```
> restart;
```

Ex.1 - Finding a flow

```
> restart;
```

Assignement

Ex.1: Finding a flow to an ODE

Find a flow to the differential equation $x' = x^p,\ \forall p \in \mathbb{N}$

```
> ode:=diff(x(t),t) = x(t)^p;
```

$$ode := \frac{d}{dt}\, x(t) = x(t)^p \qquad (1.1.1)$$

Note: Conditions on flow

Flow

Let us have a SODE

$$x'(t) = F(x(t))$$

A mapping

$$\varphi : \mathbb{R} \times \mathcal{S} \to \mathcal{S}$$

for which holds the following,

(i) $\varphi(0, x_0) = x_0, \quad \forall x_0 \in \mathcal{S}$

(ii) $\varphi(t, \varphi(s, x_0)) = \varphi(t+s, x_0), \quad \forall s, t \in \mathbb{R},\ \forall x_0 \in \mathcal{S}$

(iii) $\frac{d\varphi_x(t)}{dt} = F(\varphi_x(t)), \quad \forall t \in \mathbb{R},\ \forall x_0 \in \mathcal{S}$

is called a flow associated to the SODE.

```
> F:=x->x^p:#specify the vector field
```

1. p = 1

```
> p:='p':
> p:=1:
```

Find general solution to the ODE

```
> sol:=dsolve(ode,x(t));
  assign(sol):
```

$$sol := x(t) = \_C1\, e^t \qquad (1.2.1)$$

```
> x:=unapply(x(t),t,_C1);#resave as function (better usable)
```

$$x := (t, \_C1) \to \_C1\, e^t \qquad (1.2.2)$$

Specify _C1 in a way, that (i)-(iii) hold

(i) x(0,x0) = x0

```
> x(0,x0);
```

(1.2.3)

$$x0 \qquad (1.2.3)$$

(ii) x(t,x(s,x0)) = x(t+s,x0)

```
> evalb(simplify(x(t,x(s,x0))) = simplify(x(t+s,x0)));
```

true **(1.2.4)**

(iii) is derivative of the solution our vector field?

```
> evalb(D[1](x)(t,x0) = F(x(t,x0)));
```

true **(1.2.5)**

Note: This equations should always hold, as we found x(t,x0) as solution to the differential equation

Note: In this case, we got everything "for free". We should not get used to it.

2. p >=1

```
> p:='p':x:='x':x0:='x0':
> assume(p,'integer',p>0):
> about(p);
```

```
Originally p, renamed p~:
  is assumed to be: AndProp(integer,RealRange(1,infinity))
```

Find general solution to the ODE

```
> sol:=dsolve(ode,x(t));
  assign(sol):
```

$$sol := x(t) = \frac{1}{(-p\sim t + \_C1 + t)^{\frac{1}{p\sim - 1}}} \qquad (1.3.1)$$

```
> x:=unapply(simplify(x(t)),t,_C1);#resave as function (better
  usable)
```

$$x := (t, \_C1) \rightarrow (-p\sim t + \_C1 + t)^{-\frac{1}{p\sim - 1}} \qquad (1.3.2)$$

Specify _C1 in a way, that (i)-(iii) hold

(i) x(0,x0) = x0

```
> eq:=x(0,x0)=x0;
```

$$eq := x0^{-\frac{1}{p\sim - 1}} = x0 \qquad (1.3.3)$$

```
> eq:=simplify(lhs(eq)^((p-1))) = rhs(eq)^((p-1));#solution
  step by step
  eq:=simplify(lhs(eq)*x0) = simplify(rhs(eq)*x0);
  x0:=lhs(eq)^(1/p);
```

$$eq := \frac{1}{x0} = x0^{p\sim - 1}$$

$$eq := 1 = x0^{p\sim}$$

$$x0 := 1 \qquad (1.3.4)$$

```
> x0:='x0':x0:=solve(x(0,x0)=x0,x0);#maple solution
```

$$x0 := 1 \qquad (1.3.5)$$

Note: (i) may hold iff _C1 = 1

(ii) x(t,x(s,x0)) = x(t+s,x0)

```
> eq:=simplify(x(t,x(s,x0))) = simplify(x(t+s,x0));
```

$$eq := \left(-t\,p\sim + (-p\sim s + s + 1)^{-\frac{1}{p\sim -1}} + t\right)^{-\frac{1}{p\sim -1}} = (-p\sim s - p\sim t + s + t + 1)^{-\frac{1}{p\sim -1}} \quad (1.3.6)$$

```
> evalb(simplify(x(t,x(s,x0))) = simplify(x(t+s,x0)));
```

false **(1.3.7)**

Note: (ii) does not hold

(iii) is derivative of the solution our vector field?

```
> evalb(simplify(D[1](x)(t,x0)) = simplify(F(x(t,x0))));
```

true **(1.3.8)**

Note: This equations should always hold, as we found x(t,x0) as solution to the differential equation

Result: The flow associated to the ODE can be found only for the case p = 1.

Ex.2 - Multistep and competitive reactions

```
> restart;
```

Assignement

Ex.2: Competitive reactions

Let us have a reaction scheme,

$$\begin{aligned} a &\xrightarrow{k_1} b \xrightarrow{k_2} c \\ a &\xrightarrow{k_3} e \end{aligned}$$

generating the corresponding differential equations,

$$x' = \begin{pmatrix} a' \\ b' \\ c' \\ e' \end{pmatrix} = \begin{pmatrix} -k_1 a - k_3 a \\ k_1 a - k_2 b \\ k_2 b \\ k_3 a \end{pmatrix} = \begin{pmatrix} -k_1 - k_3 & 0 & 0 & 0 \\ k_1 & -k_2 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ k_3 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ e \end{pmatrix} = Ax$$

Find dependence of contcentrations on time corresponding to the following parameter values,

$$a_0 = 1,\ b_0 = c_0 = e_0 = 0,\ k_1 = 1, k_2 = 1/2, k_3 = 1/10$$

```
> A:=Matrix([[ -k1-k3 , 0 , 0 , 0 ],
            [ k1 , -k2 , 0 , 0 ],
            [ 0 , k2 , 0 , 0 ],
```

```
[ k3 , 0 , 0 , 0 ]]);#system matrix
```

$$A := \begin{bmatrix} -k1 - k3 & 0 & 0 & 0 \\ k1 & -k2 & 0 & 0 \\ 0 & k2 & 0 & 0 \\ k3 & 0 & 0 & 0 \end{bmatrix} \quad (2.1.1)$$

```
> with(LinearAlgebra):
```

1. Eigenvalues

```
> charPol:=CharacteristicPolynomial(A,lambda);#compute the characteristic polynomial
```

$$charPol := \lambda^4 + (k2 + k1 + k3)\,\lambda^3 + k2\,(k1 + k3)\,\lambda^2 \quad (2.2.1)$$

```
> eigValsA:=solve(charPol,lambda);
```

$$eigValsA := 0, 0, -k2, -k1 - k3 \quad (2.2.2)$$

```
> charPol:=Determinant(A-lambda*IdentityMatrix(4,4));#compute the characteristic polynomial in a way we do it
```

$$charPol := -\left(-k1 - k3 - \lambda\right)\left(k2 + \lambda\right)\lambda^2 \quad (2.2.3)$$

```
> eigValsA:=solve(charPol,lambda);#get matrix eigenvalues
```

$$eigValsA := -k2, -k1 - k3, 0, 0 \quad (2.2.4)$$

```
> eigValsA:=Eigenvalues(A);#get matrix eigenvalues by direct maple command
```

$$eigValsA := \begin{bmatrix} 0 \\ 0 \\ -k2 \\ -k1 - k3 \end{bmatrix} \quad (2.2.5)$$

2. Eigenvectors

```
> eigValsA,eigVecsA:=Eigenvectors(A);#by direct Maple command
```

$$eigValsA, eigVecsA := \begin{bmatrix} 0 \\ 0 \\ -k1 - k3 \\ -k2 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -\frac{k1 + k3}{k3} & 0 \\ 0 & 0 & \frac{(k1 + k3)\,k1}{k3\,(k1 + k3 - k2)} & -1 \\ 0 & 1 & -\frac{k1\,k2}{k3\,(k1 + k3 - k2)} & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (2.3.1)$$

