

Hexagon LIVE Software User Conference CZ&SK

Využití Pythonu v prostředí Adams

Škoda Auto a.s.

Jan Blokeš

jan.blokes@skoda-auto.cz

Zaječí 19.6.2024



SKODA

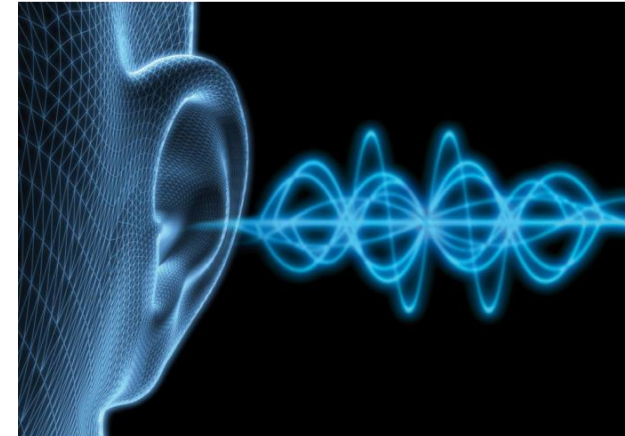
OBSAH

1. Úvod
2. Python: základní info
3. Ukázka možností spuštění úloh v Adams/Car
4. Python: Postprocessing
5. Python: Optimalizace
6. Závěr

ODDĚLENÍ

EIG – Fyzika a vlastnosti celého vozu

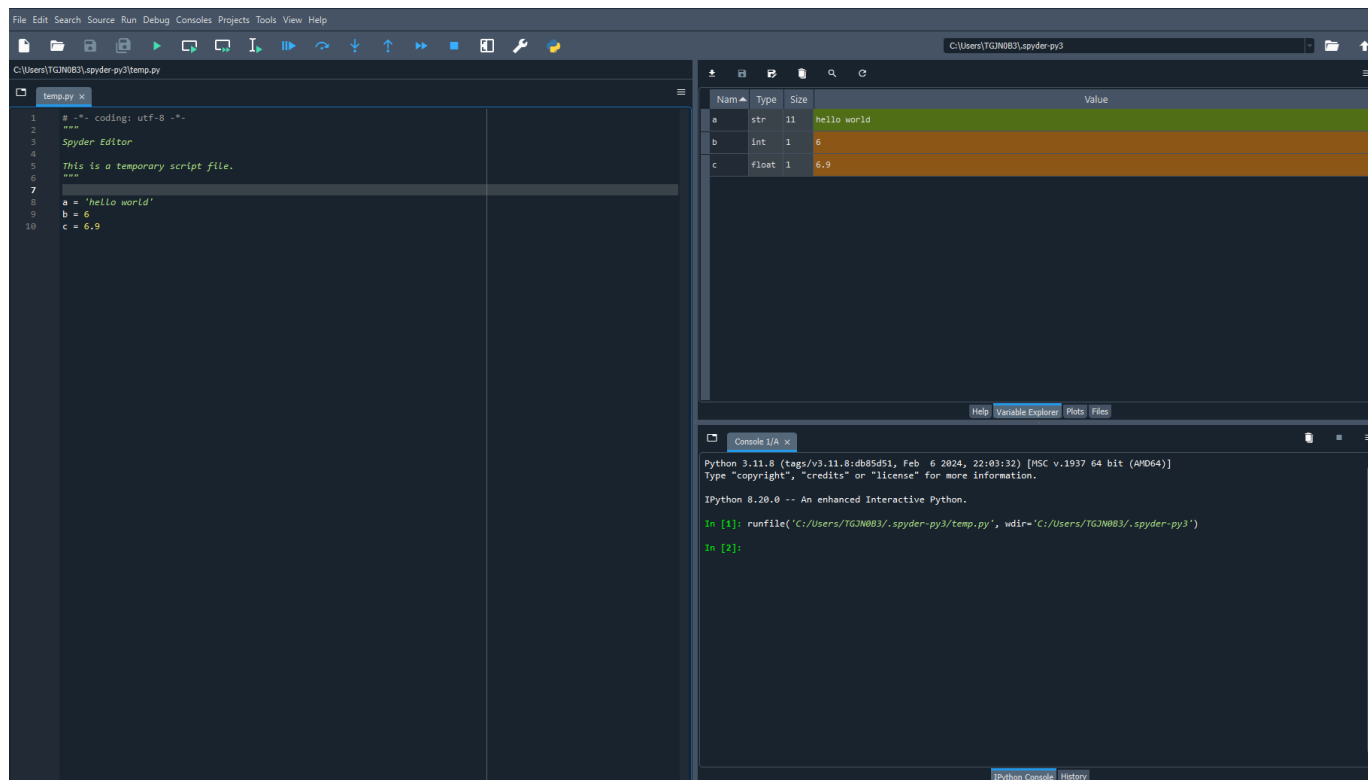
1. Životnost karoserie
 - Jízda po naskenované trati (zkušební polygon VW)
 2. Světla výška při jízdě
 - Jízda po naskenovaných překážkách (obrubník, žel. přejezd, apod.)
 3. Akustika a vibrace
 - Odvalovací hluky
 - Jízdní komfort
- Využití software Adams ve všech zmíněných disciplínách



PYTHON

Základní info

- Programovací jazyk
- Editor
 - Spyder (Linux i Windows)
- Existují jiné editory
 - Visual Studio
 - IDLE
 - VIM
 - PyCharm
 - Online compilers
 -

