Lineare Differentialgleichungen höherer Ordnung

Die allgemeine Form einer Differentialgleichung n-ter Ordnung ist:

$$y^{(n)} + a_1(x)\, y^{(n-1)} + \ldots + a_n(x)\, y = f(x)$$

wobei a_1, ..., a_n, f stetige Funktionen sind.

Testen von Lösungen

Fur eine gegebene Differentialgleichung kann man überprüfen, ob eine Funktion eine Losung der DGL ist.

Wir betrachten der folgende DGL:

$$xy'' - (x+3)y' + 2y = 0$$

und die Funktion:

$$u(x) = x^2 + 4x + 6$$

Wir presentieren zwei Methoden um zu checken ob eine Funktion die DGL genügt.

In [1]:
```
reset()
x=var('x')
y=function('y')(x)
deq=x*diff(y,x,2)-(x+3)*diff(y,x)+2*y==0
```

In [2]:
```
u=function('u')(x)
u(x)=x^2+4*x+6
```

Bei der ersten Methode checkt man direkt, imdem man die Funktion $u(x)$ in der linken Seite der Gleichung einsetzt. Falls man 0 bekommen, ist die Funktion eine Lösung der DGL.

In [3]:
```
ans=x*diff(u,x,2)-(x+3)*diff(u,x)+2*u
ans(x)
```

Out[3]:
```
-2*(x + 3)*(x + 2) + 2*x^2 + 10*x + 12
```

In [4]:
```
expand(ans(x))
```

Out[4]:
```
0
```

Da wir $ans(x) = 0$ bekommen, ist $u(x)$ ist eine Losung der Gleichung.

Bei der zweite Methode definiert man die linke Seite der Differentialgleichung als ein Operator, das

von einem Funktion abhängt.

In [5]:
```
def L(y):
  a(x)=expand(x*diff(y,x,2)-(x+3)*diff(y,x)+2*y)
  return a(x)
```

Danach berechnen wir $L(u)$

In [6]:
```
L(u)
```

Out[6]:
```
0
```

Weil $L(u) = 0$, erschliessen wir dass u eine Lösung der DGL ist.

Lassen Sie uns prüfen, ob die Funktion $u(x) = e^{x^2}$ eine Lösung der DGL ist.

Ob wir direkt in der linken Seite der Gleichung einsetyen, bekommen wir

In [7]:
```
u(x)=exp(x^2)
ans=x*diff(u,x,2)-(x+3)*diff(u,x)+2*u
ans(x)
```

Out[7]:
```
-2*(x + 3)*x*e^(x^2) + 2*(2*x^2*e^(x^2) + e^(x^2))*x + 2*e^(x^2)
```

In [8]:
```
expand(ans(x))
```

Out[8]:
```
4*x^3*e^(x^2) - 2*x^2*e^(x^2) - 4*x*e^(x^2) + 2*e^(x^2)
```

Also, ist das Ergebnis verschieden von 0, deswegen ist $u(x) = e^{x^2}$ keine Losung.

Oder können wir das Operator L von u anwenden.

In [9]:
```
L(u)
```

Out[9]:
```
4*x^3*e^(x^2) - 2*x^2*e^(x^2) - 4*x*e^(x^2) + 2*e^(x^2)
```

und wir bekommen das gleiche Ergebnis, d.h., nicht 0, also ist $u(x) = e^{x^2}$ keine Lösung.

Einige bestimmte Lösungen finden

Lasst uns die folgende DGL betrachten:

$$xy'' - (x + 3)y' + 2y = 0$$

Wir suchen eine Lösung, die in der Form eines Polynoms zweiten Grades ist.