From MAPLE help:

With an eigenvalue of multiplicity $k > 1$, there may be fewer than k linearly independent eigenvectors. In this case, the matrix is called *defective*. By design, the returned matrix always has full column dimension. Therefore, in the defective case, some of the columns that are returned are zero. Thus, they are not eigenvectors. With the option, output=list, only

eigenvectors are returned. For more information, see LinearAlgebra[JordanForm] and LinearAlgebra[SchurForm].
=>
We can't be really sure, that we have the real eigenvectors

```
> for i from 1 to numelems(eigValsA) do:
   auxMat:=A-eigValsA[i]*IdentityMatrix(4,4):#substitute the current eigenvalue to the matrix A-lambdaE
   eigValsBasis[i]:=NullSpace(auxMat):#find null space -> solve homogeneous system
  od:
> for i from 1 to numelems(eigValsA) do:#check, if we really found the eigenvectors (does the equation lambda*h = A*h hold?)
   print([op(eigValsBasis[i])*eigValsA[i],A . op(eigValsBasis[i])]);
  od:
```

$$[[0, 0]]$$

$$[[0, 0]]$$

$$\left[\left[\begin{array}{c} -\frac{(-k1-k3)(k1+k3)}{k3} \\ \frac{(-k1-k3)(k1+k3)\,k1}{k3\,(k1+k3-k2)} \\ -\frac{(-k1-k3)\,k1\,k2}{k3\,(k1+k3-k2)} \\ -k1-k3 \end{array}\right], \left[\begin{array}{c} -\frac{(-k1-k3)(k1+k3)}{k3} \\ -\frac{k1\,(k1+k3)}{k3} - \frac{k2\,(k1+k3)\,k1}{k3\,(k1+k3-k2)} \\ \frac{k2\,(k1+k3)\,k1}{k3\,(k1+k3-k2)} \\ -k1-k3 \end{array}\right]\right]$$

$$\left[\left[\begin{array}{c} 0 \\ k2 \\ -k2 \\ 0 \end{array}\right], \left[\begin{array}{c} 0 \\ k2 \\ -k2 \\ 0 \end{array}\right]\right] \quad (2.3.2)$$

```
> eigValsBasis:=convert(
   [seq(op(eigValsBasis[i]),i=1..numelems(eigValsBasis))],
   Matrix
  );#convert the output to a matrix form (better usable)
```

$$eigValsBasis := \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & -\frac{k1+k3}{k3} & 0 \\ 0 & 0 & 0 & 0 & \frac{(k1+k3)\,k1}{k3\,(k1+k3-k2)} & -1 \\ 0 & 1 & 0 & 1 & -\frac{k1\,k2}{k3\,(k1+k3-k2)} & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right] \quad (2.3.3)$$

```
> delList:=[]:#create list for positions of duplicate columns
  for i from 1 to ColumnDimension(eigValsBasis) do:
   for j from i+1 to ColumnDimension(eigValsBasis) do:
     if Equal(eigValsBasis[1..-1,i],eigValsBasis[1..-1,j],
```

```
compare=entries) then:
    delList:=[op(delList),j]:
    fi:
 od:
od:
eigVecsA:=DeleteColumn(eigValsBasis,delList):#resave
eigenvectors in a new matrix
> eigValsA,eigVecsA;
```

$$\left[\begin{array}{c} 0 \\ 0 \\ -k1 - k3 \\ -k2 \end{array}\right], \left[\begin{array}{cccc} 0 & 0 & -\frac{k1 + k3}{k3} & 0 \\ 0 & 0 & \frac{(k1 + k3)\, k1}{k3\,(k1 + k3 - k2)} & -1 \\ 0 & 1 & -\frac{k1\, k2}{k3\,(k1 + k3 - k2)} & 1 \\ 1 & 0 & 1 & 0 \end{array}\right] \quad \textbf{(2.3.4)}$$

3. General solution

Note:
- We found lambda_1 = lambda_2 = 0 ... real root of the characteristic polynomial with *algebraic* multiplicity = 2
BUT
- for this eigenvalue (lambda = 0), dim(N(A-lambdaE)) = 2 -> there exist 2 linearly independent eigenvectors vectors -> *geometric* multiplicity of eigenvalue lambda = 0 is 1
=>
We have 4 real eigenvalues with 4 distinctive eigenvectors
=>
We seek the solution to the system x' = Ax in the form x(t) = Sum(C_j*exp(lambda_j*t)*eigVecsA(j),j=1..4)

```
> cVec:=Vector[column](numelems(eigValsA),symbol=C):#prepare
vector of constants
> xGS:=add(cVec[i]*exp(eigValsA[i]*t)*eigVecsA[1..-1,i],i=1..
numelems(eigValsA));#this is general solution to the system
```

$$xGS := \left[\begin{array}{c} -\frac{C_3\, e^{(-k1 - k3)\, t}\,(k1 + k3)}{k3} \\ \frac{C_3\, e^{(-k1 - k3)\, t}\,(k1 + k3)\, k1}{k3\,(k1 + k3 - k2)} - C_4\, e^{-k2\, t} \\ C_2 - \frac{C_3\, e^{(-k1 - k3)\, t}\, k1\, k2}{k3\,(k1 + k3 - k2)} + C_4\, e^{-k2\, t} \\ C_1 + C_3\, e^{(-k1 - k3)\, t} \end{array}\right] \quad \textbf{(2.4.1)}$$

4. Particular solution - initial value problem (IVP)

```
> IC:=a0 = 1,b0 = 0, c0 = 0, e0 = 0:
> x0:=Vector[column]([seq(rhs(IC[i]),i=1..numelems([IC]))]);
  #combine the initial conditions to a vector
```

$$x0 := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.5.1}$$

```
> eqs:=[seq(x0[i]=subs(t=0,xGS[i]),i=1..numelems(x0))];
```

$$eqs := \left[1 = -\frac{C_3 \, e^0 \, (k1 + k3)}{k3}, 0 = \frac{C_3 \, e^0 \, (k1 + k3) \, k1}{k3 \, (k1 + k3 - k2)} - C_4 \, e^0, 0 = C_2 - \frac{C_3 \, e^0 \, k1 \, k2}{k3 \, (k1 + k3 - k2)} + C_4 \, e^0, 0 = C_1 + C_3 \, e^0\right] \tag{2.5.2}$$

```
> B,b:=GenerateMatrix(eqs,convert(cVec,list));
```

$$B, b := \begin{bmatrix} 0 & 0 & \frac{k1 + k3}{k3} & 0 \\ 0 & 0 & -\frac{(k1 + k3) \, k1}{k3 \, (k1 + k3 - k2)} & 1 \\ 0 & -1 & \frac{k1 \, k2}{k3 \, (k1 + k3 - k2)} & -1 \\ -1 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.5.3}$$

```
> cVec:=LinearSolve(B,b);
```

$$cVec := \begin{bmatrix} \frac{k3}{k1 + k3} \\ \frac{k1}{k1 + k3} \\ -\frac{k3}{k1 + k3} \\ -\frac{k1}{k1 + k3 - k2} \end{bmatrix} \tag{2.5.4}$$