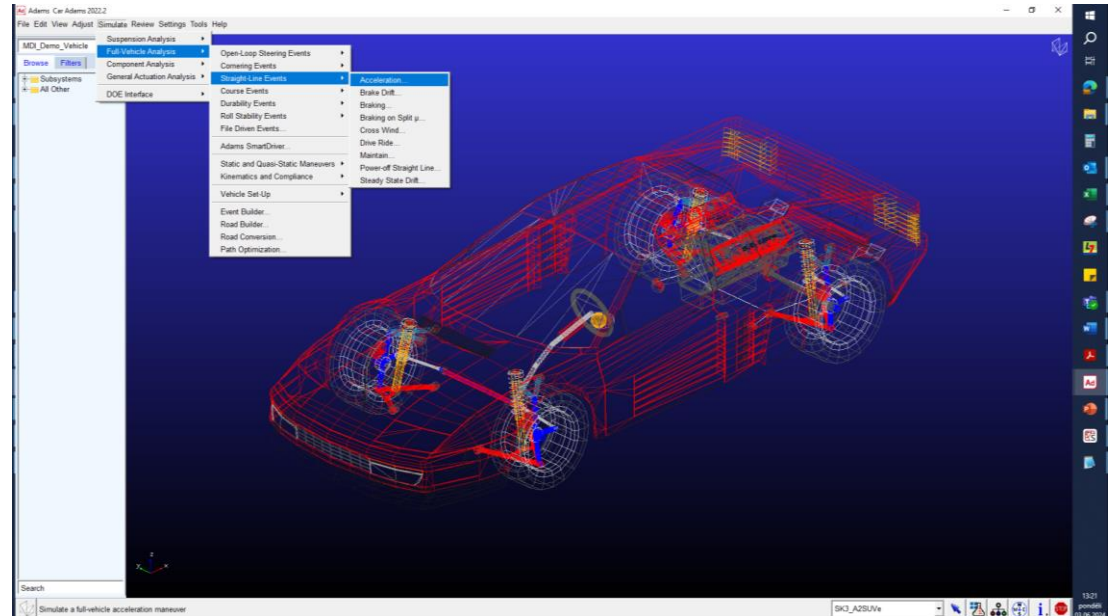


PYTHON

Ukázka – spuštění úlohy v Adams/Car

1. GUI

- Vhodné pro odladění úloh
- Pro spuštění jednoho manévru je vhodný, ale pro desítky manévrů ne
- Vozidlo MDI_Demo_Vehicle.asy z databáze acar_shared
- Simulate – Full Vehicle Analysis – Acceleration

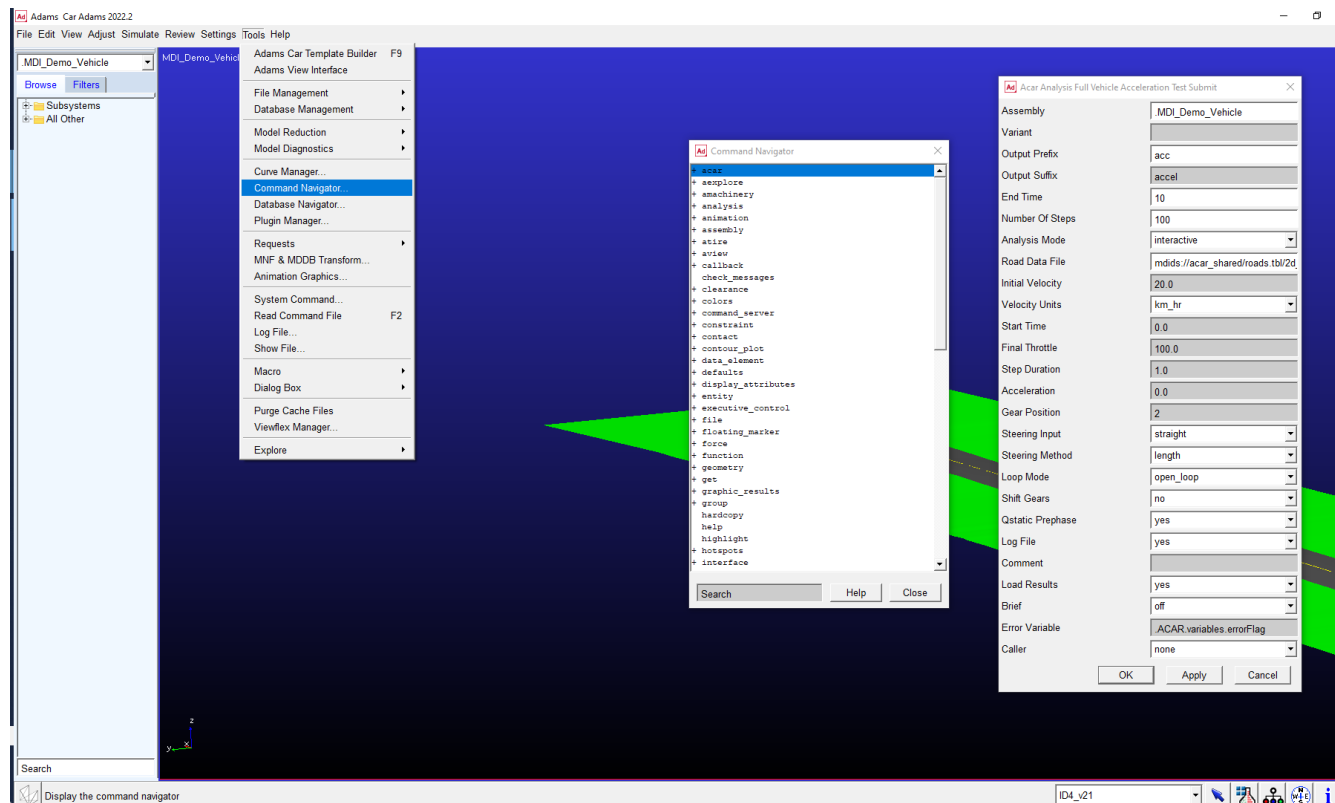


PYTHON

Ukázka – spuštění úlohy v Adams/Car

2. Import cmd

- Úloha se dá spustit přes command navigator: čistší command – bez grafických zbytečností
- Stačí kopírovat kód z CMD command window

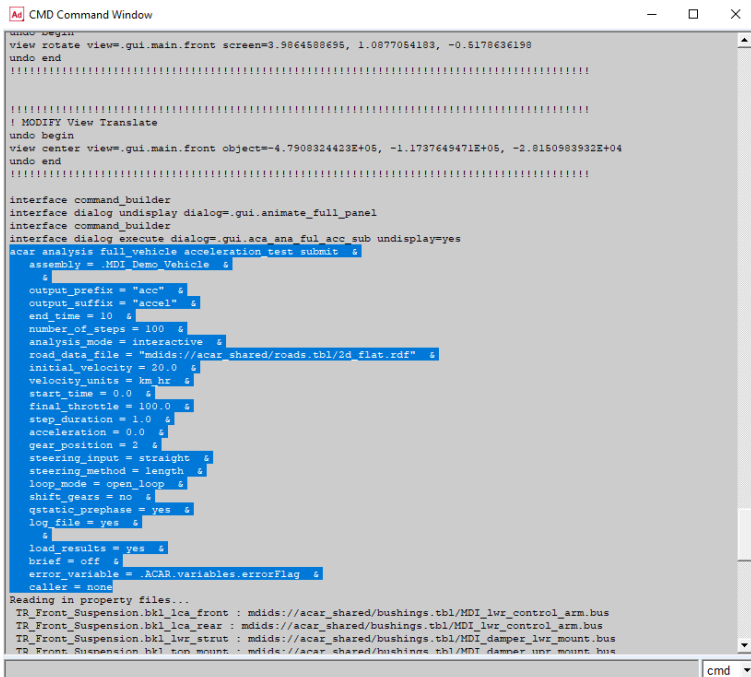


PYTHON

Ukázka – spuštění úlohy v Adams/Car

2. Import cmd

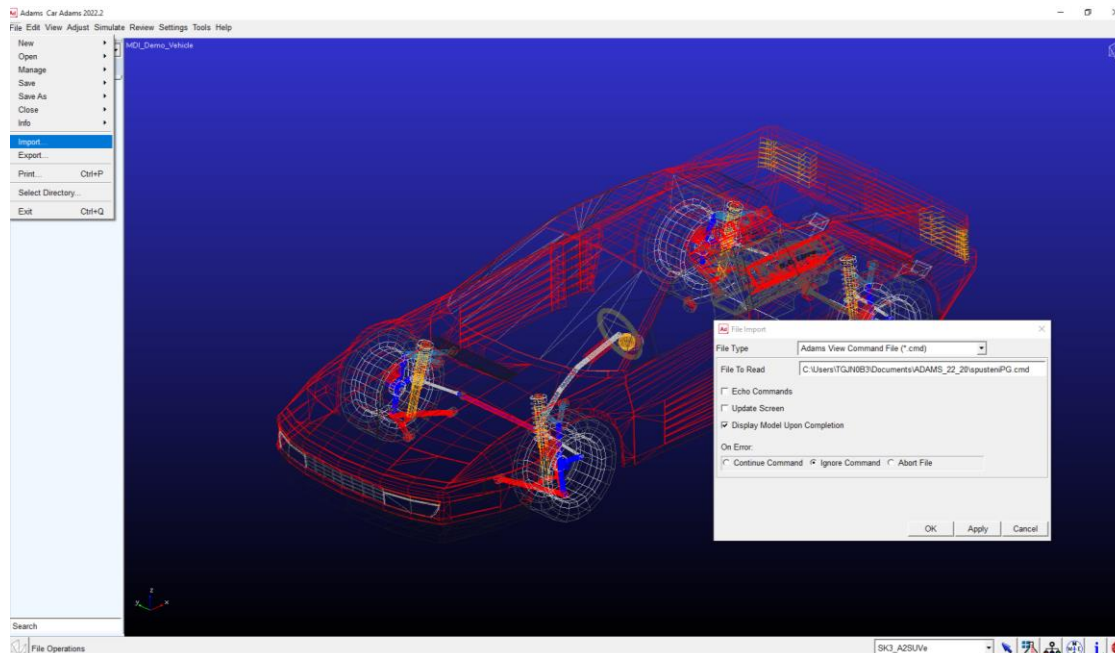
- Kód kopírovat buď z CMD Command Window nebo z acar.log
- Uložit jako soubor s příponou *.cmd a pomocí F2 (nebo File->Import->*.cmd) vložit
- Takto se dá vložit více manévrů za sebou - > počítají se automaticky
- Pro jinou assembly se jen přepíše název sestavy – v textovém editoru nebo v programovacím jazyce



```
view rotate view=.gui.main.front screen=3.9864688696, 1.0877064183, -0.6178636198
undo end

! MODIFY View Translate
undo begin
view center view=.gui.main.front object=-4.7908324423E+05, -1.1737649471E+05, -2.8150983932E+04
undo end

interface command_builder
interface dialog undisplay dialog=.gui.animate_full_panel
interface command_builder
interface dialog execute dialog=.gui.aca ana ful_acc sub undisplay=yes
aca analysis fullvehicle acceleration_test_submit &
assembly = .MDI Demo Vehicle &
&
output_prefix = "acc" &
output_suffix = "accel" &
end_time = 10 &
number_of_steps = 100 &
analysis_mode = interactive &
road_data_file = "mdids://acar_shared/roads.tbl/2d_flat.rdf" &
initial_velocity = 20.0 &
velocity_units = km/hr &
start_time = 0.0 &
final_throttle = 100.0 &
step_duration = 1.0 &
acceleration = 0.0 &
gear_position = 2 &
steering_input = straight &
steering_method = length &
loop_mode = open_loop &
shift_gears = no &
qustatic_prephase = yes &
log_file = yes &
&
load_results = yes &
brief = off &
error_variable = .ACAB.variables.errorFlag &
caller = none
Reading in property files...
TR_Front_Suspension.bkl_lca_front : mdids://acar_shared/bushings.tbl/MDI_lvr_control_arm.bus
TR_Front_Suspension.bkl_lca_rear : mdids://acar_shared/bushings.tbl/MDI_lvr_control_arm.bus
TR_Front_Suspension.bkl_lvr_strut : mdids://acar_shared/bushings.tbl/MDI_damper_lvr_mount.bus
TR_Front_Suspension.bkl_ton_mount : mdids://acar_shared/bushings.tbl/MDI_damper_lvr_mount.bus
```



PYTHON

Ukázka – spuštění úlohy v Adams/Car

3. Spuštění Adamsu z příkazového řádku

- V Adams car je výhoda toho, že se dá spustit assembly se subsystémem, který GUI nenačte
- *.bat soubor

```
del aview.cmd  
  
cd C:\Users\TGJN0B3\Documents\ADAMS_22_20\  
adams2022_2 acar ru-so acc_accel.acf  
  
pause
```

- Spustí se buď *.acf nebo *.cmd
- Pro full_vehicle musí být soubory *.acf, *.adm a *.xml v jedné složce!

