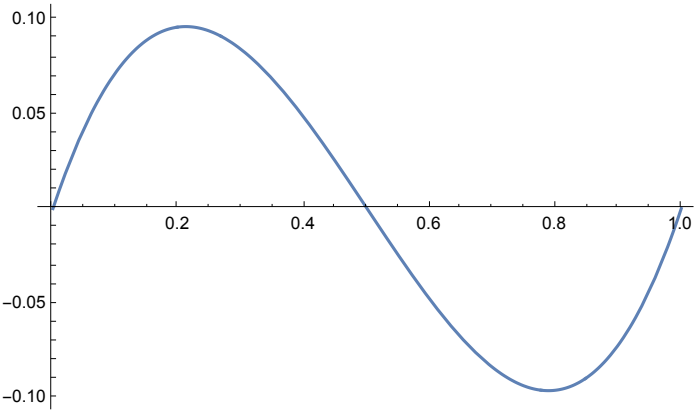
SGDs

Game1

```
In[•]:= payoffmat1 = {{-1, 1}, {0, 0}};

In[•]:= sgd1fitness[strat_, x1_, x2_] := {payoffmat1.{x1, x2}}[[1, strat]];

In[•]:= sgd1avgfit[x1_, x2_] := {x1, x2}.(payoffmat1.{x1, x2});

In[•]:= Plot[x (1 - x) (sgd1fitness[1, x, 1 - x] - sgd1fitness[2, x, 1 - x]), {x, 0, 1}]
```

Out[•]=




```
In[•]:= sgd1eq = Solve[(sgd1fitness[1, x, 1 - x] - sgd1fitness[2, x, 1 - x]) == 0, x][[1, 1, 2]]
```

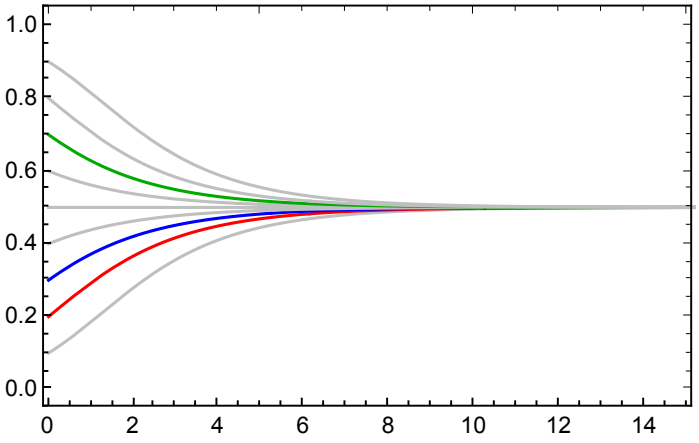
Out[•]=

$$\frac{1}{2}$$

```
In[•]:= tlim = 50;

In[•]:= sgd1sol[init_] := NDSolve[{
          x1'[t] == x1[t] (sgd1fitness[1, x1[t], x2[t]] - sgd1avgfit[x1[t], x2[t]]),
          x2'[t] == x2[t] (sgd1fitness[2, x1[t], x2[t]] - sgd1avgfit[x1[t], x2[t]]),
          x1[0] == init, x2[0] == 1 - init}, {x1, x2}, {t, 0, tlim}]
```

```
In[•]:= Plot[{
          Evaluate[{x1[iter]} /. sgd1sol[0.1]],
          Evaluate[{x1[iter]} /. sgd1sol[0.2]],
          Evaluate[{x1[iter]} /. sgd1sol[0.3]],
          Evaluate[{x1[iter]} /. sgd1sol[0.4]],
          Evaluate[{x1[iter]} /. sgd1sol[0.5]],
          Evaluate[{x1[iter]} /. sgd1sol[0.6]],
          Evaluate[{x1[iter]} /. sgd1sol[0.7]],
          Evaluate[{x1[iter]} /. sgd1sol[0.8]],
          Evaluate[{x1[iter]} /. sgd1sol[0.9]]}, {iter, 0, 30},
         PlotRange → {{-0.1, 15.1}, {-0.05, 1.05}}, Frame → True, PlotStyle →
          {Lighter[Gray, 0.5], Red, Blue, Lighter[Gray, 0.5], Lighter[Gray, 0.5],
           Lighter[Gray, 0.5], Darker[Green], Lighter[Gray, 0.5], Lighter[Gray, 0.5]},
         FrameTicksStyle → Directive[Black, 12], FrameStyle →
          Directive[Thickness[0.003]]]
```

