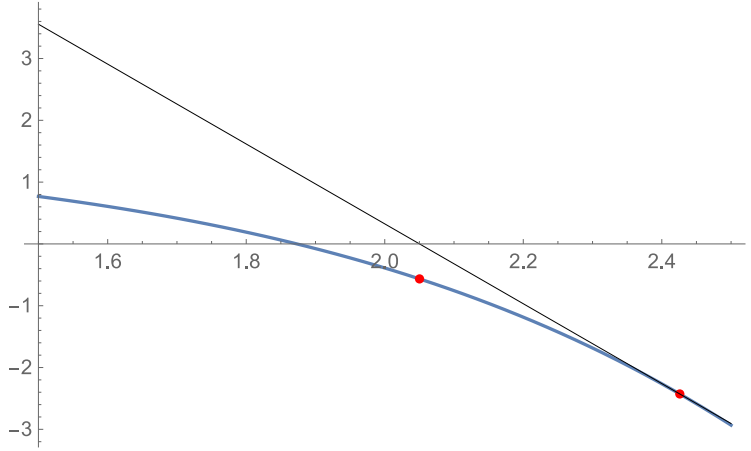
$$\frac{|\epsilon_{k+1}|}{|\epsilon_k|^\alpha} \to C \quad \text{when} \quad k \to \infty$$

we say that the convergence has **order** α with **convergence constant** C.

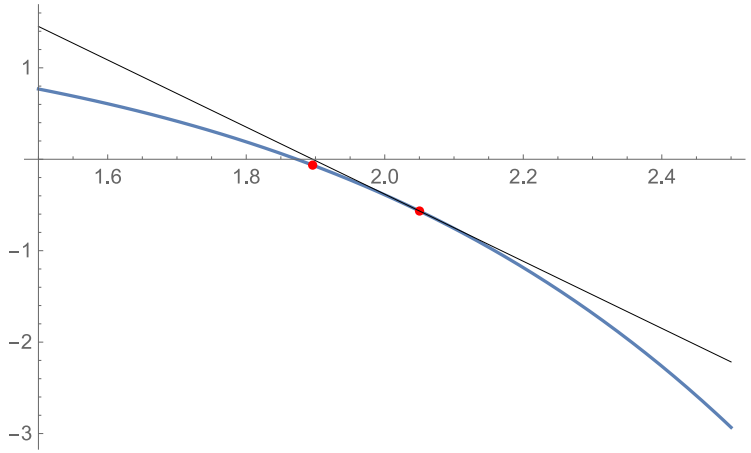
A given method has the same order for all root finding problems (except in singular cases), whereas the convergence constant is problem dependent.

Newtons method

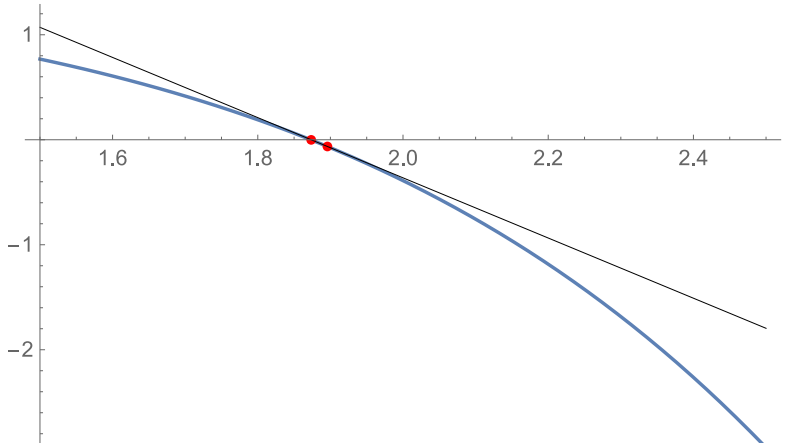
$$x_{i+1} = x_i - \frac{1}{f'(x_i)} f(x_i)$$



