```
xRange = {x, 0, 1000};
tRange = {t, 0, 200};
```

Reaction Term

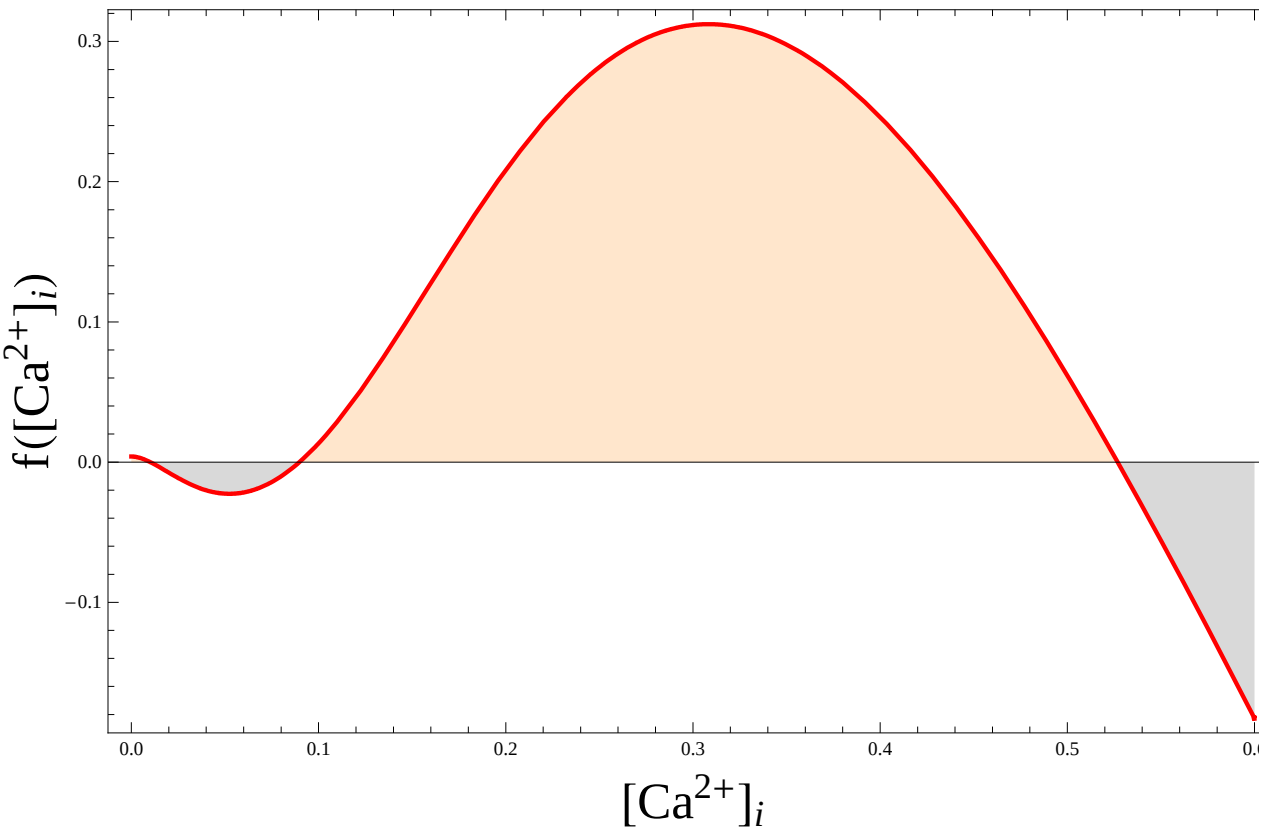
```
Q2 = d2 * (IP3 + d1) / (IP3 + d3);
wInf[c_] := Q2 / (Q2 + c);
mInf[c_] := (IP3 / (IP3 + d1) * c / (c + d5));
VIP3R[c_] := (mInf[c] * wInf[c])^3;
f[c_] := ( fiVLeak + fiVIP3R * VIP3R[c]) * (cER - c) - fiVSERCA * c^2 / (c^2 + KSerca^2);
```

Parameter Values

```
ParamRules = {fiVLeak → 0.004, fiVIP3R → 20, fiVSERCA → 1.2,  cER → 1,
   KSerca → 0.15, IP3 → 0.7, a2 → 0.2, d1 → 0.1, d2 → 2, d3 → 0.2, d5 → 0.2};
```

