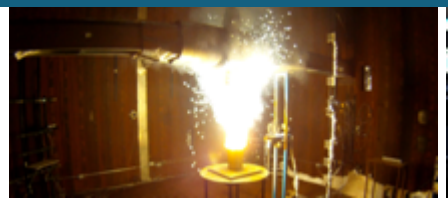
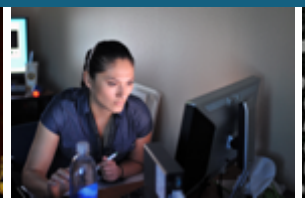
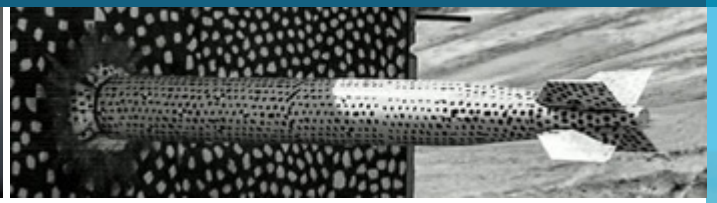
SAND2021-12001C

Sandia National Laboratories

Xyce Python Model Interpreter (Xyce-PyMi) for enabling ML advancements in production circuit simulation software







Presenter: Paul Kuberry*

Team members: Ting Mei*, Eric Keiter*, Biliana Paskaleva*, Pavel Bochev*, Joshua Hanson*#

* Sandia National Laboratories, # University of Illinois Urbana-Champaign

Funding: ASC Advanced Machine Learning Initiative

MMLDT CSET 2021

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The need for Python models

National Laboratories Software Ecosystem (mostly C++)

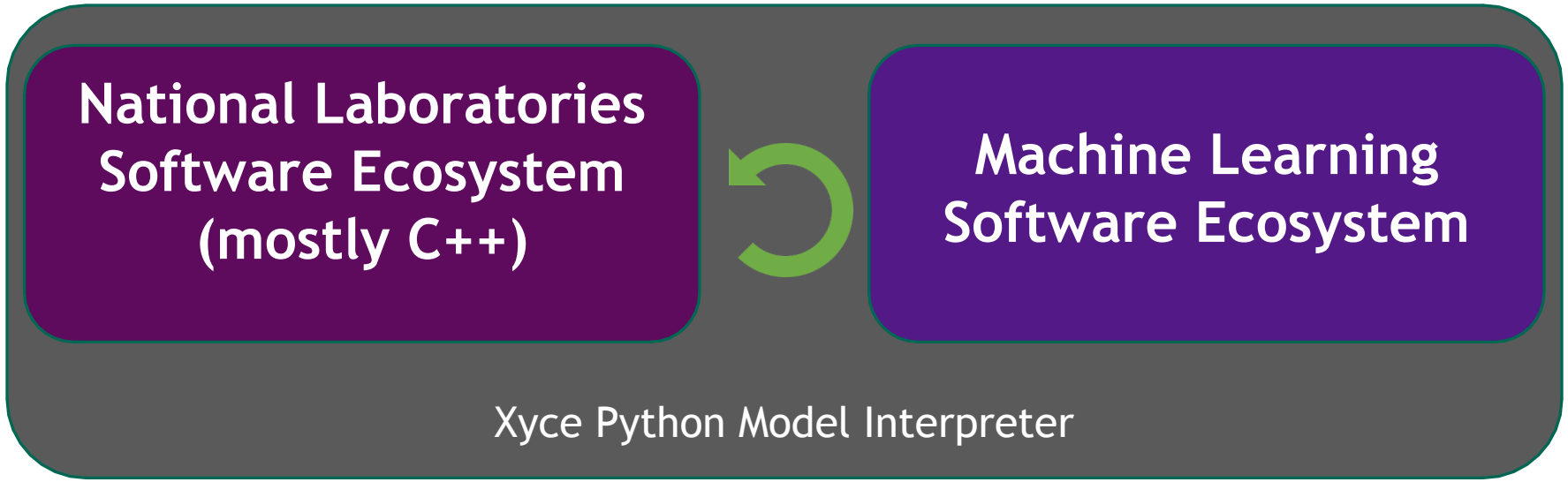
Machine Learning Software Ecosystem

"75% of ML developers and data scientists use Python"
- State of the Developer Nation (Slashdata.co 2020)

