Towards the Optimization of Integrated Transmission-Distribution Networks via the Rapid Prototyping of OPF Formulations with **PowerModelsITD.jl**

Juan Ospina, *Ph.D.*

Postdoctoral Researcher with the A-1 Information Systems and Modeling group at Los Alamos National Laboratory

July 14, 2022

LA-UR-22-25908

Fifth Workshop on Autonomous Energy Systems

Acknowledgements - Team & Funding

- David Fobes (A-1 LANL)
- Russell Bent (T-5 LANL)
- Andreas Wächter (Northwestern University)
- Xinyi Luo (Northwestern University)

This work was performed with the support of the **U.S. Department of Energy (DOE) Office of Electricity (OE) Advanced Grid Modeling (AGM)** Research Program under program manager **Ali Ghassemian**. We gratefully acknowledge Ali's support of this work.

Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

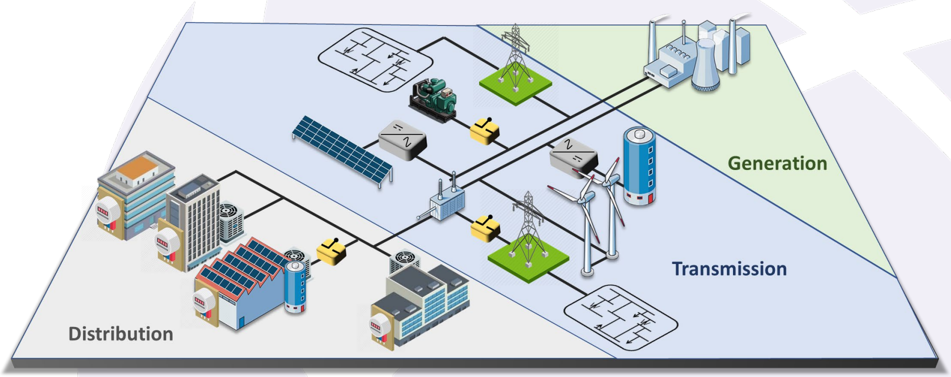
Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

Background

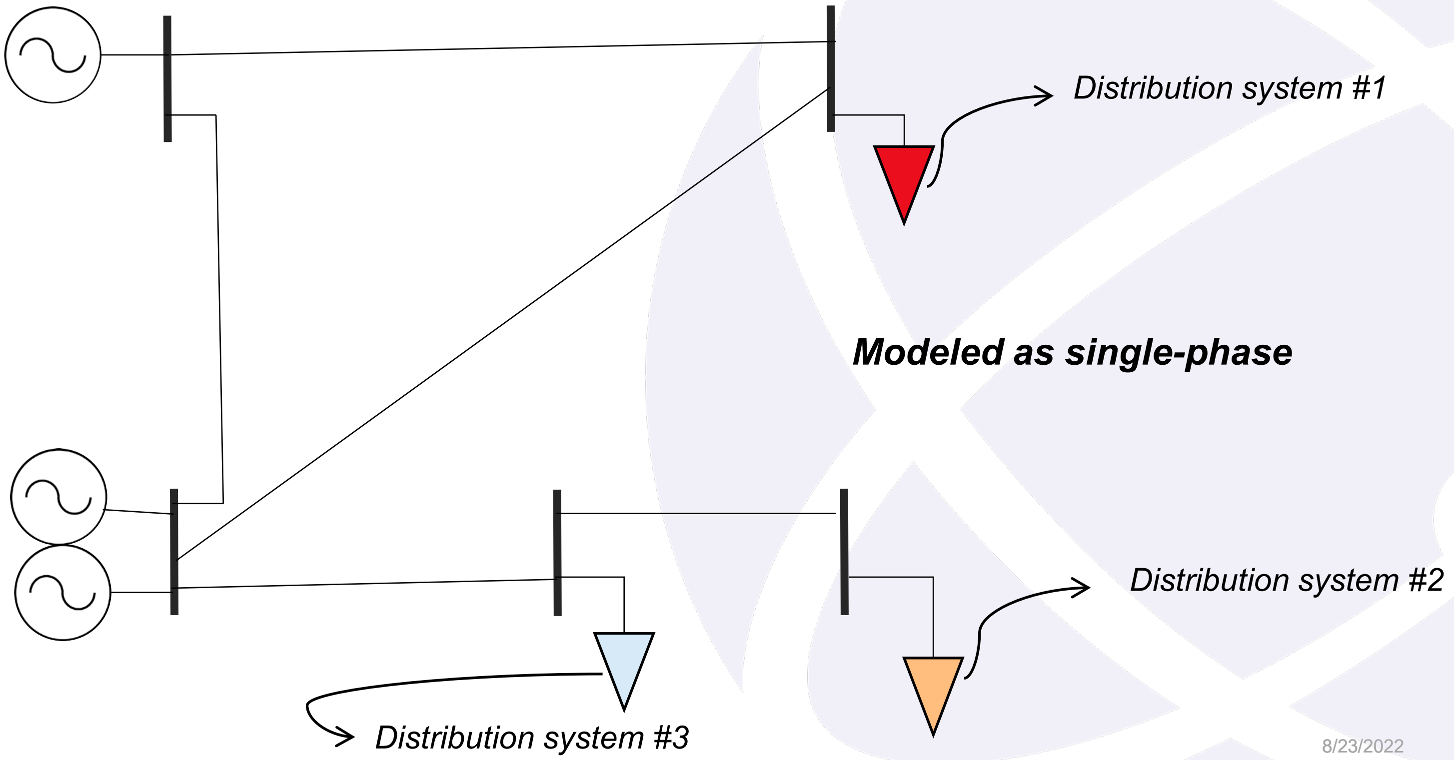
- Conventional electric power systems (EPS) are composed of:
  - **Generation**
  - **Transmission**
  - **Distribution**

- Managed independently by:
  - Transmission system (TSOs)
  - Distribution system operators (DSOs).



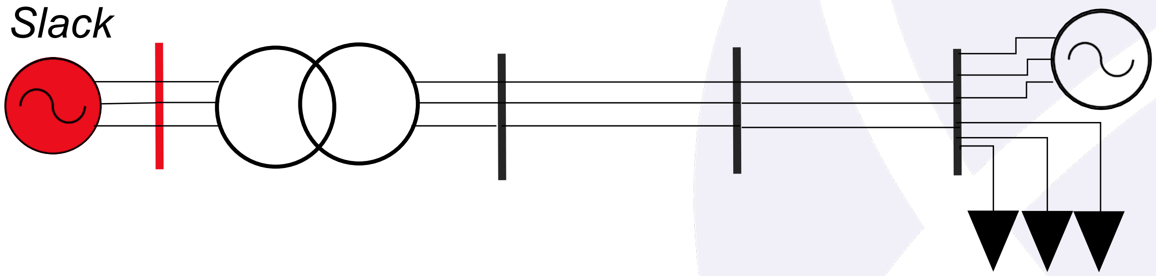
Background: TSOs

- TSOs traditionally model distribution systems as consumers (**loads**).



Background: DSOs

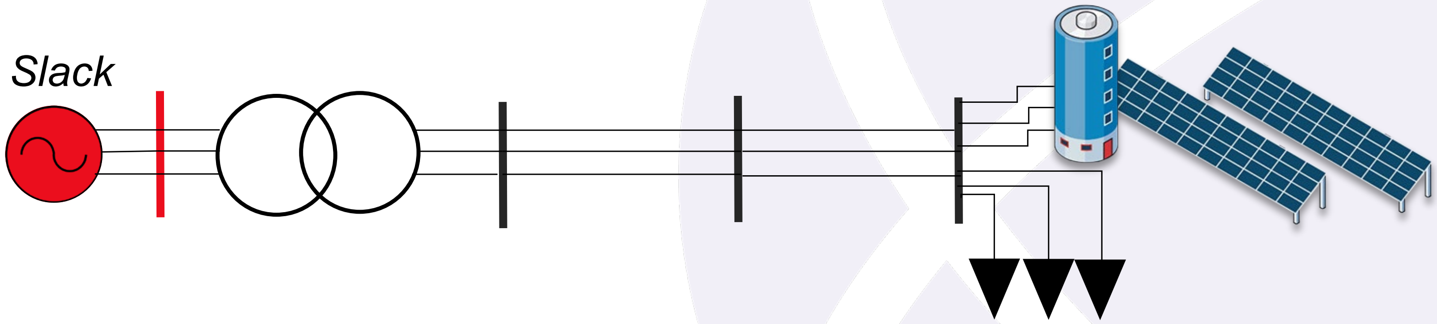
- DSOs traditionally regard transmission systems as slack buses with unlimited resources (often modeled as **voltage sources**).



Modeled as three-phase (multiconductor)

Background: Integration of DERs

- Distribution systems are becoming more **active**:
  - Integration of **Distributed Energy Resources** (DERs)
  - **Demand Response** (DR) Programs
  - Integration of **Information & Communication** Technologies (ICTs).



The **common** assumption of the distribution system being **just a load** seen from the **transmission system-side** is now unreasonable

Challenges

- Traditionally owned and operated by **separate entities**.
  - Competitive relationship -> unwillingness to share and/or combine models.
  - Assumption: centralized models may not be scalable and hard to solve.