Plotting Reaction Term

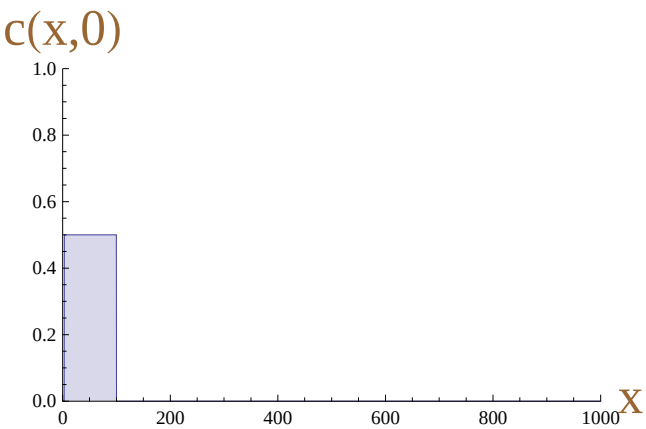
```
Plot[f[c] /. ParamRules, {c, 0, 0.6}, PlotStyle → {Thick, Red},
 Filling → Axis, FillingStyle → {LightGray, LightOrange}, Frame → True,
 FrameLabel → {Style["[Ca2+]i", Large], Style["f([Ca2+]i)", Large]}, ImageSize → {600, 400}]
```



Step Initial Condition

```
StepInputSize = 0.5;
InitialConditionProfile = StepInputSize *
   (1 / 2 + 1 / 2 * Tanh[50 * Sqrt[1 / 2] * (x - 3)] - (1 / 2 + 1 / 2 * Tanh[50 * Sqrt[1 / 2] * (x - 100)]));

Plot[InitialConditionProfile, {x, 0, 1000}, PlotRange → {{0, 1000}, {0, 1}},
 Filling → Bottom,
 AxesLabel → {StyleForm["x", Large, Brown], StyleForm["c(x,0)", Large, Brown]}]
```



