**Model Solver**

**Kratko uputstvo za modelSolver.exe**

modelSolver se nalazi na linku:

<https://www.researchgate.net/publication/373144163_modelSolver_20230816>

**Instalacija i korištenje**

Raspakovati sadržaj 7zip arhive (sa gornjeg linka) u neki folder na Windows sistemu.

Unutar tog foldera se nalazi aplikacija **modelSolver.exe** i **plotSol.py.**

Ukoliko se aplikacija modelSolver starta bez ikakvih parametara dobije se kratko uputstvo za korištenje:

izudindzafic@IDzM1Max Release % **./modelSolver**

Usage: modelSolver modelType domain inputFileName outputFileName [t0] [dT] [t1]

where:

modelType can be: ODE for solving Ordinary Diferential Equations;

DAE for solving Diferential Algegraic Equations

NR required to solve set of nonlinear equations using Newton-Raphson method

WLS for solving WLS (EC) state estimation problems

DIFF for testing differentiation.

domain can be: real for problems defined in real domain (R);

cmplx for problems defined in complex domain (C).

inputFileName: is input file name in xml format.

outputFileName: is output (result) file name in txt format.

t0: is required for ODE and DAE. Ignored by other methods. Represents the starting time of simulaiton.

dT: is required for ODE and DAE. Ignored by other methods. Represents the time increment of simulaiton.

t1: is required for ODE and DAE. Ignored by other methods. Represents the final time of simulaiton.

Napr. Da bi se startao DAE solver za primjer sa klatnom potrebno je otkucati slijedeće:

**izudindzafic@IDzM1Max Release % ./modelSolver DAE real real/PendulumDAE2.xml Res.txt 0 1e-4 2.1**

Model analizirano sistema se unosi u xml formatu. Vidjeti primjere. Ukoliko je potrebno dodatno objašnjenje možemo organizovati kratak sastanak na fakultetu.

**Ispis rezultata dinamičke simulacije**

Za crtanje grafova iz dobivenih dinamičkih simulacija moguće je koristiti priloženu plotSol.py skriptu. Ona je napisana u python-u. Kolega Botić razvija C++ API za crtanje (ko želi ovu varijantu neka se obrati kolegi Botiću).

Da bi skripta radila potrebno je instalirati python3 ukoliko već nije instaliran na sistemu.

Također, skripta zahtjeva i **matplotlib** paket. Ukoliko se nakon izvršenja skripte ne prikaže graf onda je potrebno instalirati paket **matplotlib** kucanjem naredne instrukcije u komandnom promptu:

**pip install matplotlib**

Napr, za naš prethodni primjer kreiran je fajl Res.txt kojeg crtamo na slijedeći način:

**./plotSol Res.txt**

Nakon ovoga bi trebala da se dobije slika slična ovoj ispod. Ukoliko se dobije onda ili nije instaliran python ili nije instaliran paket matplotlib.

A picture containing screenshot, plot

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidence

A picture containing screenshot, plot

Description automatically generated

Teme koje se mogu odabrati korištenjem modelSolver-a

Implementacija primjera (xml modela) za: ODE, DAE u elektrotehnici (preferabilno smart grid).

Testiranje cmplx i real izvoda. Implementacija i verifikacija od jednostavnih do složenih testnih slučajeva.

Implementacija xml primjera za Newton Raphson PF i State Estimation (NE i EC) za power systems (od jednostavnih do složenijih u real i kompleksnoj domeni).

Solver podržava if-then-else instrukcije kao i slijedeće funkcije za real i cmplx domene:

{"abs","acos","asin","atg","cos","exp","sqrt","ln","log","sin","tg", "sqr", "atan2", "sign",

"sinh", "cosh", "tgh", "asnh", "acsh", "atgh", "disc"};

Solver u kompleksnoj domeni dodaje još i funkcije:

"conj", "real", "imag",

**Changes 2023-08-16**

Added support for Implicit Runge-Kutta RK2, RK4, and RK6 methods when solving DAE

Added support for Tranfsfer functions.

Other improvements.