25 February 2025

Solutions 2 : General Properties of Nonlinear Systems

1 Picard Approach to the Harmonic Oscillator

We solve the harmonic oscillator equation $\ddot{x} + \omega^2 x = 0$ with initial conditions $x(0) = c_0$ and $x'(0) = \omega s_0$.

(a) Rewrite the harmonic oscillator equation in the $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t))$ form and analytically compute the first two Picard iterations $\vec{x}_1(t)$ and $\vec{x}_2(t)$. Do you see where the iterations are going ? Otherwise continue with $\vec{x}_3(t)$ and so on to guess $\lim_{k\to\infty} \vec{x}_k(t)$.

Our second order system is $(x^{(1)}, x^{(2)}) = (x, x')$. So $f(x^{(1)}, x^{(2)}) = (x^{(2)}, -\omega^2 x^{(1)})$
The first iterations give :

$$\vec{x}_1(t) = \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \int_0^t \begin{bmatrix} \omega s_0 \\ -\omega^2 c_0 \end{bmatrix} \mathrm{d}s = \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \begin{bmatrix} \omega s_0 \\ -\omega^2 c_0 \end{bmatrix} t$$

$$\vec{x}_2(t) = \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \int_0^t \vec{f}\left(\begin{bmatrix} c_0 + s_0 \omega x \\ \omega(-c_0 \omega x + s_0) \end{bmatrix} \right) \mathrm{d}x = \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \int_0^t \begin{bmatrix} \omega(-c_0 \omega x + s_0) \\ -c_0 \omega^2 - s_0 \omega^3 x \end{bmatrix} \mathrm{d}x$$

$$= \begin{bmatrix} c_0(1 - \frac{\omega^2}{2} t^2) + s_0 \omega t \\ \omega[-c_0(\omega t) + s_0(1 - \frac{\omega^2}{2} t^2)] \end{bmatrix}$$

$$\vec{x}_3(t) = \begin{bmatrix} c_0(1 - \frac{\omega^2}{2} t^2) + s_0(\omega t - \frac{\omega^3}{6} t^3) \\ \omega[-c_0(\omega t - \frac{\omega^3}{6} t^3) + s_0(1 - \frac{\omega^2}{2} t^2)] \end{bmatrix}$$

We can see the first order of cosine and sine forming, with the highest order term proportional to t^k. We can guess the solution will be $x^{(1)}(t) = x(t) = c_0 \cos(\omega t) + s_0 \sin(\omega t)$, with its derivative $x^{(2)}(t) = x'(t) = \omega[-c_0 \sin(\omega t) + s_0 \cos(\omega t)]$. This is our good friend, the solution of the harmonic oscillator.

(b) Prove rigorously that Picard's iterations converge to the result you guessed in (a).

Reminder : For a system $\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t))$ Picard's method states that $\vec{x}_0(t) = \vec{x}(t_0)$ and $\vec{x}_{k+1}(t) = \vec{x}(t_0) + \int_{t_0}^t \vec{f}(s, \vec{x}_k(s))\mathrm{d}s$

To prove it we can proceed by induction. The general form is the partial sum of the sine and cosine series. Using the first terms that we computed, the assumption is that :

$$\vec{x}_k(t) = \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} \sum_{n=0}^{\alpha(k)} (-1)^n \frac{(\omega t)^{2n}}{(2n)!} + \begin{bmatrix} s_0 \\ -\omega c_0 \end{bmatrix} \sum_{n=0}^{\beta(k)} (-1)^n \frac{(\omega t)^{2n+1}}{(2n+1)!} \tag{1}$$

The upper bound of the partial sums for the cosine and sine series are such that the highest order term is $\propto t^k$. This means that we have to differentiate the even and odd cases.

$$\alpha(k) = \begin{cases} \frac{k-1}{2} & \text{if } k \text{ is odd} \\ \frac{k}{2} & \text{if } k \text{ is even} \end{cases} \qquad \beta(k) = \begin{cases} \frac{k-1}{2} & \text{if } k \text{ is odd} \\ \frac{k}{2} - 1 & \text{if } k \text{ is even} \end{cases}$$

