Deterministické modely růstu a existenčního boje

Modely růstu

```
> unassign('x','t','a','g','k');
```

Nechť $x(t)$ značí velikost růstu populace v čase $\langle T_0, T_1 \rangle$. Předpokládáe, že $x(t)$ plňuje rovnici:

```
> r[1]:=diff(x(t),t)=a*x(t)*g(x(t)/k);
```

$$r_1 := \frac{d}{dt}x(t) = a\,x(t)\,g\left(\frac{x(t)}{k}\right)$$

pro nějkou funkci g a konstanty a a k.
Rovnice r_1 vyjadřuje, jak rychlost růstu populce závisí na její velikosti

Malthusův předpoklad $g(y) = 1$

Za předpokladu, že funkce je konstantní

```
g:=y->1;
```

$$g := y \to 1$$

má rovnice tvar:

```
Malt:=r[1];
```

a její řešení je:

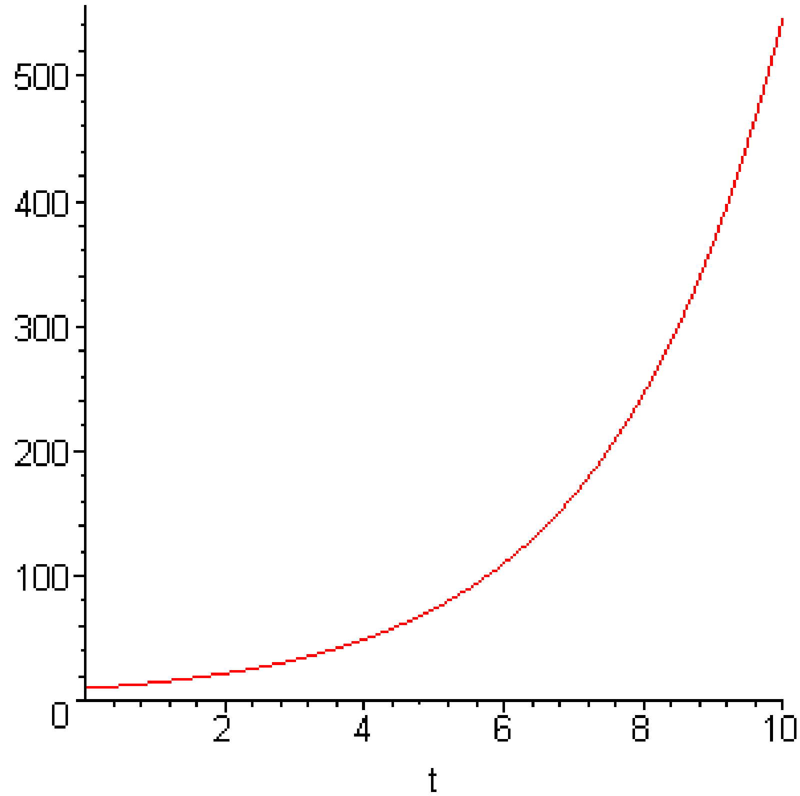
```
M:=dsolve(Malt);
g:='g':
```

$$M := x(t) = \_C1\ \mathbf{e}^{(a\,t)}$$

```
> C1:=solve(subs(t=0,a=.4,k=1,rhs(M))=10);
```

$$C1 := \frac{10}{\mathbf{e}^0}$$

```
> plot(subs(_C1=C1,a=.4,rhs(M)),t=0..10);
```

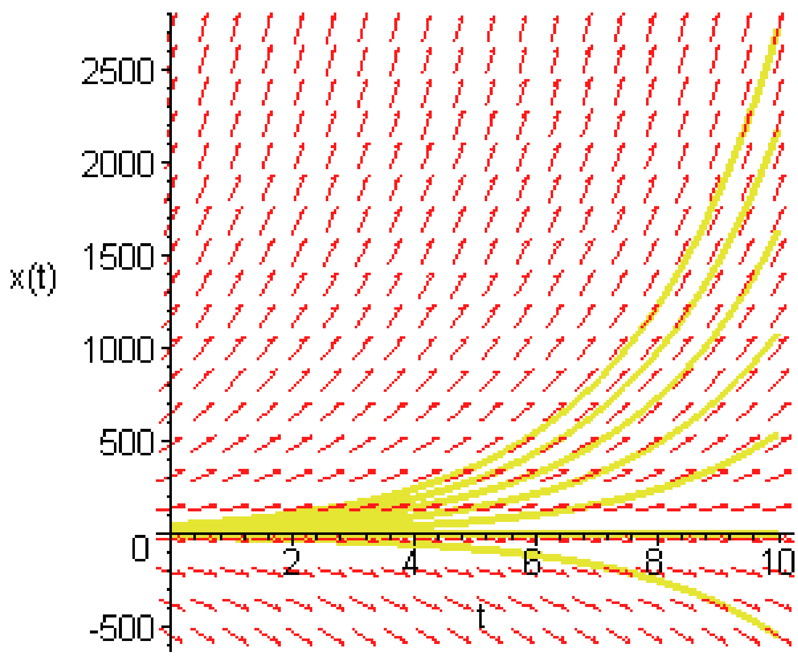


nakreslíme řešení pro různé počáteční podmínky:

```
> with(DEtools);
ivs:=[seq(x(0)=i*10,i=-1..5)];
DEplot( subs(a=.4,Malt), x(t), t=0..10, ivs );
#phaseportrait( subs(a=.4,Malt), x(t), t=0..10, ivs );
```

[*AreSimilar*, *DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *DFactorLCLM*, *DFactorsols*, *Dchangevar*, *FunctionDecomposition*, *GCRD*, *Gosper*, *Heunsols*, *Homomorphisms*, *IsHyperexponential*, *LCLM*, *MeijerGsols*, *MultiplicativeDecomposition*, *PDEchangecoords*, *PolynomialNormalForm*, *RationalCanonicalForm*, *ReduceHyperexp*, *RiemannPsols*, *Xchange*, *Xcommutator*, *Xgauge*, *Zeilberger*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *caseplot*, *casesplit*, *checkrank*, *chinisol*, *clairautsol*, *constcoeffsols*, *convertAlg*, *convertsys*, *dalembertsol*, *dcoeffs*, *de2diffop*, *dfieldplot*, *diff_table*, *diffop2de*, *dperiodic_sols*, *dpolyform*, *dsubs*, *eigenring*, *endomorphism_charpoly*, *equinv*, *eta_k*, *eulersols*, *exactsol*, *expsols*, *exterior_power*, *firint*, *firtest*, *formal_sol*, *gen_exp*, *generate_ic*, *genhomosol*, *gensys*, *hamilton_eqs*, *hypergeomsols*, *hyperode*, *indicialeq*, *infgen*, *initialdata*, *integrate_sols*, *intfactor*, *invariants*, *kovacicsols*, *leftdivision*, *liesol*, *line_int*, *linearsol*, *matrixDE*, *matrix_riccati* , *maxdimsystems* , *moser_reduce* , *muchange* , *mult* , *mutest* , *newton_polygon* , *normalG2* , *ode_int_y* , *ode_y1* , *odeadvisor* , *odepde* , *parametricsol* , *particularsol* , *phaseportrait* , *poincare* , *polysols* , *power_equivalent* , *ratsols* , *redode* , *reduceOrder* , *reduce_order* , *regular_parts* , *regularsp* , *remove_RootOf* , *riccati_system* , *riccatisol* , *rifread* , *rifsimp* , *rightdivision* , *rtaylor* , *separablesol* , *singularities* , *solve_group* , *super_reduce* , *symgen* , *symmetric_power* , *symmetric_product* , *symtest* , *transinv* , *translate* , *untranslate* , *varparam* , *zoom*]

