

In[540]:=

enddt = 60;

In[541]:=

A[x_] = D[7 / 60 (x - 30) (Exp[100 (x - 30)]), x]
|微分 |指数関数

B[x_] = D[-5 (x) / 60 (Exp[-100 (x)]), x]
|微分 |指数関数

Out[541]=

$$\frac{7}{60} e^{100(-30+x)} + \frac{35}{3} e^{100(-30+x)} (-30+x)$$

Out[542]=

$$-\frac{1}{12} e^{-100x} + \frac{25}{3} e^{-100x} x$$

In[543]:=

A[30]

Out[543]=

$$\frac{7}{60}$$

In[544]:=

N[A[0], 10]
|数値

Out[544]=

$$-4.575910754 \times 10^{-1301}$$

In[545]:=

N[B[30], 10]
|数値

Out[545]=

$$3.268507681 \times 10^{-1301}$$

In[546]:=

B[0]

Out[546]=

$$-\frac{1}{12}$$

In[547]:=

55 / 5

Out[547]=

11

In[548]:=

85 / 5

Out[548]=

17

In[549]:=

```

a = 0; b = 30;
z[x_] := (x - a) / (b - a);
w[x_] := z[x]^2 (1 - z[x])^2; (*端で値も傾きも 0*)

```

(*例：ガウスを元にする*)

```
gA[x_] := 1 / 2 Exp[-7 / 10 (x - 20)^2];
```

|指数関数

```
gB[x_] := 1 / 2 Exp[-7 / 10 (x - 10)^2];
```

|指数関数

```
Fa0[x_] := w[x] × gA[x];
```

```
Fb0[x_] := w[x] × gB[x];
```

(*厳密チェック*)

```
Simplify[D[Fa0[x], x] /. x → a] (*0*)
```

|簡単な… |微分

```
Simplify[D[Fa0[x], x] /. x → b] (*0*)
```

|簡単な… |微分

```
Simplify[Fa0[a]] (*0*)
```

|簡単な形式に

```
Simplify[Fa0[b]] (*0*)
```

|簡単な形式に

Out[556]=

0

Out[557]=

0

Out[558]=

0

Out[559]=

0

In[560]:=

```
(*パラメータ*) rfb = 1; rmb = 1; rrfb = 1; rfa = 1; rma = 1; rrfa = 1;
```

```
centerr = 20; centerl = 10; lossr = 1 / 10; lossf = 1 / 10;
```

```
rra[t_, x_] = 1 / 10;
```

```
rrb[t_, x_] = 1 / 10;
```

```
cA = -2;
```

```
cAR = -2;
```

```
cB = 2;
```

```
cBR = 2;
```

```
dAR = 5 / 100;
```

```
dBR = 5 / 100;
```

```
dAF = 5 / 100;
```

```
dBf = 5 / 100;
```

```
alpha = 100; beta = 40;
```

```
Sigmoid[x_] := 1 / (1 + Exp[-alpha (x - 0.1)]);
```

