

Vibration of a Rectangular Membrane

```
Manipulate[
Module[{xx, yy,
  opt = {PerformanceGoal -> "Quality", ImageMargins -> 1}, r, r0, c0, a00, b00, color0},
  c0 = Sqrt[tension * 1000 / ( $\rho_0 * 10^6$ )];
  a00 = a0 / 1000; (*mm to meter*)
  b00 = b0 / 1000;
  r = w[x, y, a00, b00, m0, n0, c0, t, t0]; (*saved for tooltip to show analytical form*)
  r0 = r /. {t -> t0, x -> xx, y -> yy}; (*used for plotting*)
  Which[
    color == 1, color0 = Automatic,
    color == 2, color0 = (ColorData["TemperatureMap"])[#3] &),
    color == 3, color0 = Function[{x, y, z}, Hue[z]],
    color == 4, color0 = "BlueGreenYellow",
    color == 5, color0 = "Rainbow",
    color == 6, color0 = Function[{x, y}, ColorData["NeonColors"][y]]
  ];

Grid[{
  {
    Which[plotType == "3D",

      Plot3D[Tooltip[Evaluate@r0,
        Text[Style[TraditionalForm[r], 12]]], {xx, 0, a00}, {yy, 0, b00},
        Evaluate@opt,
        ImageSize -> {350, 436},
        PlotRange -> {{0, a00}, {0, b00}, {-yMax, yMax}},
        Mesh -> meshLines,
        AxesLabel -> {x, y, u},
        PreserveImageOptions -> True,
        SphericalRegion -> True,
        PlotStyle -> Directive[Opacity[opacity]],
        ColorFunction -> color0,
        ImagePadding -> {{25, 5}, {10, 10}}],

      plotType == "contour",

      ContourPlot[Evaluate@r0, {xx, 0, a00}, {yy, 0, b00},
        Evaluate@opt,
        ImageSize -> {350, 436},
        FrameLabel -> {{y, None}, {x, None}},
        ColorFunction -> "Pastel",
        ImagePadding -> 30,
        FrameTicksStyle -> 8,
        Contours -> contoursLines],
```

```

plotType == "both",

Grid[{
  {Plot3D[Tooltip[Evaluate@r0,
    Text[Style[TraditionalForm[r], 12]]], {xx, 0, a00}, {yy, 0, b00},
    Evaluate@opt,
    ImageSize → {350, 210},
    PlotRange → {{0, a00}, {0, b00}, {-yMax, yMax}},
    Mesh → meshLines,
    ImagePadding → {{25, 5}, {10, 10}},
    PreserveImageOptions → True,
    SphericalRegion → True,
    PlotStyle → Directive[Opacity[opacity]],
    AxesLabel → {x, y, u}]
  },
  {
    ContourPlot[Evaluate@r0, {xx, 0, a00}, {yy, 0, b00},
      Evaluate@opt,
      ImageSize → {350, 210},
      FrameLabel → {{y, None}, {x, None}},
      ColorFunction → "Pastel",
      ImagePadding → 30,
      FrameTicksStyle → 8,
      Contours → contoursLines]
  }
], Spacings → {0, 0}, Alignment → Center
]
], SpanFromLeft}
}, Alignment → Center, Spacings → 0
],
Text@Grid[{
  {
    Style[TraditionalForm[Defer[D[u[x, y, t], {x, 2}] + D[u[x, y, t], {y, 2}]] ==
      Defer[HoldForm[1/c^2] D[u[x, y, t], {t, 2}]]], 14], SpanFromLeft
  },
  {
    Grid[{
      {Grid[{
        {
          Row[{Style[
            TraditionalForm[HoldForm[ $\omega_{mn} = c \pi \sqrt{\frac{m^2}{a^2} + \frac{n^2}{b^2}}$ ]], 14], Spacer[3], "(hz)"]}
        },
        {
          Dynamic[getW[Sqrt[tension/ρ0], a0/1000, b0/1000]]
        }
      ]
    }
  }
}