https://s3-eu-west-1.amazonaws.com/vm-blog/uploads/2020/04/DE18-SoN-Digital-.pdf

Our goal is to enable ML advancements to impact design phases of electrical systems via data-driven compact device models

What is Xyce-PyMi?



- Leverages General External interface (C++) from Xyce, by Tom Russo at Sandia National Laboratories
- Easily installable and callable capability for executing Python ML models in a production circuit simulation software (Xyce ←→ PyBind11)
- Product of active engagement with compact model development groups and circuit designers
- Provide data-driven compact device modeling approaches (GMLS, splines, deep neural networks) from ongoing research projects at Sandia such as *Data-Driven, Radiation-Aware, Agile Modeling Approach for Rapid Nuclear Deterrence Design Assessment* and *Physics-Informed, Rapid and Automated MachIn learning for compact model Development (PIRAMID)*

How to call and specify device/subcircuit behavior

```
>> Xyce-PyMi example.cir
```

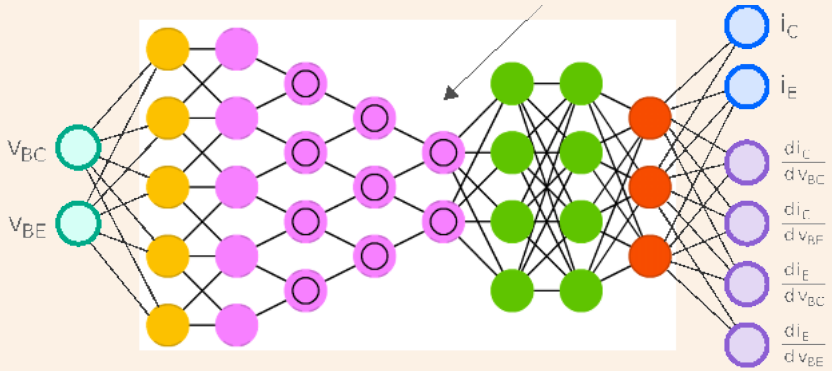
Xyce-PyMi
(Xyce + **Py**thon **M**odel **I**nterpreter)

- Full Xyce functionality + devices / subcircuits / circuits defined in Python
- Python class defines how F, Q, B, dF/dX, and dQ/dX vectors/matrices are populated for Xyce DAE equation:
  - residual = f(x,t) + dq(x,t)/dt - b(t)
- Supports all popular machine learning frameworks such as TensorFlow, Keras, PyTorch, Jax, Numba, Numpy, etc…

```
* example.cir (Example of easy inclusion in a netlist)
YGENEXT devicename terminal1 terminal2
+ SPARAMS={NAME=MODULENAME VALUE=PythonDevice.py}
```

```
# PythonDevice.py
import numpy as np
from BaseDevice import BaseDevice

class Device(BaseDevice):
    def computeXyceVectors(…):
        # definition of device in Python goes here
```



BJT DNN Compact Model

Installation Tutorial

Generalized Moving Least Squares

1. (GMLS Approximate) Constrained Optimization formulation:

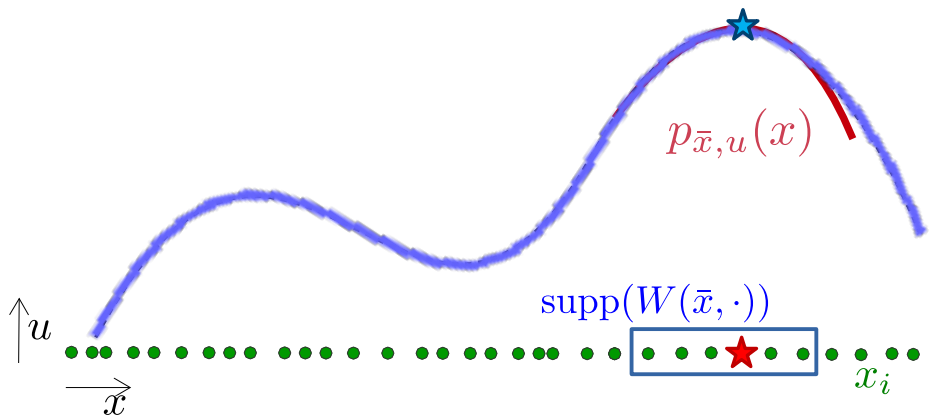
$$\tau^h_{\bar{\mathbf{x}}}(u) := \sum_{i \in I_{\bar{\mathbf{x}}}} a^i_{\tau_{\bar{\mathbf{x}}}} \lambda_i(u), \quad \{a^i_{\tau_{\mathbf{x}}}\} = \arg\min_{a^i} \sum_{i \in I_{\bar{\mathbf{x}}}} \frac{|a^i|^2}{W(\bar{\mathbf{x}}, \mathbf{x}_i)},$$

$$\text{s. t. } \tau_{\bar{\mathbf{x}}}(p) = \sum_{i \in I_{\bar{\mathbf{x}}}} a^i \lambda_i(p), \ \forall p \in P \qquad \left(\tau^h_{\bar{\mathbf{x}}}(p) = \tau_{\bar{\mathbf{x}}}(p)\right)$$

2. (Practical Recipe) Least Squares formulation:

$$\tau^h_{\bar{\mathbf{x}}}(u) := \tau_{\bar{\mathbf{x}}}(p_{\bar{\mathbf{x}},u}), \quad p_{\bar{\mathbf{x}},u} = \arg\min_{p \in P} \sum_{i \in I_{\bar{\mathbf{x}}}} \left(\lambda_i(u) - \lambda_i(p)\right)^2 W(\bar{\mathbf{x}}, \mathbf{x}_i)$$