ivs := [x(0) = -10, x(0) = 0, x(0) = 10, x(0) = 20, x(0) = 30, x(0) = 40, x(0) = 50]



Verhulstův přdpoklad $g(y) = 1 - y$ (logistická křivka)

```
> g:=y->1-y;
Verh:=r[1];
V:=dsolve(Verh);
g:='g':
```

$$g := y \to 1 - y$$

$$Verh := \frac{d}{dt}\mathrm{x}(t) = a\,\mathrm{x}(t)\left(1 - \frac{\mathrm{x}(t)}{k}\right)$$

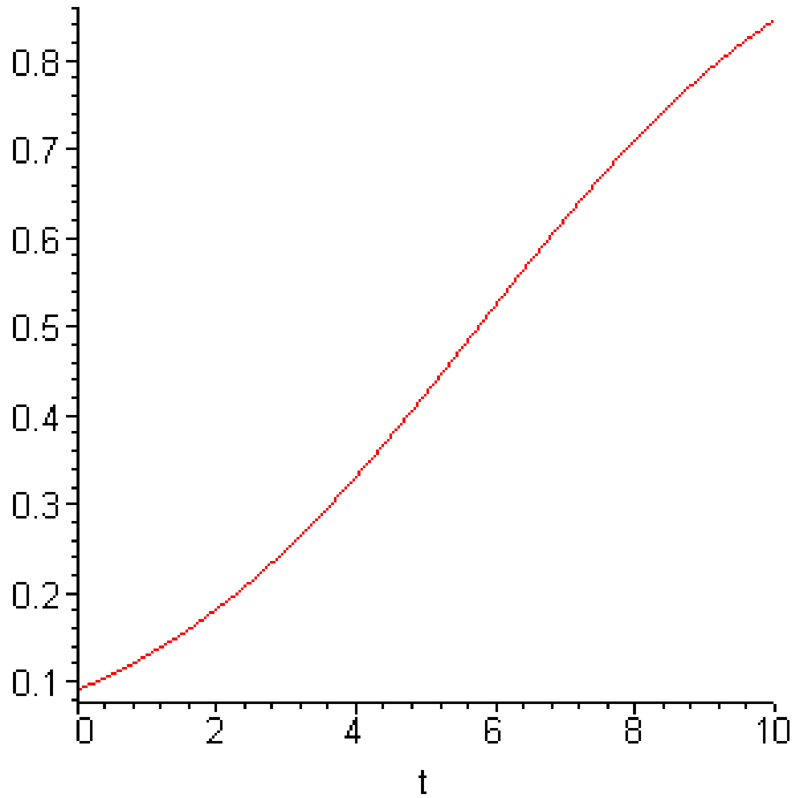
$$V := x(t) = \frac{k}{1 + e^{(-a\,t)}\ \_C1\ k}$$

H. Hotelling, stanovil v roce 1940 hodnotu parametr; a=0.031239, b=67.5352, k= 2900.235729 pro po4et obzvatel v USA.

```
> C1:=solve(subs(t=0,a=.4,k=1,rhs(V))=1);
```

$$C1 := 0$$

```
> plot(subs(_C1=10,a=.4,k=1,rhs(V)),t=0..10);
```




```
> InlexniBod=solve(
simplify(diff(rhs(V),t,t))=0
,t);
```

$$InlexniBod = -\frac{\ln\left(\frac{1}{\_C1\ k}\right)}{a}$$

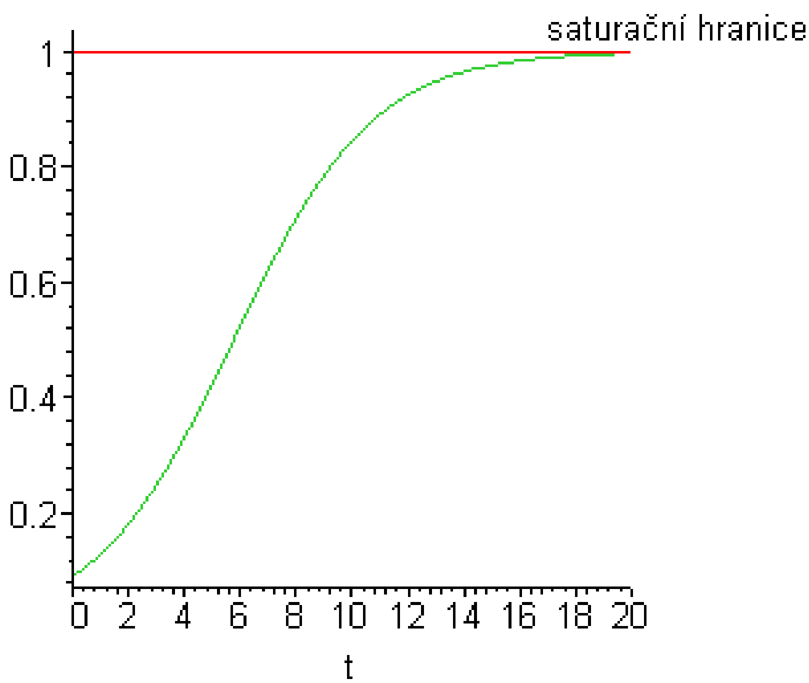
```
> assume(k>0,a>0,_C1>0);
#with(RealDomain);
Limit(rhs(V),t=infinity)=limit(rhs(V),t=infinity);
```

$$\lim_{t \to \infty} \frac{k\sim}{1 + e^{(-a\sim\,t)}\ \_C1\sim\ k\sim} = k\sim$$

```
> with(plots);
p1:=plot(subs(_C1=10,a=.4,k=1,{rhs(V),k}),t=0..20):
p2:=textplot([17,1.02,"saturační hranice"],align={above,right}):
```