- This **'independent' optimization** does **not** allow **optimal** dispatch of both T&D resources simultaneously.

Coordination between **T&D** networks will be **imperative** for the **optimal operation** of the power grid.

Challenges: Technical

"Coupling [transmission-distribution] **models** and **formulations** is a **non-trivial task**"

- How to model T&D '**Boundaries**'?

Common modeling practices are:

- Transmission systems as **single-phase**, and
- Distribution systems as **phase-unbalanced (multi-conductor)**

Challenges: Other Technical

- **Variable coefficient scaling**
  Powers and voltages over feeders can differ by **orders of magnitude**

- **Problem scaling**
  Distribution models **can be many times bigger** due to explicit multiconductor modeling

- **Convergence issues with AC OPF (nonlinear, nonconvex formulations)**

Challenges: Questions

- How can **grid operators optimally** manage resources across operational boundaries?
- How can **integrated utilities** (i.e., those who own both T&D) reduce operational costs?
- Is there a way to examine different formulations for T&D (e.g., **nonlinear, approximations, convex relaxations**) in a centralized problem specification (OPF, PF, etc.)?

Overall strategy: **Co-optimization of T&D networks**

Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

InfrastructureModels.jl

- **Core** package for **multi-infrastructure modeling** and **optimization** ecosystem

https://github.com/lanl-ansi/InfrastructureModels.jl

Core Design: PowerModels.jl Example

Separation of

- Formulations (AC polar, AC rectangular, DC polar, etc.)
- Problem Specifications (PF, OPF, etc.)

IV

Var…

M…

Su…

AC Rectan…

Variables:

$S_{g,k}, \forall k \in G$

$V_i^{\Re}, \forall i \in N$

$V_i^{\Im}, \forall i \in N$

$S_{ij}, \forall (i,j) \in$ …

Minimize:

$\sum_{k \in G_i} c_{2k} (\Re($ …

Subject to:

$S_{g,k}^l \le S_{g,k} \le$ …

$(v_i^l)^2 \le (V_i^{\Re}$ …

$\sum_{k \in G_i} S_{g,k} - S$ …

$|S_{ij}| \le s_{ij}^u, \forall$ …

…

AC Polar

Variables:

$S_{g,k}, \forall k \in G$

$V_i, \forall i \in N$

$\theta_i, \forall i \in N$

$S_{ij}, \forall (i,j) \in E \cup E_R$

Minimize:

$$\sum_{k \in G_i} c_{2k} (\Re(S_{g,k}))^2 + c_{1k} (\Re(S_{g,k})) + c_{0k}$$

Subject to:

$\theta_r = 0, \forall r \in R$

$S_{g,k}^l \le S_{g,k} \le S_{g,k}^u, \forall k \in G$

$v_i^l \le |V_i| \le v_i^u, \forall i \in N$

$\sum_{k \in G_i} S_{g,k} - S_{d,i} - \sum_{(i,j) \in E \cup E_R} S_{ij} = 0, \forall i \in N$

$\theta_{ij}^{\Delta l} \le \theta_i - \theta_j \le \theta_{ij}^{\Delta u}, \forall (i,j) \in E$

$|S_{ij}| \le s_{ij}^u, \forall (i,j) \in E \cup E_R$

…

```
function build_opf(pm::AbstractPowerModel)
    variable_gen_power(pm)
    variable_bus_voltage(pm)
    variable_branch_power(pm)

    objective_min_fuel_and_flow_cost(pm)

    constraint_model_voltage(pm)

    for i in ids(pm, :ref_buses)
        constraint_theta_ref(pm, i)
    end

    for i in ids(pm, :bus)
        constraint_power_balance(pm, i)
    end

    for i in ids(pm, :branch)
        constraint_ohms_yt_from(pm, i)
        constraint_ohms_yt_to(pm, i)

        constraint_voltage_angle_difference(pm,i)

        constraint_thermal_limit_from(pm, i)
        constraint_thermal_limit_to(pm, i)
    end
end
```

Core language feature: Multiple dispatch

PowerModels.jl

- **PowerModels.jl** (PM) is free and open-source software library to solve **Transmission Systems**
- **Fueled** by:
  - **Explosion in the number** of power flow nonconvex, approximations, and relaxations
  - Difficulty of **evaluation** using a **common platform**
- Written in **Julia** and **JuMP.jl**

https://github.com/lanl-ansi/PowerModels.jl

PowerModels.jl

- **Perform** various **quasi-steady-state optimizations** of power **transmission networks.**

Problem Specifications

Power Flow (pf)
Optimal Power Flow (opf)
Optimal Transm. Switching (ots)
Transmission Net. Expansion (tnep)

Formulations

AC polar (ACP)
AC rectangular (ACR)
DC polar (DCP) – approximation
IV rectangular (IVR)
SDP – relaxation
SOC – relaxation
…

Modeled as single-phase

Coffrin, C., Bent, R., Sundar, K., Ng, Y., & Lubin, M. (2018, June). Powermodels. jl: An open-source framework for exploring power flow formulations. In 2018 Power Systems Computation Conference (PSCC) (pp. 1-8). IEEE.

PowerModels.jl

PowerModels.jl

Status: CI passing | codecov 94% | Documentation passing

PowerModels.jl is a Julia/JuMP package for Steady-State Power Network Optimization. It is designed to enable computational evaluation of emerging power network formulations and algorithms in a common platform. The code is engineered to decouple problem specifications (e.g. Power Flow, Optimal Power Flow, ...) from the power network formulations (e.g. AC, DC-approximation, SOC-relaxation, ...). This enables the definition of a wide variety of power network formulations and their comparison on common problem specifications.

Core Problem Specifications

- Power Flow (pf)
- Optimal Power Flow (opf)
- Optimal Transmission Switching (ots)
- Transmission Network Expansion Planning (tnep)

Core Network Formulations

- AC (polar and rectangular coordinates)
- DC Approximation (polar coordinates)
- LPAC Approximation (polar coordinates)
- SDP Relaxation (W-space)
- SOC Relaxation (W-space)
- QC Relaxation (W+L-space)
- IV (rectangular coordinates)

Network Data Formats

- Matpower ".m" files
- PTI ".raw" files (PSS(R)E v33 specification)

Provides a **common platform** for **baseline** implementations

https://github.com/lanl-ansi/PowerModels.jl

PowerModelsDistribution.jl

- Building on the success of **PM**, **PMD** was built to addresses similar issues as PM, but for **phase unbalanced power systems**

- **Quasi-steady-state multi-conductor** (e.g., three-phase, explicit neutral & ground) **phase-unbalanced** optimization problems

https://github.com/lanl-ansi/PowerModelsDistribution.jl

PowerModelsDistribution.jl

- **Perform** various **quasi-steady-state optimizations** of power unbalanced multi-conductor **distribution networks.**

Problem Specifications

Power Flow (pf)
Optimal Power Flow (opf)
Optimal Power Flow with
on-load tap-changer (opf_oltc) …

Formulations

AC polar unbalanced (ACPU)
AC rectangular unbalanced (ACRU)
IV rectangular unbalanced (IVRU)
SDP – relaxation
SOC – relaxation
…

Modeled as phase unbalanced multi-conductor

Fobes, D. M., Claeys, S., Geth, F., & Coffrin, C. (2020). PowerModelsDistribution. jl: An open-source framework for exploring distribution power flow formulations. Electric Power Systems Research, 189, 106664.

PowerModelsDistribution.jl

PowerModelsDistribution.jl

CI passing | Documentation passing

PowerModelsDistribution.jl is an extension package of PowerModels.jl for Steady-State Power Distribution Network Optimization. It is designed to enable computational evaluation of emerging power network formulations and algorithms in a common platform. The code is engineered to decouple problem specifications (e.g. Power Flow, Optimal Power Flow, ...) from the power network formulations (e.g. AC, linear-approximation, SOC-relaxation, ...). This enables the definition of a wide variety of power network formulations and their comparison on common problem specifications.

