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

1 - 3 Vibrating string
Using the present method, solve numbered lines (1) through (4) with h = k = 0.2 for the given initial deflection f[x] and initial velocity 0 on the given t-interval.

1. $f[x\_] = \begin{cases} x & 0 \leq x < \frac{1}{5} \\ \frac{1}{4}(1-x) & \frac{1}{5} \leq x \leq 1 \;\&\&\; 0 \leq t \leq 1 \end{cases}$

Hyperbolic means, iconically, the wave equation, and I should convey numbered lines (1), (2), (3), and (4) on page 942.

$u_{tt} = u_{xx}$ for $0 \leq x \leq 1$, $t \geq 0$

u[x, 0] = f[x] The given displacement

$u_t[x, 0] = g[x]$ The given initial velocity

u[0, t] = u[1, t] = 0 Boundary conditions.

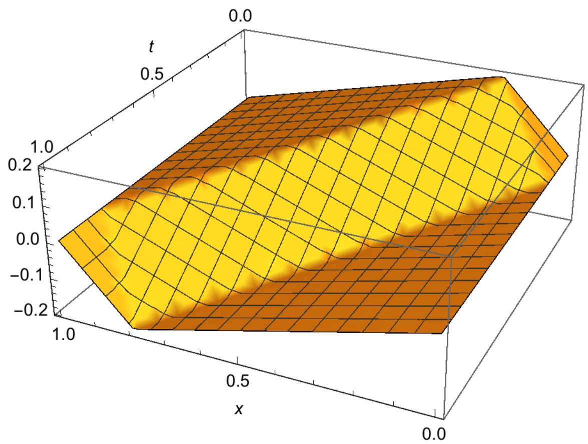
```
Clear["Global`*"]
```

I am slowly learning some things about NDSolve. One place I picked up pointers was at *https://mathematica.stackexchange.com/questions/128516/numerically-solve-the-initial-value-problem-for-the-1-d-wave-equation*. The answer by user21. MaxCellMeasure and MaxStepFraction should be helpful in making sure the landscape of the problem is recognized and mapped sufficiently well. In this case the MaxCellMeasure shown was the smallest I could use, because reducing it further caused an internal error test to fail.

```
uif = NDSolveValue[{D[u[x, t], t, t] == D[u[x, t], x, x],
   u[x, 0] == Piecewise[{{x, 0 ≤ x < 0.2}, {1/4 (1 - x), 0.2 ≤ x ≤ 1}}],
   (D[u[x, t], t] /. t → 0) == 0, u[1, t] == u[0, t] == 0}, u, {t, 0, 1},
  {x, 0, 1}, Method → {"MethodOfLines", "SpatialDiscretization" →
     {"FiniteElement", "MeshOptions" → {"MaxCellMeasure" → 0.0015}}},
  AccuracyGoal → 9, WorkingPrecision → MachinePrecision,
  MaxStepFraction → 0.001]
```

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

```
Plot3D[uif[x, t], {x, 0, 1}, {t, 0, 1},
 AxesLabel → Automatic, ImageSize → 300]
```




```
tableW = TableForm[Table[uif[h, k], {h, 0, 1, 0.1}, {k, 0, 1, 0.2}]]
```

| | | | | | |
|---|---|---|---|---|---|
| 0. | 0. | 0. | 0. | 0. | 0. |
| 0.1 | 0.0375019 | -0.0249958 | -0.0250006 | -0.0249999 | -0.025 |
| 0.19991 | 0.0750035 | -0.0497648 | -0.049999 | -0.0500044 | -0.050 |
| 0.175 | 0.112503 | -0.0124984 | -0.0749995 | -0.0749994 | -0.075 |
| 0.15 | 0.149802 | 0.0250026 | -0.0997484 | -0.0999992 | -0.100 |
| 0.125 | 0.124998 | 0.0624973 | -0.0624977 | -0.125 | -0.124 |
| 0.1 | 0.1 | 0.0997669 | -0.0249997 | -0.149736 | -0.149 |
| 0.075 | 0.075 | 0.0749984 | 0.0125029 | -0.1125 | -0.175 |
| 0.05 | 0.05 | 0.0499978 | 0.0497447 | -0.0750008 | -0.199 |
| 0.025 | 0.025 | 0.0249993 | 0.0250015 | -0.037497 | -0.099 |
| 0. | 0. | 0. | 0. | $1.0842 \times 10^{-19}$ | -2.168 |

In the right most column of the above table the approximations to the text’s 0, -0.05, -0.10, -0.15, and -0.20, and 0 can be seen (on alternating lines, showing S5 to S3).

