نکات خاص pencile code:

چطور چند هسته را به یک خروجی بفرستیم؟

در makefile باید نوع IO را بصورت io_collect تعریف کنیم ، در این صورت داده های هسته های مختلف تجمیع می شوند و در data/allprocs ذخیره می شوند ، اما همچنان بصورت مجزا هم داده هر هسته در دسترس است.

چگونه پایتون را برای تحلیل داده های pencil راه اندازی کنیم؟

ابتدا برای اینکه پایتون دستورات pencil code را شناسایی کند باید در فایل .bashrc ، متغیر محیطی PYTHONPATH را بصورت زیر تعریف کنیم:

```
export PYTHONPATH={PENCIL_HOME}/python
# or
export PYTHONPATH={PENCIL_HOME}/python:{PYTHONPATH}
```

سپس هر بار در پایتون باید دستور " *import pencil as pc* " را اجرا کنیم تا ماژول pencil فراخوانی شود. به این منظور وهمچنین فراخوانی های دیگر بصورت زیر عمل می کنیم:

ابتدا فایل pythonrc. را ایجاد بصورت زیر ایجاد کرده؛

```
#!/usr/bin/python
import numpy as np
import pylab as plt
import pencil as pc
import atexit
#import readline
import rlcompleter
# enables search with Ctrl+r in the history
try:
import readline
except ImportError:
print "Module readline not available."
else:
import rlcompleter
readline.parse_and_bind("tab: complete")
# enables command history
historyPath = os.path.expanduser("~/.pyhistory")
def save_history(historyPath=historyPath):
import readline
```

```
readline.write_history_file(historyPath)
if os.path.exists(historyPath):
readline.read_history_file(historyPath)
atexit.register(save_history)
del os, atexit, readline, rlcompleter, save_history, historyPath
plt.ion()
```

و در ادامه فایل .pythonhistory را ایجاد می کنیم و سپس در .bashrc اضافه می کنیم:

```
export PYTHONSTARTUP=~/.pythonrc
```

چطور در پایتون فایل را باز کنیم و ببینیم؟

داده ها در فایل های var ذخیره می شوند و برای خواندن آن ها در پایتون بصورت زیر عمل می کنیم:

```
var = pc.read.var( varfile='filename' , dir ='data directory' , proc=-1)
```

اسم فایل می تواند هرکدام از var ها مثل VAR10 یا var.dat باشد.
proc، درواقع مشخص کننده شماره هسته ای هست که می خواهیم داده ها را از آن بخوانیم که اگر 1 - باشد، داده های تمام هسته ها تجمیع می شوند (درحالتی که io_collect نمیباشد) ، در غیر اینصورت بسته به تعداد هسته ها می تواند مقادیر ۰، ۱ ،.... را بپذیرد.
حال برای استخراج داده های var تعریف شده هم کافیست از **Tab+ .** استفاده کنیم تا ببینیم چه داده هایی را شامل می شود. مثلا برای چگالی:

```
Rho = var.rho
```

خود-تست اولیه را انجام ندهیم؟

در دفعه اول اجرای کد بهتر است این مورد را فعال بگذاریم تا ایرادهای احتمالی را شناسایی کند اما در دفعات بعدی اجرای کد به جهت سرعت بیشتر در اجرای کد میتوان در run.in ، lpencil_check= F قرار دهیم.

شرایط مرزی را چطور تعیین کنیم؟

در هر یک از فایل های start.in یا run.in می توانیم شرایط مرزی را بسته به فیزیک مسئله تعیین کنیم، برای مثال در راستای محور z می توان تعریف کرد:

bcz= 's' ,'s' ,'s' ,'a' ,'a'

که s مربوط حالت متقارن و a مربوط به حالت پادمتقارن است.
ترتیب قرارگیری کمیت ها در تعیین شرط مرزی در حالت کلی(3d) به شرح زیر است :

$$u_x, u_y, u_z, ln\rho, \frac{s}{c_p}, A_x, A_y, A_z, lnc$$

در pencil ، بجای میدان مغناطیسی B ، با میدان برداری A سروکار داریم، چطور از A به B برسیم؟

برای این کار کافیست از اپراتور کِرل در کتابخانه math خود pencil استفاده کنیم:

pc.math.derivatives.curl( )

بعد از هربار تغییر فایل Makefile.local میتوان کد را بلافاصله اجرا کرد؟

خیر، باید مجددا فیزیک مسئله ساخته شود چراکه از اساس فیزیک مسئله در Makefile تعیین می شود، لذا با تغییر در آن باید مجدد pc_build را انجام داد.
توجه کنید که بعد از تغییر در فایل run و start نیازی به pc_build نیست.

پس از اتمام یک run ، اگر مجدد pc_run اجرا شود، شبیه سازی از اول شروع می شود؟

خیر، از ادامه اجرای قبلی (از ادامه آخرین تایم استپ و زمان)، شروع می شود.