Core Problem Specifications

- Power Flow (pf)
  - ACP, ACR, IVR, LinDist3Flow, NFA, DCP
- Optimal Power Flow (opf)
  - ACP, ACR, IVR, LinDist3Flow, NFA, DCP
- Continuous load shed, minimum load delta (mld)
  - ACP, LinDist3Flow, NFA
- Optimal Power Flow with on-load tap-changer (opf_oltc)
  - ACP

Provides a **common platform** for **baseline** implementations

https://github.com/lanl-ansi/PowerModelsDistribution.jl

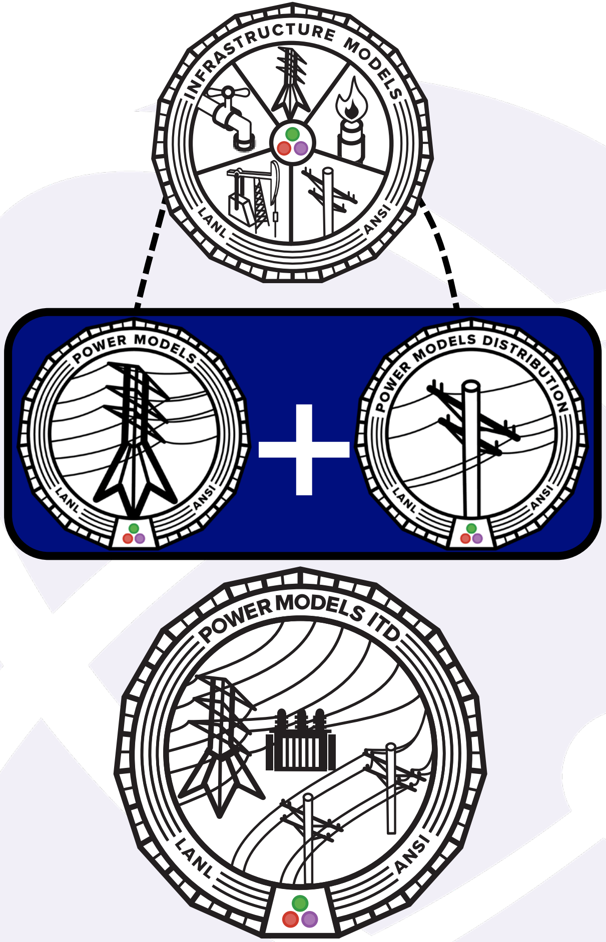
PowerModelsITD.jl

- **PMITD** enables
  - rapid prototyping of **integrated transmission-distribution (ITD) optimization problems**

- **PMITD** provides

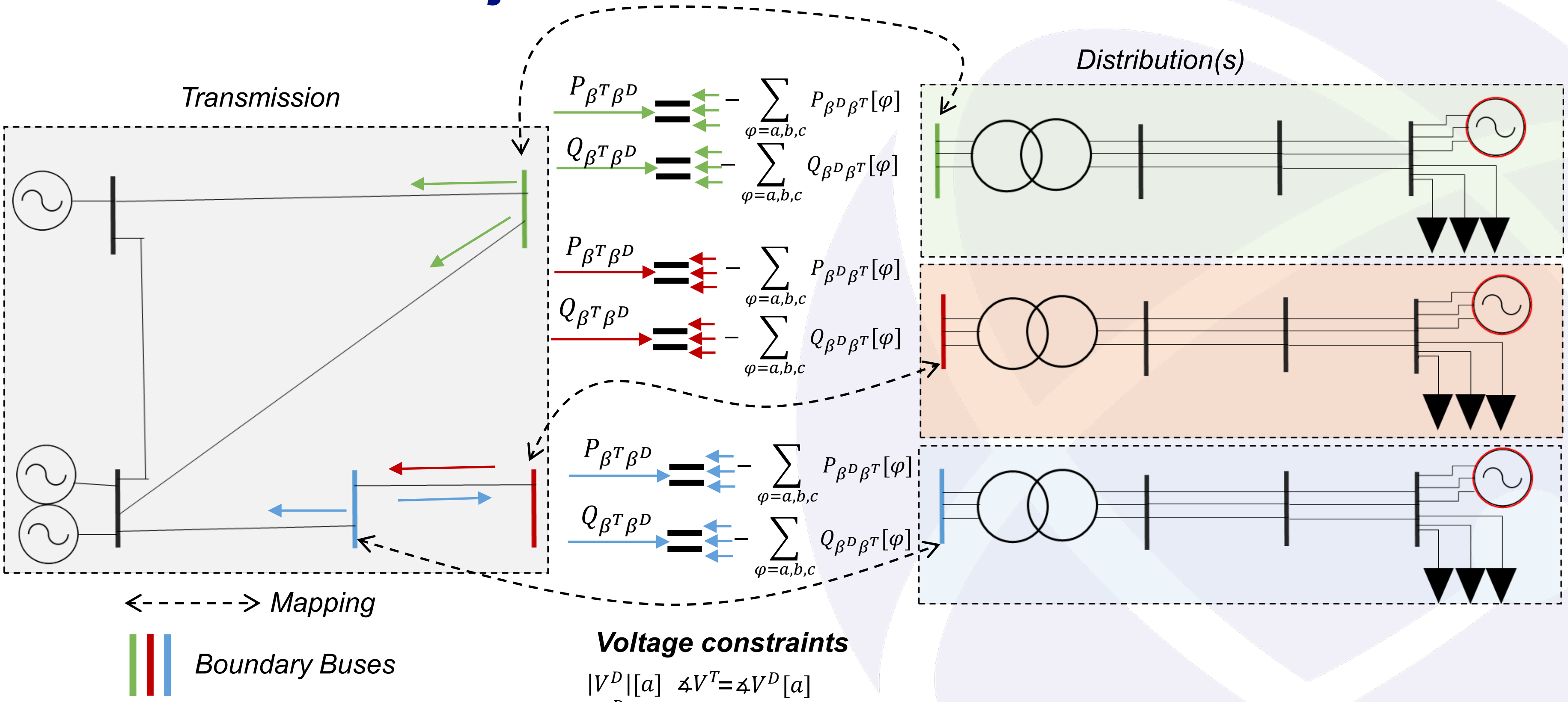
  - **baseline implementations** of **steady-state ITD optimization** problems

  - **a common platform** for the **evaluation of emerging formulations** and optimization problems.



https://github.com/lanl-ansi/PowerModelsITD.jl

PowerModelsITD.jl



Voltage constraints

$$|V^T| = \begin{matrix} |V^D|[a] \\ |V^D|[b] \\ |V^D|[c] \end{matrix}$$

$$\measuredangle V^T = \measuredangle V^D[a]$$

$$\measuredangle V^D[b] = \measuredangle V^D[a] + 120°$$

$$\measuredangle V^D[c] = \measuredangle V^D[a] - 120°$$

PowerModelsITD.jl

Problem Specifications

Integrated T&D Power Flow (pfitd)
Integrated T&D Optimal Power Flow (opfitd)
Integrated T&D Optimal Power Flow with on-load tap-changer (opfitd_oltc)
…

Formulations

ACP-ACPU
ACR-ACRU
IVR-IVRU
NFA-NFAU
SOCBFM- LinDis3Flow
…

PowerModelsITD.jl

PowerModelsITD.jl

CI passing | Documentation passing

PowerModelsITD.jl is an extention package of PowerModels.jl and PowerModelsDistribution.jl for Steady-State Integrated Power Transmission-Distribution Network Optimization. It is designed to enable computational evaluation of emerging power network formulations and algorithms in a common platform. The code is engineered to decouple problem specifications (e.g. Power Flow, Optimal Power Flow, ...) from the power network formulations (e.g. AC, linear-approximation, SOC-relaxation, ...) on both transmission and distribution system. Thus, enabling the definition of a wide variety of power network formulations and their comparison on common problem specifications.

Core Problem Specifications

- Integrated T&D Power Flow (pfitd)
- Integrated T&D Optimal Power Flow (opfitd)
- Integrated T&D Optimal Power Flow with on-load tap-changer (opfitd_oltc)
- Integrated T&D Optimal power flow at transmission and minimum load delta at distribution system (opfitd_dmld)

https://github.com/lanl-ansi/PowerModelsITD.jl

Provides a **common platform** for **baseline** implementations

Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

Integrated Transmission-Distribution (ITD) Formulation

- Mathematical Formulation **ACP-ACPU** - Notation

Sets

$\mathcal{T}$	Belongs to transmission network.
$\mathcal{D}$	Belongs to distribution network.
$\mathcal{B}$	Set of boundary buses.
Λ	Set of boundary links.
N	Set of buses.
R	Set of reference buses.
G	Set of generators.
G_i	Generator at bus i .
E, E_R	Set of branches (forward and reverse).

Parameters

$\Re$	Real part.
$\Im$	Imaginary part.

Parameters

$\Re$	Real part.
$\Im$	Imaginary part.
$\Phi = a, b, c$	Multi-conductor phases.
$\chi \to \mathcal{T}, \mathcal{D}$	Belongs to $\mathcal{T}$ or $\mathcal{D}$.
v_i^l	Voltage lower bounds.
v_i^u	Voltage upper bounds.
$P_{g,k}^{\chi,l}$	Gen. active power lower bounds.
$P_{g,k}^{\chi,u}$	Gen. active power upper bounds
$Q_{g,k}^{\chi,l}$	Gen. reactive power lower bounds.
$Q_{g,k}^{\chi,u}$	Gen. reactive power upper bounds
c_2, c_1, c_0	Gen. cost components
$P_{d,i}^{\chi}$	Active power demand at bus i
g_i^s	Shunt conductance at bus i.
b_i^s	Shunt susceptance at bus i.
$p_{ij}^{\chi,l}$	Active power flow on line (i, j) lower bounds.
$p_{ij}^{\chi,u}$	Active power flow on line (i, j) upper bounds.
$q_{ij}^{\chi,l}$	Reactive power flow on line (i, j) lower bounds.
$q_{ij}^{\chi,u}$	Reactive power flow on line (i, j) upper bounds.
τ_{ij}	Transformer tap ratio on line (i, j).
ϕ_{ij}	Transformer angle on line (i, j).
g_{ij}^E	Conductance on line (i, j).
b_{ij}^E	Susceptance on line (i, j).
b_{ij}^C	Branch charging susceptance of line (i, j).
x_{ij}	Reactance on line $(i, j) \in E$.
$\theta_{ij}^{\Delta l}$	Branch voltage angle difference lower bounds.
$\theta_{ij}^{\Delta u}$	Branch voltage angle difference upper bounds.

