ExaGO™ Users Manual
Version 1.0

Contents

Chapter 1

Introduction

Exascale Grid Optimization (ExaGO$^{\text{TM}}$) toolkit is an open source package for solving large-scale power grid optimization problems on parallel and distributed architectures, particularly targeted for exascale machines with heteregenous architectures (GPU). ExaGO$^{\text{TM}}$ is written in C/C++ and makes heavy use of the functionality provided by the PETSc[1] library. It uses RAJA [3]and Umpire [4]libraries for execution on the GPU and can make use of several optimization solvers - Ipopt [7], HiOp [6, 5], and TAO [1]- for solving the optimization problems.

All ExaGO$^{\text{TM}}$ applications use a nonlinear formulation based on full AC optimal power flow. The different applications available with ExaGO are listed in Table 1.1

Table 1.1: ExaGO applications

Application	Description	Notes
OPFLOW	AC optimal power flow	
SCOPFLOW	Security-constrained AC optimal power flow	Uses TCOPFLOW for multi-period contingencies
TCOPFLOW	Multi-period AC optimal power flow	
SOPFLOW	Stochastic AC optimal power flow	Uses SCOPFLOW for multi-contingency scenarios

While ExaGO is targeted for making use of distributed computing environments and GPUs, its full support is still under development. Table 1.2 lists the architecure support for ExaGO applications.

Table 1.2: ExaGO application execution on different hardware

Application	CPU (serial)	CPU (parallel)	GPU
OPFLOW	Y	x	Y
SCOPFLOW	Y	Y (embarrasingly parallel)	N
SOPFLOW	Y	Y (embarrasingly parallel)	N
TCOPFLOW	Y	N	N

Together, these application cater to problems spanning the dimensions of security (contingencies), stochasticity, and time as shown in Figure 1.1.
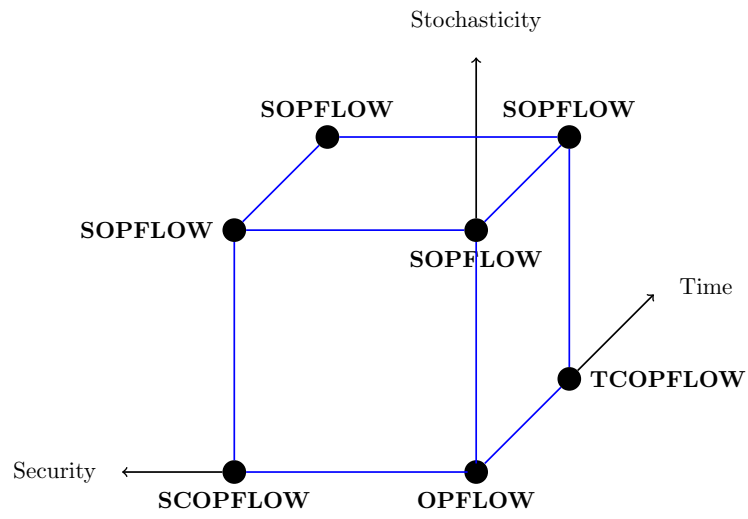
Figure 1.1: `ExaGO` provides applications along the dimensions of security (contingencies), time, and stochasticity. The label vertices denote different `ExaGO` applications available.

Chapter 2

Getting Started

2.1 System requirements

`ExaGO` is currently only built on 64b OSX and Linux machines, compiled with GCC >= 7.3. We build `ExaGO` on Intel and IBM Power9 architectures.

2.2 Prerequisites

This section assumes that you already have the `ExaGO` source code, and that the environment variable `EXAGODIR` is the directory of the `ExaGO` source code. `ExaGO` may be acquired via the PNNL git repository linked here, like so:

```
> git clone https://gitlab.pnnl.gov/exasgd/frameworks/exago.git exago
> export EXAGODIR=$PWD/exago
```

Paths to installations of third party software in examples are abbreviated with placeholder paths. For example, `/path/to/cuda` is a placeholder for a path to a valid `CUDA Toolkit` installation.

2.3 Dependencies

`ExaGO` has dependencies in table 2.1.

Table 2.1: Dependency Table

Dependency	Version Constraints	Mandatory	Notes
PETSc [1]	>= 3.13.0	✓	Only needed for the setup stage
CMake	>= 3.10	✓	Only a build dependency
MPI	>= 3.1.3		Only tested with `openmpi` and `spectrummpi`
Ipopt [7]	>= 3.12		
HiOp [6, 5]	>= 0.3.0		Prefer dynamically linked
RAJA [3]	>= 0.11.0		
Umpire [2]	>= 2.1.0		Only when RAJA is enabled
MAGMA	>= 2.5.2		Only when GPU acceleration is enabled
CUDA Toolkit	>= 10.2.89		Only when GPU acceleration is enabled

These may all be toggled via `CMake` which will be discussed in the section Building and installation.

2.3.1 Notes on environment modules

Many of the dependencies are available via environment modules on institutional clusters. To get additional information on your institution's clusters, please ask your institution's system administrators. Some end-to-end examples in this document will use system-specific modules and are not expected to expected to run on other clusters.

For example, the modules needed to build and run `ExaGO` on Newell, an IBM Power9 PNNL cluster, are as follows:

```
> module load gcc/7.4.0
> module load openmpi/3.1.5
> module load cuda/10.2
> module load magma/2.5.2_cuda10.2
> module load metis/5.1.0
> module load cmake/3.16.4
```

2.3.2 Additional Notes on GPU Accelerators

As of January 2021, `CUDA` is the only GPU accelerator platform `ExaGO` fully supports. We have preliminary support for `HIP`, however this is not in any of our main development branches yet. Full `HIP` support should arrive in early 2021.

2.3.3 Additional Notes on Umpire

`Umpire` is an implicit dependency of `RAJA`. If a user enables `RAJA`, they must also provide a valid installation of `Umpire`. Additionally, if a user would like to run `ExaGO` with `RAJA` and without `CUDA`, they must provide a CPU-only build of `Umpire` since an `Umpire` build with `CUDA` enabled will link against `CUDA`.

2.4 Building and installation

2.4.1 Default Build

`ExaGO` may be built with a standard `CMake` workflow:

Listing 2.1: Example `CMake` workflow

```
> cd $EXAGODIR
> export BUILDDIR=$PWD/build INSTALLDIR=$PWD/install
> mkdir $BUILDDIR $INSTALLDIR
> cd $BUILDDIR
> cmake .. -DCMAKE_INSTALL_PREFIX=$INSTALLDIR
> make install
```

Following sections will assume the user is following the basic workflow outlined above.
Note: For changes to the `CMake` configuration to take effect, the code wil have to be rebuilt.

2.4.2 Additional Options

To enable additional options, `CMake` variables may be defined via `CMake` command line arguments, `ccmake`, or `cmake-gui`. `CMake` options specific to `ExaGO` have an `EXAGO_` prefix. For example, the following shell commands will build `ExaGO` with `MPI`:

```
> cmake .. -DCMAKE_INSTALL_PREFIX=$INSTALLDIR -DEXAGO_ENABLE_MPI=ON
```

`ExaGO`'s `CMake` configuration will search the usual system locations for an `MPI` installation.

For dependencies not installed to a system-wide location, users may also directly specify the location of a dependency. For example, this will build `ExaGO` with `IPOPT` enabled and installed to a user directory:

```
> cmake .. \
    -DCMAKE_INSTALL_PREFIX=$INSTALLDIR \
    -DEXAGO_ENABLE_IPOPT=ON \
    -DIPOPT_DIR=/path/to/ipopt
```

Notice that the `CMake` variable `IPOPT_DIR` does not have an `EXAGO_` prefix. This is because the variables specifying locations often belong to external `CMake` modules. `CMake` variables indicating installation directories do not have an `EXAGO_` prefix.

Some `CMake` options effect others. This is especially common when the user enables `ExaGO`'s GPU options. For example, if the user enables `EXAGO_ENABLE_GPU` and `EXAGO_ENABLE_RAJA`, the user must provide a GPU-enabled `RAJA` installation. `Umpire` is also an implicit dependency of `RAJA`, so if the user enables `EXAGO_ENABLE_GPU` they must **also** provide a GPU-enabled `Umpire` installation.

Below is a complete shell session on PNNL's cluster Newell in which a more complicated `ExaGO` configuration is built, where each dependency installation is explicitly passed to `CMake`. Environment modules specific to Newell are provided to make the example thorough, even though they are not likely to work on another machine.

Listing 2.2: `ExaGO` build with all options enabled

```
> module load gcc/7.4.0
> module load cmake/3.16.4
> module load openmpi/3.1.5
> module load magma/2.5.2_cuda10.2
> module load metis/5.1.0
> module load cuda/10.2
> git clone https://gitlab.pnnl.gov/exasgd/frameworks/exago.git exago
> export EXAGODIR=$PWD/exago
> cd $EXAGODIR
> export BUILDDIR=$PWD/build INSTALLDIR=$PWD/install
> mkdir $BUILDDIR $INSTALLDIR
> cd $BUILDDIR
> cmake .. \
  -DCMAKE_INSTALL_PREFIX=$INSTALLDIR \
  -DCMAKE_BUILD_TYPE=Debug \
  -DEXAGO_ENABLE_GPU=ON \
  -DEXAGO_ENABLE_HIOP=ON \
  -DEXAGO_ENABLE_IPOPT=ON \
```

9

```
      -DEXAGO_ENABLE_MPI=ON \
      -DEXAGO_ENABLE_PETSC=ON \
      -DEXAGO_RUN_TESTS=ON \
      -DEXAGO_ENABLE_RAJA=ON \
      -DEXAGO_ENABLE_IPOPT=ON \
      -DIPOPT_DIR=/path/to/ipopt \
      -DRAJA_DIR=/path/to/raja \
      -Dumpire_DIR=/path/to/umpire \
      -DHIOP_DIR=/path/to/hiop \
      -DMAGMA_DIR=/path/to/magma \
      -DPETSC_DIR=/path/to/petsc
> make -j 8 install
> # For the following commands, a job scheduler command may be needed.
> # Run test suite
> make test
> # Run an ExaGO application:
> $INSTALLDIR/bin/opflow
```

2.5   Usage

Each `ExaGO` application has the following format for execution

```
  ./app <app_options>
```

Here, `app_options` are the command line options for the application. Each application has many options through which the input files and the control options can be set for the application. All application options have the form `-app_option_name` followed by the `app_option_value`. For instance,

```
  ./opflow -netfile case9mod.m -opflow_model POWER_BALANCE_POLAR \
  -opflow_solver IPOPT
```

will execute OPFLOW application using `case9mod.m` input file with the model `POWER_BALANCE_POLAR`
↪ and Ipopt [7]solver.

   Options can also be passed to each application through `-options_file`, or through a combination of the command line and options file. The configuration specified last on the command line overrides any previous options. For example, if `-options_file opflowoptions` specified `-netfile case9mod.m` within it's settings:

```
./opflow -netfile case118.m -options_file opflowoptions # Uses case9mod.m
./opflow -options_file opflowoptoins -netfile case118.m # Uses case118.m
```

If no options file is specified through the command line, ExaGO applications will attempt to locate the default options file for a given application in `.,./options,<install_dir>/options`. Each solver that can be used within `ExaGO` also have their own unique options. Some of these options are specified internally within `ExaGO`, but any other options that are needed should either be passed through the command line, or additionally specified in the options file.

Chapter 3

Optimal power flow (OPFLOW)

OPFLOW solves the full AC optimal power flow problem and provides various flexible features that can be toggled via run-time options. It has interfaces to different optimization solvers and includes different representations of the underlying equations (power-balance-polar and power-balance-cartesian) that can be used. By selecting the appropriate solver, OPFLOW can be executed on CPU (in serial or parallel) or on GPUs.

3.1   Formulation

Optimal power flow is a general nonlinear programming problem with the following form

$$\text{min. } f(x) \tag{3.1}$$
$$\text{s.t.}$$
$$g(x) = 0 \tag{3.2}$$
$$h(x) \leq 0 \tag{3.3}$$
$$x^{\min} \leq x \leq x^{\max} \tag{3.4}$$

Here, x are the decision variables with lower and upper bounds $x^{\min}$ and $x^{\max}$, respectively, $f(x)$ is the objective function, $g(x)$ and $h(x)$ are the equality and inequality constraints, respectively. In the following sections we describe what constitutes these different terms as used by OPFLOW.

