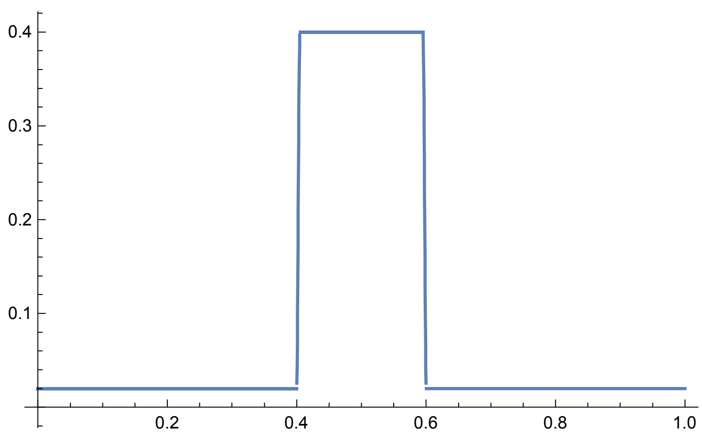
In[1]:=
```
(*Alpha and Epsilon Values*)
alphaRod = 0.4;
alphaSheet = 0.02;
epsilon = 0.1;
```

In[4]:=
```
(*To make the diffusion function continuous and differentiable,
I used a quartic function for the part where the value changes.
  This is because it was easy for me to make one of its local maxima
 my upper value and one of its local minima my lower value. They needed
 to be minima and maxima to ensure that entire function is smooth. *)
BB = 0.01;
initFunc[x_] := x^2 (x - BB) ^2;
maxVal = initFunc[BB / 2];
funcUse[x_] := (1 / maxVal) * (alphaRod - alphaSheet) * initFunc[x] + alphaSheet;
diffConst[x_] := Piecewise[{{alphaSheet, x < 0.5 - epsilon},
    {funcUse[x - (0.5 - epsilon)], x ≥ 0.5 - epsilon && x < 0.5 - epsilon + BB / 2},
    {alphaRod, x ≥ 0.5 - epsilon + BB / 2 && x < 0.5 + epsilon - BB / 2},
    {funcUse[x - (0.5 + epsilon - BB)], x ≥ 0.5 + epsilon - BB / 2 && x < 0.5 + epsilon},
    {alphaSheet, x ≥ 0.5 + epsilon}}];
```

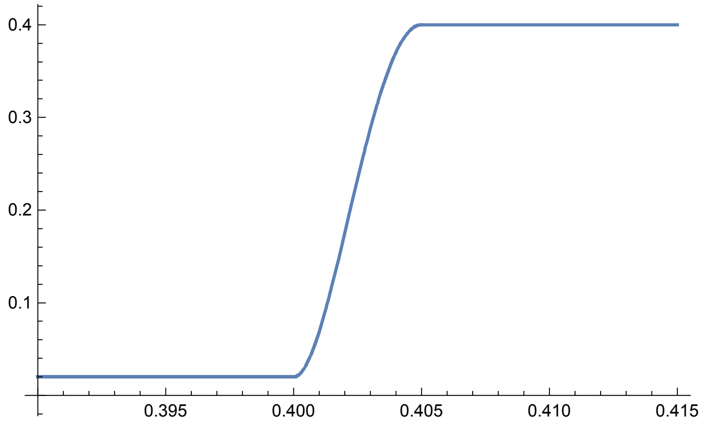
In[9]:=
```
(*Here is the entire function from 0 to 1*)
Plot[diffConst[x], {x, 0, 1}]
```

Out[9]=



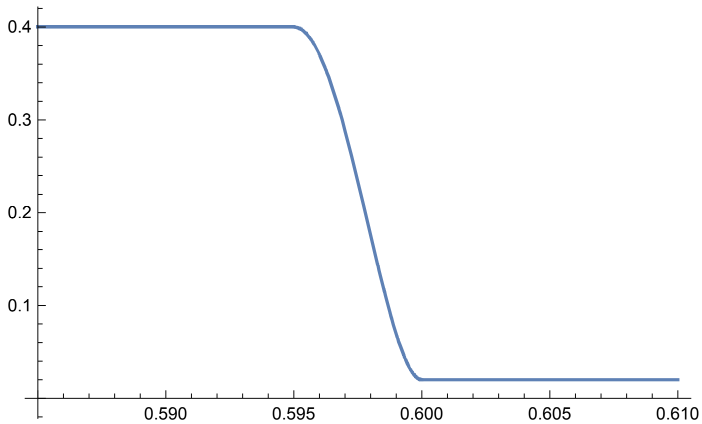
In[10]:=
```
(*Shows the transition at x=
 0.5-epsilon to verify smoothness at that transition neighborhood*)
Plot[diffConst[x], {x, 0.5 - epsilon - epsilon / 10, 0.5 - epsilon + 1.5 * epsilon / 10}]
```