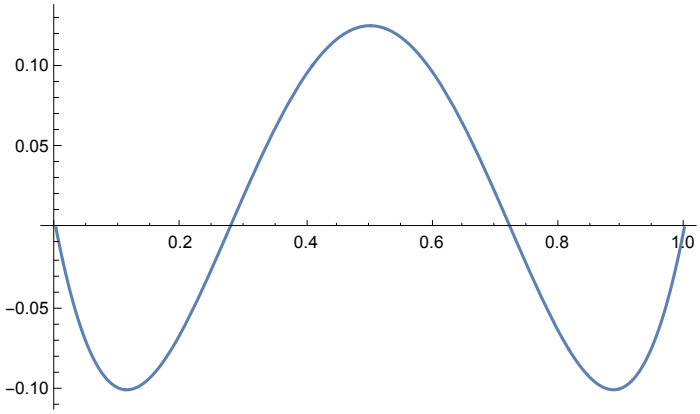
Out[•]=



Game2

```
In[•]:= payoffmat2 = {{-2, 3, -2}, {0, 0, 0}};

In[•]:= sgd2fitness[strat_, x1_, x2_] := {payoffmat2.{x1^2, 2 x1 x2, x2^2}}[[1, strat]];

In[•]:= sgd2avgfit[x1_, x2_] := {x1, x2}.(payoffmat2.{x1^2, 2 x1 x2, x2^2});

In[•]:= Plot[x (1 - x) (sgd2fitness[1, x, 1 - x] - sgd2fitness[2, x, 1 - x]), {x, 0, 1}]
```

Out[•]=



```
In[•]:= sgd2eq = Solve[(sgd2fitness[1, x, 1 - x] - sgd2fitness[2, x, 1 - x]) == 0, x][[2, 1, 2]]
```

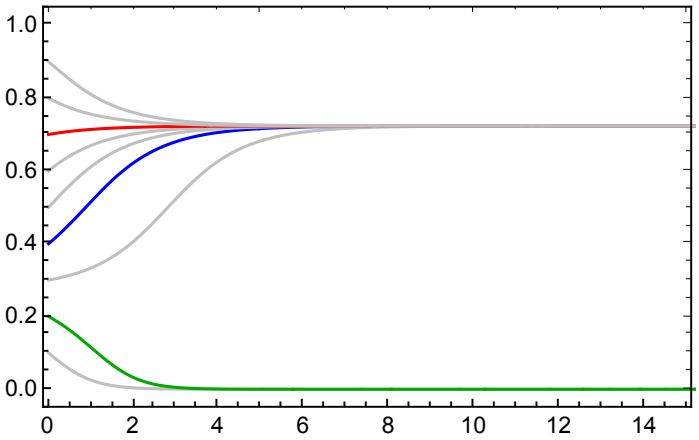
$$\text{Out[•]=}\quad \frac{1}{10}\left(5+\sqrt{5}\right)$$

```
In[•]:= tlim = 50;
```

```
In[•]:= sgd2sol[init_] := NDSolve[{
          x1'[t] == x1[t] (sgd2fitness[1, x1[t], x2[t]] - sgd2avgfit[x1[t], x2[t]]),
          x2'[t] == x2[t] (sgd2fitness[2, x1[t], x2[t]] - sgd2avgfit[x1[t], x2[t]]),
          x1[0] == init, x2[0] == 1 - init}, {x1, x2}, {t, 0, tlim}]
```

```
In[•]:= Plot[{
          Evaluate[{x1[iter]} /. sgd2sol[0.1]],
          Evaluate[{x1[iter]} /. sgd2sol[0.2]],
          Evaluate[{x1[iter]} /. sgd2sol[0.3]],
          Evaluate[{x1[iter]} /. sgd2sol[0.4]],
          Evaluate[{x1[iter]} /. sgd2sol[0.5]],
          Evaluate[{x1[iter]} /. sgd2sol[0.6]],
          Evaluate[{x1[iter]} /. sgd2sol[0.7]],
          Evaluate[{x1[iter]} /. sgd2sol[0.8]],
          Evaluate[{x1[iter]} /. sgd2sol[0.9]]}, {iter, 0, 30},
         PlotRange → {{-0.1, 15.1}, {-0.05, 1.05}}, Frame → True,
         PlotStyle → {Lighter[Gray, 0.5], Darker[Green], Lighter[Gray, 0.5],
           Blue, Lighter[Gray, 0.5], Lighter[Gray, 0.5], Red, Lighter[Gray, 0.5],
           Lighter[Gray, 0.5]}, FrameTicksStyle → Directive[Black, 12],
         FrameStyle → Directive[Thickness[0.003]]]
```