3.1.1   Variables and bounds

The different variables used in OPFLOW formulation are described in Table 3.1.

Power imbalance variables are non-physical (slack) variables that measure the violation of power balance at buses. Having these variables (may) help in making the optimization problem easier to solve since they always ensure feasibility of the bus power balance constraints.

3.1.2   Objective Function

The objective function for OPFLOW is given in (3.5)

$$\text{min. } C(p^{\mathrm{g}}) + C(\Delta p^{\mathrm{g}}) + C(\Delta p^{\mathrm{l}}, \Delta q^{\mathrm{l}}) + C(\Delta p, \Delta q) \tag{3.5}$$

Table 3.1: Optimal power flow (OPFLOW) variables

Variable	Symbol	Bounds	Notes
Generator real power dispatch	p_j^{g}	$p_j^{\mathrm{gmin}} \leq p_j^{\mathrm{g}} \leq p_j^{\mathrm{gmax}}$	
Generator reactive power dispatch	q_j^{g}	$q_j^{\mathrm{gmin}} \leq q_j^{\mathrm{g}} \leq q_j^{\mathrm{gmax}}$	
Generator real power deviation	Δp_j^{g}	$-p_j^{\mathrm{r}} \leq \Delta p_j^{\mathrm{g}} \leq p_j^{\mathrm{r}}$	• Only used when `-opflow_has_gensetpoint` or `-opflow_use_agc` option is active • Δp_j^{g} is the deviation from real power generation set-point p_j^{gset}.
Generator real power set-point	p_j^{gset}	$p_j^{\mathrm{gmin}} \leq p_j^{\mathrm{gset}} \leq p_j^{\mathrm{gmax}}$	• Only used when `-opflow_has_gensetpoint` or `-opflow_use_agc` option is active.
System power excess/deficit	ΔP	Unbounded	Only used when `-opflow_use_agc` is active
Bus voltage angle	θ_i	$-\pi \leq \theta_i \leq \pi$	• Used with power balance polar model (`-opflow_model POWER_BALANCE_CARTESIAN`) • θ_i is unbounded, except reference bus angle θ_i^{ref} which is fixed to 0
Bus voltage magnitude	v_i	$v_i^{\mathrm{min}} \leq v_i \leq v_i^{\mathrm{max}}$	• Used with power balance polar model (`-opflow_model POWER_BALANCE_POLAR`) • $v_i^{\mathrm{min}} = v_i^{\mathrm{max}} = v_i^{\mathrm{set}}$ if fixed generator set point voltage option is active (`-opflow_has_gensetpoint`)
Bus voltage real part	e_i	$-v_i^{\mathrm{max}} \leq e_i \leq v_i^{\mathrm{max}}$	Used with power balance cartesian model (`-opflow_model POWER_BALANCE_CARTESIAN`)
Bus voltage imaginary part	f_i	$-v_i^{\mathrm{max}} \leq f_i \leq v_i^{\mathrm{max}}$	Used with power balance cartesian model (`-opflow_model POWER_BALANCE_CARTESIAN`)
Bus real power mismatch	Δp_i	Unbounded	Used when power mismatch variable option is active (`-opflow_include_powerimbalance_variables1`)
Bus reactive power mismatch	Δq_i	Unbounded	Used when power mismatch variable option is active (`-opflow_include_powerimbalance_variables1`)
Real power load loss	Δp_j^{l}	$0 \leq \Delta p_j^{\mathrm{l}} \leq p_j^{\mathrm{l}}$	Used when load loss variable option is active (`-opflow_include_loadloss_variables1`)
Reactive power load loss	Δq_j^{l}	$0 \leq \Delta q_j^{\mathrm{l}} \leq q_j^{\mathrm{l}}$	Used when load loss variable option is active (`-opflow_include_loadloss_variables1`)

Total generation cost $C(p^{\mathbf{g}})$

Needs option `-opflow_objective MIN_GEN_COST`

$$C(p^{\mathbf{g}}) = \sum_{j \in J^{\text{gen}}} C_j^{\text{g}}(p_j^{\text{g}})$$ (3.6)

Here, C_j^{g} is a quadratic function of the form $C_j^{\text{g}} = a_j^{\text{g}} p_j^{\text{g}2} + b_j^{\text{g}} p_j^{\text{g}} + c_j^{\text{g}}$.

Total generation setpoint deviation $C(\Delta p^{\mathbf{g}})$

Needs option `-opflow_objective MIN_GENSETPOINT_DEVIATION` option

$$C(\Delta p^{\mathbf{g}}) = \sum_{j \in J^{\text{gen}}} (\Delta p_j^{\text{gu}2})$$ (3.7)

Load loss $C(\Delta p^{\mathbf{l}}, \Delta q^{\mathbf{l}})$

This term gets added to the objective when `-opflow_include_loadloss_variables` option is active.

$$C(\Delta p^{\mathbf{l}}, \Delta q^{\mathbf{l}}) = \sum_{j \in J^{\text{ld}}} \sigma_j^{\text{l}}(\Delta p_j^{\text{l}^2} + \Delta q_j^{\text{l}^2})$$ (3.8)

The load loss penalty σ_j^{l} can be set via the option `-opflow_loadloss_penalty`. The default is $1000/MW for all loads.

Power imbalance $C(\Delta p, \Delta q)$

This term gets added to the objective when `-opflow_include_powerimbalance_variables` option is active.

$$C(\Delta p, \Delta q) = \sum_{i \in J^{bus}} \sigma_i(\Delta p_i^2 + \Delta q_i^2)$$ (3.9)

The power imbalance cost σ_i can be set via the option `-opflow_powerimbalance_penalty`. The default is $10,000/MW for all buses. Though the power imbalance variables $\Delta p_i, \Delta q_i$ are slack or non-physical, they can help in solving infeasible cases and provide a meaasure of the infeasibility.

3.1.3 Equality constraints

Nodal power balance

$$\sum_{\substack{j \in J^{\text{gen}} \\ A_{ij}^{\text{g}} \neq 0}} p_j^{\text{g}} = p_i^{\text{sh}} + \sum_{\substack{j \in J^{\text{ld}} \\ A_{ij}^{\text{l}} \neq 0}} p_j^{\text{l}} + \sum_{\substack{j \in J^{\text{br}} \\ A_{oi}^{\text{br}} \neq 0}} p_{j_{od}}^{\text{br}} + \sum_{\substack{j \in J^{\text{br}} \\ A_{id}^{\text{br}} \neq 0}} p_{j_{do}}^{\text{br}}$$ (3.10)

$$\sum_{\substack{j \in J^{\text{gen}} \\ A_{ij}^{\text{g}} \neq 0}} q_j^{\text{g}} = q_i^{\text{sh}} + \sum_{\substack{j \in J^{\text{ld}} \\ A_{ij}^{\text{l}} \neq 0}} q_j^{\text{l}} + \sum_{\substack{j \in J^{\text{br}} \\ A_{oi}^{\text{br}} \neq 0}} q_{j_{od}}^{\text{br}} + \sum_{\substack{j \in J^{\text{br}} \\ A_{id}^{\text{br}} \neq 0}} q_{j_{do}}^{\text{br}}$$ (3.11)

(3.12)

where, the real and reactive power shunt consumption is given by (3.19) and (3.20)

The real and reactive power flow $p_{j_{od}}^{\mathrm{br}}$, $q_{j_{od}}^{\mathrm{br}}$ for line j from the origin bus o to bus d is given by (3.21) – (3.22) and from destination bus d to origin bus o is given by (3.23) – (3.24)

Generator real power output

When using -opflow_has_gensetpoint, two extra variables p_j^{gset} and Δp_j^{g} are added for each generator. The generator real power output p_j^{g} is related to the power deviation Δp_j^{g} by the following relations

$$p_j^{\mathrm{gset}} + \Delta p_j^{\mathrm{g}} - p_j^{\mathrm{g}} = 0 \tag{3.13}$$

$$p_j^{\mathrm{gset}} - p_j^{\mathrm{g}*} = 0 \tag{3.14}$$

Here, $p_j^{\mathrm{g}*}$ is the set-point for the generator real power output, which can be thought of as an operator set or contractual agreement set-point.

3.1.4 Inequality constraints

MVA flow on branches

MVA flow limits at origin and destination buses for each line.

$$p_{j_{od}}^{\mathrm{br}\,2} + q_{j_{od}}^{\mathrm{br}\,2} \leq s_j^{\mathrm{rateA}\,2}, \quad j \in J^{\mathrm{br}} \tag{3.15}$$

$$p_{j_{do}}^{\mathrm{br}\,2} + q_{j_{do}}^{\mathrm{br}\,2} \leq s_j^{\mathrm{rateA}\,2}, \quad j \in J^{\mathrm{br}} \tag{3.16}$$

To reduce the size of inequality constraints, only lines that are in service and having MVA A rating s_j^{rateA} less than 10000 MVA are considered.

Automatic generation control (AGC)

With -opflow_use_agc, two additional constraints are added for each participating generator to enforce the proportional generator redispatch participation as done in automatic generation control (AGC). These two equations are

$$
\begin{aligned}
\left(\alpha_j^{\mathrm{g}} \Delta P - \Delta p_j^{\mathrm{g}} \right) \left(p_j^{\mathrm{g}} - p_j^{\mathrm{gmax}} \right) \geq 0 \\
\left((\Delta p_j^{\mathrm{g}} - \alpha_j \Delta P) \right) \left(p_j^{\mathrm{gmin}} - p_j^{\mathrm{g}} \right) \geq 0
\end{aligned}
\tag{3.17}
$$

Eq. 3.17 enforces the generator set-point deviation to be equal to the generation participation when the generator has head-room available $p_j^{\mathrm{gmin}} \leq p_j^{\mathrm{g}} \leq p_j^{\mathrm{gmax}}$. Here, α_j^{g} is the generator participation factor which is the proportion of the power deficit/excess ΔP that the generator provides.

Generator bus voltage control

When the option -opflow_genbusvoltage FIXED_WITHIN_QBOUNDS is used, the generator bus voltage is fixed when the total reactive power generation available at the bus is within bounds. When

it reaches its bounds, the voltage varies with the generator reactive power fixed at its bound. To implement this behavior, two inequality constraints are added for each generator bus

$$
\begin{aligned}
(v_i^{\text{set}} - v_i)(q_i - q^{\text{max}_i}) &\geq 0 \\
(v_i - v_i^{\text{set}})(q^{\text{min}_i} - q_i) &\geq 0
\end{aligned}
\tag{3.18}
$$

Here, q_i, q^{max_i}, and q^{min_i} are the generated, maximum, and minimum reactive power at the bus, respectively.

3.2 Models

Table 3.2: OPFLOW models

Model type	OPFLOW option	Compatible solvers	CPU-GPU
Power balance with polar coordinates	`-opflow_model POWER_BALANCE_POLAR`	IPOPT	CPU
Power balance with cartesian coordinates	`-opflow_model POWER_BALANCE_CARTESIAN`	IPOPT, TAO	CPU
Power balance with polar coordinates used with HIOP	`-opflow_model POWER_BALANCE_HIOP`	HiOp [6, 5]	CPU/GPU
Power balance with polar coordinates used with HIOP on GPU	`-opflow_model PBPOLRAJAHIOP`	IPOPT, TAO	GPU

3.2.1 Power balance polar

Shunt power

$$
p_i^{\text{sh}} = g_i^{\text{sh}} v_i^2 \tag{3.19}
$$

$$
q_i^{\text{sh}} = -b_i^{\text{sh}} v_i^2 \tag{3.20}
$$

Branch flows

The real and reactive power flow $p_{j_{od}}^{\text{br}}$, $q_{j_{od}}^{\text{br}}$ from bus o to bus d on line j is given by (3.21) $-$ (3.22)

$$
p_{j_{od}}^{\text{br}} = g_{oo} v_o^2 + v_o v_d (g_{od} \cos(\theta_o - \theta_d) + b_{od} \sin(\theta_o - \theta_d)) \tag{3.21}
$$

$$
q_{j_{od}}^{\text{br}} = -b_{oo} v_o^2 + v_o v_d (-b_{od} \cos(\theta_o - \theta_d) + g_{od} \sin(\theta_o - \theta_d)) \tag{3.22}
$$

and from bus i to bus d is given by (3.23) $-$ (3.24)

$$
p_{j_{do}}^{\text{br}} = g_{dd} v_d^2 + v_d v_o (g_{do} \cos(\theta_d - \theta_o) + b_{do} \sin(\theta_d - \theta_o)) \tag{3.23}
$$

$$
q_{j_{do}}^{\text{br}} = -b_{dd} v_d^2 + v_d v_o (-b_{do} \cos(\theta_d - \theta_o) + g_{do} \sin(\theta_d - \theta_o)) \tag{3.24}
$$

15

Table 3.3: OPFLOW Model-solver compatibility

Model	Ipopt [7]	HiOp [6, 5]	TAO [1]
POWER_BALANCE_POLAR	✓		
POWER_BALANCE_CARTESIAN	✓		✓
POWER_BALANCE_HIOP		✓	
PBPOLRAJAHIOP		✓	

3.2.2 Power balance with HiOp

Power balance formulation with polar coordinates used with HiOp [6, 5][6] library. This model runs on CPU only, though the HiOp [6, 5]solver can run on GPU.

3.2.3 Power balance with HiOp,RAJA,and Umpire

Power balance formulation with polar coordinates used with HiOp [6, 5]and RAJA [3]. This model uses RAJA [3] [3] and Umpire [4] libraries to run OPFLOW calculations (objective, constraints, etc.) on the GPU.

3.3 Solvers

OPFLOW can be used with a few different solvers. All the three solvers solve the optimization problem via a nonlinear interior-point algorithm.

1. Ipopt [7] is a popular open-source package for solving nonlinear optimization problems. It is the most robust of the solvers implemented for solving OPFLOW. However, it can be run only on a single process and does not have GPU support. Option:
   `-opflow_solver IPOPT`

2. The Toolkit for Applied Optimization (TAO [1]) is a toolkit available through PETSc [1] for solving optimization problems. TAO includes a interior-point algorithm that is used by OPFLOW. The TAO algorithm is fully parallel, but is less robust compared to IPOPT. Currently, the TAO solver can only used with the power balance cartesian model. Option:
   `-opflow_solver TAO` -opflow_model POWER_BALANCE_CARTESIAN

3. HiOp [6, 5] is a high-performance optimization library that implements an interior-point algorithm for solving nonlinear optimization problems. The library supports execution both on the CPU and the GPU. Options:
   CPU: `-opflow_solver HIOP` -opflow_model POWER_BALANCE_HIOP
   GPU: `-opflow_solver HIOP` -opflow_model PBPOLRAJAHIOP
   The option `-hiop_ipopt_debug` can be used with the CPU option in order to use an adapter that verifies the correctness of the first and second order derivates when compared to Ipopt [7]. This may be useful when debugging experimental changes.

3.4 Input and Output

The current `ExaGO` version only supports reading network file in MATPOWER format and can (optionally) write the output back in MATPOWER data file format.

3.5 Usage

```
./opflow -netfile <netfilename>  <opflowoptions>
```

3.6 Options

See table 3.4

Table 3.4: OPFLOW options

Option	Meaning	Values (Default value)
-netfile	Network file name	string ¡ 4096 characters (case9mod.m)
-print_output	Print output to screen	0 or 1 (0)
-save_output	Save output to file	0 or 1 (0)
-opflow_model	Representation of network balance equations and bus voltages	See Table 3.2 (POWER_BALANCE_POLAR)
-opflow_solver	Optimization solver	See section 3.3
-opflow_initialization	Type of initialization	See Table 3.5 (MIDPOINT)
-opflow_has_gensetpoint	Uses generation set point and activates ramping variables	0 or 1 (0)
-opflow_use_agc	Uses AGC formulation in OPF	0 or 1 (0)
-opflow_objective	type of objective function	See table 3.7 (MIN_GEN_COST)
-opflow_genbusvoltage	Type of generator bus voltage control	See Table 3.6 (FIXED_WITHIN_QBOUNDS)
-opflow_ignore_lineflow_constraints	Ignore line flow constraints	0 or 1 (0)
-opflow_include_loadloss_variables	Include load loss	0 or 1 (0)
-opflow_include_powerimbalance_variables	Include power imbalance	0 or 1 (0)
-opflow_loadloss_penality	$ penalty for load loss	real (1000)
-opflow_powerimbalance_penalty	$ penalty for power imbalance	real (10000)
-opflow_tolerance	Optimization solver tolerance	real (1e-6)

Table 3.5: OPFLOW initializations

Initialization type	Meaning
MIDPOINT	Use mid-point of bounds
FROMFILE	Use values from network file
ACPF	Run AC power flow for initialization
FLATSTART	Flat-start

Table 3.6: OPFLOW generator bus voltage control modes

Voltage control type	Meaning
FIXED_AT_SETPOINT	Fixed at given set-point. Reactive power limits are ignored
FIXED_WITHIN_QBOUNDS	Fixed within reactive power bounds
VARIABLE_WITHIN_BOUNDS	Variable within voltage bounds

17

Table 3.7: OPFLOW objective function types

Objective function	Meaning
MIN_GEN_COST	Minimize generation cost
MIN_GENSETPOINT_DEVIATION	Minimize deviation (ramp up-down) from generataor set-point
NO_OBJ	No objecive function

3.7   Examples

Some OPFLOW example runs are provided with some sample output. Options values are the default values in table 3.4 unless otherwise specified. -print_output is only used in the first example to save space. Sample output is generated from running examples from the installation directory.

Example using the TAO [1]solver:

```
$ ./bin/opflow -netfile case9mod.m -opflow_solver TAO -opflow_model \
POWER_BALANCE_CARTESIAN -print_output
[ExaGO INFO]: -- Checking ... -options_file not passed    exists: no
[ExaGO INFO]: Creating OPFlow
TAO solve converged due to CONVERGED_GRTOL iterations 12
==============================================================
                Optimal Power Flow
==============================================================
Model                         POWER_BALANCE_CARTESIAN
Solver                        TAO
Objective                     MIN_GEN_COST
Initialization                MIDPOINT
Load loss allowed             NO
Power imbalance allowed       NO
Ignore line flow constraints  NO

Number of variables           24
Number of equality constraints    19
Number of inequality constraints  27

Convergence status               CONVERGED
Objective value                  4144.46


-----------------------------------------------------------------
Bus       Pd      Qd      Vm      Va       mult_Pmis      mult_Qmis
-----------------------------------------------------------------
1        0.00    0.00   1.100  -0.000     2102.91         0.00
2        0.00    0.00   1.095   3.927     2059.18         0.00
3        0.00    0.00   1.087   2.120     2065.15        -0.00
4        0.00    0.00   1.097  -1.993     2103.16         0.08
5       75.00   50.00   1.079  -3.060     2113.46         7.29
6       90.00   30.00   1.087  -3.927     2129.85         1.62
7        0.00    0.00   1.100   0.535     2059.57        -0.04
8      100.00   35.00   1.089  -1.720     2079.34         2.99
```

```
9         0.00    0.00   1.100   -0.135      2065.43       -0.09

|--------------------------------------------------------------------
|From       To      Status    Sft     Stf     Slim    mult_Sf  mult_St
|--------------------------------------------------------------------
|1          4         1      73.17   72.98   380.00    0.00     0.00
|2          7         1     114.18  114.68   250.00    0.00     0.00
|3          9         1      83.57   84.60   300.00    0.00     0.00
|4          5         1      29.68   40.50   250.00    0.00     0.00
|4          6         1      44.86   46.03   250.00    0.00     0.00
|5          7         1      51.29   49.04   250.00    0.00     0.00
|6          9         1      48.94   51.43   150.00    0.00     0.00
|7          8         1      66.61   68.11   250.00    0.00     0.00
|8          9         1      38.87   34.15   150.00    0.00     0.00

|--------------------------------------------------------------------
|Gen  Status   Fuel      Pg      Qg     Pmin    Pmax     Qmin    Qmax
|--------------------------------------------------------------------
|1     1    UNDEFINED   72.86    6.79   10.00  350.00  -300.00  300.00
|2     1    UNDEFINED  114.07   -5.13   10.00  300.00  -300.00  300.00
|3     1    UNDEFINED   80.21  -23.47   10.00  270.00  -300.00  300.00
|[ExaGO INFO]: Finalizing opflow application.
```

Example using the Ipopt [7] solver:

```
$ ./bin/opflow -opflow_solver IPOPT -opflow_model POWER_BALANCE_POLAR \
-opflow_initialisation ACPF -opflow_objective MIN_GENSETPOINT_DEVIATION
[ExaGO INFO]: -- Checking ... -options_file not passed     exists: no
[ExaGO INFO]: Creating OPFlow

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the
Eclipse Public License (EPL).
******************************************************************************

This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      125
Number of nonzeros in inequality constraint Jacobian.:      106
Number of nonzeros in Lagrangian Hessian.............:      159

Total number of variables............................:       32
                    variables with only lower bounds:        0
               variables with lower and upper bounds:       24
                    variables with only upper bounds:        0
Total number of equality constraints.................:       24
Total number of inequality constraints...............:       36
```

19

```
         inequality constraints with only lower bounds:       15
     inequality constraints with lower and upper bounds:       21
         inequality constraints with only upper bounds:        0

iter objective  inf_pr inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  5.9999880e-04 1.80e+00 2.60e-01 -1.0 0.00e+00 - 0.00e+00 0.00e+00   0
1  1.1156839e-02 1.31e+00 7.65e+01 -1.0 1.25e+00 - 9.33e-02 2.67e-01h 1
2  1.9076073e-02 1.26e+00 7.36e+01 -1.0 1.84e+00 - 3.08e-02 4.25e-02f 1
3  4.6334972e-02 1.17e+00 6.85e+01 -1.0 1.20e+00 - 1.51e-02 7.14e-02f 1
...
86  8.8262569e-02 3.01e-12 2.18e-03 -7.0 2.12e-05 - 1.00e+00 1.00e+00h 1
87  8.8262567e-02 3.55e-15 1.33e-10 -7.0 2.54e-09 - 1.00e+00 1.00e+00h 1

Number of Iterations....: 87

                              (scaled)                   (unscaled)
Objective............:  8.8262566709882084e-02   8.8262566709882084e-02
Dual infeasibility..:  1.3274141855146966e-10   1.3274141855146966e-10
Constraint violation:  3.5527136788005009e-15   3.5527136788005009e-15
Complementarity.....:  9.0929465340161350e-08   9.0929465340161350e-08
Overall NLP error...:  9.0929465340161350e-08   9.0929465340161350e-08


Number of objective function evaluations         = 120
Number of objective gradient evaluations         = 88
Number of equality constraint evaluations        = 120
Number of inequality constraint evaluations      = 120
Number of equality constraint Jacobian evaluations  = 88
Number of inequality constraint Jacobian evaluations = 88
Number of Lagrangian Hessian evaluations         = 87
Total CPU secs in IPOPT (w/o function evaluations) =      0.039
Total CPU secs in NLP function evaluations        =      0.011

EXIT: Optimal Solution Found.
[ExaGO INFO]: Finalizing opflow application.
```

Example using the HiOp [6, 5]solver on the CPU:

```
$ ./bin/opflow -opflow_solver HIOP -opflow_model POWER_BALANCE_HIOP \
-opflow_initialisaition FLATSTART -hiop_compute_mode CPU \
-hiop_verbosity_level 3
[ExaGO INFO]: -- Checking ... -options_file not passed    exists: no
[ExaGO INFO]: Creating OPFlow
[Warning] Detected 1 fixed variables out of a total of 24.
[Warning] Fixed variables will be relaxed internally.
===============
Hiop SOLVER
===============
```

```
#
# Hiop options
#
...
# end of Hiop options

Using 1 MPI ranks.
---------------
Problem Summary
---------------
Total number of variables: 24
      lower/upper/lower_and_upper bounds: 16 / 16 / 16
Total number of equality constraints: 18
Total number of inequality constraints: 18
      lower/upper/lower_and_upper bounds: 18 / 18 / 18
iter  objective   inf_pr    inf_du   lg(mu)   alpha_du   alpha_pr linesrch
0  1.0318125e+04 1.800e+00  4.460e+03  -1.00  0.000e+00  0.000e+00  -(-)
1  7.7139414e+03 1.167e+00  2.900e+03  -1.00  6.255e-01  3.496e-01  1(s)
2  7.3644945e+03 1.075e+00  2.654e+03  -1.00  9.785e-03  8.498e-02  1(s)
...
13   4.1444605e+03 3.989e-10  3.273e-08  -7.00  1.000e+00  1.000e+00  1(h)
Successfull termination.
Total time 0.003 sec
Hiop internal time:     total 0.001 sec     avg iter 0.000 sec
    internal total std dev across ranks 0.000 percent
Fcn/deriv time:total=0.001sec (obj=0.000 grad=0.000 cons=0.000 \
                                Jac=0.000 Hess=0.001)
    Fcn/deriv total std dev across ranks 0.000 percent
Fcn/deriv #: obj 14 grad 14 eq cons 14 ineq cons 14 eq Jac 14 ineq Jac 14
Total KKT time 0.001 sec
        update init 0.000sec   update linsys 0.000 sec     fact 0.001 sec
        solve rhs-manip 0.000 sec     triangular solve 0.000 sec

[ExaGO INFO]: Finalizing opflow application.
```