First iteration



Second iteration



Third iteration

1.1 Newtons method

The most well known iterative method for root finding is Newtons method (sometimes called the Newton-Raphson method). To use Newtons method, the function $f(x)$ must be continuous differentiable. The algorithm is then

$$x_{i+1} = x_i - \frac{1}{f'(x_i)} f(x_i)$$

1.1.1 Convergence properties

An analysis (which we shall skip) shows that if f is twice continuous differentiable, Newtons method has

$$\begin{aligned} \alpha &= 2 \\ C &= \frac{1}{2}\frac{f''(\tilde{x})}{f'(\tilde{x})} \end{aligned}$$

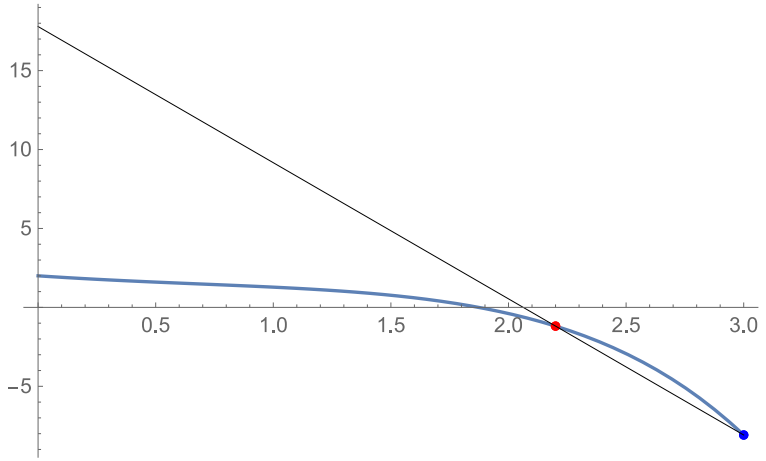
$$\epsilon_k = x_k - \tilde{x}$$

$$\frac{|\epsilon_{k+1}|}{|\epsilon_k|^\alpha} \to C \qquad \textbf{order } \alpha$$

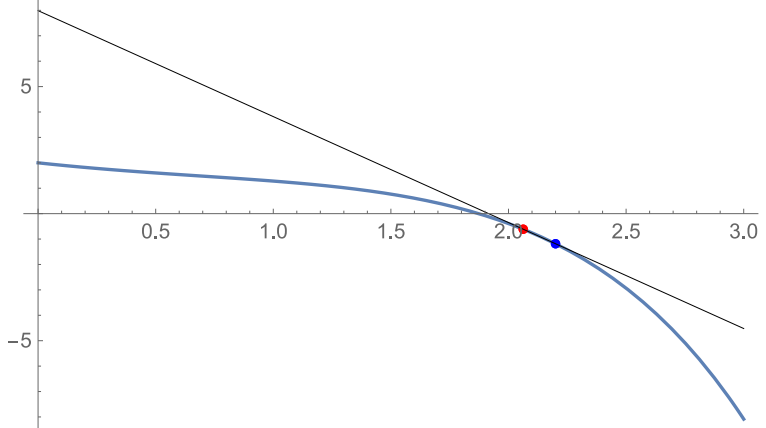
convergence constant C

Secant Method

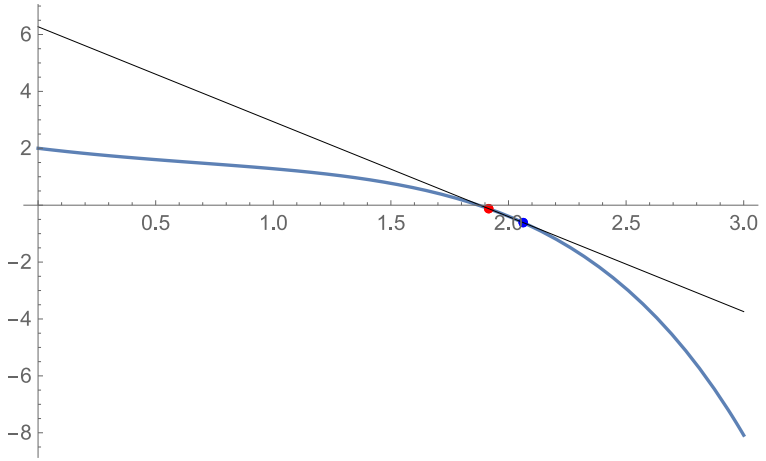
$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$