Transmission Variables

$P_{g,k}^{\mathcal{T}}$	Gen. k active power output.
$Q_{g,k}^{\mathcal{T}}$	Gen. k reactive power output.
$V_i^{\mathcal{T}}$	Voltage magnitude at bus i.
$\theta_i^{\mathcal{T}}$	Voltage angle at bus i.
$P_{ij}^{\mathcal{T}}$	Active power flow on line (i, j).
$Q_{ij}^{\mathcal{T}}$	Reactive power flow on line (i, j).

Boundary Variables

$P_{\beta^{\mathcal{T}}\beta^{\mathcal{D}}}^{\mathcal{T}}$	Active power flow from $\beta^{\mathcal{T}}$ to $\beta^{\mathcal{D}}$.
$Q_{\beta^{\mathcal{T}}\beta^{\mathcal{D}}}^{\mathcal{T}}$	Reactive power flow from $\beta^{\mathcal{T}}$ to $\beta^{\mathcal{D}}$.
$P_{\beta^{\mathcal{D}}\beta^{\mathcal{T}}}^{\mathcal{D},\varphi}$	Active power flow from $\beta^{\mathcal{D}}$ to $\beta^{\mathcal{T}}$ phase φ.
$Q_{\beta^{\mathcal{D}}\beta^{\mathcal{T}}}^{\mathcal{D},\varphi}$	Reactive power flow from $\beta^{\mathcal{D}}$ to $\beta^{\mathcal{T}}$ phase φ.

Distribution Variables

$P_{g,m}^{\mathcal{D},\varphi}$	Gen. m active power output on phase φ.
$Q_{g,m}^{\mathcal{D},\varphi}$	Gen. m reactive power output on phase φ.
$v_i^{\mathcal{D},\varphi}$	Voltage magnitude at bus i phase φ.
$\theta_i^{\mathcal{D},\varphi}$	Voltage angle at bus i phase φ.
$P_{ij}^{\mathcal{D},\varphi}$	Active power flow on line (i, j) phase φ.
$Q_{ij}^{\mathcal{D},\varphi}$	Reactive power flow on line (i, j) phase φ.

Integrated Transmission-Distribution (ITD) Formulation

- Mathematical Formulation **ACP-ACPU** – ITD Cost Function

$$\min \left(\sum_{k \in G^{\mathcal{T}}} c_{2k}(P_{g,k}^{\mathcal{T}})^2 + c_{1k}(P_{g,k}^{\mathcal{T}}) + c_{0k} \right) + \left(\sum_{m \in G^{\mathcal{D}}} c_{2m}(\sum_{\varphi \in \Phi} P_{g,m}^{\mathcal{D},\varphi})^2 + c_{1m}(\sum_{\varphi \in \Phi} P_{g,m}^{\mathcal{D},\varphi}) + c_{0m} \right) \quad (1)$$

Transmission generation cost

Distribution generation cost

Integrated Transmission-Distribution (ITD) Formulation

- Mathematical Formulation **ACP-ACPU** - Transmission

Reference

$$\theta_r^{\mathcal{T}} = 0, \ \forall r \in R \quad (2)$$

Gen. limits

$$P_{g,k}^{\mathcal{T},l} \leq P_{g,k}^{\mathcal{T}} \leq P_{g,k}^{\mathcal{T},u}, \quad \forall k \in G^{\mathcal{T}} \quad (3)$$

$$Q_{g,k}^{\mathcal{T},l} \leq Q_{g,k}^{\mathcal{T}} \leq Q_{g,k}^{\mathcal{T},u}, \quad \forall k \in G^{\mathcal{T}} \quad (4)$$

Volt. limits

$$v_i^l \leq |V_i| \leq v_i^u, \ \forall i \in N^{\mathcal{T}} \quad (5)$$

Active power line flows

$$P_{ij}^{\mathcal{T}} = \frac{1}{\tau_{ij}^2} g_{ij}^{E^{\mathcal{T}}} V_i^2 - \frac{1}{\tau_{ij}} V_i V_j (g_{ij}^{E^{\mathcal{T}}} cos(\theta_i - \theta_j - \phi_{ij}) \ldots + b_{ij}^{E^{\mathcal{T}}} sin(\theta_i - \theta_j - \phi_{ij})), \quad \forall (i,j) \in E^{\mathcal{T}} \quad (6)$$

$$P_{ji}^{\mathcal{T}} = g_{ij}^{E^{\mathcal{T}}} V_j^2 - \frac{1}{\tau_{ij}} V_i V_j (g_{ij}^{E^{\mathcal{T}}} cos(\theta_j - \theta_i + \phi_{ij}) \ldots + b_{ij}^{E^{\mathcal{T}}} sin(\theta_j - \theta_i + \phi_{ij})), \quad \forall (i,j) \in E^{\mathcal{T}} \quad (7)$$

Reactive power line flows

$$Q_{ij}^{\mathcal{T}} = -\frac{1}{\tau_{ij}^2} \left(b_{ij}^{E^{\mathcal{T}}} + \frac{b_{ij}^C}{2} \right) V_i^2 - \frac{1}{\tau_{ij}} V_i V_j (g_{ij}^{E^{\mathcal{T}}} cos(\theta_i \ldots - \theta_j - \phi_{ij}) - b_{ij}^{E^{\mathcal{T}}} sin(\theta_i - \theta_j - \phi_{ij})), \ldots \forall (i,j) \in E^{\mathcal{T}} \quad (8)$$

$$Q_{ji}^{\mathcal{T}} = -\left(b_{ij}^{E^{\mathcal{T}}} + \frac{b_{ij}^C}{2} \right) V_j^2 - \frac{1}{\tau_{ij}} V_i V_j (g_{ij}^{E^{\mathcal{T}}} cos(\theta_j \ldots - \theta_i + \phi_{ij}) - b_{ij}^{E^{\mathcal{T}}} sin(\theta_j - \theta_i + \phi_{ij})), \ldots \forall (i,j) \in E^{\mathcal{T}} \quad (9)$$

Active power balance constraints

$$\sum_{k \in G_i^{\mathcal{T}}} P_{g,k}^{\mathcal{T}} - P_{d,i}^{\mathcal{T}} - (V_i)^2 g_i^s - \sum_{(i,j) \in E^{\mathcal{T}} \cup E_R^{\mathcal{T}}} P_{ij}^{\mathcal{T}} \ldots - \sum_{(i,\beta) \in \Lambda, \beta \in N^{\mathcal{D}} \cap N^{\mathcal{B}}} P_{i\beta}^{\mathcal{T}} = 0, \quad \forall i \in N^{\mathcal{T}} \quad (10)$$

Boundary flow

Reactive power balance constraints

$$\sum_{k \in G_i^{\mathcal{T}}} Q_{g,k}^{\mathcal{T}} - Q_{d,i}^{\mathcal{T}} - (V_i)^2 b_i^s - \sum_{(i,j) \in E^{\mathcal{T}} \cup E_R^{\mathcal{T}}} Q_{ij}^{\mathcal{T}} \ldots - \sum_{(i,\beta) \in \Lambda, \beta \in N^{\mathcal{D}} \cap N^{\mathcal{B}}} Q_{i\beta}^{\mathcal{T}} = 0, \quad \forall i \in N^{\mathcal{T}} \quad (11)$$

Boundary flow

Active/Reactive Power limits

$$|P_{ij}| \leq p_{ij}^{\mathcal{T},u}, \quad \forall (i,j) \in E^{\mathcal{T}} \cup E_R^{\mathcal{T}} \quad (12)$$

$$|Q_{ij}| \leq q_{ij}^{\mathcal{T},u}, \quad \forall (i,j) \in E^{\mathcal{T}} \cup E_R^{\mathcal{T}} \quad (13)$$

Angle diff. limits

$$\theta_{ij}^{\Delta l} \leq \theta_i - \theta_j \leq \theta_{ij}^{\Delta u}, \quad \forall (i,j) \in E^{\mathcal{T}} \quad (14)$$

Integrated Transmission-Distribution (ITD) Formulation

- Mathematical Formulation **ACP-ACPU** - Distribution

$$\begin{bmatrix} P_{g,m}^{\mathcal{D},l,a} \\ P_{g,m}^{\mathcal{D},l,b} \\ P_{g,m}^{\mathcal{D},l,c} \end{bmatrix} \le \begin{bmatrix} P_{g,m}^{\mathcal{D},a} \\ P_{g,m}^{\mathcal{D},b} \\ P_{g,m}^{\mathcal{D},c} \end{bmatrix} \le \begin{bmatrix} P_{g,m}^{\mathcal{D},u,a} \\ P_{g,m}^{\mathcal{D},u,b} \\ P_{g,m}^{\mathcal{D},u,c} \end{bmatrix}, \quad \forall m \in G^{\mathcal{D}} \quad (23)$$

