

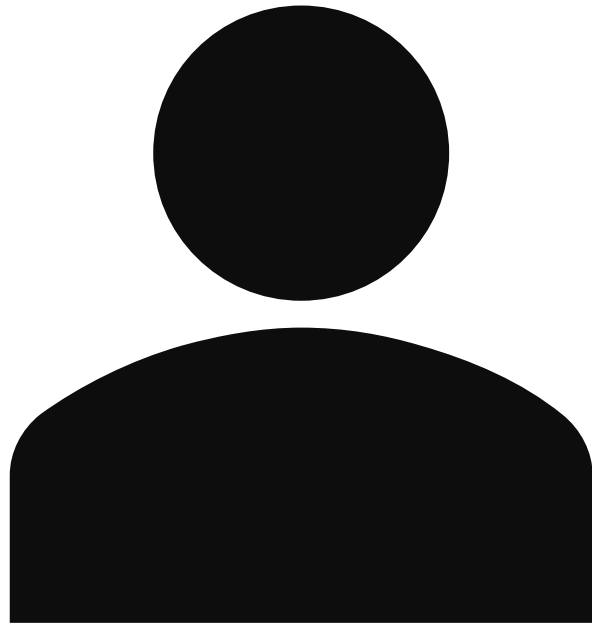


# Numerical Analysis

Dr :Manal Elsaid

Lecture 3 : Special Functions (Gamma )





**Team name : DataVerse**

Made by :  
Zeinab Talaat Antar Samaha

- Project Link :

<https://drive.google.com/file/d/1ICpAsUVGgpRegZKJ4dpflTwEF5soGPL/view?usp=drivesdk>



# OutPut Code

شاشة اختيار اللغة

```
205 def language_selector():
206     root = tk.Tk()
207     root.title("Language / اللغة")
208     root.geometry("300x200")
209     root.configure(bg="#fdf0f9")
210
211     tk.Label(root, text="اختاري اللغة / Choose Language", font=("Segoe UI", 13, "bold"), bg="#fdf0f9", fg="#a24ac3").pack(
212
213     tk.Button(root, text="العربية", font=("Segoe UI", 12), bg="#e0c7ff", fg="#3d0066",
214         |         |         command=lambda: [root.destroy(), arabic_frame()]).pack(pady=5, ipadx=10, ipady=5)
215
216     tk.Button(root, text="English", font=("Segoe UI", 12), bg="#e0c7ff", fg="#3d0066",
217         |         |         command=lambda: [root.destroy(), english_frame()]).pack(pady=5, ipadx=10, ipady=5)
218
219     root.mainloop()
220
221 language_selector()
222
```

