```
(*****************************************************************)
(*                                                               *)
(* Generalized Collocation Methods:Solutions to Nonlinear Problems *)
(* Bellomo,N.,Lods,B.,Revelli,R.,Ridolfi,L.                      *)
(* A Birkhäuser book                                             *)
(* ISBN:978-0-8176-4525-0                                        *)
(*                                                               *)
(* Program DiffNL                                                *)
(*                                                               *)
(*****************************************************************)

$TextStyle = {FontFamily -> "Times", FontSize → 12};
Off[General::"spell", General::"spell1"]

DiffNL[InitBoundData_, Nodes_, η_, Type_] := Module[{},

   (*** INITIAL CONDITION ***)

   φ[x_] := InitBoundData[[2]];

   hh = 1/(Nodes - 1);

   Which[Type === Sinc, x1_i_ := (i - 1) * hh,
    Type === Lagrange, x1_i_ := -1/2. (Cos[(i - 1) * π/(Nodes - 1)] - 1)];


   If[Type === Lagrange, Lagr[j_, x_] := ∏_{p=1}^{Nodes} (If[p ≠ j, (x - x1_p)/(x1_j - x1_p), 1]);

    InitValue[x_] := ∑_{k=1}^{Nodes} (φ[x] /. x → x1_k) * Lagr[k, x]];
   If[Type === Sinc, Sinc[j_, x_] := Which[0 ≤ x ≤ 1 && x ≠ (j - 1) * hh,
      Sin[π * (x - (j - 1) * hh)/hh] / (π * (x - (j - 1) * hh)/hh), x == (j - 1) * hh, 1];
    InitValue[x_] := ∑_{k=1}^{Nodes} (φ[x] /. x -> x1_k) * Sinc[k, x]];

   Print["Initial condition interpolation:"];
   Plot[Evaluate[{φ[x], InitValue[x]}], {x, 0, 1},
    PlotStyle → {GrayLevel[0], Dashing[{0.15, 0.05}]}, AxesLabel → {x, u[x, 0]}];

   Print["Don't worry! I'm working"];

   (*** Derivative Matrices ***)

   If[Type === Lagrange,
    FDM[j_, i_] := Which[i == j, ∑_{k=1}^{Nodes} If[k == i, 0, 1/(x1_i - x1_k)], i != j,
      (∏_{p=1}^{Nodes} (If[p == i, 1, x1_i - x1_p])) / ((x1_i - x1_j) ∏_{p=1}^{Nodes} (If[p == j, 1, x1_j - x1_p]))];

    FirstDer = Table[FDM[j, i], {j, 1, Nodes}, {i, 1, Nodes}];
    SecondDer =
```

$$\text{Table}\left[\text{If}\left[i \neq j,\ 2\left(\text{FirstDer}[[j, i]] * \text{FirstDer}[[i, i]] - \frac{\text{FirstDer}[[j, i]]}{x1_i - x1_j}\right), 0\right],\ \{j, 1, \text{Nodes}\}, \{i, 1, \text{Nodes}\}\right];$$

$$\text{Do}\left[\text{SecondDer}[[i, i]] = -\sum_{k=1}^{\text{Nodes}} \text{SecondDer}[[k, i]], \{i, 1, \text{Nodes}\}\right]\Big];$$

$\text{If}\Big[\text{Type} === \text{Sinc},$

$$\text{FDM}[j\_, i\_] := \text{Which}\left[i\ != j,\ \frac{(-1)^{i-j}}{hh * (i-j)},\ i == j,\ 0\right];$$

$$\text{SDM}[j\_, i\_] := \text{Which}\left[i\ != j,\ \frac{2 * (-1)^{i-j+1}}{hh^2 * (i-j)^2},\ i == j,\ -\frac{\pi^2}{3 * hh^2}\right];$$

```
FirstDer = Table[FDM[j, i], {j, 1, Nodes}, {i, 1, Nodes}];
SecondDer = Table[SDM[j, i], {j, 1, Nodes}, {i, 1, Nodes}]];

(*** Nodal Equations ***)
```

$$\text{NdEq}[i\_] := C_i\,'[t] == (2\,C_i[t] - 1) * \left(\sum_{k=1}^{\text{Nodes}} \text{FirstDer}[[k, i]] * C_k[t]\right) + \eta\,(C_i[t])^2\,(1 - C_i[t])\left(\sum_{k=1}^{\text{Nodes}} \text{SecondDer}[[k, i]] * C_k[t]\right) + \eta\,(C_i[t])\,(2 - 3\,C_i[t])\left(\sum_{k=1}^{\text{Nodes}} \text{FirstDer}[[k, i]] * C_k[t]\right)^2\ /.\ x \to x1_i;$$