Out[•]=



W-E

```
In[•]:= esols = Solve[x11 + x12 == sgd1eq && x11 + x21 == sgd2eq &&
          x11 + x12 + x21 + x22 == 1 && x11 x22 == x21 x12, {x11, x12, x21, x22}]
```

$$\text{Out[•]=}\quad \left\{\left\{x11 \to \frac{1}{20}\left(5+\sqrt{5}\right),\ x12 \to \frac{1}{20}\left(5-\sqrt{5}\right),\ x21 \to \frac{1}{20}\left(5+\sqrt{5}\right),\ x22 \to \frac{1}{20}\left(5-\sqrt{5}\right)\right\}\right\}$$

```
In[•]:= solutionlist =
         Table[{esols[[1, 1, 2]] + t, esols[[1, 2, 2]] - t, esols[[1, 3, 2]] - t, esols[[1, 4, 2]] + t},
          {t, -esols[[1, 2, 2]], esols[[1, 2, 2]], 0.01}]
```

```
Out[•]= {{0.223607, 0.276393, 0.5, 0.}, {0.233607, 0.266393, 0.49, 0.01},
         {0.243607, 0.256393, 0.48, 0.02}, {0.253607, 0.246393, 0.47, 0.03},
         {0.263607, 0.236393, 0.46, 0.04}, {0.273607, 0.226393, 0.45, 0.05},
         {0.283607, 0.216393, 0.44, 0.06}, {0.293607, 0.206393, 0.43, 0.07},
         {0.303607, 0.196393, 0.42, 0.08}, {0.313607, 0.186393, 0.41, 0.09},
         {0.323607, 0.176393, 0.4, 0.1}, {0.333607, 0.166393, 0.39, 0.11},
         {0.343607, 0.156393, 0.38, 0.12}, {0.353607, 0.146393, 0.37, 0.13},
         {0.363607, 0.136393, 0.36, 0.14}, {0.373607, 0.126393, 0.35, 0.15},
         {0.383607, 0.116393, 0.34, 0.16}, {0.393607, 0.106393, 0.33, 0.17},
         {0.403607, 0.0963932, 0.32, 0.18}, {0.413607, 0.0863932, 0.31, 0.19},
         {0.423607, 0.0763932, 0.3, 0.2}, {0.433607, 0.0663932, 0.29, 0.21},
         {0.443607, 0.0563932, 0.28, 0.22}, {0.453607, 0.0463932, 0.27, 0.23},
         {0.463607, 0.0363932, 0.26, 0.24}, {0.473607, 0.0263932, 0.25, 0.25},
         {0.483607, 0.0163932, 0.24, 0.26}, {0.493607, 0.0063932, 0.23, 0.27}}
```

```
In[•]:= eline = Line[Map[new3d[#] &, solutionlist]];
```