In [16]:
```
reset()
x=var('x')
y=function('y')(x)
deq=x*diff(y,x,2)-(x+3)*diff(y,x)+2*y==0
```

In [17]:
```
a,b,c=var('a,b,c')
u=function('u')(x)
u(x)=a*x^2+b*x+c
```

Wir konstruieren das Operator, das von der linken Seite erzeugt ist:

In [18]:
```
def L(y):
  a(x)=expand(x*diff(y,x,2)-(x+3)*diff(y,x)+2*y)
  return a(x)
```

In [19]:
```
L(u)
```

Out[19]:
```
-4*a*x + b*x - 3*b + 2*c
```

Wir konnen die Koeffizienten der x-Potenzen mit dem "coefficients" Befehl gruppieren.

In [20]:
```
ans=L(u)
```

In [23]:
```
ans.coefficients(x)
```

Out[23]:
```
[[-3*b + 2*c, 0], [-4*a + b, 1]]
```

Die Antwort ist eine Liste der Liste der Koeffizienten die die Potenzen von x entsprechen.

In [24]:
```
coeff=ans.coefficients(x)
```

Das Koeffizient für x^0 ist das erste Element der ersten Liste.

In [25]:
```
coeff[0][0]
```

Out[25]:
```
-3*b + 2*c
```

Das Koeffizient fur x ist das erste Element der zweiten Liste.

In [26]:
```
coeff[1][0]
```

Out[26]:
```
-4*a + b
```

Falls die Funktion $u(x) = ax^2 + bx + c$ eine Lösung der DGL ist, müssen alle diese Koeffizienten 0 sein. Deswegen finden wir die Parameter a, b, c von dieser Bedingung.

In [27]:
```
c0=coeff[0][0]
c1=coeff[1][0]
```

und wir lösen das entsprechende System in Bezug auf a, b, c.

In [28]:
```
solve([c0==0,c1==0],a,b,c)
```

Out[28]:
```
[[a == r1, b == 4*r1, c == 6*r1]]
```

Merken Sie dass die Antwort unendlich viele Lösungen besitzt. Wenn wir $r1 = 1$ einsetzen, bekommen wir die Funktion $u\,(x) = x^2 + 4x + 6$, die Funktion dass wir in der vorherigen Sektion eine Lösung überprüft wurde.

Falls wir für die gegebene DGL eine Lösung der Form $u(x) = e^{ax}(bx + c)$ finden möchten, berechen wir die Parameter wie folgendes.

In [31]:
```
u(x)=exp(a*x)*(b*x+c)
```

In [32]:
```
L(u)
```

Out[32]:
```
a^2*b*x^2*e^(a*x) + a^2*c*x*e^(a*x) - a*b*x^2*e^(a*x) - a*b*x*e^(a*x) - a
*c*x*e^(a*x) - 3*a*c*e^(a*x) + b*x*e^(a*x) - 3*b*e^(a*x) + 2*c*e^(a*x)
```

In [33]:
```
factor(L(u))
```

Out[33]:
```
(a^2*b*x^2 + a^2*c*x - a*b*x^2 - a*b*x - a*c*x - 3*a*c + b*x - 3*b + 2*c)
*e^(a*x)
```

Der Faktor e^{ax} kann nicht 0 sein, deshalb muss der erste Faktor 0 sein. Am ersten, brauchen wir diese Ausdruck mit e^{ax} zu kurzen.

In [34]:
```
ans=L(u)/exp(a*x)
ans
```

Out[34]:
```
(a^2*b*x^2*e^(a*x) + a^2*c*x*e^(a*x) - a*b*x^2*e^(a*x) - a*b*x*e^(a*x) -
a*c*x*e^(a*x) - 3*a*c*e^(a*x) + b*x*e^(a*x) - 3*b*e^(a*x) + 2*c*e^(a*x))*
e^(-a*x)
```

Merken Sie dass Sage nicht die Vereinfachung der Exponentialfunktion macht. Ob wir kürzen möchten, müssen wir "log_simplify" benutzen.

