問 2 (3)

In [2]:
```
using SymPy

# x を記号変数として宣言する。
# ここでは x が独立変数である。
@syms x

# y を「未知関数」として宣言する。
# これは単なる記号 y ではなく、
# x に依存する関数 y(x) を扱うための準備である。
y = SymFunction("y")

# 微分方程式
#
#     y' = y / (x(x+1))
#
# を SymPy の等式 Eq(...) として定義する。
#
# diff(y(x), x) は y(x) を x で微分したもの、
# つまり数学で書く dy/dx または y' を表す。
eq = Eq(
    diff(y(x), x),
    y(x) / (x * (x + 1))
)

# 定義した微分方程式を表示する。
# Jupyter Notebook では数式としてきれいに表示される。
display(eq)

# 微分方程式 eq を、未知関数 y(x) について解く。
# dsolve は differential equation solve の意味だと思えばよい。
sol = dsolve(eq, y(x))

# 解を表示する。
# たとえば
#
#     Eq(y(x), C1*x/(x + 1))
#
# のような結果が表示される。
display(sol)

# sol は
#
#     y(x) = C1*x/(x+1)
#
# という「等式」である。
# rhs(sol) は right hand side、つまり右辺を取り出す関数である。
#
# したがって sol_rhs には
#
#     C1*x/(x+1)
#
# が入る。
sol_rhs = rhs(sol)
```

```
# 得られた解が本当に微分方程式を満たすかを確認する。
#
# 元の微分方程式は
#
#     y' = y / (x(x+1))
#
# なので、左辺 - 右辺 を計算して 0 になるかを見る。
#
# つまり
#
#     d/dx(sol_rhs) - sol_rhs / (x(x+1))
#
# を計算している。
check = simplify(
    diff(sol_rhs, x) - sol_rhs / (x * (x + 1))
)

# check が 0 ならば、得られた解は確かに微分方程式を満たしている。
display(check)
```

$\frac{d}{dx}y(x) = \frac{y(x)}{x(x+1)}$

$y(x) = \frac{C_1 x}{x+1}$

0

問 4 (1)

In [3]:
```
using SymPy

# x を記号変数として宣言する。
# ここでは x が独立変数である。
@syms x

# y を未知関数として宣言する。
# 数学でいう y(x) を扱うための準備である。
y = SymFunction("y")

# 微分方程式
#
#     y' - y = e^(2x)
#
# を SymPy の等式として定義する。
#
# diff(y(x), x) は y'(x) を表す。
# exp(2*x) は e^(2x) を表す。
eq = Eq(
    diff(y(x), x) - y(x),
    exp(2 * x)
)

# 定義した微分方程式を表示する。
display(eq)

# 微分方程式 eq を、未知関数 y(x) について解く。
sol = dsolve(eq, y(x))
```

```
# 解を表示する。
#
# 結果は例えば
#
#     Eq(y(x), C1*exp(x) + exp(2*x))
#
# または
#
#     Eq(y(x), (C1 + exp(x))*exp(x))
#
# のような形で表示される。
#
# これらは同じ意味である。
display(sol)

# sol は
#
#     y(x) = ...
#
# という等式である。
# rhs(sol) により、その右辺だけを取り出す。
sol_rhs = rhs(sol)

# 得られた解が本当に微分方程式を満たすかを確認する。
#
# 元の微分方程式は
#
#     y' - y = e^(2x)
#
# なので、
#
#     y' - y - e^(2x)
#
# を計算して 0 になるかを調べる。
check = simplify(
    diff(sol_rhs, x) - sol_rhs - exp(2 * x)
)

# check が 0 ならば、得られた解は確かに微分方程式を満たしている。
display(check)
```

$$-y(x) + \frac{d}{dx}y(x) = e^{2x}$$

$$y(x) = \left(C_1 + e^{x}\right) e^{x}$$

$$0$$

問 5 (1)

In [4]:
```
using SymPy

# x, y を記号変数として宣言する。
# 今回は y を未知関数 y(x) としてではなく、
# 2変数関数 F(x,y) の第2変数として扱う。
@syms x y C

# 微分方程式
#
```

```
#     (-x + y + 2) dx + (x - y + 1) dy = 0
#
# を
#
#     M(x,y) dx + N(x,y) dy = 0
#
# と見る。
M = -x + y + 2
N =  x - y + 1

# M_y = ∂M/∂y を計算する。
My = diff(M, y)

# N_x = ∂N/∂x を計算する。
Nx = diff(N, x)

# 完全微分形であるための条件は
#
#     ∂M/∂y = ∂N/∂x
#
# である。
#
# そこで、その差を計算して 0 になるかを確認する。
exact_check = simplify(My - Nx)

display(My)
display(Nx)
display(exact_check)

# exact_check が 0 ならば、
#
#     ∂M/∂y = ∂N/∂x
#
# なので、この微分方程式は完全微分形である。
```

1

1

0

In [5]:

```
using SymPy

@syms x y C

M = -x + y + 2
N =  x - y + 1

# まず F_x = M を使う。
#
# F_x = -x + y + 2
#
# なので、M を x について積分すれば F の候補が得られる。
#
# ただし、x で積分したとき、y だけの関数は「積分定数」として残る。
# そこでまずは y だけの未知関数を除いた部分 F0 を求める。
F0 = integrate(M, x)

display(F0)
```

```
# F0 は
#
#     -x^2/2 + x*y + 2*x
#
# になる。
#
# しかし本当は、ここに y だけの関数 g(y) を足して
#
#     F(x,y) = F0 + g(y)
#
# と考える必要がある。
#
# 次に F_y = N を使って g(y) を決める。
#
# F0 を y で微分すると、F_y のうち F0 から来る部分がわかる。
F0_y = diff(F0, y)

display(F0_y)

# 本来ほしいのは
#
#     F_y = N
#
# である。
#
# いま
#
#     F = F0 + g(y)
#
# と考えているので、
#
#     F_y = ∂F0/∂y + g'(y)
#
# である。
#
# したがって
#
#     g'(y) = N - ∂F0/∂y
#
# である。
gprime = simplify(N - F0_y)

display(gprime)

# g'(y) が求まったので、y について積分して g(y) を求める。
g = integrate(gprime, y)

display(g)

# これでポテンシャル関数 F(x,y) が求まる。
F = simplify(F0 + g)

display(F)

# 解は
#
#     F(x,y) = C
#
# で与えられる。
solution = Eq(F, C)
```

```
display(solution)

# 念のため、F_x = M, F_y = N になっているか確認する。
check_Fx = simplify(diff(F, x) - M)
check_Fy = simplify(diff(F, y) - N)

display(check_Fx)
display(check_Fy)
```

$$-\frac{x^2}{2} + x\,(y+2)$$

$$x$$

$$1 - y$$

$$-\frac{y^2}{2} + y$$

$$-\frac{x^2}{2} + x\,(y+2) - \frac{y^2}{2} + y$$

$$-\frac{x^2}{2} + x\,(y+2) - \frac{y^2}{2} + y = C$$

$$0$$

$$0$$

In [ ]: