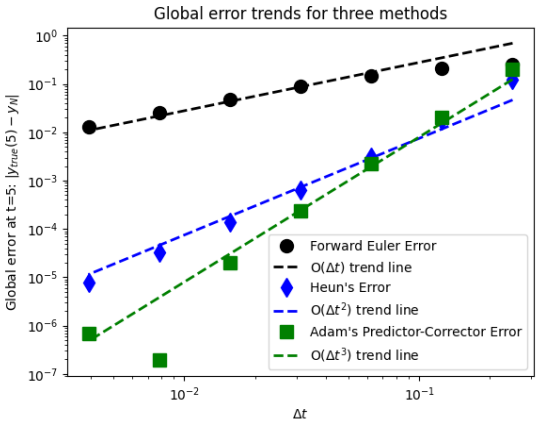
AMATH 581 Homework #1

Tyler Shakibai

1. Log-log Plot of the Error Versus Δt for the Forward Euler Method, Heun's Method, and Adams Predictor-Corrector Method.



Code:

```
dydt = lambda t, y: -3*y*np.sin(t)
y0 = np.pi/np.sqrt(2)
ytrue = lambda t: np.pi*np.exp(3*(np.cos(t) - 1))/np.sqrt(2)

## a - Forward Euler

def forward_euler(f, t, y0):
dt = t[2] - t[1]
y = np.zeros(len(t))
y[0] = y0
for k in range(len(y)-1):
y[k+1] = y[k] + dt*f(t[k], y[k])
return y

dt_vals = 2**(-np.linspace(2, 8, 7))
err = np.zeros(len(dt_vals))
for k, dt in enumerate(dt_vals):
N = int(5/dt)
t = np.arange(0, 5+dt, dt)
y = forward_euler(dydt, t, y0)
err[k] = np.abs(y[-1] - ytrue(5))

A1 = y.reshape(-1, 1)
A2 = err.reshape(1, -1)

pfit = np.polyfit(np.log(dt_vals), np.log(err), 1)
A3 = pfit[0]

fig, ax = plt.subplots()
ax.loglog(dt_vals, err, 'k.', markersize=20, label='Forward Euler Error')
ax.loglog(dt_vals, 2.8*dt_vals, 'k--', linewidth=2, \
```

```
label=r'O($\Delta t$) trend line')

## b - Heun's method

def heun(f, t, y0):
dt = t[2]-t[1]
y = np.zeros(len(t))
y[0] = y0
for k in range(len(y)-1):
y[k+1] = y[k] + 0.5*dt*(f(t[k], y[k]) + f(t[k+1], y[k]+dt*f(t[k], y[k])))
return y

err2 = np.zeros(len(dt_vals))
for k, dt in enumerate(dt_vals):
t = np.arange(0, 5+dt, dt)
yh = heun(dydt, t, y0)
err2[k] = np.abs(yh[-1] - ytrue(5))

A4 = yh.reshape(-1, 1)
A5 = err2.reshape(1, -1)

pfit2 = np.polyfit(np.log(dt_vals), np.log(err2), 1)
A6 = pfit2[0]

ax.loglog(dt_vals, err2, 'bd', markersize=10, markerfacecolor='b', \
label="Heun's Error")
ax.loglog(dt_vals, 0.75*dt_vals**2, 'b--', linewidth=2, \
label=r'O($\Delta t^2$) trend line')

## c - Adams predictor-corrector method

def adam(f, t, y0):
```

```
dt = t[2]-t[1]
y = np.zeros(len(t))
y[0] = y0
y[1] = y0 + dt*f(t[0] + dt/2, y0 + 0.5*dt*f(t[0], y0))
for k in range(1,len(y)-1):
yp = y[k] + 0.5*dt*(3*f(t[k], y[k]) - f(t[k-1], y[k-1]))
y[k+1] = y[k] + 0.5*dt*( \
f(t[k+1], yp) + f(t[k], y[k]) )
return y

err = np.zeros(len(dt_vals))
for k, dt in enumerate(dt_vals):
t = np.arange(0, 5+dt, dt)
y = adam(dydt, t, y0)
err[k] = np.abs(y[-1] - ytrue(5))

A7 = y.reshape(-1, 1)
A8 = err.reshape(1, -1)

pfit = np.polyfit(np.log(dt_vals), np.log(err), 1)
A9 = pfit[0]

### 2D Plot

ax.loglog(dt_vals, err, 'gs', markersize=10, markerfacecolor='g', \
label="Adam's Predictor-Corrector Error")
ax.loglog(dt_vals, 8*dt_vals**3, 'g--', linewidth=2, \
label=r'O($\Delta t^3$) trend line')
ax.legend(loc='lower right')
ax.set_xlabel(r'$\Delta t$')
ax.set_ylabel(r'Global error at t=5: $|y_{true}(5) - y_N|$')
ax.set_title('Global error trends for three methods')
```