Cody Fizette

1) $f_X(x) = f_Y(x) = \begin{cases} e^{-x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$

Let $Z = X + Y$

Then $f_Z(z) = \int_0^\infty f_X(x) f_Y(z-x)\,dx$

$= \int_0^z e^{-x} e^{x-z}\,dx$

$= \boxed{\begin{matrix} ze^{-z} & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{matrix}}$

2a) $\det(\Sigma) = 4 - 1 = 3$

$\Sigma^{-1} = \frac{1}{\det(\Sigma)} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}$

2b) $f_{X,Y}(x,y) = \frac{1}{\sqrt{12\pi^2}} e^{\left(-\frac{x^2}{3} - \frac{xy}{3} - y^2/3\right)}$

$= \boxed{\frac{1}{2\sqrt{3\pi^2}} \exp\left(-\frac{1}{3}\left(x^2 + xy + y^2\right)\right)}$

2c) $f_{X|Y}(X|y) = \dfrac{f_{X,Y}(x,y)}{f_Y(y)}$ note that $Y \sim N(0,2)$

$\dfrac{y^2}{4}$

$$= \frac{\frac{1}{2\pi\sqrt{3}} \exp\left(-\frac{x^2}{3} - \frac{xy}{3} - \frac{y^2}{3}\right)}{\frac{1}{2\sqrt{\pi}} \exp\left(\frac{-y^2}{4}\right)}$$

$$= \frac{1}{\sqrt{3\pi}} \exp\left(-\frac{x^2}{3} - \frac{xy}{3} - \frac{y^2}{12}\right)$$

$$\boxed{= \frac{1}{\sqrt{3\pi}} \exp\left(\frac{-1}{3}\left(x + \frac{y}{2}\right)^2\right)}$$

2d) $\boxed{\mu_{X|Y} = \frac{-y}{2}}$

Scanned by CamScanner

3a) $X = U - Y \qquad Y = X - V$

$X = U - X + V \qquad Y = U - Y - V$

$X = \frac{U+V}{2} \qquad Y = \frac{U-V}{2}$

$\frac{\partial x}{\partial u} = \frac{1}{2} \qquad \frac{\partial x}{\partial v} = \frac{1}{2}$

$\frac{\partial y}{\partial u} = \frac{1}{2} \qquad \frac{\partial y}{\partial v} = \frac{-1}{2}$

$$\boxed{J = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}}$$

3b) $\det(J) = -\frac{1}{4} - \frac{1}{4} = -1/2$

$f_{u,v}(u,v) = \frac{1}{2} f_{x,y}\left(\frac{u+v}{2}, \frac{u-v}{2}\right)$

$$= \frac{1}{2} \cdot \frac{1}{2\pi} \exp\left(-\frac{1}{2}\left(\left(\frac{u+v}{2}\right)^2 + \left(\frac{u-v}{2}\right)^2\right)\right)$$

$$= \frac{1}{4\pi} \exp\left(-\frac{1}{2}\left(\frac{u^2 + \cancel{2uv} + v^2}{4} + \frac{u^2 - \cancel{2uv} + v^2}{4}\right)\right)$$

$$= \boxed{\frac{1}{4\pi} \exp\left(-\frac{1}{4}(u^2+v^2)\right)}$$

3c) $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Let $\vec{Y} = \begin{bmatrix} u \\ v \end{bmatrix}$, $\vec{X} = \begin{bmatrix} x \\ y \end{bmatrix}$

Then $\vec{Y} = A\vec{X}$

$\mu_{\vec{Y}} = A\mu_{\vec{X}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Sigma_Y = A\Sigma_X A^T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ $\quad \det(\Sigma_Y) = 4 \quad \Sigma_Y^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$

Then $f_{U,V}(u,v) = \frac{1}{4\pi} \exp\left(-\frac{1}{2}\left(\frac{1}{2}u^2 + \frac{1}{2}v^2\right)\right)$

$= \boxed{\frac{1}{4\pi} \exp\left(-\frac{1}{4}(u^2 + v^2)\right)}$

5a) $h(x) = \frac{f(x)}{g(x)} = 2x^{5/2} e^{-x/2}$

$h'(x) = 5x^{3/2} e^{-x/2} - x^{5/2} e^{-x/2} = (5-x)x^{3/2} e^{-x/2}$

not even close to optimal, but it works

$h'(x) = 0$ when $x = 0$, $x = 5$ $\quad h(0) = 0 \quad h(5) = 2 \cdot 5^{5/2} e^{-5/2} < 2 \cdot 5^3 = 250$

So let $\boxed{M = 250}$

In [23]:
```
import numpy as np
import seaborn as sns
from numpy.random import rand
from math import sqrt, pi, exp
from mpl_toolkits import mplot3d
from numpy.linalg import eig
from scipy.stats import expon, gamma
from numpy import log, cos, sin
import matplotlib.pyplot as plt
```

Problem 4a

In [2]:
```
sigma = np.array([[4,2,2], [2,4,0], [0,0,1]])
```

In [3]:
```
eigenvalues, eigenvectors = eig(sigma)
```

In [4]:
```
eigenvalues
```

Out[4]:
```
array([6., 2., 1.])
```

In [5]:
```
eigenvectors
```

Out[5]:
```
array([[ 0.70710678, -0.70710678, -0.68376346],
       [ 0.70710678,  0.70710678,  0.45584231],
       [ 0.        ,  0.        ,  0.56980288]])
```