MGD

This is a function for

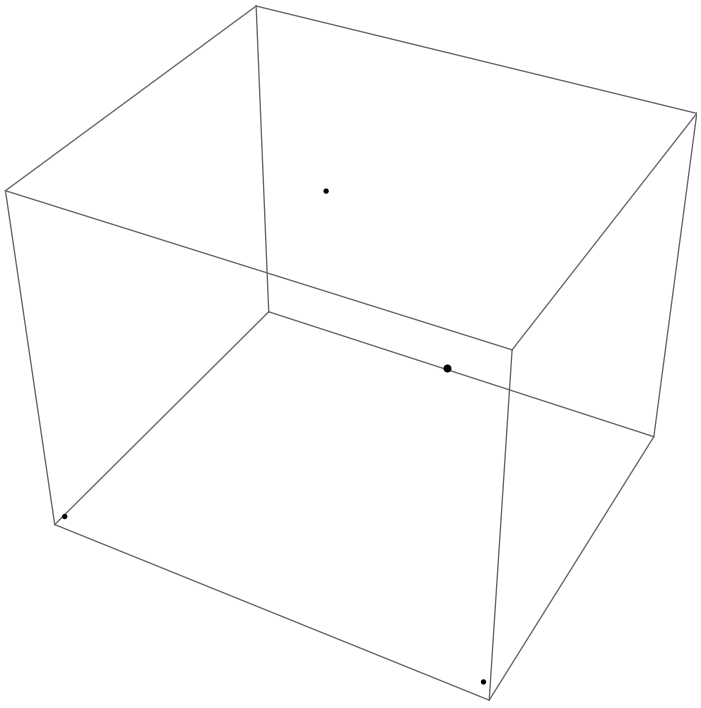
```
In[•]:= payoffmgd[x11_, x12_, x21_, x22_] :=
          {(sgd1fitness[1, x11 + x12, x21 + x22] - sgd1avgfit[x11 + x12, x21 + x22]) +
             (sgd2fitness[1, x11 + x21, x12 + x22] - sgd2avgfit[x11 + x21, x12 + x22]),
            (sgd1fitness[1, x11 + x12, x21 + x22] - sgd1avgfit[x11 + x12, x21 + x22]) +
             (sgd2fitness[2, x11 + x21, x12 + x22] - sgd2avgfit[x11 + x21, x12 + x22]),
            (sgd1fitness[2, x11 + x12, x21 + x22] - sgd1avgfit[x11 + x12, x21 + x22]) +
             (sgd2fitness[1, x11 + x21, x12 + x22] - sgd2avgfit[x11 + x21, x12 + x22]),
            (sgd1fitness[2, x11 + x12, x21 + x22] - sgd1avgfit[x11 + x12, x21 + x22]) +
             (sgd2fitness[2, x11 + x21, x12 + x22] - sgd2avgfit[x11 + x21, x12 + x22])};
```

```
In[•]:= mgdfitness[strat_, x11_, x12_, x21_, x22_] :=
          {payoffmgd[x11, x12, x21, x22]}[[1, strat]];
```

```
In[•]:= mgdavgfit[x11_, x12_, x21_, x22_] :=
          {x11, x12, x21, x22}.payoffmgd[x11, x12, x21, x22];
```