|指数関数

```
dfax[t_, x_] := D[Fa[t, s], s] /. s → x;
```

|微分

```
dfbx[t_, x_] := D[Fb[t, s], s] /. s → x;
```

|微分

```

dRax[t_, x_] := D[Resa[t, s], s] /. s -> x;
|微分
dRbx[t_, x_] := D[Resb[t, s], s] /. s -> x;
|微分
Resa0[x_] := (70 / (60 × 5) (x - 30) (Exp[100 (x - 30)])) (*;/;x<30*);
|指数関数
Resb0[x_] := (-50 (x) / (60 × 5) (Exp[-100 (x)])) (*;/;x>0*);
|指数関数

(*PDE と初期・境界条件*)
pdes =
{D[Resa[t, x], t] == -rra[t, x] × Resa[t, x] - rrfb Fb[t, x] - cAR D[Resa[t, x], {x, 1}] +
|微分 |微分
dAR (D[Resa[t, x], {x, 2}]) + beta (Sqrt[(Resa[t, x])^2] - Resa[t, x]) +
|微分 |平方根
NeumannValue[7 / (5 × 12), x == 30] + NeumannValue[0, x == 0],
|ノイマン境界値 |ノイマン境界値
D[Resb[t, x], t] == -rrb[t, x] × Resb[t, x] - rrfb Fa[t, x] - cBR D[Resb[t, x], {x, 1}] +
|微分 |微分
dBR (D[Resb[t, x], {x, 2}]) + beta (Sqrt[(Resb[t, x])^2] - Resb[t, x]) +
|微分 |平方根
NeumannValue[5 / (5 × 12), x == 0] + NeumannValue[0, x == 30], D[Fa[t, x], t] ==
|ノイマン境界値 |ノイマン境界値 |微分
-rfb Fb[t, x] + Ma[t, x] + (-cA D[Fa[t, x], {x, 1}] + dAF D[Fa[t, x], {x, 2}]) -
|微分 |微分
lossf Fa[t, x] + beta (Sqrt[(Fa[t, x])^2] - Fa[t, x]) +
|平方根
NeumannValue[0, x == 0] + NeumannValue[0, x == 30], D[Fb[t, x], t] ==
|ノイマン境界値 |ノイマン境界値 |微分
-rfa Fa[t, x] + Mb[t, x] + (-cB D[Fb[t, x], x] + dBF D[Fb[t, x], {x, 2}]) -
|微分 |微分
lossf Fb[t, x] + beta (Sqrt[(Fb[t, x])^2] - Fb[t, x]) +
|平方根
NeumannValue[0, x == 0] + NeumannValue[0, x == 30],
|ノイマン境界値 |ノイマン境界値
D[Ma[t, x], t] == -rmb Fb[t, x] + rra[t, x] × Resa[t, x] - lossr Ma[t, x] +
|微分
beta (Sqrt[(Ma[t, x])^2] - Ma[t, x]), D[Mb[t, x], t] == -rma Fa[t, x] +
|平方根 |微分
rrb[t, x] × Resb[t, x] - lossr Mb[t, x] + beta (Sqrt[(Mb[t, x])^2] - Mb[t, x]),
|平方根

(*初期分布*) Resa[0, x] == 0, Resb[0, x] == 0, Fa[0, x] == 5 × 10 Fa0[x],
Fb[0, x] == 7 × 10 Fb0[x], Ma[0, x] == 4 × 7 × 10 / (5 × 12) gA[x],
Mb[0, x] == 7 × 10 / (5 × 12) gB[x] (*境界条件*);

(*解法設定*)
sol =
NDSolve[pdes, {Fa, Fb, Ma, Mb, Resa, Resb}, {t, 0, endt}, {x, 0, 30}, MaxStepSize -> 0.5,
|微分方程式の数値解 |最大刻み幅
WorkingPrecision -> MachinePrecision, AccuracyGoal -> 10, PrecisionGoal -> 10];
|作業精度 |機械精度 |確度目標 |精度目標
SetDirectory[$UserDocumentsDirectory]
|ディレクトリの設定 |デフォルトのドキュメントディレクトリ

```

```
DumpSave["Solution.mx", sol] (*読み出しはGet["Solution.mx"]*)
```

定義を書き出す

(*結果のプロット例*)

```
(*Plot3D[Evaluate[Fa[t,x]/.sol],{t,0,15},{x,0,30},PlotRange→All,
```

3Dプロット 評価

プロット範囲

すべて

```
ColorFunction→"TemperatureMap",AxesLabel→{"t","x","Fa"}]*)
```

色関数

軸のラベル

Out[575]=

C:\Users\georg\OneDrive\ドキュメント

Out[576]=

```
{ { { Fa → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ],
  Fb → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ],
  Ma → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ],
  Mb → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ],
  Resa → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ],
  Resb → InterpolatingFunction [  Domain: {{0., 60.}, {0., 30.}} Output: scalar ] ] ] }
```

データが保存されていません。今すぐ保存する 

データが保存されていません。今すぐ保存する 

データが保存されていません。今すぐ保存する 

データが保存されていません。今すぐ保存する 

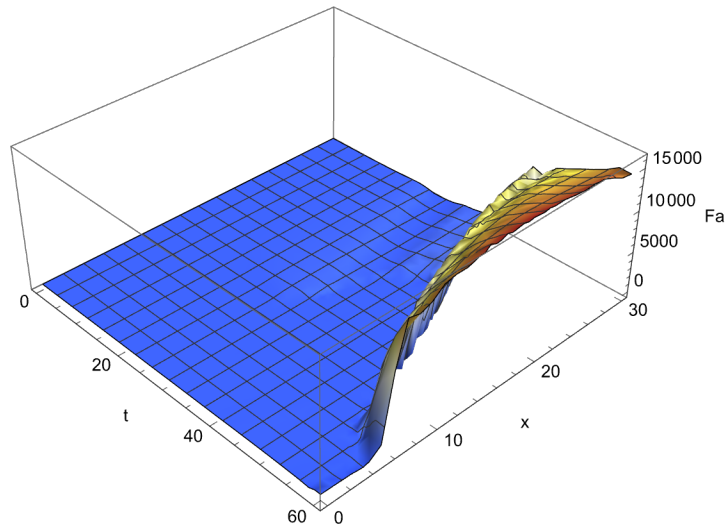
データが保存されていません。今すぐ保存する 

データが保存されていません。今すぐ保存する 

In[577]:=

```
Plot3D[Evaluate[Fa[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange -> All,
3Dプロット 評価 プロット範囲 すべて
ColorFunction -> "TemperatureMap", AxesLabel -> {"t", "x", "Fa"}]
色関数 軸のラベル
```

