1 Vector Matrix Calculations, pen and paper

1. v =

$$\begin{bmatrix} -2 \\ b \\ 7 \\ 2 \\ a \end{bmatrix}$$

vTv =

$$\begin{bmatrix} -2 \; b \; 7 \; 2 \; a \end{bmatrix} \begin{bmatrix} -2 \\ b \\ 7 \\ 2 \\ a \end{bmatrix}$$

vTv = [-2*-2 + b*b + 7*7 + 2*2 + a*a] =

$$[57 + b^2 + a^2]$$

vvT =

$$\begin{bmatrix} -2 \\ b \\ 7 \\ 2 \\ a \end{bmatrix} \begin{bmatrix} -2 \; b \; 7 \; 2 \; a \end{bmatrix}$$

vvT =

$$\begin{bmatrix} -2 * -2 & -2 * b & -2 * 7 & -2 * 2 & -2 * a \\ b * -2 & b * b & b * 7 & b * 2 & b * a \\ 7 * -2 & 7 * b & 7 * 7 & 7 * 2 & 7 * a \\ 2 * -2 & 2 * b & 2 * 7 & 2 * 2 & 2 * a \\ a * -2 & a * b & a * 7 & a * 2 & a * a \end{bmatrix}$$

vvT =

$$\begin{bmatrix} 4 & -2b & -14 & -4 & -2a \\ -2b & b^2 & 7b & 2b & ba \\ -14 & 7b & 49 & 14 & 7a \\ -4 & 2b & 14 & 4 & 2a \\ -2a & ab & 7a & 2a & a^2 \end{bmatrix}$$

2. Let A and B the following matrices

1.2. $v = B\cdot[x,y]^T = \begin{bmatrix} -4 & 8 \\ -5 & -9 \\ 4 & -7 \end{bmatrix}\cdot\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -4x+8y \\ -5x-9y \\ 4x-7y \end{bmatrix}$

$$w = A\cdot V = \begin{bmatrix} -5 & 4 & -6 \\ 3 & 4 & -6 \\ 8 & -6 & 8 \end{bmatrix}\cdot\begin{bmatrix} -4x+8y \\ -5x-9y \\ 4x-7y \end{bmatrix} = \begin{bmatrix} -5(-4x+8y)+4(-5x-9y)-6(4x-7y) \\ 3(-4x+8y)+4(-5x-9y)-6(4x-7y) \\ 8(-4x+8y)-6(-5x-9y)+8(4x-7y) \end{bmatrix}$$

$$\therefore\ w = A\cdot v = \begin{bmatrix} -24x-34y \\ -56x+30y \\ +30x+62y \end{bmatrix}$$

$$C = A\cdot B = \begin{bmatrix} -5 & 4 & -6 \\ 3 & 4 & -6 \\ 8 & -6 & 8 \end{bmatrix}\cdot\begin{bmatrix} -4 & 8 \\ -5 & -9 \\ 4 & -7 \end{bmatrix} = \begin{bmatrix} -24 & -34 \\ -56 & 30 \\ 30 & 62 \end{bmatrix}.$$

$$z = C\cdot[x,y]^T = \begin{bmatrix} -24 & -34 \\ -56 & 30 \\ 30 & 62 \end{bmatrix}\cdot\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -24x-34y \\ -56x+30y \\ 30x+62y \end{bmatrix}$$

3. Let A be the matrix

1.3. $A^2 = A\cdot A = \begin{bmatrix} 1 & -1 & -1 \\ 8-x & x-6 & x-6 \\ x-7 & 5-x & 5-x \end{bmatrix}^2 =$

$$= \begin{bmatrix} 1-8+x-x+7 & -1-x+6-5+x & -1-x+6-5+x \\ 8-x+(x-6)(8-x)+(x-6)(x-7) & -8+x+(x-6)^2+(x-6)(5-x) & -8+x+(x-6)^2+(x-6)(5-x) \\ x-7+(5-x)(8-x)+(5-x)(x-7) & -x+7+(5-x)(x-6)+(5-x)^2 & -x+7+(5-x)(x-6)+(5-x)^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 2 & -2 & -2 \\ -2 & 2 & 2 \end{bmatrix}$$

$$A^3 = A^2\cdot A = \begin{bmatrix} 0 & 0 & 0 \\ 2 & -2 & -2 \\ -2 & 2 & 2 \end{bmatrix}\cdot\begin{bmatrix} 1 & -1 & -1 \\ 8-x & x-6 & x-6 \\ x-7 & 5-x & 5-x \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 2-2(8-x)-2(x-7) & -2-2(x-6)-2(5-x) & -2-2(x-6)-2(5-x) \\ -2+2(8-x)+2(x-7) & 2+2(x-6)+2(5-x) & 2+2(x-6)+2(5-x) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4. Let A be the matrix