3. f[x] = 0.2 $(x - x^2)$, $0 \le t \le 2$

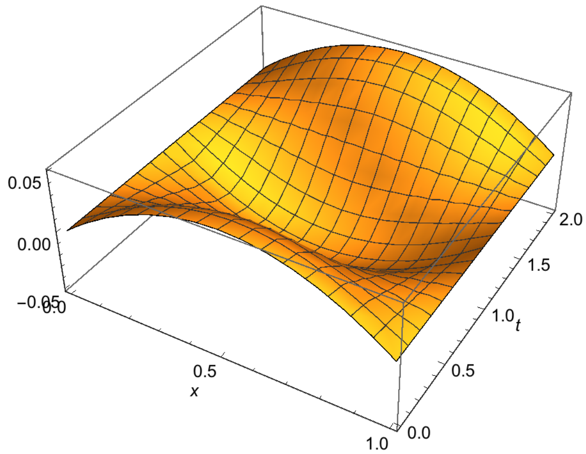
```
Clear["Global`*"]
```

Getting an answer back is pretty slow when I have the MaxStepFraction set so tight, but I like to think I’m keeping the accuracy up.

```
uig = NDSolveValue[{D[u[x, t], t, t] == D[u[x, t], x, x],
    u[x, 0] == 0.2 (x - x^2), (D[u[x, t], t] /. t → 0) == 0, u[1, t] == u[0, t] == 0},
   u, {t, 0, 2}, {x, 0, 1}, AccuracyGoal → 9,
   WorkingPrecision → MachinePrecision, MaxStepFraction → 0.001]
```

```
InterpolatingFunction[ Domain: {{0., 1.}, {0., 2.}} Output: scalar ]
```

```
Plot3D[uig[x, t], {x, 0, 1}, {t, 0, 2},
 AxesLabel → Automatic, ImageSize → 300]
```




```
tableW = TableForm[Table[uig[h, k], {h, 0, 1, 0.2}, {k, 0.2, 0.6, 0.2}]]
```

| | | |
|---|---|---|
| 0. | 0. | 0. |
| 0.0239997 | 0.008 | -0.008 |
| 0.04 | 0.0159996 | -0.0160006 |
| 0.04 | 0.0159996 | -0.0160006 |
| 0.0239997 | 0.008 | -0.008 |
| $-6.06418\times10^{-17}$ | $-4.43147\times10^{-17}$ | $-3.09471\times10^{-17}$ |

The table is off by exactly one order of magnitude compared to the text answer.

5. Nonzero initial displacement and speed. Illustrate the starting procedure when both f and g are not identically zero, say, f[x] = 1 - Cos[2 π x], g[x] = x (1 - x), h = k = 0.1, 2 time steps.

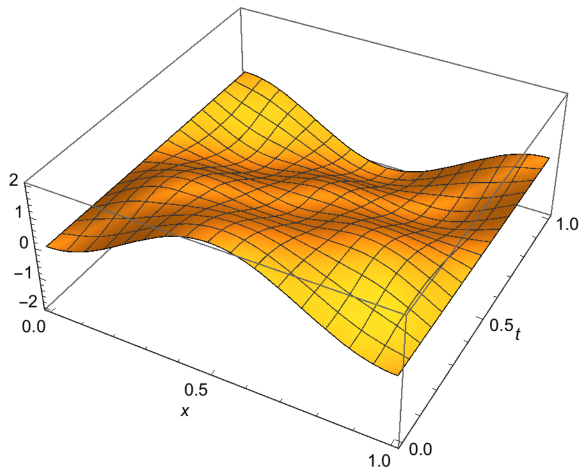
```
Clear["Global`*"]
```

This was a strange one. I started out with MaxStepFraction at only 0.1 to see how it would go, and got a warning of large inaccuracy, which persisted with MSF at 0.01. When returned to 0.001 the problem went away, though execution time was slow, as expected. Then something strange happened. I changed the time from 2 to 1, and the error came back. When I returned it to 2, the error went away again. It doesn't seem logical, unless the step fraction effectively was larger with the larger time. That must be my working assumption.

```
uig =
 NDSolveValue[{D[u[x, t], t, t] == D[u[x, t], x, x], u[x, 0] == 1 - Cos[2 π x],
   (D[u[x, t], t] /. t → 0) == x (1 - x), u[1, t] == u[0, t] == 0},
  u, {t, 0, 2}, {x, 0, 1}, AccuracyGoal → 9,
  WorkingPrecision → MachinePrecision, MaxStepFraction → 0.001]
```

```
InterpolatingFunction[ Domain: {{0., 1.}, {0., 2.}} Output: scalar ]
```

```
Plot3D[uig[x, t], {x, 0, 1}, {t, 0, 1},
 AxesLabel → Automatic, ImageSize → 300]
```




```
tableW = TableForm[Table[uig[h, k], {h, 0, 1, 0.1}, {k, {0.1, 0.2}}]]
```

| | |
|---|---|
| 0. | 0. |
| 0.354172 | 0.574684 |
| 0.765667 | 0.933867 |
| 1.27067 | 1.13482 |
| 1.67818 | 1.29533 |
| 1.83368 | 1.35635 |
| 1.67818 | 1.29533 |
| 1.27067 | 1.13482 |
| 0.765667 | 0.933867 |
| 0.354172 | 0.574684 |
| $-5.55741 \times 10^{-16}$ | $1.71715 \times 10^{-16}$ |

The table looks good, the associated text values in the first column being 0, 0.354, 0.766, 1.271, 1.679, 1.834, and for the second column 0, 0.575, 0.935, 1.135, 1.296, 1.357.