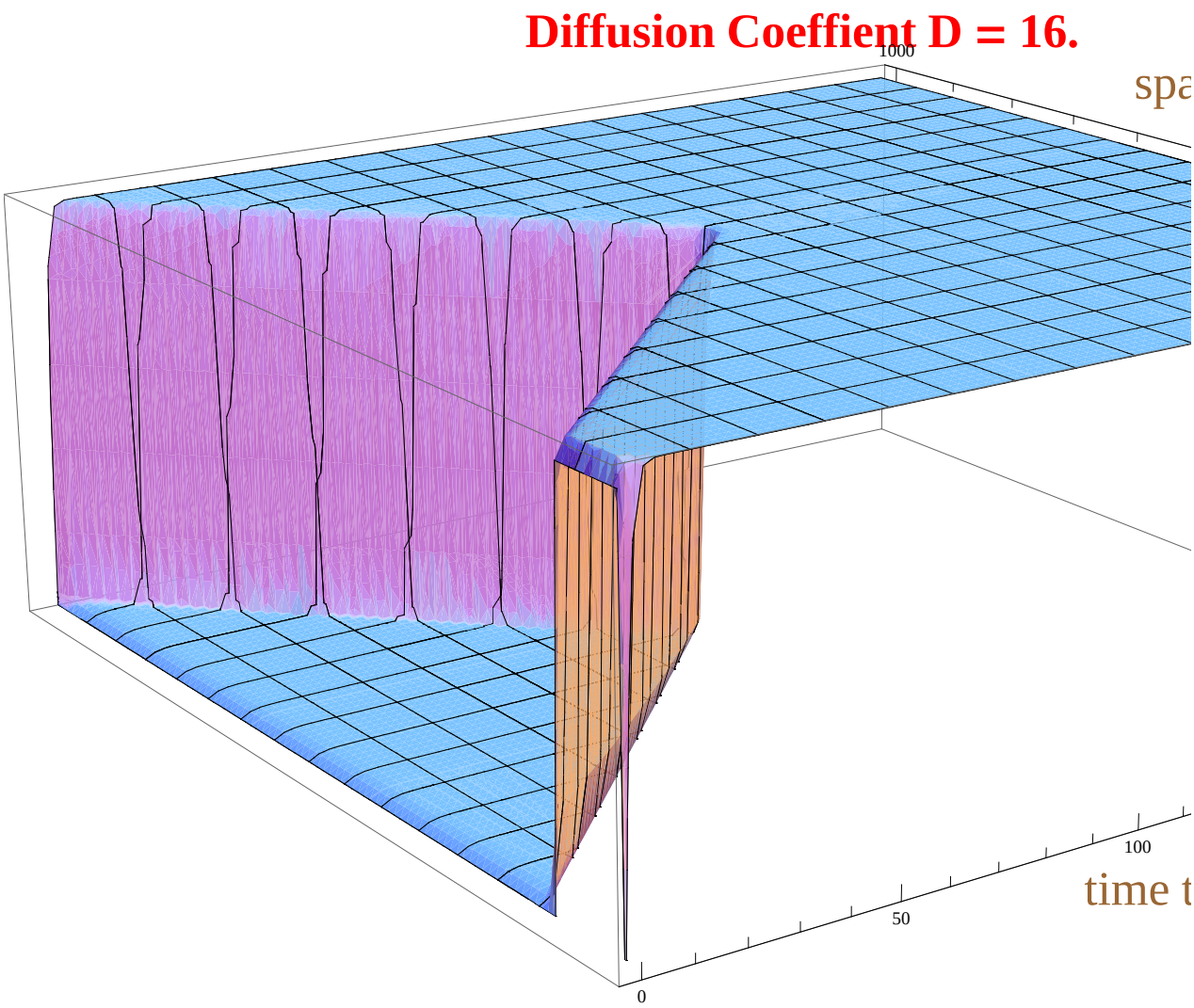
Setting Diffusion Coefficient and Solve PDE

```
CoeffD = 16.0;

NDSol = NDSolve[{∂t u[t, x] == CoeffD * ∂x,x u[t, x] + (f[u[t, x]] /. ParamRules),
    u[0, x] == InitialConditionProfile (* initial condition *),
    u[t, xRange // Last] == u[t, 0] (* periodic boundary condition *)},
   u,
   tRange, xRange,
   Method → {"MethodOfLines", "SpatialDiscretization" → {"TensorProductGrid"}} ];
```