1.4. $A^{-1} = \begin{bmatrix} -15 & 12 & 4 \\ -24 & 19 & 6 \\ -76 & 57 & 13 \end{bmatrix}$

$\therefore A \cdot \vec{f_1} = \vec{e_1} \Rightarrow \vec{f_1} = A^{-1}\vec{e_1} = \begin{bmatrix} -15 & 12 & 4 \\ -24 & 19 & 6 \\ -76 & 57 & 13 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -15 \\ -24 \\ -76 \end{bmatrix}$

$\therefore \vec{f_2} = A^{-1} \cdot \vec{e_2} = \begin{bmatrix} 12 \\ 19 \\ 57 \end{bmatrix}$

$\therefore \vec{f_3} = A^{-1}\vec{e_3} = \begin{bmatrix} 4 \\ 6 \\ 13 \end{bmatrix}$

$B = \begin{bmatrix} -15 & 12 & 4 \\ -24 & 19 & 6 \\ -76 & 57 & 13 \end{bmatrix}$ $\therefore A \cdot B = \begin{bmatrix} -95 & 72 & -4 \\ -144 & 109 & -6 \\ 76 & -57 & 3 \end{bmatrix}\begin{bmatrix} -15 & 12 & 4 \\ -24 & 19 & 6 \\ -76 & 57 & 13 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$B \cdot A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Conclusion:

- In come cases AB = BA for matrix A and B.
- Matrix B is the inverse matrix of A.

5. [optional]

1.5. Let $A = \begin{bmatrix} \vec{a_1} \\ \vdots \\ \vec{a_n} \end{bmatrix}_{n\times n}$, $\vec{a_i}$ is a $1\times n$ row vector, $i \in [0, n]$.

Let $B = [\vec{b_1} \cdots \vec{b_n}]_{n\times n}$, $\vec{b_i}$ is a $n\times 1$ column vector, $i \in [0, n]$.

$\therefore A \cdot B = \begin{bmatrix} \vec{a_1}\cdot\vec{b_1} & \cdots & \vec{a_1}\cdot\vec{b_n} \\ \vdots & \ddots & \\ \vec{a_n}\cdot\vec{b_1} & & \vec{a_n}\cdot\vec{b_n} \end{bmatrix}$ where $\vec{a_i}\cdot\vec{b_j}$ is a scalar. $i,j \in [0,n]$.

$\therefore (A \cdot B)^T = \begin{bmatrix} \vec{a_1}\vec{b_1} & \cdots & \vec{a_n}\vec{b_1} \\ & \ddots & \\ \vec{a_1}\cdot\vec{b_n} & & \vec{a_n}\cdot\vec{b_n} \end{bmatrix}$

$A^T = [\vec{a_1}^T \ \vec{a_2}^T \cdots \vec{a_n}^T]_{n\times n}$, where $\vec{a_i}^T$ is a $n\times 1$ column vecter, $i \in [0,n]$

$B^T = \begin{bmatrix} \vec{b_1}^T \\ \vdots \\ \vec{b_n}^T \end{bmatrix}_{n\times n}$, where $\vec{b_i}^T$ is a $1\times n$ row vector, $i \in [0,n]$

$\therefore B^T \cdot A^T = \begin{bmatrix} \vec{b_1}^T\cdot\vec{a_1}^T & \cdots & \vec{b_1}^T\cdot\vec{a_n}^T \\ \vdots & \ddots & \\ \vec{b_n}^T\cdot\vec{a_1}^T & & \vec{b_n}^T\cdot\vec{a_n}^T \end{bmatrix}$ where $\vec{b_j}^T \cdot \vec{a_i}^T$ is a scalar. $i,j \in [0,n]$

and $\vec{b_j}^T \cdot \vec{a_i}^T = \vec{a_i}\cdot\vec{b_j}$ for i and j in $[0,n]$. $\therefore (A\cdot B)^T = B^T A^T$