First iteration



Second iteration



Third iteration

1.2 The Secant Method

We may also use a finite difference version of Newtons method called the Secant method. We estimate

$$f'(x_i) \simeq \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Choose first a random $x_1 \neq x_0$. The sequence is then generated as

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

1.2.1 Convergence properties

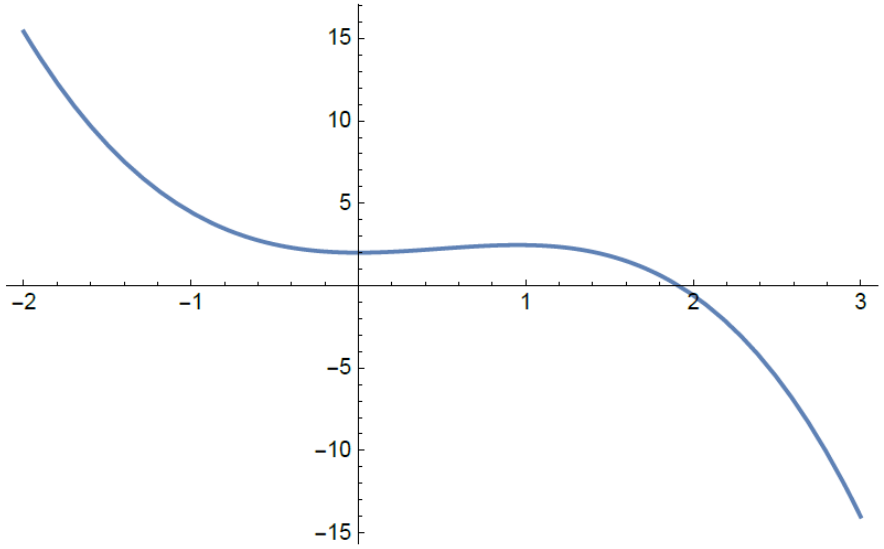
An analysis (which we shall again skip) shows that if f is twice continuous differentiable, the Secant method has

$$\begin{aligned} \alpha &= \frac{1}{2}(1+\sqrt{5}) \simeq 1.62 \\ C &= \left| \frac{1}{2} \frac{f''(\tilde{x}}{f'(\tilde{x})} \right|^{\frac{2}{1+\sqrt{5}}} \end{aligned}$$

$$\epsilon_k = x_k - \tilde{x}$$

$$\frac{|\epsilon_{k+1}|}{|\epsilon_k|^\alpha} \to C \qquad \textbf{order } \alpha$$