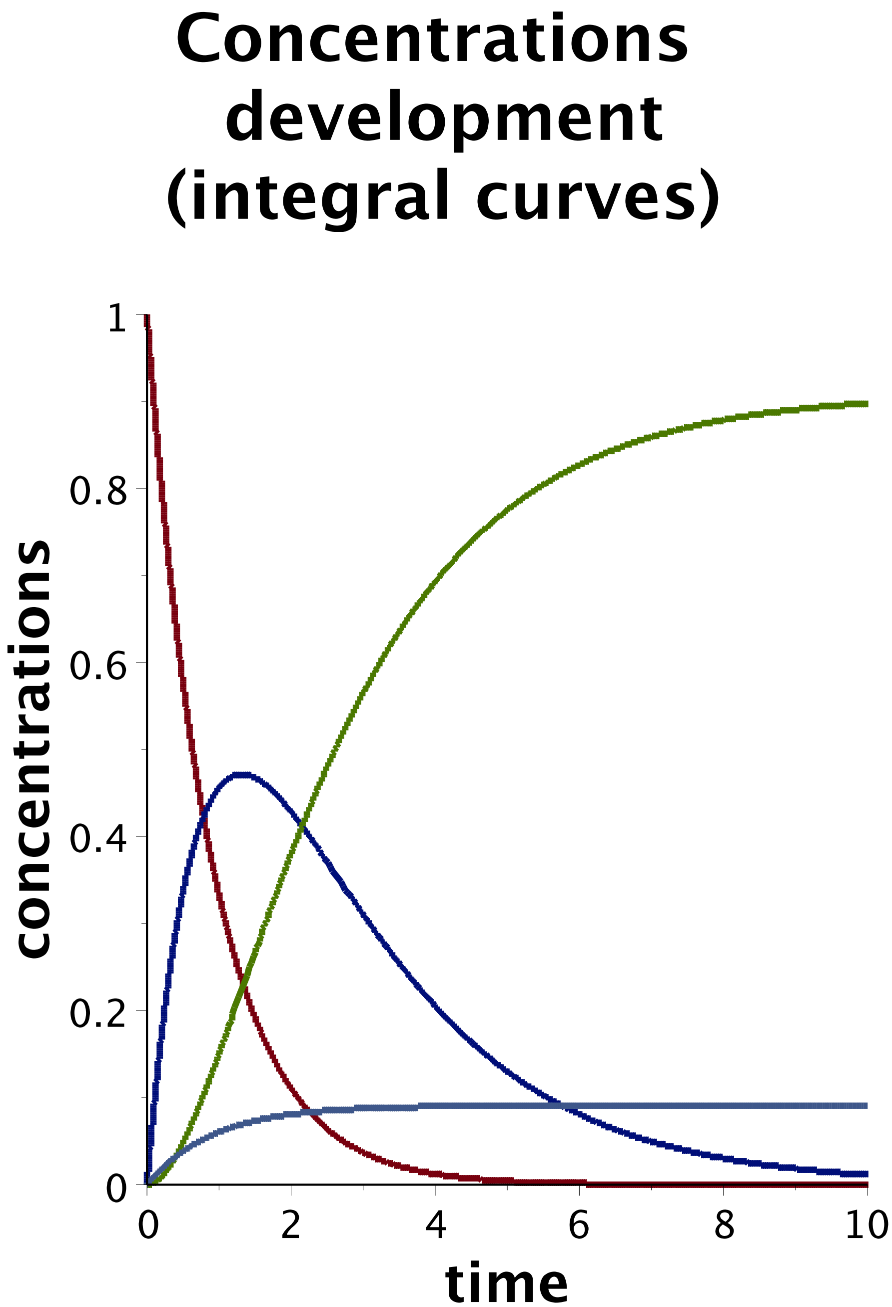
```
> xPS:=add(cVec[i]*exp(eigValsA[i]*t)*eigVecsA[1..-1,i],i=1..
  numelems(eigValsA));#this is particular solution to the
  system
```

(2.5.5)

$$xPS := \begin{bmatrix} e^{(-k1-k3)\,t} \\ -\frac{e^{(-k1-k3)\,t}\,k1}{k1+k3-k2} + \frac{k1\,e^{-k2\,t}}{k1+k3-k2} \\ \frac{k1}{k1+k3} + \frac{e^{(-k1-k3)\,t}\,k1\,k2}{(k1+k3)\,(k1+k3-k2)} - \frac{k1\,e^{-k2\,t}}{k1+k3-k2} \\ \frac{k3}{k1+k3} - \frac{k3\,e^{(-k1-k3)\,t}}{k1+k3} \end{bmatrix} \quad \textbf{(2.5.5)}$$

Specify reaction rate constants and plot the solution (compare with the plot bellow)

```
> plot(
    subs(k1=1,k2=1/2,k3=1/10,xPS),t=0..10,
    thickness=3,
    legend=["a(t)","b(t)","c(t)","e(t)"],
    legendstyle=[font=["HELVETICA", 15],location=bottom],
    labels=["time","concentrations"],
    labeldirections=["horizontal","vertical"],
    labelfont=["Helvetica",20,Bold],
    title="Concentrations development\n(integral curves)",
    titlefont=["Helvetica",24,Bold],
    axesfont=["Helvetica",14],
    numpoints=5000,size=[800,600]
  );
```



5. Test - solve the system using Maple built-in methods

```
> xVec:=Vector[column]([x1(t),x2(t),x3(t),x4(t)]);#vector of uknown functions
```

$$xVec := \begin{bmatrix} x1(t) \\ x2(t) \\ x3(t) \\ x4(t) \end{bmatrix} \quad (2.6.1)$$

```
> systRHS:=MatrixVectorMultiply(A,xVec);
```

$$systRHS := \begin{bmatrix} (-k1 - k3)\, x1(t) \\ k1\, x1(t) - k2\, x2(t) \\ k2\, x2(t) \\ k3\, x1(t) \end{bmatrix} \quad (2.6.2)$$

```
> for i from 1 to 4 do:
   ode[i]:=diff(xVec[i],t)=systRHS[i]:
  od:
> ode:=convert(ode,list):
  xVec:=convert(xVec,list):
> IC:=x1(0)=a0,x2(0)=b0,x3(0)=c0,x4(0)=e0:
> sol:=dsolve([op(ode), IC],xVec):#Such a system is "always"
  analyticaly solvable
> assign(sol);
> sol:
> x1(t);
  x2(t);
  x3(t);
  x4(t);
```

$$a0\, e^{-(k1 + k3)\, t}$$

$$\frac{\left(-\frac{k1\, k2\, k3\, a0\, e^{-(k1+k3)\, t + k2\, t}}{k1 + k3 - k2} + \frac{k2\, (a0\, k1 + b0\, k1 - b0\, k2 + b0\, k3)\, k3}{k1 + k3 - k2}\right) e^{-k2\, t}}{k3\, k2}$$

$$- \frac{1}{(k1 + k3 - k2)\, k3\, k2} \left(- \frac{k2^2\, e^{-(k1+k3)\, t}\, k3\, a0\, k1}{k1 + k3} + \frac{e^{-k2\, t}\, k2\, (a0\, k1 + b0\, k1 - b0\, k2 + b0\, k3)\, k1\, k3}{k1 + k3 - k2} - \frac{k2^2\, (a0\, k1 + b0\, k1 - b0\, k2 + b0\, k3)\, e^{-k2\, t}\, k3}{k1 + k3 - k2} + \frac{k2\, (a0\, k1 + b0\, k1 - b0\, k2 + b0\, k3)\, k3^2\, e^{-k2\, t}}{k1 + k3 - k2} - \frac{(a0\, k1 + b0\, k1 + b0\, k3 + c0\, k1 + c0\, k3)\, k1\, k3\, k2}{k1 + k3} + \frac{k2^2\, (a0\, k1 + b0\, k1 + b0\, k3 + c0\, k1 + c0\, k3)\, k3}{k1 + k3} - \frac{k3^2\, (a0\, k1 + b0\, k1 + b0\, k3 + c0\, k1 + c0\, k3)\, k2}{k1 + k3} \right)$$

(2.6.3)

$$\frac{a0\,k3 + e0\,k1 + e0\,k3}{k1 + k3} - \frac{k3\,a0\,e^{-(k1 + k3)\,t}}{k1 + k3} \tag{2.6.3}$$

```
> ICP:=a0=1,b0=0,c0=0,e0=0,k1=1,k2=1/2,k3=1/10:
> aP:=subs(ICP,x1(t));
  bP:=subs(ICP,x2(t));
  cP:=subs(ICP,x3(t));
  eP:=subs(ICP,x4(t));
```

