```
(*                    Elliptic PDE

  Task1: 1.1. Solve the Laplace equation: ∂x,x u(x,y)+∂y,y u(x,y)=
 0 over the interval Ω=[0,π/2]x[0,π], with the boundary conditions: u[0,y]=Sin[y],
u[x,0]=Sin[2x],u[π/2,y]=0,u[x,π]=0. Take a 4 by 3 grid.
          1.2 Increase the size of the grid by an order of magnitude. Try
  to adjust the number of iterations for a better precission  *)
Clear["`*"];
a = 0; b = π/2; c = 0; d = π;
f[x_, y_] = 0;
m = 4; n = 3;
h = (b - a)/m; k = (d - c)/n; r = k/h;
ω = 4/(2 + Sqrt[4 - (Cos[π/m] + Cos[π/(r*n)])^2]);
Print["The optimal weighted average is ", N[ω]];
(* The list of points,
where x0 and y0 are replaced by x1 and y1, respectively: *)
x1 = a;  y1 = c;
For[i = 1, i ≤ m, i++,
  x(i+1) = x(i) + h; Print["x"(i+1), "=", x(i+1)];];
For[j = 1, j ≤ n, j++,
  y(j+1) = y(j) + k; Print["y"(j+1), "=", y(j+1)];];
(* As a consequence of the increased initial index,
the last point will be given by (x(m+1),y(n+1)). *)
(* Initial conditions: *)
u = Table[5.00, {m + 1}, {n + 1}];
For[ i = 1,  i <= m + 1,  i++, u[[i,1]] = Sin[2 x(i)] ];
For[ i = 1,  i <= m + 1,  i++, u[[i,n +1]] = 0 ];
For[ j = 1,  j <= n + 1,  j++, u[[1,j]] = Sin[y(j)]];
For[ j = 1,  j <= n + 1,  j++, u[[m +1,j]] = 0 ];
u0 = u;  Print["u=", MatrixForm[N[u]]];
Err = 1.0;
While[ And[Err > 0.001 , k ≤ 20],
  Err = 0.0;
  For[ i = 2,  i ≤ m,  i++,
   For[ j = 2,  j ≤ n,  j++,
     u[[i,j]] = r^2/(2 (1 + r^2)) ω
        (u0[[i+1,j]] + u[[i-1,j]] + 1/r^2 (u0[[i,j+1]] + u[[i,j-1]]) - h^2 f[x(i), y(j)]) + (1 - ω) u0[[i,j]];
     Err = Max[Err, Abs[u[[i,j]] - u0[[i,j]]]]; ]; ];
  u0 = u;
  Print["Max grid change = ", Err];
  k = k + 1; ];
```

```
Print["The numerical solution to the P.D.E."];
Print["∂x,x u(x,y)+∂y,y u(x,y)=0"];
ListPlot3D[Transpose[u], AxesLabel → {"y", "x", "u   "},
  ViewPoint → {4, 2, 3}, Lighting → False, ColorFunction → Hue];
Print["The numerical solution to the P.D.E."];
Print["∂x,x u(x,y)+∂y,y u(x,y)=0"];
Print[NumberForm[TableForm[N[u]], 4]];
Print["The numerical solution to the P.D.E."];
Print["∂x,x u(x,y)+∂y,y u(x,y)=0"];
ListContourPlot[Reverse[u], ColorFunction → Hue];
(* Task2: Solve Poissons's equation: ∂x,x u(x,y)+∂y,y u(x,y)=
 e^-(x^2+y^2) over the interval Ω=[0,1]x[0,1],
with the boundary conditions: u|∂Ω=0. Take a 9 by 9 grid. *)
(* Task3: 3.1 Solve the problem formulated at task1 for a large grid (30x50),
but instead of the Succesive Over Relaxation method,
use Mathematica's predefined NSove to find the solution
 of the system approximating the given Dirichlet problem.
