Reaction diffusion equation with Schnakenberg kinetics Turing pattern formation parallel-computed by MPI using two nodes

Schnakenberg kinetics

力学系の定義

```
deq = {u'[t] == k1 - k2 u[t] + k3 u[t]^2 v[t],
       v'[t] == k4 - k3 u[t]^2 v[t],
       u[0] == u0, v[0] == v0}
```

$$\{u'[t] == k1 - k2\, u[t] + k3\, u[t]^2\, v[t],\ v'[t] == k4 - k3\, u[t]^2\, v[t],\ u[0] == u0,\ v[0] == v0\}$$

パラメータの設定

```
para = {k1 → 0.2, k2 → 1, k3 → 1, k4 → 0.5}
```

$$\{k1 \to 0.2,\ k2 \to 1,\ k3 \to 1,\ k4 \to 0.5\}$$

初期値の設定

```
deqN = deq /. para /. {u0 → 0.1, v0 → 0.2}
```

$$\{u'[t] == 0.2 - u[t] + u[t]^2\, v[t],\ v'[t] == 0.5 - u[t]^2\, v[t],\ u[0] == 0.1,\ v[0] == 0.2\}$$

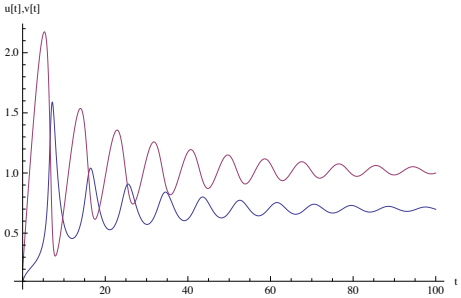
数値的に解く

```
tEnd = 100;
sol = NDSolve[deqN, {u[t], v[t]}, {t, 0, tEnd}]
```

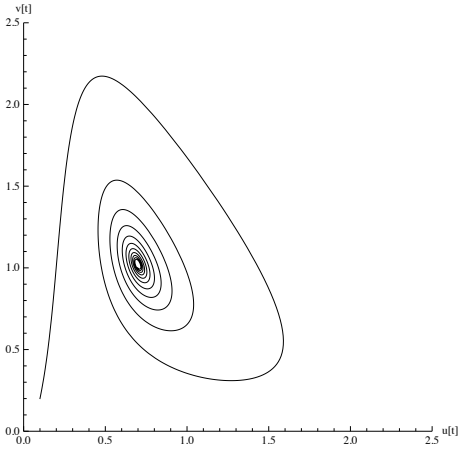
```
{{u[t] → InterpolatingFunction[{{0., 100.}}, <>][t],
  v[t] → InterpolatingFunction[{{0., 100.}}, <>][t]}}
```

視覚可

```
Plot[ Evaluate[ {u[t], v[t]} /. sol], {t, 0, tEnd},
 PlotRange → All, AxesLabel → {"t", "u[t],v[t]"}]
```




```
gdyna = ParametricPlot[ Evaluate[ {u[t], v[t]} /. sol], {t, 0, tEnd},
  PlotRange → {{0, 2.5}, {0, 2.5}}, AxesLabel → {"u[t]", "v[t]"},
  AspectRatio → 1, PlotStyle → RGBColor[0, 0, 0]]
```

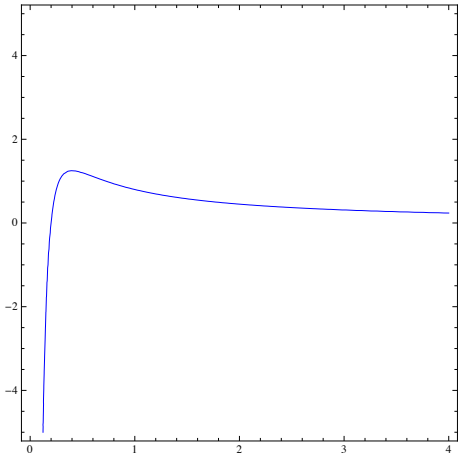



```
isoU = k1 - k2 u + k3 u^2 v;
isoV = k4 - k3 u^2 v;

isoUN = isoU /. para
isoVN = isoV /. para
```