```

```

    }
  }, Spacings → .1
]
},
{
  Dynamic[Grid[
    {
      Row[{Style["c", Italic, 12], Spacer[2], "(m/s)"}],
      Row[{ $\lambda$ , " = ", Style[a/b, Italic]}]],
      {padIt2[Sqrt[tension/ $\rho$ 0], {4, 2}], padIt2[N[a0/b0], {4, 2}]}
    }, Frame → All,
    FrameStyle → Directive[Thickness[.001], Gray], Spacings → {.8, .4}
  ]]
}
}, Alignment → Center, Spacings → {.1, .1}
]
},
{Grid[{
  {
    Grid[{
      {
        Style["T", Italic, 12],
        Manipulator[Dynamic[tension, {tension = #} &], {0.1, 10, 0.1}, ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[tension, {3, 1}]], "N/mm"
      },
      {
        Style[ $\rho$ , Italic, 12],
        Manipulator[Dynamic[ $\rho$ 0, { $\rho$ 0 = #} &], {0.1, 2, 0.1}, ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[ $\rho$ 0, {2, 1}]], "kg/mm2"
      },
      {
        Style["a", Italic, 12],
        Manipulator[Dynamic[a0, {a0 = #} &], {10, 1000, 1}, ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[a0, 4]], "mm"
      },
      {
        Style["b", Italic, 12],
        Manipulator[Dynamic[b0, {b0 = #} &], {10, 1000, 1}, ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[b0, 4]], "mm"
      }
    ], Spacings → {.4, 0}, Alignment → Left
  ]
}
}, Spacings → {.5, .6}, Alignment → Center
], SpanFromLeft
},
{
  Grid[{
    {

```

```

Grid[{
  {Item[Style["modes to excite", 12], Alignment → Center], SpanFromLeft},
  {
    Grid[{
      {Style["m", Italic, 12],
        TogglerBar[Dynamic[m0, {m0 = #} &], Range[5]]
      },
      {Style["n", Italic, 12],
        TogglerBar[Dynamic[n0, {n0 = #} &], Range[5]]
      }
    ], Spacings → {.3, .2}, Alignment → Left
  ]
}, Spacings → {.5, .6}, Alignment → Center
],

Grid[{
  {
    Style["plot type", 10],
    PopupMenu[Dynamic[plotType, {plotType = #} &],
      {
        "3D" → Style["3D plot", 10],
        "contour" → Style["contour plot", 10],
        "both" → Style["3D+contour", 10]
      }, ImageSize -> All
    ]
  },
  {
    Style["3D color", 10],
    PopupMenu[Dynamic[color, {color = #} &],
      {
        1,
        2,
        3,
        4,
        5,
        6
      }, ImageSize -> All
    ]
  }, Spacings → {0.2, 0.1}, Alignment → Left
]

}
], Spacings → {.3, .6}, Alignment → Center,
Frame → True, FrameStyle -> Directive[Thickness[.001], Gray]
]
}
,
{

```

```

Grid[{
  {
    Grid[{
      {
        Text@"animate", Trigger[Dynamic[t0, {t0 = #} &], {0, 10000},
        AnimationRepetitions → Infinity, AnimationRate → 10, ImageSize → Tiny,
        AppearanceElements → {"TriggerButton", "PauseButton", "ResetButton"},
        DisplayAllSteps → True], Dynamic[padIt2[t0, {7, 2}]], " sec"
      },
      {Text@"plot range",
        Manipulator[Dynamic[yMax, {yMax = #} &], {0.1, 15, 0.1}, ImageSize → Small,
        ContinuousAction → True], Dynamic[padIt2[yMax, {3, 1}]], " m"
      },
      {Text@"mesh lines",
        Manipulator[Dynamic[meshLines, {meshLines = #} &], {0, 20, 1},
        ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[meshLines, 2]]
      },
      {Text@"opacity", Manipulator[Dynamic[opacity, {opacity = #} &], {0.01, 1, 0.01},
        ImageSize → Small, ContinuousAction → True], Dynamic[padIt2[opacity, {2, 2}]]
      },
      {
        Text@"contour lines",
        Manipulator[
          Dynamic[contoursLines, {contoursLines = #} &], {1, 16, 1}, ImageSize → Small,
          ContinuousAction → True], Dynamic[padIt2[contoursLines, 2]], ""
        }
    ], Spacings → {.3, .1}, Alignment → Left
  ]
}, Spacings → {.5, .6}, Alignment → Center, Frame → False], SpanFromLeft
]

}, Alignment → Center, Spacings → {0.5, .4},
Frame → True, FrameStyle -> Directive[Thickness[.001], Gray]
],

{{color, 1}, None},
{{tension, 0.1}, None},
{{ρ0, 1}, None},
{{opacity, .8}, None},
{{yMax, 1.6}, None},
{{t0, 0}, None},
{{a0, 100}, None},
{{b0, 100}, None},
{{m0, {1}}, None},
{{n0, {1}}, None},
{{meshLines, 10}, None},
{{plotType, "3D"}, None},
{{contoursLines, 10}, None},

SynchronousUpdating → False,