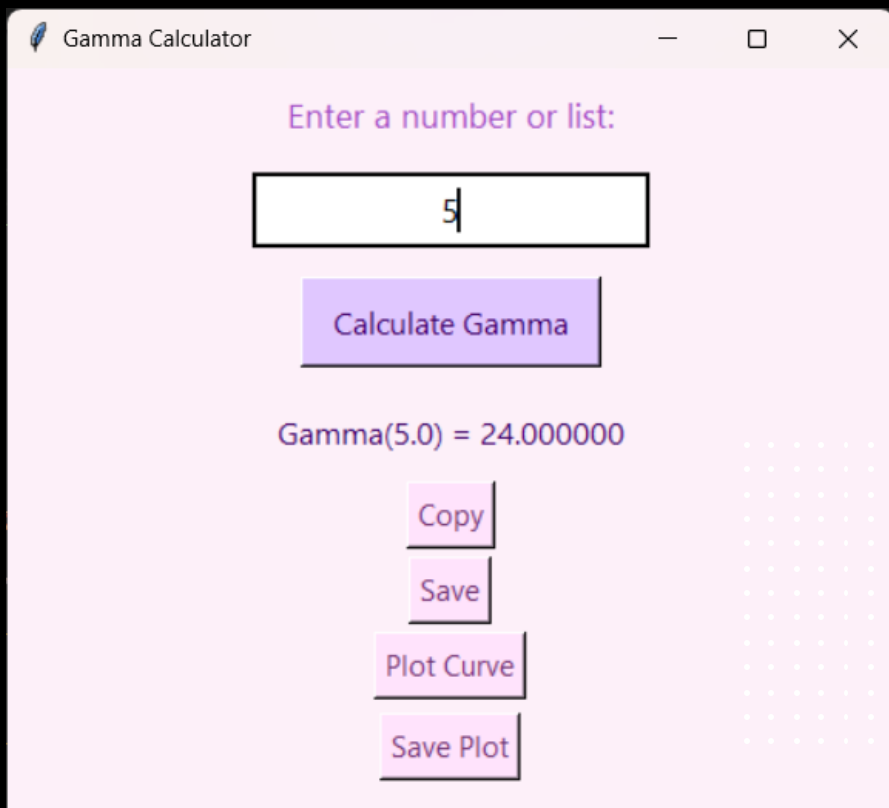


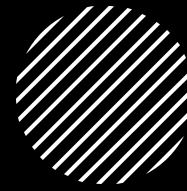
# شاشة اللغة العربية

```
25 def arabic_frame():
26     def calculate_gamma():
27         try:
28             numbers = entry.get().split(",")
29             results = []
30             for num in numbers:
31                 x = float(num.strip())
32                 if x <= 0:
33                     results.append(f"غير صحيح: {x}")
34                 else:
35                     results.append(f"Gamma({x}) = {gamma(x):.6f}")
36             result_text = "\n".join(results)
37             result_label.config(text=result_text)
38             global last_result
39             last_result = result_text
40         except ValueError:
41             messagebox.showerror("خطأ", "أدخل رقمًا أو قائمة أرقام مفصولة بفواصل (مثال: 2.5, 4)")
42
43     def copy_result():
44         if last_result:
45             pyperclip.copy(last_result)
46             messagebox.showinfo("تم نسخ النتيجة", "تم النسخ")
47         else:
48             messagebox.showwarning("تحذير", "لا توجد نتيجة لنسخها")
49
50     def save_result():
51         if last_result:
52             file_path = filedialog.asksaveasfilename(defaultextension=".txt",
53                                                         filetypes=[("Text Files", "*.txt")])
54             if file_path:
55                 with open(file_path, "w", encoding="utf-8") as f:
56                     f.write(last_result)
57                 messagebox.showinfo("تم حفظ النتيجة في", f"تم الحفظ {file_path}")
58             else:
59                 messagebox.showwarning("تحذير", "لا توجد نتيجة لحفظها")
```