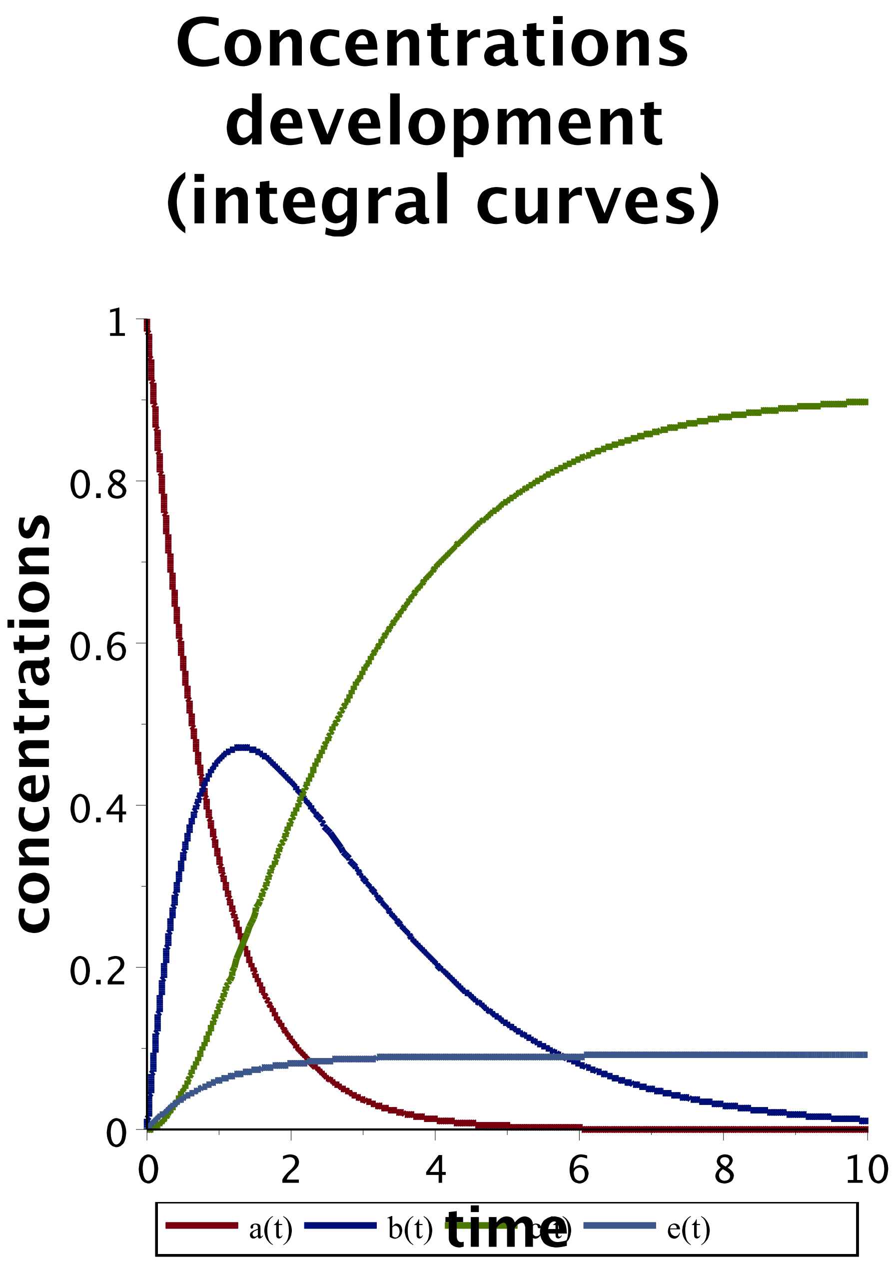
$$aP := e^{-\frac{11}{10}\,t}$$

$$bP := 20\left(-\frac{1}{12}\,e^{-\frac{3}{5}\,t} + \frac{1}{12}\right)e^{-\frac{1}{2}\,t}$$

$$cP := \frac{10}{11} + \frac{25}{33}\,e^{-\frac{11}{10}\,t} - \frac{5}{3}\,e^{-\frac{1}{2}\,t}$$

$$eP := \frac{1}{11} - \frac{1}{11}\,e^{-\frac{11}{10}\,t} \tag{2.6.4}$$

```
> plot(
   [aP,bP,cP,eP],t=0..10,
   thickness=3,
   legend=["a(t)","b(t)","c(t)","e(t)"],
   legendstyle=[font=["HELVETICA", 15],location=bottom],
   labels=["time","concentrations"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="Concentrations development\n(integral curves)",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
   numpoints=5000,size=[800,600]
  ) ;
```



Ex.3 - Real, non-equal eigenvalues

```
> restart;
```

Assignement

Assignement

Solve the Cauchy problem,

$$x' = Ax, \quad A = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix}, \quad x(t=0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

```
> with(LinearAlgebra):
> A:=Matrix([[ 2 , 3 ],
            [ 1 , 4 ]]);
```

$$A := \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \quad (3.1.1)$$

1. Eigenvalues

```
> charPol:=lambda^2-Trace(A)*lambda + Determinant(A);
```

$$charPol := \lambda^2 - 6\lambda + 5 \quad (3.2.1)$$

```
> eigValsA:=solve(charPol,lambda);
```

$$eigValsA := 5, 1 \quad (3.2.2)$$

```
> eigValsA:=convert([eigValsA],Vector):#convert the output to a vector form (better usability)
```

2. Eigenvectors

```
> for i from 1 to numelems(eigValsA) do:
   auxMat:=A-eigValsA[i]*IdentityMatrix(Dimension(A)):
   eigValsBasis[i]:=NullSpace(auxMat):#find null space -> solve homogeneous system
  od:
> for i from 1 to numelems(eigValsA) do:#check, if we really found the eigenvectors (does the equation lambda*h = A*h hold?)
   print([op(eigValsBasis[i])*eigValsA[i],A . op(eigValsBasis[i])]);
  od:
```

$$\left[\begin{bmatrix} 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 5 \end{bmatrix}\right]$$

$$\left[\begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -3 \\ 1 \end{bmatrix}\right] \quad (3.3.1)$$

```
> eigValsBasis:=convert(
   [seq(op(eigValsBasis[i]),i=1..numelems(eigValsBasis))],
   Matrix
  );#convert the output to a matrix form (better usability)
```

(3.3.2)

$$eigValsBasis := \begin{bmatrix} 1 & -3 \\ 1 & 1 \end{bmatrix} \quad \textbf{(3.3.2)}$$

==

the following step is just a precaution - its usability becames apparent from the Ex2 and Ex5

```
> delList:=[]:#create list for positions of duplicate columns
  for i from 1 to ColumnDimension(eigValsBasis) do:
   for j from i+1 to ColumnDimension(eigValsBasis) do:
     if Equal(eigValsBasis[1..-1,i],eigValsBasis[1..-1,j],
  compare=entries) then:
       delList:=[op(delList),j]:
       fi:
   od:
  od:
  eigVecsA:=DeleteColumn(eigValsBasis,delList):#resave
  eigenvectors in a new matrix
```

==

3. General solution

```
> cVec:=Vector[column](numelems(eigValsA),symbol=C):#prepare
  vector of constants
> xGS:=add(cVec[i]*exp(eigValsA[i]*t)*eigVecsA[1..-1,i],i=1..
  numelems(eigValsA));
```

$$xGS := \begin{bmatrix} C_1 e^{5t} - 3\, C_2 e^{t} \\ C_1 e^{5t} + C_2 e^{t} \end{bmatrix} \quad \textbf{(3.4.1)}$$

4. Particular solution - initial value problem (IVP)

```
> x0:=Vector[column]([1,0]):#specify the initial condition
> eqs:=[seq(x0[i]=subs(t=0,xGS[i]),i=1..numelems(x0))];
```

$$eqs := \left[1 = C_1 e^{0} - 3\, C_2 e^{0},\ 0 = C_1 e^{0} + C_2 e^{0}\right] \quad \textbf{(3.5.1)}$$

```
> B,b:=GenerateMatrix(eqs,convert(cVec,list));
```

$$B, b := \begin{bmatrix} -1 & 3 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \textbf{(3.5.2)}$$

```
> cVec:=LinearSolve(B,b);
```

$$cVec := \begin{bmatrix} \frac{1}{4} \\ -\frac{1}{4} \end{bmatrix} \quad \textbf{(3.5.3)}$$

```
> xPS:=add(cVec[i]*exp(eigValsA[i]*t)*eigVecsA[1..-1,i],i=1..
  numelems(eigValsA));
```