Gen. limits

$$\begin{bmatrix} Q_{g,m}^{\mathcal{D},l,a} \\ Q_{g,m}^{\mathcal{D},l,b} \\ Q_{g,m}^{\mathcal{D},l,c} \end{bmatrix} \le \begin{bmatrix} Q_{g,m}^{\mathcal{D},a} \\ Q_{g,m}^{\mathcal{D},b} \\ Q_{g,m}^{\mathcal{D},c} \end{bmatrix} \le \begin{bmatrix} Q_{g,m}^{\mathcal{D},u,a} \\ Q_{g,m}^{\mathcal{D},u,b} \\ Q_{g,m}^{\mathcal{D},u,c} \end{bmatrix}, \quad \forall m \in G^{\mathcal{D}} \quad (24)$$

$$\begin{bmatrix} v_i^{l,a} \\ v_i^{l,b} \\ v_i^{l,c} \end{bmatrix} \le \begin{bmatrix} |v_i^a| \\ |v_i^b| \\ |v_i^c| \end{bmatrix} \le \begin{bmatrix} v_i^{u,a} \\ v_i^{u,b} \\ v_i^{u,c} \end{bmatrix}, \ \forall i \in N^{\mathcal{D}} \quad (25)$$

Volt. limits

$$P_{ij}^{\mathcal{D},\varphi} = \frac{1}{(\tau_{ij}^{\varphi})^2} g_{ij}^{\varphi} (v_i^{\varphi})^2 - \frac{1}{\tau_{ij}^{\varphi}} v_i^{\varphi} \sum_{\rho=a,b,c} v_j^{\rho} (g_{ij}^{\varphi\rho} cos(\theta_i^{\varphi}$$
$$\ldots - \theta_j^{\rho} - \phi_{ij}^{\varphi\rho}) + b_{ij}^{\varphi\rho} sin(\theta_i^{\varphi} - \theta_j^{\rho} - \phi_{ij}^{\varphi\rho})),$$
$$\ldots \quad \forall \varphi \in \Phi, \ \forall (i,j) \in E^{\mathcal{D}} \quad (26)$$

... (27)

Active power line flows

$$Q_{ij}^{\mathcal{D},\varphi} = -\frac{1}{(\tau_{ij}^{\varphi})^2} \left(b_{ij}^{\varphi} + \frac{b_{ij}^{C,\varphi}}{2} \right) (v_i^{\varphi})^2$$
$$\ldots - \frac{1}{\tau_{ij}^{\varphi}} v_i^{\varphi} \sum_{\rho=a,b,c} v_j^{\rho} (g_{ij}^{\varphi\rho} cos(\theta_i^{\varphi} - \theta_j^{\rho} - \phi_{ij}^{\varphi\rho})$$
$$\ldots - b_{ij}^{\varphi\rho} sin(\theta_i^{\varphi} - \theta_j^{\rho} - \phi_{ij}^{\varphi\rho})), \ \forall \varphi \in \Phi, \ \forall (i,j) \in E^{\mathcal{D}} \quad (28)$$

... (29)

Reactive power line flows

$$\sum_{m \in G_i^{\mathcal{D}}} \sum_{\varphi \in \Phi} P_{g,m}^{\mathcal{D},\varphi} - \sum_{\varphi \in \Phi} P_{d,i}^{\mathcal{D},\varphi} - \sum_{\varphi \in \Phi} (v_i^{\varphi})^2 g_i^{s,\varphi}$$
$$\ldots - \sum_{(i,j) \in E^{\mathcal{D}} \cup E_R^{\mathcal{D}}} \sum_{\varphi \in \Phi} P_{ij}^{\mathcal{D},\varphi} \quad (30)$$
$$\ldots - \sum_{(i,\beta) \in \Lambda, \beta \in N^{\mathcal{T}} \cap N^{\mathcal{B}}} \sum_{\varphi \in \Phi} P_{i\beta}^{\mathcal{D},\varphi} = 0, \quad \forall i \in N^{\mathcal{D}}$$

Boundary flow

Active power balance constraints

$$\sum_{m \in G_i^{\mathcal{D}}} \sum_{\varphi \in \Phi} Q_{g,m}^{\mathcal{D},\varphi} - \sum_{\varphi \in \Phi} Q_{d,i}^{\mathcal{D},\varphi} - \sum_{\varphi \in \Phi} (v_i^{\varphi})^2 b_i^{s,\varphi}$$
$$\ldots - \sum_{(i,j) \in E^{\mathcal{D}} \cup E_R^{\mathcal{D}}} \sum_{\varphi \in \Phi} Q_{ij}^{\mathcal{D},\varphi} \quad (31)$$
$$\ldots - \sum_{(i,\beta) \in \Lambda, \beta \in N^{\mathcal{T}} \cap N^{\mathcal{B}}} \sum_{\varphi \in \Phi} Q_{i\beta}^{\mathcal{D},\varphi} = 0, \quad \forall i \in N^{\mathcal{D}}$$

Boundary flow

Reactive power balance constraints

$$\begin{bmatrix} |P_{ij}^{\mathcal{D},a}| \\ |P_{ij}^{\mathcal{D},b}| \\ |P_{ij}^{\mathcal{D},c}| \end{bmatrix} \le \begin{bmatrix} p_{ij}^{\mathcal{D},u,a} \\ p_{ij}^{\mathcal{D},u,b} \\ p_{ij}^{\mathcal{D},u,c} \end{bmatrix}, \ \forall (i,j) \in E^{\mathcal{D}} \cup E_R^{\mathcal{D}} \quad (32)$$

$$\begin{bmatrix} |Q_{ij}^{\mathcal{D},a}| \\ |Q_{ij}^{\mathcal{D},b}| \\ |Q_{ij}^{\mathcal{D},c}| \end{bmatrix} \le \begin{bmatrix} q_{ij}^{\mathcal{D},u,a} \\ q_{ij}^{\mathcal{D},u,b} \\ q_{ij}^{\mathcal{D},u,c} \end{bmatrix}, \ \forall (i,j) \in E^{\mathcal{D}} \cup E_R^{\mathcal{D}} \quad (33)$$

Active/Reactive Power limits

$$\begin{bmatrix} \theta_{ij}^{\Delta l,a} \\ \theta_{ij}^{\Delta l,b} \\ \theta_{ij}^{\Delta l,c} \end{bmatrix} \le \begin{bmatrix} \theta_i^{\mathcal{D},a} \\ \theta_i^{\mathcal{D},b} \\ \theta_i^{\mathcal{D},c} \end{bmatrix} - \begin{bmatrix} \theta_j^{\mathcal{D},a} \\ \theta_j^{\mathcal{D},b} \\ \theta_j^{\mathcal{D},c} \end{bmatrix} \le \begin{bmatrix} \theta_{ij}^{\Delta u,a} \\ \theta_{ij}^{\Delta u,b} \\ \theta_{ij}^{\Delta u,c} \end{bmatrix}, \forall (i,j) \in E^{\mathcal{D}} \quad (34)$$

Angle diff. limits

Integrated Transmission-Distribution (ITD) Formulation

- Mathematical Formulation **ACP-ACPU** - Boundary

$$\sum_{\varphi \in \Phi} P^{\mathcal{D},\varphi}_{\beta^{\mathcal{D}}\beta^{\mathcal{T}}} + P^{\mathcal{T}}_{\beta^{\mathcal{T}}\beta^{\mathcal{D}}} = 0, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (15)$$

$$\sum_{\varphi \in \Phi} Q^{\mathcal{D},\varphi}_{\beta^{\mathcal{D}}\beta^{\mathcal{T}}} + Q^{\mathcal{T}}_{\beta^{\mathcal{T}}\beta^{\mathcal{D}}} = 0, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (16)$$

Active/Reactive power flows @ **boundary(ies)**

$$V_{\beta^{\mathcal{T}}} = v^{a}_{\beta^{\mathcal{D}}}, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (17)$$

$$V_{\beta^{\mathcal{T}}} = v^{b}_{\beta^{\mathcal{D}}}, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (18)$$

$$V_{\beta^{\mathcal{T}}} = v^{c}_{\beta^{\mathcal{D}}}, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (19)$$

Voltage mag. equality @ **boundary(ies)**

$$\theta_{\beta^{\mathcal{T}}} = \theta^{a}_{\beta^{\mathcal{D}}}, \ \forall(\beta^{\mathcal{T}}, \beta^{\mathcal{D}}) \in \Lambda \quad (20)$$

$$\theta^{b}_{\beta^{\mathcal{D}}} = (\theta^{a}_{\beta^{\mathcal{D}}} - 120°), \ \forall \beta^{\mathcal{D}} \in N^{\mathcal{B}} \cap N^{\mathcal{D}} \quad (21)$$

$$\theta^{c}_{\beta^{\mathcal{D}}} = (\theta^{a}_{\beta^{\mathcal{D}}} + 120°), \ \forall \beta^{\mathcal{D}} \in N^{\mathcal{B}} \cap N^{\mathcal{D}} \quad (22)$$

Voltage angle equality/shift @ **boundary(ies)**

Integrated Transmission-Distribution (ITD) Formulation

Assumptions at the Boundary

- **Transmission** system is **balanced** (This may/may not be valid in reality) - 1Ø modeling
- **Distribution** system(s) are not balanced at the **substation** (See **(15)** and **(16)** summations)
  - **Vanilla** implementation
  - Additional **constraints** can be added to force **± X balance between** 3Ø