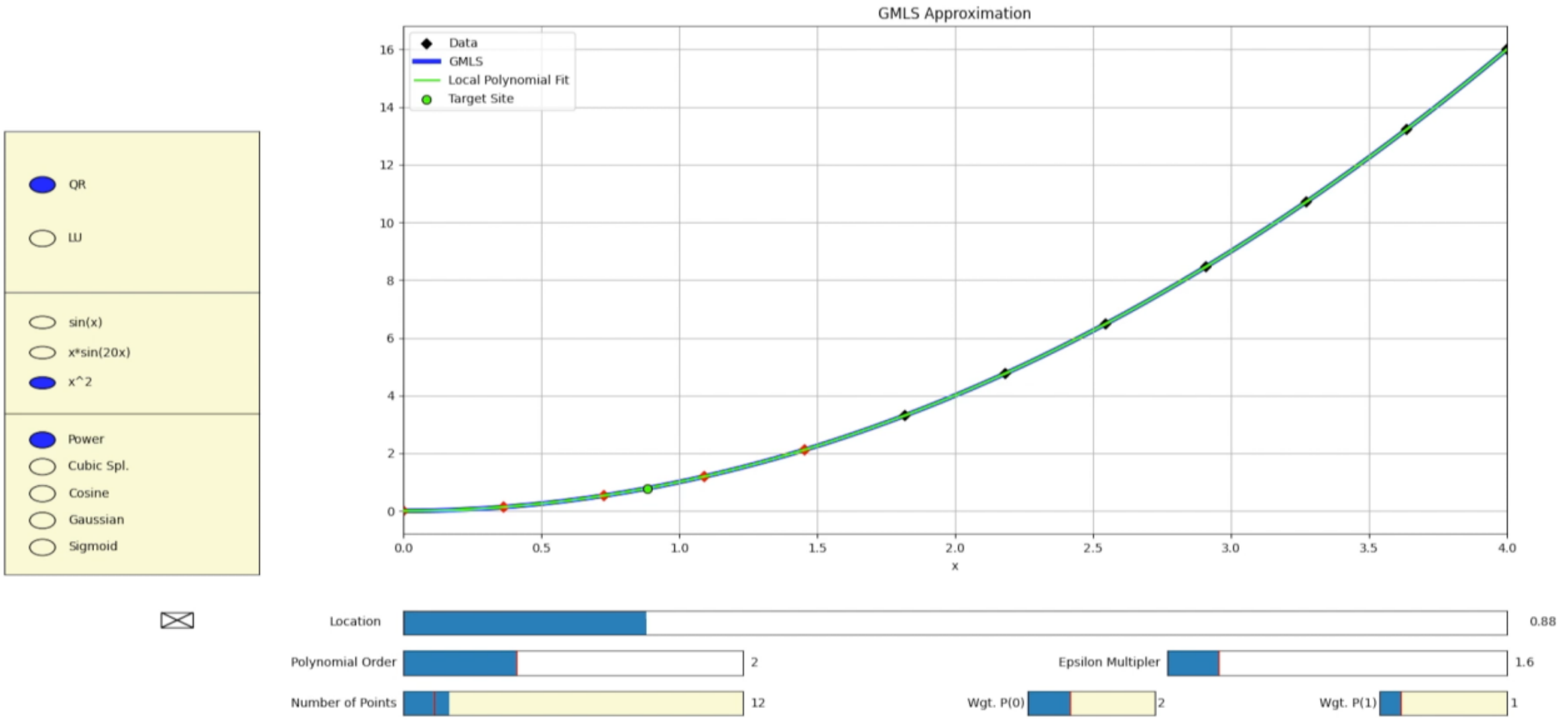
$$\mathrm{I}_{\bar{\mathbf{x}}} := \{i : W(\bar{\mathbf{x}}, \mathbf{x}_i) > 0\}$$



GMLS is a non-parametric regression: Evaluation at any point $\bar{x}$ requires solution of this LS (quadratic) problem.

DEMO [1D Regression]

```
>> git clone https://github.com/sandialabs/compadre.git
>> cd compadre/examples
>> python pycompadre_example_1d.py
```