In [6]:
```
eigenvalues = np.flip(eigenvalues, axis=0)
```

In [7]:
```
U = np.flip(eigenvectors, axis=1)
```

In [8]:
```
lam = np.diag(eigenvalues)
print(lam)
```

```
[[1. 0. 0.]
 [0. 2. 0.]
 [0. 0. 6.]]
```

In [9]:
```
print(U)
```

```
[[-0.68376346 -0.70710678  0.70710678]
 [ 0.45584231  0.70710678  0.70710678]
 [ 0.56980288  0.          0.        ]]
```

Problem 4b

In [10]:
```
sqrt_lam = np.sqrt(lam)
print(sqrt_lam)
```

```
[[1.         0.         0.        ]
 [0.         1.41421356 0.        ]
 [0.         0.         2.44948974]]
```

In [11]:
```
U_sqrt_lam = np.matmul(U, sqrt_lam)
print(U_sqrt_lam)
```

```
[[-0.68376346 -1.          1.73205081]
 [ 0.45584231  1.          1.73205081]
 [ 0.56980288  0.          0.        ]]
```

Problem 4c

In [12]:
```
# Need 1000 vectors in R3, to do this we will generate 3000 independent var
iables to form the vectors
def box_muller(n):
    u = -2*log(rand(n//2))
    v = 2*pi*rand(n//2)
    x = np.sqrt(u)*cos(v)
    y = np.sqrt(u)*sin(v)
    return np.append(x,y)
```

In [13]:
```
random_vars = box_muller(3000)
```

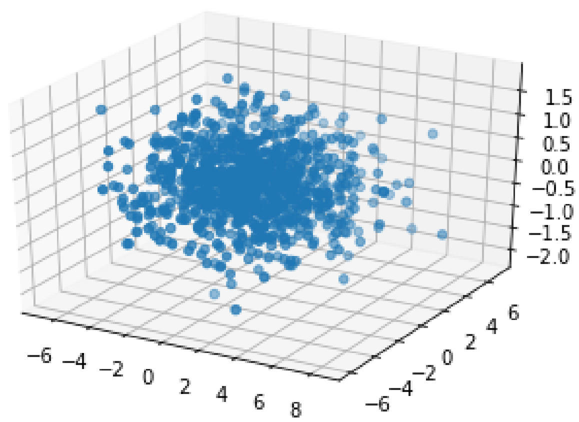
In [14]:
```
random_points = random_vars.reshape((3,-1))
xs=random_points[0]
ys=random_points[1]
zs=random_points[2]
```

In [15]:
```
transformed_points = np.matmul(U_sqrt_lam, random_points)
```

In [16]:
```
xs=transformed_points[0]
ys=transformed_points[1]
zs=transformed_points[2]
```

In [17]:
```
%matplotlib inline
ax = plt.axes(projection='3d')
ax.scatter3D(xs=xs, ys=ys, zs=zs)
```

Out[17]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1a13fab828>



Problem 4d

It seems to spread out along the line y=x. The eigenvector with the largest eigenvalue is [1,1,0] so this makes sense

Problem 5b

In [18]:
```
M = 250
```

In [19]:
```
def solve_problem_5():
    random_numbers = []
    n_expon_generated = 0
    n_accepted = 0
    while n_accepted < 1000:
        u = rand()
        y = expon.rvs(scale=2)
        f_y = y**(5/2)*exp(-y)
        g_y = 0.5*exp(-y/2)
        n_expon_generated += 1
        if u <= f_y/(M*g_y):
            random_numbers.append(y)
            n_accepted += 1
    return random_numbers, n_expon_generated, n_accepted
```

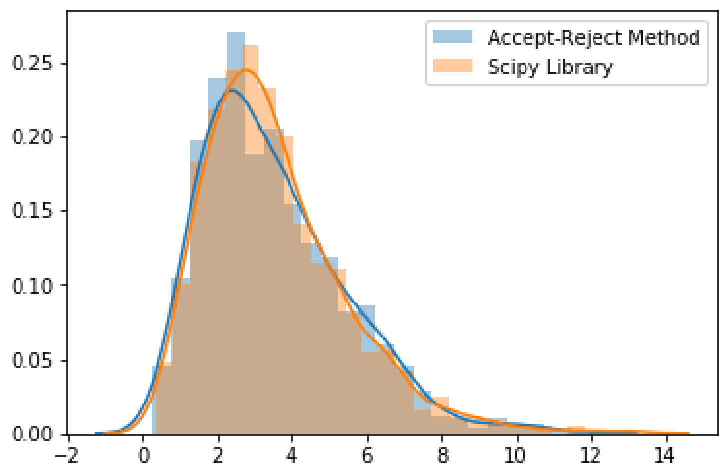
In [20]:
```
random_numbers, n_expon_generated, n_accepted = solve_problem_5()
```

In [29]:
```
real_random_numbers = gamma.rvs(a=7/2, scale=1, size=1000)
```

In [30]:
```
sns.distplot(random_numbers)
sns.distplot(real_random_numbers)
plt.legend(['Accept-Reject Method', 'Scipy Library'])
```

```
/Users/cfizette/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axe
s.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been rep
laced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
/Users/cfizette/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axe
s.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been rep
laced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[30]: `<matplotlib.legend.Legend at 0x1a16ab0198>`



Looks good to me.....

Problem 5c

In [32]:
```
accept_rate = n_accepted / n_expon_generated
accept_rate
```

Out[32]: 0.012517054486673818

The accept rate is around 1.2%