(3.5.4)

$$xPS := \begin{bmatrix} \frac{1}{4} e^{5t} + \frac{3}{4} e^{t} \\ \frac{1}{4} e^{5t} - \frac{1}{4} e^{t} \end{bmatrix} \quad (3.5.4)$$

5. Test - solve the system using Maple built-in methods

```
> xVec:=Vector[column]([x1(t),x2(t)]):
> systRHS:=MatrixVectorMultiply(A,xVec);
```

$$systRHS := \begin{bmatrix} 2\,x1(t) + 3\,x2(t) \\ x1(t) + 4\,x2(t) \end{bmatrix} \quad (3.6.1)$$

```
> for i from 1 to numelems(xVec) do:
   ode[i]:=diff(xVec[i],t)=systRHS[i]:
  od:
> ode:=convert(ode,list):
  xVec:=convert(xVec,list):
> print(ode);
```

$$\left[\frac{d}{dt}\,x1(t) = 2\,x1(t) + 3\,x2(t),\ \frac{d}{dt}\,x2(t) = x1(t) + 4\,x2(t) \right] \quad (3.6.2)$$

```
> sol:=dsolve(ode,xVec):
> xGS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

$$xGS\_MAPLE := \begin{bmatrix} -3\,\_C1\,e^{t} + \_C2\,e^{5t} \\ \_C1\,e^{t} + \_C2\,e^{5t} \end{bmatrix} \quad (3.6.3)$$

```
> IC:=x1(0)=1,x2(0)=0:
> sol:=dsolve([op(ode),IC],xVec):
> xPS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

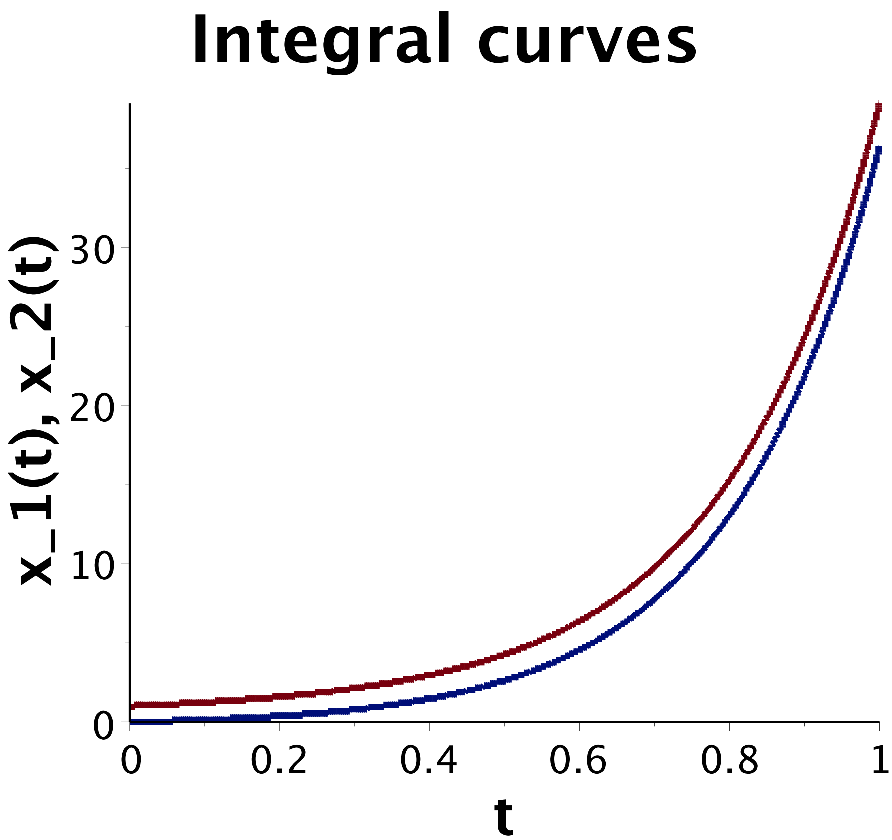
$$xPS\_MAPLE := \begin{bmatrix} \frac{1}{4} e^{5t} + \frac{3}{4} e^{t} \\ \frac{1}{4} e^{5t} - \frac{1}{4} e^{t} \end{bmatrix} \quad (3.6.4)$$

6. Plot resulting integral curve and trajectory

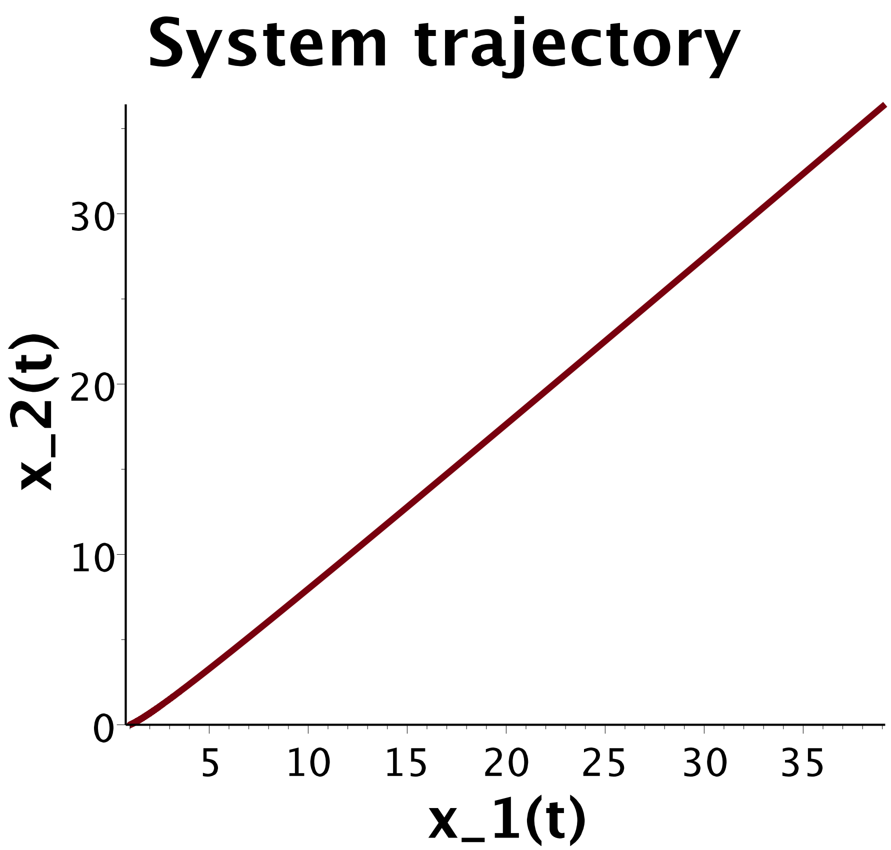
Note: All the possible trajectories -> *phase portrait*

```
> plot(xPS,t=0..1,
   thickness=3,
   legend=["x1(t)","x2(t)"],
   legendstyle=[font=["HELVETICA", 15],location=bottom],
   labels=["t","x_1(t), x_2(t)"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="Integral curves",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
```

```
  numpoints=5000,size=[600,400]
 );
```




```
> plot([op(convert(xPS,list)),t=0..1],
  thickness=3,
  labels=["x_1(t)","x_2(t)"],
  labeldirections=["horizontal","vertical"],
  labelfont=["Helvetica",20,Bold],
  title="System trajectory",
  titlefont=["Helvetica",24,Bold],
  axesfont=["Helvetica",14],
  numpoints=5000,size=[600,400]
 );
```