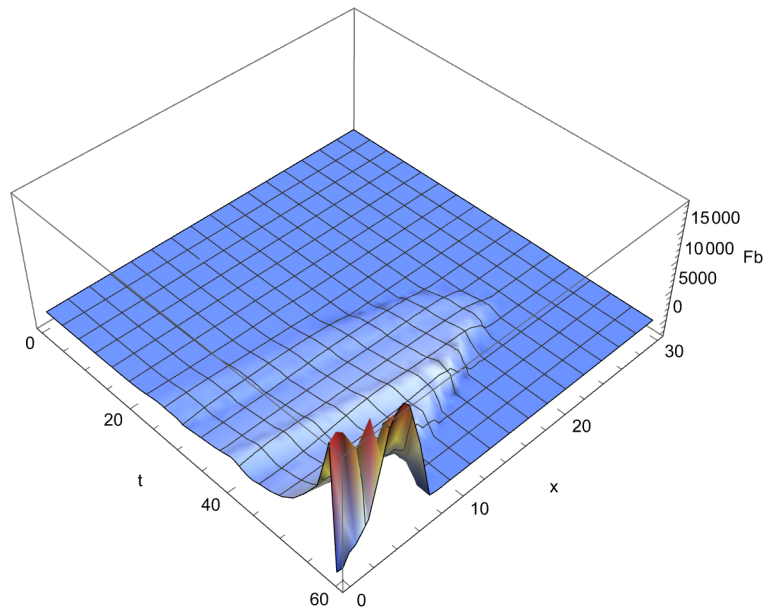
Out[577]:=



In[578]:=

```
Plot3D[Evaluate[Fb[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange -> All,
3Dプロット 評価 プロット範囲 すべて
ColorFunction -> "TemperatureMap", AxesLabel -> {"t", "x", "Fb"}]
色関数 軸のラベル
```

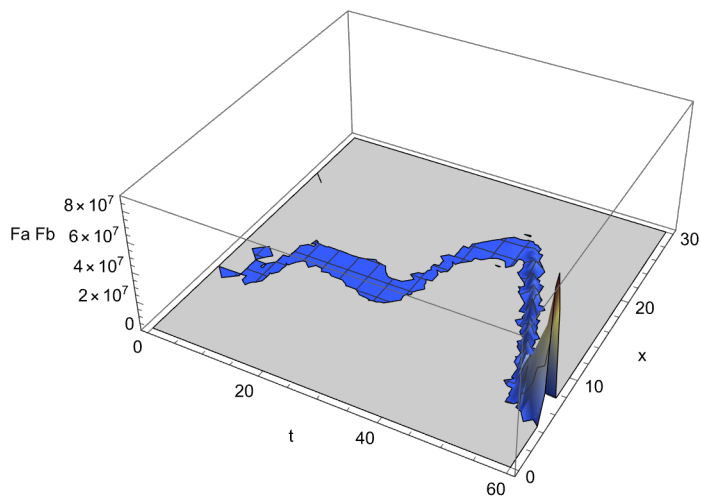
Out[578]:=



In[579]:=

```
Plot3D[Evaluate[(Fa[t, x] × Fb[t, x]) /. sol], {t, 0, endt}, {x, 0, 30},
  3Dプロット 評価
  PlotRange → {0, All}, ColorFunction → "TemperatureMap", AxesLabel → {"t", "x", "Fa Fb"}]
  プロット範囲 すべて 色関数 軸のラベル
```