Integrated Transmission-Distribution (ITD) Formulations

- Built-in Formulations (Tested)
  - ACP-ACPU
    - Power-Voltage, polar coordinates, non-linear (NLP)
  - ACR-ACRU
    - Power-Voltage, rectangular coordinates, non-linear (NLP)
  - IVR-IVRU
    - Current-Voltage, rectangular coordinates, non-linear (NLP)
  - NFA-NFAU
    - Network Flow Approximation
    - Active power only, lossless, linear (LP)

- Other Formulations (Experimental)
  - ACR-FOTRU
    - Power-Voltage, rectangular coordinates, First-Order Taylor Approximation
  - ACP-FOTPU
    - Power-Voltage, polar coordinates, First-Order Taylor Approximation
  - ACR-FBSU
    - Power-Voltage, rectangular coordinates, Forward-Backward Sweep Approximation
  - SOCBFM-LinDist3Flow
    - Second Order Cone Branch Flow Model Relaxation – W-space.
    - Linear Approximation.
  - BFA-LinDist3Flow
    - Branch Flow Approximation

Integrated Transmission-Distribution (ITD): Problem Specification

Declarative modeling ➡ Rapid development / testing

Code Block 1 Problem specification for OPFITD

```
function build_opfitd(pmitd::AbstractPowerModelITD)
    pm_model = ... # Transmission model
    pmd_model = ... # Distribution model

    # PM(Transmission) Variables
    PM.variable_bus_voltage(pm_model)
    ...

    # PMD(Distribution) Variables
    PMD.variable_mc_bus_voltage(pmd_model)
    ...

    # PMITD (Boundary) Variables
    variable_boundary_power(pmitd)

    # --- PM(Transmission) Constraints ---
    PM.constraint_model_voltage(pm_model)
    ...

    # --- PMD(Distribution) Constraints ---
    PMD.constraint_mc_model_voltage(pmd_model)
    ...

    # --- PMITD-related Constraints -------
    for i in ids(pmitd, :boundary)
        constraint_boundary_power(pmitd, i)
        constraint_boundary_voltage_
                            magnitude(pmitd, i)
        constraint_boundary_voltage_
                            angle(pmitd, i)
    end

    # ---- Transmission Power Balance ---
    boundary_buses = Vector{Int64}()
    for i in PM.ids(pm_model, :bus)
        for j in ids(pmitd, :boundary)
            constraint_transmission_power_
                        balance_boundary(pmitd, i,
                                j, boundary_buses)
        end
        if !(i in boundary_buses)
            PM.constraint_power_
                            balance(pm_model, i)
        end
    end

    # ---- Distribution Power Balance ---
    for i in PMD.ids(pmd_model, :bus)
        for j in ids(pmitd, :boundary)
            constraint_distribution_power_
                    balance_boundary(pmitd, i,
                            j, boundary_buses)
        end
        if !(i in boundary_buses)
            PMD.constraint_mc_power_
                            balance(pmd_model, i)
        end
    end

    # --- PMITD Cost Functions --------
    objective_itd_min_fuel_cost(pmitd)

end
```

Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

Using PowerModelsITD.jl

The **files** needed to run **OPFITD** are:

Transmission file

```
function mpc = case5
mpc.version = '2';
mpc.baseMVA = 100.0;

%% bus data
%       bus_i   type    Pd      Qd      Gs      Bs      area    Vm      Va      baseKV  zone
mpc.bus = [
        1       2       0.0     0.0     0.0     0.0     1       1.07762         2.80377
        2       1       300.0   98.61   0.0     0.0     1       1.08407         -0.73465
        3       2       300.0   98.61   0.0     0.0     1       1.10000         -0.55972
        4       3       390.0   131.47  0.0     0.0     1       1.06414         0.00000
        5       1       8.0     1.2     0.0     0.0     1       1.00000         0.00000
        10      2       0.0     0.0     0.0     0.0     1       1.06907         3.59033
];

%% generator data
%       bus     Pg      Qg      Qmax    Qmin    Vg      mBase   status  Pmax    Pmin
mpc.gen = [
        1       40.0    30.0    30.0    -30.0   1.07762         100.0   1       40.0    0.0;
        1       170.0   127.5   127.5   -127.5  1.07762         100.0   1       170.0   0.0;
        3       324.498         390.0   390.0   -390.0  1.1     100.0   1       520.0   0.0;
        4       0.0     -10.802         150.0   -150.0  1.06414         100.0   1       200.0
        10      470.694         -165.039        450.0   -450.0  1.06907         100.0   1
];

%% generator cost data
%       2       startup shutdown        n       c(n-1)  ...     c0
mpc.gencost = [
        2       0.0     0.0     3       0.000000        14.000000       0.000000        2.000
        2       0.0     0.0     3       0.000000        15.000000       0.000000        2.000
        2       0.0     0.0     3       0.000000        30.000000       0.000000        2.000
        2       0.0     0.0     3       0.000000        40.000000       0.000000        2.000
        2       0.0     0.0     3       0.000000        10.000000       0.000000        2.000
];

%% branch data
%       fbus    tbus    r       x       b       rateA   rateB   rateC   ratio   angle   status
mpc.branch = [
        1       2       0.00281         0.0281  0.00712         400.0   400.0   400.0   0.0
        1       4       0.00304         0.0304  0.00658         426     426     426     0.0
        1       10      0.00064         0.0064  0.03126         426     426     426     0.0
        2       3       0.00108         0.0108  0.01852         426     426     426     0.0
        3       4       0.00297         0.0297  0.00674         426     426     426     1.05
        4       10      0.00297         0.0297  0.00674         240.0   240.0   240.0   0.0
        2       5       0.00297         0.0297  0.00674         426     426     426     0.0
];
```

MATPOWER (".m")

PSS(R)E v33 specification (".raw")

(support PowerWorld for PSSE conversions)

Distribution file(s)

```
New Circuit.3bus_bal
!  define a really stiff source
~ basekv=230 pu=1.00 MVAsc3=200000 MVAsc1=210000

! Substation Transformer
New Transformer.SubXF Phases=3 Windings=2 Xhl=0.01
~ wdg=1 bus=sourcebus conn=wye kv=230   kva=25000    %r=0.0005
~ wdg=2 bus=Substation   conn=wye kv=13.8  kva=25000   %r=0.0005

!Define Linecodes
New linecode.556MCM nphases=3 basefreq=60  ! ohms per 5 mile
~ rmatrix = ( 0.1000 | 0.0400    0.1000 |  0.0400    0.0400    0.1000)
~ xmatrix = ( 0.0583 |  0.0233    0.0583 | 0.0233    0.0233    0.0583)
~ cmatrix = (50.92958178940651  | -0  50.92958178940651 | -0 -0 50.92958178940651  ) ! small cap

New linecode.4/0QUAD nphases=3 basefreq=60  ! ohms per 100ft
~ rmatrix = ( 0.1167 | 0.0467    0.1167 | 0.0467    0.0467    0.1167)
~ xmatrix = (0.0667  |  0.0267    0.0667  |  0.0267    0.0267    0.0667 )
~ cmatrix = (50.92958178940651  | -0  50.92958178940651 | -0 -0 50.92958178940651  )  ! small ca

!Define lines
New Line.OHLine  bus1=Substation.1.2.3  Primary.1.2.3  linecode = 556MCM   length=1 normamps=600
New Line.Quad    Bus1=Primary.1.2.3  loadbus.1.2.3  linecode = 4/0QUAD  length=1 normamps=6000 e

!Loads - single phase
New Load.L1 phases=1  loadbus.1.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1
New Load.L2 phases=1  loadbus.2.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1
New Load.L3 phases=1  loadbus.3.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1

!GENERATORS DEFINITIONS
New generator.gen1 Bus1=loadbus.1.2.3 Phases=3 kV=( 13.8 3 sqrt / )  kW=2000 pf=1 conn=wye Model

Set VoltageBases = "230,13.8"
Set tolerance=0.000001
set defaultbasefreq=60
```

```
3178940651  ) ! small cap
)
3178940651  )  ! small ca
!   length=1 normamps=600
 length=1 normamps=6000 e

!Loads - single phase
New Load.L1 phases=1  loadbus.1.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1
New Load.L2 phases=1  loadbus.2.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1
New Load.L3 phases=1  loadbus.3.0   ( 13.8 3 sqrt / )   kW=3000   kvar=1500  model=1

!GENERATORS DEFINITIONS
New generator.gen1 Bus1=loadbus.1.2.3 Phases=3 kV=( 13.8 3 sqrt / )  kW=2000 pf=1 conn=wye Model

Set VoltageBases = "230,13.8"
Set tolerance=0.000001
set defaultbasefreq=60
```

OpenDSS (".dss")

Boundary file

```
[
    {
    "transmission_boundary": "5",
    "distribution_boundary": "3bus_unbal.voltage_source.source"
    },

    {
    "transmission_boundary": "6",
    "distribution_boundary": "3bus_bal.voltage_source.source"
    }
]
```

JSON (".json")

other proprietary file formats supported via **DiTTo [25]**

https://lanl-ansi.github.io/PowerModelsITD.jl/stable/manual/fileformat.html

[25] "DiTTo (Distribution Transformation Tool)," 2021, Accessed: Aug. 06, 2021. [Online]. Available: https://github.com/NREL/ditto