- Výhoda: nemusí se klikat!

```
9 import subprocess  
10  
11 batch_script = 'adams.bat'  
12  
13  
14 try:  
15     process = subprocess.Popen(batch_script, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)  
16     out, err = process.communicate()  
17  
18     if process.returncode == 0:  
19         print("Script executed successfully")  
20         print("Output:", out.decode())  
21     else:  
22         print("Script encountered an error")  
23         print("Error:", err.decode())  
24 except Exception as e:  
25     print("Error:", e)
```

```
del aview.cmd  
  
cd C:\Users\TGJN0B3\Documents\ADAMS_22_20\springOpt1\  
adams2022_2 -c acar ru-st b cmd.cmd  
  
pause
```

PYTHON

Postprocessing

- Příklad: analýza sil na bushingu předního horního ramena při jízdě
- Full vehicle analysis – Smart Driver – Country road

Full-Vehicle Analysis: SmartDriver

Vehicle Assembly: MDI_Demo_Vehicle

Assembly Variant: default

Output Prefix: country

Solver Settings File: [icon]

End Time / Duration: 20.0 [Sec]

Number of Steps: 4000.0 [-]

Analysis Mode: interactive

Road Data File: [icon] mdids://acar_shared/roads.tbl/2d_flat.rdf

Velocity: 16.0 km/hr

Gear Position: [lightbulb icon] 2

☒ Quasi-Static Straight or Skidpad Set-Up

Course Type: ☐ Driver Course ☒ 3D Spline, CRG or RGR Road

3D Road Data File: [icon] mdids://acar_shared/roads.tbl/country_road_reposit

☐ Auto-Calculate Initial Velocity

Shift Cycle Time: 0.5

Smart Driver Task: User Defined ☐ Power-based Shift Control

Maximum Acceleration [%]

Driving: 0 50 100

Braking: 0 50 100

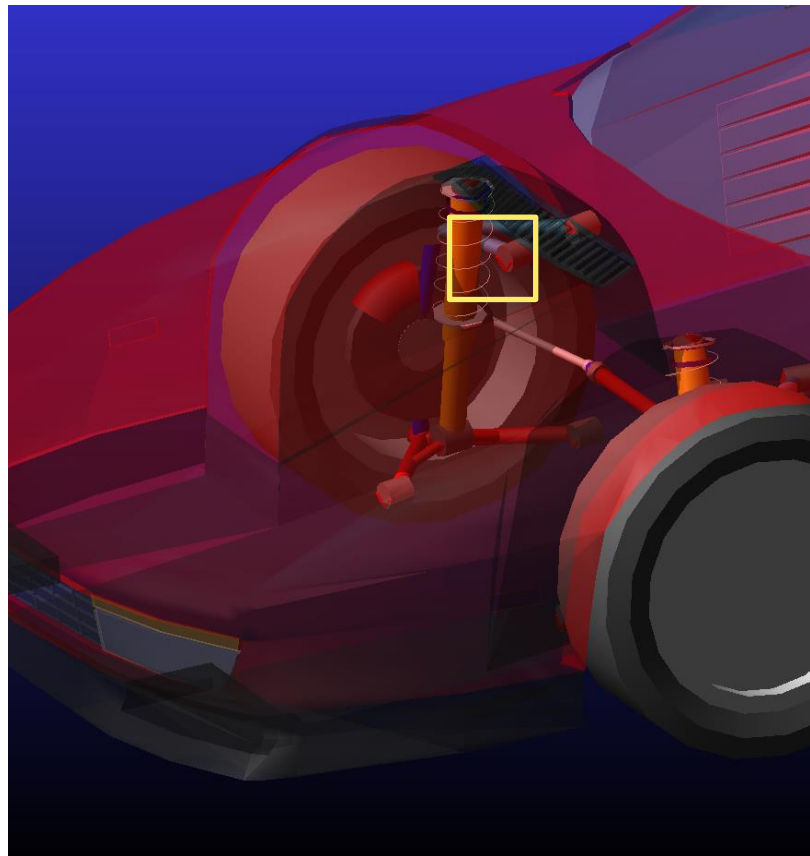
Left Right

Cornering: 50 50

☒ Create Event Log File ☐ Add Vehicle Dynamics Requests

☐ Compute Characteristic Values

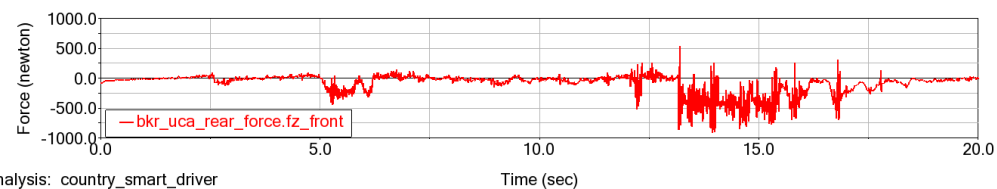
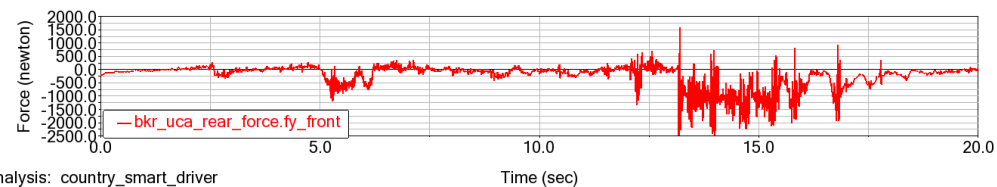
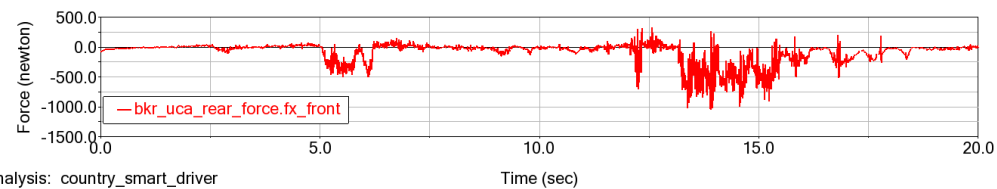
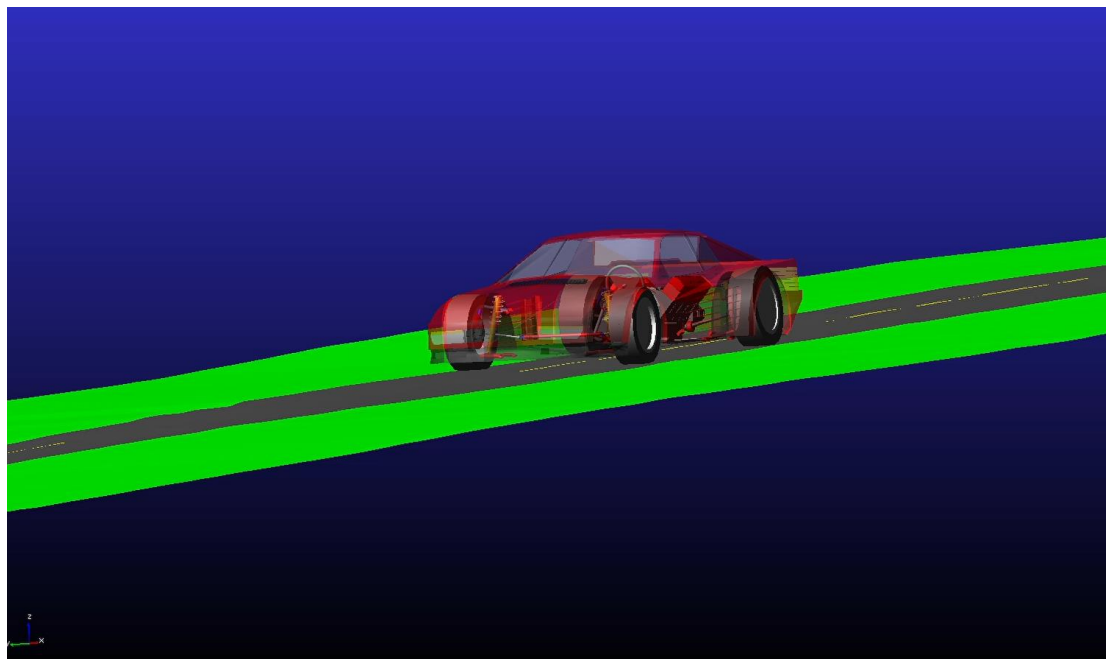
OK Apply Cancel



PYTHON

Postprocessing

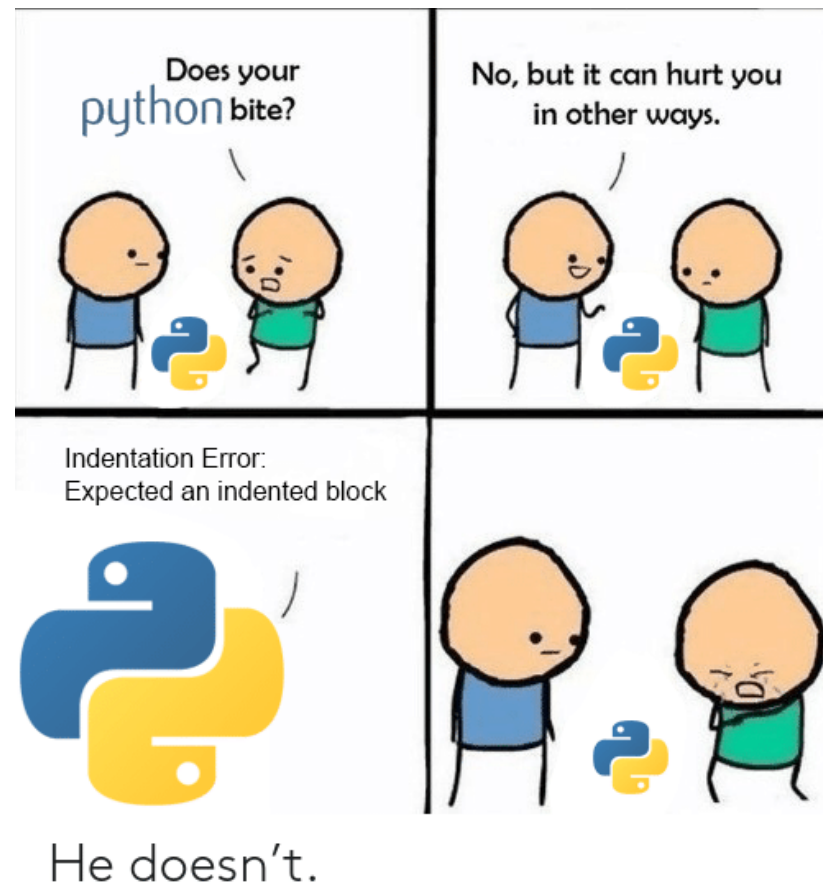
– Příklad: analýza sil na bushingu předního horního ramena