In [35]:
```
ans.log_simplify()
```

Out[35]:
```
(a^2 - a)*b*x^2 - (3*a - 2)*c - ((a - 1)*b - (a^2 - a)*c)*x - 3*b
```

In [36]:
```
ans=ans.log_simplify()
```

In [37]:
```
coeff=ans.coefficients(x)
coeff
```

Out[37]:
```
[[-(3*a - 2)*c - 3*b, 0], [-(a - 1)*b + (a^2 - a)*c, 1], [(a^2 - a)*b,
2]]
```

In [28]:
```
c0=coeff[0][0]
c1=coeff[1][0]
c2=coeff[2][0]
```

In [29]:
```
solve([c0==0,c1==0,c2==0],a,b,c)
```

Out[29]:
```
[[a == r2, b == 0, c == 0], [a == 1, b == r3, c == -3*r3]]
```

Wir erhalten zwei Moglichkeiten $u(x) = 0$ oder $u(x) = e^x(bx - 3b)$. Wir wissen dass die null Funktion immer eine Lösung einer linearen homogenen Gleichung ist, aber die kann nicht benutzt werden um linear unabhängige Lösungen zu konstruieren, deshalb nehmen wir die zweite Lösung für $b = 1$, $u(x) = e^x(x - 3)$.

In [40]:
```
u(x)=exp(x)*(x-3)
```

In [41]:
```
L(u)
```

Out[41]:
```
0
```

Linear unabhängige Lösungen

Betrache $S = \{y_1(x), \ldots, y_n(x)\}$ eine Menge von n Funktionen und jede Funktion ist mindestens $n - 1$ mal differenzialbar. Die Wronskideterminante der Funktionen S ist $W(x; y_1, \ldots, y_n)$, d.h., das folgende Determinant:

$$W(x; y_1, \ldots, y_n) = \begin{vmatrix} y_1 & y_2 & \cdots & y_{n1} \\ y_1' & y_2' & \cdots & y_n' \\ \vdots & \vdots & & \vdots \\ y_1^{(n-1)} & y_2^{(n-1)} & \cdots & y_n^{(n-1)} \end{vmatrix}$$

Das Wronskische Kriterium wird verwendet, um zu überprüfen, ob ein gegebenes System $\{y_1(x), \ldots, y_n(x)\}$ linear abhängig oder unabhängig ist.

Wenn die Wronskideterminate nicht 0 ist, dann sind die Lösungen linear unabhängig, also ist die allgemeine Lösung der linearen homogenen Gleichung eine lineare Kombination dieser Lösungen.

Lassen Sie uns die folgende lineare homogene DGL betrachten:

$$x^2\, y'' - 2x\, y' + 2y = 0;$$

Wir prüfen ob die Lösungen $\{x,\ x^2\}$ linear unabhängig sind.

Zuerst überprüfen wir, ob diese Funktionen Lösungen des DGL sind und zweitens überprüfen wir, ob die Wronskideterminaten verschieden von 0 ist.

In [51]:
```
reset()
x=var('x')
y=function('y')(x)
deq=x^2*diff(y,x,2)-2*x*diff(y,x)+2*y==0
```

In [52]:
```
def L(y):
  a(x)=expand(x^2*diff(y,x,2)-2*x*diff(y,x)+2*y)
  return a(x)
```

In [53]:
```
y1=function('y1')(x)
y2=function('y2')(x)
y1(x)=x
y2(x)=x^2
```

In [54]:
```
L(y1)
```

Out[54]:
```
0
```

In [55]:
```
L(y2)
```

Out[55]:
```
0
```

Die Funktionen $y_1(x) = x$ und $y_2(x) = x2$ sind also Lösungen der DGL.

Als nächstes konstruieren wir die Matrix mit der wir die Wronskideterminante berechen.

In [56]:
```
W = matrix([[y1(x),y2(x)], [diff(y1,x)(x),diff(y2,x)(x)]])
W
```

Out[56]:
```
[  x x^2]
[  1 2*x]
```

In [57]:
```
det(W)
```

Out[57]:
```
x^2
```

Wir erhalten, dass die Wronskideterminante x^2 ist, also ist sie nicht 0, da wir die Gleichung auf einem Intervall lösen, das es nicht den Punkt $x = 0$ enthält, folglich sind die Lösungen linear unabhängig.