2 Vector Matrix Calculations, Python

In [1]:
```
import numpy as np
```

1. An inner and an outer product. Compute the inner product

In [2]:
```
a = np.array([17,6,-5.5,3])
b = np.array([-4.9,1.17,0,2])
```

In [3]:
```
# inner product of a and b is:
print(a.dot(b))
print(np.dot(a, b))
print(a @ b)
```

```
-70.28000000000002
-70.28000000000002
-70.28000000000002
```

In [4]:
```
# outer product of a and b is :
a = a.reshape((a.shape[0], 1))
b = b.reshape((b.shape[0], 1))
# print(a.shape, b.shape)
print(a@b.T)
```

```
[[-83.3    19.89    0.     34.   ]
 [-29.4     7.02    0.     12.   ]
 [ 26.95   -6.435   0.    -11.   ]
 [-14.7     3.51    0.      6.   ]]
```

2. Let t = [−3, 2, 11]T . Compute the cross-product

In [5]:
```
# my implementation of cross product:
def CrossProduct(a, b):
    return np.array([a[1]*b[2] - a[2]*b[1], -(a[0]*b[2] - a[2]*b[0]
```

In [20]:
```
t = np.array([-3, 2, 11])
print('the cross-product t × t is:', np.cross(t, t))
print('the cross-product t × t is:', CrossProduct(t, t))
```

```
the cross-product t × t is: [0 0 0]
the cross-product t × t is: [0 0 0]
```

In [7]:
```
a = np.array([-2, 6, 1])
b = np.cross(t, a)
c = np.cross(a, t)
print('b: ', b)
print('c: ', c)
print('tTb: ', t.T@b)
print('aTb: ', a@b)
```

```
b:  [-64 -19 -14]
c:  [64 19 14]
tTb:  0
aTb:  0
```

3. Write a function that takes a 3D vector t and associates the corresponding matrix ˆt.

In [8]:
```
def GettHat(t):
    return np.array([[0, -t[2], t[1]], [t[2], 0, -t[0]], [-t[1], t[

t_hat = GettHat(t)
print('t: ', t)
print('t_hat: \n', t_hat)
print('t_hat @ t:', np.dot(t_hat, t))
print('t_hat @ a:', np.dot(t_hat, a))
# the product of t_hat and t is a zero vector. The result equals to
# the product of t_hat and a is a none zero vector. The result equa
```

```
t:  [-3  2 11]
t_hat:
 [[  0 -11   2]
 [ 11   0   3]
 [ -2  -3   0]]
t_hat @ t: [0 0 0]
t_hat @ a: [-64 -19 -14]
```

4. For a 3D vector v, the operation v → t × v is linear. So it can be represented by a matrix. Which one?

In [9]:
```
# Let v = t, we have t → t × t = [0, 0, 0]T, recall t_hat dot t equ
v = np.array([1, 2, 3])
print(np.dot(t_hat, v))
print(np.cross(t, v))
# Matrix t_hat can represent the linear mapping of v → t × v

# methord 2:
# reconstruct t x v as B·v, B is the matrix that represent the mapp
```

```
[-16  20  -8]
[-16  20  -8]
```

5. Check that ˆtT = −ˆt.

In [10]:
```
print('t_hatT:\n', t_hat.T)
print('-t_hat:\n', -t_hat)
```

```
t_hatT:
 [[  0  11  -2]
 [-11   0  -3]
 [  2   3   0]]
-t_hat:
 [[  0  11  -2]
 [-11   0  -3]
 [  2   3   0]]
```

In [11]:
```
t_hat_square = t_hat.dot(t_hat)
print('t̂^2 = t̂. t̂:\n', t_hat_square)
```

```
t̂^2 = t̂. t̂:
 [[-125   -6  -33]
 [  -6 -130   22]
 [ -33   22  -13]]
```

In [12]:
```
print('transpose of t̂^2:\n', t_hat_square.T)
# the transpose of tˆ^2 is tˆ^2, because tˆ^2 is a symmetric matrix
```

```
transpose of t̂^2:
 [[-125   -6  -33]
 [  -6 -130   22]
 [ -33   22  -13]]
```

In [13]:
```
print('t̂^3 = t̂. t̂. t̂:\n', t_hat_square.dot(t_hat))
# tˆ^3 = tˆ. tˆ. tˆ is a symmetric matrix, with zeros in diagonal
```

```
t̂^3 = t̂. t̂. t̂:
 [[    0  1474  -268]
 [-1474     0  -402]
 [  268   402     0]]
```