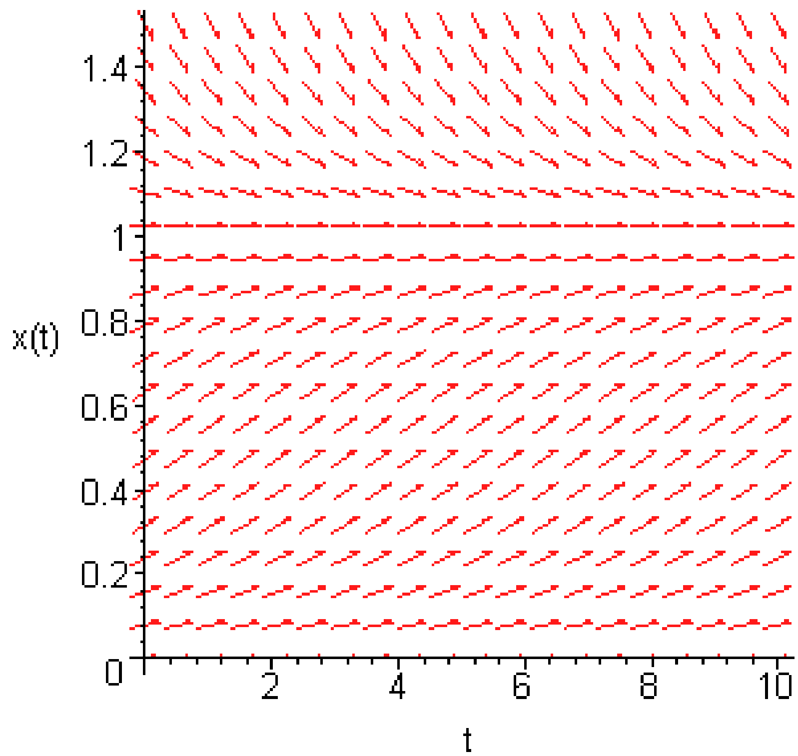
```
display({p1,p2});
unassign('k','a','_C1');
```

[*animate* , *animate3d* , *animatecurve* , *arrow* , *changecoords* , *complexplot* , *complexplot3d* , *conformal* , *conformal3d* , *contourplot* , *contourplot3d* , *coordplot* , *coordplot3d* , *densityplot* , *display* , *fieldplot* , *fieldplot3d* , *gradplot* , *gradplot3d* , *graphplot3d* , *implicitplot* , *implicitplot3d* , *inequal* , *interactive* , *interactiveparams* , *intersectplot* , *listcontplot* , *listcontplot3d* , *listdensityplot* , *listplot* , *listplot3d* , *loglogplot* , *logplot* , *matrixplot* , *multiple* , *odeplot* , *pareto* , *plotcompare* , *pointplot* , *pointplot3d* , *polarplot* , *polygonplot* , *polygonplot3d* , *polyhedra_supported* , *polyhedraplot* , *rootlocus* , *semilogplot* , *setcolors* , *setoptions* , *setoptions3d* , *spacecurve* , *sparsematrixplot* , *surfdata* , *textplot* , *textplot3d* , *tubeplot*]




```
> ivs:=[seq(x(0)=i*.1,i=0..15)];
DEplot( subs(a=.4,k=1,Verh), x(t), t=0..10, ivs, animatecurves =
true );
```

ivs := [x(0) = 0., x(0) = 0.1, x(0) = 0.2, x(0) = 0.3, x(0) = 0.4, x(0) = 0.5, x(0) = 0.6, x(0) = 0.7, x(0) = 0.8, x(0) = 0.9, x(0) = 1.0, x(0) = 1.1, x(0) = 1.2, x(0) = 1.3, x(0) = 1.4, x(0) = 1.5]



>

Gompertzův předpoklad $g(y) = -\ln(y)$

```
> g:=y->-ln(y);
with(PDEtools,declare);
declare(x(t),prime=t);;
Gomp:=r[1];
infolevel[dsolve] := 3;
#odeadvisor(Gomp, help);
G:=simplify(dsolve(Gomp));
g:='g':
```

$$g := y \to -\ln(y)$$

$$[declare]$$

$x(t)$ *will now be displayed as* x

derivatives with respect to t

of functions of one variable will now be displayed with '

$$Gomp := x' = -a\, x \ln\left(\frac{x}{k}\right)$$

$$infolevel_{dsolve} := 3$$

```
Methods for first order ODEs:
--- Trying classification methods ---
trying a quadrature
```

```
trying 1st order linear
trying Bernoulli
trying separable
<- separable successful
```

$$G := x = e^{\left(e^{(-a\,(t + \_C1\,))}\right)}\,k$$

```
> C1:=solve(subs(t=0,a=.4,k=1,rhs(G))=.2);
```

$$C1 := -1.189712488 - 7.853981634\,I$$

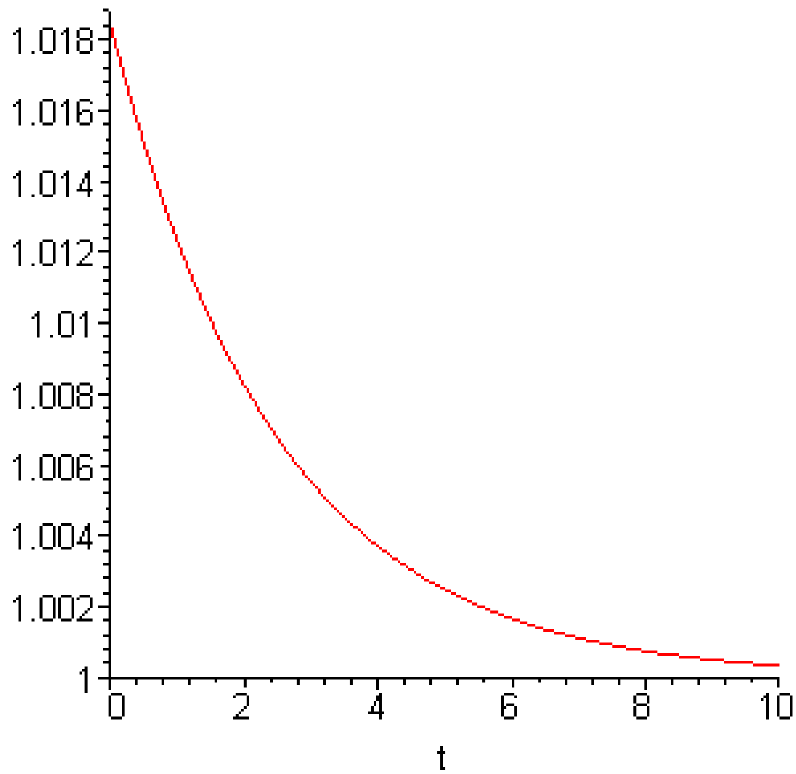
```
> with(RealDomain);
G2:=dsolve({Gomp,x(0)=.1});
```