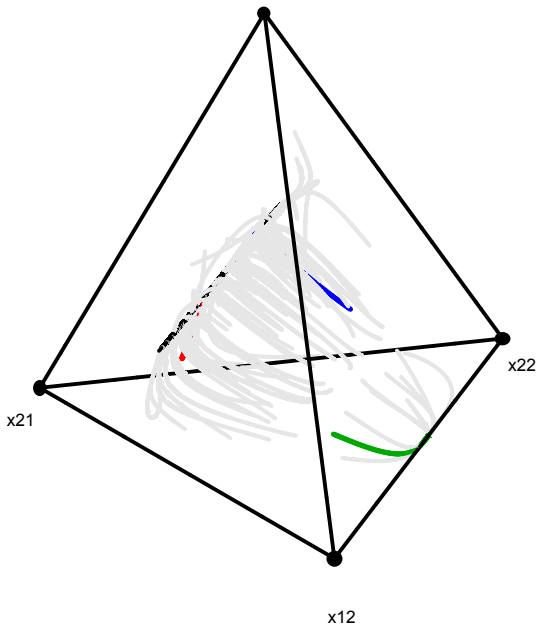
```
In[•]:= tlim = 100;
mgdsol[init_] := NDSolve[{
   x11'[t] == x11[t] (mgdfitness[1, x11[t], x12[t], x21[t], x22[t]] -
       mgdavgfit[x11[t], x12[t], x21[t], x22[t]]),
   x12'[t] == x12[t] (mgdfitness[2, x11[t], x12[t], x21[t], x22[t]] -
       mgdavgfit[x11[t], x12[t], x21[t], x22[t]]),
   x21'[t] == x21[t] (mgdfitness[3, x11[t], x12[t], x21[t], x22[t]] -
       mgdavgfit[x11[t], x12[t], x21[t], x22[t]]),
   x22'[t] == x22[t] (mgdfitness[4, x11[t], x12[t], x21[t], x22[t]] -
       mgdavgfit[x11[t], x12[t], x21[t], x22[t]]),
   x11[0] == init[[1]], x12[0] == init[[2]], x21[0] == init[[3]],
   x22[0] == 1 - init[[1]] - init[[2]] - init[[3]]}, {x11, x12, x21, x22}, {t, 0, tlim}]

In[•]:= new3d[r_] := {r[[2]] + 1 / 2 * (r[[3]] + r[[4]]), √3 * (r[[3]] / 2 + r[[4]] / 6), √6 * r[[4]] / 3};

In[•]:= list1 = Table[Map[new3d[#] &, Flatten[Table[Evaluate[
         {x11[t], x12[t], x21[t], x22[t]} /. mgdsol[init]], {t, 0, tlim, 0.05}], 1]],
   {init, {{0.1, 0.1, 0.6}, {0.2, 0.1, 0.2}, {0.1, 0.6, 0.1}}}];

In[•]:= raninitlis = Table[temp = RandomReal[1, 4];
    temp / Total[temp][[1 ;; 3]], {i, 100}];

In[•]:= ranlis = Table[Map[new3d[#] &,
     Flatten[Table[Evaluate[{x11[t], x12[t], x21[t], x22[t]} /. mgdsol[init]],
       {t, 0, tlim, 0.05}], 1]], {init, raninitlis}];

In[•]:= ranlines = Table[{Lighter[Gray, 0.8], Thick, Line[ranlis[[i]]]}, {i, 100}];

In[•]:= Graphics3D[{Point[new3d[{1, 0, 0, 0}]], Point[new3d[{0, 1, 0, 0}]],
   Point[new3d[{0, 0, 1, 0}]], Point[new3d[{0, 0, 0, 1}]]}]
```

Out[•]=



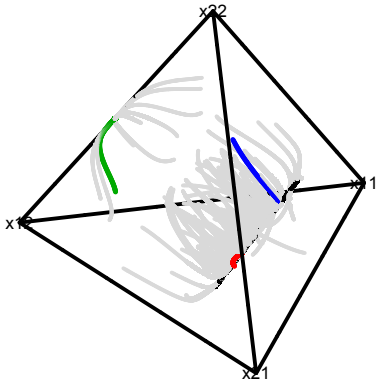
In[•]:=
```
tetra = Graphics3D[{
    {Thick, Line[{new3d[{1, 0, 0, 0}], new3d[{0, 1, 0, 0}]}]},
    {Thick, Line[{new3d[{1, 0, 0, 0}], new3d[{0, 0, 1, 0}]}]},
    {Thick, Line[{new3d[{1, 0, 0, 0}], new3d[{0, 0, 0, 1}]}]},
    {Thick, Line[{new3d[{0, 1, 0, 0}], new3d[{0, 0, 1, 0}]}]},
    {Thick, Line[{new3d[{0, 1, 0, 0}], new3d[{0, 0, 0, 1}]}]},
    {Thick, Line[{new3d[{0, 0, 1, 0}], new3d[{0, 0, 0, 1}]}]},
    {Red, Thickness[0.008], Line[list1[[1]]]},
    {Blue, Thickness[0.0074], Line[list1[[2]]]},
    {Darker[Green], Thickness[0.0073], Line[list1[[3]]]}, ranlines,
    {PointSize[0.02], Point[new3d[{1, 0, 0, 0}]]},
    {PointSize[0.02], Point[new3d[{0, 1, 0, 0}]]},
    {PointSize[0.02], Point[new3d[{0, 0, 1, 0}]]},
    {PointSize[0.02], Point[new3d[{0, 0, 0, 1}]]}, Text[x12, new3d[{0, 1.1, 0, 0}]],
    Text[x21, new3d[{0, 0, 1.1, 0}]], Text[x22, new3d[{0, 0, 0, 1.1}]],
    {Black, Thick, Dashed, eline}}, Boxed → False
]
```

Out[•]=



In[•]:=
```
Show[tetra, SphericalRegion → True,
 ViewPoint → {Pi, 15, 10}, Boxed → False, Axes → False]
```

Out[•]=

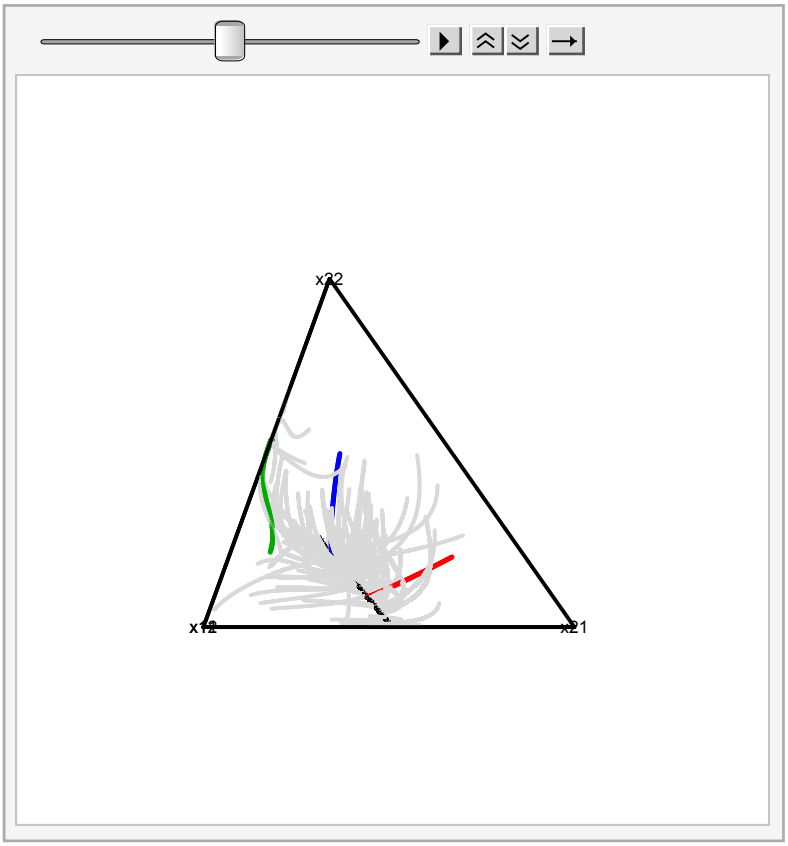


In[•]:=
```
autoRotate[gr_Graphics3D, rate_: 5] := DynamicModule[{vp, va, vv, vc},
  {vp, va, vv, vc} = gr~AbsoluteOptions~#~OptionValue~# &@
    {ViewPoint, ViewAngle, ViewVertical, ViewCenter};
  Overlay[{Show[Graphics3D[], ViewPoint → Dynamic[vp], ViewAngle → Dynamic[va],
     SphericalRegion → True, Boxed → False], Show[gr, SphericalRegion → True,
     ViewPoint → Dynamic[RotationMatrix[Clock[2 π, rate], vv].vp],
     ViewAngle → Dynamic[va], Boxed → False, Axes → False]}, All, 1]]
```

In[•]:=
```
ani = Table[Show[tetra, SphericalRegion → True,
   ViewVector → {5 Cos[t], 5 Sin[t], 10 Sin[t / 2]}], {t, 0, 4 Pi, π / 2}];
```

