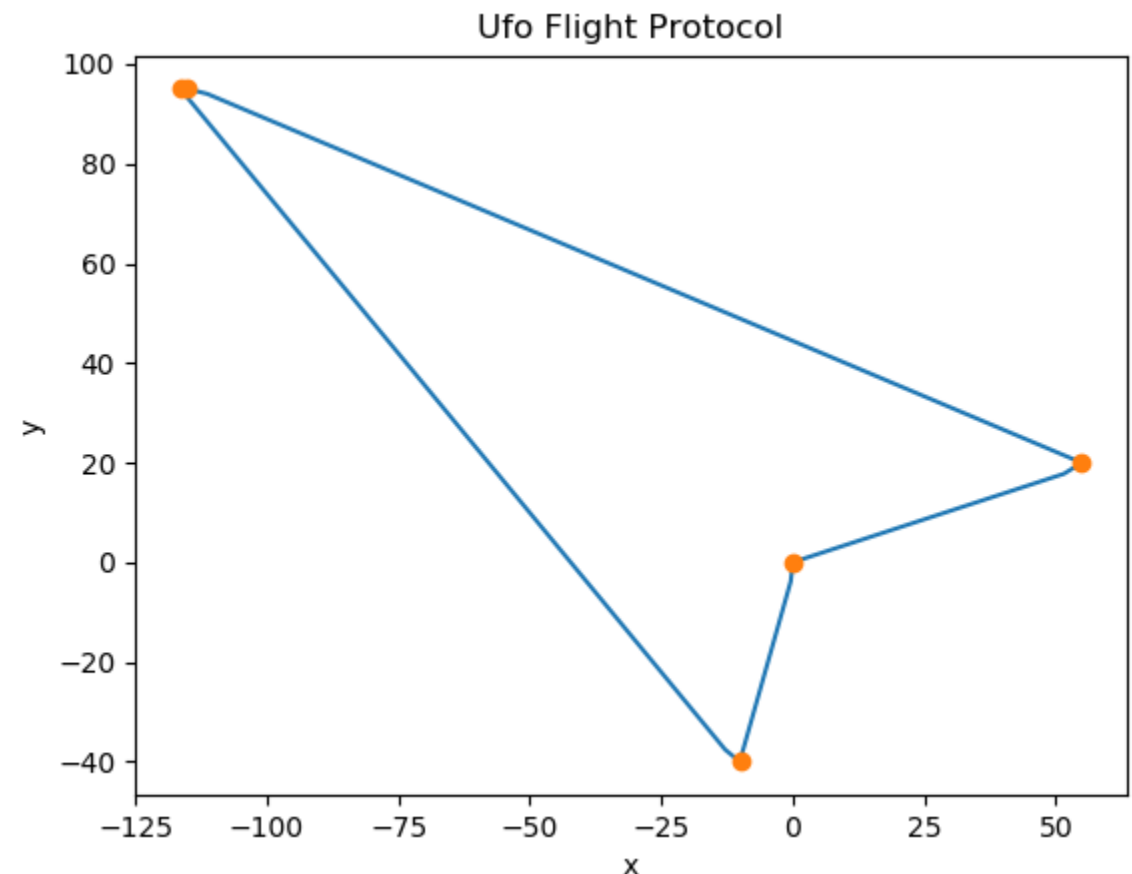


- Im **vierten Teil des Praktikums** soll der Flug des Ufos protokolliert werden:

- Es soll der Flug in der zweidimensionalen Draufsicht dargestellt werden.
- Es reicht, wenn die Punkte, an denen eine Richtungsänderung erfolgt, abgespeichert werden. Diese Punkte können dann durch Linien verbunden werden.
- Es soll die plot-Funktion von matplotlib verwendet werden.
- Das Protokoll der in der letzten Praktikumsaufgabe verwendeten Beispielroute könnte etwa folgendermaßen aussehen:





1. Erstellen Sie eine neue Datei `ufo_protocol.py` und legen Sie darin zwei globale Variablen an:

```
protocol_x = []  
protocol_y = []
```

2. Implementieren Sie in `ufo_protocol.py` die folgende Funktion:

```
def add_to_protocol(p):
```

Parameter: tuple p : Punkt bestehend aus zwei float

Rückgabewert: keiner

Die Funktion hängt die erste Komponente von p hinten an `protocol_x` an und entsprechend die zweite Komponente an `protocol_y`.

3. Rufen Sie die Funktion `add_to_protocol` an geeigneten Stellen in der Funktion `cruise` auf, sodass alle nötigen Punkte zum Zeichnen des Flugweges abgespeichert werden.



4. Erstellen Sie in ufo\_protocol.py die folgende Funktion:

```
def plot_protocol(destinations):
```

Parameter: list destinations : Ziele (Liste von Tupel mit je zwei float)

Rückgabewert: keiner

Die Funktion zeichnet das in protocol\_x und protocol\_y gespeicherte Flugprotokoll mit Hilfe der plot-Funktion von matplotlib (=blaue Linien im Beispiel auf einer der vorangegangenen Folien).

Außerdem sollen die Ziele des Ufos eingezeichnet werden. Deshalb hat die Funktion den Parameter destinations. Der Start- und Endpunkt soll auch geplottet werden (=orange Kreise im Beispiel).

Der Plot und die Achsen sollen beschriftet sein (siehe Beispiel).

5. Rufen Sie die Funktion plot\_protocol am Ende des Hauptprogramms in ufo\_main.py auf.

- Abgabe:
  - Testen Sie das fertige Programm ausgiebig. Wenn alle Tests erfolgreich waren, verpacken Sie die vier py-Dateien `ufo_routing.py`, `ufo_autopilot.py`, `ufo_protocol.py` und `ufo_main.py` in eine zip-Datei. Laden Sie die zip-Datei anschließend in Moodle hoch. Zwar hat sich `ufo_routing.py` nicht geändert, packen Sie die Datei aber trotzdem mit ein.
  - Bitte laden Sie die zip-Datei rechtzeitig hoch.
- Checkliste: Überprüfen Sie vor dem Hochladen der Abgabe, ob Folgendes erfüllt ist:
  - Die Abgabe ist eine zip-Datei, die vier py-Dateien enthält.
  - Die py-Datei `ufo_routing.py` enthält zusätzlich zu den bereits vorhandenen Funktionen die beiden neuen Funktionen `add_to_protocol`, `plot_protocol` und zwei globale Listen `protocol_x`, `protocol_y`. Außer den vorhandenen und neuen Funktionen, der beiden globalen und Listen und der benötigten Import-Anweisungen ist kein weiterer Code enthalten.

- Die py-Datei `ufo_main.py` enthält das an nur zwei Stellen ergänzte Hauptprogramm. Außer dem Hauptprogramm und benötigten Import-Anweisungen ist kein weiterer Code enthalten.
- Die py-Datei `ufo_routing.py` ist unverändert.
- Das Programm hat keine Syntaxfehler.
- Das Programm ist mit verschiedenen Eingabewerten, d.h. Zielelisten, getestet und fehlerfrei. Es reicht, wenn Sie als Routing-Strategie „kürzesten Route“ verwenden.