[$\Im$, $\Re$, ^, arccos , arccosh , arccot , arccoth , arccsc , arccsch , arcsec , arcsech , arcsin , arcsinh , arctan , arctanh , cos , cosh , cot , coth , csc , csch , *eval* , exp , *expand* , *limit* , ln , log , sec , sech , signum , *simplify* , sin , sinh , *solve* , sqrt , *surd* , tan , tanh]

```
Methods for first order ODEs:
--- Trying classification methods ---
trying a quadrature
trying 1st order linear
trying Bernoulli
trying separable
<- separable successful
```

$$G2 := x = e^{\left(\frac{\ln\left(\frac{1}{10\,k}\right) + 2\,I\,\pi\,\_Z18\sim}{e^{(a\,t)}}\right)}\,k$$

```
> plot(subs(_C1=10,a=.4,k=1,_Z2=10,{rhs(G),rhs(G)}),t=0..10);
```

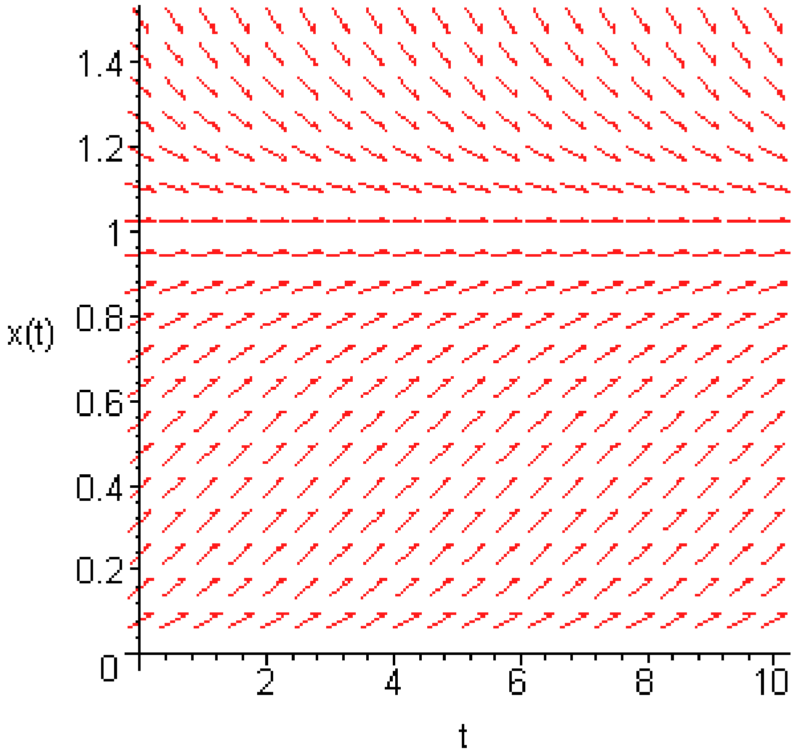



```
> infolevel[dsolve] := 0;
ivs:=[seq(x(0)=i*.1,i=0..15)];
DEplot( subs(a=.4,k=1,Gomp), x(t), t=0..10, ivs, animatecurves =
true );
```

$$infolevel_{dsolve} := 0$$

$$ivs := [x(0) = 0., x(0) = 0.1, x(0) = 0.2, x(0) = 0.3, x(0) = 0.4, x(0) = 0.5, x(0) = 0.6, x(0) = 0.7, x(0) = 0.8, x(0) = 0.9, x(0) = 1.0, x(0) = 1.1, x(0) = 1.2, x(0) = 1.3, x(0) = 1.4, x(0) = 1.5]$$

```
Warning, plot may be incomplete, the following errors(s) were issued:
   cannot evaluate the solution past the initial point, problem may be complex,
initially singular or improperly set up
```




```
> xxx:=subs(a=.4,k=1,Gomp);
zzz:=dsolve({xxx,x(0)=.1},numeric);
```

$$xxx := x' = -0.4\, x \ln(x)$$

$$zzz := \mathbf{proc}(x\_rkf45\ )\ \ldots\ \mathbf{end\ proc}$$

```
> f:=t->op(2,zzz(t));
```

$$f := t \rightarrow \mathrm{op}(2, \mathrm{zzz}(t))$$

```
> plot(rhs(zzz(t)[2]),t=0..10);
Error, invalid input: rhs received zzz(t)[2], which is not valid for its 1st
argument, expr

> aaa:=rhs(zzz(t)[2]);
> 
Error, invalid input: rhs received zzz(t)[2], which is not valid for its 1st
argument, expr

> plot(aaa,t=0..10);
Warning, unable to evaluate the function to numeric values in the region; see the
plotting command's help page to ensure the calling sequence is correct

Plotting error, empty plot
> 
```

Dvě populace soupeřící o obživu

```
> with(DEtools);unassign('x','t','a','k','c','_r');
```