Example using HiOp [6, 5]solver on the GPU:

```
$mpirun -n 1 ./bin/opflow -opflow_model PBPOLRAJAHIOP -opflow_solver HIOP\
-hiop_compute_mode GPU -hiop_verbosity_level 3 \
-opflow_initialisation ACPF -opflow_ignore_lineflow_constraints 1 \
-netfile datafiles/case_ACTIVSg200.m
[ExaGO INFO]: -- Checking ... -options_file not passed       exists: no
[ExaGO INFO]: Creating OPFlow
[Warning] Detected 1 fixed variables out of a total of 476.
[Warning] Fixed variables will be relaxed internally.
==============
Hiop SOLVER
==============
```

```
#
# Hiop options
#
...
# end of Hiop options

Using 1 MPI ranks.
---------------
Problem Summary
---------------
Total number of variables: 476
     lower/upper/lower_and_upper bounds: 277 / 277 / 277
Total number of equality constraints: 400
Total number of inequality constraints: 0
     lower/upper/lower_and_upper bounds: 0 / 0 / 0
iter  objective   inf_pr   inf_du   lg(mu)  alpha_du   alpha_pr linesrch
0  2.6329021e+04 1.725e+00  2.321e+03  -1.00  0.000e+00  0.000e+00  -(-)
1  2.6227762e+04 1.723e+00  2.318e+03  -1.00  8.506e-04  1.026e-03  1(s)
2  2.6134620e+04 1.721e+00  2.314e+03  -1.00  1.697e-03  8.630e-04  1(s)
...
24  2.7557571e+04 3.921e-09  3.801e-07  -7.00  1.000e+00  1.000e+00  1(h)
Successfull termination.
Total time 4.555 sec
Hiop internal time:     total 4.538 sec      avg iter 0.189 sec
     internal total std dev across ranks 0.000 percent
Fcn/deriv time:     total=0.015 sec  ( obj=0.001 grad=0.001 cons=0.003 \
                                       Jac=0.004 Hess=0.006)
     Fcn/deriv total std dev across ranks 0.000 percent
Fcn/deriv #: obj 27 grad 25 eq cons 27 ineq cons 27 eq Jac 25 ineq Jac 25
Total KKT time 4.464 sec
         update init 0.004sec   update linsys 0.046 sec     fact 4.327 sec
         solve rhs-manip 0.072 sec     triangular solve 0.015 sec

[ExaGO INFO]: Finalizing opflow application.
```

Chapter 4

Multi-period optimal power flow (TCOPFLOW)

TCOPFLOW solves a full AC multi-period optimal power flow problem with the objective of minimizing the total cost over the given time horizon while adhering to intra-time and inter-temporal constraints.

4.1 Formulation

The multi-period optimal power flow problem is a series of optimal power flow problems coupled via temporal constraints. The generator real power deviation $(p_{jt}^{\text{g}} - p_{jt-\Delta t}^{\text{g}})$ constrained within the ramp limits form the temporal constraints. An illustration of the temporal constraints is shown in Fig. 4.1 with four time steps. Each time-step t is coupled with its preceding time $t - \Delta t$, where Δt is the time-step where the objective is to find a least cost dispatch for the given time horizon.
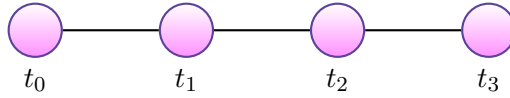
Figure 4.1: Multi-period optimal power flow example with four time-steps. The lines connecting the different time-periods denote the coupling between them.

In general form, the equations for multi-period optimal power flow are given by (4.1) – (4.5). TCOPFLOW solves to minimize the total generation cost $\sum_{t=0}^{N_t-1} f(x_t)$ over the time horizon, where N_t is the number of time-steps. At each time-step, the equality constraints $(g(x_t))$, inequality $h(x_t)$, and the lower/upper limit (x^-, x^+) constraints need to be satisfied. Equation (4.5) represents the coupling between the consecutive time-steps. It is a most common form of coupling that limits the deviation of the real power generation at time t from its preceding time-step $t - \Delta t$ to within its ramping capability $\delta_t x$.

$$\min \sum_{t=0}^{N_t-1} f(x_t) \tag{4.1}$$

$$\text{s.t.}$$

$$g(x_t) = 0, \qquad\qquad\qquad t \in [0, N_t - 1] \tag{4.2}$$

$$h(x_t) \leq 0, \qquad\qquad\qquad t \in [0, N_t - 1] \tag{4.3}$$

$$x^- \leq x_t \leq x^+, \qquad\qquad\qquad t \in [0, N_t - 1] \tag{4.4}$$

$$-\delta_t x \leq x_t - x_{t-\Delta t} \leq \delta_t x, \qquad\qquad\qquad t \in [1, N_t - 1] \tag{4.5}$$

4.2 Solvers

TCOPFLOW solves the coupled multi-period using the solver Ipopt [7]. Currently, ExaGO only supports solving TCOPFLOW using Ipopt [7] , which is also the default solver. Solving TCOPFLOW is supported only on single process.

4.3 Input and Output

- **Network file:** The network file describing the network details. Only MATPOWER format files are currently supported.

- **Load data:** One file for load real power and one fo reactive power. The files need to be in CSV format. An example of the format for the 9-bus case is here.

- **Wind generation:** The wind generation time-series described in CSV format. See an example of the format here.

If the load data and/or wind generation profiles are not set then a flat profile is assumed, i.e., the load and wind generation for all hours is constant.

The TCOPFLOW output is saved to a directory named *tcopflowout*. This directory contains N_t files, one for each time-step, in MATPOWER data file format.

4.4 Usage

```
./tcopflow -netfile <netfilename> -tcopflow_pload_profile <ploadprofile> \
-tcopflow_windgen_profile <windgenprofile> <tcopflowoptions> \
-tcopflow_qload_profile <qloadprofile>
```

4.5 Options

See table 4.1. In addition, all OPFLOW options given in Table 3.4 can be used.

Table 4.1: TCOPFLOW options

Option	Meaning	Values (Default value)
-netfile	Network file name	string ¡ 4096 characters (case9mod_gen3_wind.m)
-save_output	Save output to file	0 or 1 (0)
-tcopflow_pload_profile	Real power load profile	string ¡ 4096 characters (load_P.csv)
-tcopflow_qload_profile	Reactive power load profile	string ¡ 4096 characters (load_Q.csv)
-tcopflow_windgen_profile	Wind generation profile	string ¡ 4096 characters (case9/scenarios_9bus.csv)
-tcopflow_dT	Length of time-step (minutes)	double (5.0)
-tcopflow_duration	Total duration (hours)	double (0.5)
-tcopflow_iscoupling	Coupling between time steps (ramp constraints)	0 or 1 (0)

4.6   Examples

Some TCOPFLOW example runs are provided with some sample output. Options are the default options provided in table 3.4 and 4.1 unless otherwise specified. Sample output is generated from running examples from the installation directory.

Example using Ipopt [7]solver:

```
$ ./bin/tcopflow -tcopflow_iscoupling 1 -print_output
[ExaGO INFO]: -- Checking ... -options_file not passed          exists:
    ↪   no
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.500000 hours, timestep = 5.000000 minutes, \
                                     number of time-steps = 7
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
Setting: "0" is not a valid setting for Option: tol.
Check the option documentation.

### tol (Real Number) ###
Category: Convergence
Description: Desired convergence tolerance (relative).
0 < (1e-08) <= +inf


************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the Eclipse
Public License (EPL).
************************************************************************


This is Ipopt version 3.12.10, running with linear solver ma27.


Number of nonzeros in equality constraint Jacobian...:       770
```

```
Number of nonzeros in inequality constraint Jacobian.:      610
Number of nonzeros in Lagrangian Hessian.............:      630

Total number of variables...........................:      161
                     variables with only lower bounds:        0
                variables with lower and upper bounds:      105
                     variables with only upper bounds:        0
Total number of equality constraints.................:      126
Total number of inequality constraints...............:      186
        inequality constraints with only lower bounds:       42
   inequality constraints with lower and upper bounds:      144
        inequality constraints with only upper bounds:        0

iter  objective  inf_pr  inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  5.2094875e+04 1.80e+00 5.02e+01  -1.0 0.00e+00 - 0.00e+00 0.00e+00    0
1  3.6277420e+04 1.40e+00 1.06e+02  -1.0 2.05e+00 - 1.88e-01 2.20e-01f  1
...
33 2.0843675e+04 1.13e-14 3.32e-09  -8.6 1.76e-08 - 1.00e+00 1.00e+00h  1

Number of Iterations....: 33


                                   (scaled)                 (unscaled)
Objective...........:   4.6734697780313832e+02    2.0843675210019970e+04
Dual infeasibility..:   3.3166774138882895e-09    1.4792381265941772e-07
Constraint violation:   1.1296519275560968e-14    1.1296519275560968e-14
Complementarity.....:   4.9900570925632554e-09    2.2255654632832121e-07
Overall NLP error...:   4.9900570925632554e-09    2.2255654632832121e-07


Number of objective function evaluations            = 47
Number of objective gradient evaluations            = 34
Number of equality constraint evaluations           = 47
Number of inequality constraint evaluations         = 47
Number of equality constraint Jacobian evaluations  = 34
Number of inequality constraint Jacobian evaluations = 34
Number of Lagrangian Hessian evaluations            = 33
Total CPU secs in IPOPT (w/o function evaluations)  =      0.025
Total CPU secs in NLP function evaluations          =      0.032


EXIT: Optimal Solution Found.
================================================================
        Multi-Period Optimal Power Flow
================================================================
OPFLOW Model                    POWER_BALANCE_POLAR
Solver                          IPOPT
Duration (minutes)              30.00
Time-step (minutes)             5.00
```

```
Number of steps                        7
Active power demand profile            datafiles/case9/load_P.csv
Rective power demand profile           datafiles/case9/load_Q.csv
Wind generation profile                datafiles/case9/scenarios_9bus.csv
Load loss allowed                      NO
Power imbalance allowed                NO
Ignore line flow constraints           NO

Number of variables                    168
Number of equality constraints         126
Number of inequality constraints       168
Number of coupling constraints         18

Convergence status                     CONVERGED
Objective value                        20843.68
```

Bus	Pd	Qd	Vm	Va	mult_Pmis	mult_Qmis
1	0.00	0.00	1.040	0.000	2063.35	−0.00
2	0.00	0.00	1.025	4.553	2013.40	0.00
3	0.00	0.00	1.025	2.955	2017.53	0.00
4	0.00	0.00	1.037	−2.174	2063.59	−0.74
5	75.00	30.00	1.025	−3.405	2074.28	2.68
6	90.00	30.00	1.023	−4.213	2092.22	4.26
7	0.00	0.00	1.033	0.783	2014.09	3.85
8	100.00	35.00	1.022	−1.667	2035.30	7.65
9	0.00	0.00	1.036	0.264	2018.00	3.11

From	To	Status	Sft	Stf	Slim	mult_Sf	mult_St
1	4	1	71.31	71.13	380.00	−0.00	−0.00
2	7	1	111.80	112.67	250.00	−0.00	−0.00
3	9	1	86.66	87.56	300.00	−0.00	−0.00
4	5	1	28.34	34.62	250.00	−0.00	−0.00
4	6	1	42.81	45.47	250.00	−0.00	−0.00
5	7	1	47.86	51.21	250.00	−0.00	−0.00
6	9	1	49.48	52.48	150.00	−0.00	−0.00
7	8	1	63.89	65.25	250.00	−0.00	−0.00
8	9	1	41.64	36.64	150.00	−0.00	−0.00

Gen	Status	Fuel	Pg	Qg	Pmin	Pmax	Qmin	Qmax
1	1	COAL	71.06	5.89	10.00	350.00	−300.00	300.00
2	1	COAL	111.38	−9.71	10.00	300.00	−300.00	300.00

```
3       1       WIND    85.00   -16.87     0.00      85.00   -300.00   300.00
[ExaGO INFO]: Finalizing tcopflow application.
```

Chapter 5

Security-constrained optimal power flow (SCOPFLOW)

SCOPFLOW solves a contingency-constrained optimal power flow problem. The problem is set up as a two-stage optimization problem where the first-stage (base-case) represents the normal operation of the grid and the second-stage comprises of N_c contingency scenarios. Each contingency scenario can be single or multi-period.

