2D_Diff_xi

June 25, 2023

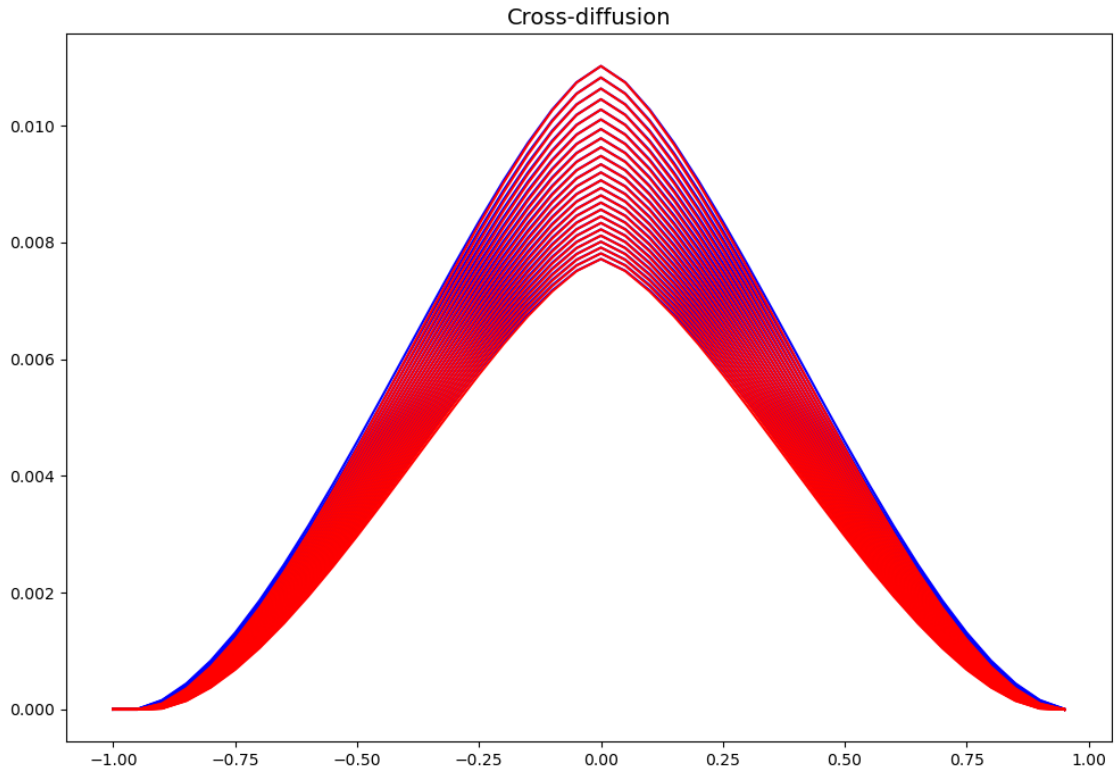
[66]:
```
from math import*
import numpy as np
import matplotlib.pyplot as plt

a=1
a2=a
k=1.0
m=2.3
p=2.1
p2=2.02

t_T=2

alfa=1.0/(1-p)
beta=-alfa*(p-k*m)/(m+1)
beta2=-alfa*(p2-k*m)/(m+1)
g1=1+1.0/m
g2=m/(m*k-1)
A=(fabs(beta)**(1.0/m)/(k*g1*g2))**g2
A2=(fabs(beta2)**(1.0/m)/(k*g1*g2))**g2
print(f"alf={alfa}\tbeta={beta}\tg1={g1}\tg2={g2}\tA={A}\tA2={A2}")

N=40
M=25
b1=-1.0
b2=1.0
T=1.0
hx=(b2-b1)*1.0/N
ht=T*1.0/M
print(f"hx={hx}\tht={ht}")

x=np.zeros(shape=N,dtype=float)
for i in range(0,N):
    x[i]=b1+hx*i
    #print(f"x[{i}]={x[i]}\t")
t=np.zeros(shape=M,dtype=float)
for j in range(0,M):
```

```
    t[j]=j*ht
    #print(f"t[{j}]={t[j]}\t")
xi=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        xi[i,j]=Xi(fabs(x[i]),t[j],-beta)
        #print(f"xi[{i}]={xi[i]}\t")

f=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        f_f=a**g1-(xi[i,j]**g1)
        if(f_f<=0):
            f[i,j]=0
        else:
            f[i,j]=A*(f_f**g2)
            #print(f"f[{i}]={f[i]}\t")

xi2=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        xi2[i,j]=Xi(fabs(x[i]),t[j],-beta2)
        #print(f"xi[{i}]={xi[i]}\t")

f2=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        f_f=a**g1-(xi2[i,j]**g1)
        if(f_f<=0):
            f2[i,j]=0
        else:
            f2[i,j]=A*(f_f**g2)
        #print(f"f[{i}]={f[i]}\t")

u=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        u[i,j]=((t_T+t[j])**alfa)*f[i,j]
        #print(f"u[{i}]={u[i]}\t")
v=np.zeros(shape=(N,M),dtype=float)
for i in range(0,N):
    for j in range(0,M):
        v[i,j]=((t_T+t[j])**alfa)*f2[i,j]
        #print(f"v[{i}]={v[i]}\t")

def Xi(m,p,k):
    return m*((p+t_T)**k)
```

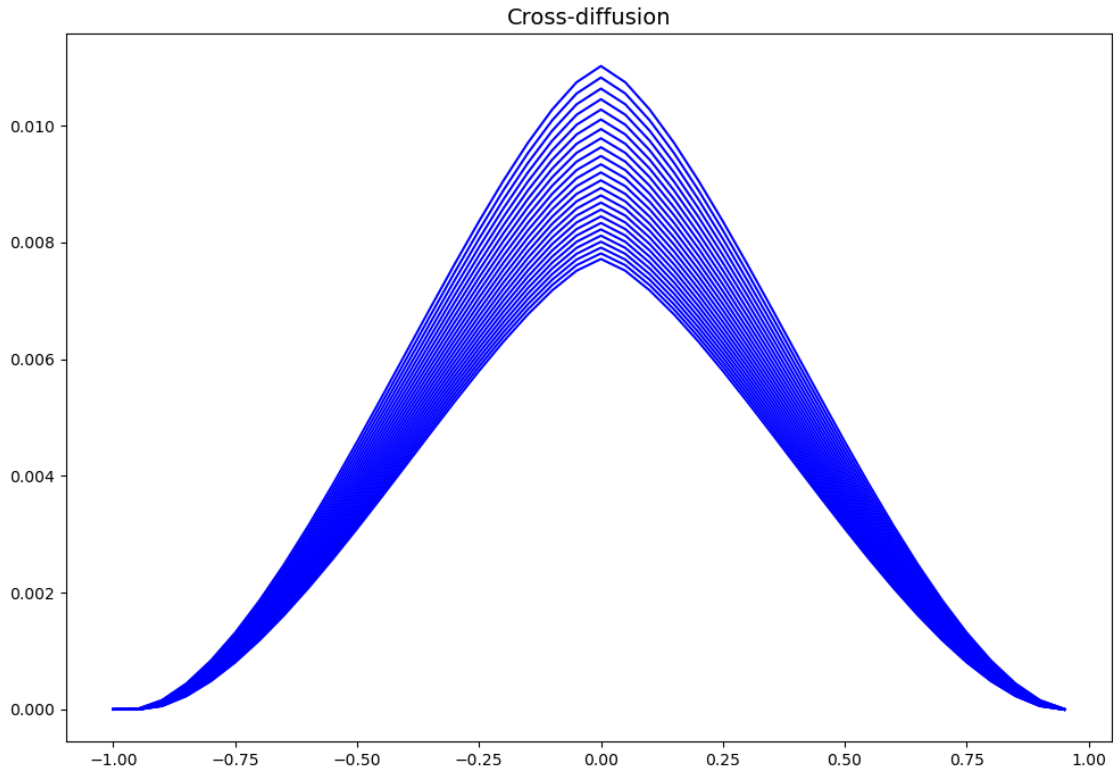
```
def grafik(k,m,p,p2,a,a2):
    xpoints = np.array(x)
    upoints = np.array(u)
    vpoints = np.array(v)
    plt.rcParams["figure.figsize"] = [10, 7]
    plt.rcParams["figure.autolayout"] = True
    fig, ax = plt.subplots()
    ax.plot(x, upoints, label='u', color='blue')
    ax.plot(x, vpoints, label='v', color='red')
    ax.set_title('Cross-diffusion', size=14)
    #plt.legend()
    plt.show()

def grafik_u(k,m,p,p2,a,a2):
    xpoints = np.array(x)
    upoints = np.array(u)
    vpoints = np.array(v)
    plt.rcParams["figure.figsize"] = [10, 7]
    plt.rcParams["figure.autolayout"] = True
    fig, ax = plt.subplots()
    ax.plot(x, upoints, label='u', color='blue')
    ax.set_title('Cross-diffusion', size=14)
    plt.show()

def grafik_v(k,m,p,p2,a,a2):
    xpoints = np.array(x)
    upoints = np.array(u)
    vpoints = np.array(v)
    plt.rcParams["figure.figsize"] = [10, 7]
    plt.rcParams["figure.autolayout"] = True
    fig, ax = plt.subplots()
    ax.plot(x, vpoints, label='v', color='red')
    ax.set_title('Cross-diffusion', size=14)
    plt.show()
```

```
alf=-0.9090909090909091 beta=-0.0550964187327823        g1=1.4347826086956523
g2=1.7692307692307694   A=0.02069448533756767   A2=0.02680777757309592
hx=0.05 ht=0.04
```