This is verified by $\vec{x}_0(t)$ and $\vec{x}_1(t)$, which initializes our induction. (Note that both $k = 0$ and $k = 1$ must be verified since the form in Eq. (1) is different for even and odd k.) To prove heredity, the integration of the sum is straightforward, then some manipulations are necessary.

$$\begin{aligned}
\vec{x}_{k+1}(t) &= \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \int_0^t \left\{ \begin{bmatrix} \omega s_0 \\ -\omega^2 c_0 \end{bmatrix} \sum_{n=0}^{\alpha(k)} (-1)^n \frac{(\omega s)^{2n}}{(2n)!} + \begin{bmatrix} -\omega c_0 \\ -\omega^2 s_0 \end{bmatrix} \sum_{n=0}^{\beta(k)} (-1)^n \frac{(\omega s)^{2n+1}}{(2n+1)!} \right\} \mathrm{d}s \\
&= \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \begin{bmatrix} \omega s_0 \\ -\omega^2 c_0 \end{bmatrix} \sum_{n=0}^{\alpha(k)} (-1)^n \frac{\omega^{2n} t^{2n+1}}{(2n+1)!} + \begin{bmatrix} -\omega c_0 \\ -\omega^2 s_0 \end{bmatrix} \sum_{n=0}^{\beta(k)} (-1)^n \frac{\omega^{2n+1} t^{2n+2}}{(2n+2)!} \\
&= \begin{bmatrix} s_0 \\ -\omega c_0 \end{bmatrix} \sum_{n=0}^{\alpha(k)} (-1)^n \frac{(\omega t)^{2n+1}}{(2n+1)!} + \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} + \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} \sum_{n=0}^{\beta(k)} (-1)^{n+1} \frac{(\omega t)^{2n+2}}{(2n+2)!} \\
&= \begin{bmatrix} s_0 \\ -\omega c_0 \end{bmatrix} \sum_{n=0}^{\alpha(k)} (-1)^n \frac{(\omega t)^{2n+1}}{(2n+1)!} + \begin{bmatrix} c_0 \\ \omega s_0 \end{bmatrix} \sum_{m=0}^{\beta(k)+1} (-1)^m \frac{(\omega t)^{2m}}{(2m)!}
\end{aligned}$$

In the last step, the second sum is shifted, $m = n + 1$. For the heredity to be correct, we need $\alpha(k) = \beta(k+1)$ and $\beta(k) + 1 = \alpha(k+1)$. If k is even :

$$\begin{aligned}
\alpha(k) &= \frac{k}{2} = \frac{(k+1)-1}{2} = \beta(k+1) \\
\beta(k) + 1 &= \frac{k}{2} - 1 + 1 = \frac{(k+1)-1}{2} = \alpha(k+1)
\end{aligned}$$

If k is odd :

$$\begin{aligned}
\alpha(k) &= \frac{k-1}{2} = \frac{(k+1)-2}{2} = \frac{k+1}{2} - 1 = \beta(k+1) \\
\beta(k) + 1 &= \frac{k-1}{2} + 1 = \frac{k+1}{2} = \alpha(k+1)
\end{aligned}$$

So indeed the heredity is valid and our general form is true.