3 Derivatives

1. Compute the derivative f ′ (x) of f (x)

3.1. $f(x) = e^{-\frac{x^2}{2}} \quad \therefore f'(x) = e^{-\frac{x^2}{2}} \cdot \left(-\frac{x^2}{2}\right)' = -e^{-\frac{x^2}{2}} \cdot x = -x \cdot f(x)$

2. Compute its second derivative, f ′′ (x)

3.2. $\cancel{f''(x) = -((}$

$$f''(x) = -(1 \cdot f(x) + x \cdot f'(x))$$
$$= -f(x) - x f'(x)$$
$$= -f(x) + x^2 f(x)$$
$$\therefore f''(x) = f(x) \cdot (x^2 - 1)$$

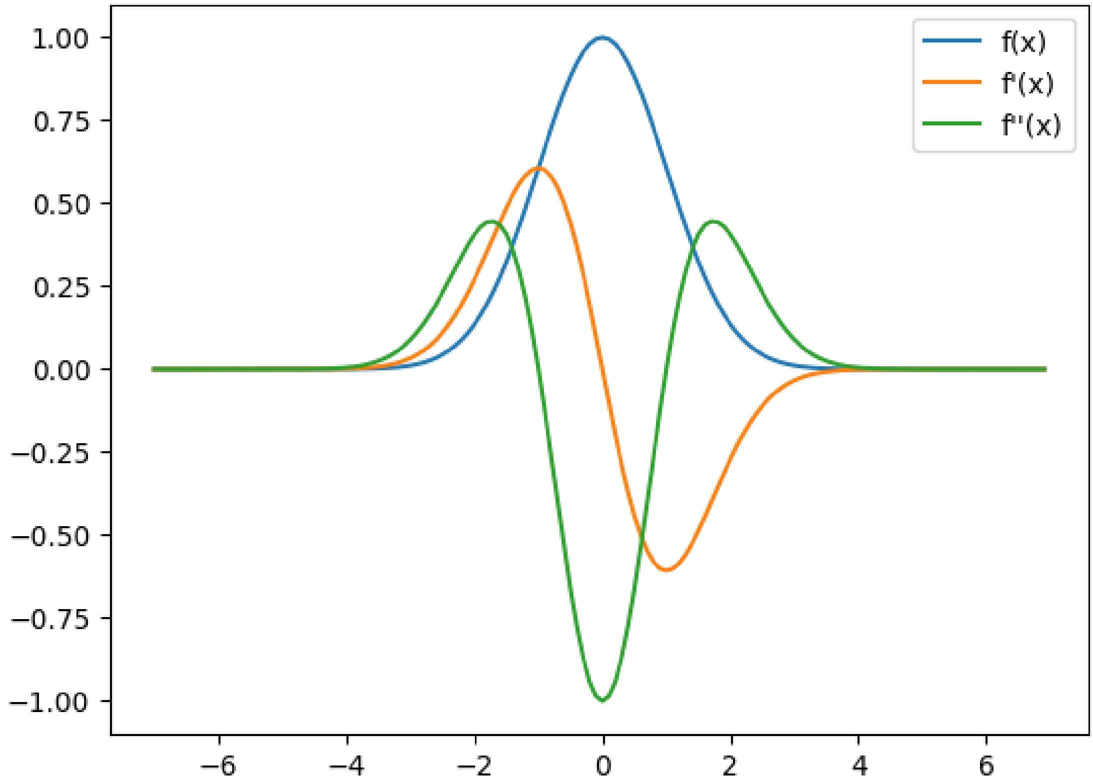
3. Plot on the same graph f (x), f ′ (x) and f ″ (x)

In [14]:
```
import matplotlib.pyplot as plt
def f(x):
    return pow(np.e, -1 * x * x / 2)
def f_prime(x):
    return -1 * x * f(x)
def f_double_pirme(x):
    return (x * x - 1) * f(x)


# auto calculation of derivates from ChatGPT
# import sympy as sp
# x = sp.symbols('x')
# fx = pow(np.e, -1 * x * x / 2)
# df_dx = sp.diff(fx, x)
# df_dx_fn = sp.lambdify(x, df_dx, 'numpy')
# y = df_dx_fn(X)
```

In [15]:
```
X = np.arange(-7, 7., 0.1)
Y = [f(x) for x in X]
Y_1 = [f_prime(x) for x in X]
Y_2 = [f_double_pirme(x) for x in X]
plt.plot(X, Y, label='f(x)')
plt.plot(X, Y_1, label='f\'(x)')
plt.plot(X, Y_2, label='f\'\'(x)')
plt.legend()
```