5.1   Formulation

5.1.1   Single-period

The contingency-constrained optimal power flow (popularly termed as security-constrained optimal power flow (SCOPF) in power system parlance) attempts to find a least cost dispatch for the base case (or no contingency) while ensuring that if any of contingencies do occur then the system will be secure. This is illustrated in Fig. 5.1 for a SCOPF with a base-case c_0 and three contingencies.
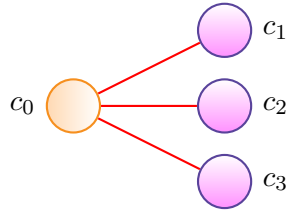
Figure 5.1: Contingency constrained optimal power flow example with three contingencies. c_0 represents the base case (or no contingency case). c_1, c_2, c_3 are the three contingency cases. Each of the contingency states is coupled with the base-case through ramping constraints (denoted by red lines)

In general form, the equations for contingency-constrained optimal power flow are given by (5.1) – (5.5). This is a two-stage stochastic optimization problem where the first stage is the base case c_0 and each of the contingency states $c_i, i \in [1, N_c - 1]$ are second-stage subproblems. SCOPFLOW aims to minimize the objective $\sum_{c=0}^{N_c-1} f(x_c)$, while adhering to the equality $(g(x_c))$, inequality $h(x_c)$, and the lower/upper bound (x^-, x^+) constraints. Equation (5.5) represents the coupling between the base-case and each of the contingency states c_i. Equation (5.5) is the most typical form of coupling that limits the deviation of the contingency variables x_c from the base x_0 to within $\delta_c x$. An example of this constraint could be the allowed real power output deviation for the generators constrained by their ramp limit.

$$\min \ f(x_0) \tag{5.1}$$

$$\text{s.t.}$$

$$g(x_c) = 0, \qquad\qquad\qquad c \in [0, N_c - 1] \tag{5.2}$$

$$h(x_c) \leq 0, \qquad\qquad\qquad c \in [0, N_c - 1] \tag{5.3}$$

$$x^- \leq x_c \leq x^+, \qquad\qquad\qquad c \in [0, N_c - 1] \tag{5.4}$$

$$-\delta_c x \leq x_c - x_0 \leq \delta_c x, \qquad\qquad\qquad c \in [1, N_c] \tag{5.5}$$

5.1.2 Multiperiod

In the multi-period version,each contingency comprises of multiple time-periods. The multiple periods have variables and constraints as described in chapter 4. An example of multi-contingency multi-period optimal power flow is illustrated in Fig. 5.2 with two contingencies c_0 and c_1. Here, c_0 is the case with no contingencies, i.e., the base-case. Each contingency is multi-period with four time-periods. Each time-step is coupled with its adjacent one through ramping constraints. We assume that the contingency is incident at the first time-step, i.e. at t_0. This results in the coupling between the contingency cases $c_i, i \in [1, N_c - 1]$ and the base-case c_0 only at time-step t_0 as shown in Fig. 5.2.
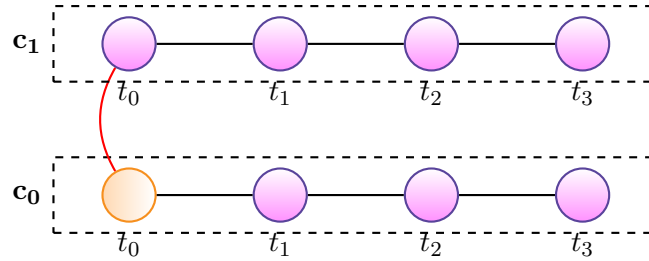
Figure 5.2: Multi-period contingency constrained optimal power flow example with two contingencies c_0 and c_1, each with four time-periods t_0, t_1, t_2, t_3. State c_0, t_0 represent the base case (no contingency) case. We assume that any contingency is incident at the first time-step, i.e., at t_0. The contingency states c_1, t_0 is coupled with the no-contingency state c_0, t_0 at time t_0. The red line denotes the coupling between the contingency.

The overall objective of this contingency-constrained multi-period optimal power flow is to find a secure dispatch for base-case c_0 while adhering to contingency and temporal constraints. Its general formulation is given in Eqs. (5.6) – (5.12).

$$\min \sum_{c=0}^{N_c-1} \sum_{t=0}^{N_t-1} f(x_{c,t}) \tag{5.6}$$

s.t.

$$g(x_{c,t}) = 0, \qquad\qquad c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{5.7}$$

$$h(x_{c,t}) \leq 0, \qquad\qquad c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{5.8}$$

$$x^- \leq x_{c,t} \leq x^+, \qquad\qquad c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{5.9}$$

$$-\delta_t x \leq x_{c,t} - x_{c,t-\Delta t} \leq \delta_t x, \qquad\qquad c \in [0, N_c - 1], t \in [1, N_t - 1] \tag{5.10}$$

$$-\delta_c x \leq x_{c,0} - x_{0,0} \leq \delta_c x, \qquad\qquad c \in [1, N_c - 1] \tag{5.11}$$

$$\tag{5.12}$$

In this formulation, the objective is to reduce the cost for the base-case time horizon, where $f(x_{0,t})$ is the objective cost for contingency c_0 at time t. Equation (5.12) represents the coupling between the base case c_0 and each contingency c_i at time-step t_0. We use a simple box constraint $\delta_c x$ to restrict the deviation of decision variables $x_{c,0}$ from the base-case $x_{0,0}$. The bound $\delta_c x$ could represent here, for example, the allowable reserve for each generator.

5.2   Solvers

SCOPFLOW can be solved with Ipopt [7]. If one wants to solve each contingency independently, i.e., without any coupling constraints then use *EMPAR* solver. *EMPAR* distributes the contingencies to different processes when executed in parallel.

5.3   Input and Output

To execute SCOPFLOW, the following files are required:

- **Network file:** The network file describing the network details. Only MATPOWER format files are currently supported.

- **Contingency file:** The file describing the contingencies. Contingencies can be single or multiple outages. The contingency file needs to be described in PTI format.

If the multi-period option is chosen, then additional files describing the load and wind generation can be (optionally) set.

- **Load data:** One file for load real power and one fo reactive power. The files need to be in CSV format. An example of the format for the 9-bus case is here.

- **Wind generation:** The wind generation time-series described in CSV format. See an example of the format here.

The SCOPFLOW output is saved to a directory named *scopflowout*. This directory contains N_c files to save the solution for each contingency in MATPOWER datafile format. Each file has the name *cont_xx* where *xx* is the contingency number.

If multi-period option is chosen then N_c subdirectories are created (one for each contingency), and each subdirectory contains N_t output files, one for each time-period. The subdirectories have the naming convention *cont_xx* and the output file are named as *t_yy* where *yy* is the time-step number.

5.4 Usage

```
  ./scopflow -netfile <netfilename> -ctgcfile <ctgcfilename> \
  <scopflowoptions> [-scopflow_enable_multiperiod 1]
```

5.5 Options

See table 5.1. In addition, all OPFLOW options in Table 3.4 and TCOPFLOW options in Table 4.1 can be used.

Table 5.1: SCOPFLOW options

Option	Meaning	Values (Default value)
-netfile	Network file name	string ¡ 4096 characters (case9mod_gen3_wind.m)
-ctgcfile	Contingency file name	string ¡ 4096 characters (case9.cont)
-save_output	Save output to directory	0 or 1 (0)
-scopflow_solver	Set solver for scopflow	IPOPT or EMPAR
-scopflow_Nc	Number of contingencies	int (0. Passing -1 results in all contingencies in the file used)
-scopflow_mode	Operation mode: Preventive or corrective	0 or 1 (0)
-scopflow_iscoupling	Coupling between first and second stage	0 or 1 (0)
-scopflow_enable_multiperiod	Multi-period SCOPFLOW	0 or 1 (0)
-scopflow_pload_profile	Real power load profile	string (load_P.csv)
-scopflow_qload_profile	Reactive power load profile	string (load_Q.csv)
-scopflow_windgenprofile	Wind generation profile	string (case9/scenarios_9bus.csv)
-scopflow_dT	Length of time-step (minutes)	double (5.0)
-scopflow_duration	Total duration (hours)	double (0.5)

Depending on the *mode*, SCOPFLOW can either be *preventive* (mode = 0) or *corrective* (mode = 1). In the preventive mode, the PV and PQ generator real power is fixed to its corresponding base-case values. The generators at the referencce bus pick up any make-up power required for the contingency. The corrective mode allows deviation of the PV and PQ generator real power from the base-case dispatch constrained by its 30-min. ramp rate capability.

-scopflow_enable_multiperiod 1 must be used in order to enable any of the options listed in table 5.1 that are below -scopflow_enable_multiperiod.

5.6 Examples

Some SCOPFLOW example runs are provided with some sample output. Options are the default options given in table 3.4, 4.1 and 5.1 unless otherwise specified. Sample output is generated from running examples in the installation directory.

Example using the Ipopt [7]solver:

```
$ ./bin/scopflow -scopflow_solver IPOPT -tcopflow_iscoupling 1 \
```

```
-scopflow_iscoupling 1 -print_output
[ExaGO INFO]: -- Checking ... -options_file not passed      exists: no
SCOPFLOW: Application created
SCOPFLOW running with 2 contingencies (base case + 1 contingencies)
Rank 0 has 2 contingencies, range [0 -- 2]
SCOPFLOW: Using IPOPT solver
SCOPFLOW: Setup completed


******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the Eclipse
Public License (EPL).
******************************************************************************


This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      231
Number of nonzeros in inequality constraint Jacobian.:      182
Number of nonzeros in Lagrangian Hessian.............:      249


Total number of variables............................:       55
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       39
                     variables with only upper bounds:        0
Total number of equality constraints.................:       44
Total number of inequality constraints...............:       59
        inequality constraints with only lower bounds:       21
   inequality constraints with lower and upper bounds:       38
        inequality constraints with only upper bounds:        0


iter    objective    inf_pr   inf_du lg(mu)  ||d|| lg(rg) alpha_du alpha_pr  ls
0  7.4421250e+03 1.80e+00 1.69e+01  -1.0 0.00e+00  -  0.00e+00 0.00e+00   0
1  6.9108233e+03 1.65e+00 1.56e+01  -1.0 1.24e+00  -  4.46e-02 8.38e-02f  1
2  5.9991119e+03 1.30e+00 5.10e+01  -1.0 7.41e-01  -  7.07e-02 2.08e-01f  1
...
43 3.0556401e+03 1.07e-14 3.95e-10  -8.6 2.79e-09  -  1.00e+00 1.00e+00f  1


Number of Iterations....: 43


                                   (scaled)                 (unscaled)
Objective...........:  6.8512110967619464e+01    3.0556401491558281e+03
Dual infeasibility..:  3.9453703519053556e-10    1.7596351769497887e-08
Constraint violation:  1.0658141036401503e-14    1.0658141036401503e-14
Complementarity.....:  5.0027986112663857e-09    2.2312481806248081e-07
Overall NLP error...:  5.0027986112663857e-09    2.2312481806248081e-07
```

```
Number of objective function evaluations          = 59
Number of objective gradient evaluations          = 44
Number of equality constraint evaluations         = 59
Number of inequality constraint evaluations       = 59
Number of equality constraint Jacobian evaluations   = 44
Number of inequality constraint Jacobian evaluations = 44
Number of Lagrangian Hessian evaluations          = 43
Total CPU secs in IPOPT (w/o function evaluations)  =      0.019
Total CPU secs in NLP function evaluations          =      0.009


EXIT: Optimal Solution Found.
================================================================
        Security-Constrained Optimal Power Flow
================================================================
OPFLOW Formulation               POWER_BALANCE_POLAR
Solver                           IPOPT
Initialization                   MIDPOINT
Number of contingencies          1
Load loss allowed                NO
Power imbalance allowed          NO
Ignore line flow constraints     NO

Number of variables              57
Number of equality constraints   42
Number of inequality constraints 58
Number of coupling constraints   3

Convergence status              CONVERGED
Objective value                 3055.64


----------------------------------------------------------------
Bus      Pd      Qd      Vm      Va      mult_Pmis      mult_Qmis
----------------------------------------------------------------
1       0.00    0.00   1.040   0.000     2158.80        -0.00
2       0.00    0.00   1.025   4.572     2108.44         0.00
3       0.00    0.00   1.025   1.912     2118.81         0.00
4       0.00    0.00   1.038  -2.306     2159.07        -0.61
5      75.00   30.00   1.026  -3.550     2170.87         2.93
6      90.00   30.00   1.023  -4.551     2191.34         4.30
7       0.00    0.00   1.033   0.613     2109.20         4.12
8     100.00   35.00   1.022  -2.070     2133.92         7.92
9       0.00    0.00   1.036  -0.462     2119.23         2.84


----------------------------------------------------------------
From     To     Status     Sft     Stf     Slim    mult_Sf  mult_St
----------------------------------------------------------------
1        4        1        75.61   75.44   380.00   -0.00    -0.00
```

```
2           7           1      117.34    118.26    250.00     -0.00     -0.00
3           9           1       76.90     77.68    300.00     -0.00     -0.00
4           5           1       28.64     34.88    250.00     -0.00     -0.00
4           6           1       46.84     48.91    250.00     -0.00     -0.00
5           7           1       47.56     50.96    250.00     -0.00     -0.00
6           9           1       45.96     48.57    150.00     -0.00     -0.00
7           8           1       69.77     70.78    250.00     -0.00     -0.00
8           9           1       37.09     30.76    150.00     -0.00     -0.00


--------------------------------------------------------------------
Gen    Status   Fuel     Pg        Qg       Pmin     Pmax     Qmin     Qmax
--------------------------------------------------------------------
1       1      COAL    75.40      5.62     10.00   350.00  -300.00   300.00
2       1      COAL   116.97     -9.31     10.00   300.00  -300.00   300.00
3       1      WIND    75.00    -16.97      0.00    75.00  -300.00   300.00
[ExaGO INFO]: Finalizing scopflow application
```