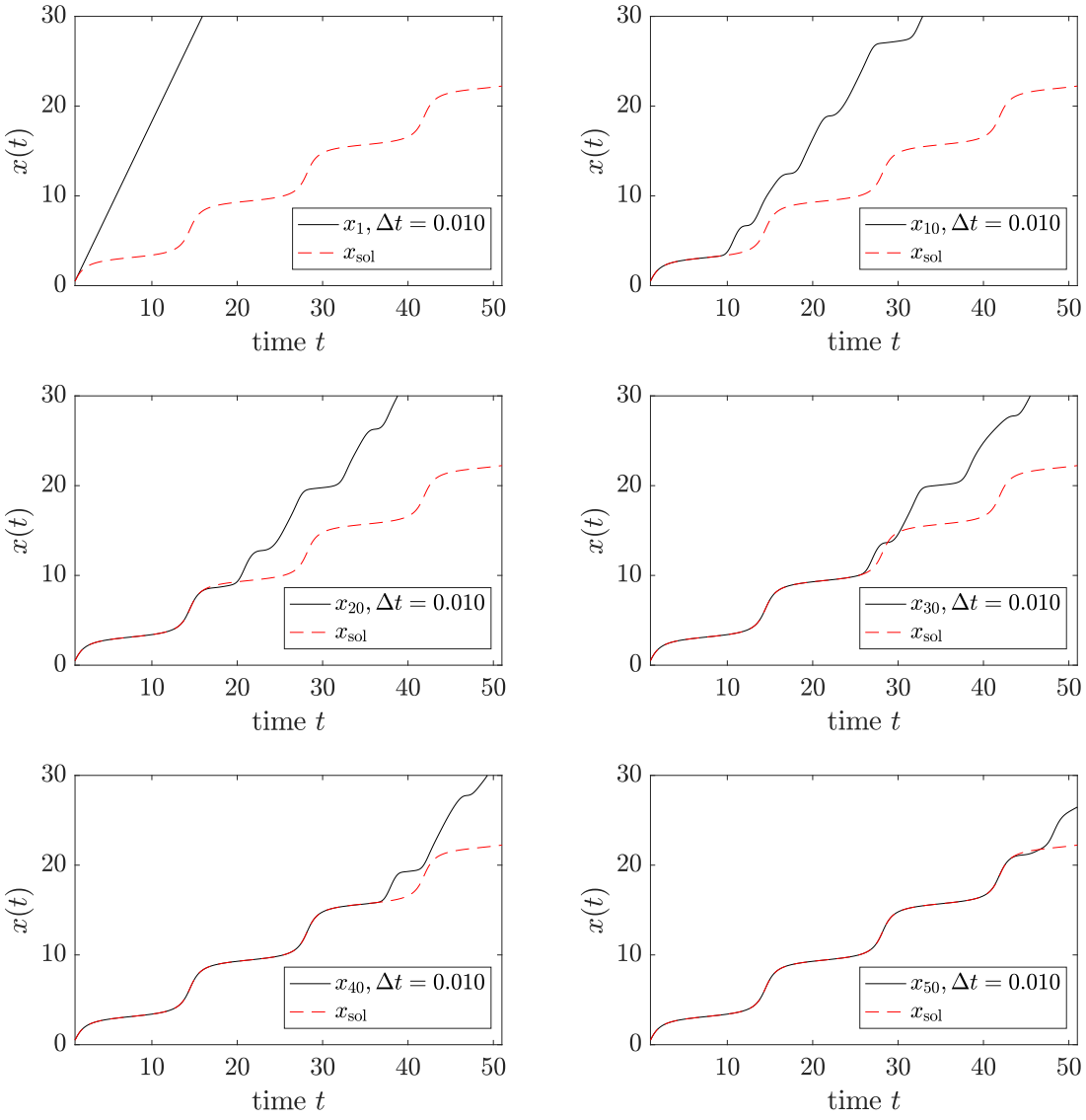
2 Numerical Implementation of Picard Iteration

It is possible to use Picard iteration method to solve differential equations numerically. Write a Matlab program that numerically approximates the first N Picard iterations on a set of discrete times $t_i = t_0 + i\Delta t,\ i \in \mathbb{N}$ of an interval $[t_0, t_f]$. Start only with a first order differential equation $\dot{x}(t) = f(x(t))$ and consider $f(x) = \cos(x) + 1.1$, with $t_0 = 1$, $t_f = 51$ and $x_0 = 1/2$.

(a) Start by computing the first Picard iteration $x_1(t)$. Initialise all the necessary parameters, i.e. the discretised time vector $\vec{t} = (t_0, t_0 + \Delta t, \ldots, t_f)$, the zeroth Picard interation, which is constant $x_0(t_i) = x_0, \forall i$, and so on. The next iteration is $x_1(t) = x_0 + \int_{t_0}^{t} f(x_0(s))\mathrm{d}s$, therefore numerical integration is needed. This integral needs to be computed for all the discrete values of time $t = t_i$. To limit redundant computations and gain in speed, when evaluating $\int_{t_0}^{t_{i+1}} f(x_k(s))\mathrm{d}s$ you can reuse the integral $\int_{t_0}^{t_i} f(x_k(s))\mathrm{d}s$ computed for $x_{k+1}(t_i)$ and add the next bit $\int_{t_i}^{t_{i+1}} f(x_k(s))\mathrm{d}s$. We suggest to use the trapezoidal rule. Verify that $x_1(t)$ is a straight line.

See the code in the `integration` function.

(b) Now, just repeat the process to get the N^{th} Picard iteration and show visually that the iterations converge to x_{sol}, the solution of the differential equation computed with ode45.

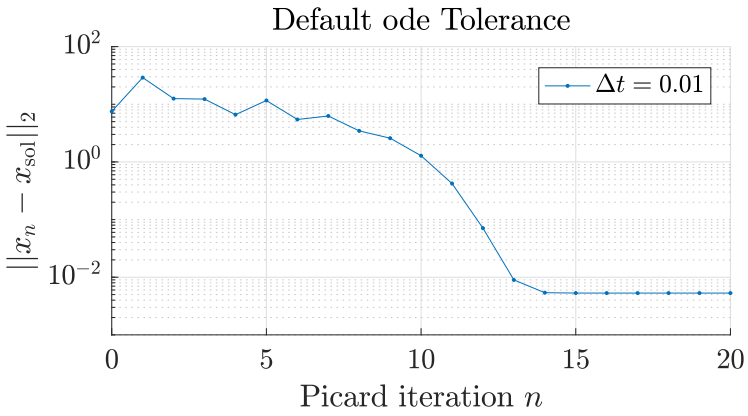


It is clear that the Picard iterations are converging to the correct result. In class we even proved that it converges uniformly for such a closed interval.

Make sure the relative error tolerance and the Δt are small enough to get this result.

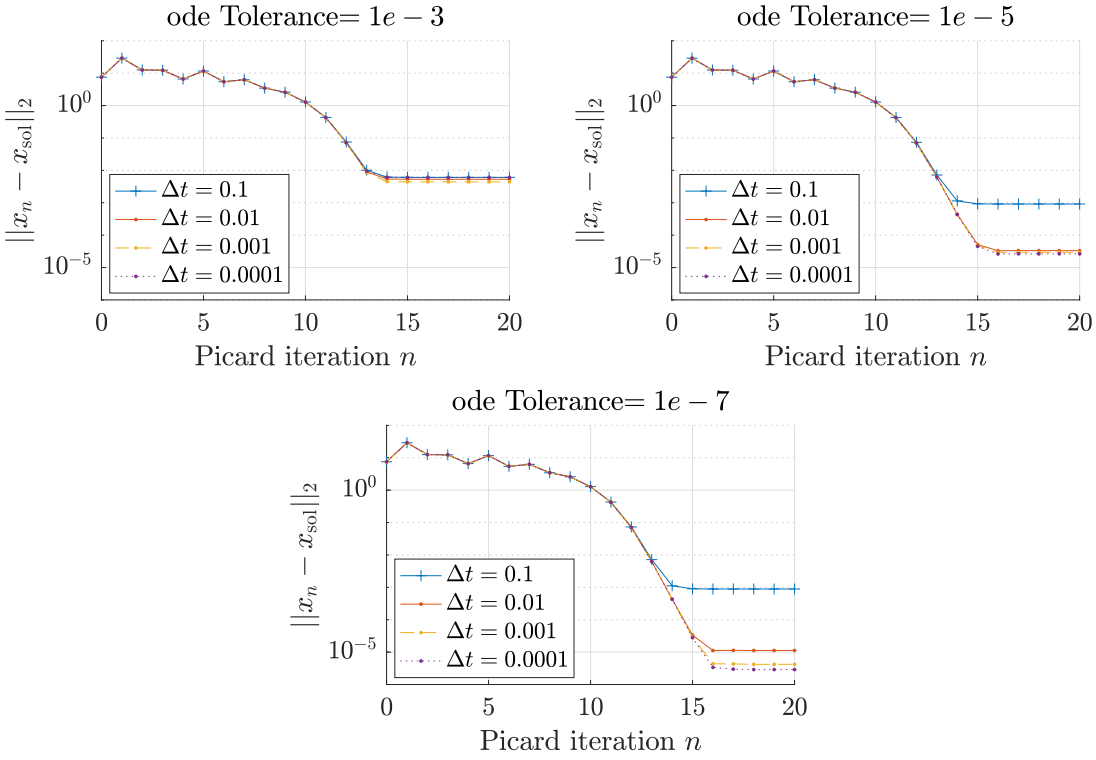
(c) The norm $||x_a - x_b||_2 = \sqrt{\int_{t_0}^{t_f} [x_a(t) - x_b(t)]^2 \mathrm{d}t}$ can be used to measure the distance between two functions. For $t_f = 11$ and $\Delta t = 10^{-1}$ plot the distance with the final solution $||x_n - x_{\text{sol}}||_2$ as a function of n (ranging from $n = 0$ to 20). Use the default relative error tolerance `rel_tol=1e-3`. Identify what causes the error to plateau for high-n Picard iterations.