```

```

Alignment → Center,
ImageMargins → 2, (*important*)
FrameMargins → 1,
SynchronousInitialization → True,
ContinuousAction → False,
Alignment -> Center,
Paneled → True,
Frame → False,
AutorunSequencing → Automatic,

ControlPlacement → Left,
Initialization :->
{
  integerStrictPositive = (IntegerQ[#] && # > 0 &);
  integerPositive = (IntegerQ[#] && # ≥ 0 &);
  numericStrictPositive = (Element[#, Reals] && # > 0 &);
  numericPositive = (Element[#, Reals] && # ≥ 0 &);
  numericStrictNegative = (Element[#, Reals] && # < 0 &);
  numericNegative = (Element[#, Reals] && # ≤ 0 &);
  bool = (Element[#, Booleans] &);
  numeric = (Element[#, Reals] &);
  integer = (Element[#, Integers] &);

  (*-----*)
  (* helper function for formatting *)
  (*-----*)
  padIt1[v_?numeric, f_List] := AccountingForm[Chop[v], f,
    NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*-----*)
  (* helper function for formatting *)
  (*-----*)
  padIt2[v_?numeric, f_List] := AccountingForm[Chop[v], f,
    NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
  padIt2[v_?numeric, f_Integer] := AccountingForm[Chop[v], f,
    NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*-----*)
  w[x_, y_, a_, b_, mm_, nn_, c_, t_, t0_] := Module[{m, n, f},
    Sum[Sum[
      If[MemberQ[mm, m] && MemberQ[nn, n],
        f = N@freq[c, a, b, m, n];
        Sin[ $\frac{m \text{ Pi } x}{a}$ ] Sin[ $\frac{n \text{ Pi } y}{b}$ ] If[t0 == 0, 1, (Cos[f t] + Sin[f t])], 0],
      {m, 1, 5}],
      {n, 1, 5}]
    ];
  (*-----*)

  freq[c_, a_, b_, m_, n_] := Pi c  $\sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2}$ ;
  (*-----*)