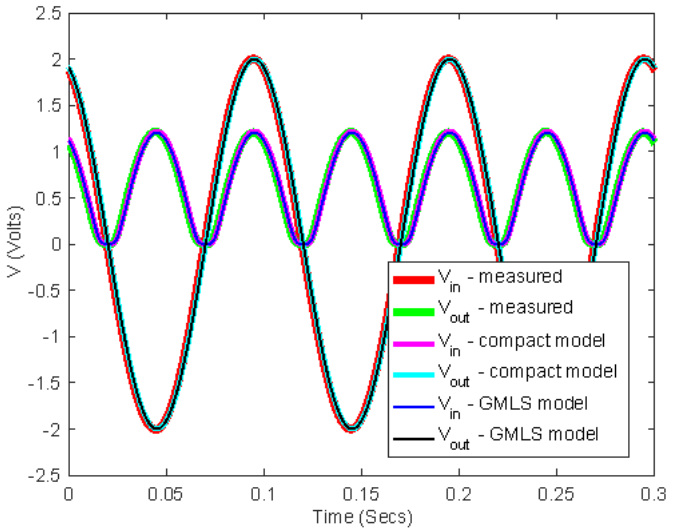


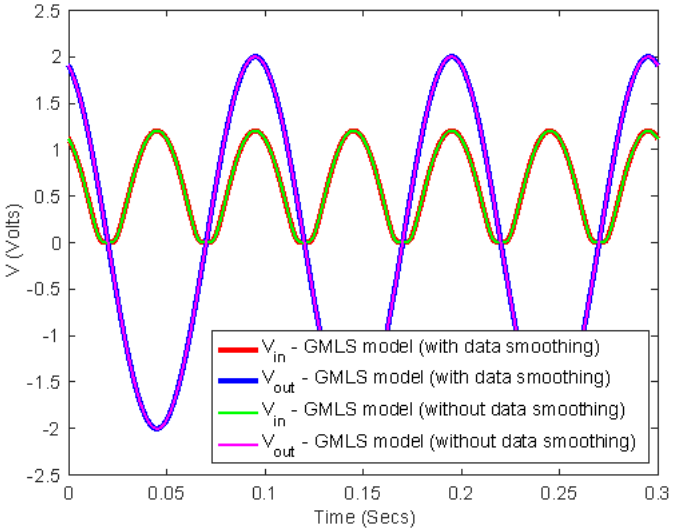
Fast switching 1N4148 diode in bridge rectifier

* All runs on collected laboratory data for 1N4148 diode

GMLS

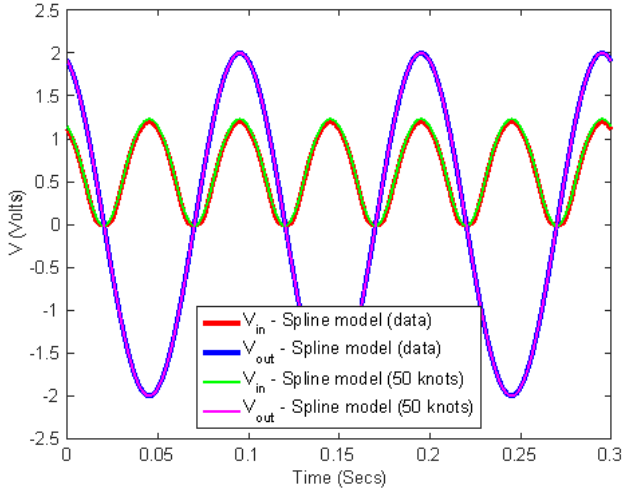
```
**************
* Netlist for Bridge Rectifier
**************
V3 1 2 SIN (0 2 10 0 0 -72)
R3 3 0 10M
R4 3 4 100K
YGENEXT pyd3 1 4
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py.../data/1N4148.dat}
YGENEXT pyd1 3 1
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py.../data/1N4148.dat}
YGENEXT pyd4 3 2
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py.../data/1N4148.dat}
YGENEXT pyd2 2 4
+ SPARAMS={NAME=MODULENAME,DATAFILE VALUE=../models/gmls_diode_1N4148.py.../data/1N4148.dat}
.TRAN 0 0.3s
.PRINT TRAN V(1) V(2) V(3) V(4) V(2,1) V(4,3)
.END
```

