



Univerzitet u Novom Sadu
Prirodno - matematički fakultet,
Departman za fiziku

INTERPOLACIJA POLJA TEMPERATURE KORISTEĆI PROGRAMSKI JEZIK PYTHON

-MASTER RAD-

Mentor: Prof.dr Ilija Arsenić

Kandidat: Martin Petraš

Novi Sad, 2018.

Sadržaj

1	Postupak za instaliranje	3
1.1	Provera verzije pythona	3
1.2	Izbegavanje problema sa dve verzije	4
2	Korišćenje biblioteka	5
2.1	NumPy	6
2.2	Pandas	6
2.3	MetPy	7

Python

Programski jezik Python spada u programske jezike visokog nivoa. Njegova sintaksa omogućava pisanje veoma preglednih programa, brzo i lako uči. Programski jezik Python nastao je krajem devedesetih godina prošlog veka. Njegov autor je Gvido van Rosum. Uspeh Python-a se oslanja na nekoliko prednosti koje pruža kako početnicima tako i stručnjacima: Python se lako uči. Broj funkcija u samom jeziku je skroman, pa zahteva relativno malo uloženog vremena ili napora da se naprave prvi programi. Pythonova sintaksa je dizajnirana da bude čitljiva i jednostavna. Ova jednostavnost čini Python idealnim nastavnim jezikom i omogućava početnicima da brzo napreduju. Programeri provode više vremena razmišljajući o problemu koji pokušavaju da reše, a manje vremena razmišljaju o kompleksnosti jezika ili dešifrujući kod koji su pisali drugi. Najosnovnije korišćenje Pythona je kao jezik skriptovanja i automatizacije, koristi se za nauku o podacima i mašinsko učenje, za veb usluge, metaprogramiranje. Njegova popularnost se ogleda u velikom broju biblioteka. Što se tiče nedostataka istakli bi njegovu brzinu. Ono za što Pythonu treba šest sekundi u drugom programskom jeziku se može uraditi za delić sekunde. Međutim, brzina kojom se može napisati funkcionalan program je daleko brži nego u nekom drugom jeziku.

Uvod

U ovom radu biće predstavljeno kako uz pomoć Pythona i njegove biblioteke Metpy možemo nacrtati polje temperature. Konkretno će obrađena oblast u području planina Karpati. Biće pokazano kako instalirati odgovarajuće biblioteke potrebne za rad. Osvrnućemo se i na postupak interpolacije koji je korišćen kod metpy biblioteke. Prvi deo rada obuhvatiće postupke za konfigurisanje a u drugom delu će biti predstavljeni rezultati.

1 Postupak za instaliranje

Python kao programski jezik postoji za sve tri popularne sistemske platforme Windows, Linux i OSX. Ovde će biti prikazan postupak instaliranja potrebnih paketa za rad pod Linux operativnim sistemom. Konkretno ćemo raditi na distribuciji baziranoj na Debian(stable) paketima, međutim za ostale distribucije poput Suse, Fedora, Arch postupak je sličan. Prvo što treba da uradimo jeste da odemo na <https://www.anaconda.com/download/#linux> i preuzmemo *Anaconda3-5.2.0-Linux-x86_64.sh*¹. U pitanju je *shell* skripta koja se pokreće u terminalu na sledeći način:

```
bash Anaconda3-5.2.0-Linux-x86_64.sh
```

Nakon što se prihvati licenca određujemo gde da instaliramo Anacondu, najbolje je da se ostavi podrazumevana putanja */home/USER/anaconda*. Nakon završene instalacije najbolje je uraditi modifikaciju *.bashrc* dodavanjem sledeće putanje:

```
export PATH=/home/$USER/anaconda/bin:$PATH
```

Nakon ovoga uspešno smo instalirali anacondu koju ćemo koristiti za instaliranje odnosno ažuriranje Python paketa. Paketi se uz pomoć Anaconde instaliraju upisivanjem sledeće komande u terminal:

```
conda install ime_paketa
```

Za ažuriranje paketa koristi se sledeća komanda:

```
conda update ime_paketa
```

Postoji još jedan metod kojim se mogu instalirati paketi, a to je uz pomoć komande *pip*:

```
pip install ime_paketa
```

Međutim, jako bitno je istaći da ne treba mešati ova dva metoda za instaliranje paketa. Najbolje je da se izabere jedan od njih i uvek koristi, jer mogu nastati problemi ako se koriste oba metoda.

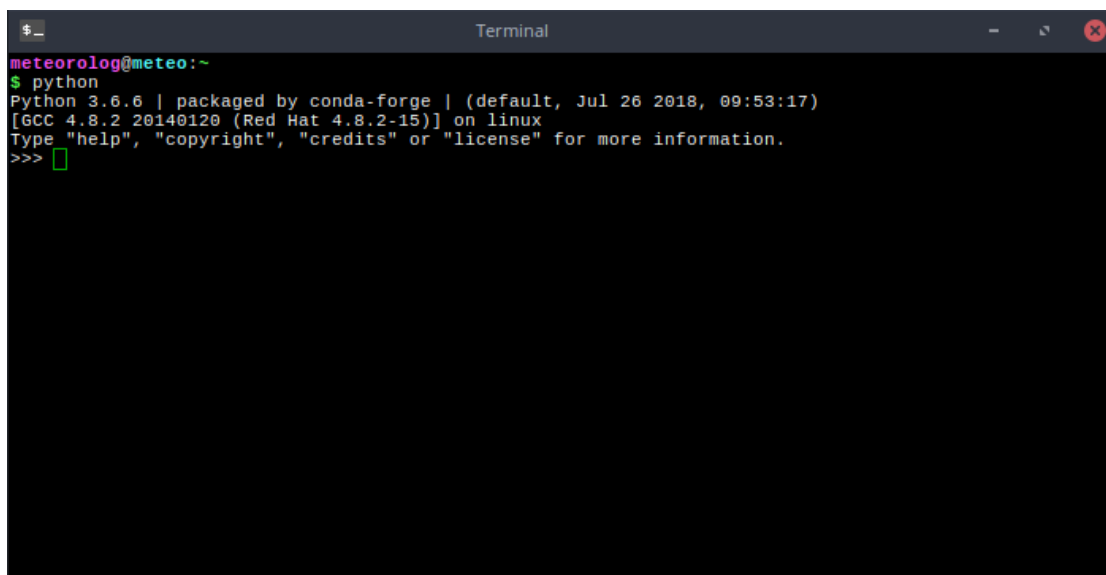
1.1 Provera verzije pythona

Prvo proveravamo da li imamo instaliran python kucanjem sledeće komande:

```
python
```

Ukoliko je python instaliran pojavi se sledeće. Uglavnom sve distribucije linuxa dolaze sa instaliranim pythonom, razlika može biti u verziji. Postoje dve verzije pythona, 2.7 čija podrška polako prestaje i novija 3.x (trenutna je 3.6.6).

¹U pitanju je trenutna verzija

A terminal window titled "Terminal" with a dark background. The prompt is "meteorolog@meteo:~". The command "python" has been entered, and the output shows: "Python 3.6.6 | packaged by conda-forge | (default, Jul 26 2018, 09:53:17) [GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux". It also includes instructions to type "help", "copyright", "credits" or "license" for more information. The prompt ">>>" is visible at the end of the line.

```
meteorolog@meteo:~  
$ python  
Python 3.6.6 | packaged by conda-forge | (default, Jul 26 2018, 09:53:17)  
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

Slika 1. Pokretanje pythona

Razlike između ove dve verzije postoje, ali metpy biblioteka koja nama je potrebna podržava obe verzije. Više o razlikama između ove dve verzije pogledajte na <https://wiki.python.org/moin/Python2orPython3>. Koja verzija pythona je instalirana možemo proveriti i kucanjem sledećeg koda u terminal:

```
python --version  
Python 3.6.6
```

U budućе svaki kod će biti vezan za ovu verziju.

1.2 Izbegavanje problema sa dve verzije

Kao što sam već spomenuo sve linux distribucija dolaze sa instaliranim pythonom, razlike mogu biti u verziji. Ukoliko vaša linux distribucija ima instaliranu samo verziju 2.7 a vi bi hteli i verziju 3.x, onda treba obratiti pažnju na sledeće. Kod mene, ja koristim MX linux distribuciju baziranu na Debian(stable) verziji, kucanjem u terminalu:

```
python --version  
Python 3.6.6
```

prikazuje da je podrazumevana verzija 3.6. Međutim, ukoliko je kod vas instalirana samo verzija 2.7 a vi dodatno instalirate verziju 3.x, može se desiti da će verzija 2.7 i dalje ostati podrazumevana, pa se prilikom kompajliranja može javiti problem. Pošto su kod mene obe verzije, kao podrazumevana verzija je 3.6.6 dok je za pozivanje verzije 2.7 potrebno kucati sledeće:

```
$ python
Python 3.6.6 | packaged by conda-forge | (default, Jul 26 2018, 09:53:17)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Slika 2. Pokretanje pythona verzije 3.6.6

```
python2 --version
Python 2.7.13
```

```
$ python2
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Slika 3. Pokretanje pythona verzije 2.7

Kako bi otklonili ovaj problem, u direktorijumu `/home/user/.bashrc`, koristeći jedna od tekst editora, upišemo sledeću komandu:

```
alias python=python3
```

Ovom komandom prilikom svakog pozivanja pythona iz terminala kao podrazumevana verzija će biti 3.6.6, što će biti jako bitno u daljem radu.

2 Korišćenje biblioteka

Mnoge python funkcije su sadržane u specijalizovanim bibliotekama, tzv. modulima. Kako bi se ove funkcije mogle koristiti, one se moraju učitati. Učitavanje modula se postiže naredbom `import`. Ova mogućnost je pythonu donela popularnost jer postoje monogo modula koji jako pomažu prilikom rada. Python je svoju popularnost stekao pri izradi veb programa međutim usavršavanjem podrške za dodtane module otvorilo je vrata pythonu i u drugim oblastima. Moduli poput *numpy* i *pandas*, koji se koriste za analizu i vizuelno prikazivanje podataka, su python svrstali uz rame sa ostalim kako komercijalnim programima tako i sa programima otvorenog koda kao što su R, MATLAB, SAS, Stata i ostali. Kada pogledamo da python spada u besplatni programski jezik, njegova popularnost nije slučajnost. Jedan deo ove popularnosti u obradi podataka pripada i laka integracija sa C, C++ i FORTRAN kodom. Pošto su moduli jako značajni za python istaći ćemo neke od naznačajnijih. Samo da napomene, sve nabrojane mogućnosti koje pružaju biblioteke je moguće izvesti koristeći isključivo samo Pythonov kod.

Prednost biblioteka je u efikasnosti i lakoći kojim se postižu isti rezultati, uz manje utrošenog vremena.

2.1 NumPy

NumPy (Numerical Python) je jedan od glavnih modula koji se koristi za numeričke proračune u pythonu. Sadrži sledeće stvari:

- objekt *ndarray*, brz i efikasan višedimenzioni niz
- sadrži funkcije za brzi rad sa redovima
- alate za čitanje i pisanje nizova iz datoteka
- operacije sa linearnom algebrom, Furijeve transformacije, nasumično kreiranje brojeva
- omogućava interakciju sa C i C++ kodom

Za numeričke proračune, nizovi kod NumPy su dosta efikasniji prilikom njihove manipulacije nego bilo koja druga izgrađena struktura unutar Pythona. Instaliranjem Anaconde ova biblioteka bi trebalo da bude instalirana, ukoliko to nekim slučajem nije, to se radi na sledeći način:

```
conda install numpy
```

Ukoliko želimo ažurirati novu verziju:

```
conda update numpy
```

2.2 Pandas

Pandas je dizajniran kako bi se ubrzao rad sa strukturiranim ili tabelarnim podacima i kao takav, postao jako moćna i produktivna alatka za analizu podataka. Sadrži sve osobine NumPy s mogućnošću brže i efikasnije manipulacije bazom podataka. Brzo i lako indeksiranje, preoblikovanje, sečenje, spajanje, brisanje su samo od nekih mogućnosti koje pruža ova biblioteka u radu sa podacima. Instaliranje pandas biblioteke:

```
conda install pandas
```


2.3 MetPy

Metpy predstavlja kolekciju alata u Pythonu za pregled, vizuilizaciju i proračuna meteoroloških podataka. Paket je nastao od strane *Unidata* udruženja, koja se bavi izradom alata za naučne svrhe. MetPy paket sadrži dosta opcija, koristi se za vizelizaciju radarskih slika, rad sa netCDF podacima, vizuelizaciju radio-sondažnih merenja, vizuelizaciju meteoroloških podataka i ostalo. Mi ćemo koristiti ovu biblioteku u svrhu interpolacije polja temperature. Za instaliranje MetPy u terminal kucamo:

```
conda install -c -conda-forge metpy
```

Instaliranjem ovog paketa dobijamo razne alate, nama će najviše značiti mogućnost interpolacije za polje temperature kao i njegove vizuelizacije. Za vizuilizaciju se koristi *Cartopy*, Pythonov paket za crtanje mapa.