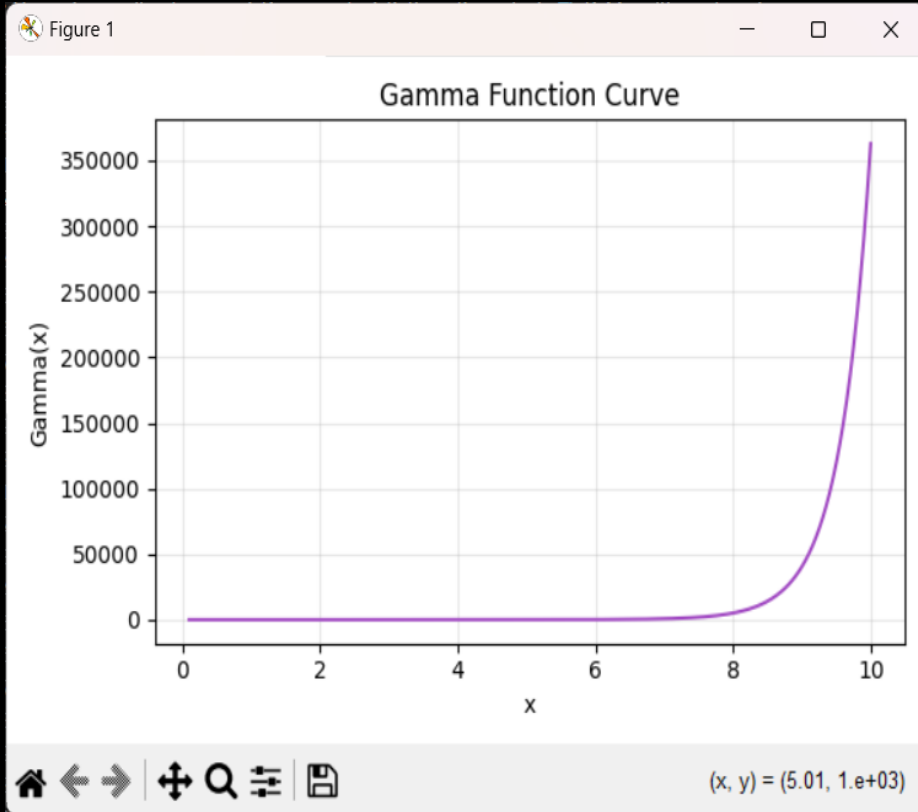
# اللغة الانجليزية شاشة

```
108 def english_frame():
109     def calculate_gamma():
110         try:
111             numbers = entry.get().split(",")
112             results = []
113             for num in numbers:
114                 x = float(num.strip())
115                 if x <= 0:
116                     results.append(f"Invalid: {x}")
117                 else:
118                     results.append(f"Gamma({x}) = {gamma(x):.6f}")
119             result_text = "\n".join(results)
120             result_label.config(text=result_text)
121             global last_result
122             last_result = result_text
123         except ValueError:
124             messagebox.showerror("Error", "Enter a valid number or comma-separated list (e.g. 2.5, 4)")
125
126     def copy_result():
127         if last_result:
128             pyperclip.copy(last_result)
129             messagebox.showinfo("Copied", "Result copied to clipboard.")
130         else:
131             messagebox.showwarning("Warning", "No result to copy.")
132
133     def save_result():
134         if last_result:
135             file_path = filedialog.asksaveasfilename(defaultextension=".txt",
136                                                         filetypes=[("Text Files", "*.txt")])
137             if file_path:
138                 with open(file_path, "w", encoding="utf-8") as f:
139                     f.write(last_result)
140                 messagebox.showinfo("Saved", f"Result saved to:\n{file_path}")
141             else:
142                 messagebox.showwarning("Warning", "No result to save.")
143
```





# منحنى دالة جاما



```
150
151 def plot_gamma_curve():
152     x = np.linspace(0.1, 10, 400)
153     y = gamma(x)
154     plt.figure(figsize=(6, 4))
155     plt.plot(x, y, color="#a24ac3")
156     plt.title("Gamma Function Curve")
157     plt.xlabel("x")
158     plt.ylabel("Gamma(x)")
159     plt.grid(True, alpha=0.3)
160     plt.tight_layout()
161     plt.show()
162
163 def save_plot_image():
164     file_path = filedialog.asksaveasfilename(defaultextension=".png",
165                                               filetypes=[("PNG Image", "*.png")])
166     if file_path:
167         save_gamma_plot(file_path)
168         messagebox.showinfo("Saved", f"Plot image saved to:\n{file_path}")
169
170 win = tk.Toplevel()
171 win.title("Gamma Calculator")
172 win.geometry("460x500")
173 win.configure(bg="#fdf0f9")
174
175 try:
176     img = Image.open("icon.png").resize((60, 60))
177     photo = ImageTk.PhotoImage(img)
178     tk.Label(win, image=photo, bg="#fdf0f9").pack(pady=5)
179     win.iconphoto(False, photo)
180     win.image = photo
181 except:
182     pass
183
184 tk.Label(win, text="Enter a number or list:", font=("Segoe UI", 13), bg="#fdf0f9", fg="#a24ac3").pack(pady=10)
185 entry = tk.Entry(win, font=("Segoe UI", 13), justify="center", relief="solid", bd=2)
```