        3.2  Compare the result with the exact solution. *)
Clear["`*"];
a = 0; b = π/2; c = 0; d = π;
f[x_, y_] = 0;
m = 30; n = 50;
h = (b - a)/m; k = (d - c)/n; r = k/h;
(* Initial conditions: *)
f_l[y_] = Sin[y]; (* left *) f_r[y_] = 0; (* right *)
f_t[x_] = 0; (* top *)  f_b[x_] = Sin[2 * x]; (* bottom *)
For[i = 0, i ≤ m, i++,
 x_i = a + i * h; u_i,0 = f_b[x_i]; u_i,n = f_t[x_i]];
For[j = 0, j ≤ n, j++,
  y_j = c + j * k; u_0,j = f_l[y_j]; u_m,j = f_r[y_j]];
(* 3.1 *) (* Unknowns: *)
Nec = Flatten[Table[u_i,j, {i, 1, m - 1}, {j, 1, n - 1}], 1];
Eq =
  Flatten[Table[r^2 * (u_i-1,j + u_i+1,j) + u_i,j-1 + u_i,j+1 - 2 * (r^2 + 1) * u_i,j == r^2 * h^2 * f[x_i, y_j],
    {i, 1, m - 1}, {j, 1, n - 1}], 1];
sol = NSolve[Eq, Nec];
solut = First[Nec /. sol];
(* the approximated solution: *)
Unum = Table[0, {i, 0, m}, {j, 0, n}];
Unum[[1]] = Table[u_0,j, {j, 0, n}]; Unum[[m + 1]] = Table[u_m,j, {j, 0, n}];
For[i = 2, i ≤ m, i++, Unum[[i]] =
   Join[{u_i-1,0}, Table[solut[[j]], {j, n * (i - 2) - (i - 3), (i - 1) * (n - 1)}], {u_i-1,n}]];
Print["The approximated solution is: "];
Print["Unum= ", N[Unum // MatrixForm]];
Print["The numerical solution to the P.D.E."];
Print["∂x,x u(x,y)+∂y,y u(x,y)=0"];
ListPlot3D[Transpose[Unum], AxesLabel → {"y", "x", "u   "},
  ViewPoint → {4, 2, 3}, Lighting → False, ColorFunction → Hue];
```

```
(* 3.2 *) (* The exact solution fex*)
fex[x_, y_] = Sinh[π/2 - x]/Sinh[π/2] * Sin[y] + Sinh[2 * (π - y)]/Sinh[2 * π] * Sin[2 * x];
Uex = Table[fex[xi, yj], {i, 0, m}, {j, 0, n}];
Print["The exact solution is: "];
Print["Uex= ", N[Uex // MatrixForm]];
Print["The largest error is: ", Max[Abs[Unum - Uex]]];

(* Task4: 4.1  Solve the problem with Neumann and Dirichlet conditions:
    Δu=1 with u(0,y)=u(x,0)=0; ∂xu(π/2,y)=∂yu(x,π)=0, in the domain Ω=(0,π/2)x(0,π).
        4.2  Compare the result with the exact solution.  *)
Clear["`*"];
a = 0; b = π/2; c = 0; d = π;
f[x_, y_] = 1;
m = 6; n = 12;
h = (b - a)/m; k = (d - c)/n; r = k/h;
(* Set of points: *)
For[i = 0, i ≤ m, i++, xi = a + i * h;];
For[j = 0, j ≤ n, j++, yj = c + j * k;];

(* 4.1 *) (* Unknowns: *)
Nec = Flatten[Table[ui,j, {i, 0, m}, {j, 0, n}], 1];
Eq =
  Flatten[Table[r^2 * (ui-1,j + ui+1,j) + ui,j-1 + ui,j+1 - 2 * (r^2 + 1) * ui,j == r^2 * h^2 * f[xi, yj],
      {i, 1, m - 1}, {j, 1, n - 1}], 1] ⋃
   Table[ui,0 == 0, {i, 0, m}] ⋃ Table[u0,j == 0, {j, 1, n}] ⋃
   Table[um-2,j - 4 * um-1,j + 3 * um,j == 0, {j, 1, n}] ⋃
   Table[ui,n-2 - 4 * ui,n-1 + 3 * ui,n == 0, {i, 1, m - 1}];
sol = NSolve[Eq, Nec];
solut = First[Nec /. sol];

(* the approximated solution: *)
Unum = Table[0, {i, 0, m}, {j, 0, n}];
Unum[[1]] = Table[u0,j, {j, 0, n}]; Unum[[m + 1]] = Table[um,j, {j, 0, n}];
For[i = 1, i ≤ m + 1, i++, Unum[[i]] = Table[solut[[j]], {j, n * (i - 1) + i, i * (n + 1)}]];
Print["The approximated solution is: "];
Print["Unum= ", N[Unum // MatrixForm]];

Print["The numerical solution to the P.D.E."];
Print["∂x,xu(x,y)+∂y,yu(x,y)=0"];
ListPlot3D[Transpose[Unum], AxesLabel → {"y", "x", "u   "},
  ViewPoint → {4, 2, 3}, Lighting → False, ColorFunction → Hue];

(* 4.2 *) (* The exact solution fex*)
fex[x_, y_] = -4/π * Sum[(1/(2 * l + 1)^3 * (1 - Cosh[(2 * l + 1) * (π - y)]/Cosh[(2 * l + 1) * π]) * Sin[(2 * l + 1) * x]), {l, 0, 100}];
Uex = Table[fex[xi, yj], {i, 0, m}, {j, 0, n}];
Print["The exact solution is: "];
Print["Uex= ", N[Uex // MatrixForm]];
Print["The largest error is: ", Max[Abs[Unum - Uex]]];
```