Out[15]: <matplotlib.legend.Legend at 0x7f85f83206d0>



4. Compute the derivative g ′ (t) of g(t) = cos(t)/ sin(t).

3.4 $g(t) = \frac{\cos(t)}{\sin(t)}$

$$\therefore g'(t) = \frac{\cos'(t)\cdot\sin(t) - \cos(t)\cdot\sin'(t)}{\sin^2(t)} = \frac{-\sin^2(t) - \cos^2(t)}{\sin^2(t)} = -\frac{1}{\sin^2(t)}$$

5. Plot f (x, t)

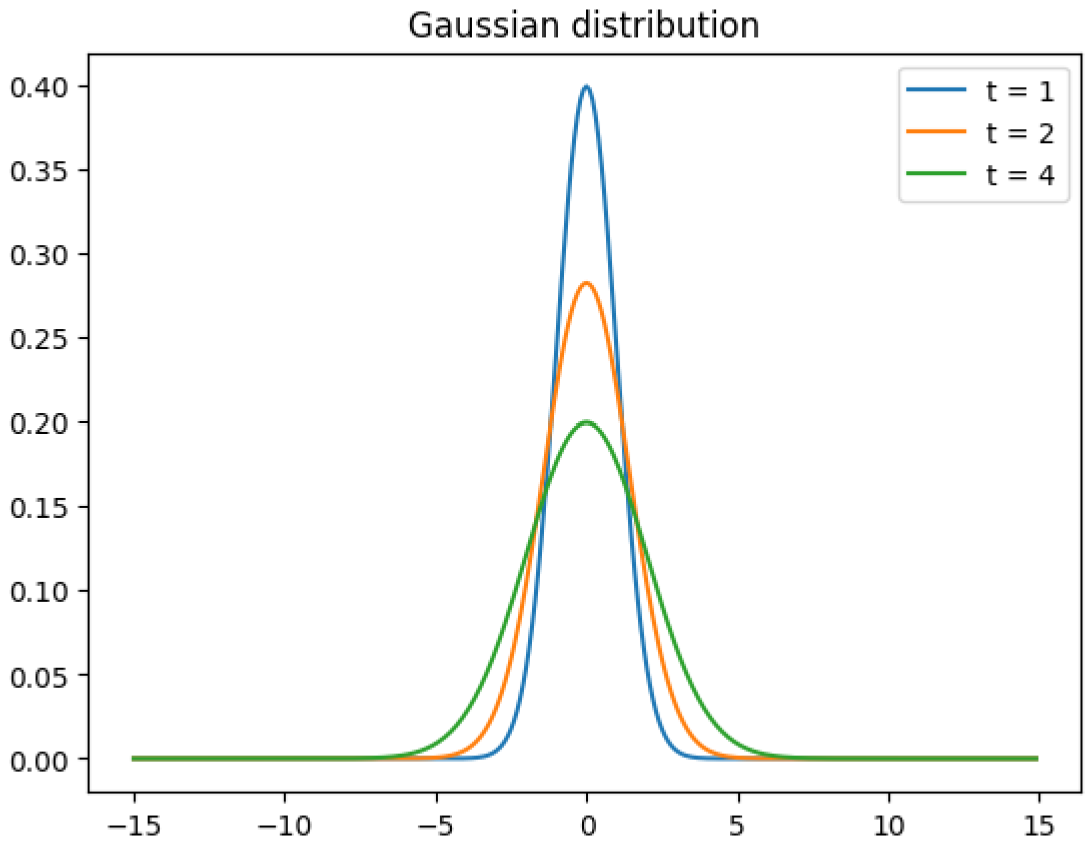
In [16]:
```
X = np.arange(-15, 15, 0.1)

def gd_function(x, t):
    return 1 / np.sqrt((2 * np.pi * t)) * pow(np.e, -1 * x * x / 2

Y = [gd_function(x, 1) for x in X]
Y_2 = [gd_function(x, 2) for x in X]
Y_4 = [gd_function(x, 4) for x in X]

plt.plot(X, Y, label='t = 1')
plt.plot(X, Y_2, label='t = 2')
plt.plot(X, Y_4, label='t = 4')
plt.title('Gaussian distribution')
plt.legend()
```