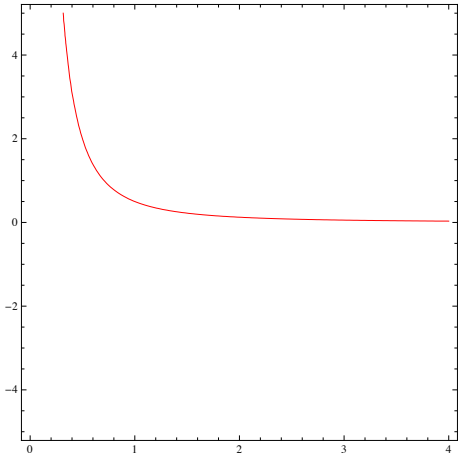
$$0.2 - u + u^2\, v$$

$$0.5 - u^2\, v$$

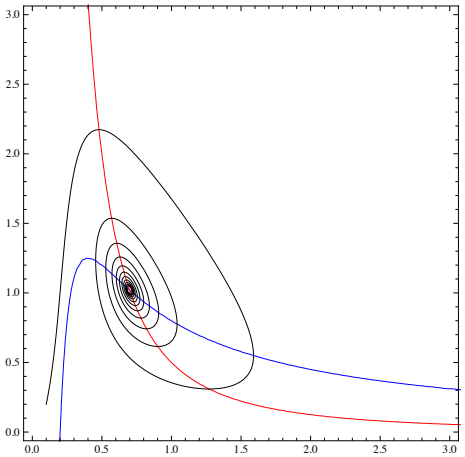
```
gisoU = ContourPlot[isoUN == 0, {u, 0, 4}, {v, -5, 5},
   ContourStyle → RGBColor[0, 0, 1], AspectRatio → 1]
```



```
gisoV = ContourPlot[isoVN == 0, {u, 0, 4}, {v, -5, 5},
   ContourStyle → RGBColor[1, 0, 0], AspectRatio → 1]
```




```
Show[gisoU, gisoV, gdyna, AspectRatio → 1, PlotRange → {{0, 3}, {0, 3}}]
```



Schnakenberg kinetics + diffusion

```
SetDirectory["/Users/takasu/home/情報科学科の仕事/講義/平成２３年度/H23
    計算機実験2/MPI/testMPI/build/Debug/"]
```

```
/Users/takasu/home/情報科学科の仕事/講義/平成２３年度/H23
  計算機実験2/MPI/testMPI/build/Debug
```

- **Non MPI with one node**

ファイルの読み込み

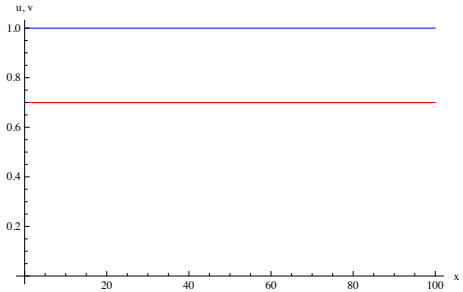
```
size = 101;

datau = ReadList["data-u", {Real, Real}];
datau = Partition[datau, size];

datav = ReadList["data-v", {Real, Real}];
datav = Partition[datav, size];

Length[datau]
Length[datav]
```

```
100
```

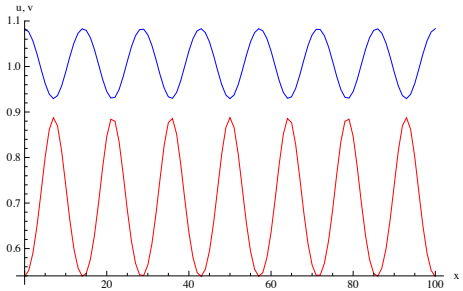
```
100
```

```
drawFigure[x1_List, x2_List, opt___] :=
 Block[{g1, g2}, g1 = ListPlot[x1, Joined → True,
    PlotStyle → RGBColor[1, 0, 0], opt, DisplayFunction → Identity];
  g2 = ListPlot[x2, Joined → True, PlotStyle → RGBColor[0, 0, 1],
    opt, DisplayFunction → Identity];
  Return[Show[g1, g2, DisplayFunction → $DisplayFunction]]]
```

```
drawFigure[datau[[1]], datav[[1]],
 PlotRange → All, AxesLabel → {"x", "u, v"}]
```




```
drawFigure[Last[datau], Last[datav],
 PlotRange → All, AxesLabel → {"x", "u, v"}]
```