Out[10]=



In[11]:=
```
(*Shows the transition at x=
 0.5+epsilon to verify smoothness at that transition neighborhood*)
Plot[diffConst[x], {x, 0.5 + epsilon - 1.5 * epsilon / 10, 0.5 + epsilon + epsilon / 10}]
```

Out[11]=



In[12]:=
```
(*This is my unit square system that features a rod in the middle of it*)
```

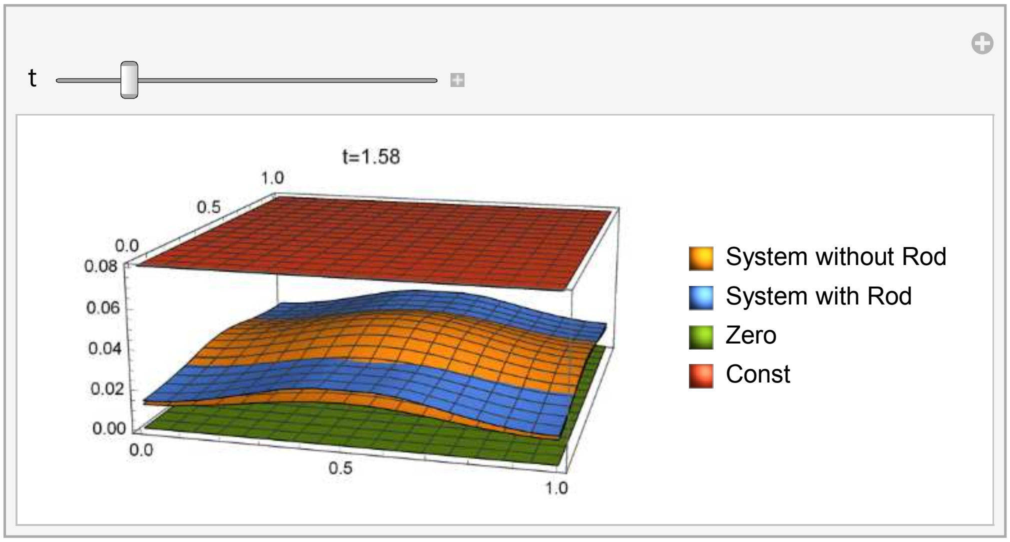
In[13]:=
```
initF[x_, y_] := 20 * x^2 * (x - 1) ^2 * y^2 * (y - 1) ^2;
systemWithRod =
  NDSolveValue[{D[u[x, y, t], t] - diffConst[x] * Laplacian[u[x, y, t], {x, y}] == 0,
    DirichletCondition[u[x, y, t] == initF[x, y], t == 0]},
   u, {x, 0, 1}, {y, 0, 1}, {t, 0, 10}];
systemNoRod = NDSolveValue[{D[u[x, y, t], t] - alphaSheet * Laplacian[u[x, y, t], {x, y}] ==
     0, DirichletCondition[u[x, y, t] == initF[x, y], t == 0]},
   u, {x, 0, 1}, {y, 0, 1}, {t, 0, 10}];
```

NDSolveValue: The PDE is convection dominated and the result may not be stable. Adding artificial diffusion may help.

NDSolveValue: The PDE is convection dominated and the result may not be stable. Adding artificial diffusion may help.