```

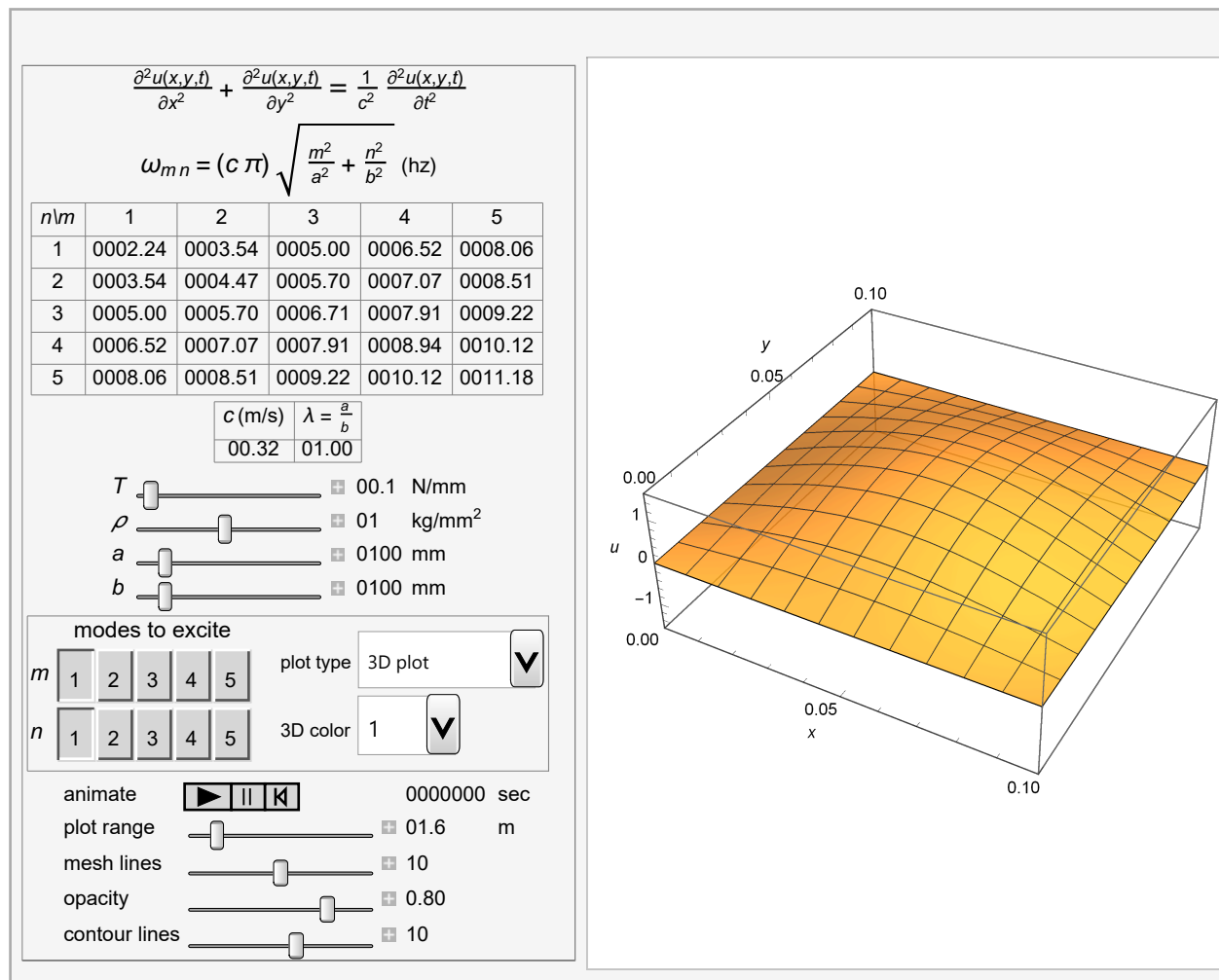
```

getW[c_, a_, b_] := Module[{m, n, g, wmn},
  g = Table[0, {6}, {6}];
  wmn = Table[PadIt2[N@freq[c, a, b, m, n] / (2 * Pi), {6, 2}], {m, 1, 5}, {n, 1, 5}];
  g[[1, All]] = {"", 1, 2, 3, 4, 5};
  g[[All, 1]] = {"", 1, 2, 3, 4, 5};
  g[[2 ;;, 2 ;;]] = wmn;
  g[[1, 1]] = Style["n\\m", Italic];
  Grid[g, Frame -> All,
    FrameStyle -> Directive[Thickness[.001], Gray], Spacings -> {.8, .8}]
];

}

]

```



CAPTION

DETAILS

THIS NOTEBOOK IS THE SOURCE CODE FROM

Nasser M. Abbasi

"Vibration of a Rectangular Membrane"

<http://demonstrations.wolfram.com/VibrationOfARectangularMembrane/>

Wolfram Demonstrations Project

Published: September 5, 2013

A full-function Wolfram *Mathematica* system is required to edit this notebook.

[GET WOLFRAM MATHEMATICA »](#)