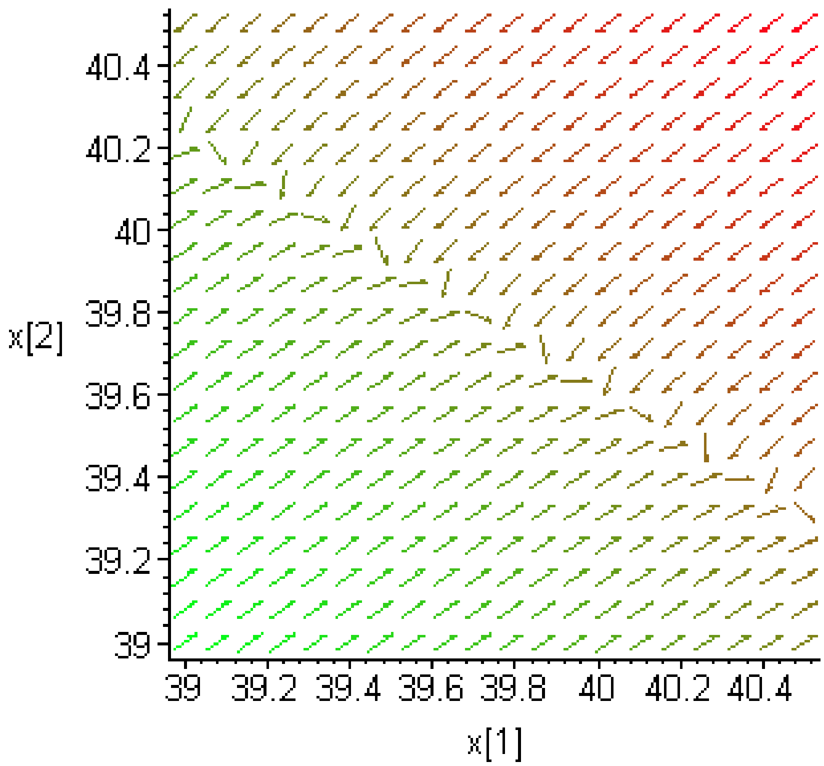
[*AreSimilar*, *DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *DFactorLCLM*, *DFactorsols*, *Dchangevar*, *FunctionDecomposition*, *GCRD*, *Gosper*, *Heunsols*, *Homomorphisms*, *IsHyperexponential*, *LCLM*, *MeijerGsols*, *MultiplicativeDecomposition*, *PDEchangecoords*, *PolynomialNormalForm*, *RationalCanonicalForm*, *ReduceHyperexp*, *RiemannPsols*, *Xchange*, *Xcommutator*, *Xgauge*, *Zeilberger*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *caseplot*, *casesplit*, *checkrank*, *chinisol*, *clairautsol*, *constcoeffsols*, *convertAlg*, *convertsys*, *dalembertsol*, *dcoeffs*, *de2diffop*, *dfieldplot*, *diff_table*, *diffop2de*, *dperiodic_sols*, *dpolyform*, *dsubs*, *eigenring*, *endomorphism_charpoly*, *equinv*, *eta_k*, *eulersols*, *exactsol*, *expsols*, *exterior_power*, *firint*, *firtest*, *formal_sol*, *gen_exp*, *generate_ic*, *genhomosol*, *gensys*, *hamilton_eqs*, *hypergeomsols*, *hyperode*, *indicialeq*, *infgen*, *initialdata*, *integrate_sols*, *intfactor*, *invariants*, *kovacicsols*, *leftdivision*, *liesol*, *line_int*, *linearsol*, *matrixDE*, *matrix_riccati*, *maxdimsystems*, *moser_reduce*, *muchange*, *mult*, *mutest*, *newton_polygon*, *normalG2*, *ode_int_y*, *ode_y1*, *odeadvisor*, *odepde*, *parametricsol*, *particularsol*, *phaseportrait*, *poincare*, *polysols*, *power_equivalent*, *ratsols*, *redode*, *reduceOrder*, *reduce_order*, *regular_parts*, *regularsp*, *remove_RootOf*, *riccati_system*, *riccatisol*, *rifread*, *rifsimp*, *rightdivision*, *rtaylor*, *separablesol*, *singularities*, *solve_group*, *super_reduce*, *symgen*, *symmetric_power*, *symmetric_product*, *symtest*, *transinv*, *translate*, *untranslate*, *varparam*, *zoom*]

```
> _r[1]:=diff(x[1](t),t)=a[1]*x[1](t)*(k[1]-x[1](t)-c[1]*x[2](t))/k[1];
_r[2]:=diff(x[2](t),t)=a[2]*x[2](t)*(k[2]-x[2](t)-c[2]*x[1](t))/k[2];
```

$$\_r_1 := \frac{d}{dt} x_1(t) = \frac{a_1\, x_1(t)\, (k_1 - x_1(t) - c_1\, x_2(t))}{k_1}$$

$$\_r_2 := \frac{d}{dt} x_2(t) = \frac{a_2\, x_2(t)\, (k_2 - x_2(t) - c_2\, x_1(t))}{k_2}$$

```
> dfieldplot(subs(k[1]=105,k[2]=64,a[1]=1.124,a[2]=.794,c[1]=1.64,c[2]=.61,[_r[1],_r[2]]),
[x[1](t),x[2](t)],t=-2..2, x[1]=39..40.5, x[2]=39..40.5,
arrows=SMALL,
title=``, color=[.3*x[2](t)*(x[1](t)-1),x[1](t)*(1-x[2](t)),.1]);
```




```
> A:='A';B:='B';
xxx:=solve({k[2]=A,k[1]/c[1]=B,k[2]/c[2]=E,k[1]=F},{k[1],k[2],c[1
],c[2]});
```