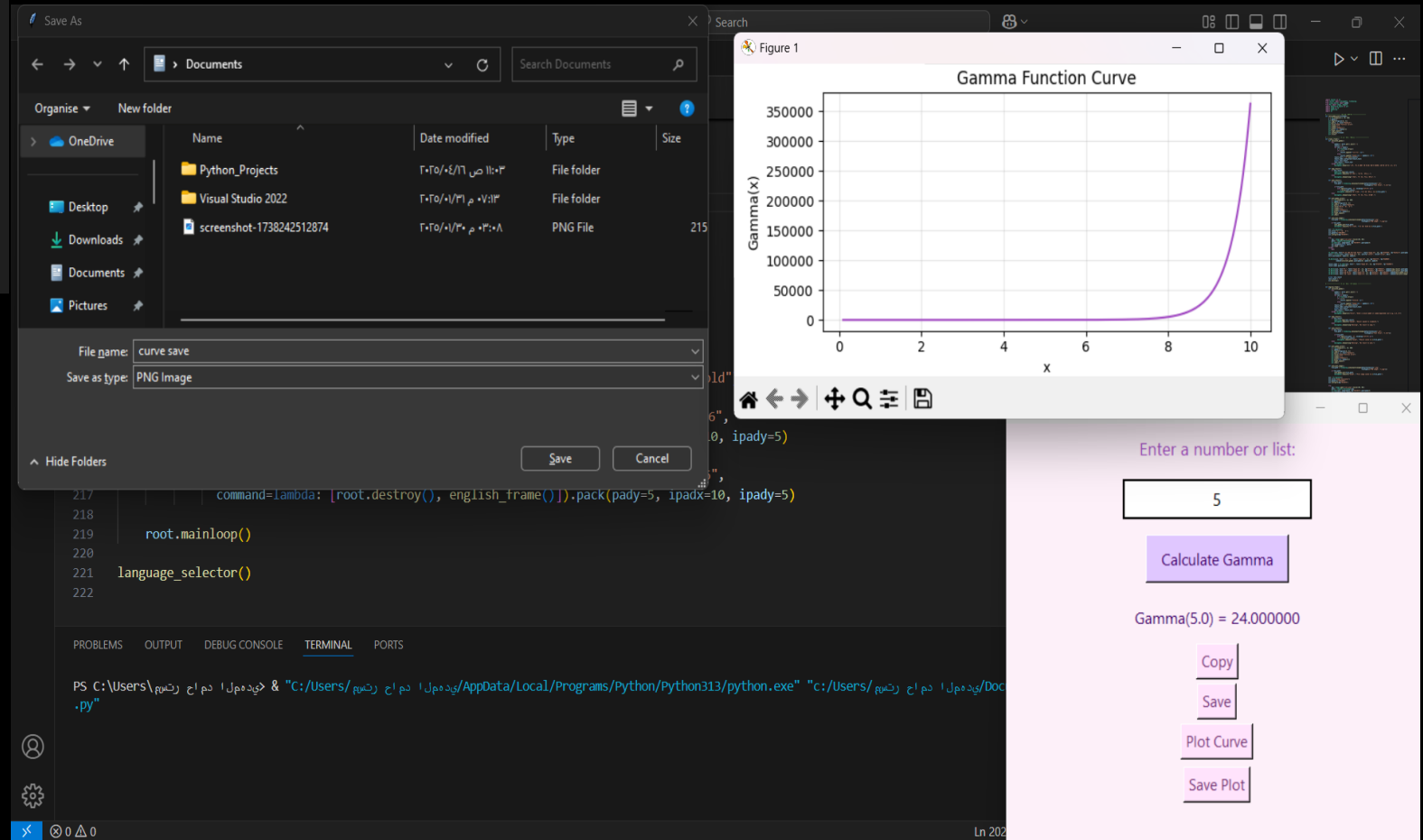
0

```

11 def save_gamma_plot(filename):
12     x = np.linspace(0.1, 10, 400)
13     y = gamma(x)
14     plt.figure(figsize=(6, 4))
15     plt.plot(x, y, color="#a24ac3")
16     plt.title("Gamma Function Curve")
17     plt.xlabel("x")
18     plt.ylabel("Gamma(x)")
19     plt.grid(True, alpha=0.3)
20     plt.tight_layout()
21     plt.savefig(filename)
22     plt.close()
23

```

# حفظ منحنى دالة جاما







Thank you

- Project summary
  - In this project, a graphical user interface was developed using Python and the Tkinter library to calculate the Gamma function for a single number or a list of numbers, with features such as:
- Displaying results clearly.
- Copying or saving results to a file.
- Plotting the Gamma function curve
- Supporting both Arabic and English interfaces