Ex.4 - Complex eigenvalues

```
> restart;
```

Assignement

Assignement

Solve the Cauchy problem,

$$x' = Ax, \quad A = \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}, \quad x(t=0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

```
> with(LinearAlgebra):
> A:=Matrix([[ 2 , -1 ],
            [ 1 , 2 ]]);
```

$$A := \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \qquad (4.1.1)$$

1. Eigenvalues

```
> charPol:=lambda^2-Trace(A)*lambda + Determinant(A);
```

$$charPol := \lambda^2 - 4\,\lambda + 5 \qquad (4.2.1)$$

```
> eigValsA:=solve(charPol,lambda);
```

$$eigValsA := 2 + I, 2 - I \qquad (4.2.2)$$

```
> eigValsA:=convert([eigValsA],Vector):#convert the output to a vector form (better usability)
```

2. Eigenvectors

```
> for i from 1 to numelems(eigValsA) do:
   auxMat:=A-eigValsA[i]*IdentityMatrix(Dimension(A)):
   eigValsBasis[i]:=NullSpace(auxMat):#find null space -> solve homogeneous system
  od:
> for i from 1 to numelems(eigValsA) do:#check, if we really found the eigenvectors (does the equation lambda*h = A*h hold?)
   print([op(eigValsBasis[i])*eigValsA[i],A . op(eigValsBasis[i])]);
  od:
```

$$\left[\left[\begin{array}{c} -1+2\,I \\ 2+I \end{array}\right], \left[\begin{array}{c} -1+2\,I \\ 2+I \end{array}\right]\right]$$

$$\left[\left[\begin{array}{c} -1-2\,I \\ 2-I \end{array}\right], \left[\begin{array}{c} -1-2\,I \\ 2-I \end{array}\right]\right] \qquad (4.3.1)$$

```
> eigValsBasis:=convert(
   [seq(op(eigValsBasis[i]),i=1..numelems(eigValsBasis))],
   Matrix
  );#convert the output to a matrix form (better usability)
```

$$eigValsBasis := \left[\begin{array}{cc} I & -I \\ 1 & 1 \end{array}\right] \qquad (4.3.2)$$

Note: Finding eigenvector is enough, the other one is its complex conjugate

===

the following step is just a precaution - its usability becames apparent from the Ex2 and Ex5

```
> delList:=[]:#create list for positions of duplicate columns
  for i from 1 to ColumnDimension(eigValsBasis) do:
   for j from i+1 to ColumnDimension(eigValsBasis) do:
     if Equal(eigValsBasis[1..-1,i],eigValsBasis[1..-1,j],compare=entries) then:
      delList:=[op(delList),j]:
      fi:
   od:
  od:
  eigVecsA:=DeleteColumn(eigValsBasis,delList):#resave eigenvectors in a new matrix
```

===

Note: I have two complex eigenvectors in a form h_{1,2} = u +- iv => I need to separate the vectors u and v

```
> realEigVecsA:=<Re(eigVecsA[1..-1,1])|Im(eigVecsA[1..-1,1])>; #construct a matrix [u,v]
```

$$realEigVecsA := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4.3.3)$$

3. General solution

```
> cVec:=Vector[column](numelems(eigValsA),symbol=C):#prepare vector of constants
> xGS:=exp(Re(eigValsA[1])*t)*(cVec[1]*(realEigVecsA[1..-1,1]*cos(Im(eigValsA[1])*t) - realEigVecsA[1..-1,2]*sin(Im(eigValsA[1])*t)) + cVec[2]*(realEigVecsA[1..-1,2]*cos(Im(eigValsA[1])*t) + realEigVecsA[1..-1,1]*sin(Im(eigValsA[1])*t)));
```

$$xGS := \begin{bmatrix} e^{2t}\left(-C_1 \sin(t) + C_2 \cos(t)\right) \\ e^{2t}\left(C_1 \cos(t) + C_2 \sin(t)\right) \end{bmatrix} \quad (4.4.1)$$

4. Particular solution - initial value problem (IVP)

```
> x0:=Vector[column]([1,0]):#specify the initial condition
> eqs:=[seq(x0[i]=subs(t=0,xGS[i]),i=1..numelems(x0))];
```

$$eqs := \left[1 = e^0\left(-C_1 \sin(0) + C_2 \cos(0)\right), 0 = e^0\left(C_1 \cos(0) + C_2 \sin(0)\right)\right] \quad (4.5.1)$$

```
> B,b:=GenerateMatrix(eqs,convert(cVec,list));
```

$$B, b := \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (4.5.2)$$

```
> cVec:=LinearSolve(B,b);
```

$$cVec := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.5.3)$$

```
> xPS:=exp(Re(eigValsA[1])*t)*(cVec[1]*(realEigVecsA[1..-1,1]*cos(Im(eigValsA[1])*t) - realEigVecsA[1..-1,2]*sin(Im(eigValsA[1])*t)) + cVec[2]*(realEigVecsA[1..-1,2]*cos(Im(eigValsA[1])*t) + realEigVecsA[1..-1,1]*sin(Im(eigValsA[1])*t)));
```

$$xPS := \begin{bmatrix} e^{2t}\cos(t) \\ e^{2t}\sin(t) \end{bmatrix} \quad (4.5.4)$$

5. Test - solve the system using Maple built-in methods

```
> xVec:=Vector[column]([x1(t),x2(t)]):
```

```
> systRHS:=MatrixVectorMultiply(A,xVec);
```

$$systRHS := \begin{bmatrix} 2\,x1(t) - x2(t) \\ x1(t) + 2\,x2(t) \end{bmatrix} \quad (4.6.1)$$

```
> for i from 1 to numelems(xVec) do:
   ode[i]:=diff(xVec[i],t)=systRHS[i]:
  od:
> ode:=convert(ode,list):
  xVec:=convert(xVec,list):
> print(ode);
```

$$\left[\frac{d}{dt}\,x1(t) = 2\,x1(t) - x2(t),\ \frac{d}{dt}\,x2(t) = x1(t) + 2\,x2(t)\right] \quad (4.6.2)$$

```
> sol:=dsolve(ode,xVec):
> xGS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

$$xGS\_MAPLE := \begin{bmatrix} e^{2t}\left(\cos(t)\,\_C1 - \sin(t)\,\_C2\right) \\ e^{2t}\left(\_C2\cos(t) + \_C1\sin(t)\right) \end{bmatrix} \quad (4.6.3)$$

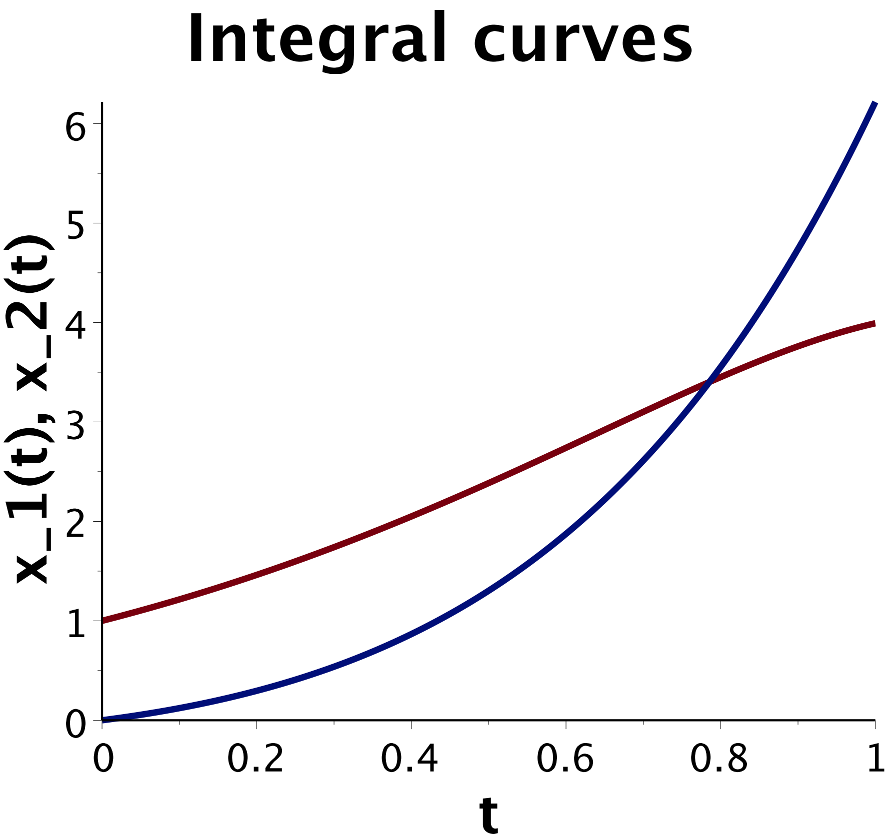
```
> IC:=x1(0)=1,x2(0)=0:
> sol:=dsolve([op(ode),IC],xVec):
> xPS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

$$xPS\_MAPLE := \begin{bmatrix} e^{2t}\cos(t) \\ e^{2t}\sin(t) \end{bmatrix} \quad (4.6.4)$$

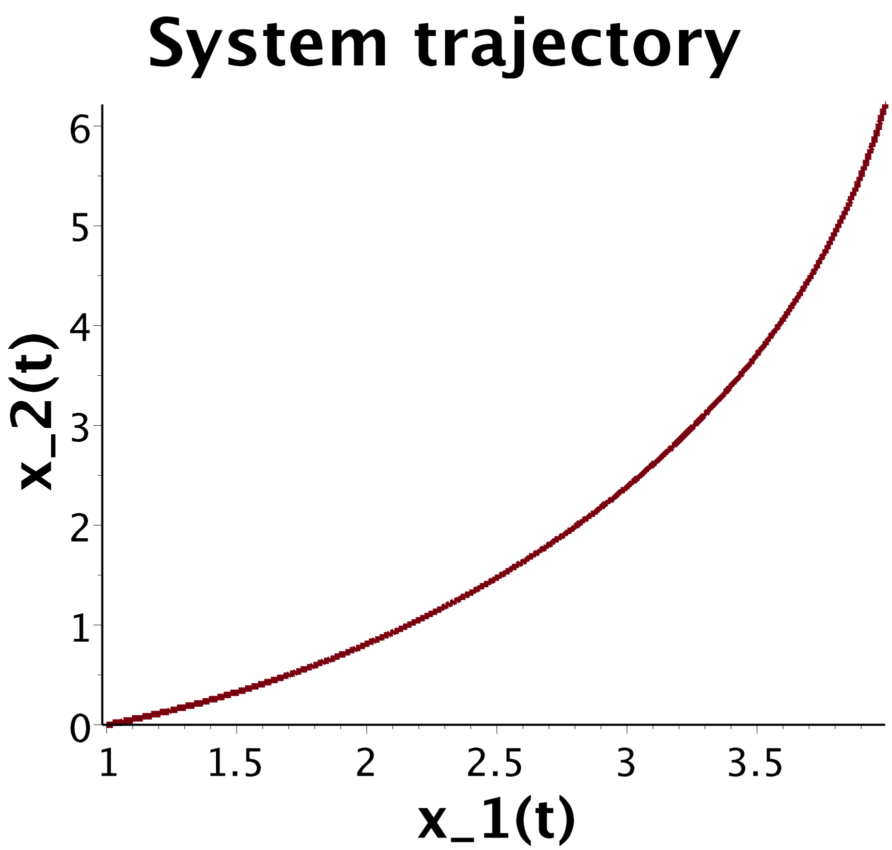
6. Plot resulting integral curve and trajectory

Note: All the possible trajectories -> *phase portrait*

```
> plot(xPS,t=0..1,
   thickness=3,
   legend=["x1(t)","x2(t)"],
   legendstyle=[font=["HELVETICA", 15],location=bottom],
   labels=["t","x_1(t), x_2(t)"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="Integral curves",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
   numpoints=5000,size=[600,400]
  ) ;
```




```
> plot([op(convert(xPS,list)),t=0..1],
   thickness=3,
   labels=["x_1(t)","x_2(t)"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="System trajectory",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
   numpoints=5000,size=[600,400]
  ) ;
```



Ex.5 - Real, equal eigenvalues - generalized eigenvector

```
> restart;
```

Assignement

Assignement

Solve the Cauchy problem,

$$x' = Ax, \quad A = \begin{pmatrix} 3 & 1 \\ -1 & 5 \end{pmatrix}, \quad x(t=0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

```
> with(LinearAlgebra):
> A:=Matrix([[ 3 , 1 ],
            [ -1 , 5 ]]);
```

$$A := \begin{bmatrix} 3 & 1 \\ -1 & 5 \end{bmatrix} \tag{5.1.1}$$

1. Eigenvalues

```
> charPol:=lambda^2-Trace(A)*lambda + Determinant(A);
```

$$charPol := \lambda^2 - 8\lambda + 16 \quad (5.2.1)$$

```
> eigValsA:=solve(charPol,lambda);
```

$$eigValsA := 4, 4 \quad (5.2.2)$$

```
> eigValsA:=convert([eigValsA],Vector):#convert the output to a
  vector form (better usability)
```

2. Eigenvectors

```
> for i from 1 to numelems(eigValsA) do:
   auxMat:=A-eigValsA[i]*IdentityMatrix(Dimension(A)):
   eigValsBasis[i]:=NullSpace(auxMat):#find null space -> solve
  homogeneous system
  od:
> for i from 1 to numelems(eigValsA) do:#check, if we really
  found the eigenvectors (does the equation lambda*h = A*h
  hold?)
   print([op(eigValsBasis[i])*eigValsA[i],A . op(eigValsBasis
  [i])]);
  od:
```

$$\left[\begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \end{bmatrix}\right]$$

$$\left[\begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \end{bmatrix}\right] \quad (5.3.1)$$

```
> eigValsBasis:=convert(
   [seq(op(eigValsBasis[i]),i=1..numelems(eigValsBasis))],
   Matrix
  );#convert the output to a matrix form (better usability)
```

$$eigValsBasis := \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (5.3.2)$$

Note: We found 2 same eigenvectors -> we need to find the generalized eigenvector

===

Remove the duplicate columns from *eigValsBasis*

```
> delList:=[]:#create list for positions of duplicate columns
  for i from 1 to ColumnDimension(eigValsBasis) do:
   for j from i+1 to ColumnDimension(eigValsBasis) do:
     if Equal(eigValsBasis[1..-1,i],eigValsBasis[1..-1,j],
  compare=entries) then:
      delList:=[op(delList),j]:
      fi:
   od:
  od:
  eigVecsA:=DeleteColumn(eigValsBasis,delList):#resave
  eigenvectors in a new matrix
```

===

2-a. Generalized Eigenvector

```
> print(eigVecsA);
```

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (5.3.3)$$

Note: I found only 1 eigenvector -> in order to find a general solution to the SODE I need to obtain a generalized eigenvector -> solve SLAR (A-lambdaE)k = h

```
> genEigVecA:=LinearSolve(A-eigValsA[1]*IdentityMatrix(2,2),
  eigVecsA,method=none,free=t);#system depend on 1 parameter
```

$$genEigVecA := \begin{bmatrix} -1 + t_{1,1} \\ t_{1,1} \end{bmatrix} \quad (5.3.4)$$

```
> genEigVecA:=subs(t[1,1]=0,genEigVecA);#I can choose any non-
  zero t
```

$$genEigVecA := \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (5.3.5)$$

```
> eigVecsA:=<<eigVecsA|genEigVecA>>;
```

$$eigVecsA := \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \quad (5.3.6)$$

3. General solution

```
> cVec:=Vector[column](numelems(eigValsA),symbol=C):#prepare
  vector of constants
> xGS:=add(cVec[i]*exp(eigValsA[i]*t)*(eigVecsA[1..-1,1]*t^
  (i-1) + (i-1)*eigVecsA[1..-1,2]),i=1..numelems(eigValsA));
```

$$xGS := \begin{bmatrix} C_1 e^{4t} + C_2 e^{4t} (t-1) \\ C_1 e^{4t} + C_2 e^{4t} t \end{bmatrix} \quad (5.4.1)$$