ψ = InitBoundData[[1]];
ξ = InitBoundData[[3]];

(******* Determination of u_1 and u_n *******)

$$\gamma = \sum_{k=1}^{\text{Nodes}} \text{FirstDer}[[k, 1]] * C_k[t];$$

$$\delta = \sum_{k=1}^{\text{Nodes}} \text{FirstDer}[[k, \text{Nodes}]] * C_k[t];$$

halpha = -1 - (1 - α) γ η + α γ η;
kbeta = -1 - (1 - β) δ η + β δ η;
hgamma = - (1 - α) α η;
kdelta = - (1 - β) β η;

```
aa = halpha + hgamma * FirstDer[[1, 1]];
bb = hgamma * FirstDer[[Nodes, 1]];
dd = kdelta * FirstDer[[1, Nodes]];
ee = kbeta + kdelta * FirstDer[[Nodes, Nodes]];
```

$$cc = \partial_t \psi - \text{hgamma}\left(\sum_{k=2}^{\text{Nodes}-1} (\text{FirstDer}[[k, 1]] * \text{NdEq}[k][[2]])\right);$$

$$ff = \partial_t \xi - \text{kdelta}\left(\sum_{k=2}^{\text{Nodes}-1} (\text{FirstDer}[[k, \text{Nodes}]] * \text{NdEq}[k][[2]])\right);$$

$$\text{dealpha} = \frac{cc\ ee - bb\ ff}{-bb\ dd + aa\ ee};$$

$$\text{debeta} = \frac{cc\ dd - aa\ ff}{bb\ dd - aa\ ee};$$

```
Equat1 := C1 '[t] == dealpha;
EquatNodes := CNodes '[t] == debeta;

(******* End of Determination of u1 and un *******)

EqsSys = Flatten[Table[{NdEq[i] , Ci [0] == (φ[x] /. x → x1i)}, {i, 2, Nodes - 1}]];
NdEqsSys = Flatten[Join[{EqsSys, Equat1, C1 [0] == (φ[x] /. x → x11), EquatNodes,
      CNodes [0] == (φ[x] /. x → x1Nodes)}]] /. {α -> C1 [t], β -> CNodes [t]};
UnKnown = Join[Table[Ci , {i, 1, Nodes}]];
NumSol = NDSolve[NdEqsSys, UnKnown,
    {t, 0, 1}, MaxSteps → Infinity, Method → {StiffnessSwitching,
      Method → {ExplicitEuler, ImplicitRungeKutta}}] // Flatten;


(*** Solution ***)

Which[Type === Sinc, TotSol[x_, t_] := Sum[NumSol[[k, 2]][t] * Sinc[k, x], {k, 1, Nodes}],

 Type === Lagrange, TotSol[x_, t_] := Sum[NumSol[[k, 2]][t] * Lagr[k, x], {k, 1, Nodes}]];

Print["Solution:"];

Plot3D[Evaluate[TotSol[x, t]], {x, 0, 1}, {t, 0, 1}, PlotRange →
   {{0, 1}, {0, 1}, {-0.001, 0.3}}, AxesLabel → TraditionalForm /@ {x, t, "u(x,t)"},
  Ticks → {{0, 1}, {0, 1}, {0.0, 0.3}}, PlotPoints → 101];
];


{Nodes, η} = {21, 0.1};
InitCond = 4 (x)^2 (x - 1)^2 ;
Bound0Cond = 0;
Bound1Cond = 0.01 t Exp[1 - t];
IBData = {Bound0Cond, InitCond, Bound1Cond};

DiffNL[IBData, Nodes, η, Lagrange]
```