Using PowerModelsITD.jl

Simple User Interface → Easy User Adoption

Case w/ 1 distro. system

```julia
using PowerModelsITD
import Ipopt
ipopt = Ipopt.Optimizer

# Path for the files
pmitd_path = joinpath(dirname(pathof(PowerModelsITD)), "..")

# Files
pm_file = joinpath(pmitd_path, "test/data/transmission/case5_withload.m")
pmd_file = joinpath(pmitd_path, "test/data/distribution/case3_balanced.dss")
boundary_file = joinpath(pmitd_path, "test/data/json/case5_case3_bal.json")

pmitd_type = NLPowerModelITD{ACPPowerModel, ACPUPowerModel}

result = solve_opfitd(pm_file, pmd_files, boundary_file, pmitd_type, ipopt)
```

Case w/ 2 distro. systems

```julia
using PowerModelsITD
import Ipopt
ipopt = Ipopt.Optimizer

# Path for the files
pmitd_path = joinpath(dirname(pathof(PowerModelsITD)), "..")

# Files
pm_file = joinpath(pmitd_path, "test/data/transmission/case5_with2loads.m")
pmd_file1 = joinpath(pmitd_path, "test/data/distribution/case3_unbalanced.dss")
pmd_file2 = joinpath(pmitd_path, "test/data/distribution/case3_balanced.dss")
boundary_file = joinpath(pmitd_path, "test/data/json/case5_case3x2_unbal_bal.json")

pmd_files = [pmd_file1, pmd_file2] # vector of files
pmitd_type = NLPowerModelITD{ACPPowerModel, ACPUPowerModel}

result = solve_opfitd(pm_file, pmd_files, boundary_file, pmitd_type, ipopt)
```

Using PowerModelsITD.jl

Results

```
julia> result
Dict{String, Any} with 8 entries:
  "solve_time"         => 0.12712
  "optimizer"          => "Ipopt"
  "termination_status" => LOCALLY_SOLVED
  "dual_status"        => FEASIBLE_POINT
  "primal_status"      => FEASIBLE_POINT
  "objective"          => 18146.3
  "solution"           => Dict{String, Any}("multiinfrastructure"=>true, "it"=>Dict{String, Any}("pmd…
  "objective_lb"       => -Inf
```

Transmission

```
julia> result["solution"]["it"]["pm"]
Dict{String, Any} with 6 entries:
  "baseMVA"      => 100.0
  "branch"       => Dict{String, Any}("3"=>Dict{String, Any}("qf"=>206.656, "qt"=>-202.276, "pt"=>221.006, "pf"=>-220.308), "4"=>Dict{String, Any}("qf"=>-217.108, "qt"=>221.882, "pt"=>79.0383, "pf"=>-78.3924), "1"=…
  "gen"          => Dict{String, Any}("4"=>Dict{String, Any}("qg"=>56.3262, "pg"=>18.0328), "1"=>Dict{String, Any}("qg"=>30.0, "pg"=>40.0), "5"=>Dict{String, Any}("qg"=>-201.205, "pg"=>461.003), "2"=>Dict{String, A…
  "multinetwork" => false
  "bus"          => Dict{String, Any}("4"=>Dict{String, Any}("va"=>-1.06955e-34, "vm"=>0.9), "1"=>Dict{String, Any}("va"=>3.95367, "vm"=>0.917681), "5"=>Dict{String, Any}("va"=>-0.949629, "vm"=>0.937736), "2"=>Dict…
  "per_unit"     => false
```

Distribution

```
julia> result["solution"]["it"]["pmd"]
Dict{String, Any} with 7 entries:
  "line"        => Dict{String, Any}("3bus_unbal.quad"=>Dict{String, Any}("qf"=>[1344.85, 1503.97, 1502.46], "qt"=>[-1333.33, -1500.0, -1500.0], "pt"=>[-3333.33, -2333.33, -2333.33], "pf"=>[3351.62, 2340.39, 2344.9…
  "settings"    => Dict{String, Any}("sbase"=>100000.0)
  "transformer" => Dict{String, Any}("3bus_bal.subxf"=>Dict{String, Any}("q"=>[[1508.51, 1508.51, 1508.51], [-1508.41, -1508.41, -1508.41]], "p"=>[[2351.59, 2351.59, 2351.59], [-2351.58, -2351.58, -2351.58]]), "3bu…
  "generator"   => Dict{String, Any}("3bus_unbal.gen1"=>Dict{String, Any}("qg_bus"=>[-0.0, -0.0, -0.0], "qg"=>[-0.0, -0.0, -0.0], "pg"=>[666.668, 666.668, 666.668], "pg_bus"=>[666.668, 666.668, 666.668]), "3bus_bal…
  "load"        => Dict{String, Any}("3bus_unbal.l2"=>Dict{String, Any}("qd_bus"=>[1500.0], "pd_bus"=>[3000.0], "qd"=>[1500.0], "pd"=>[3000.0]), "3bus_bal.l3"=>Dict{String, Any}("qd_bus"=>[1500.0], "pd_bus"=>[3000.…
  "bus"         => Dict{String, Any}("3bus_unbal.loadbus"=>Dict{String, Any}("va"=>[-1.10106, -120.971, 119.172], "vm"=>[7.38801, 7.42776, 7.41273]), "3bus_bal.substation"=>Dict{String, Any}("va"=>[-1.08179, -121.0…
  "per_unit"    => false
```

Boundary

```
julia> result["solution"]["it"]["pmitd"]["boundary"]
Dict{String, Any} with 4 entries:
  "(100001, 5, voltage_source.3bus_unbal.source)" => Dict{String, Any}("pbound_fr"=>[8068.8], "qbound_fr"=>[4367.42])
  "(100001, voltage_source.3bus_unbal.source, 5)" => Dict{String, Any}("pbound_to"=>[-3367.36, -2346.47, -2354.97], "qbound_to"=>[-1355.14, -1507.53, -1504.75])
  "(100002, voltage_source.3bus_bal.source, 6)"   => Dict{String, Any}("pbound_to"=>[-2351.62, -2351.62, -2351.62], "qbound_to"=>[-1508.64, -1508.64, -1508.64])
  "(100002, 6, voltage_source.3bus_bal.source)"   => Dict{String, Any}("pbound_fr"=>[7054.87], "qbound_fr"=>[4525.93])
```

Using PowerModelsITD.jl

Running Multinetwork (Time-series)

```julia
using PowerModelsITD
import Ipopt
ipopt = Ipopt.Optimizer

# Path for the files
pmitd_path = joinpath(dirname(pathof(PowerModelsITD)), "..")

# Files
pm_file = joinpath(pmitd_path, "test/data/transmission/case5_with2loads.m")
pmd_file = joinpath(pmitd_path, "test/data/distribution/case3_unbalanced_withoutgen_mn.dss")
boundary_file = joinpath(pmitd_path, "test/data/json/case5_case3x2.json")

pmd_files = [pmd_file, pmd_file] # vector of files
pmitd_type = NLPowerModelITD{ACPPowerModel, ACPUPowerModel}

result = solve_mn_opfitd(pm_file, pmd_files, boundary_file, pmitd_type, Ipopt.Optimizer; auto_rename=true)
```

```
!Loads - single phase
New Loadshape.ls1 pmult=(file=load_profile.csv)

New Load.L1 phases=1  loadbus.1.0  ( 13.8 3 sqrt / )  kW=4000  kvar=1333.33  model=1 daily=ls1
New Load.L2 phases=1  loadbus.2.0  ( 13.8 3 sqrt / )  kW=3000  kvar=1500  model=1 daily=ls1
New Load.L3 phases=1  loadbus.3.0  ( 13.8 3 sqrt / )  kW=3000  kvar=1500  model=1 daily=ls1
```

```
0.3
0.3
0.3
0.3
```

Solve multinetwork opfitd

Using PowerModelsITD.jl

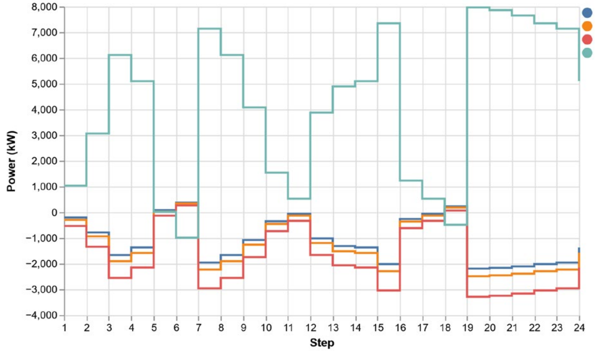
Results Multinetwork (Time-series)