In[•]:=
```
ani // ListAnimate
```

Out[•]=

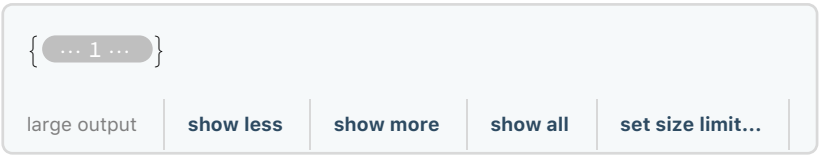


In[•]:=
```
Export[NotebookDirectory[] <> "mov.avi", ani]
```

Out[•]=
```
$Aborted
```

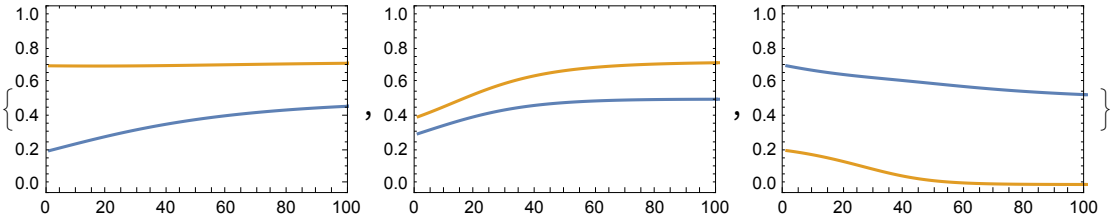
In[•]:=
```
recoversgd = Table[
  Flatten[Table[{iter, Evaluate[{x11[iter] + x12[iter], x11[iter] + x21[iter]} /.
       mgdsol[init]]}, {iter, 0, tlim, 0.05}], 1],
 {init, {{0.1, 0.1, 0.6}, {0.2, 0.1, 0.2}, {0.1, 0.6, 0.1}}}]
```

Out[•]=



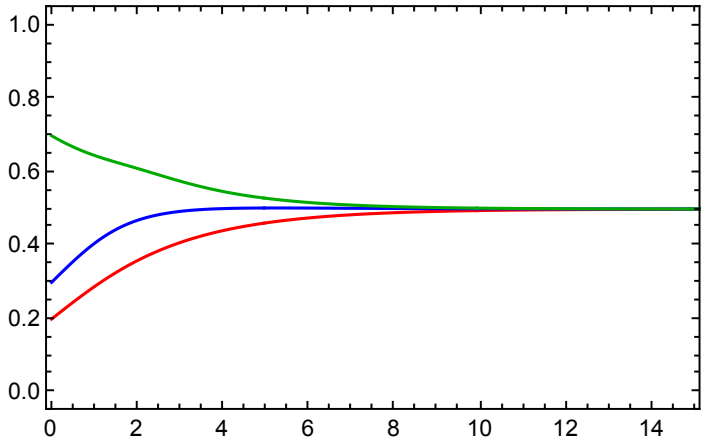
In[•]:=
```
ListPlot[{#[[All, 1]], #[[All, 2]]}, Joined → True,
   PlotRange → {{-0.1, 100.1}, {-0.05, 1.05}}, Frame → True] & /@ recoversgd
```

Out[•]=



In[•]:=
```
ListPlot[{Partition[Flatten[recoversgd[[1]], 2], 3][[All, {1, 2}]],
  Partition[Flatten[recoversgd[[2]], 2], 3][[All, {1, 2}]],
  Partition[Flatten[recoversgd[[3]], 2], 3][[All, {1, 2}]]},
 Joined → True, PlotRange → {{-0.1, 15.1}, {-0.05, 1.05}}, Frame → True,
 PlotStyle → {Red, Blue, Darker[Green]}, FrameTicksStyle → Directive[Black, 12],
 FrameStyle → Directive[Thickness[0.003]]]
```

Out[•]=



In[•]:=
```
Partition[Flatten[recoversgd[[1]], 2], 3][[1, {1, 3}]]
```

Out[•]=
```
{0., 0.7}
```

In[•]:=
```
ListPlot[{Partition[Flatten[recoversgd[[1]], 2], 3][[All, {1, 3}]],
  Partition[Flatten[recoversgd[[2]], 2], 3][[All, {1, 3}]],
  Partition[Flatten[recoversgd[[3]], 2], 3][[All, {1, 3}]]},
 Joined → True, PlotRange → {{-0.1, 15.1}, {-0.05, 1.05}}, Frame → True,
 PlotStyle → {Red, Blue, Darker[Green]}, FrameTicksStyle → Directive[Black, 12],
 FrameStyle → Directive[Thickness[0.003]]]
```

Out[•]=