Out[16]:
```
<matplotlib.legend.Legend at 0x7f85cf61b190>
```



6. Plot ∂f/∂x (x, t)

3.6 $f(x,t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}$

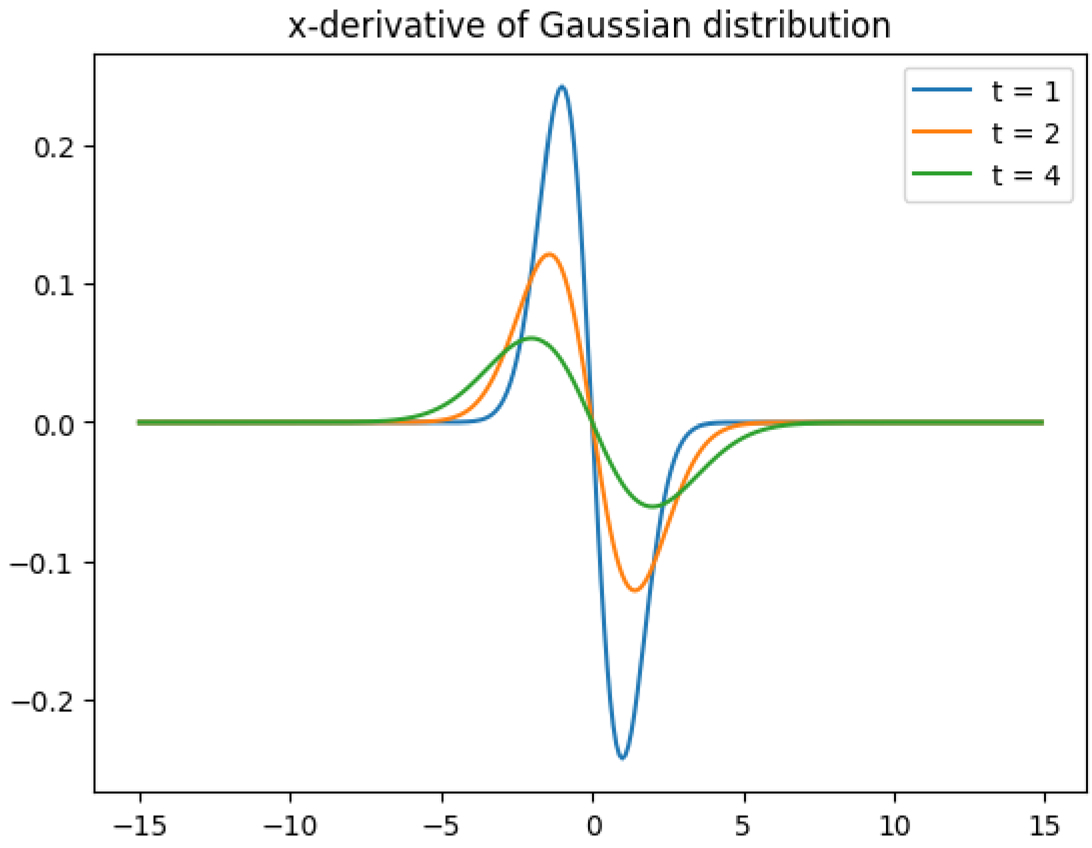
$$\therefore \frac{\partial f}{\partial x}(x,t) = \frac{1}{\sqrt{2\pi t}} \cdot \frac{x}{-t} \cdot e^{-\frac{x^2}{2t}} = -\frac{x}{t} \cdot f(x,t)$$

In [17]:
```
def gd_function_dx(x, t):
    return -1 * x / t * gd_function(x, t)

Y_dx = [gd_function_dx(x, 1) for x in X]
Y_2_dx = [gd_function_dx(x, 2) for x in X]
Y_4_dx = [gd_function_dx(x, 4) for x in X]

plt.plot(X, Y_dx, label='t = 1')
plt.plot(X, Y_2_dx, label='t = 2')
plt.plot(X, Y_4_dx, label='t = 4')
plt.title('x-derivative of Gaussian distribution')
plt.legend()
```