convergence constant C



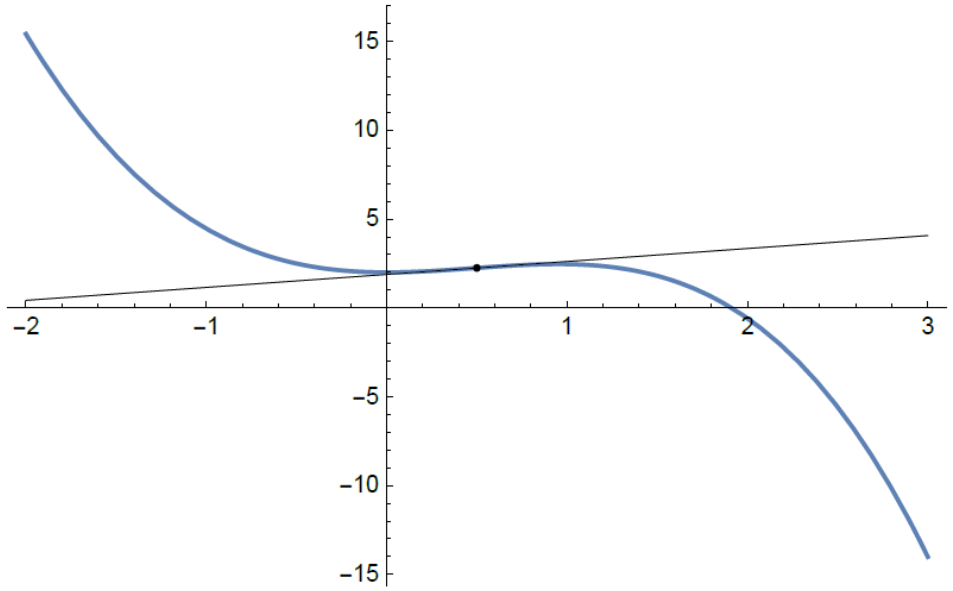
Convergence problems

- Will often occur for both Newton and Secant

```
exact = x //. FindRoot[f[x] == 0, {x, 0.5}]
```

FindRoot: The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances.

-6.02475×10^{-6}



2 Iterative methods using bracketing

We will define the bracketing of a solution as having x_0, y_0 so that $f(x_0) < 0$ and $f(y_0) > 0$ or vice versa. We will abbreviate this condition as $f(x_0)f(y_0) < 0$. With this condition, as f is assumed to be continuous, there will be a solution to $f(x) = 0$ between x_0 and y_0.

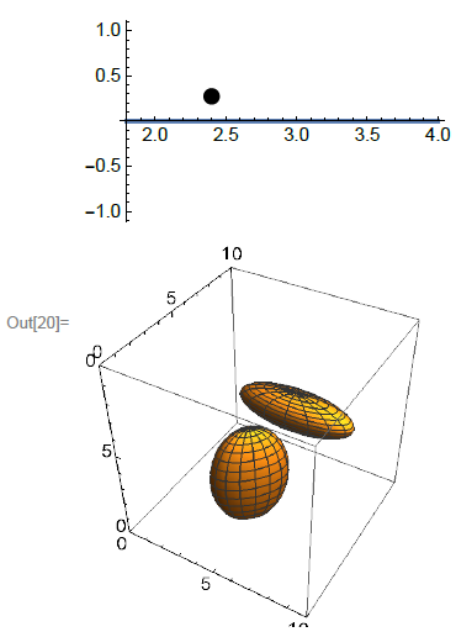
An iterative method using bracketing derives a sequence of point pairs (x_0, y_0); (x_1, y_1); (x_2, y_2); ...; ... so that

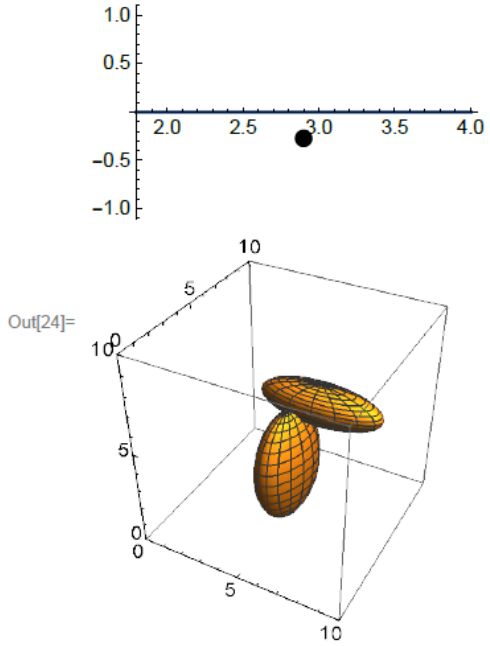
$$\begin{aligned} f(x_i)f(y_i) &< 0 \quad \text{for } i = 0, 1, 2, \ldots \\ x_i &\to \tilde{x} \quad \text{for } i \to \infty \end{aligned}$$

All methods using bracketing will obviously converge to a solution $\tilde{x}$ due to the two conditions above. In the methods below, we will still use the x_i's as our primary members for the sequence, whereas the y_i's are control points to keep the bracketing.

Bracketing is often directly offered by the application

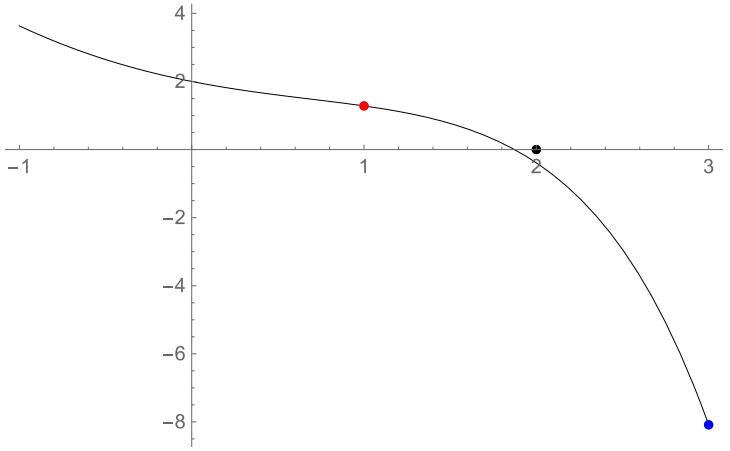
No collision (f>0) and collision (f<0)

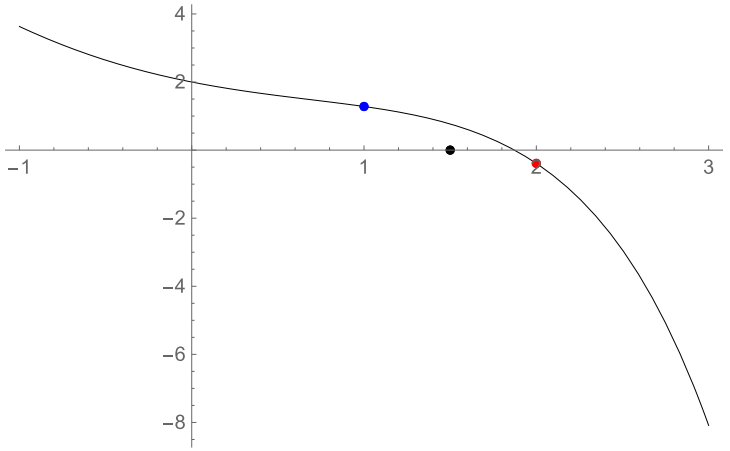


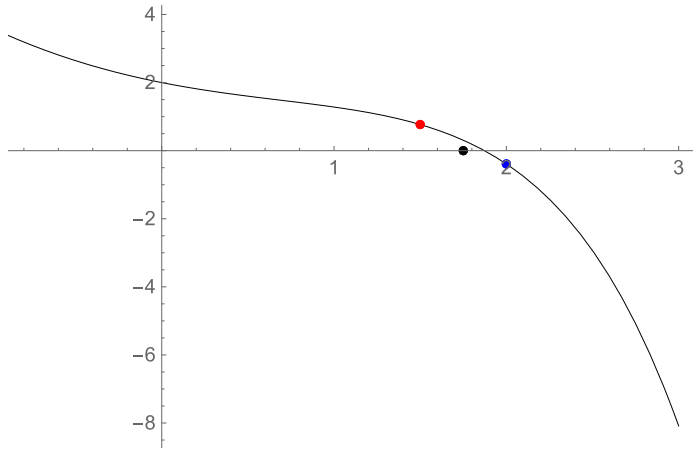


Bisection

First three iterations.
Red dots are x_i's
Blue dots are y_i's
Black dots are new midpoints







For $i = 0, 1, 2, \ldots$

$$x_{i+1} = (x_i + y_i)/2$$

If $f(x_{i+1})f(y_i) < 0$

$$y_{i+1} = y_i$$

Else

$$y_{i+1} = x_i$$

2.1.1 Convergence properties

Clearly, we get that the size of the interval is divided by two in each step. Hence, we get

$$\begin{aligned} \alpha &= 1 \\ \overline{C} &= \frac{1}{2} \end{aligned}$$

where $\overline{C}$ is the average value of C over all the iterations.

False Position (Regula Falsi)

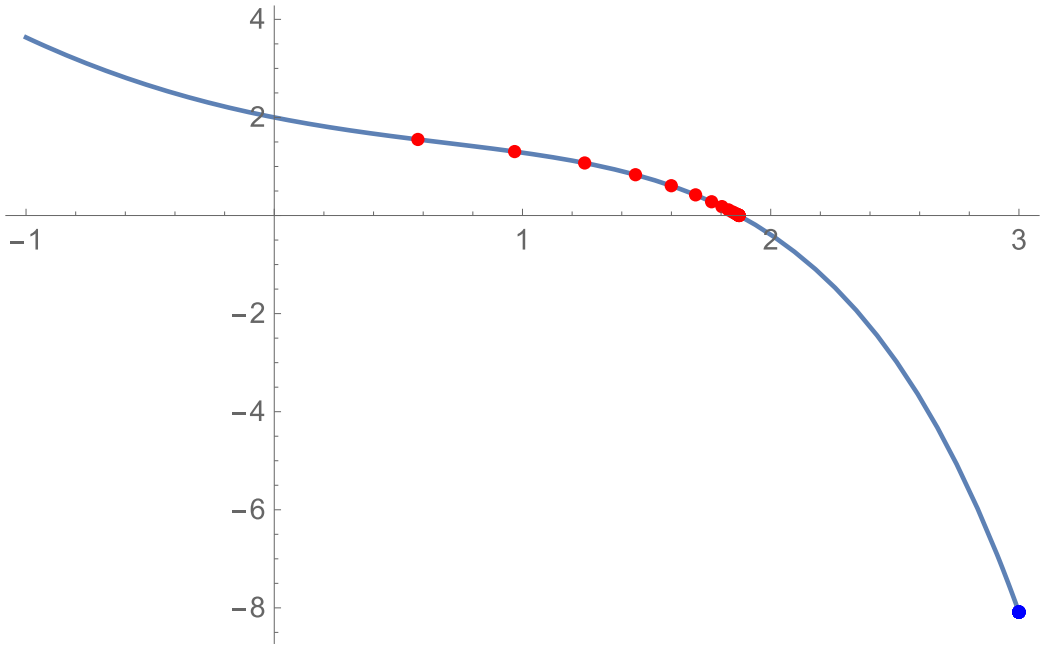
$$x_{i+1} = x_i - \frac{x_i - y_i}{f(x_i) - f(y_i)} f(x_i)$$

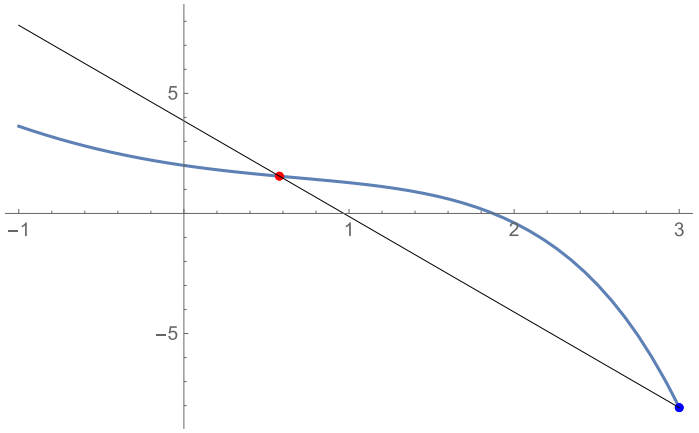
If $f(x_{i+1}) f(y_i) < 0$

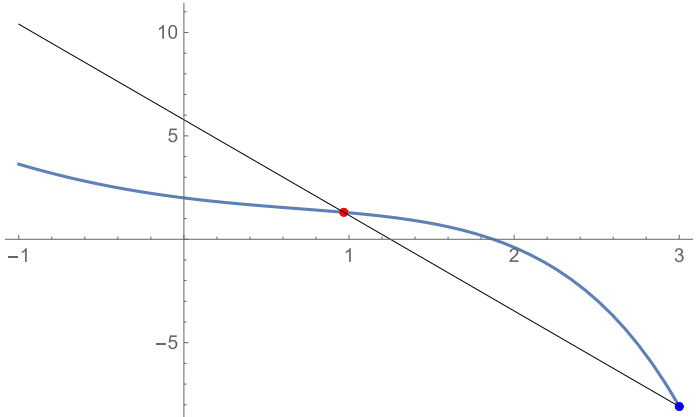
$$y_{i+1} = y_i$$

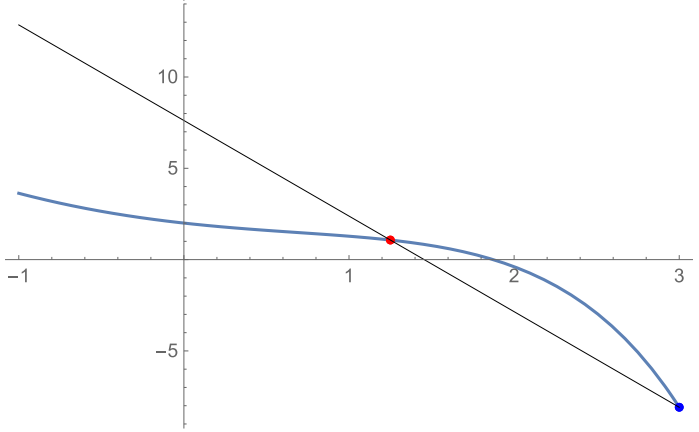
Else

$$y_{i+1} = x_i$$









First three iterations.
Red dots are x_i's
Blue dots are y_i's

2.2 False Position (Regula Falsi)

False position is a bracketing variant of the Secant method inspired by Bisection. The algorithm is

$$x_{i+1} = x_i - \frac{x_i - y_i}{f(x_i) - f(y_i)} f(x_i)$$

If $f(x_{i+1})f(y_i) < 0$

$$y_{i+1} = y_i$$

Else

$$y_{i+1} = x_i$$

2.2.1 Convergence properties

An analysis (which we shall again skip) shows that if f is twice continuous differentiable and $f''(\tilde{x}) \neq 0$, we will have that from some m that

$$y_m = y_{m+1} = y_{m+2} = \dots$$

We then get that the Regula falsi method has

$$\begin{aligned} \alpha &= 1 \\ C &= 1 - f'(\tilde{x}) \frac{y_m - \tilde{x}}{f(y_m)} \end{aligned}$$

$$\epsilon_k = x_k - \tilde{x}$$

$$\frac{|\epsilon_{k+1}|}{|\epsilon_k|^\alpha} \to C \qquad \textbf{order } \alpha$$

convergence constant C

Ridders method

$$z_i = (x_i + y_i)/2$$

$$x_{i+1} = z_i + (z_i - x_i)\frac{\mathrm{sign}[f(x_i) - f(y_i)]f(z_i)}{\sqrt{f(z_i)^2 - f(x_i)f(y_i)}}$$