Aufbau einer homogenen linearen ODE wenn die Lösungen gegeben sind

Es ist möglich, die homogene lineare ODE zu konstruieren, wenn das Lösungsfundamentalsystem gegeben ist.

Angenommen, dass das Funktionssystem $\{y_1(x), \ldots, y_n(x)\}$ ein Lösungsfundamentalsystem für einige homogene lineare DGL ist. Wenn man eine andere Lösung y(x) der Gleichung auswählt, dann ist das System $\{y(x), y_1(x), \ldots, y_n(x)\}$ linear abhängig, weil $y(x)$ eine lineare Kombination der

Lösungen aus dem Lösungsfundamentalsystem ist, so dass die Wronskideterminante

$$W\left(x; y(x), y_1(x), \ldots, y_n(x)\right) = 0.$$

Mit dieser Eigenschaft können wir die DGL erhalten, für die das System $\{y_1(x), \ldots, y_n(x)\}$ ein Lösungsfundamentalsystem ist.

Zum Beispiel ist das Funktionssystem $\left\{x,\ x^2\right\}$ linear unabhängig. Wir werden die entsprechende lineare homogene DGL konstruieren, für die dieses System ein Lösungsfundamentalsystem ist:

In [58]:
```
y1=function('y1')(x)
y2=function('y2')(x)
y1(x)=x
y2(x)=x^2
```

In [59]:
```
W = matrix([[y(x),y1(x),y2(x)], [diff(y,x)(x), diff(y1,x)(x),diff(y2,x)(x)]
W
```

Out[59]:
```
[            y(x)                 x              x^2]
[   diff(y(x), x)                 1              2*x]
[diff(y(x), x, x)                 0                2]
```

In [60]:
```
det(W)==0
```

Out[60]:
```
x^2*diff(y(x), x, x) - 2*x*diff(y(x), x) + 2*y(x) == 0
```

So erhalten wir die lineare homogene DGL:

$$x^2\ y'' - 2x\ y' + 2y = 0;$$

Lösen einer linearen homogenen DGL zweiter Ordnung mit nicht konstanten Koeffizienten

Wir können eine lineare homogene DG zweiter Ordnung lösen

$$y'' + a_1(x)\ y' + a_2(x)\ y = 0$$

wenn wir mindestens eine Lösung $y_1(x)$ kennt. Mit der Substitution

$$y(x) = z(x)\ y_1(x)$$

erhalten wir eine lineare homogene DGL zweiter Ordnung der Form

$$z'' + b_1(x)\ z' = 0$$

welche mit der Substitution $zprime(x) = u(x)$ zu einer linearen homogenen DGL erster Ordnung zurückgeführt wird.

$$u' + b_1(x)\ u = 0$$

Nachedem wir diese Gleichung lösen, erhalten wir $u(x)$ und lösen die Gleichung

$z\prime(x) = u(x)$

erhalten wir z(x). Mit dem Ergebnis erhalten wir die Lösung y(x).

Lasst uns die allgemeine Lösung der folgenden DGL finden:

$xy'' - (x+3)y' + 2y = 0$

wenn die Funktion $y_1\ (x) = x^2 + 4x + 6$ eine Lösung der DGL ist.

In [62]:
```
reset()
x=var('x')
y=function('y')(x)
deq=x*diff(y,x,2)-(x+3)*diff(y,x)+2*y==0
```

In [63]:
```
def L(y):
  a(x)=expand(x*diff(y,x,2)-(x+3)*diff(y,x)+2*y)
  return a(x)
```

In [64]:
```
y1=function('y1')(x)
y1(x)=x^2+4*x+6
```

In [65]:
```
L(y1)
```

Out[65]:
```
0
```

In [66]:
```
z=function('z')(x)
u=function('u')(x)
```

Wir machen die Substitution $y(x) = z(x) \cdot y_1(x)$ in der linken Seite der DGL durch die Auswertung von $L(z \cdot y_1)$