Example using the IPOPT solver with multiperiod enabled:

```
$./bin/scopflow -scopflow_solver IPOPT -scopflow_enable_multiperiod 1 \
-tcopflow_iscoupling 1 -scopflow_iscoupling 1
[ExaGO INFO]: -- Checking ... -options_file not passed      exists: no
SCOPFLOW: Application created
SCOPFLOW running with 2 contingencies (base case + 1 contingencies)
Rank 0 has 2 contingencies, range [0 -- 2]
SCOPFLOW: Using IPOPT solver
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes, \
          number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes, \
          number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
SCOPFLOW: Setup completed


*************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the
Eclipse Public License (EPL).
*************************************************************************


This is Ipopt version 3.12.10, running with linear solver ma27.


Number of nonzeros in equality constraint Jacobian...:      640
Number of nonzeros in inequality constraint Jacobian.:      494
```

```
Number of nonzeros in Lagrangian Hessian.............:      540

Total number of variables...........................:      138
                    variables with only lower bounds:        0
               variables with lower and upper bounds:       90
                    variables with only upper bounds:        0
Total number of equality constraints................:      110
Total number of inequality constraints..............:      151
        inequality constraints with only lower bounds:       36
   inequality constraints with lower and upper bounds:      115
        inequality constraints with only upper bounds:        0


iter objective inf_pr  inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
0   4.4652750e+04 1.80e+00 5.01e+01 -1.0 0.00e+00  - 0.00e+00 0.00e+00    0
1   3.1127309e+04 1.41e+00 1.01e+02 -1.0 1.97e+00  - 1.87e-01 2.18e-01f  1
...
55  1.7423413e+04 9.91e-15 8.48e-10 -8.6 9.66e-09  - 1.00e+00 1.00e+00h  1

Number of Iterations....: 55


                                   (scaled)                 (unscaled)
Objective...........:    3.9065947675497836e+02    1.7423412663272036e+04
Dual infeasibility..:    8.4833681960093011e-10    3.7835822154201486e-08
Constraint violation:    9.9087404947795221e-15    9.9087404947795221e-15
Complementarity.....:    3.3875752035319637e-09    1.5108585407752560e-07
Overall NLP error...:    3.3875752035319637e-09    1.5108585407752560e-07



Number of objective function evaluations             = 91
Number of objective gradient evaluations             = 56
Number of equality constraint evaluations            = 91
Number of inequality constraint evaluations          = 91
Number of equality constraint Jacobian evaluations   = 56
Number of inequality constraint Jacobian evaluations = 56
Number of Lagrangian Hessian evaluations             = 55
Total CPU secs in IPOPT (w/o function evaluations)   =      0.045
Total CPU secs in NLP function evaluations           =      0.052


EXIT: Optimal Solution Found.
[ExaGO INFO]: Finalizing scopflow application.
```

Example using the *EMPAR* solver with multiperiod enabled:

```
$ ./bin/scopflow -scopflow_solver EMPAR -scopflow_enable_multiperiod 1 \
-tcopflow_iscoupling 1 -scopflow_iscoupling 1
[ExaGO INFO]: -- Checking ... -options_file not passed      exists: no
SCOPFLOW: Application created
SCOPFLOW running with 2 contingencies (base case + 1 contingencies)
```

```
Rank 0 has 2 contingencies, range [0 -- 2]
SCOPFLOW: Using EMPAR solver
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes,
          number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes,
          number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
SCOPFLOW: Setup completed
Setting: "0" is not a valid setting for Option: tol.
Check the option documentation.

### tol (Real Number) ###
Category: Convergence
Description: Desired convergence tolerance (relative).
0 < (1e-08) <= +inf


**********************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under
the Eclipse Public License (EPL).
**********************************************************************

This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      330
Number of nonzeros in inequality constraint Jacobian.:      258
Number of nonzeros in Lagrangian Hessian.............:      270

Total number of variables............................:       69
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       45
                     variables with only upper bounds:        0
Total number of equality constraints.................:       54
Total number of inequality constraints...............:       78
        inequality constraints with only lower bounds:       18
   inequality constraints with lower and upper bounds:       60
        inequality constraints with only upper bounds:        0

iter  objective   inf_pr  inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  2.2326375e+04 1.80e+00 5.02e+01  -1.0 0.00e+00  -  0.00e+00 0.00e+00   0
1  1.5533385e+04 1.40e+00 1.02e+02  -1.0 1.95e+00  -  1.88e-01 2.20e-01f  1
...
```

```
32  8.6735595e+03 7.99e-15 3.70e-09  -8.6 1.93e-08 - 1.00e+00 1.00e+00h 1


Number of Iterations....: 32


                                (scaled)                    (unscaled)
Objective...........:  1.9447442817603635e+02    8.6735594966512217e+03
Dual infeasibility..:  3.6983296779767669e-09    1.6494550363776382e-07
Constraint violation:  7.9936057773011271e-15    7.9936057773011271e-15
Complementarity.....:  4.9913149587308981e-09    2.2261264715939807e-07
Overall NLP error...:  4.9913149587308981e-09    2.2261264715939807e-07



Number of objective function evaluations          = 40
Number of objective gradient evaluations          = 33
Number of equality constraint evaluations         = 40
Number of inequality constraint evaluations       = 40
Number of equality constraint Jacobian evaluations  = 33
Number of inequality constraint Jacobian evaluations = 33
Number of Lagrangian Hessian evaluations          = 32
Total CPU secs in IPOPT (w/o function evaluations) =      0.019
Total CPU secs in NLP function evaluations         =      0.014


EXIT: Optimal Solution Found.
Setting: "0" is not a valid setting for Option: tol.
Check the option documentation.

### tol (Real Number) ###
Category: Convergence
Description: Desired convergence tolerance (relative).
0 < (1e-08) <= +inf
This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:     306
Number of nonzeros in inequality constraint Jacobian.:     234
Number of nonzeros in Lagrangian Hessian.............:     270


Total number of variables............................:      69
                    variables with only lower bounds:       0
               variables with lower and upper bounds:      45
                    variables with only upper bounds:       0
Total number of equality constraints.................:      54
Total number of inequality constraints...............:      72
        inequality constraints with only lower bounds:      18
   inequality constraints with lower and upper bounds:      54
        inequality constraints with only upper bounds:       0


iter objective  inf_pr   inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
```

```
0  2.2326375e+04 1.80e+00 4.89e+01  -1.0 0.00e+00  - 0.00e+00 0.00e+00    0
1  1.5535979e+04 1.41e+00 1.01e+02  -1.0 1.97e+00  - 1.87e-01 2.18e-01f  1
...
51  8.7491131e+03 8.88e-15 4.80e-09  -8.6 1.07e-08  - 1.00e+00 1.00e+00h  1

Number of Iterations....: 51

                              (scaled)                 (unscaled)
Objective...........:    1.9616845456090007e+02    8.7491130734161434e+03
Dual infeasibility..:    4.8019002633480731e-09    2.1416475174532407e-07
Constraint violation:    8.8817841970012523e-15    8.8817841970012523e-15
Complementarity.....:    4.9761717933322852e-09    2.2193726198261992e-07
Overall NLP error...:    4.9761717933322852e-09    2.2193726198261992e-07


Number of objective function evaluations           = 54
Number of objective gradient evaluations           = 52
Number of equality constraint evaluations          = 54
Number of inequality constraint evaluations        = 54
Number of equality constraint Jacobian evaluations  = 52
Number of inequality constraint Jacobian evaluations = 52
Number of Lagrangian Hessian evaluations           = 51
Total CPU secs in IPOPT (w/o function evaluations) =       0.024
Total CPU secs in NLP function evaluations         =       0.016

EXIT: Optimal Solution Found.
[ExaGO INFO]: Finalizing scopflow application.
```

Chapter 6

Stochastic optimal power flow (SOPFLOW)

SOPFLOW solves a stochastic security-constrained multi-period optimal power flow problem. The problem is set up as a two-stage optimization problem where the first-stage (base-case) represents the normal operation of the grid (or the most likely forecast) and the second-stage comprises of N_s scenarios of forecast deviation. Each scenario can have multiple contingencies and each contingency can be multi-period. Thus, depending on the options selected, each stochastic scenario can be

- Single-period, no contingency

- Single-period contingencies

- Multi-period contingencies

6.1 Formulation

An illustration of SOPFLOW in Fig. 6.1 for a case with two scenarios s_0 and s_1. Each scenario has two contingencies c_0, c_1, and each contingency has four time-periods.

The formulation for the stochastic security-constrained multi-period optimal power flow is given in (6.1) – (6.7). In this formulation, the objective is to reduce the expected cost, where $f(x_{s,c,t})$ is the cost for scenario s with contingency c at time t. π_s is the probability of scenario s.

$$\min \sum_{s=0}^{N_s-1} \pi_s \sum_{c=0}^{N_c-1} \sum_{t=0}^{N_t-1} f(x_{s,c,t}) \tag{6.1}$$

$$\text{s.t.}$$

$$g(x_{s,c,t}) = 0, \qquad s \in [1, N_s - 1], c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{6.2}$$

$$h(x_{s,c,t}) \leq 0, \qquad s \in [1, N_s - 1], c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{6.3}$$

$$x^- \leq x_{s,c,t} \leq x^+, \qquad s \in [1, N_s - 1], c \in [0, N_c - 1], t \in [0, N_t - 1] \tag{6.4}$$

$$-\delta_t x \leq x_{s,c,t} - x_{s,c,t-\Delta t} \leq \delta_t x, \qquad s \in [1, N_s - 1], c \in [0, N_c - 1], t \in [1, N_t - 1] \tag{6.5}$$

$$-\delta_c x \leq x_{s,c,0} - x_{s,0,0} \leq \delta_c x, \qquad s \in [1, N_s - 1], c \in [1, N_c - 1] \tag{6.6}$$

$$-\delta_s x \leq x_{s,0,0} - x_{0,0,0} \leq \delta_s x, \qquad s \in [1, N_s - 1] \tag{6.7}$$
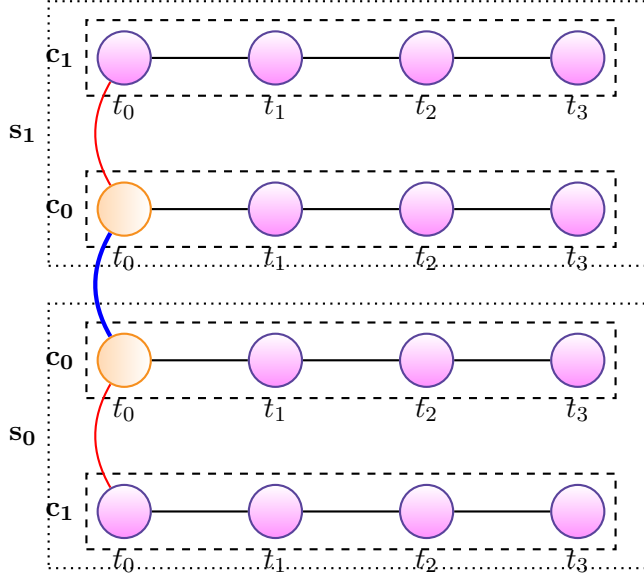
Figure 6.1: Stochastic multi-period contingency constrained example with two scenarios s_0 and s_1. Each scenario has two contingencies c_0,c_1 and each contingency consists of four time-periods t_0, t_1, t_2, t_3. State s_0, c_0, t_0 represent the base case (no contingency) case for the two scenarios. We assume that any contingency is incident at the first time-step, i.e., at t_0. Thus, the contingency states c_1, t_0 is coupled with the no-contingency state c_0, t_0 at time t_0 for both the scenarios. The red line denotes the coupling between the contingency and the no-contingency states.The blue line denotes the coupling between the scenarios

SOPFLOW uses all the modeling details used for modeling an optimal power flow problem, i.e., each of the circles shown in Fig. 6.1 has the modeling details of an optimal power flow problem (OPFLOW). Incorporating the probabilities π_s for each scenario is not implemented yet which leads to each scenario having an equal probability.

Currently, SOPFLOW uses wind power generation as the stochastic variables and each scenario is a realization of the power output from wind generators. A zero fuel cost is used for wind power generation to ensure wind generation would be the dispatched to the given target level (upper limit).

For contingencies, SOPFLOW supports generation and/or transmission outages. A contingency can have multiple outages, but, it should not cause any islanding. The coupling between the no-contingency and the contingency case for each scenario is also the difference in real power output $(p^{\text{g}}_{jsct} - p^{\text{g}}_{js0t}, \ j \in J^{\text{gen}})$ that must be within the 30 minute generator ramp rate.

For multi time-period, we use ramping constraints on the generator real power output between successive time steps.

SOPFLOW can be run in two modes: preventive and corrective. In the preventive mode, generator real power output is fixed to the base-case values for generators at PV bus(es). In this mode, the generators at the reference bus provide/absorb any deficit/surplus power. The corrective mode allows deviation of the PV and PQ generator real power from the base-case dispatch constrained by its 30-min. ramp rate capability. Note that the preventive/corrective mode is only applied at the first step t_0. In the successive time-steps, the generator dispatch is dictated by the previous step dispatch and the ramp limits.

41

6.2 Solvers

SOPFLOW can be solved with Ipopt [7]. If one wants to solve each scenario independently, i.e., without any coupling constraints then use *EMPAR* solver. *EMPAR* distributes the contingencies to different processes when executed in parallel.

6.3 Input and Output

The following files are needed for executing a SOPFLOW.

- **Network file:** The network file describing the network details. Only MATPOWER format files are currently supported.

- **Scenario file:** SOPFLOW only supports reading wind generation scenarios in a CSV format. An example of this format for the 9-bus case is here.

- **Contingency file:** Contingencies can be specified via PTI format file as described in chapter 5.The option -sopflow_enable_multicontingency should be set for multi-contingency problems.

- **Load data:** One file for load real power and one fo reactive power. The files need to be in CSV format. An example of the format for the 9-bus case is here.

The SOPFLOW output is saved to a directory named *sopflowout*. This directory contains N_s subdirectories to save the solution for each scenario. Each of these subdirectories contain N_c subdirectories, one for each contingency. Each contingency subdirectory has N_t MATPOWER format files to store the output for each time-period for the given contingency and scenario. The subdirectories have the directory name format *scen_x* where x is the scenario number, *cont_y* where y is the contingency number, and the output files have the file name format *t_z* where z is the time-step number.

6.4 Usage

```
./sopflow -netfile <netfilename>  -scenfile <scenfilename> \
[-sopflow_enable_multicontingency 1] <sopflowoptions>
```

6.5 Options

See table 6.1. With multi-contingency SOPFLOW, all SCOPFLOW options given in Table 5.1 can be used to tune the contingencies. All OPFLOW options in Table 3.4, along with TCOPFLOW options in Table 4.1 can also be used. Multi-contingency SOPFLOW also allows the options listed after -sopflow_enable_multicontingency in Table 6.1 to be used.

Depending on the *mode*, SOPFLOW can either be *preventive* (mode = 0) or *corrective* (mode = 1). In the preventive mode, the PV and PQ generator real power is fixed to its corresponding base-case values. The generators at the reference bus pick up any make-up power required for the contingency. The corrective mode allows deviation of the PV and PQ generator real power from the base-case dispatch constrained by its 30-min. ramp rate capability.

Table 6.1: SOPFLOW options

Option	Meaning	Values (Default value)
-netfile	Network file name	string ¡ 4096 characters (case9mod_gen3_wind.m)
-scenfile	Scenario file name	string ¡ 4096 characters (case9/scenarios_9bus.csv)
-sopflow_mode	Operation mode: Preventive or corrective	0 or 1 (0)
-sopflow_solver	Set solver for sopflow	IPOPT or EMPAR
-sopflow_enable_multicontingency	Multi-contingency SOPFLOW	0 or 1 (0)
-ctgcfile	Contingency file name	string (case9.cont)
-sopflow_Ns	Number of scenarios	int (0. Passing -1 results in all scenarios in the file being used)

6.6 Examples

Some SOPFLOW example runs are provided with some sample output. Options are the default options given in Tables 3.4, 4.1, 5.1 and 6.1 unless otherwise specified. Sample output is generated from running examples in the installation directory.

Example using the Ipopt [7] solver:

```
$ ./bin/sopflow -print_output -sopflow_solver IPOPT
[ExaGO INFO]: -- Checking ... -options_file not passed    exists: no
SOPFLOW: Application created
Rank[0]: color = 0, ns = 2, sstart= 0, send=2
SOPFLOW running with 2 scenarios (base case + 1 scenarios)
Rank 0 scenario range [0 -- 2]
SOPFLOW: Using IPOPT solver
SOPFLOW: Setup completed


**********************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the
Eclipse Public License (EPL).
**********************************************************************


This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      222
Number of nonzeros in inequality constraint Jacobian.:      168
Number of nonzeros in Lagrangian Hessian.............:      180

Total number of variables............................:       46
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       30
                     variables with only upper bounds:        0
Total number of equality constraints.................:       37
Total number of inequality constraints...............:       50
```

```
         inequality constraints with only lower bounds:        12
   inequality constraints with lower and upper bounds:        38
         inequality constraints with only upper bounds:         0


iter objective     inf_pr   inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  1.4884250e+04 1.80e+00 5.02e+01  -1.0 0.00e+00  - 0.00e+00 0.00e+00    0
1  1.0889088e+04 1.45e+00 7.87e+01  -1.0 1.92e+00  - 1.69e-01 1.93e-01f  1
2  1.0548318e+04 1.33e+00 7.28e+01  -1.0 6.01e-01  - 2.76e-02 8.30e-02f  1
...
29   5.9075224e+03 1.24e-14 2.96e-12  -7.0 4.91e-10  - 1.00e+00 1.00e+00h  1


Number of Iterations....: 29


                                (scaled)                 (unscaled)
Objective...........: 1.3245566021454471e+02   5.9075224455686939e+03
Dual infeasibility..: 2.9592247352450450e-12   1.3198142319192901e-10
Constraint violation: 1.2434497875801753e-14   1.2434497875801753e-14
Complementarity.....: 9.0910764708133484e-08   4.0546201059827535e-06
Overall NLP error...: 9.0910764708133484e-08   4.0546201059827535e-06



Number of objective function evaluations            = 33
Number of objective gradient evaluations            = 30
Number of equality constraint evaluations           = 33
Number of inequality constraint evaluations         = 33
Number of equality constraint Jacobian evaluations  = 30
Number of inequality constraint Jacobian evaluations = 30
Number of Lagrangian Hessian evaluations            = 29
Total CPU secs in IPOPT (w/o function evaluations)  =      0.006
Total CPU secs in NLP function evaluations          =      0.003

EXIT: Optimal Solution Found.
================================================================
        Stochastic Optimal Power Flow
================================================================
OPFLOW Formulation              POWER_BALANCE_POLAR
Solver                          IPOPT
Initialization                  MIDPOINT
Number of scenarios             1
Load loss allowed               NO
Power imbalance allowed         NO
Ignore line flow constraints    NO


Number of variables             48
Number of equality constraints  0
Number of inequality constraints 0
Number of coupling constraints  0
```

```
Convergence status                    CONVERGED
Objective value                       5907.52


----------------------------------------------------------------------
Bus        Pd       Qd      Vm      Va       mult_Pmis       mult_Qmis
----------------------------------------------------------------------
1        0.00     0.00   1.040   0.000       2218.76          -0.00
2        0.00     0.00   1.025   4.156       2170.20           0.00
3        0.00     0.00   1.025   1.685       2180.00           0.00
4        0.00     0.00   1.038  -2.389       2219.06          -0.56
5       75.00    30.00   1.026  -3.714       2231.93           2.96
6       90.00    30.00   1.023  -4.683       2252.92           4.40
7        0.00     0.00   1.033   0.291       2170.95           4.07
8      100.00    35.00   1.022  -2.352       2196.02           8.01
9        0.00     0.00   1.036  -0.689       2180.43           2.83


----------------------------------------------------------------------
From       To       Status    Sft      Stf      Slim    mult_Sf  mult_St
----------------------------------------------------------------------
1          4          1       78.31    78.15    380.00    -0.00    -0.00
2          7          1      114.58   115.48    250.00    -0.00    -0.00
3          9          1       76.90    77.69    300.00    -0.00    -0.00
4          5          1       30.36    36.13    250.00    -0.00    -0.00
4          6          1       47.82    49.77    250.00    -0.00    -0.00
5          7          1       45.94    49.28    250.00    -0.00    -0.00
6          9          1       45.11    47.63    150.00    -0.00    -0.00
7          8          1       68.77    69.83    250.00    -0.00    -0.00
8          9          1       37.83    31.76    150.00    -0.00    -0.00


----------------------------------------------------------------------
Gen   Status    Fuel     Pg       Qg       Pmin     Pmax     Qmin     Qmax
----------------------------------------------------------------------
1      1        COAL    78.13     5.42     10.00   350.00  -300.00   300.00
2      1        COAL   114.18    -9.55     10.00   300.00  -300.00   300.00
3      1        WIND    75.00   -17.00      0.00    75.00  -300.00   300.00
[ExaGO INFO]: Finalizing sopflow application.
```

Example using the *EMPAR* solver with multicontingency enabled:

```
$ ./bin/sopflow -sopflow_solver EMPAR -sopflow_enable_multicontingency 1
[ExaGO INFO]: -- Checking ... -options_file not passed      exists: no
SOPFLOW: Application created
Rank[0]: color = 0, ns = 2, sstart= 0, send=2
SOPFLOW running with 2 scenarios (base case + 1 scenarios)
Rank 0 scenario range [0 -- 2]
SOPFLOW: Using EMPAR solver
SCOPFLOW: Application created
```

```
SCOPFLOW running with 2 contingencies (base case + 1 contingencies)
Rank 0 has 2 contingencies, range [0 -- 2]
SCOPFLOW: Using IPOPT solver
SCOPFLOW: Setup completed
SCOPFLOW: Application created
SCOPFLOW running with 2 contingencies (base case + 1 contingencies)
Rank 0 has 2 contingencies, range [0 -- 2]
SCOPFLOW: Using IPOPT solver
SCOPFLOW: Setup completed
SOPFLOW: Setup completed


******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the
Eclipse Public License (EPL).
******************************************************************************


This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      226
Number of nonzeros in inequality constraint Jacobian.:      168
Number of nonzeros in Lagrangian Hessian.............:      215


Total number of variables............................:       51
                    variables with only lower bounds:        0
               variables with lower and upper bounds:       35
                    variables with only upper bounds:        0
Total number of equality constraints.................:       42
Total number of inequality constraints...............:       58
        inequality constraints with only lower bounds:       21
   inequality constraints with lower and upper bounds:       37
        inequality constraints with only upper bounds:        0


iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
0  7.4421250e+03 1.80e+00 1.04e+01  -1.0 0.00e+00  -  0.00e+00 0.00e+00   0
1  6.5388630e+03 1.51e+00 2.24e+01  -1.0 1.07e+00  -  5.38e-01 1.58e-01f  1
2  5.9394843e+03 1.28e+00 2.56e+02  -1.0 7.32e-01  -  3.31e-02 1.56e-01f  1
...
29  3.0556401e+03 1.07e-14 5.62e-10  -8.6 4.33e-09  -  1.00e+00 1.00e+00f  1


Number of Iterations....: 29


                                   (scaled)                 (unscaled)
Objective...........:  6.8512110964678271e+01    3.0556401490246512e+03
Dual infeasibility..:  5.6231688979129259e-10    2.5079333284691651e-08
Constraint violation:  1.0658141036401503e-14    1.0658141036401503e-14
Complementarity.....:  2.7880113972390153e-09    1.2434530831686009e-07
```

```
Overall NLP error...:   2.7880113972390153e-09    1.2434530831686009e-07


Number of objective function evaluations            = 48
Number of objective gradient evaluations            = 30
Number of equality constraint evaluations           = 48
Number of inequality constraint evaluations         = 48
Number of equality constraint Jacobian evaluations  = 30
Number of inequality constraint Jacobian evaluations = 30
Number of Lagrangian Hessian evaluations            = 29
Total CPU secs in IPOPT (w/o function evaluations)  =      0.010
Total CPU secs in NLP function evaluations          =      0.006

EXIT: Optimal Solution Found.
This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:      226
Number of nonzeros in inequality constraint Jacobian.:      168
Number of nonzeros in Lagrangian Hessian.............:      215
Total number of variables............................:       51
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       35
                     variables with only upper bounds:        0
Total number of equality constraints.................:       42
Total number of inequality constraints...............:       58
        inequality constraints with only lower bounds:       21
   inequality constraints with lower and upper bounds:       37
        inequality constraints with only upper bounds:        0

iter objective  inf_pr  inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  7.4421250e+03 1.80e+00 1.04e+01  -1.0 0.00e+00  - 0.00e+00 0.00e+00   0
1  6.2852600e+03 1.53e+00 2.73e+01  -1.0 1.34e+00  - 3.21e-01 1.50e-01f  1
...
28  2.8488309e+03 1.07e-14 3.60e-09  -8.6 1.94e-08  - 1.00e+00 1.00e+00h  1

Number of Iterations....: 28


                                (scaled)                 (unscaled)
Objective...........:  6.3875131478884526e+01    2.8488308639582501e+03
Dual infeasibility..:  3.5985088591418374e-09    1.6049349511772596e-07
Constraint violation:  1.0658141036401503e-14    1.0658141036401503e-14
Complementarity.....:  4.7446211848960068e-09    2.1161010484636192e-07
Overall NLP error...:  4.7446211848960068e-09    2.1161010484636192e-07


Number of objective function evaluations            = 43
Number of objective gradient evaluations            = 29
```

```
Number of equality constraint evaluations           = 43
Number of inequality constraint evaluations         = 43
Number of equality constraint Jacobian evaluations  = 29
Number of inequality constraint Jacobian evaluations = 29
Number of Lagrangian Hessian evaluations            = 28
Total CPU secs in IPOPT (w/o function evaluations)  =      0.016
Total CPU secs in NLP function evaluations          =      0.009

EXIT: Optimal Solution Found.
```

Example using the Ipopt [7]solver with multicontingency and multiperiod enabled:

```
./bin/sopflow -sopflow_solver IPOPT -sopflow_enable_multicontingency 1\
-scopflow_enable_multiperiod 1 -sopflow_Ns -1 -scopflow_Nc -1
[ExaGO INFO]: -- Checking ... -options_file not passed    exists: no
SOPFLOW: Application created
Rank[0]: color = 0, ns = 4, sstart= 0, send=4
SOPFLOW running with 4 scenarios (base case + 3 scenarios)
Rank 0 scenario range [0 -- 4]
SOPFLOW: Using IPOPT solver
SCOPFLOW: Application created
SCOPFLOW running with 10 contingencies (base case + 9 contingencies)
Rank 0 has 10 contingencies, range [0 -- 10]
SCOPFLOW: Using IPOPT solver
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes,
        number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
TCOPFLOW: Application created
TCOPFLOW: Duration = 0.166667 hours, timestep = 5.000000 minutes,
        number of time-steps = 3
TCOPFLOW: Using IPOPT solver
TCOPFLOW: Setup completed
...
SCOPFLOW running with 10 contingencies (base case + 9 contingencies)
Rank 0 has 10 contingencies, range [0 -- 10]
SCOPFLOW: Using IPOPT solver
...
TCOPFLOW: Setup completed
SCOPFLOW: Setup completed
SOPFLOW: Setup completed


********************************************************************
This program contains Ipopt, a library for large-scale nonlinear
optimization. Ipopt is released as open source code under the
Eclipse Public License (EPL).
********************************************************************
```

```
This is Ipopt version 3.12.10, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:    12678
Number of nonzeros in inequality constraint Jacobian.:     9636
Number of nonzeros in Lagrangian Hessian.............:    10404


Total number of variables............................:     2688
                     variables with only lower bounds:        0
                variables with lower and upper bounds:     1728
                     variables with only upper bounds:        0
Total number of equality constraints.................:     2223
Total number of inequality constraints...............:     2922
        inequality constraints with only lower bounds:      648
   inequality constraints with lower and upper bounds:     2274
        inequality constraints with only upper bounds:        0

iter   objective   inf_pr  inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr  ls
0  8.0374950e+05 2.59e+00 6.64e+01  -1.0 0.00e+00  - 0.00e+00 0.00e+00    0
1  7.8996010e+05 2.48e+00 6.98e+01  -1.0 1.43e+00 2.0 2.04e-02 3.97e-02f 1
2  7.7076082e+05 2.31e+00 9.50e+01  -1.0 1.16e+00 2.4 2.69e-02 6.75e-02f 1
...
197 4.2506331e+05 1.42e-14 3.34e-10  -7.0 2.10e-09  - 1.00e+00 1.00e+00h  1

Number of Iterations....: 197


                                   (scaled)                 (unscaled)
Objective...........:   9.5305673828307717e+03    4.2506330527425243e+05
Dual infeasibility..:   3.3401254461705077e-10    1.4896959489920465e-08
Constraint violation:   1.4210854715202004e-14    1.4210854715202004e-14
Complementarity.....:   9.1353878464689626e-08    4.0743829795251576e-06
Overall NLP error...:   9.1353878464689626e-08    4.0743829795251576e-06


Number of objective function evaluations             = 283
Number of objective gradient evaluations             = 198
Number of equality constraint evaluations            = 283
Number of inequality constraint evaluations          = 283
Number of equality constraint Jacobian evaluations   = 198
Number of inequality constraint Jacobian evaluations = 198
Number of Lagrangian Hessian evaluations             = 197
Total CPU secs in IPOPT (w/o function evaluations)   =      2.136
Total CPU secs in NLP function evaluations           =      4.421

EXIT: Optimal Solution Found.
[ExaGO INFO]: Finalizing sopflow application.
```

Appendices

Appendix A

Symbol reference

Units of measurement are given in Table (A.1), indices and index sets in Table (A.2), subsets in Table (A.3), special set elements in Table (A.4), and real-valued parameters in Table (A.5).

Table A.1:   Units of measurement

Symbol	Description
1	dimensionless. Dimensionless real number quantities are indicated by a unit of 1.
USD	US dollar. Cost, penalty, and objective values are expressed in USD.
h	hour. Time is expressed in h.
pu	per unit. Voltage magnitude is expressed in a per unit system under given base values, and the unit is denoted by pu
rad	radian. Voltage angles are expressed in rad.
MW	megawatt. Real power is expressed in MW.
MVar	megavolt-ampere-reactive. Reactive power is expressed in MVar.
MVA	megavolt-ampere. Apparent power is expressed in MVA.
MW at 1 pu	megawatt at unit voltage. Conductance is expressed in MW at 1 pu, meaning the conductance is such as to yield a real power flow equal to the indicated amount when the voltage is equal to 1 pu
MVar at 1 pu	megavolt-ampere-reactive at unit voltage. Susceptance is expressed in MVar at 1 pu, meaning the susceptance is such as to yield a reactive power flow equal to the indicated amount when the voltage is equal to 1 pu

Table A.2:   Index sets

Symbol	Description
$a \in A$	areas
$i \in I$	buses

Symbol	Description
$j \in J$	bus-connected grid components, i.e. loads, shunts, generators, stochastic resources, branches
$k \in K$	security contingencies, i.e. NERC $(n-1)$- or $(n-k)$-style contingencies, different from the severe event we are modeling
$s \in S$	stochastic scenarios
$t \in T$	time periods

Table A.3:   Subsets

Symbol	Description
$J^{\mathrm{gen}} \subset J$	generators
$J^{\mathrm{ld}} \subset J$	loads
$J^{\mathrm{br}} \subset J$	branches, i.e. lines, transformers
$J^{\mathrm{sh}} \subset J$	shunts
$J_i^{\mathrm{o}} \subset J$	branches with origin bus at bus i
$J_i^{\mathrm{d}} \subset J$	branches with destination bus at bus i

Table A.4:   Special set elements

Symbol	Description
$a_i \in A$	area of bus i
$i_j^{\mathrm{d}} \in I$	destination bus of branch $j \in J$
$i_j^{\mathrm{o}} \in I$	origin bus of branch $j \in J^{\mathrm{br}}$

Table A.5:   Real-valued parameters

Symbol	Description
$b_j^{\max}$	maximum susceptance for shunt $j \in J^{\mathrm{sh}}$ (MVar at 1 pu)

Symbol	Description
b_j^{min}	minimum susceptance for shunt $j \in J^{\text{sh}}$ (MVar at 1 pu)
b_j^{ch}	charging susceptance for branch $j \in J^{\text{br}}$ (MVar at 1 pu)
b_j^{s}	series susceptance for branch $j \in J^{\text{br}}$ (MVar at 1 pu)
p_j^{gset}	real power set point of generator $j \in J^{\text{gen}}$ (MW)
p_j^{gmax}	maximum real power output for generator $j \in J^{\text{gen}}$ (MW)
p_j^{gmin}	minimum real power output for generator $j \in J^{\text{gen}}$ (MW)
p_j^{r}	maximum ramp rate for generator $j \in J^{\text{gen}}$ (MW/h)
q_j^{gmax}	maximum reactive power output for generator $j \in J^{\text{gen}}$ (MVar)
q_j^{gmin}	minimum reactive power output for generator $j \in J^{\text{gen}}$ (MVar)
v_i^{max}	maximum voltage magnitude for bus $i \in I$ (pu)
v_i^{min}	minimum voltage magnitude for bus $i \in I$ (pu)
π_s	probability of scenario s (1)

Bibliography

[1] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dmitry Karpeyev, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.12, Argonne National Laboratory, 2019.

[2] D. Beckingsale, M. Mcfadden, J. Dahm, R. Pankajakshan, and R. Hornung. Umpire: Application-focused management and coordination of complex hierarchical memory. *IBM Journal of Research and Development*, 2019.

[3] David A Beckingsale, Jason Burmark, Rich Hornung, Holger Jones, William Killian, Adam J Kunen, Olga Pearce, Peter Robinson, Brian S Ryujin, and Thomas RW Scogland. Raja: Portable performance for large-scale scientific applications. In *2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pages 71–81. IEEE, 2019.

[4] David A Beckingsale, Marty J Mcfadden, Johann PS Dahm, Ramesh Pankajakshan, and Richard D Hornung. Umpire: Application-focused management and coordination of complex hierarchical memory. *IBM Journal of Research and Development*, 64(3/4):00–1, 2019.

[5] C. G. Petra, I.Aravena Solis, N. Gawande, V. Amatya, A. Li, J. Li, S. Abhyankar, S. Peles, M. Schanen, K. Kim, A. Maldonado, M. Anitescu. ExaSGD - Optimization grid dynamics at Exascale, 2020.

[6] Cosmin Petra. HiOP User Guide version 0.3, 2017.

[7] Andreas Waechter and L. Biegler. On the implementation of an interior-point filter line-search algorithm for large scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.