Out[17]:
```
<matplotlib.legend.Legend at 0x7f85cf3eba10>
```



7. Plot the t-derivative of f (x, t)

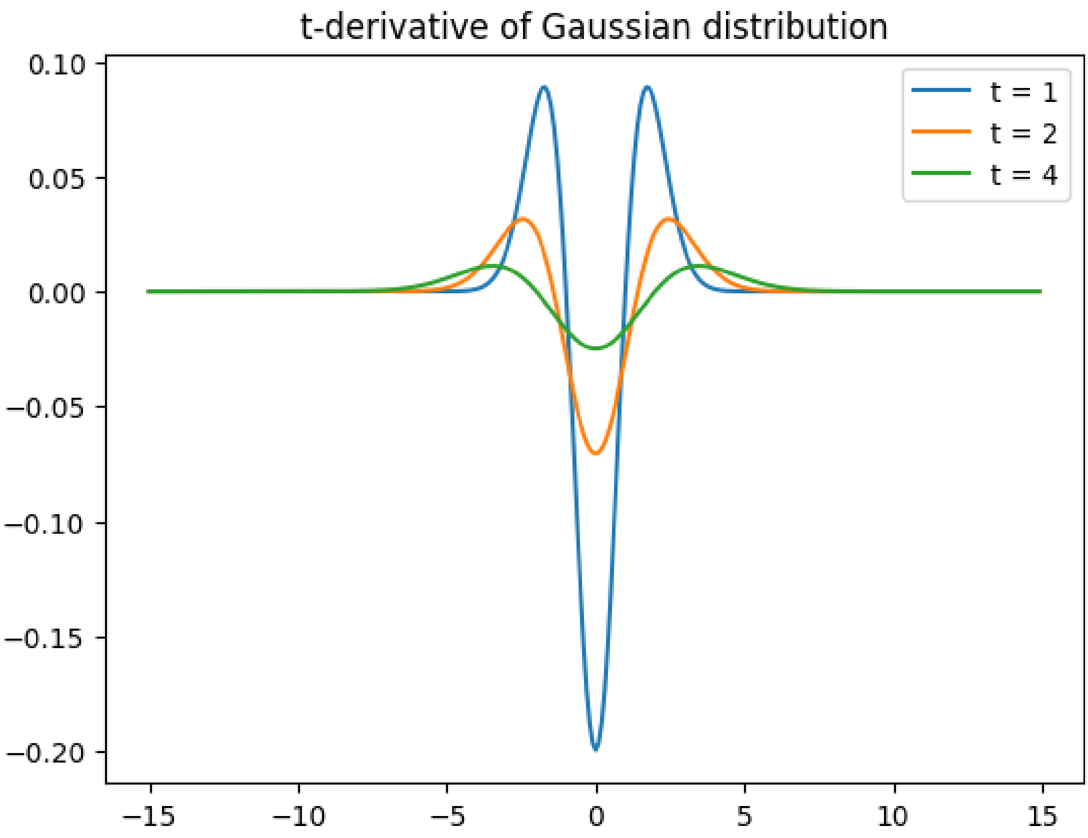
3.7 $$\frac{\partial f}{\partial t}(x,t) = \left(\frac{1}{\sqrt{2\pi t}}\right)'\Big|_t e^{-\frac{x^2}{2t}} + \frac{1}{\sqrt{2\pi t}}\cdot\left(e^{-\frac{x^2}{2t}}\right)'\Big|_t$$
$$= -\frac{1}{2t\sqrt{2\pi t}}\cdot e^{-\frac{x^2}{2t}} + \frac{1}{\sqrt{2\pi t}}\cdot\frac{x^2}{2t^2}\, e^{-\frac{x^2}{2t}}$$
$$= -\frac{1}{2t}\cdot f(x,t) + \frac{x^2}{2t^2} f(x,t)$$
$$= \left(\frac{x^2}{2t^2} - \frac{1}{2t}\right) f(x,t)$$

In [18]:
```
def gd_function_dt(x, t):
    return (x * x / 2 / t / t - 1 / 2 / t) * gd_function(x, t)

Y_dt = [gd_function_dt(x, 1) for x in X]
Y_2_dt = [gd_function_dt(x, 2) for x in X]
Y_4_dt = [gd_function_dt(x, 4) for x in X]

plt.plot(X, Y_dt, label='t = 1')
plt.plot(X, Y_2_dt, label='t = 2')
plt.plot(X, Y_4_dt, label='t = 4')
plt.title('t-derivative of Gaussian distribution')
plt.legend()
```