In [47]:
```
L(z*y1)
```

Out[47]:
```
-x^3*diff(z(x), x) + x^3*diff(z(x), x, x) - 3*x^2*diff(z(x), x) + 4*x^2*d
iff(z(x), x, x) - 10*x*diff(z(x), x) + 6*x*diff(z(x), x, x) - 18*diff(z
(x), x)
```

Um die Koeffizienten von z'' und z' zu sehen, verwenden wir den "collect" Befehl in Bezug auf z'' und z'

In [67]:
```
L(z*y1).collect(diff(z,x,2)).collect(diff(z,x))
```

Out[67]:
```
-(x^3 + 3*x^2 + 10*x + 18)*diff(z(x), x) + (x^3 + 4*x^2 + 6*x)*diff(z(x),
x, x)
```

Jetzt bauen wir die ODE in z

In [68]:
```
deq1=L(z*y1).collect(diff(z,x,2)).collect(diff(z,x))==0
deq1
```

Out[68]:
```
-(x^3 + 3*x^2 + 10*x + 18)*diff(z(x), x) + (x^3 + 4*x^2 + 6*x)*diff(z(x),
x, x) == 0
```

und wir ersetzen z'' mit u' und z' mit u

In [69]:
```
deq2=deq1.substitute(diff(z(x), x, x)==diff(u,x)).substitute(diff(z(x), x)=
deq2
```

Out[69]:
```
-(x^3 + 3*x^2 + 10*x + 18)*u(x) + (x^3 + 4*x^2 + 6*x)*diff(u(x), x) == 0
```

Wenn wir diese Gleichung lösen, bekommen wir die Lösung u

In [70]:
```
desolve(deq2,u)
```

Out[70]:
```
_C*x^3*e^x/(x^2 + 4*x + 6)^2
```

Als nächstes konstruieren wir die Lösung als Funktion in Bezug auf die Variable x und die Integrationskonstante $\_C$, wir verwenden die Variable uu weil u bereits in der Definition von $deq2$ verwendet wird.

In [71]:
```
uu(x,_C)=desolve(deq2,u)
uu
```

Out[71]:
```
(x, _C) |--> _C*x^3*e^x/(x^2 + 4*x + 6)^2
```

Als nächstes lösen wir die Gleichung $z'(x) = u(x)$ und das Lösen erhalten wir die Funktion z.

In [72]:
```
C1,C2=var('C1,C2')
```

In [73]:
```
deq=diff(z,x)==uu(x,C1)
```

In [74]:
```
zz(x,C1,_C)=desolve(deq,[z,x])
zz
```

Out[74]:
```
(x, C1, _C) |--> C1*(x - 3)*e^x/(x^2 + 4*x + 6) + _C
```

Mit dem Ausdruck von z (die Lösung wird in der Variablen zz) können wir die Lösung y aus der Relation konstruieren $y(x) = z(x) \cdot y_1(x)$

In [75]:
```
zz(x,C1,C2)*y1(x)
```

Out[75]:
```
(x^2 + 4*x + 6)*(C1*(x - 3)*e^x/(x^2 + 4*x + 6) + C2)
```

Beachten Sie, dass in diesem Ausdruck die Vereinfachungen nicht getan werden, um zu tun, dass wir "simplify_full" atribute verwenden müssen und um die Koeffizienten der Konstanten $C1$ und $C2$ zu sehen, müssen wir das "collect" atribute verwenden.

In [76]:
```
(zz(x,C1,C2)*y1(x)).simplify_full().collect(C1).collect(C2)
```

Out[76]:
```
(x*e^x - 3*e^x)*C1 + (x^2 + 4*x + 6)*C2
```

In [77]:
```
yy(x,C1,C2)=(zz(x,C1,C2)*y1(x)).simplify_full().collect(C1).collect(C2)
yy
```

Out[77]:
```
(x, C1, C2) |--> (x*e^x - 3*e^x)*C1 + (x^2 + 4*x + 6)*C2
```

In [ ]:

In [ ]: