Ввод [1]:

```
from PIL import ImageGrab
from IPython.display import display, Image
def ins(ratio=1.0):
    im_data = ImageGrab.grabclipboard()
    new_size = tuple([int(i*ratio) for i in im_data.size])
    thumb = im_data.resize(new_size)
    fn = "temp.PNG"
    thumb.save(fn)
    img = Image(filename=fn)
    display(img)
```

Ввод [2]:

```
ins(1)
```

1.24. (С использованием «Jupyter Notebook» или «Wolfram Mathematica») Обобщенные биномиальных коэффициентов. Напомним, $\mathrm{C}_n^k = \frac{n!}{k!(n-k)!}$. Вспомним, что $n! = \Gamma(n+1)$, где

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx, \qquad \Re(z) > 0\,.$$

Определим формально символ C_z^w для $z, w \in \mathbb{C}$:

$$\mathrm{C}_z^w = \frac{\Gamma(z+1)}{\Gamma(w+1)\Gamma(z-w+1)}.$$

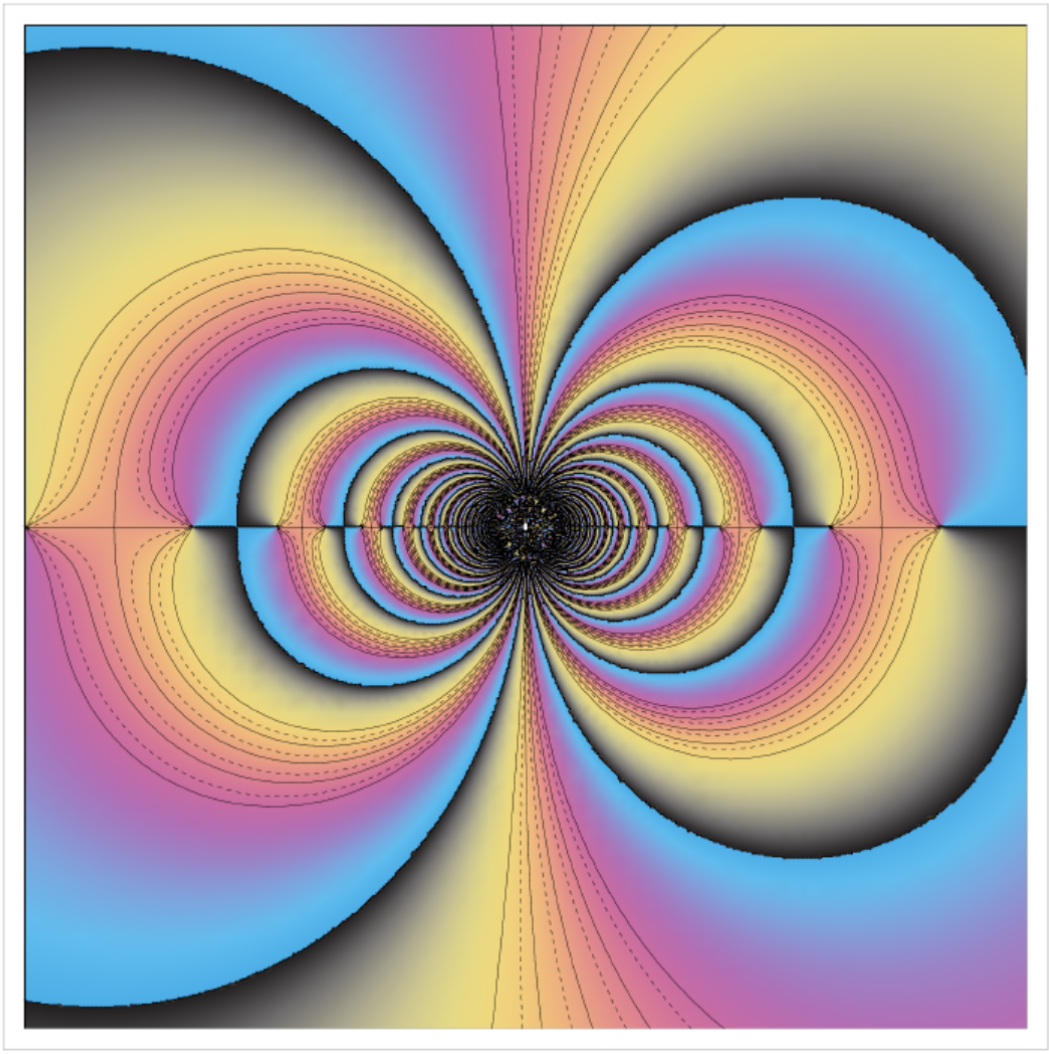
a) Найдите $\mathrm{C}_{1+\mathrm{i}}^{1-\mathrm{i}}$.
Ответ: $4.36745 + 4.85163\mathrm{i}$

b) Найдите $\mathrm{Arg}(\mathrm{C}_{1+\mathrm{i}}^{1-\mathrm{i}})$.
Ответ: 0.837869 **(**$48.0064°$**)**

c) Построить линии уровня $\mathrm{Arg}[\mathrm{C}_z^{\frac{1}{z}} = const]$, где $z = x + \mathrm{i}y$.

Ввод [3]:
```
1 ins(1)
```

Ответ:



Ввод [4]:
```
1 from scipy.special import gamma # импортируем гамма функцию
2 from numpy import angle # Импортируем функцию, которая находит угол комплексного числа
3 import matplotlib.pyplot as plt
4 import numpy as np
```

Ввод [5]:
```
1
2 def C(z, w): # Создаем функцию нахождения биномиальных коэффициентов
3     return gamma(z + 1) / (gamma(w + 1) * gamma(z - w + 1))
```

Ввод [6]:
```
1 print('Ответ: ', C(1 + 1.j, 1 - 1.j))
2 print('Ответ: ', angle(C(1 + 1.j, 1 - 1.j)))
```

```
Ответ:  (4.367449140640135+4.851629765157813j)
Ответ:  0.8378693695500425
```

Ввод [7]:
```
1 rad=angle(C(1 + 1.j, 1 - 1.j))
2 rad
```

Out[7]:
```
0.8378693695500425
```

Ввод [8]:
```
1 np.rad2deg(rad)
```

Out[8]:
```
48.00637865850453
```

Ввод [9]:

```
#from sympy import *
import locale as loc
loc.setlocale(loc.LC_ALL, 'ru')
#init_printing(use_unicode=True,use_latex=True)
```

Out[9]: 'ru'

Ввод [23]:

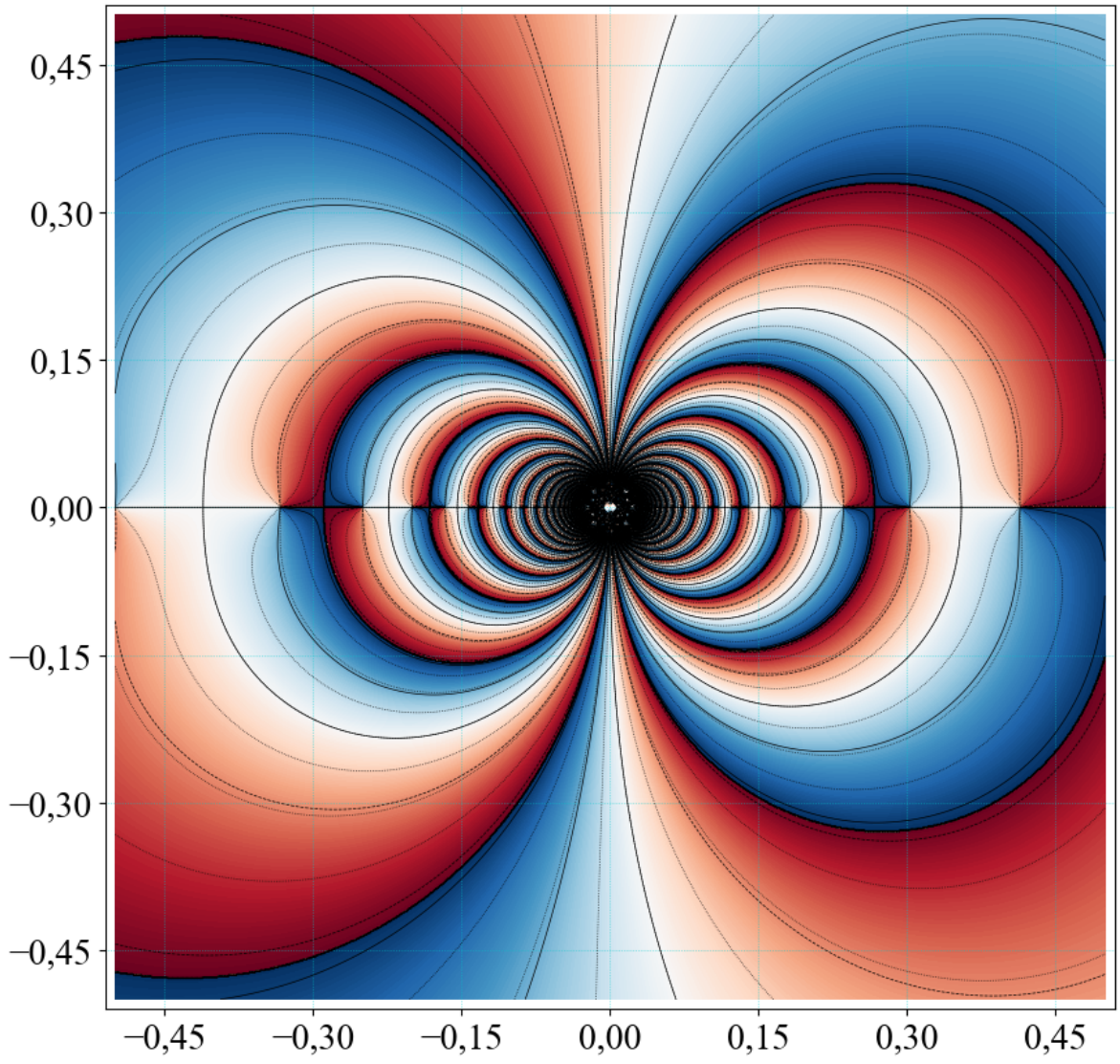
```
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from matplotlib import rcParams
#########################
import locale
locale.setlocale(locale.LC_NUMERIC, 'russian')
plt.rcParams['axes.formatter.use_locale'] = True
#########
#########
plt.rcParams['font.size'] = 16
plt.rcParams["font.family"] = "Times New Roman"
plt.rcParams['mathtext.fontset'] = 'cm'
###################
```

Ввод [25]:

```
import warnings

warnings.simplefilter(action = "ignore", category = RuntimeWarning)
```

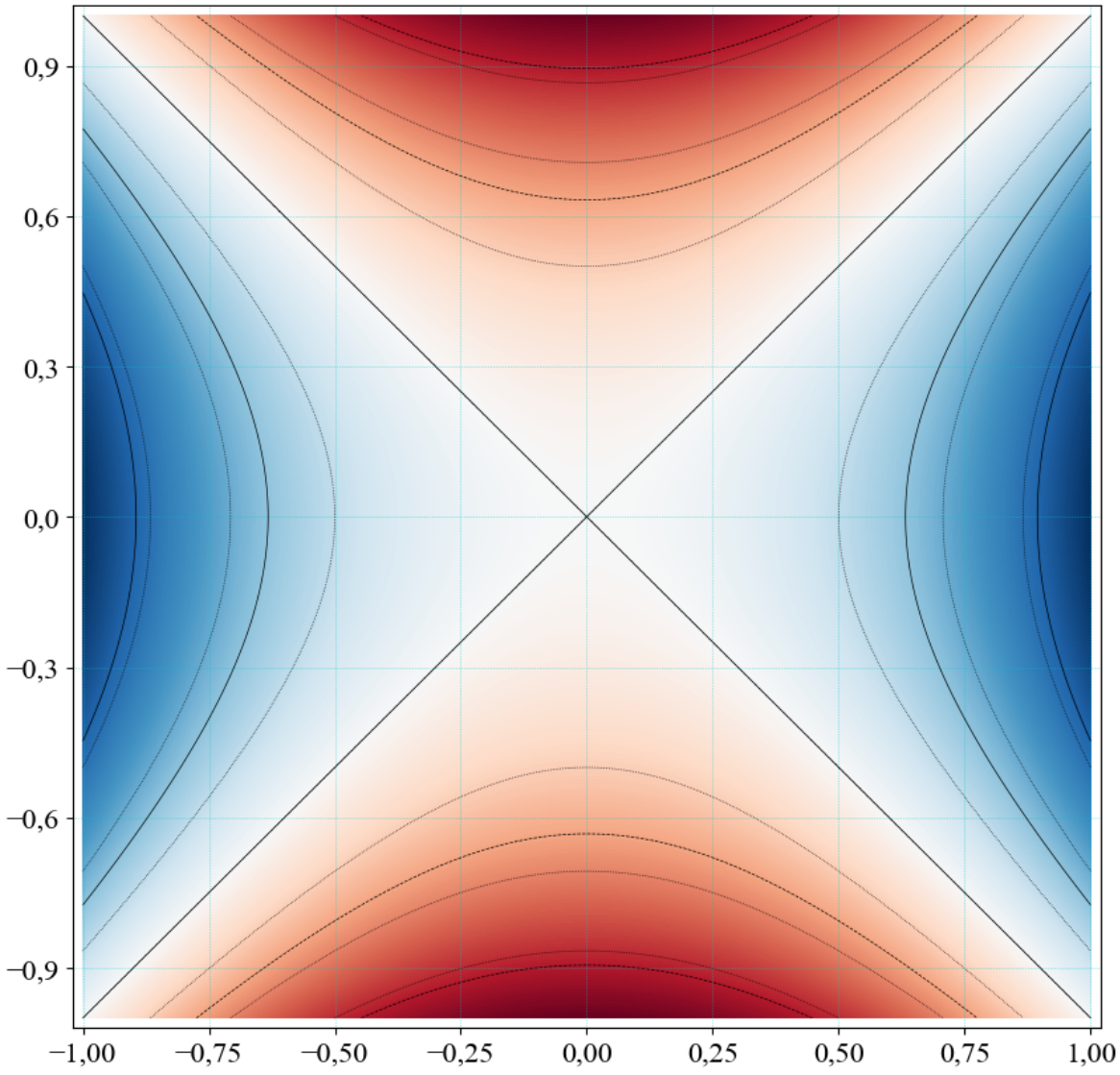
Ввод [35]:

```
lim = 0.5 # Пределы по x и по y
x = np.linspace(-lim, lim, 1000) # Создаем массив из которого будет браться x и y
y = np.linspace(-lim, lim, 1000)

#fig, ax = plt.subplots() # Создаем фигуру, чтобы изменить размер картинки
fig,ax =plt.subplots(figsize=(10, 5))
plt.tick_params(labelsize = 20)
plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=0.5)
ax.xaxis.set_major_locator(ticker.MaxNLocator(10))
ax.yaxis.set_major_locator(ticker.MaxNLocator(8))
#ax.yaxis.set_minor_locator(ticker.MaxNLocator(10))

X, Y = np.meshgrid(x, y) # Создаем пары точек (x, y)
Z = angle(C( X + Y*1j, 1 / (X + Y*1j)))  # Применяем функцию, которая нужна по заданию.

plt.contourf(X, Y, Z,  200, cmap='RdBu')#RdYlBu')#RdBu')#gist_earth')#twilight_shifted')#
plt.contour(X, Y, Z,  7, colors = 'black', linewidths = 0.5, linestyles = 'dotted') # Рис
plt.contour(X, Y, Z,  5, colors = 'black', linewidths = 0.5) # Рисуем сплошные линии уров


fig.set_figwidth(10) # Размер фигуры
fig.set_figheight(10)

ax.set_xlim([-lim - lim/50, lim + lim/50]) # Рамка вокруг фигуры
ax.set_ylim([-lim - lim/50, lim + lim/50])

plt.show()
#fig.savefig("sol_1_24.pdf", bbox_inches='tight')
fig.savefig("sol_1_24.png", bbox_inches='tight')
```



Ввод [32]:

```
lim = 1 # Пределы по x и по y
x = np.linspace(-lim, lim, 1000) # Создаем массив из которого будет браться x и y
y = np.linspace(-lim, lim, 1000)

#fig, ax = plt.subplots() # Создаем фигуру, чтобы изменить размер картинки
fig,ax =plt.subplots(figsize=(10, 5))
plt.tick_params(labelsize = 16)
plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=0.5)
ax.xaxis.set_major_locator(ticker.MaxNLocator(10))
ax.yaxis.set_major_locator(ticker.MaxNLocator(8))
#ax.yaxis.set_minor_locator(ticker.MaxNLocator(10))

X, Y = np.meshgrid(x, y) # Создаем пары точек (x, y)
Z = X**2-Y**2

plt.contourf(X, Y, Z,  700, cmap='RdBu')#gist_earth')#twilight_shifted')#coolwarm')# plas
plt.contour(X, Y, Z,  7, colors = 'black', linewidths = 0.5, linestyles = 'dotted') # Рис
plt.contour(X, Y, Z,  5, colors = 'black', linewidths = 0.5) # Рисуем сплошные линии уров


fig.set_figwidth(10) # Размер фигуры
fig.set_figheight(10)

ax.set_xlim([-lim - lim/50, lim + lim/50]) # Рамка вокруг фигуры
ax.set_ylim([-lim - lim/50, lim + lim/50])

plt.show()
#fig.savefig("sol_1_24.pdf", bbox_inches='tight')
fig.savefig("sol_1_24a.png", bbox_inches='tight')
```



Ввод [ ]:

```
1
```