PYTHON

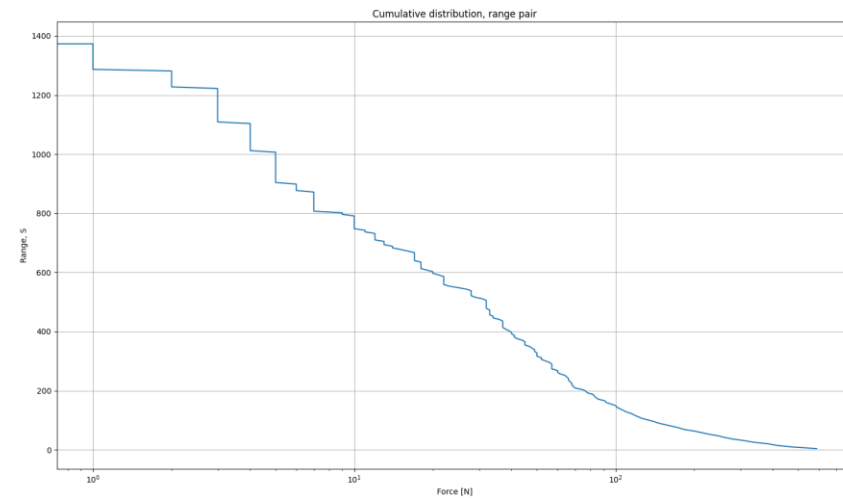
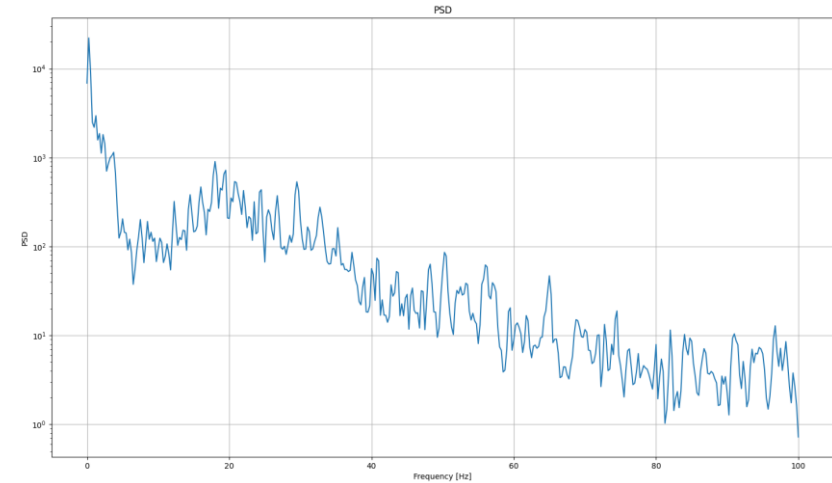
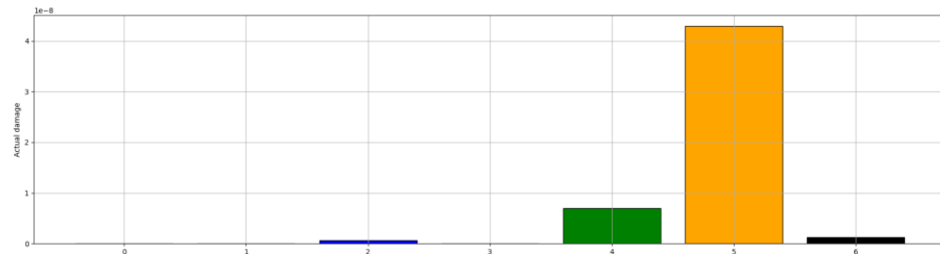
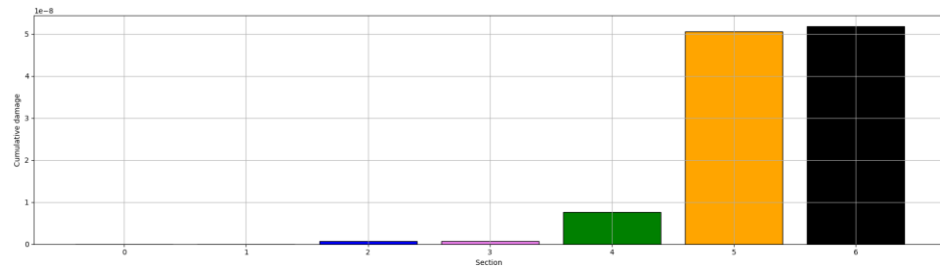
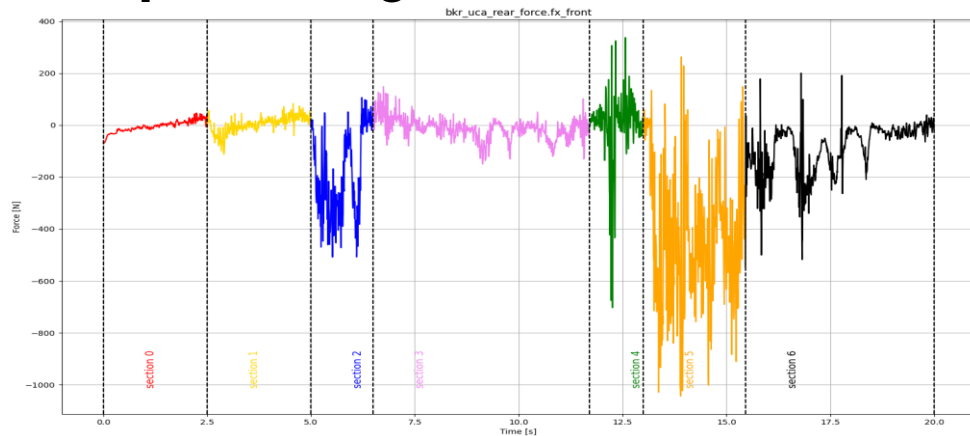
Postprocessing

- Úprava signálů + vyhodnocení (random signal)
 - Filtry, průměry, rms, min, max,...
 - Hladiny zatížení
 - Pseudo damage
 - Časová -> Frekvenční doména - FFT, PSD, spektrogram
 - ...
- Vhodné knihovny – volně stažitelné
 - Numpy
 - Scipy
 - Fatpack
 - pyLife
 - wafo



PYTHON

Postprocessing

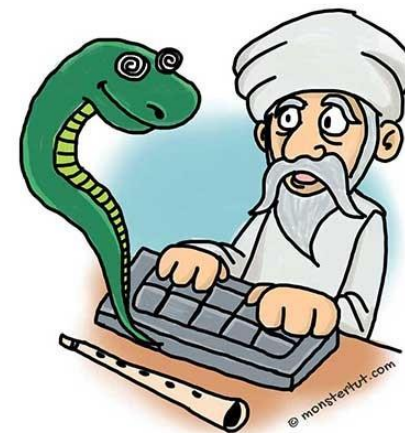


PYTHON

Postprocessing

- Automatické uložení obrázků
 - *.png, *.jpg,...
- Tvorba reportů – ze simulací nebo měření
 - Word
 - Latex
- Čtení, úprava, zápis jiných formátů
 - *.txt
 - *.xml
 - Excel – knihovna xlrd, openpyxl
 - Měřicí ústředny
 - NI - *.tdms
 - DEWEsoft – *.d7d,*.dxd,*.dxz - knihovna dwdatareader
 - Výhoda – dá se vyhodnocovat ihned po naměření dat
 - Jiné komerční software –např. Tecware

