Numerical Solution of a Nonlinear Dynamical System
by a Collocation Method based on Cubic B-spline Basis

Author : Francesca Pitolli

References :

1) F. Pitolli, Fractals&Fractionals, 2018
2) F. Pitolli, Axioms, 2018
3) E. Pellegrino, L. Pezza, F. Pitolli, submitted, 2019
4) F. Pitolli, Algorithms, 2019

Version: July, 2019

Define the Nonlinear Dynamical System

Test Problems :

1) Linear dynamical system
2) Nonlinear dynamical system

In[ ]:=

```
(* Known term of the dynamical system *)
Clear[y, z, w, F, JF]
Problem = 1;
Which[Problem == 1, F = {y + z + w, -y + z - w, -Gamma[3/2] w};,
 Problem == 2, F = {y + z + 100 Sqrt[π] y^2 + w, -y + z + 100 Sqrt[π] y^2 - w, w^2 - 1/200 Sqrt[π] w};
]
Y = {y, z, w};
nF = Dimensions[F][[1]];

(* Initial conditions *)
Clear[Y0]
Y0  = {0.05, 0.05, 0.05};

(* Jacobian matrix of F *)
Clear[JF]
JF = Outer[D, F, Y];
MatrixForm[JF]

(* Order of the time fractional derivative ( 0 < b < 1) *)
b = 0.5;
```

Basis Functions

Left-edge boundary functions

In[ ]:=
```
(* First left edge function *)
Clear[N03]

N03[x_] := Piecewise[{{(1 - x)^3, 0 ≤ x ≤ 1}}]

(* Second left edge function *)
Clear[N13]
N13[x_] := Piecewise[{{1/4 (2 - x)^3 - 2 (1 - x)^3, 0 ≤ x ≤ 1}, {1/4 (2 - x)^3, 1 ≤ x ≤ 2}}]

(* Third left edge function *)
Clear[N23]

N23[x_] :=
 Piecewise[{{1/6 (3 - x)^3 - 3/4 (2 - x)^3 + 3/2 (1 - x)^3, 0 ≤ x ≤ 1}, {1/6 (3 - x)^3 - 3/4 (2 - x)^3, 1 ≤ x ≤ 2}, {1/6 (3 - x)^3, 2 ≤ x ≤ 3}}]
```

Cubic B - Spline

In[ ]:=
```
Clear[B3]
B3[x_] := Piecewise[{{1/6 x^3, 0 ≤ x ≤ 1}, {1/6 (x^3 - 4 (x - 1)^3), 1 ≤ x ≤ 2},
  {1/6 (x^3 - 4 (x - 1)^3 + 6 (x - 2)^3), 2 ≤ x ≤ 3}, {1/6 (x^3 - 4 (x - 1)^3 + 6 (x - 2)^3 - 4 (x - 3)^3), 3 ≤ x ≤ 4}}]
```

In[ ]:=
```
Plot[{N03[x], N13[x], N23[x], B3[x]}, {x, 0, 4}, PlotRange → All]
```

Caputo derivative of fractional order b (0 < b < 1)

Caputo Derivative of the interior functions

In[ ]:=
```
(* Derivative of the first left edge function *)
Clear[derN03]
```

$$\text{derN03}[x\_, b\_] := \frac{-3}{\text{Gamma}[1-b]\,(-3+b)\,(-2+b)\,(-1+b)}\ \text{Piecewise}\Big[\Big\{\big\{-x^{1-b}\left(6-5b+b^2-6x+2bx+2x^2\right),\ 0\le x\le 1\big\},\ \big\{2\,(-1+x)^{3-b}-6x^{1-b}+5bx^{1-b}-b^2x^{1-b}+6x^{2-b}-2bx^{2-b}-2x^{3-b},\ 1\le x\big\}\Big\}\Big]$$

```
(* Derivative of the second left edge function *)
Clear[derN13]
```

$$\text{derN13}[x\_, b\_] := \frac{-3}{\text{Gamma}[1-b]\,(-3+b)\,(-2+b)\,(-1+b)}\ \text{Piecewise}\Big[\Big\{\big\{6x^{1-b}-5bx^{1-b}+b^2x^{1-b}-9x^{2-b}+3bx^{2-b}+\frac{7x^{3-b}}{2},\ 0\le x\le 1\big\},\ \big\{-4\,(-1+x)^{3-b}+6x^{1-b}-5bx^{1-b}+b^2x^{1-b}-9x^{2-b}+3bx^{2-b}+\frac{7x^{3-b}}{2},\ 1\le x\le 2\big\},\ \big\{\frac{1}{2}\,(-2+x)^{3-b}-4\,(-1+x)^{3-b}+6x^{1-b}-5bx^{1-b}+b^2x^{1-b}-9x^{2-b}+3bx^{2-b}+\frac{7x^{3-b}}{2},\ 2\le x\big\}\Big\}\Big]$$

```
(* Derivative of the third left edge function *)
Clear[derN23]
```

$$\text{derN23}[x\_, b\_] := \frac{-3}{\text{Gamma}[1-b]\,(-3+b)\,(-2+b)\,(-1+b)}\ \text{Piecewise}\Big[\Big\{\big\{x^{\wedge}(1-b)\left(-\frac{11}{6}x^{\wedge}2+(3-b)\,x\right),\ 0\le x\le 1\big\},\ \big\{x^{\wedge}(1-b)\left(-\frac{11}{6}x^{\wedge}2+(3-b)\,x\right)+3\,(x-1)^{\wedge}(3-b),\ 1\le x\le 2\big\},\ \big\{x^{\wedge}(1-b)\left(-\frac{11}{6}x^{\wedge}2+(3-b)\,x\right)+3\,(x-1)^{\wedge}(3-b)-\frac{3}{2}\,(x-2)^{\wedge}(3-b),\ 2\le x\le 3\big\},\ \big\{x^{\wedge}(1-b)\left(-\frac{11}{6}x^{\wedge}2+(3-b)\,x\right)+3\,(x-1)^{\wedge}(3-b)-\frac{3}{2}\,(x-2)^{\wedge}(3-b)+\frac{1}{3}\,(x-3)^{\wedge}(3-b),\ 3\le x\big\}\Big\}\Big]$$

Caputo Derivative of the cubic B – spline

In[]:=
```
Clear[derB3]
```

```
derB3[x_, b_] := 1/Gamma[4 - b]
  Piecewise[{{x^(3 - b), 0 ≤ x ≤ 1}, {x^(3 - b) - 4 (x - 1)^(3 - b), 1 ≤ x ≤ 2}, {x^(3 - b) - 4 (x - 1)^(3 - b) + 6 (x - 2)^(3 - b), 2 ≤ x ≤ 3},
    {x^(3 - b) - 4 (x - 1)^(3 - b) + 6 (x - 2)^(3 - b) - 4 (x - 3)^(3 - b), 3 ≤ x ≤ 4},
    {x^(3 - b) - 4 (x - 1)^(3 - b) + 6 (x - 2)^(3 - b) - 4 (x - 3)^(3 - b) + (x - 4)^(3 - b), 4 ≤ x}}]
```

In[]:=
```
Plot[{derN03[x, b], derN13[x, b], derN23[x, b], derB3[x, b]}, {x, 0, 8}, PlotRange → All]
```

Inputs for the collocation method

In[ ]:=
```
(* Discretization interval: [0,T] *)
T = 8;

(* Refinement step *)
js = 4;
h = 1/(2^js)

(* Number of interior basis functions *)
Nkint = T / h;

(* Number of left edge basis functions *)
Nkedge = 3;

(* Number of basis functions *)
Nk = Nkint + Nkedge
```

Out[ ]= $\frac{1}{16}$

Out[ ]= 131

In[ ]:=

Collocation Points

In[ ]:=
```
Clear[TPoints]
dt = h/2;
Ns = T / dt
TPoints = Table[i dt, {i, 1, Ns}]

If[Ns < Nk, Print["ERROR: Ns should be greater than or equal to Nk"]]
```

Out[ ]= 256

Collocation Matrix for the Cubic B-spline Basis

In[ ]:=
```
(* Interior functions *)
Clear[Mcoll0int]
Mcoll0int = Table[N[B3[TPoints[[i]]/h - k]], {i, 2, Ns}, {k, 0, Nkint - 1}];

(* Edge functions *)
Clear[Mcoll0edge]
Mcoll0edge = Transpose[Table[{N[N13[TPoints[[i]]/h]], N[N23[TPoints[[i]]/h]]}, {i, 2, Ns}]];

(* Collocation matrix *)
Mcoll0 = Join[Mcoll0edge, Transpose[Mcoll0int]];
Dimensions[Mcoll0]
Clear[Mcoll0int, Mcoll0edge]
```

