```
(*****************************************************************)
(*                                                               *)
(* 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 TwoDLaSiErr                                           *)
(*                                                               *)
(*****************************************************************)

<< Graphics`MultipleListPlot`

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

TwoDLaSiErr[function_, nodes_, deltax_, deltay_] := Module[{},
  (*** INITIAL CONDITION ***)
  {NodesX, NodesY} = {nodes, nodes};
  φ[x_, y_] := function;
  {hhX, hhY} = {1/(NodesX - 1), 1/(NodesY - 1)};
  x_i_ := (i - 1) * hhX;
  y_i_ := (i - 1) * hhY;
  x1_i_ := -1/2 (Cos[(i - 1) * π/(NodesX - 1)] - 1);
  y1_i_ := -1/2 (Cos[(i - 1) * π/(NodesY - 1)] - 1);

  (*** Lagrange polynomial definition ***)

  Lagr1X[j_, x_] := Product[(If[p ≠ j, (x - x1_p)/(x1_j - x1_p), 1]), {p, 1, NodesX}];

  Lagr1Y[j_, y_] := Product[(If[p ≠ j, (y - y1_p)/(y1_j - y1_p), 1]), {p, 1, NodesY}];

  FunctionLagr1[x_, y_] :=
   Sum[Sum[((φ[x, y] /. {x → x1_k, y → y1_m}) * Lagr1X[k, x] * Lagr1Y[m, y]), {k, 1, NodesX}], {m, 1, NodesY}];

  (*** Sinc function definition ***)

  SincX[j_, x_] := Which[0 ≤ x ≤ 1 && x != (j - 1) * hhX,
    Sin[(π * (x - (j - 1) * hhX))/hhX] / ((π * (x - (j - 1) * hhX))/hhX), x == (j - 1) * hhX, 1];
  SincY[j_, y_] := Which[0 ≤ y ≤ 1 && y != (j - 1) * hhY,
    Sin[(π * (y - (j - 1) * hhY))/hhY] / ((π * (y - (j - 1) * hhY))/hhY), y == (j - 1) * hhY, 1];

  FunctionSinc[x_, y_] :=
   Sum[Sum[(φ[x, y] /. {x -> x_k, y → y_m}) * SincX[k, x] * SincY[m, y], {k, 1, NodesX}], {m, 1, NodesY}];

  (*** Error Evaluation ***)
```

```
  ErrSinc = Table[Evaluate[Abs[(φ[x, y] - FunctionSinc[x, y])/1]],
    {x, 0, 1, deltax}, {y, 0, 1, deltay}];
  ErrLagr = Table[Evaluate[Abs[(φ[x, y] - FunctionLagr1[x, y])/1]],
    {x, 0, 1, deltax}, {y, 0, 1, deltay}];

  {MaxSinc, MaxLagr} = {Max[Flatten[ErrSinc]], Max[Flatten[ErrLagr]]};
  {MeanSinc, MeanLagr} = {Mean[Flatten[ErrSinc]], Mean[Flatten[ErrLagr]]};

  Print["Nodes ", " MaxSinc ", " MaxLagr ", " MeanSinc ", " MeanLagr "];
  Print[nodes, " ", MaxSinc, " ", MaxLagr, " ", MeanSinc, " ", MeanLagr];

  ErrorMax[NodesX] = {NodesX, MaxSinc, MaxLagr};
  ErrorMean[NodesX] = {NodesX, MeanSinc, MeanLagr};];

{function, MinNodes, MaxNodes, deltax, deltay} =
  {Exp[-25 (2 x - 1)^2 - 25 (2 y - 1)^2], 2, 41, 0.05, 0.05};
For[i = 2, i < MaxNodes, TwoDLaSiErr[function, i, deltax, deltay]; i = i + 1]

SincMaxOdd =
  Table[{ErrorMax[j][[1]], ErrorMax[j][[2]]}, {j, MinNodes + 1, MaxNodes - 1, 2}];
LagrMaxOdd = Table[{ErrorMax[j][[1]], ErrorMax[j][[3]]},
   {j, MinNodes + 1, MaxNodes - 1, 2}];
SincMeanOdd = Table[{ErrorMean[j][[1]], ErrorMean[j][[2]]},
   {j, MinNodes + 1, MaxNodes - 1, 2}];
LagrMeanOdd = Table[{ErrorMean[j][[1]], ErrorMean[j][[3]]},
   {j, MinNodes + 1, MaxNodes - 1, 2}];

(*** Max  Error for odd number of nodes ***)

p2a = ListPlot[SincMaxOdd, PlotJoined → True,
   PlotRange → {{-0.01, 40}, {-0.02, 1.05}}, Frame → True, FrameTicks →
    {{{0, 0, .02}, {10, 10, .02}, {20, 20, .02}, {30, 30, .02}, {40, 40, 0.02}},
     {{-1, -1, .02}, {0, 0, .02}, {1, 1, 0.02}}, None, None}, PlotStyle → {GrayLevel[0]},
   FrameLabel → TraditionalForm /@ {n, ℰ^n}, DisplayFunction → Identity];
p2b = ListPlot[LagrMaxOdd, PlotJoined → True, PlotRange → {{-0.01, 40}, {-0.02, 1.05}},
   Frame → True, FrameTicks → {{{0, 0, .02}, {10, 10, .02}, {20, 20, .02}, {30, 30, .02},
      {40, 40, 0.02}}, {{-1, -1, .02}, {0, 0, .02}, {1, 1, 0.02}}, None, None},
   PlotStyle → {GrayLevel[0], Dashing[{Dash, Dash}]},
   FrameLabel → TraditionalForm /@ {n, ℰ^n}, DisplayFunction → Identity];
 p2 = Show[{p2a, p2b}, DisplayFunction → $DisplayFunction];

(*** Mean  Error for odd number of nodes ***)

p4a = ListPlot[SincMeanOdd, PlotJoined → True,
   PlotRange → {{-0.01, 40}, {-0.02, 0.65}}, Frame → True, FrameTicks →
    {{{0, 0, .02}, {10, 10, .02}, {20, 20, .02}, {30, 30, .02}, {40, 40, 0.02}},
     {{-1, -1, .02}, {0, 0, .02}, {0.6, 0.6, 0.02}}, None, None},
   PlotStyle → {GrayLevel[0]}, FrameLabel → TraditionalForm /@ {n, ℰ1^n},
   DisplayFunction → Identity];
 p4b = ListPlot[LagrMeanOdd, PlotJoined → True,
   PlotRange → {{-0.01, 40}, {-0.02, 0.65}}, Frame → True, FrameTicks →
    {{{0, 0, .02}, {10, 10, .02}, {20, 20, .02}, {30, 30, .02}, {40, 40, 0.02}},
     {{-1, -1, .02}, {0, 0, .02}, {0.6, 0.6, 0.02}}, None, None},
   PlotStyle → {GrayLevel[0], Dashing[{Dash, Dash}]},
   FrameLabel → TraditionalForm /@ {n, ℰ1^n}, DisplayFunction → Identity];
 p4 = Show[{p4a, p4b}, DisplayFunction → $DisplayFunction];
```