Tip : Instead of giving `ode45` only the initial and final time `[t0,tf]` and let it choose at which times it evaluates the solution, you can give `ode45` the time vector $\vec{t}$ where the solutions should be evaluated. Also, to set the relative error tolerance, use `ode45(@f,t,x0,options)` with `options = odeset('RelTol',1e-5)`.



The Picard iterations indeed converge to the final solution as the distance drops from over 10^1 at $n = 1$ to below 10^{-2} at $n = 13$. Then the distance plateaus off, due to numerical error in x_{sol} solution controlled by the `rel_tol` of `ode45` and the error in x_n controlled by Δt.

(d) Plot again $||x_n - x_{\text{sol}}||_2$ as a funtion of n to compare on the same plot the time steps $\Delta t = 0.1$, 0.01, 0.001 and 0.0001. Make the plot for x_{sol} evaluated with `rel_tol=1e-3`, `1e-5` and `1e-7`. Explain the results.



For the default `rel_tol=1e-3` the precision of x_{sol} is the limiting factor. The distance $||x_n - x_{\text{sol}}||_2$ should in theory decrease to zero, but it plateaus at 10^{-2} for all the time steps, due to the lack of precision of x_{sol} computed by `ode45`. When `rel_tol` is increased to `1e-5`, still the distance $||x_n - x_{\text{sol}}||_2$ plateaus for three smaller time steps but this time at a more precise $5 \cdot 10^{-5}$. For $\Delta t = 0.1$ the precision of x_n is the limiting factor since the plateau is at a higher 10^{-3}. Finally for `rel_tol=1e-7`, the precision of x_{sol} is no longer the limiting factor, as all the time steps plateau at different precisions, and making `rel_tol` smaller does not change the graph.

BONUS

(e) If your code is well organized, you can try with only a few changes to compute the Picard iterations for a system of arbitrary dimension !

3 Lipschitz Constant

Find a Lipschitz constant, if it exists, for the following functions in the indicated domains.
Reminder : A Lipschitz constant, K, on the domain D is a constant such that $||\vec{f}(\vec{x}) - \vec{f}(\vec{y})||_2 < K||\vec{x} - \vec{y}||_2$, $\forall \vec{x}, \vec{y} \in D$, where $||\cdot||_2$ is the standard Euclidean norm.

(a) $f(x) = \cos(\omega x)$, $x \in [-\pi, \pi]$

For a $\mathcal{C}^1$ function on a closed bounded domain D, the smallest Lipschitz constant K is given by $K = \max_{\vec{x} \in D} ||\vec{\nabla} f(\vec{x})||_2$.

The derivative is $|f'(x)| = |\omega \sin(\omega x)|$. Since the sine is odd, then we only need to find the maximum on the interval $[0, \pi]$, where $|f'(x)| = |\omega| \sin(|\omega| x)$.

- If $|\omega|\pi < \pi/2$ then the funciton is monotonic increasing and the maximum is at the edge of the interval, $\max_{x \in [-\pi, \pi]} |f'(x)| = |f'(\pm\pi)| = |\omega| \sin(|\omega|\pi) = K$
- Otherwise if $|\omega|\pi \geq \pi/2$ then $\max_{x \in [-\pi, \pi]} |f'(x)| = |f'(\frac{\pi}{2|\omega|})| = |\omega| \sin(\pi/2) = |\omega| = K$

(b) $f(x) = \sqrt[3]{x}$, $x \in [-1, 1]$

No Lipschitz constant exists since $\lim_{x \to 0} f'(x) = +\infty$.

(c) $f(x_1, x_2) = \frac{x_1 x_2}{1 + x_1^2 + x_2^2}$, $x_1^2 + x_2^2 \leq 16$

We first evaluate

$$\left|\left|\vec{\nabla} f(x,y)\right|\right|_2 = \left|\left|\begin{bmatrix} \frac{y(1+x^2+y^2) - xy2x}{(1+x^2+y^2)^2} \\ \frac{x(1+x^2+y^2) - xy2y}{(1+x^2+y^2)^2} \end{bmatrix}\right|\right|_2 = \frac{1}{(1+x^2+y^2)^2} \left|\left|\begin{bmatrix} y(1+x^2+y^2) - 2x^2 y \\ x(1+x^2+y^2) - 2xy^2 \end{bmatrix}\right|\right|_2$$

$$= \sqrt{y^2(1 - x^2 + y^2)^2 + x^2(1 + x^2 - y^2)^2}/(1 + x^2 + y^2)^2$$

$$= \sqrt{y^2(1 + x^4 + y^4 - 2x^2 + 2y^2 - 2x^2y^2) + x^2(1 + x^4 + y^4 + 2x^2 - 2y^2 - 2x^2y^2)}/(1 + r^2)^2$$

$$= \sqrt{r^2 + x^4 r^2 + y^4 r^2 - 4y^2 x^2 + 2y^4 + 2x^4 - 2x^2 y^2 r^2}/(1 + r^2)^2$$

$$= \sqrt{r^2 + x^4(2 + r^2) + y^4(2 + r^2) - 2y^2 x^2 (2 + r^2)}/(1 + r^2)^2$$

$$= \sqrt{r^2 + (x^2 - y^2)^2 (2 + r^2)}/(1 + r^2)^2$$

Any number larger than the maximum of $||\vec{\nabla} f(x,y)||_2$ on the domain D is a Lipschitz constant. At a constant r, to maximize $||\vec{\nabla} f(x,y)||_2$ we need to maximize $(x^2 - y^2)^2 = (x^2 - (r^2 - x^2))^2 = (2x^2 - r^2)^2$. Since $x^2 \in [0, r^2]$, $(2x^2 - r^2)^2$ has its maximum equal to r^4 when $x^2 = \{0, r^2\}$ i.e. when $(x, y) = \{(0, r), (0, -r), (r, 0), (-r, 0)\}$. It follows

$$\left|\left|\vec{\nabla} f(x,y)\right|\right|_2 = \frac{\sqrt{r^2 + r^4(2 + r^2)}}{(1 + r^2)^2} = \frac{r\sqrt{1 + 2r^2 + r^4}}{(1 + r^2)^2} = \frac{r}{1 + r^2}$$

Now, we just need to find the maximum of $\frac{r}{1+r^2}$. We therefore evaluate $\frac{\mathrm{d}}{\mathrm{d}r}[\frac{r}{1+r^2}] = 0$, that is $\frac{1(1+r^2) - r2r}{(1+r^2)^2} = \frac{1 - r^2}{(1+r^2)^2} = 0$, which occurs at $r^2 = 1$. As a consequence the maximum is the infimum of all possible Lipschitz constants $K = 1/(1 + 1) = 1/2$ is reached at $(x, y) = \{(0, 1), (0, -1), (1, 0), (-1, 0)\}$.

Mathematica can also find this maximum as shown in `Lipschitz.nb`.