If $f(x_{i+1})f(z_i) < 0$

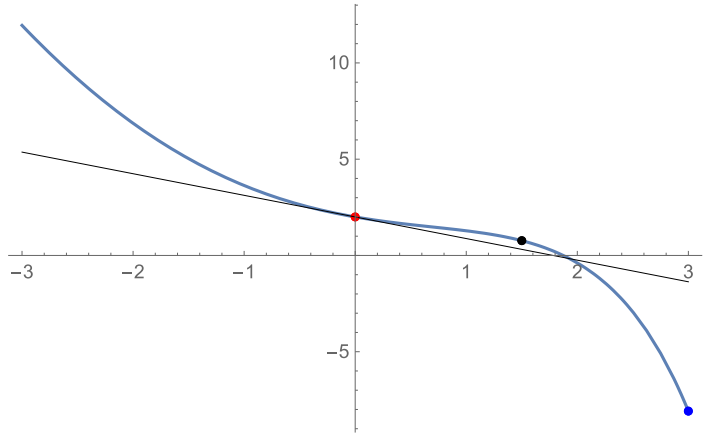
$$y_{i+1} = z_i$$

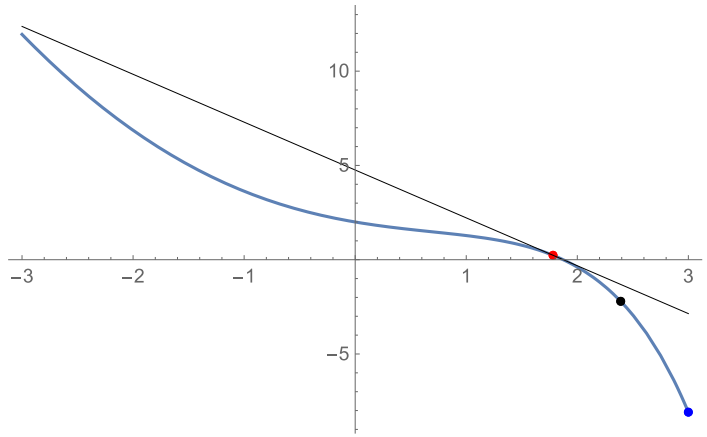
Else if $f(x_{i+1})f(y_i) < 0$

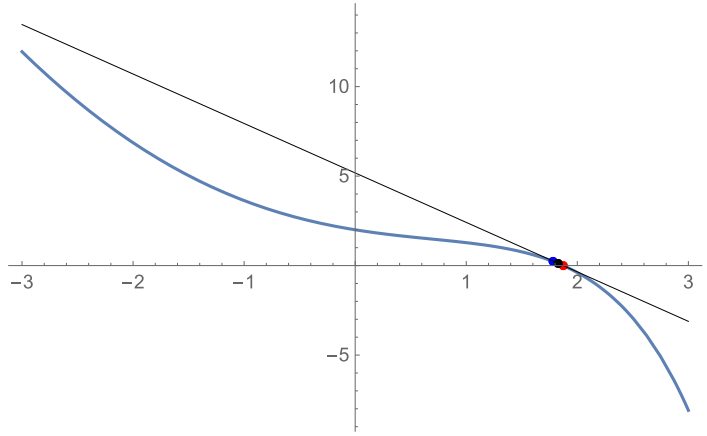
$$y_{i+1} = y_i$$

Else

$$y_{i+1} = x_i$$







First three iterations.
Red dots are x_i's
Blue dots are y_i's
Black dots are new midpoints
Zero crossing of line is next x_i

2.3.1 Convergence properties

An analysis by Ridders (which we shall again skip) shows that if f is 3 times continuous differentiable, we get

$$\begin{aligned} \alpha &= 3 \\ C &\simeq \left| \frac{f''(\tilde{x})^2 - 2f'(\tilde{x})f'''(\tilde{x})}{f'(\tilde{x})^2} \right| \end{aligned}$$

$$\epsilon_k = x_k - \tilde{x}$$

$$\frac{|\epsilon_{k+1}|}{|\epsilon_k|^\alpha} \to C \qquad \textbf{order } \alpha$$

convergence constant C

If $f'(\tilde{x}) \simeq 0$, the convergence will mostly be very slow, but Ridders method does always converge. And in the general case, the convergence of Ridders is extremely fast due to the third order nature.

Notice: There are two f-computations per iteration. Hence the “order per f-computation” is only Sqrt(3) or around 1.73.

Consider the simple problem

$$f(x)=x-\cos(x)=0$$

where there must be a solution X between 0 and Pi/2.

Implement Newton, Secant, Bisection, Regula Falsi and Ridder for this problem.

Each method produces a sequence x_0, x_1, x_2,… that is supposed to converge to the solution.

You will be asked to organize outputs of the method in a table where each line corresponds to one iteration. The table will contain the following info

i x_i x_i-x_{i-1} |x_i-x_(i-1)|/ |x_(i-1)-x_(i-2)|^k

where k is the convergence order of the method.