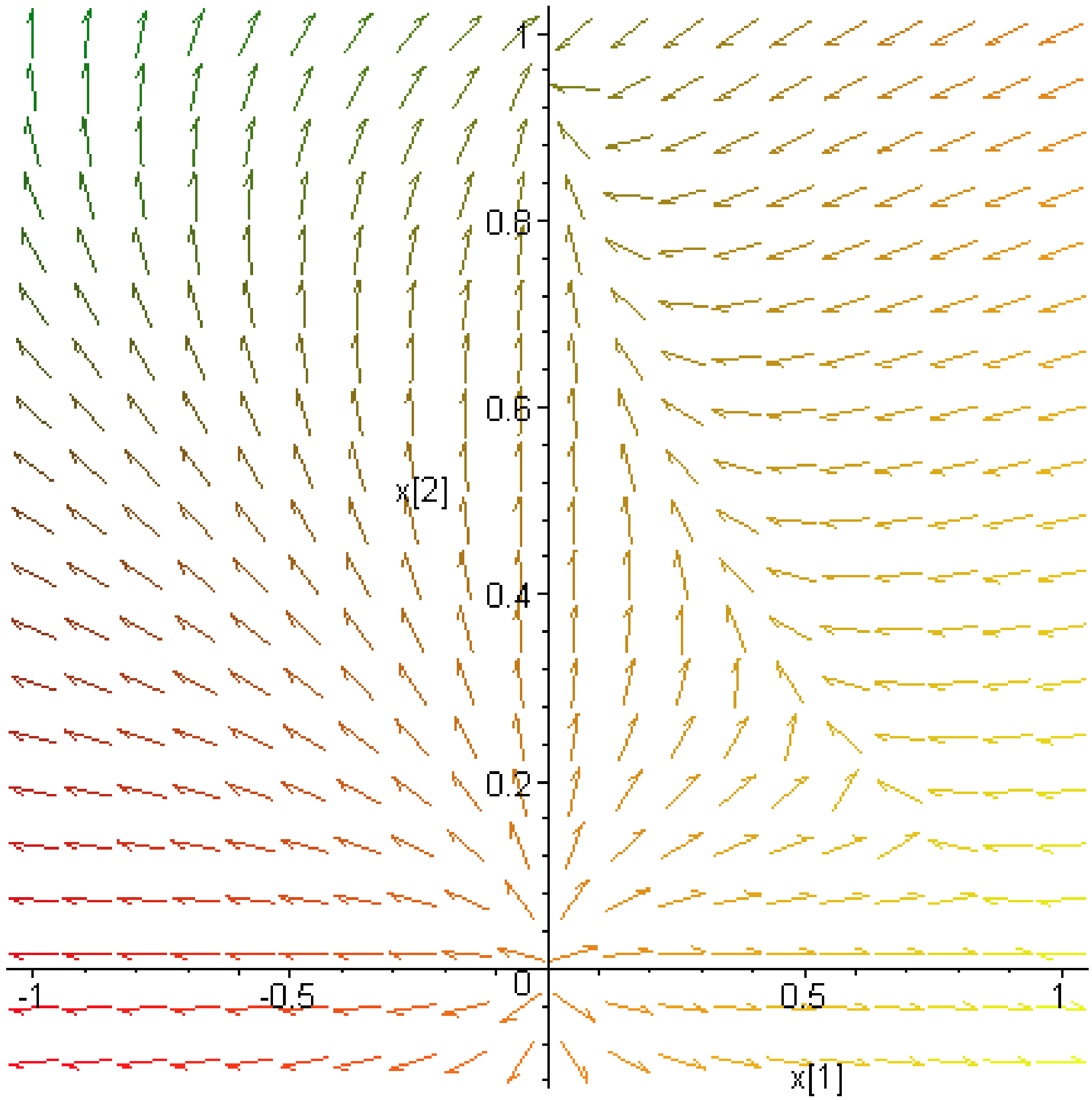
$$A := A$$

$$B := B$$

$$xxx := \{c_2 = \frac{A}{E}, c_1 = \frac{F}{B}, k_2 = A, k_1 = F\}$$

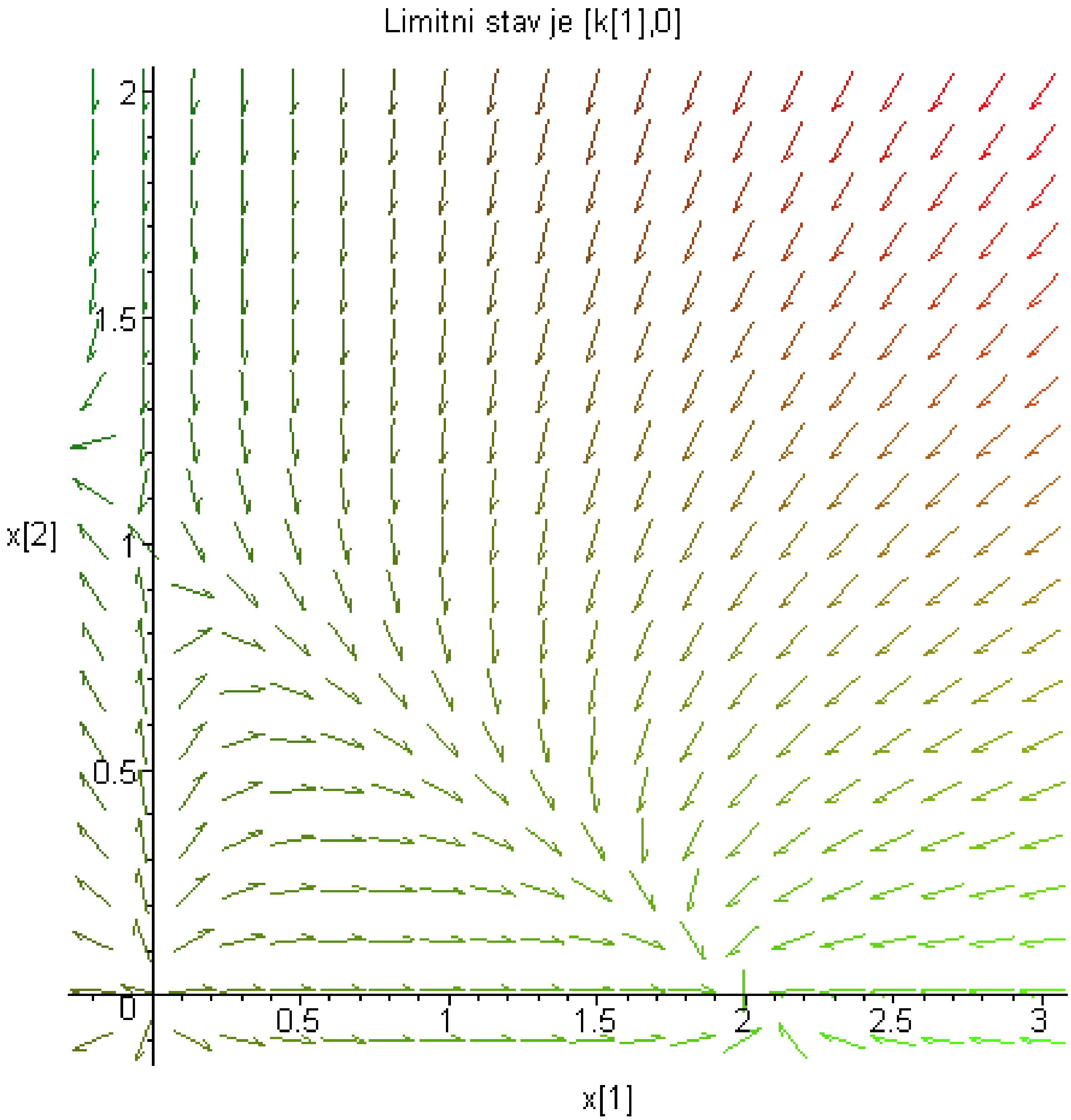
>

$$\{k_2 = 1, c_1 = 1, c_2 = 1, k_1 = 2\}$$

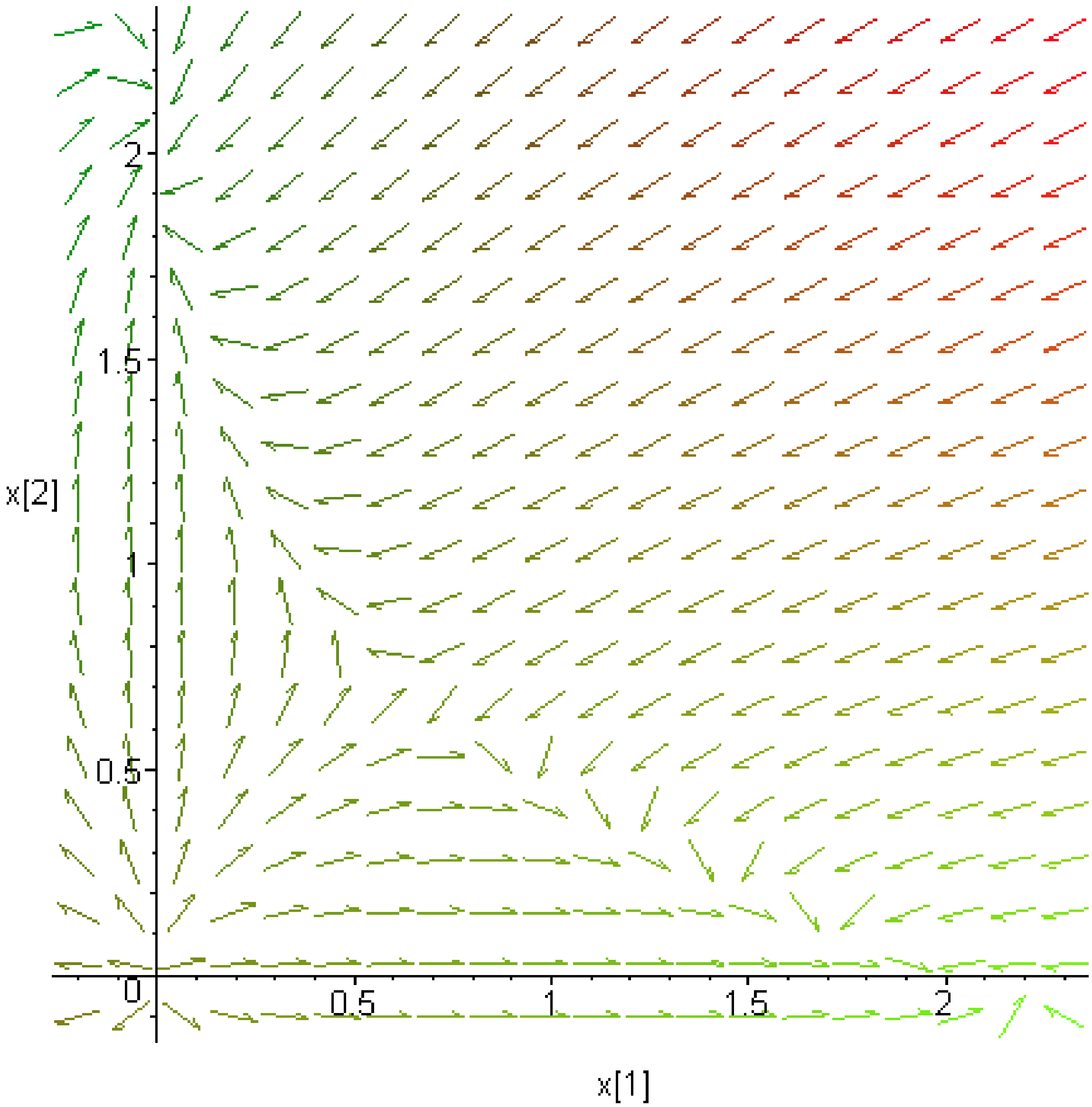


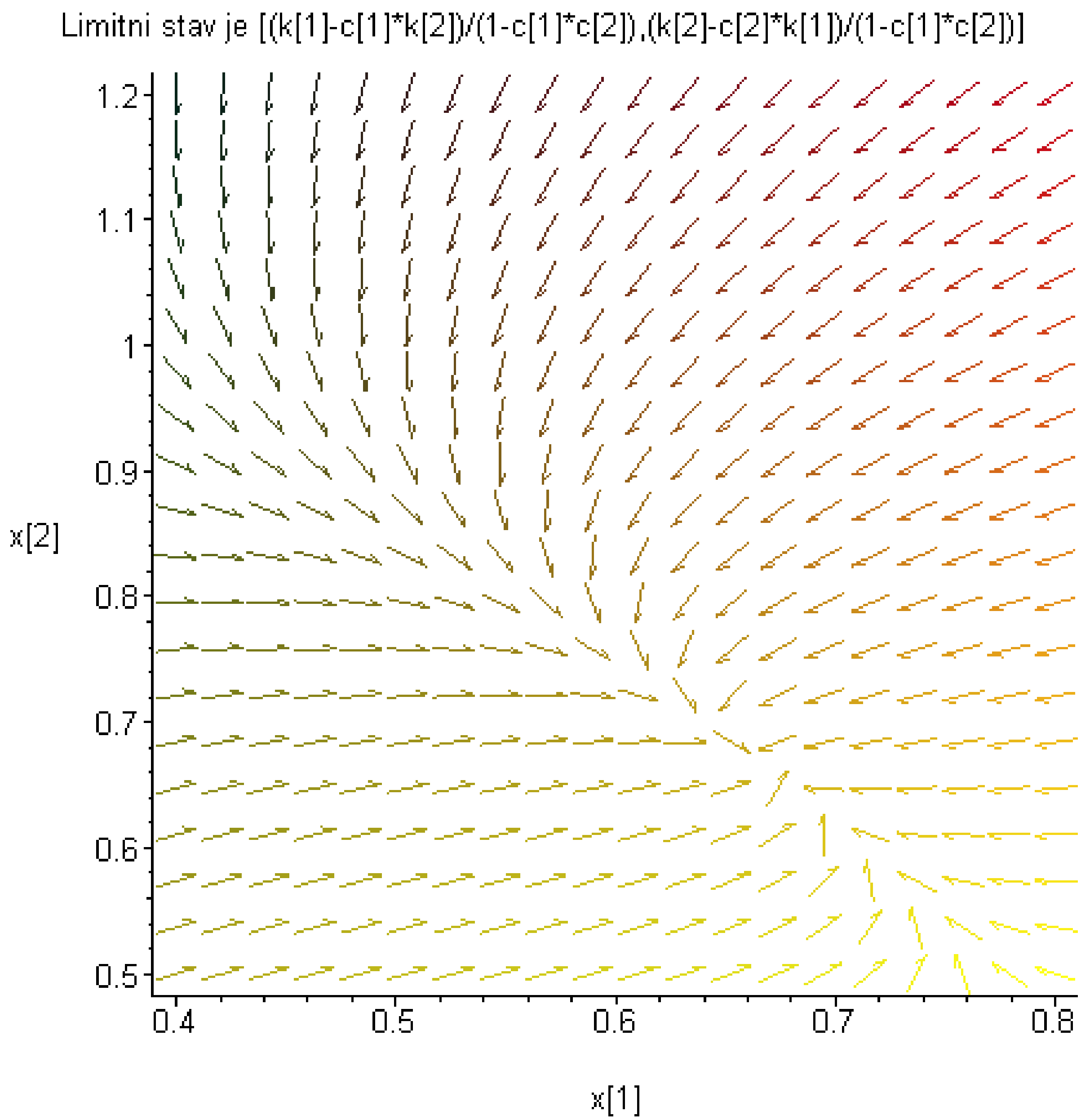

```
> dfieldplot(subs(
subs(A=1,B=2,E=1,F=2,xxx),
a[1]=2.,a[2]=1,
[_r[1],_r[2]]),
[x[1](t),x[2](t)],t=-2..2, x[1]=-.2..3, x[2]=-.1..2, arrows=SMALL,
title=`Limitni stav je [k[1],0]`,
color=[.3*x[2](t)*(x[1](t)-1),x[1](t)*(1-x[2](t)),.1]);

dfieldplot(subs(
subs(A=2,B=1,E=1,F=2,xxx),
a[1]=2.,a[2]=1,
[_r[1],_r[2]]),
[x[1](t),x[2](t)],t=-2..2, x[1]=-.2..2.3, x[2]=-.1..2.3,
```