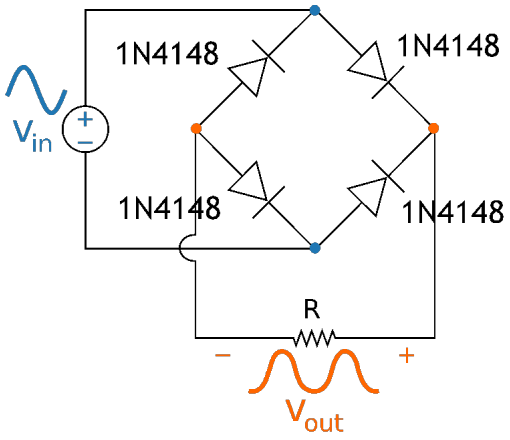




Cubic Splines

```
**************
* Netlist for Bridge Rectifier (Spline)
**************
V3 1 2 SIN (0 2 10 0 0 -72)
R3 3 0 10M
R4 3 4 100K
YGENEXT pyd3 1 4
+ SPARAMS={NAME=MODULENAME,PICKLEFILE VALUE=../models/spline_diode.py.../models/spline/spline_coeffs_1N4148_collected_smoothed_FULL.p}
YGENEXT pyd1 3 1
+ SPARAMS={NAME=MODULENAME,PICKLEFILE VALUE=../models/spline_diode.py.../models/spline/spline_coeffs_1N4148_collected_smoothed_FULL.p}
YGENEXT pyd4 3 2
+ SPARAMS={NAME=MODULENAME,PICKLEFILE VALUE=../models/spline_diode.py.../models/spline/spline_coeffs_1N4148_collected_smoothed_FULL.p}
YGENEXT pyd2 2 4
+ SPARAMS={NAME=MODULENAME,PICKLEFILE VALUE=../models/spline_diode.py.../models/spline/spline_coeffs_1N4148_collected_smoothed_FULL.p}
.TRAN 0 0.3s
.options device voltlim=0
.PRINT TRAN V(1) V(2) V(3) V(4) V(2,1) V(4,3)
.END
```