This is the modern
way to control
a python™



L^AT_EX



PYTHON

Optimalizace

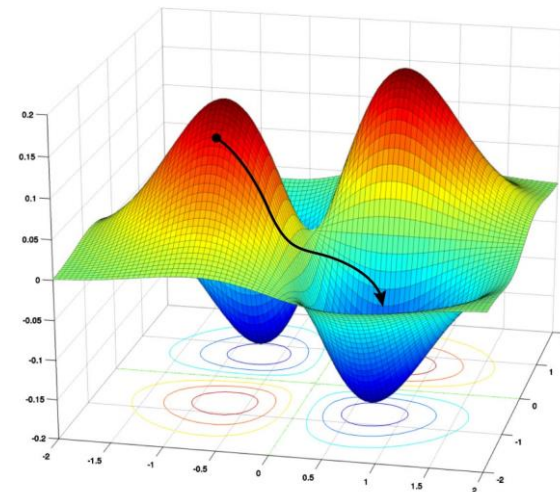
- Struktura optimalizace
 - Proměnné (cokoliv – parameter pneumatiky, hardpoint, kontaktní tuhost,...)
 - Řešič (scipy.optimize)
 - Cílová funkce (vlastní přínos – jiná pro srovnání měření a sim., jiná pro naladění statiky, apod.)
 - Počáteční podmínky (můžeme použít z Adamsu)
- Vhodná pro srovnání měření a simulace – nastavení různých parametrů
 - Buď citlivostní metodou – zkušenosti
 - Nebo optimalizací
- Ukázka na příkladu: nastavení předpětí pružiny podle měření



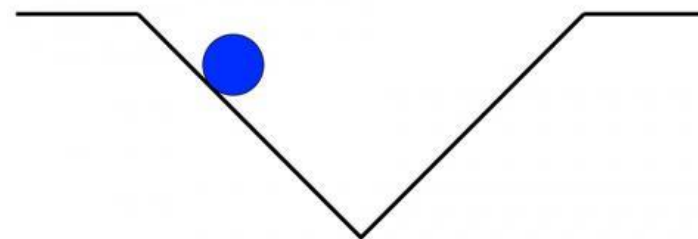
PYTHON

Optimalizace

- Stejná vzorkovací frekvence pro test i simulaci – jinak interpolace!
- Struktura skriptu
 1. Vytvoření subsystemů, kde jsou proměnné
 2. Vytvoření souboru *.cmd
 - A. Assembly
 - B. Nastavení analýzy (výběr analýzy (manévru), time step, end_time,...)
 - C. Uložení cmd
 3. Spuštění úlohy příkazovým řádkem
 4. Přečtení výsledků (*.req nebo *.res)
 5. Vyhodnocení cílové funkce (např. porovnání simulace a měření)
 6. Změna parametrů a zpět na krok 1



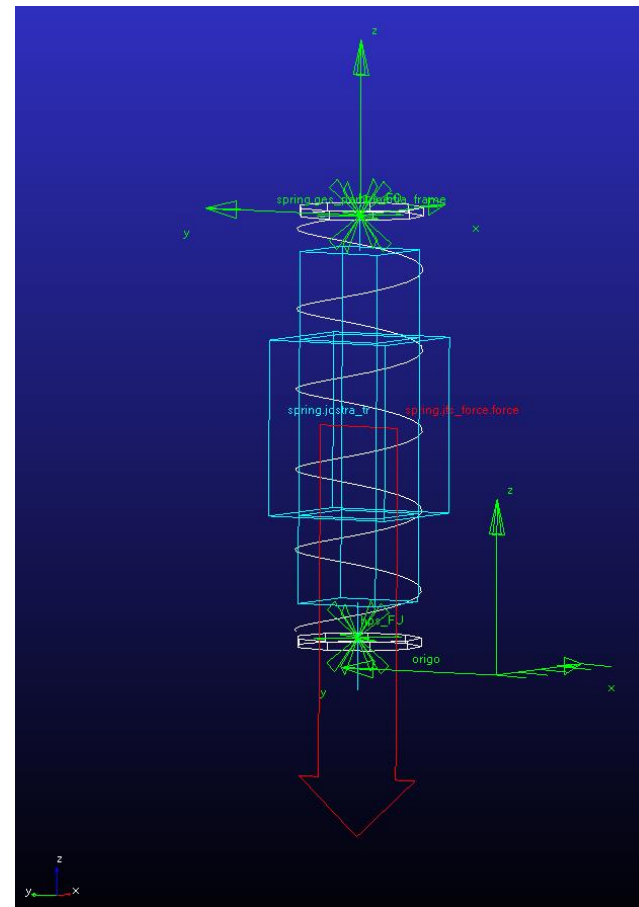
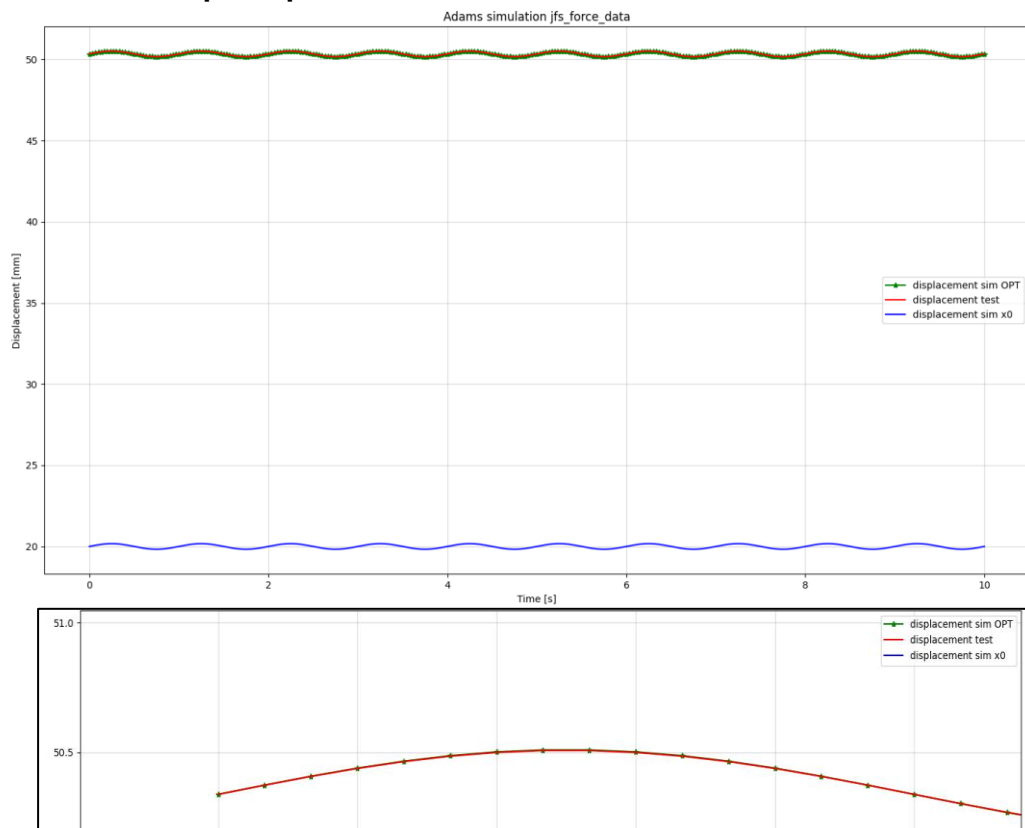
$$\min \frac{1}{2} x^T A x - x^T b$$
$$Cx \geq d$$



PYTHON

Optimalizace

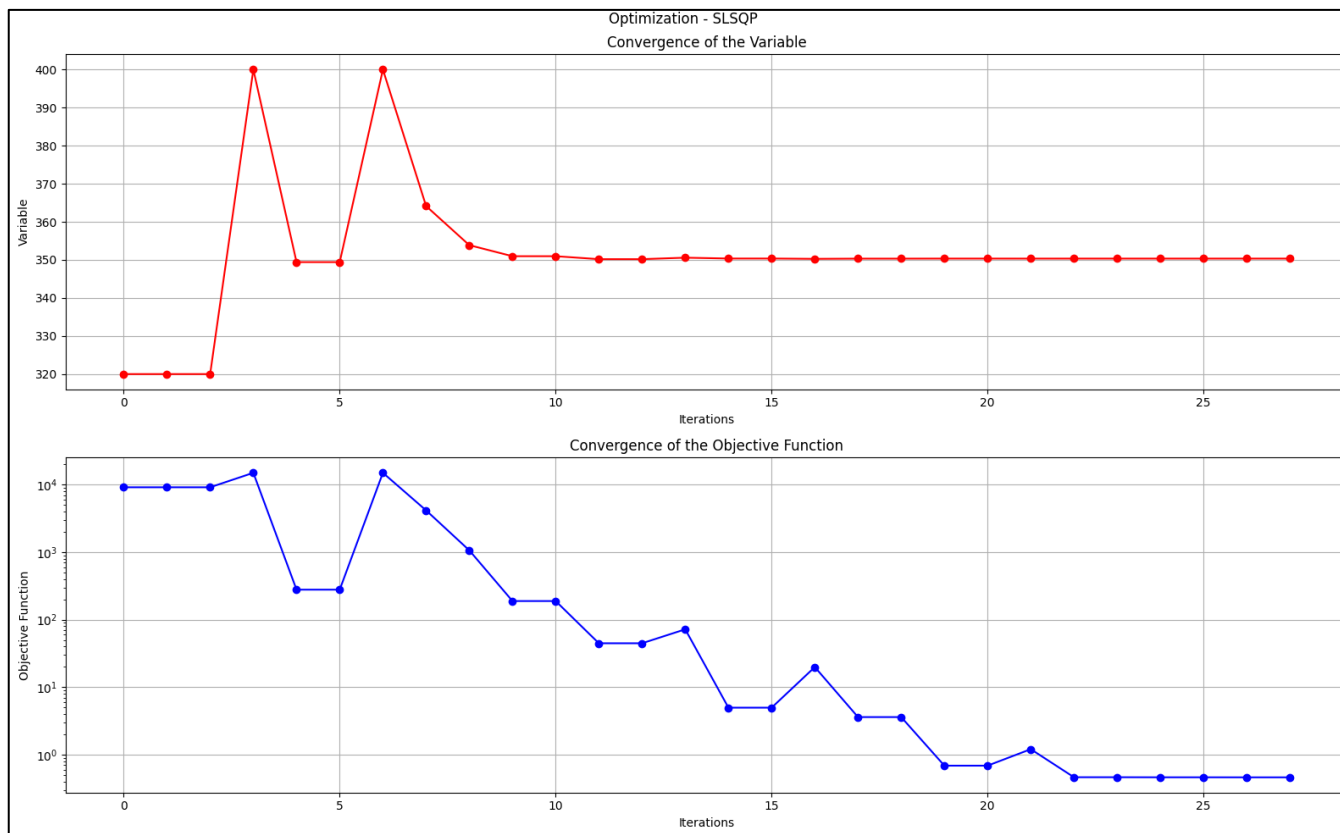
- Cíl: najdi optimální předpětí pružiny pomocí Installed Length
- Počáteční nástřel: 320.00000000
- Hodnota po optimalizaci: 350.33795771



PYTHON

Optimalizace

- Výsledky proměnné v jednotlivých iteracích
- Výsledky cílové funkce v jednotlivých iteracích



PYTHON

Optimalizace

- DOE – Adams Insight
 - Nevím, jak dostanu výsledky z měření do cílové funkce
- Python
 - Knihovna Scipy, funkce optimize
 - Funguje pro více proměnných
 - Širší škála řešičů
 - Nelder – Mead - Simplex
 - BFGS - gradient
 - Newton – CG – conjugate gradient
 - ...
- Použití pro všechny typy úloh – nezávislé na řešiči
 - Adams View
 - Adams/Car
 - Analytické výpočty

ZÁVĚR

- Využití Pythonu nám umožňuje:
 - Spouštění více manévruů najednou
 - Automatické vyhodnocení různých zátěžných stavů
 - Automatickou tvorbu zpráv ze simulací
 - Optimalizovat úlohy
 - Najít ideální varianty
 - Naladění simulace a měření
 - Propojení s dalším software: multifyzikální úlohy
- Čemu se chceme věnovat do budoucna:
 - Tvorbě vlastních pluginů
 - Vytvoření grafického prostředí
 - Zvýšení rychlosti skriptů

Děkuji za pozornost!

ŠKODA



SIMPLY CLEVER