7. Zero initial displacement. If the string governed by the wave equation (1) starts from its equilibrium position with initial velocity g[x] = Sin[π x], what is its displacement at time t = 0.4 and x = 0.2, 0.4, 0.6, 0.8? (Use the present method with h = 0.2, k = 0.2. Use (8). Compare with the exact values obtained from (12) in section 12.4.)

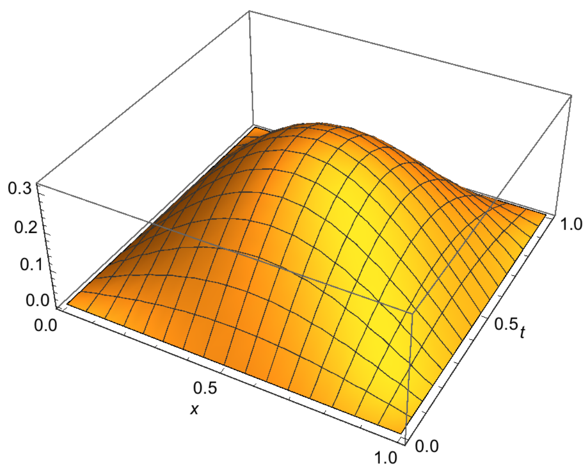
```
Clear["Global`*"]
```

I tried to streamline an equation, putting in enhancers without slowing it down much. The following seems to work pretty well.

```
uig = First[u /. NDSolve[{D[u[x, t], t, t] == D[u[x, t], x, x], u[x, 0] == 0,
      (D[u[x, t], t] /. t → 0) == Sin[π x], u[1, t] == u[0, t] == 0}, u, {t, 0, 1},
     {x, 0, 1}, AccuracyGoal → 12, WorkingPrecision → MachinePrecision,
     MaxStepFraction → 0.001, MaxStepSize → 0.001,
     Method → {"BDF"}]]
```

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

```
Plot3D[uig[x, t], {x, 0, 1}, {t, 0, 1},
 AxesLabel → Automatic, ImageSize → 300]
```




The table has good values, but they do not agree with the worked solution values given in the text answer. The exact 3S answers in the text do agree well. For k = 0.4, these are 0.178, 0.288, 0.288, 0.178, in agreement with the table.

```
tableW = TableForm[Table[uig[h, k], {h, 0, 0.8, 0.2}, {k, 0, 0.4, 0.2}]]
```

| | | |
|---|---|---|
| 0. | 0. | 0. |
| 0. | 0.109971 | 0.177937 |
| 0. | 0.177937 | 0.287908 |
| 0. | 0.177937 | 0.287908 |
| 0. | 0.109971 | 0.177937 |

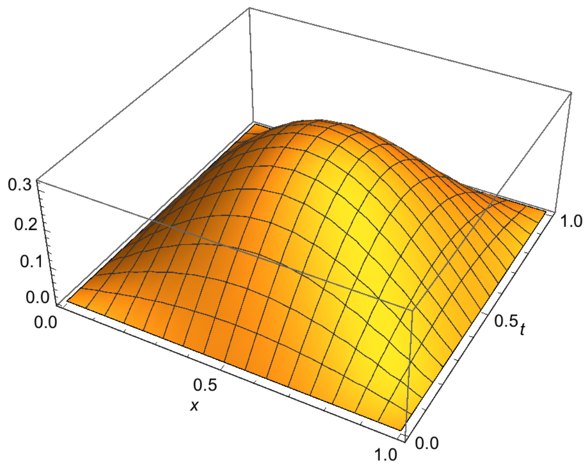
I throw in an extra FEM version, just to demonstrate they come to the same solution values.

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

```
uif = NDSolveValue[{D[u[x, t], t, t] == D[u[x, t], x, x], u[x, 0] == 0,
    (D[u[x, t], t] /. t → 0) == Sin[π x], u[1, t] == u[0, t] == 0}, u, {t, 0, 1},
   {x, 0, 1}, Method → {"MethodOfLines", "SpatialDiscretization" →
      {"FiniteElement", "MeshOptions" → {"MaxCellMeasure" → 0.0015}}},
   AccuracyGoal → 9, WorkingPrecision → MachinePrecision,
   MaxStepFraction → 0.001]
```

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

```
Plot3D[uif[x, t], {x, 0, 1}, {t, 0, 1},
 AxesLabel → Automatic, ImageSize → 300]
```




```
tableW = TableForm[Table[uif[h, k], {h, 0, 0.8, 0.2}, {k, 0, 0.4, 0.2}]]
```

| | | |
|---|---|---|
| 0. | 0. | 0. |
| $-2.9006\times10^{-12}$ | 0.109973 | 0.177936 |
| $-4.69328\times10^{-12}$ | 0.177941 | 0.287907 |
| $-4.69328\times10^{-12}$ | 0.177941 | 0.287907 |
| $-2.9006\times10^{-12}$ | 0.109973 | 0.177936 |