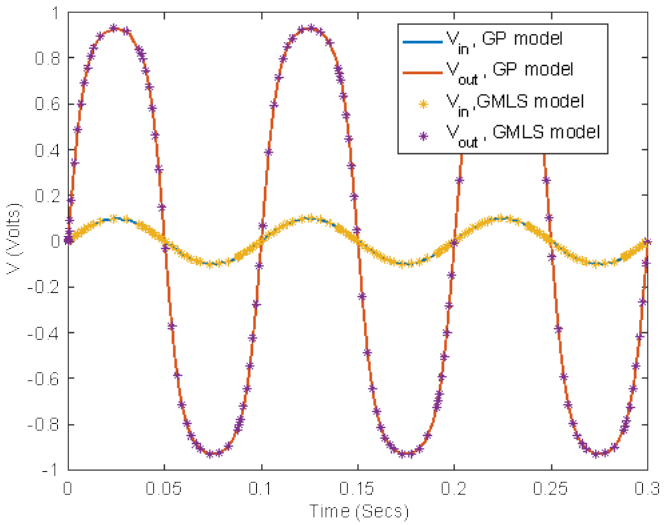


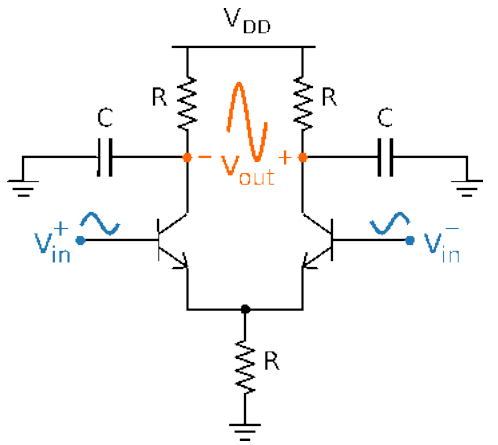
Operational Amplifier with BJTs

* All runs on data generated from synthetic MMBT2222, NPN, Fairchild

GMLS

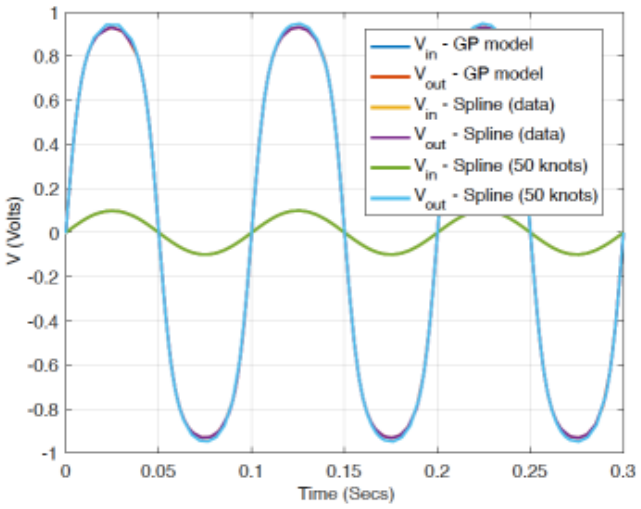
```
**************
* Netlist for Operational Amplifier (GMLS)
**************
VDD     1   0  DC   2.5
R1 1 4 1e4
R2 1 5 1e4
R3 6 0 5e3
C1 4 0 5e-12
C2 5 0 5e-12
YGENEXT pyQ1 4 7 6
+ SPARAMS={NAME=MODULENAME,DATAFILE
          VALUE=../models/gmls_bjt_2N2222.py.../data/2N2222_alan.01.dat}
RQ1 7 2 50
YGENEXT pyQ2 5 8  6
+ SPARAMS={NAME=MODULENAME,DATAFILE
          VALUE=../models/gmls_bjt_2N2222.py.../data/2N2222_alan.01.dat}
RQ2 8 3 50
Em_plus 2 0 VALUE={1+50e-3*sin(2*pi*10*time)}
Em_minus 3 0 VALUE={1-50e-3*sin(2*pi*10*time)}
```

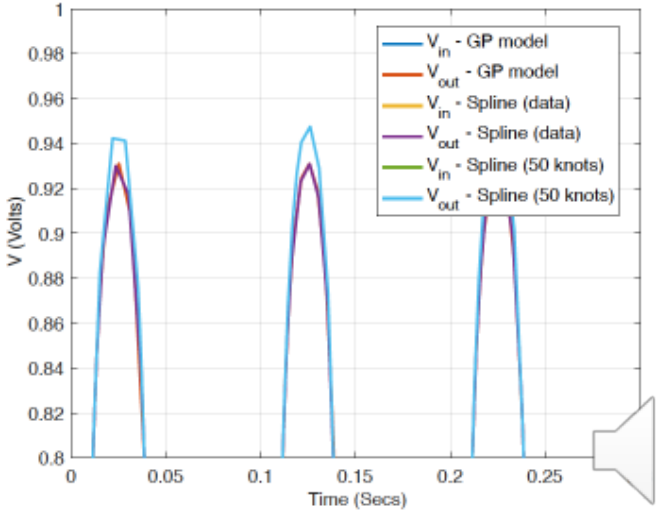




Cubic Splines

```
**************
* Netlist for Operational Amplifier (Spline)
**************
VDD     1   0  DC   2.5
R1 1 4 1e4
R2 1 5 1e4
R3 6 0 5e3
C1 4 0 5e-12
C2 5 0 5e-12
YGENEXT pyQ1 4 7 6
+ SPARAMS={NAME=MODULENAME,PICKLEFILE
          VALUE=../models/spline_bjt.py.../models/spline/spline_coeffs_2N2222_FULL.p
}
RQ1 7 2 50
YGENEXT pyQ2 5 8  6
+ SPARAMS={NAME=MODULENAME,PICKLEFILE
          VALUE=../models/spline_bjt.py.../models/spline/spline_coeffs_2N2222_FULL.p
}
RQ2 8 3 50
Em_plus 2 0 VALUE={1+50e-3*sin(2*pi*10*time)}
Em_minus 3 0 VALUE={1-50e-3*sin(2*pi*10*time)}
```





Conclusions

Xyce-PyMi

- is the glue connecting the production circuit simulator Xyce to the newest advancements in machine learning
- is easily installable using Spack
- provides ML developers and data scientists the ability to easily evaluate their models in a production circuit simulation environment and also allows circuit designers to **harness recent ML technologies.**
- can be used to **quickly prototype devices and subcircuits** using ML technology, enabling rapid compact model deployment

Reference

P. Kuberry, E. Keiter, **An embedded Python model interpreter for Xyce (Xyce-PyMi)**, *Sandia Report*, 2021. doi:10.2172/1813650. Link