Out[18]: <matplotlib.legend.Legend at 0x7f85cf460890>



8. Plot the x-derivative of ∂f/∂x (x, t)

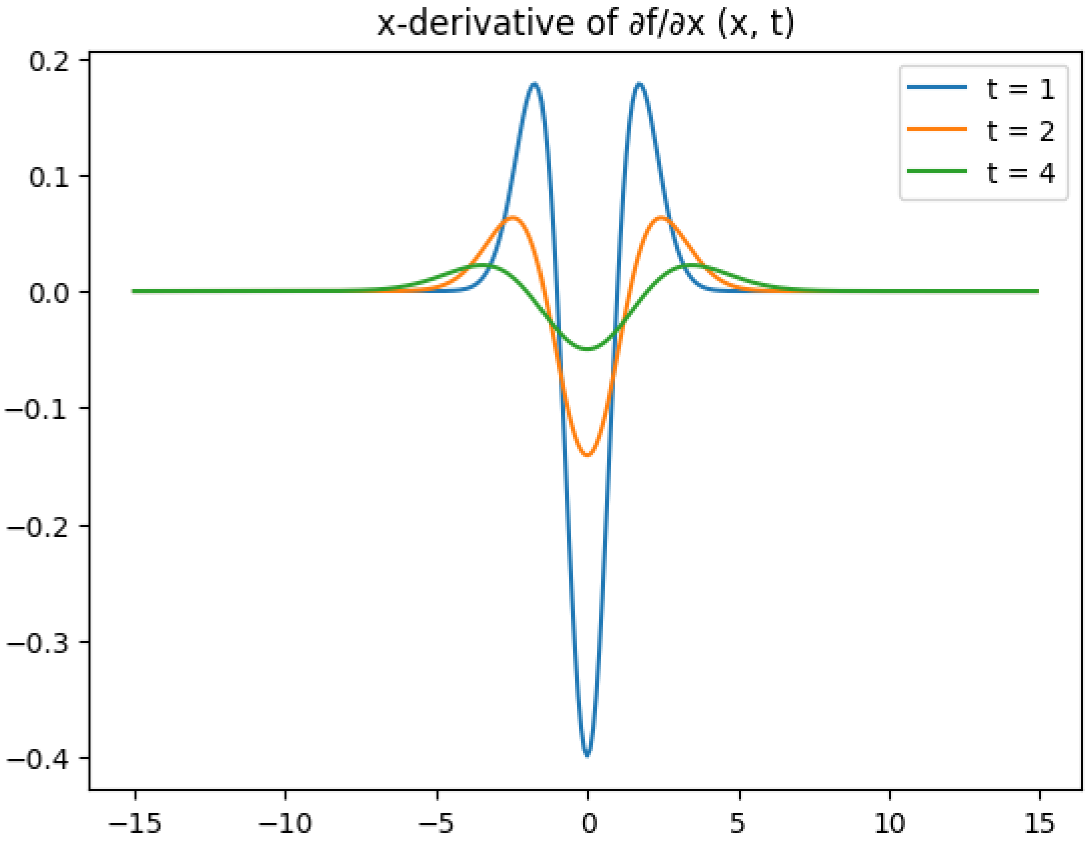
3.8
$$\frac{\partial^2 f}{\partial x^2}(x,t) = \left(-\frac{x}{t}\cdot f(x,t)\right)'\Big|_x$$
$$= -\frac{1}{t}f(x,t) - \frac{x}{t}\cdot f'(x,t)\Big|_x$$
$$= -\frac{1}{t}f(x,t) - \frac{x}{t}\left(-\frac{x}{t}\cdot f(x,t)\right)$$
$$= \left(\frac{x^2}{t^2} - \frac{1}{t}\right)\cdot f(x,t)$$

In [19]:
```
def gd_function_dx_dx(x, t):
    return 2 * gd_function_dt(x, t)

Y_dx_dx = [gd_function_dx_dx(x, 1) for x in X]
Y_2_dx_dx = [gd_function_dx_dx(x, 2) for x in X]
Y_4_dx_dx = [gd_function_dx_dx(x, 4) for x in X]

plt.plot(X, Y_dx_dx, label='t = 1')
plt.plot(X, Y_2_dx_dx, label='t = 2')
plt.plot(X, Y_4_dx_dx, label='t = 4')
plt.title('x-derivative of ∂f/∂x (x, t)')
plt.legend()
```

Out[19]:
```
<matplotlib.legend.Legend at 0x7f85cd1eb110>
```



9. Then show that

3.9. $$\frac{\partial f}{\partial t}(x,t) = \left(\frac{x^2}{2t^2} - \frac{1}{2t}\right) f(x,t)$$
$$= \frac{1}{2}\left(\frac{x^2}{t^2} - \frac{1}{t}\right) f(x,t)$$
$$= \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(x,t)$$