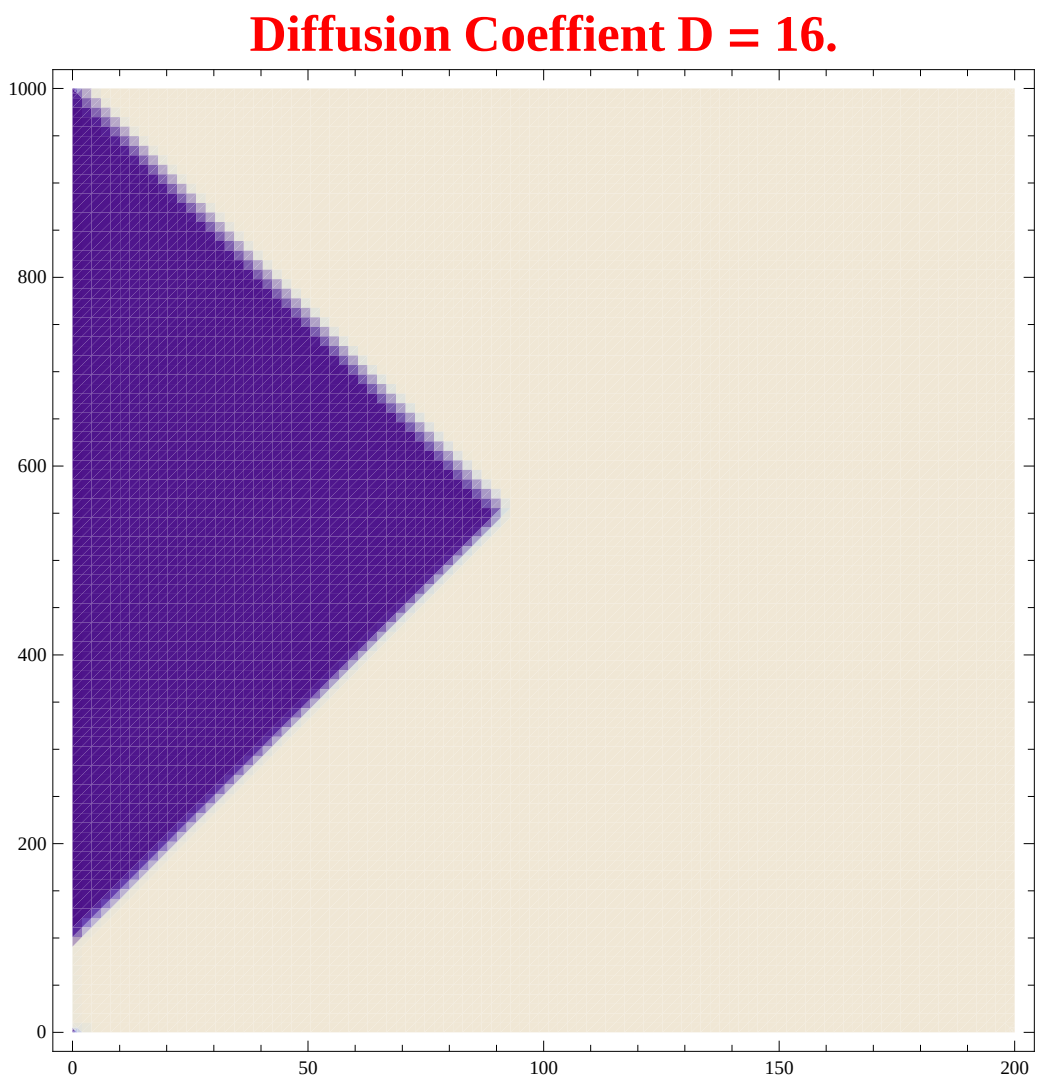
```
NDSolve::mxsst :
 Using maximum number of grid points 10000 allowed by the MaxPoints or MinStepSize
   options for independent variable x. >>
```

```
Plot3D[Evaluate[u[t, x] /. NDSol], {t, 0, tRange // Last}, {x, 0, xRange // Last},
 PlotRange → All, PlotPoints → 100, AxesLabel → {StyleForm["time t", Large, Brown],
   StyleForm["space, x", Large, Brown], StyleForm["Conc., c(x,t)", Large, Brown]},
 PlotLabel → StyleForm["Diffusion Coeffient D = " <> ToString[CoeffD], Red, Large, Bold],
 ImageSize → {900, 500}]
```



Density Plot

```
DensPlot1 =
 DensityPlot[Evaluate[u[t, x] /. NDSol], {t, 0, tRange // Last}, {x, 0, xRange // Last},
  PlotRange → All, PlotPoints → 100, AxesLabel → {StyleForm["time t", Large],
    StyleForm["space, x", Large], StyleForm["Conc., c(x,t)", Large]},
  PlotLabel → StyleForm["Diffusion Coeffient D = " <> ToString[CoeffD], Red, Large, Bold],
  ImageSize → {500, 500}]
```



Comparison of Solution for Different Diffusion Constants

```
CoeffD = 16.0 / 4;
```

```
NDSol = NDSolve[{∂t u[t, x] == CoeffD * ∂x,x u[t, x] + (f[u[t, x]] /. ParamRules),
    u[0, x] == InitialConditionProfile (* initial condition *),
    u[t, xRange // Last] == u[t, 0] (* periodic boundary condition *)}, u,
   tRange, xRange,
   Method → {"MethodOfLines", "SpatialDiscretization" → {"TensorProductGrid"}}];
```

```
NDSolve::mxsst :
 Using maximum number of grid points 10000 allowed by the MaxPoints or MinStepSize
   options for independent variable x. >>
```

```
DensPlot2 =
  DensityPlot[Evaluate[u[t, x] /. NDSol], {t, 0, tRange // Last}, {x, 0, xRange // Last},
   PlotRange → All, PlotPoints → 100, AxesLabel → {StyleForm["time t", Large],
     StyleForm["space, x", Large], StyleForm["Conc., c(x,t)", Large]},
   PlotLabel → StyleForm["Diffusion Coeffient D = " <> ToString[CoeffD], Red, Large, Bold],
   ImageSize → {500, 500}];
```

```
GraphicsArray[{{DensPlot1, DensPlot2}}]
```