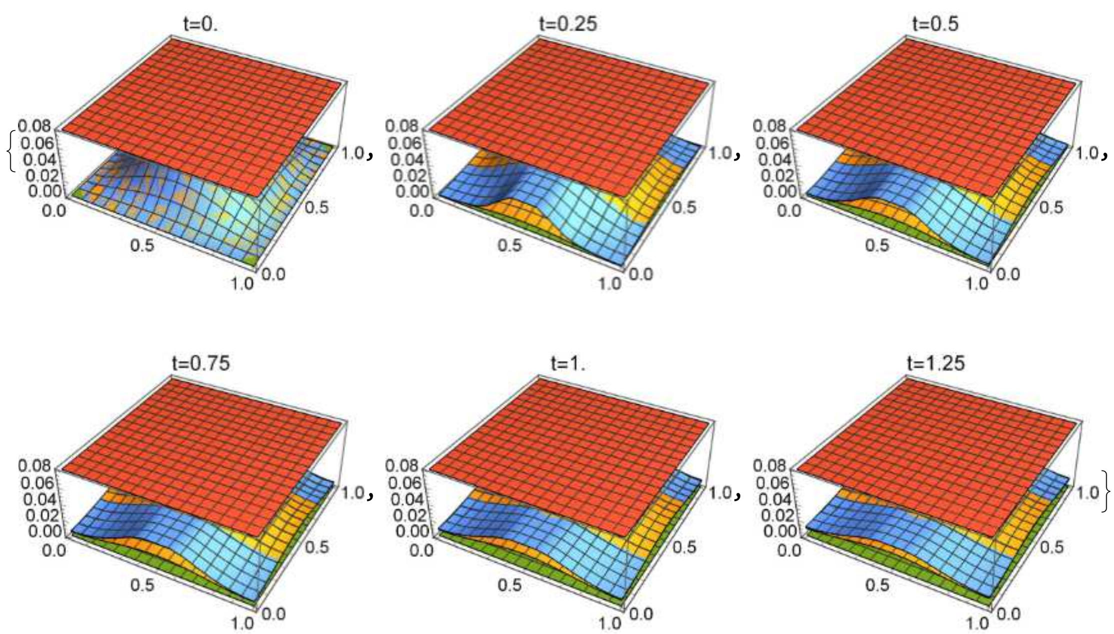
In[41]:=
```
(*3D plot of system*)
Manipulate[Plot3D[{systemNoRod[x, y, t], systemWithRod[x, y, t], 0, 0.08}, {x, 0, 1},
  {y, 0, 1}, PlotLegends → {"System without Rod", "System with Rod", "Zero", "Const"},
  PlotLabel → StringJoin["t=", ToString[t]]], {t, 0, 10}]
```

Out[41]=



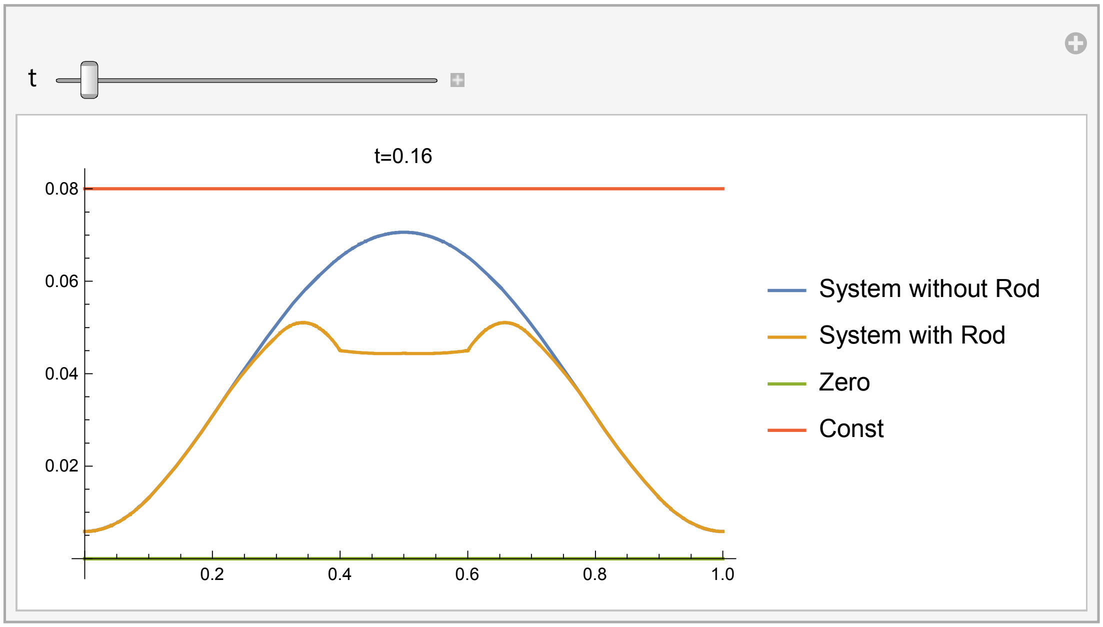
In[45]:=
```
Table[Plot3D[{systemNoRod[x, y, t], systemWithRod[x, y, t], 0, 0.08}, {x, 0, 1},
  {y, 0, 1}, PlotLabel → StringJoin["t=", ToString[t]]], {t, 0, 1.25, 0.25}]
```

