```
Clear["Global`*"]
```

Problem 4 placed here for the use of problem 7.

4. Solve the mixed boundary value problem for the Laplace equation $\nabla^2 = 0$ in the rectangle in figure 458a (using the grid in figure 458b) and the boundary conditions $u_x = 0$ on the left edge, $u_x = 3$ on the right edge, $u = x^2$ on the lower edge, and $u = x^2 - 1$ on the upper edge.

5. Do example 1 in the text for the Laplace equation (instead of the Poisson equation) with grid and boundary data as before.

My answers are not matching those of the text. What I do is to get interpolating functions for both the Laplace and Poisson functions, using the boundary conditions in example 1 on p. 932. The interpolating functions successfully reproduce all the boundary points, and create function points on the interior of the domain rectangle that are plausible and believable, though not matching the text answer. I think the u_{11}, u_{12} and so on which the text investigates are actually components of the linear solving process and not actually interior points. Either that or the solution is not unique.

The interpolating function uso is the function corresponding to the Laplace problem. I first used the function **NDSolveValue**, but I found that **NDSolve** was more cooperative in plotting.

```
eqn = D[u[x, y], {x, 2}] + D[u[x, y], {y, 2}] == 0;

bc = {u[0, y] == 0, u[1.5, y] == 3 y^3, u[x, 0] == 0, u[x, 1] == 6 x};
uso =
 First[u /. NDSolve[{eqn, bc}, u, {x, 0, 1.5}, {y, 0, 1}, PrecisionGoal → 8,
    AccuracyGoal → 8, WorkingPrecision → 10, MaxSteps → ∞]]
```

```
InterpolatingFunction[ Domain: {{0., 1.5}, {0., 1.}}
                       Output: scalar ]
```

The following grid gives the potentials at four named points. The last two columns show I missed the text answer by a lot.

```
Grid[{{"P", "x", "y", "uso", "text ans"}, {"P10", 0.5, 0, uso[0.5, 0]},
  {"P20", 1, 0, uso[1, 0]}, {"P11", 0.5, 0.5, uso[0.5, 0.5], 0.766},
  {"P21", 1, 0.5, uso[1, 0.5], 1.109}, {"P12", 0.5, 1, uso[0.5, 1], 1.957},
  {"P22", 1, 1, uso[1, 1], 3.293}}, Frame → All]
```

P	x	y	uso	text ans
P_{10}	0.5	0	-1.52656×10^{-16}	
P_{20}	1	0	-5.55112×10^{-17}	
P_{11}	0.5	0.5	1.29415	0.766
P_{21}	1	0.5	1.97798	1.109
P_{12}	0.5	1	3.	1.957
P_{22}	1	1	6.	3.293

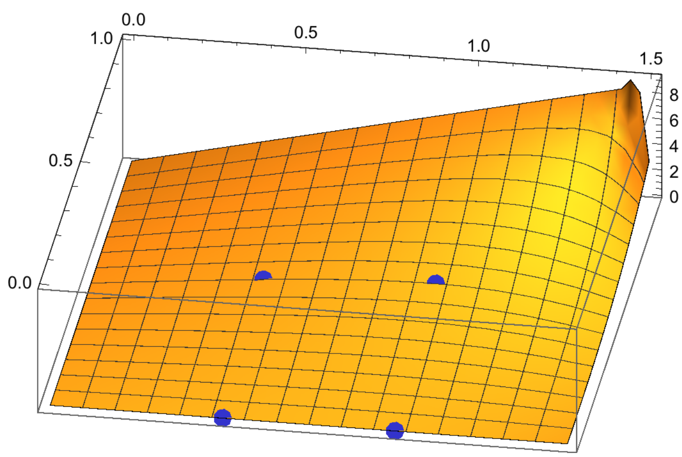
```
ua = Graphics3D[{{RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{0.5, 0, uso[0.5, 0]}]},
    {RGBColor[0.2, 0.2, 0.8], PointSize[0.03], Point[{1, 0, uso[1, 0]}]},
    {RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
     Point[{0.5, 0.5, uso[0.5, 0.5]}]}, {RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{1, 0.5, uso[1, 0.5]}]}}, ImageSize → 450];
```

The following plot shows the Laplace interpolating function with above gridded points embedded.

```
(*usm=ListPlot3D[
  Table[uso[x,y],{x,0,1.5,0.01},{y,0,1,0.02}],ImageSize→250];*)

usol = Plot3D[uso[x, y], {x, 0, 1.5}, {y, 0, 1}, ImageSize → 350,
  AspectRatio → 0.67, ViewPoint → {4000, -7000, 6000.}];

Show[usol, ua]
```



Next, the Poisson 2D wave equation.

```
eqnp = D[u[x, y], {x, 2}] + D[u[x, y], {y, 2}] == 12 x y;
```