4. Particular solution - initial value problem (IVP)

```
> x0:=Vector[column]([1,0]):#specify the initial condition
> eqs:=[seq(x0[i]=subs(t=0,xGS[i]),i=1..numelems(x0))];
```

$$eqs := \left[1 = C_1 e^0 - C_2 e^0, 0 = C_1 e^0\right] \quad (5.5.1)$$

```
> B,b:=GenerateMatrix(eqs,convert(cVec,list));
```

$$B, b := \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (5.5.2)$$

```
> cVec:=LinearSolve(B,b);
```

$$cVec := \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad (5.5.3)$$

```
> xPS:=add(cVec[i]*exp(eigValsA[i]*t)*(eigVecsA[1..-1,1]*t^(i-1) + (i-1)*eigVecsA[1..-1,2]),i=1..numelems(eigValsA));
```

$$xPS := \begin{bmatrix} -e^{4t}(t-1) \\ -e^{4t}\,t \end{bmatrix} \tag{5.5.4}$$

5. Test - solve the system using Maple built-in methods

```
> xVec:=Vector[column]([x1(t),x2(t)]):
> systRHS:=MatrixVectorMultiply(A,xVec);
```

$$systRHS := \begin{bmatrix} 3\,x1(t) + x2(t) \\ -x1(t) + 5\,x2(t) \end{bmatrix} \tag{5.6.1}$$

```
> for i from 1 to numelems(xVec) do:
   ode[i]:=diff(xVec[i],t)=systRHS[i]:
  od:
> ode:=convert(ode,list):
  xVec:=convert(xVec,list):
> print(ode);
```

$$\left[\frac{d}{dt}\,x1(t) = 3\,x1(t) + x2(t),\ \frac{d}{dt}\,x2(t) = -x1(t) + 5\,x2(t)\right] \tag{5.6.2}$$

```
> sol:=dsolve(ode,xVec):
> xGS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

$$xGS\_MAPLE := \begin{bmatrix} e^{4t}(\_C2\,t + \_C1 - \_C2) \\ e^{4t}(\_C2\,t + \_C1) \end{bmatrix} \tag{5.6.3}$$

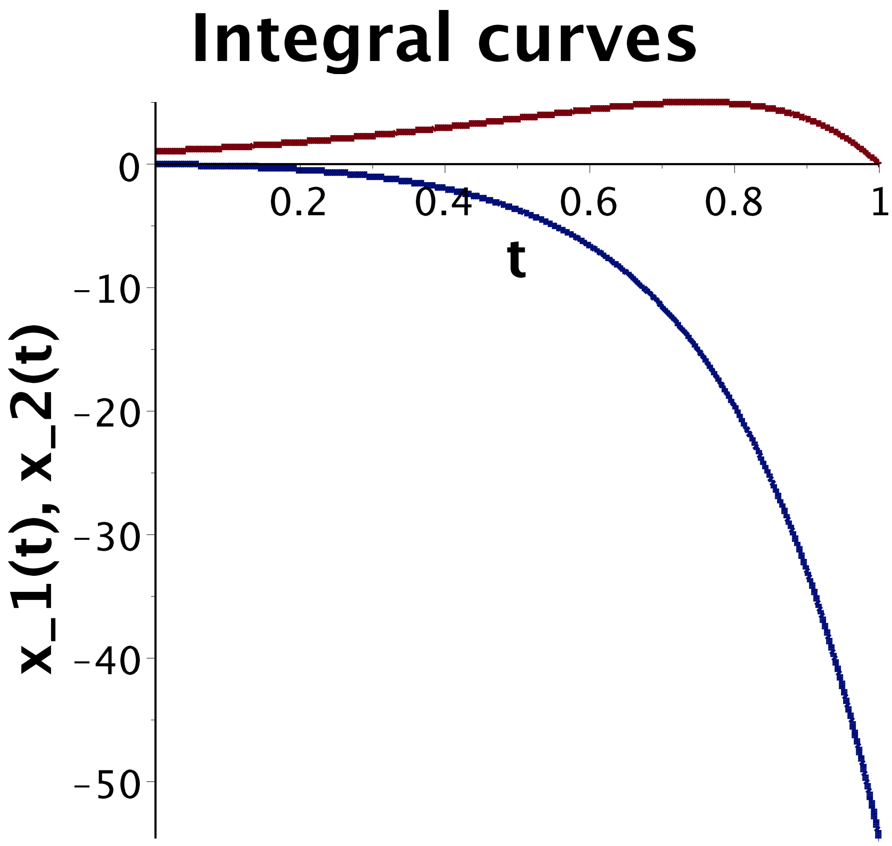
```
> IC:=x1(0)=1,x2(0)=0:
> sol:=dsolve([op(ode),IC],xVec):
> xPS_MAPLE:=Vector[column]([rhs(sol[1]),rhs(sol[2])]);
```

$$xPS\_MAPLE := \begin{bmatrix} e^{4t}(-t+1) \\ -e^{4t}\,t \end{bmatrix} \tag{5.6.4}$$

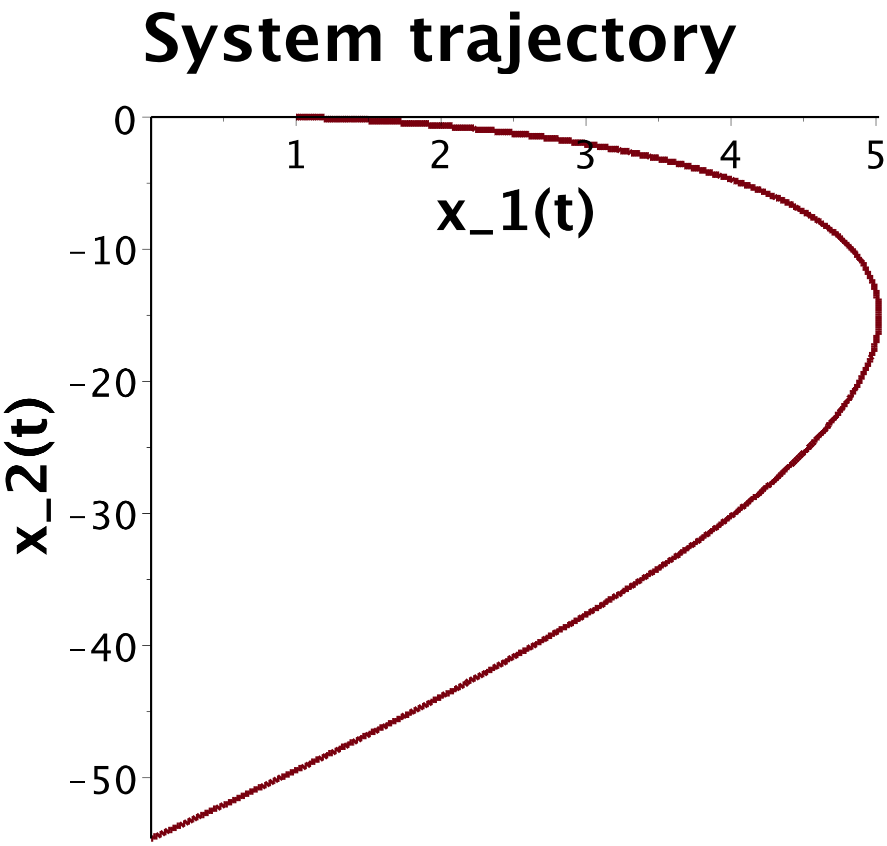
6. Plot resulting integral curve and trajectory

Note: All the possible trajectories -> *phase portrait*

```
> plot(xPS,t=0..1,
   thickness=3,
   legend=["x1(t)","x2(t)"],
   legendstyle=[font=["HELVETICA", 15],location=bottom],
   labels=["t","x_1(t), x_2(t)"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="Integral curves",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
   numpoints=5000,size=[600,400]
  );
```




```
> plot([op(convert(xPS,list)),t=0..1],
   thickness=3,
   labels=["x_1(t)","x_2(t)"],
   labeldirections=["horizontal","vertical"],
   labelfont=["Helvetica",20,Bold],
   title="System trajectory",
   titlefont=["Helvetica",24,Bold],
   axesfont=["Helvetica",14],
   numpoints=5000,size=[600,400]
  ) ;
```



System trajectory
0
1
2
3
4
5
x_1(t)
-10
-20
-30
-40
-50
x_2(t)