- **Jakobian**

Sprawdzmy, jak zachowuje sie standardowe wyrazenie przy deklaracji zagniezdzenia zmiennych

```
abc = D[g[α[x, y], β[x, y]], x]
```

```
g^(1,0)[α[x, y], β[x, y]] α^(1,0)[x, y] + g^(0,1)[α[x, y], β[x, y]] β^(1,0)[x, y]
```

Oczywiscie normalne podstawienie nie daje obliczenia pochodnej α

```
Evaluate[abc /. {α[x, y] :> x}]
```

```
g^(1,0)[x, β[x, y]] α^(1,0)[x, y] + g^(0,1)[x, β[x, y]] β^(1,0)[x, y]
```

i trzeba uzyc Pure function

```
α^(1,0)[x, y] /. {α ↦ (#1^2 &)}
```

```
2 x
```

Zdefiniujmy jakobian (macierz)

```
Jak[α_, β_] := Outer[D, {α[x, y], β[x, y]}, {x, y}]
```

```
Jak[α, β]
```

```
{{α^(1,0)[x, y], α^(0,1)[x, y]}, {β^(1,0)[x, y], β^(0,1)[x, y]}}
```

w podobny, jak wyzej sposob mamy konkretna wartosc

```
Jak[α, β] /. {α :> (#1^2 - #2^2 &), β ↦ (#1 * #2 &)}
```

```
{{2 x, -2 y}, {y, x}}
```

- **A teraz przejdzmy do transformacji potrzebnej w metodzie charakterystyk.**

zadanie bedzie polegalo na automatyzacji do pewnego stopnia rachunkow poszukiwania rownan charakterystyk

wybierzmy jakies rownanie i wymiar

```
NN = 3;
wyr = y * D[u[x, y, z], x] - x * D[u[x, y, z], y] == 0
```

```
-x u^(0,1,0)[x, y, z] + y u^(1,0,0)[x, y, z] == 0
```

wykonajmy pewne przeksztalcenia typu D[u[x, y], x_i] ↦ p_i

```
(* zamiana symboliki: *) in = {x, y, z};
out = Table[Subscript[x, i], {i, NN}]
```

```
{x_1, x_2, x_3}
```

```
rule1 = Table[in[[i]] → out[[i]], {i, NN}]
```

```
{x → x_1, y → x_2, z → x_3}
```

```
(* np.: *) x * y /. rule1
```

```
x_1 x_2
```

```
wyr = wyr /. rule1
```

$$-x_1\, u^{(0,1,0)}[x_1, x_2, x_3] + x_2\, u^{(1,0,0)}[x_1, x_2, x_3] == 0$$

```
uA = Apply[u, out]
```

$$u[x_1, x_2, x_3]$$

```
(* dziala podstawienie: *) 2 * u^(0,1)[x1, x2] /. D[u[x1, x2], x2] :> 3
```

$$6$$

```
rule2 = Table[D[uA, Subscript[x, i]] :> Evaluate[Subscript[p, i]], {i, 3}]
```

$$\left\{u^{(1,0,0)}[x_1, x_2, x_3] :\to p_1,\ u^{(0,1,0)}[x_1, x_2, x_3] :\to p_2,\ u^{(0,0,1)}[x_1, x_2, x_3] :\to p_3\right\}$$

rownanie przyjmuje postac F(x1,x2,x3,z=u,p1,p2,p3)=0:

```
eq0 = wyr /. rule2
```

$$-p_2\, x_1 + p_1\, x_2 == 0$$

```
F = %[[1]]
```

$$-p_2\, x_1 + p_1\, x_2$$

```
(* rownania pirwsze charakterystyk, x': *) Dx = Table[D[F, Subscript[p, i]], {i, NN}]
```

$$\{x_2, -x_1, 0\}$$

```
eq1 = Table[Subscript[x, i]', {i, NN}] == Dx
```

$$\{(x_1)', (x_2)', (x_3)'\} == \{x_2, -x_1, 0\}$$

```
Table[Hold[D[Subscript[x, i], (#1 &)]], {i, NN}]
```

$$\{\text{Hold}[\partial_{\#1\&}\, x_i], \text{Hold}[\partial_{\#1\&}\, x_i], \text{Hold}[\partial_{\#1\&}\, x_i]\}$$

teraz rownanie drugie, z' :

```
(* il. skalarny: *) Inner[Times, {a, b}, {x, y}, Plus]
(* lub *) {a, b}.{x, y}
```

$$a\, x + b\, y$$

$$a\, x + b\, y$$

```
pVec = Table[Subscript[p, i], {i, NN}]
```

$$\{p_1, p_2, p_3\}$$

```
(* z': *) z' == Dx.pVec
```

```
eq2 = z' == -p2 x1 + p1 x2
```

$$z' == -p_2\, x_1 + p_1\, x_2$$

Zatem uklad rownan jest:

```
Table[eq1[[1]][[i]] == eq1[[2]][[i]], {i, NN}]
```

$$\{(x_1)' == x_2,\ (x_2)' == -x_1,\ (x_3)' == 0\}$$

dolaczamy rownanie 2:

```
uklad = Join[%, {eq2, eq0}]
```

$$\{(x_1)' == x_2,\ (x_2)' == -x_1,\ (x_3)' == 0,\ z' == -p_2\, x_1 + p_1\, x_2,\ z' == -p_2\, x_1 + p_1\, x_2,\ -p_2\, x_1 + p_1\, x_2 == 0\}$$

```
Solve[uklad, {z', Subscript[x, 1]', Subscript[x, 2]'}]
```

$$\{\{z' \to 0,\ (x_1)' \to x_2,\ (x_2)' \to -x_1\}\}$$

Do czesci rownan jest potrzebne jednakze DSolve. Tu (cwiczenie: zautomatyzowac) recznie rownania te wklejono, dopisano parametr 's'.

```
FullForm[(x1)']
FullForm[x1']
```

```
Derivative[1][Subscript[x, 1]]
```

```
Derivative[1][Subscript[x, 1]]
```

```
FullForm[(x2)'[s]]
```

```
Derivative[1][Subscript[x, 2]][s]
```

```
DSolve[{(x1)'[s] == x2[s], (x2)'[s] == -x1[s]}, {x1[s], x2[s]}, s]
```

$$\{\{x_1[s] \to C[1]\ \text{Cos}[s] + C[2]\ \text{Sin}[s],\ x_2[s] \to C[2]\ \text{Cos}[s] - C[1]\ \text{Sin}[s]\}\}$$