The interpolating function usop is the function corresponding to the Poisson problem, the one with which the original text example, p. 931, was concerned.

```
usop = First[
  u /. NDSolve[{eqnp, bc}, u, {x, 0, 1.5}, {y, 0, 1}, PrecisionGoal → 8,
    AccuracyGoal → 8, WorkingPrecision → 10, MaxSteps → ∞]]
```

```
InterpolatingFunction[ Domain: {{0., 1.5}, {0., 1.}} Output: scalar ]
```

The grid of named points corresponding to the one above. I don't bother to **Chop** here.

```
Grid[{{"P", "x", "y", "usop"}, {"P10", 0.5, 0, usop[0.5, 0]},
  {"P20", 1, 0, usop[1, 0]}, {"P11", 0.5, 0.5, usop[0.5, 0.5]},
  {"P21", 1, 0.5, usop[1, 0.5]}}, Frame → All]
```

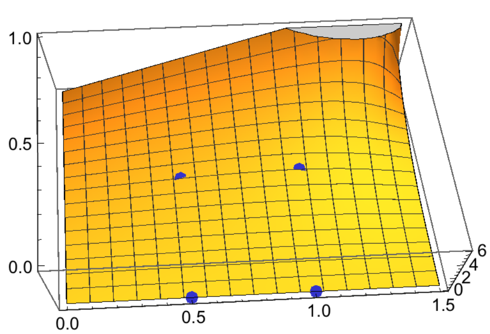
P	x	y	usop
P_{10}	0.5	0	-2.77556×10^{-17}
P_{20}	1	0	-9.71445×10^{-17}
P_{11}	0.5	0.5	0.967157
P_{21}	1	0.5	1.46853

```
uap = Graphics3D[{{RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
     Point[{0.5, 0, usop[0.5, 0]}]}, {RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{1, 0, usop[1, 0]}]},
    {RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
     Point[{0.5, 0.5, usop[0.5, 0.5]}]}, {RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{1, 0.5, usop[1, 0.5]}]}}, ImageSize → 450];
```

The following plot shows the Poisson interpolating function with embedded points.

```
usp = Plot3D[usop[x, y], {x, 0, 1.5},
   {y, 0, 1}, ImageSize → 250, AspectRatio → 0.67];
```

```
Show[usp, uap]
```



7. Solve problem 4 when $u_n = 110$ on the upper edge and u = 110 on the other edges.

This is presented in a way that makes it very similar to the problem I was just working on. I will confine myself to the Laplace case.

```
eqn = D[u[x, y], {x, 2}] + D[u[x, y], {y, 2}] == 0;
bc = {u[0, y] == 110, u[1.5, y] == 110, u[x, 0] == 110, u[x, 1] == 110};
us4 =
 First[u /. NDSolve[{eqn, bc}, u, {x, 0, 1.5}, {y, 0, 1}, PrecisionGoal → 8,
    AccuracyGoal → 8, WorkingPrecision → 10, MaxSteps → ∞]]
```

```
InterpolatingFunction[ Domain: {{0., 1.5}, {0., 1.}}
                       Output: scalar ]
```

Or, I could write out the boundary conditions specifically, like

```
bc1 = {DirichletCondition[u[x, y] == 110, y == 0 && 0 ≤ x ≤ 1.5],
   DirichletCondition[u[x, y] == 110, y == 1 && 0 ≤ x ≤ 1.5],
   DirichletCondition[u[x, y] == 110, x == 0 && 0 ≤ y ≤ 1],
   DirichletCondition[u[x, y] == 110, x == 0 && 0 ≤ y ≤ 1]};
```

In which case I would get the same interpolating function.

```
us41 = First[
  u /. NDSolve[{eqn, bc1}, u, {x, 0, 1.5}, {y, 0, 1}, PrecisionGoal → 8,
    AccuracyGoal → 8, WorkingPrecision → 10, MaxSteps → ∞]]
```

```
InterpolatingFunction[ Domain: {{0., 1.5}, {0., 1.}}
                       Output: scalar ]
```

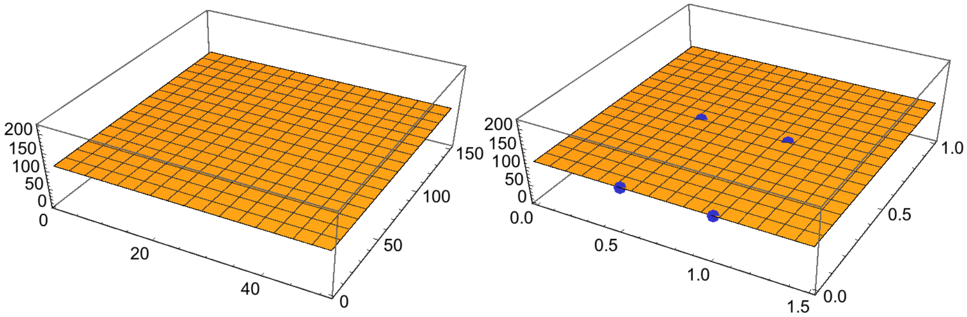
Since all the edge boundaries are equal, there is no variation. It looks like the derivative is zero everywhere. And the method of putting in boundary conditions did not make any difference to the appearance of the function.

```
uh = Graphics3D[{{RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
     Point[{0.5, 0, us41[0.5, 0]}]}, {RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{1, 0, us41[1, 0]}]},
    {RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
     Point[{0.5, 0.5, us41[0.5, 0.5]}]}, {RGBColor[0.2, 0.2, 0.8],
     PointSize[0.03], Point[{1, 0.5, us41[1, 0.5]}]}}];

uson = ListPlot3D[Table[us4[x, y], {x, 0, 1.5, 0.01}, {y, 0, 1, 0.02}],
   ImageSize → 250, AspectRatio → 0.67];

usp41 = Plot3D[us41[x, y], {x, 0, 1.5},
   {y, 0, 1}, ImageSize → 250, AspectRatio → 0.67];
```

```
Row[{Show[uson], Show[usp41, uh]}]
```



Here is a grid for the Poisson case, showing the potentials of the blue balls above.

```
Grid[{{"P", "x", "y", "us41"}, {"P10", 0.5, 0, us41[0.5, 0]},
  {"P20", 1, 0, us41[1, 0]}, {"P11", 0.5, 0.5, us41[0.5, 0.5]},
  {"P22", 1, 0.5, us41[1, 0.5]}}, Frame → All]
```

P	x	y	us41
P_{10}	0.5	0	110.
P_{20}	1	0	110.
P_{11}	0.5	0.5	110.
P_{22}	1	0.5	110.

Nothing in the above problem matches the text answer. However, the text answer explains how the answer is arrived at. To get the text answer, I am advised to use the same coefficient matrix as that used in example 1, p. 932, i.e. aa below. And to use the vector bb below. This does indeed produce the text answers.

$$aa = \begin{pmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 2 & 0 & -4 & 1 \\ 0 & 2 & 1 & -4 \end{pmatrix};$$

$$bb = \begin{pmatrix} -220 \\ -220 \\ -220 \\ -220 \end{pmatrix};$$

```
LinearSolve[aa, bb]
```

$$\left\{\left\{\frac{880}{7}\right\}, \left\{\frac{880}{7}\right\}, \left\{\frac{1100}{7}\right\}, \left\{\frac{1100}{7}\right\}\right\}$$

```
N[%]
{{125.714}, {125.714}, {157.143}, {157.143}}
```

8 - 16 Irregular boundary

13. Solve the Laplace equation in the region and for the boundary values shown in figure 463, using the indicated grid. (The sloping portion of the boundary is y = 4.5 - x.)

```
Solve[3 == 4.5 - x, x]
{{x → 1.5}}

Solve[3 == 4.5 - y, y]
{{y → 1.5}}

d1 = RegionDifference[Rectangle[{0, 0}, {3, 3}],
  Polygon[{{3, 1.5`}, {3, 3}, {1.5`, 3}, {3, 1.5`}}]]
RegionDifference[Rectangle[{0, 0}, {3, 3}],
 Polygon[{{3, 1.5}, {3, 3}, {1.5, 3}, {3, 1.5}}]]

soll = First[u /. NDSolve[{D[u[x, y], x, x] + D[u[x, y], y, y] == 0,
      DirichletCondition[u[x, y] == 3 x, y == 0 && 0 ≤ x ≤ 3],
      DirichletCondition[u[x, y] == 0, x == 0 && 0 ≤ y ≤ 3],
      DirichletCondition[u[x, y] == 0, y == 3 && 0 ≤ x ≤ 1.5],
      DirichletCondition[u[x, y] == 9 - 3 y, x == 3 && 0 ≤ y ≤ 1.5],
       DirichletCondition[u[x, y] == x^2 - 1.5 x, x ≤ 4.5 - y]},
     u, {x, y} ∈ d1, PrecisionGoal → 16, AccuracyGoal → 16,
     WorkingPrecision → 18, MaxSteps → ∞]]

InterpolatingFunction[ Domain: {{0., 3.}, {0., 3.}}  Output: scalar ]
```

The following grid gives the potentials at the four interior points.

```
Grid[{{"P", "x", "y", "soll"},
  {"P11", 1, 1, soll[1, 1]}, {"P21", 2, 1, soll[2, 1]},
  {"P12", 1, 2, soll[1, 2]}, {"P22", 2, 2, soll[2, 2]}}, Frame → All]
```

P	x	y	soll
P_{11}	1	1	2.
P_{21}	2	1	4.
P_{12}	1	2	1.
P_{22}	2	2	2.

The soll column above matches the answers in the text.

Showing a plot with embedded P_{11}, P_{21}, P_{12}, and P_{22}.

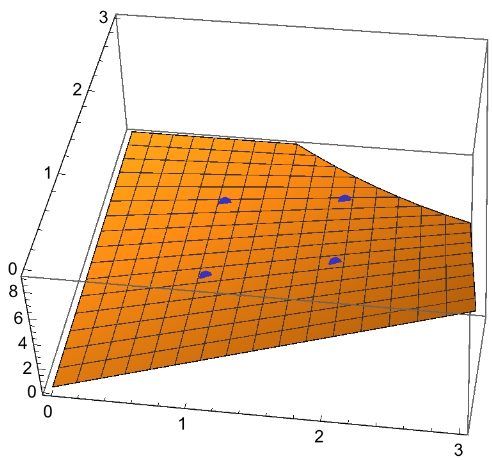
```
gp = Plot3D[soll[x, y], {x, y} ∈ d1, ImageSize → 250];
```

First a random test point.

```
soll[.5, 1]
1.
```

```
gh = Graphics3D[{{RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
      Point[{1, 1, soll[1, 1]}]}, {RGBColor[0.2, 0.2, 0.8],
      PointSize[0.03], Point[{1, 2, soll[1, 2]}]},
     {RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
      Point[{2, 1, soll[2, 1]}]}, {RGBColor[0.2, 0.2, 0.8],
      PointSize[0.03], Point[{2, 2, soll[2, 2]}]}}];
```

```
Show[gp, gh]
```



15. What potential do we have in problem 13 if u = 100 V on the axes and u = 0 on the other portion of the boundary?

This will look much like the last problem.

```
d1 = RegionDifference[Rectangle[{0, 0}, {3, 3}],
   Polygon[{{3, 1.5`}, {3, 3}, {1.5`, 3}, {3, 1.5`}}]]
```

```
RegionDifference[Rectangle[{0, 0}, {3, 3}],
 Polygon[{{3, 1.5}, {3, 3}, {1.5, 3}, {3, 1.5}}]]
```

Ready for calculating the solution function. The boundaries with zero value cannot be left blank, or the answer will be incorrect.

```
sol15 = First[u /. NDSolve[{D[u[x, y], x, x] + D[u[x, y], y, y] == 0,
      DirichletCondition[u[x, y] == 100, y == 0 && 0 ≤ x ≤ 3],
      DirichletCondition[u[x, y] == 100, x == 0 && 0 ≤ y ≤ 3],
      DirichletCondition[u[x, y] == 0, y == 3 && 0 ≤ x ≤ 1.5],
      DirichletCondition[u[x, y] == 0, x == 3 && 0 ≤ y ≤ 1.5],
       DirichletCondition[u[x, y] == 0, x ≤ 4.5 - y]}, u,
     {x, y} ∈ d1, PrecisionGoal → 16, AccuracyGoal → 16,
     WorkingPrecision → 18, MaxSteps → ∞]]
```

```
InterpolatingFunction[ Domain: {{0., 3.}, {0., 3.}}  Output: scalar ]
```

The following grid gives the potentials at the four interior points. These potentials do not quite agree with the text answers, which are included.

```
Grid[
 {{"P", "x", "y", "sol15", "text ans"}, {"P₁₁", 1, 1, sol15[1, 1], 73.68},
  {"P₂₁", 2, 1, sol15[2, 1], 47.37}, {"P₁₂", 1, 2, sol15[1, 2], 47.37},
  {"P₂₂", 2, 2, sol15[2, 2], 15.79}}, Frame → All]
```

P	x	y	sol15	text ans
P_{11}	1	1	74.6422	73.68
P_{21}	2	1	47.3269	47.37
P_{12}	1	2	47.3292	47.37
P_{22}	2	2	15.1287	15.79

Showing a plot with embedded P_{11}, P_{21}, P_{12}, and P_{22}.

```
gq = Plot3D[sol15[x, y], {x, y} ∈ d1, ImageSize → 250];

gi = Graphics3D[{{RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
      Point[{1, 1, sol15[1, 1]}]}, {RGBColor[0.2, 0.2, 0.8],
      PointSize[0.03], Point[{1, 2, sol15[1, 2]}]},
     {RGBColor[0.2, 0.2, 0.8], PointSize[0.03],
      Point[{2, 1, sol15[2, 1]}]}, {RGBColor[0.2, 0.2, 0.8],
      PointSize[0.03], Point[{2, 2, sol15[2, 2]}]}}];

Show[gq, gi]
```