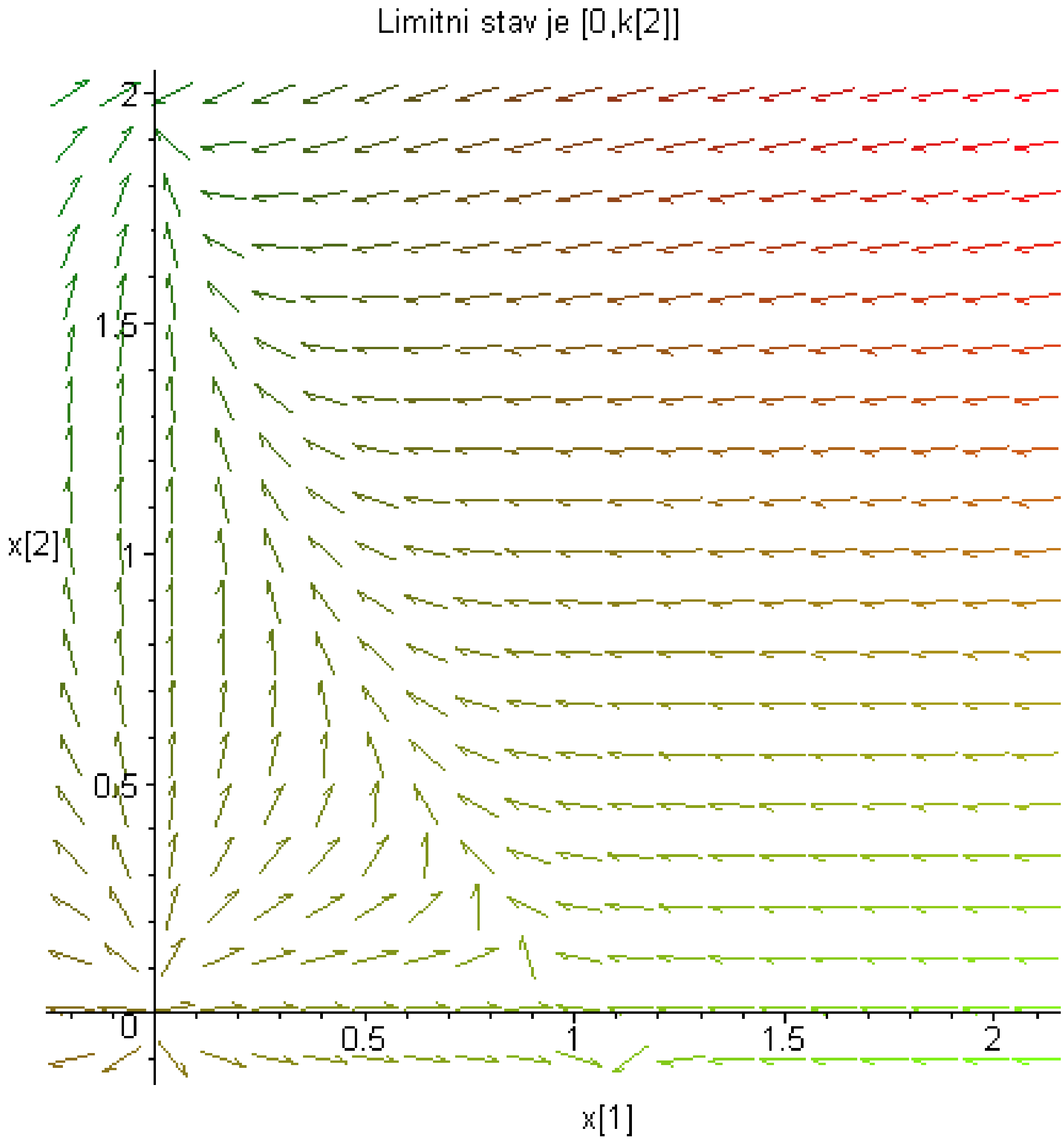
```
arrows=SMALL,
title=`Limitni stavy [k[1],0], nebo [0,k[2]] zaviseji na poc. podm.`,
color=[.3*x[2](t)*(x[1](t)-1),x[1](t)*(1-x[2](t)),.1]);

dfieldplot(subs(
subs(A=1,B=2,E=2,F=1,xxx),
a[1]=2.,a[2]=1,
[_r[1],_r[2]]),
[x[1](t),x[2](t)],t=-2..2, x[1]=.4..0.8, x[2]=0.5..1.2,
arrows=SMALL,
title=`Limitni stav je
[(k[1]-c[1]*k[2])/(1-c[1]*c[2]),(k[2]-c[2]*k[1])/(1-c[1]*c[2])]`,
color=[.3*x[2](t)*(x[1](t)-1),x[1](t)*(1-x[2](t)),.1]);


dfieldplot(subs(
subs(A=2,B=1,E=2,F=1,xxx),
a[1]=2.,a[2]=1,
[_r[1],_r[2]]),
[x[1](t),x[2](t)],t=-.2..2, x[1]=-.2..2.1, x[2]=-.1..2,
arrows=SMALL,
title=`Limitni stav je [0,k[2]]`,
color=[.3*x[2](t)*(x[1](t)-1),x[1](t)*(1-x[2](t)),.1]);
```


Limitni stav je [k[1],0]
x[2]
x[1]
0
0.5
1
1.5
2
2.5
3
0.5
1
1.5
2

Limitni stavy [k[1],0], nebo [0,k[2]] zaviseji na poc. podm.




Limitni stav je [(k[1]-c[1]*k[2])/(1-c[1]*c[2]),(k[2]-c[2]*k[1])/(1-c[1]*c[2])]
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
x[2]
0.4
0.5
0.6
0.7
0.8
x[1]



Limitni stav je [0,k[2]]
x[2]
x[1]
2
1.5
1
0.5
0
0.5
1
1.5
2

>
>
>