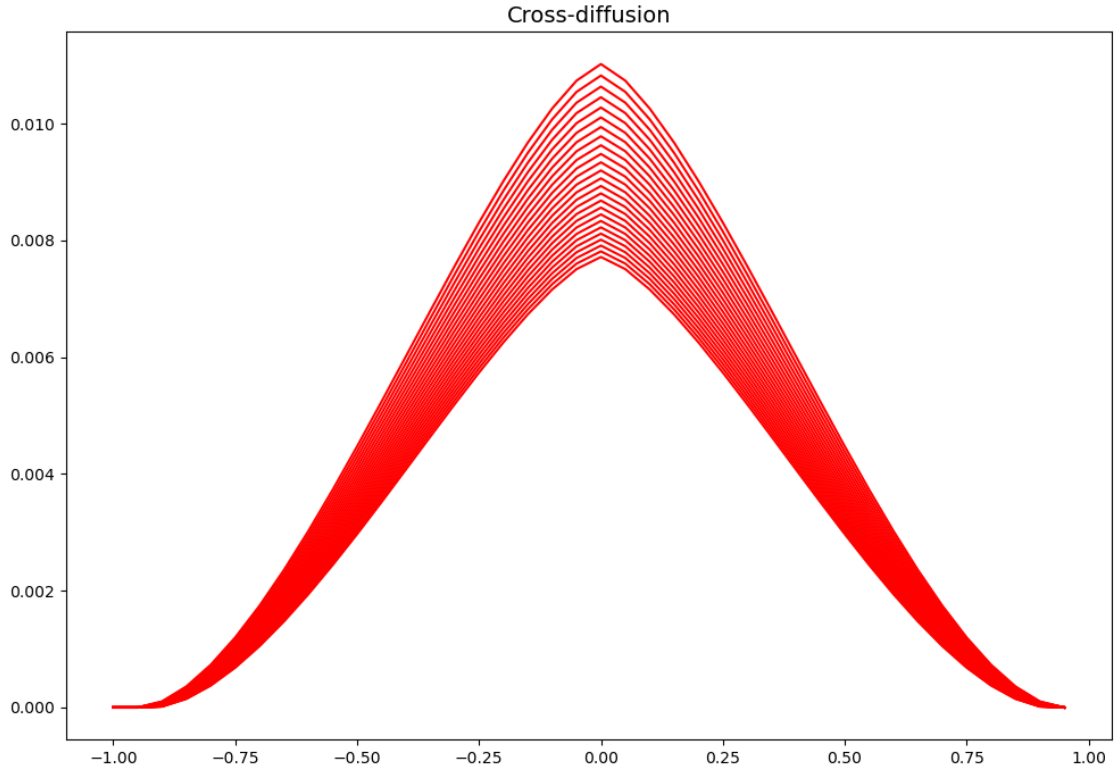
[67]:
```
grafik(1.4,1.7,1.6,1.4,1,1)
```



[68]:
```
grafik_u(1.4,1.7,1.6,1.4,1,1)
```



[69]:
```
grafik_v(1.4,1.7,1.6,1.4,1,1)
```


Cross-diffusion
0.010
0.008
0.006
0.004
0.002
0.000
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

[ ]: