# Informatika pro moderní fyziky (1) základy automatizace; jednoduché zpracování a vizualizace dat

František HAVLŮJ

*e-mail: haf@ujv.cz*

ÚJV Řež
oddělení Reaktorové fyziky a podpory palivového cyklu

semestr 2012/2013
4. prosince 2012

# Obsah

# K čemu je počítač?

- počítače udělají cokoliv, pokud na to existuje postup
- pokud na něco existuje postup, není na to potřeba člověk
- existuje-li proces, existuje také algoritmus
- kdo má algoritmus, může napsat program

# Proč se zabývat automatizací?

- mechanická práce je otravná
- program neudělá (náhodnou) chybu
- skript trvá stejně dlouho pro libovolný objem dat
- pokud je potřeba něco pozměnit nebo jen zpracování zopakovat, je ruční práce vepsí

# Zadání

## # 1

Na konci provozní směny je potřeba vyhodnotit signály ze čtyř detektorů a vykreslit je do grafu (signál v závislosti na čase). Data dostáváte v jednoduchém textovém souboru (dva sloupce, spousta řádků). Je potřeba vykreslit do jednoho grafu všechny čtyři detektory. Potíž je, že taková data přicházejí každý den - tento úkol je tedy potřeba řešit opakovaně.

## S hvězdičkou

Počet detektorů je proměnný (1 až 9).

# Zadání

## # 2

Při přípravě základního kritického experimentu je pomocí MCNP potřeba najít kritickou polohu regulační tyče R2. Jak se tato poloha změní při změně polohy tyče R1?

# Obsah

# Klasické řešení (MS Excel)

- proveďte; jaké všechny kroky je potřeba udělat?
- na který z provedených kroků byl potřeba člověk – co z toho by nemohl stejně dobře udělat počítač sám?
- jaké jsou výhody a nevýhody ručního řešení?
- jak by mělo takové automatické řešení fungovat?

# Komponenty pro automatizaci

## Funkční části

výkonné programy (např. kreslení grafů, generování tabulek/reportů, spouštění výpočtů) – předpokladem je možnost spouštět program v neinteraktivním (dávkovém) režimu

## Jak to slepit dohromady

dávkobý soubor (BAT) nebo skript – je nutno vždy vhodně volit použité prostředky ve vztahu k jednoduchosti, požadavkům na funkce, přenositelnosti

# Jak postupovat s automatickým řešením?

1. vykreslit graf s jedním detektorem

# Jak postupovat s automatickým řešením?

1. vykreslit graf s jedním detektorem
2. se všemi detektory

# Jak postupovat s automatickým řešením?

1. vykreslit graf s jedním detektorem
2. se všemi detektory
3. z příkazové řádky

# Jak postupovat s automatickým řešením?

1. vykreslit graf s jedním detektorem
2. se všemi detektory
3. z příkazové řádky
4. z batch souboru

# Jak postupovat s automatickým řešením?

1. vykreslit graf s jedním detektorem
2. se všemi detektory
3. z příkazové řádky
4. z batch souboru
5. se jménem adresáře jako parametrem

# Gnuplot

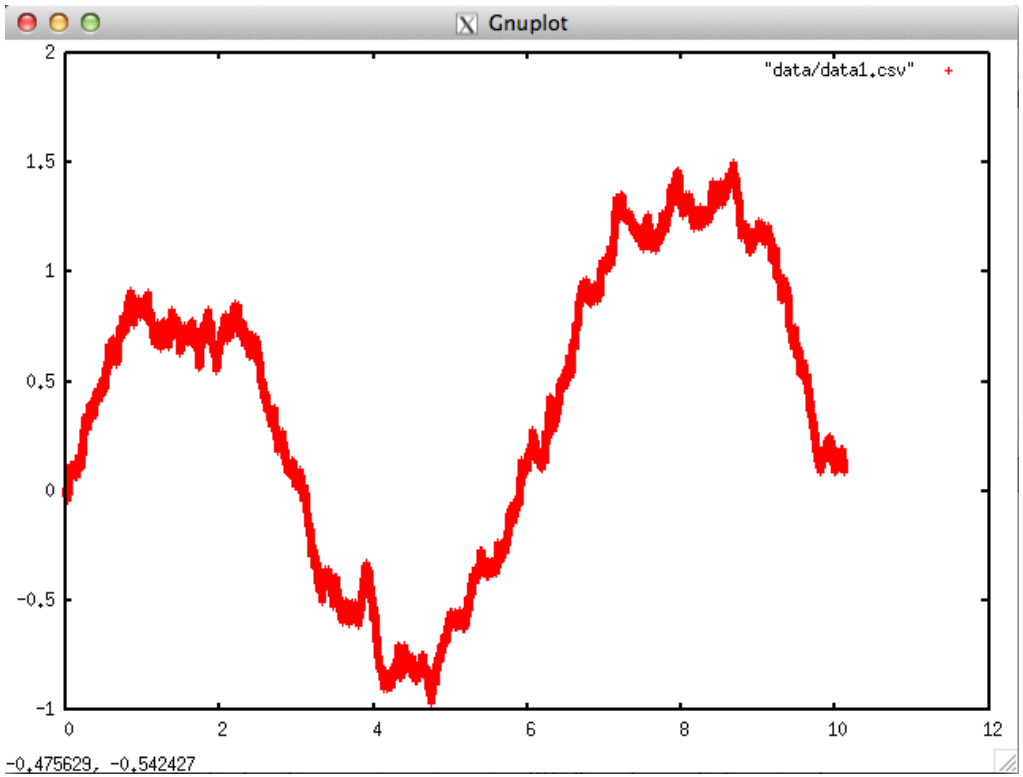
- interaktivní i dávkový režim – ideální pro automatizaci
- slušně konfigurovatelné 2D i 3D grafy
- i bez nastavení funguje velmi přijatelně
- široká paleta výstupních formátů

# Vykreslení jednoho grafu v gnuplotu

```
gnuplot> plot "data1.csv"
```

# Vykreslení jednoho grafu v gnuplotu

```
gnuplot> plot "data1.csv"
```

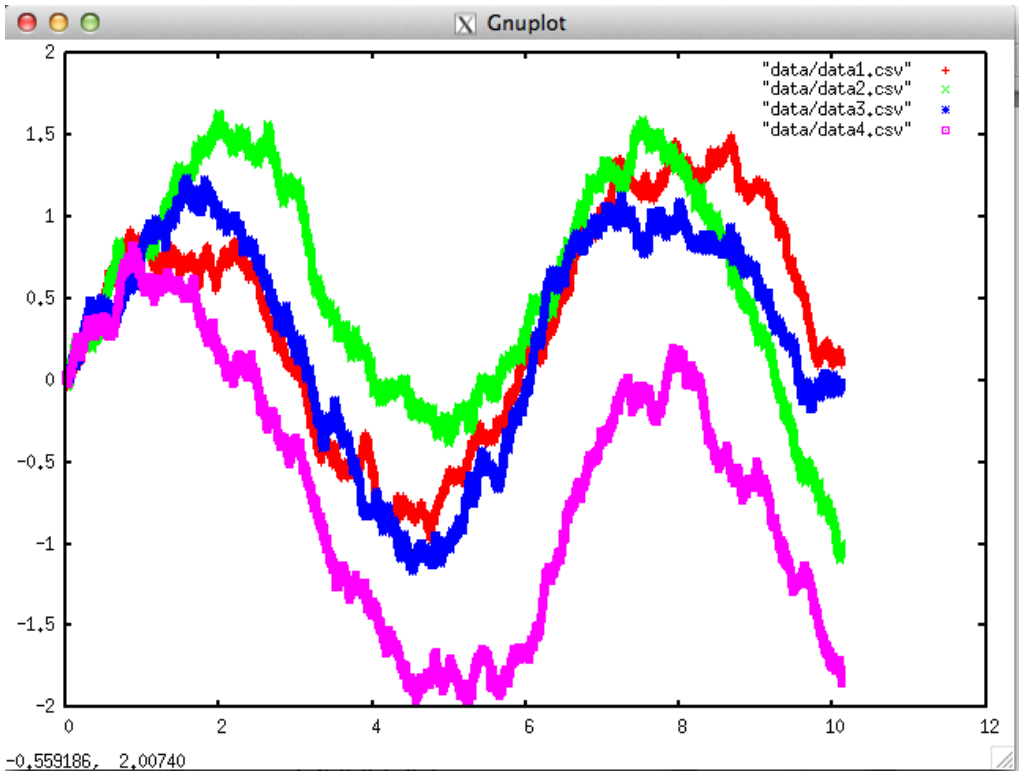


# Vykreslení všech grafů v gnuplotu

```
gnuplot> plot "data1.csv", "data2.csv",
              "data3.csv", "data4.csv"
```

# Vykreslení všech grafů v gnuplotu

```
gnuplot> plot "data1.csv", "data2.csv",
              "data3.csv", "data4.csv"
```

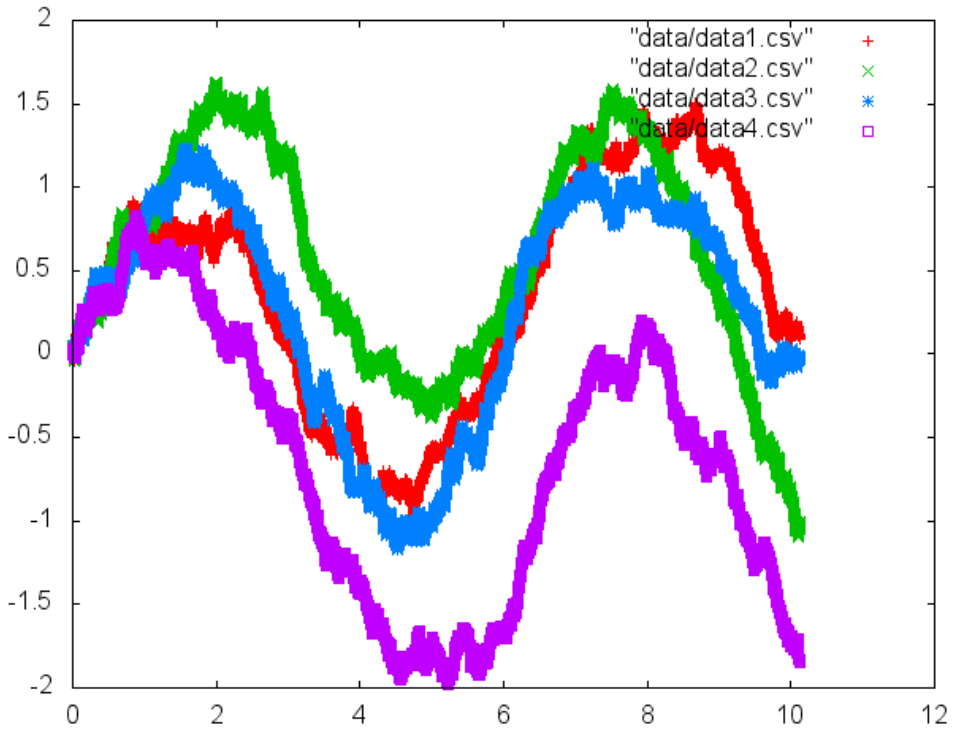


# Dávkové použití

```
set terminal png
set output "plot4.png"
plot "data/data1.csv", "data/data2.csv", \
     "data/data3.csv", "data/data4.csv"
```

# Dávkové použití

```
set terminal png
set output "plot4.png"
plot "data/data1.csv", "data/data2.csv", \
     "data/data3.csv", "data/data4.csv"
```



# BAT soubor

Je pracné pokaždé vypisovat parametry na příkazovou řádku. .BAT soubory ve Windows fungují jednoduše, prostě se do nich dá psát jako do terminálu a připravit si tak jednodušší skript.

```
gnuplot plot.gp
```

# BAT soubor s parametrem

Co takhle adresář pro každý den? Nemá smysl pokaždé ručně kopírovat vstup pro gnuplot a tak dále...
Stačí vědět, že BAT soubor může mít na příkazové řádce parametry. První parametr je uložen do proměnné `%1` a to se nám bude hodit.

```
cd %1
gnuplot ../plot.gp
cd ..
```

# BAT soubor s parametrem - vylepšení

Pokud si budeme chtít prohlédnout grafy, bude nutné vždy vlézt do adresáře a otevřít `plot.png`. Jde to ovšem vylepšit pomocí jednoduchého triku:

```
cd %1
gnuplot ../plot.gp
copy plot.png ../%1.png
cd ..
```

# Jak moc jsme si pomohli?

# Stačí nám to na řešení problému č. 2?

Zatím nevíme, jak:

- vygenerovat vstupní soubory pro MCNP
- spustit hromadu MCNP výpočtů
- vytahat výsledky z MCNP výstupního souboru

Budeme potřebovat nějaký těžší kalibr.

# Obsah

# “klasické” programování – Pascal, C++

# Interpretované jazyky / skripty

# Ukázka Ruby (1)

Každý programátor tím začíná ...

```
puts "Hello world!"
```

# Ukázka Ruby (1)

Každý programátor tím začíná ...

```
puts "Hello world!"
```

```
Hello world!
```

# Ukázka Ruby (2)

## Proměnné, `print` vs. `puts`, aritmetika

```
a = 4
b = 5
print "4 + 5 = "
puts a + b
```

# Ukázka Ruby (2)

## Proměnné, `print` vs. `puts`, aritmetika

```
a = 4
b = 5
print "4 + 5 = "
puts a + b
```

```
4 + 5 = 9
```

# Ukázka Ruby (3)

## In-line výrazy v řetězcích

```
a = 4
b = 5
puts "#{a} + #{b} = #{a+b}"
```

# Ukázka Ruby (3)

## In-line výrazy v řetězcích

```
a = 4
b = 5
puts "#{a} + #{b} = #{a+b}"
```

```
4 + 5 = 9
```

# Ukázka Ruby (4)

## Rozsahy a cykly

```
(1..5).each do |i|
  puts "#{i} * #{i} = #{i * i}"
end
```

# Ukázka Ruby (4)

## Rozsahy a cykly

```
(1..5).each do |i|
  puts "#{i} * #{i} = #{i * i}"
end
```

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
```

# Ukázka Ruby (4)

Pětkrát nic umořilo osla (opakování, ne cyklus)

```
5.times do
  puts "nic"
end
```

# Ukázka Ruby (4)

Pětkrát nic umořilo osla (opakování, ne cyklus)

```
5.times do
  puts "nic"
end
```

```
nic
nic
nic
nic
nic
```

# Určení poloh tyčí

```
c --------------------------------
c polohy tyci (z-plochy)
c --------------------------------
c
67 pz 47.6000    $ dolni hranice absoberu r1
68 pz 40.4980    $ dolni hranice hlavice r1
69 pz 44.8000    $ dolni hranice absoberu r2
70 pz 37.6980    $ dolni hranice hlavice r2
```

# A to je vše, přátelé!