Out[45]=



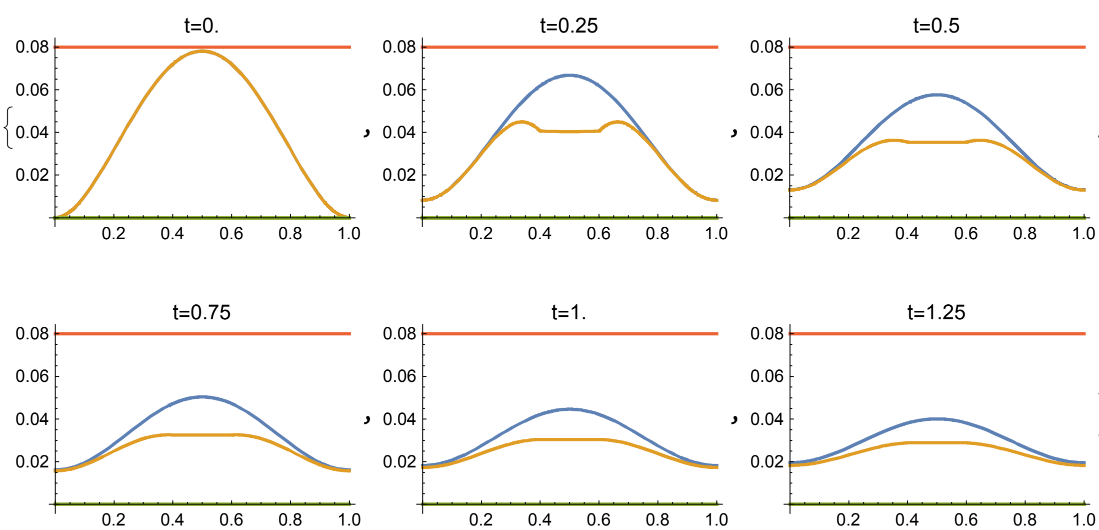
In[42]:=
```
(*2D plot of y=0.5 cross-section*)
Manipulate[Plot[{systemNoRod[x, 0.5, t], systemWithRod[x, 0.5, t], 0, 0.08}, {x, 0, 1},
  PlotLegends → {"System without Rod", "System with Rod", "Zero", "Const"},
  PlotLabel → StringJoin["t=", ToString[t]]], {t, 0, 4}]
```

Out[42]=



In[46]:=
```
Table[Plot[{systemNoRod[x, 0.5, t], systemWithRod[x, 0.5, t], 0, 0.08},
  {x, 0, 1}, PlotLabel → StringJoin["t=", ToString[t]]], {t, 0, 1.25, 0.25}]
```

Out[46]=



In[18]:=
```
(*This verifies conservation of energy for both systems*)
Energy[t_] := NIntegrate[systemWithRod[x, y, t], {x, 0, 1}, {y, 0, 1}];
Table[Energy[t], {t, 0, 5, 1}]
Energy2[t_] := NIntegrate[systemNoRod[x, y, t], {x, 0, 1}, {y, 0, 1}];
Table[Energy2[t], {t, 0, 5, 1}]
```

Out[19]= {0.0222233, 0.0222233, 0.0222233, 0.0222233, 0.0222233, 0.0222233}

Out[21]= {0.0222233, 0.0222233, 0.0222233, 0.0222233, 0.0222233, 0.0222233}

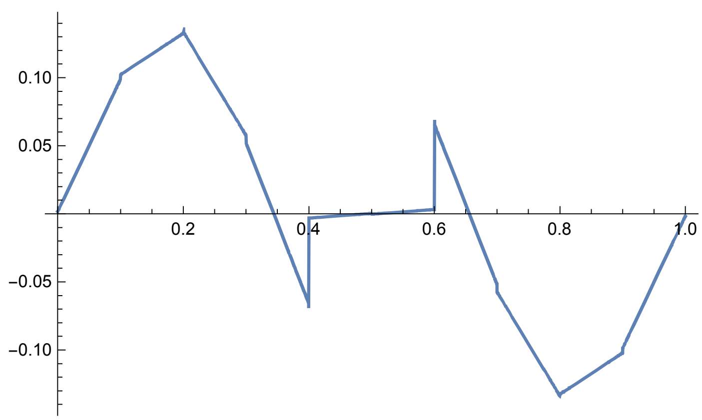
In[22]:=
```
(*Derivatives in X and Y direction so we can obtain the fluxes*)
```

In[23]:=
```
XderivNonUniformDiffusion[x_, y_, t_] :=
  D[systemWithRod[xV, yV, tV], xV] /. {tV → t, xV → x, yV → y};
YderivNonUniformDiffusion[x_, y_, t_] :=
  D[systemWithRod[xV, yV, tV], yV] /. {tV → t, xV → x, yV → y};
XderivUniformDiffusion[x_, y_, t_] :=
  D[systemNoRod[xV, yV, tV], xV] /. {tV → t, xV → x, yV → y};
YderivUniformDiffusion[x_, y_, t_] :=
  D[systemNoRod[xV, yV, tV], yV] /. {tV → t, xV → x, yV → y};
```

In[27]:=
```
(*The rest of the graphs shows the system at a select time point
 (t=0.4 currently) and verifies the properties we are looking for*)
tVV = 0.4;
```

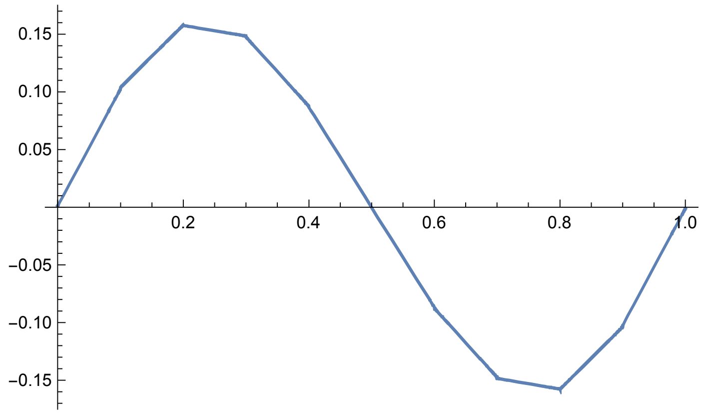
In[28]:=
```
(*Horizontal Diffusion in middle of system for non-uniform case.
    It ends up being discontinuous at x=0.5-epsilon and x=0.5+epsilon*)
Plot[XderivNonUniformDiffusion[x, 0.5, tVV], {x, 0, 1}]
```

Out[28]=



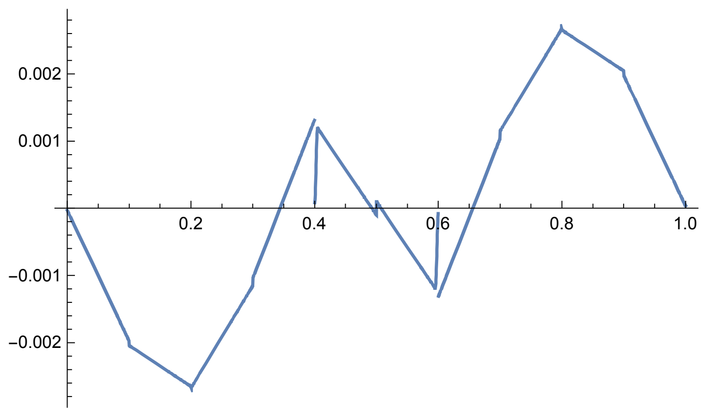
In[29]:=
```
(*Horizontal Diffusion in middle of system for uniform case as comparison*)
Plot[XderivUniformDiffusion[x, 0.5, tVV], {x, 0, 1}]
```

Out[29]=



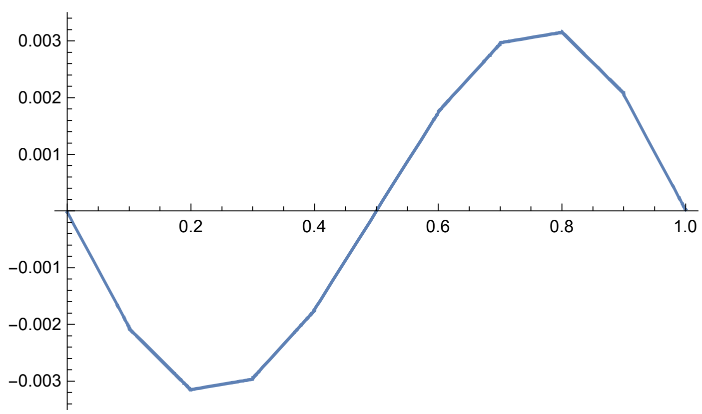
In[30]:=
```
(*Horizontal Flux in middle of system for non-uniform case.
    It seems to be mostly continusous at x=0.5-epsilon and x=0.5+epsilon*)
Plot[-XderivNonUniformDiffusion[x, 0.5, tVV] * diffConst[x], {x, 0, 1}]
```

Out[30]=



In[31]:=
```
(*Horizontal Flux in middle of system for uniform case as comparison*)
Plot[-XderivUniformDiffusion[x, 0.5, tVV] * alphaSheet, {x, 0, 1}]
```

Out[31]=



In[32]:=
```
(*Diffusion at borders for non-uniform case. should be near zero*)
Table[YderivNonUniformDiffusion[x, 1, tVV], {x, 0, 1, .1}]
Table[YderivNonUniformDiffusion[x, 0, tVV], {x, 0, 1, .1}]
Table[XderivNonUniformDiffusion[0, y, tVV], {y, 0, 1, .1}]
Table[XderivNonUniformDiffusion[1, y, tVV], {y, 0, 1, .1}]
```

Out[32]= {-0.00107796, -0.00085832, -0.0000529501, 0.00107793, -0.000138623, -0.000232363, -0.000138623, 0.00107793, -0.0000529501, -0.00085832, -0.00107796}

Out[33]= {0.00107796, 0.00085832, 0.0000529501, -0.00107793, 0.000138623, 0.000232363, 0.000138623, -0.00107793, 0.0000529501, 0.00085832, 0.00107796}

Out[34]= {0.000426179, 0.000523093, 0.000806094, 0.00119876, 0.0015367, 0.0016684, 0.0015367, 0.00119876, 0.000806094, 0.000523093, 0.000426179}

Out[35]= {-0.000426179, -0.000523093, -0.000806094, -0.00119876, -0.0015367, -0.0016684, -0.0015367, -0.00119876, -0.000806094, -0.000523093, -0.000426179}

In[36]:=
```
(*Diffusion at borders for uniform case as comparison. should be near zero*)
Table[YderivUniformDiffusion[x, 1, tVV], {x, 0, 1, .1}]
Table[YderivUniformDiffusion[x, 0, tVV], {x, 0, 1, .1}]
Table[XderivUniformDiffusion[0, y, tVV], {y, 0, 1, .1}]
Table[XderivUniformDiffusion[1, y, tVV], {y, 0, 1, .1}]
```

Out[36]= {-0.0011164, -0.00107476, -0.00101796, -0.00103093, -0.00108928, -0.00111984, -0.00108928, -0.00103093, -0.00101796, -0.00107476, -0.0011164}

Out[37]= {0.0011164, 0.00107476, 0.00101796, 0.00103093, 0.00108928, 0.00111984, 0.00108928, 0.00103093, 0.00101796, 0.00107476, 0.0011164}

Out[38]= {0.0011164, 0.00107476, 0.00101796, 0.00103093, 0.00108928, 0.00111984, 0.00108928, 0.00103093, 0.00101796, 0.00107476, 0.0011164}

Out[39]= {-0.0011164, -0.00107476, -0.00101796, -0.00103093, -0.00108928, -0.00111984, -0.00108928, -0.00103093, -0.00101796, -0.00107476, -0.0011164}