Collocation Matrix for the Caputo Derivative of the B-spline Basis

```
In[ ]:= (* Normalization factor for the derivative*)
normder = 2^(js b);

(* Interior functions *)
Clear[Mcoll1int]
Mcoll1int = Table[N[normder derB3[TPoints[[i]]/h - k, b]], {i, 2, Ns}, {k, 0, Nkint - 1}];

(* Second and third edge functions *)
Clear[Mcoll1edge]
Mcoll1edge = Transpose[Table[{N[normder derN13[TPoints[[i]]/h, b]], N[normder derN23[TPoints[[i]]/h, b]]}, {i, 2, Ns}]];

(* Collocation matrix *)
Mcoll1 = Join[Mcoll1edge, Transpose[Mcoll1int]];
Dimensions[Mcoll1]
Clear[Mcoll1int, Mcoll1edge]
```

Solution of the Nonlinear System

Iteration loop

```
In[ ]:= (* Assign the initial guess and the parameters*)
Clear[lambda0]
lambda0 = Table[0., {i , 1, nF}, {j, 1, Nk - 1}] ;
tau = 0.1;
Normlam = 10 tau;
maxiter = 10;
ell = 0;

While[Normlam > tau && ell < maxiter,
```

```
(* Evaluate the approximate solution at the previous iteration step *)
Clear[Ys];
Ys = Table[Transpose[Mcoll0].lambda0[[n]] , {n, 1, nF}] + Y0;
Print[ListPlot[Table[{TPoints[[i]], Ys[[k, i - 1]]}, {k, 1, nF}, {i, 2, Ns}],
  Joined → True, PlotRange → All, PlotLabel → "Solution"]];

(* Evaluate the Jacobian matrix at the previous iteration step *)
JFNodes = Table[JF /. {y -> Ys[[1, i]], z -> Ys[[2, i]], w -> Ys[[3, i]]} , {i, 1, Ns - 1}];

(* Construct the second term of the Jacobian matrix
  J_G by multipling the Jacobian matrix J_F with the collocation B-spline matrix *)
JFblock = Table[Outer[Times, JFNodes[[i]], Mcoll0[[i]]], {i, 1, Nk - 1}];
JJ = ArrayFlatten[Transpose[JFblock, {4, 1, 2, 3}], 2];

(* Construct the Jacobian matrix J_G *)
Clear[Mcoll];
Mcoll = KroneckerProduct[IdentityMatrix[nF], Transpose[Mcoll1]] - JJ;

(* Construct the known term *)
Clear[BG];
BG = KroneckerProduct[IdentityMatrix[nF], Transpose[Mcoll1]] .Flatten[lambda0] -
  Flatten[Transpose[Table[F /. {y -> Ys[[1, i]], z -> Ys[[2, i]], w -> Ys[[3, i]]} , {i, 1, Ns - 1}]]];

(* Solve the linear system by the least squares method *)
Clear[GamK];
GamK = LeastSquares[Mcoll, N[BG]];

(* Update the unknown coefficients *)
Clear[lambda];
lambda = lambda0 - Partition[GamK, Nk - 1];

(* Evaluate the infinity norm of the relative error *)
Normlam = Norm[lambda - lambda0, Infinity] / Norm[lambda, Infinity];

(* Update the iteration counter and the privious solution *)
```

```
ell ++;
lambda0 = lambda;

(* Print the relative error at iteration ell *)
Print["Iteration = ", ell, "   Norm = ", Normlam];
]
```

Evaluate the Numerical Solution

In[ ]:=
```
Clear[B3sint, B3sedge, B3s]

(* Points where to evaluate the numerical solution *)
h2 = h/4;
Nr = T/h2 + 1;
Tval = Table[i h2, {i, 0, Nr - 1}];

(* Function basis on the evaluation points *)
B3sint = Table[N[B3[Tval[[i]]/h - k]], {i, 1, Nr}, {k, 0, Nkint - 1}];

B3sedge = Transpose[Table[{N[N13[Tval[[i]] / h]], N[N23[Tval[[i]]/h]]}, {i, 1, Nr}]];

B3s = Join[B3sedge, Transpose[B3sint]];

(* Print the numerical solution *)
Clear[Ys]
Ys = Table[Transpose[B3s].lambda[[n]], {n, 1, nF}] + Y0;
ListPlot[Table[{Tval[[i]], Ys[[k, i]]}, {k, 1, nF}, {i, 1, Nr}], Joined → True, PlotRange → All]
```