Boundaries – all time steps (networks)

```
julia> result["solution"]["it"]["pmitd"]["nw"]
Dict{String, Any} with 4 entries:
  "4" => Dict{String, Any}("boundary"=>Dict{String, Any}("(100002, voltage_source.3bus_unbal_nogen_mn_2.source, 6)"=>Dict{String, Any}("pbound_to"=>[-1204.1, -901.81, -…
  "1" => Dict{String, Any}("boundary"=>Dict{String, Any}("(100002, voltage_source.3bus_unbal_nogen_mn_2.source, 6)"=>Dict{String, Any}("pbound_to"=>[-1204.1, -901.81, -…
  "2" => Dict{String, Any}("boundary"=>Dict{String, Any}("(100002, voltage_source.3bus_unbal_nogen_mn_2.source, 6)"=>Dict{String, Any}("pbound_to"=>[-1204.1, -901.81, -…
  "3" => Dict{String, Any}("boundary"=>Dict{String, Any}("(100002, voltage_source.3bus_unbal_nogen_mn_2.source, 6)"=>Dict{String, Any}("pbound_to"=>[-1204.1, -901.81, -…
```

Boundaries – time-step #3

```
julia> result["solution"]["it"]["pmitd"]["nw"]["3"]["boundary"]
Dict{String, Any} with 4 entries:
  "(100002, voltage_source.3bus_unbal_nogen_mn_2.source, 6)" => Dict{String, Any}("pbound_to"=>[-1204.1, -901.81, -902.678], "qbound_to"=>[-400.63, -449.178, -448.79])
  "(100002, 6, voltage_source.3bus_unbal_nogen_mn_2.source)" => Dict{String, Any}("pbound_fr"=>[3008.59], "qbound_fr"=>[1298.6])
  "(100001, 5, voltage_source.3bus_unbal_nogen_mn.source)"   => Dict{String, Any}("pbound_fr"=>[3008.59], "qbound_fr"=>[1298.59])
  "(100001, voltage_source.3bus_unbal_nogen_mn.source, 5)"   => Dict{String, Any}("pbound_to"=>[-1204.1, -901.808, -902.676], "qbound_to"=>[-400.626, -449.175, -448.787])
```



Boundary Power Flows
24 hours time-step example

Outline

- Background & Challenges
- Introduction to **PowerModelsITD.jl**
- Integrated Transmission-Distribution (ITD) OPF Problem Specification & Formulations
- Using **PowerModelsITD.jl**
- Experimental Test Cases

Experimental Test Cases

1. **Case5-Case3:**
   - PJM 5-bus system
     - New load bus #5
   - IEEE 4 Node Test Feeder
     - Connected at bus #5
     - 1 - 600 kW DG at bus #4
2. **Case118-Case3x5**
   - IEEE 118 Bus
   - x5 IEEE 4 Node Test Feeders
     - Connected at buses #2, #7, #14, #28, #44
     - Each one w/ 1 DG (different power ratings)
3. **Case500-Case30x5**
   - IEEE PGLib 500 bus
   - IEEE 30 bus system
     - Multiconductor (three-phase)
     - 1- 40 MW DG at bus #B2
4. **Case500-CaseLVx5**
   - IEEE PGLib 500 bus
   - IEEE LVTestCase (European Low-Voltage test feeder)
     - Each one w/ 3 DGs at buses #835, #539, and #619

	Transmission		Distribution		Total	
Test Case	—N—	—E—	—N—	—E—	—N—	—E—
Case 1	6	7	12	3	18	10
Case 2	118	186	12	3	178	201
Case 3	500	733	90	41	950	938
Case 4	500	733	2724	907	14120	5268

Total # of nodes & edges

*All feeders are **Kron-reduced** (any explicit neutral/ground removed)*

Experimental Test Cases

- **2 Scenarios**
  - **Independent:** Systems are optimized independently, i.e.,
    - **Step 1:** Distribution optimized assuming DSOs want to maximize DG usage, reserving 10% capacity for emergency.
    - **Step 2:** Transmission is optimized based on fixed load from Step 1.
  - **Integrated:**
    - Full use of DG is allowed due to full coordination of transmission and distribution systems.

Solved with **Ipopt** (open-source)

- MUMPS Linear solver
- Default configuration

Experimental Test Cases

Results

TABLE II
RESULTS FOR TEST CASE 1: CASE5-CASE3 WITH 1 DG

	Independent						ITD			Differences		
	PM			PMD			PMITD					
Formulation	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**
ACP-ACPU	17756.0373	0.0233	22	14.0401	0.0178	7	17770.0356	0.0727	26	0.0418	-0.0316	3
ACR-ACRU	17756.0373	0.0285	22	14.0401	0.0402	20	17770.0356	0.0652	26	0.0418	0.0036	16
IVR-IVRU	17756.0374	0.0252	20	14.0401	0.0502	22	17770.0357	0.0757	29	0.0418	-0.0003	13
NFA-NFAU	14534.2997	0.0066	14	14.0401	0.0064	7	14548.0998	0.0105	16	0.2400	0.0025	5

TABLE III
RESULTS FOR TEST CASE 2: CASE118-CASE3 ×5 DISTRIBUTION SYSTEMS WITH 1 DG.

	Independent						ITD			Differences		
	PM			PMD			PMITD					
Formulation	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**	**$/hr**	**Time(s)**	**Iterations**
ACP-ACPU	94571.9597	0.3401	27	64.7825	0.0817	35	94636.5198	0.5523	28	0.2224	-0.1304	34
ACR-ACRU	94571.9597	0.3138	28	64.7825	0.1780	98	94636.5198	0.5224	30	0.2224	-0.0305	96
IVR-IVRU	94571.9597	0.5835	32	64.7825	0.2306	110	94636.5199	0.9002	33	0.2223	-0.0861	109
NFA-NFAU	90893.1829	0.0180	15	64.7825	0.0250	35	90947.8941	0.0341	17	10.0712	0.0090	33

Experimental Test Cases

Results

TABLE IV
RESULTS FOR TEST CASE 3: CASE500-CASE30 ×5 DISTRIBUTION SYSTEMS WITH 1 DG.

	Independent						ITD			Differences		
	PM			PMD			PMITD					
Formulation	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations
ACP-ACPU	470979.6190	1.9290	42	180.0201	0.8338	65	469573.4476	5.1104	45	1586.1914	-2.3476	62
ACR-ACRU	470979.6191	1.8811	43	180.0201	3.0656	170	469573.4478	6.4275	55	1586.1914	-1.4808	158
IVR-IVRU	470979.6192	2.6622	47	180.0201	2.0838	170	469573.4479	8.4899	52	1586.1914	-3.7439	165
NFA-NFAU	450006.9826	0.1180	32	180.0201	0.0537	30	449297.2305	0.3989	34	889.7721	-0.2271	28

TABLE V
RESULTS FOR TEST CASE 4: CASE500-CASELV ×5 DISTRIBUTION SYSTEMS WITH 3 DG.

	Independent						ITD			Differences		
	PM			PMD			PMITD					
Formulation	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations	$/hr	Time(s)	Iterations
ACP-ACPU	451045.6962	1.8797	39	49.9517	15.2863	55	451093.0292	62.6413	50	2.6186	-45.4752	44
ACR-ACRU	451045.6963	1.6718	40	49.9517	31.7494	120	451093.0293	119.9938	92	2.6186	-86.5726	68
IVR-IVRU	451045.6961	2.7771	45	49.9517	39.9716	170	451093.0291	200.0407	140	2.6186	-157.2920	75
NFA-NFAU	436385.0531	0.1098	30	49.9517	0.4596	35	436434.5157	2.4760	31	0.4891	-1.9066	34

Challenges
(Currently not addressed by PowerModelsITD.jl)

1. Building realistic T&D datasets
    - Sufficiently large T&D networks
    - Realistic T&D networks
    - Lack of reliable large distribution systems datasets* (Open-source)

2. Adding support to other types of algorithms (e.g., decomposition-based)

***NREL**: Krishnan, V. K., Palmintier, B. S., Hodge, B. S., Hale, E. T., Elgindy, T., Bugbee, B., & Kadankodu, S. (2017). Smart-ds: Synthetic models for advanced, realistic testing: Distribution systems and scenarios (No. NREL/PR-5D00-68764). National Renewable Energy Lab.(NREL), Golden, CO (United States).

***PNNL**: Schneider, Kevin P., et al. *Modern grid initiative distribution taxonomy final report*. No. PNNL-18035. Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2008.

Future Work

1. Support new **ITD formulations**

    - Relaxations
    - Approximations
    - Hybrids

- Hybrid
    - ACR-FOTR (First-Order Taylor Rectangular)
    - ACP-FOTP (First-Order Taylor Polar)
    - ACR-FBS (Forward-Backward Sweep)
    - SOCBFM-LinDist3Flow
    - BFA-LinDist3Flow

2. Support **decomposition-based formulations** that allow:
    - Parallel computation of large-scale problems

3. Explore **applications** & **research (Collaborations)**
    - EVs/DERs integration and optimization Studies
    - Transformer Deferral Studies
    - Cybersecurity-related studies in T&D networks

Conclusions

1. Package designed to be a **complement** of ***PowerModels.jl*** and ***PowerModelsDistribution.jl***

2. Supports **diverse set** of **formulations enabling the co-optimization** of T&D networks.

3. Package designed to be a foundational tool that:
   - Enables the **speedy development** of **novel co-optimization formulations**
   - Improves the **state-of-the-art**

Thank you
Questions?

Contacts:

- Juan Ospina: jjospina@lanl.gov
- David M Fobes: dfobes@lanl.gov