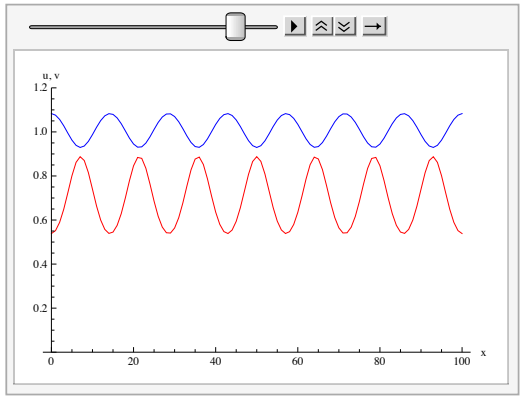


描画

```
glist = {};
Do[
 g = drawFigure[ datau[[i]], datav[[i]] ,
    PlotRange → {0, 1.2}, PlotJoined → True, AxesLabel → {"x", "u, v"}];
 AppendTo[glist, g], {i, 1, Length[datau], 10}
]
```



```
ListAnimate[glist]
```



■ MPI with two nodes

2つのファイルの読み込み

```
size = 51;

datau0 = ReadList["data-u-0", {Real, Real}];
datau0 = Partition[datau0, size];

datau1 = ReadList["data-u-1", {Real, Real}];
datau1 = Partition[datau1, size];

datav0 = ReadList["data-v-0", {Real, Real}];
datav0 = Partition[datav0, size];

datav1 = ReadList["data-v-1", {Real, Real}];
datav1 = Partition[datav1, size];


Length[datau0]
Length[datau1]
```

100

100

2つのファイルの連結

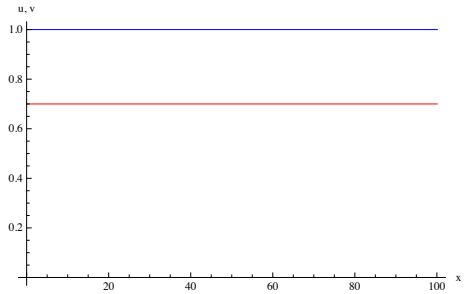
```
datauAll = {};
Do[
 AppendTo[ datauAll, Join[datau0[[i]], datau1[[i]]] ],
 {i, 1, Length[datau0]}
]

datavAll = {};
Do[
 AppendTo[ datavAll, Join[datav0[[i]], datav1[[i]]] ],
 {i, 1, Length[datav0]}
]

Length[datauAll]
Length[datavAll]
```

```
100
```

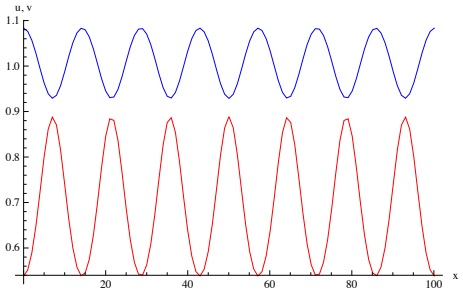
```
100
```

```
drawFigure[x1_List, x2_List, opt___] :=
 Block[{g1, g2}, g1 = ListPlot[x1, Joined → True,
    PlotStyle → RGBColor[1, 0, 0], opt, DisplayFunction → Identity];
  g2 = ListPlot[x2, Joined → True, PlotStyle → RGBColor[0, 0, 1],
    opt, DisplayFunction → Identity];
  Return[Show[g1, g2, DisplayFunction → $DisplayFunction]]]
```

```
drawFigure[datauAll[[1]], datavAll[[1]],
 PlotRange → All, AxesLabel → {"x", "u, v"}]
```



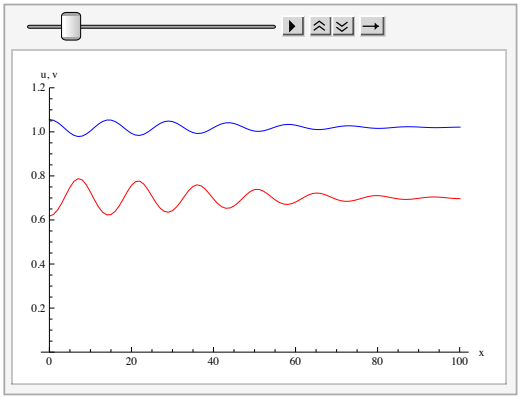

```
drawFigure[Last[datauAll], Last[datavAll],
 PlotRange → All, AxesLabel → {"x", "u, v"}]
```



描画

```
glist = {};
Do[
 g = drawFigure[ datauAll[[i]], datavAll[[i]] ,
   PlotRange → {0, 1.2}, PlotJoined → True, AxesLabel → {"x", "u, v"}];
 AppendTo[glist, g], {i, 1, Length[datauAll], 10}
]
```

```
ListAnimate[glist]
```



2

2