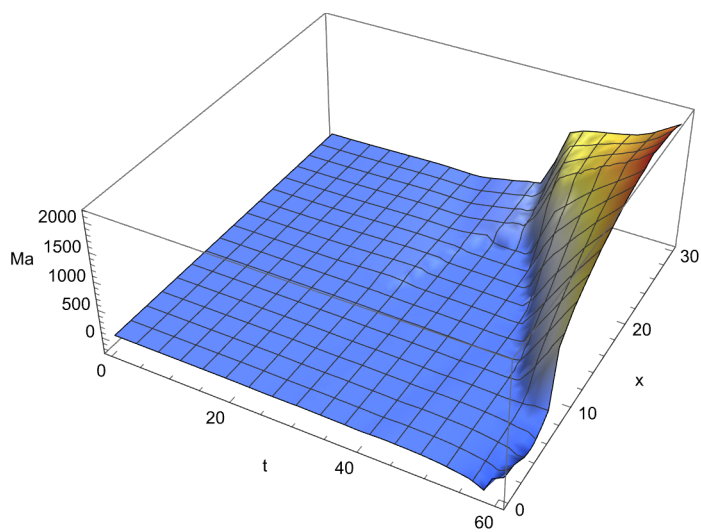
Out[579]=



In[580]:=

```
Plot3D[Evaluate[Ma[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange → All,
  3Dプロット 評価
  ColorFunction → "TemperatureMap", AxesLabel → {"t", "x", "Ma"}]
  色関数 軸のラベル
```

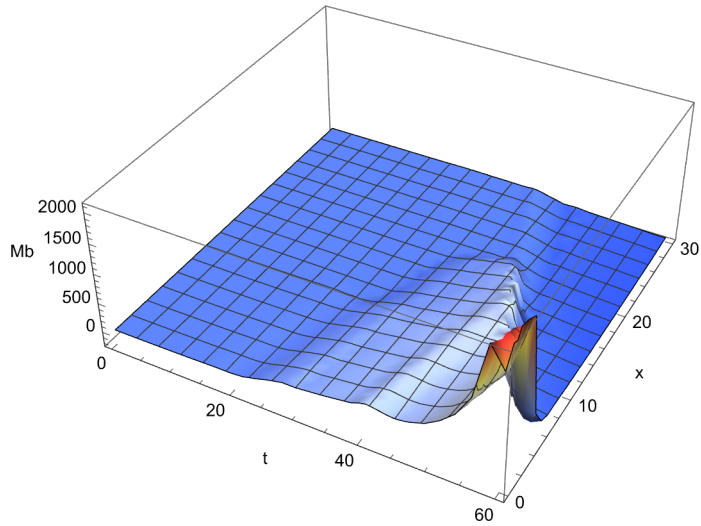
Out[580]=



In[581]:=

```
Plot3D[Evaluate[Mb[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange → All,
3Dプロット | 評価 | プロット範囲 | すべて
ColorFunction → "TemperatureMap", AxesLabel → {"t", "x", "Mb"}]
色関数 | 軸のラベル
```

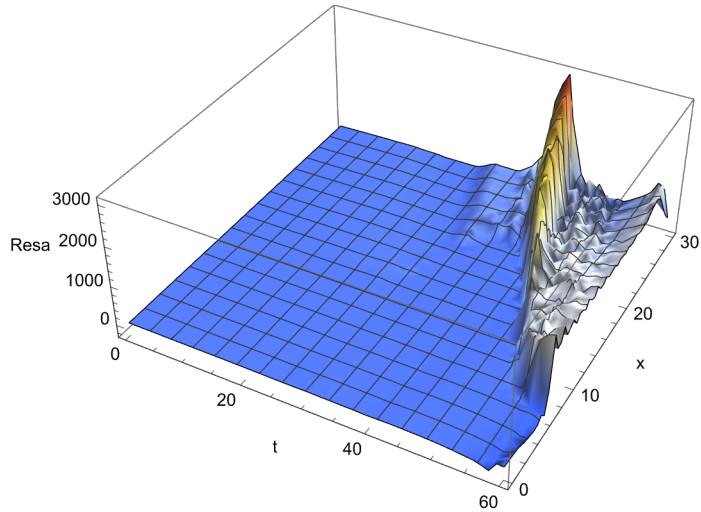
Out[581]=



In[582]:=

```
Plot3D[Evaluate[Resa[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange → All,
3Dプロット | 評価 | プロット範囲 | すべて
ColorFunction → "TemperatureMap", AxesLabel → {"t", "x", "Resa"}]
色関数 | 軸のラベル
```

Out[582]=



In[583]:=

```
Plot3D[Evaluate[Resb[t, x] /. sol], {t, 0, endt}, {x, 0, 30}, PlotRange -> All,  
3Dプロット 評価 プロット範囲 すべて  
ColorFunction -> "TemperatureMap", AxesLabel -> {"t", "x", "Resb"}]  
色関数 軸のラベル
```

Out[583]=

