

Appendix B Data File Format

There are two versions of the MATPOWER case file format. MATPOWER versions 3.0.0 and earlier used the version 1 format internally. Subsequent versions of MATPOWER have used the version 2 format described below, though version 1 files are still handled, and converted automatically, by the `loadcase` and `savecase` functions.

In the version 2 format, the input data for MATPOWER are specified in a set of data matrices packaged as the fields of a MATLAB struct, referred to as a “MATPOWER case” struct and conventionally denoted by the variable `mpc`. This struct is typically defined in a case file, either a function M-file whose return value is the `mpc` struct or a MAT-file that defines a variable named `mpc` when loaded. The fields of this struct are `baseMVA`, `bus`, `branch`, `gen` and, optionally, `gencost`. The `baseMVA` field is a scalar and the rest are matrices. Each row in the data matrices corresponds to a single bus, branch, or generator and the columns are similar to the columns in the standard IEEE and PTI formats. The `mpc` struct also has a `version` field whose value is a string set to the current MATPOWER case version, currently '2' by default. The version 1 case format defines the data matrices as individual variables rather than fields of a struct, and some do not include all of the columns defined in version 2.

Numerous examples can be found in the case files listed in Table D-18 in Appendix D. The case files created by `savecase` use a tab-delimited format for the data matrices to make it simple to transfer data seamlessly back and forth between a text editor and a spreadsheet via simple copy and paste.

The details of the MATPOWER case format are given in the tables below and can also be accessed by typing `help caseformat` at the MATLAB prompt. First, the `baseMVA` field is a simple scalar value specifying the system MVA base used for converting power into per unit quantities. For convenience and code portability, `idx_bus` defines a set of constants to be used as named indices into the columns of the `bus` matrix. Similarly, `idx_brch`, `idx_gen` and `idx_cost` define names for the columns of `branch`, `gen` and `gencost`, respectively. The script `define_constants` provides a simple way to define all the usual constants at one shot. These are the names that appear in the first column of the tables below.

The MATPOWER case format also allows for additional fields to be included in the structure. The OPF is designed to recognize fields named **A**, **l**, **u**, **H**, **Cw**, **N**, **fparm**, **z0**, **z1** and **zu** as parameters used to directly extend the OPF formulation as described in Section 7.1. Additional standard optional fields include **bus_name**, **gentype** and **genfuel**.⁶¹ Other user-defined fields may also be included, such as the **reserves** field used in the example code throughout Section 7.3. The **loadcase** function will automatically load any extra fields from a case file and, if the appropriate 'savecase' callback function (see Section 7.3.5) is added via **add_userfcn**, **savecase** will also save them back to a case file.

Table B-1: Bus Data (**mpc.bus**)

| name | column | description |
|----------------------|--------|---|
| BUS_I | 1 | bus number (positive integer) |
| BUS_TYPE | 2 | bus type (1 = PQ, 2 = PV, 3 = ref, 4 = isolated) |
| PD | 3 | real power demand (MW) |
| QD | 4 | reactive power demand (MVar) |
| GS | 5 | shunt conductance (MW demanded at $V = 1.0$ p.u.) |
| BS | 6 | shunt susceptance (MVar injected at $V = 1.0$ p.u.) |
| BUS_AREA | 7 | area number (positive integer) |
| VM | 8 | voltage magnitude (p.u.) |
| VA | 9 | voltage angle (degrees) |
| BASE_KV | 10 | base voltage (kV) |
| ZONE | 11 | loss zone (positive integer) |
| VMAX | 12 | maximum voltage magnitude (p.u.) |
| VMIN | 13 | minimum voltage magnitude (p.u.) |
| LAM_P [†] | 14 | Lagrange multiplier on real power mismatch (u /MW) |
| LAM_Q [†] | 15 | Lagrange multiplier on reactive power mismatch (u /MVar) |
| MU_VMAX [†] | 16 | Kuhn-Tucker multiplier on upper voltage limit (u /p.u.) |
| MU_VMIN [†] | 17 | Kuhn-Tucker multiplier on lower voltage limit (u /p.u.) |

[†] Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .

⁶¹All three of these are cell arrays of strings. See **gentypes** and **genfuels** for more information on the corresponding fields.

Table B-2: Generator Data (`mpc.gen`)

| name | column | description |
|-----------------------|--------|---|
| GEN_BUS | 1 | bus number |
| PG | 2 | real power output (MW) |
| QG | 3 | reactive power output (MVA _r) |
| QMAX | 4 | maximum reactive power output (MVA _r) |
| QMIN | 5 | minimum reactive power output (MVA _r) |
| VG [‡] | 6 | voltage magnitude setpoint (p.u.) |
| MBASE | 7 | total MVA base of machine, defaults to <code>baseMVA</code> |
| GEN_STATUS | 8 | machine status, > 0 = machine in-service ≤ 0 = machine out-of-service |
| PMAX | 9 | maximum real power output (MW) |
| PMIN | 10 | minimum real power output (MW) |
| PC1 [*] | 11 | lower real power output of PQ capability curve (MW) |
| PC2 [*] | 12 | upper real power output of PQ capability curve (MW) |
| QC1MIN [*] | 13 | minimum reactive power output at PC1 (MVA _r) |
| QC1MAX [*] | 14 | maximum reactive power output at PC1 (MVA _r) |
| QC2MIN [*] | 15 | minimum reactive power output at PC2 (MVA _r) |
| QC2MAX [*] | 16 | maximum reactive power output at PC2 (MVA _r) |
| RAMP_AGC [*] | 17 | ramp rate for load following/AGC (MW/min) |
| RAMP_10 [*] | 18 | ramp rate for 10 minute reserves (MW) |
| RAMP_30 [*] | 19 | ramp rate for 30 minute reserves (MW) |
| RAMP_Q [*] | 20 | ramp rate for reactive power (2 sec timescale) (MVA _r /min) |
| APF [*] | 21 | area participation factor |
| MU_PMAX [†] | 22 | Kuhn-Tucker multiplier on upper P_g limit (u /MW) |
| MU_PMIN [†] | 23 | Kuhn-Tucker multiplier on lower P_g limit (u /MW) |
| MU_QMAX [†] | 24 | Kuhn-Tucker multiplier on upper Q_g limit (u /MVA _r) |
| MU_QMIN [†] | 25 | Kuhn-Tucker multiplier on lower Q_g limit (u /MVA _r) |

^{*} Not included in version 1 case format.

[†] Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .

[‡] Used to determine voltage setpoint for optimal power flow only if `opf.use_vg` option is non-zero (0 by default). Otherwise generator voltage range is determined by limits set for corresponding bus in `bus` matrix.

Table B-3: Branch Data (`mpc.branch`)

| name | column | description |
|------------------------|--------|---|
| F_BUS | 1 | “from” bus number |
| T_BUS | 2 | “to” bus number |
| BR_R | 3 | resistance (p.u.) |
| BR_X | 4 | reactance (p.u.) |
| BR_B | 5 | total line charging susceptance (p.u.) |
| RATE_A* | 6 | MVA rating A (long term rating), set to 0 for unlimited |
| RATE_B* | 7 | MVA rating B (short term rating), set to 0 for unlimited |
| RATE_C* | 8 | MVA rating C (emergency rating), set to 0 for unlimited |
| TAP | 9 | transformer off nominal turns ratio, if non-zero (taps at “from” bus, impedance at “to” bus, i.e. if $r = x = b = 0$, $tap = \frac{ V_f }{ V_t }$; $tap = 0$ used to indicate transmission line rather than transformer, i.e. mathematically equivalent to transformer with $tap = 1$) |
| SHIFT | 10 | transformer phase shift angle (degrees), positive \Rightarrow delay |
| BR_STATUS | 11 | initial branch status, 1 = in-service, 0 = out-of-service |
| ANGMIN [†] | 12 | minimum angle difference, $\theta_f - \theta_t$ (degrees) |
| ANGMAX [†] | 13 | maximum angle difference, $\theta_f - \theta_t$ (degrees) |
| PF [‡] | 14 | real power injected at “from” bus end (MW) |
| QF [‡] | 15 | reactive power injected at “from” bus end (MVar) |
| PT [‡] | 16 | real power injected at “to” bus end (MW) |
| QT [‡] | 17 | reactive power injected at “to” bus end (MVar) |
| MU_SF [§] | 18 | Kuhn-Tucker multiplier on MVA limit at “from” bus (u /MVA) |
| MU_ST [§] | 19 | Kuhn-Tucker multiplier on MVA limit at “to” bus (u /MVA) |
| MU_ANGMIN [§] | 20 | Kuhn-Tucker multiplier lower angle difference limit (u /degree) |
| MU_ANGMAX [§] | 21 | Kuhn-Tucker multiplier upper angle difference limit (u /degree) |

* Used to specify branch flow limits. By default these are limits on apparent power with units in MVA. However, the '`opf.flow_lim`' option can be used to specify that the limits are active power or current, in which case the ratings are specified in MW or ($\text{kA} \cdot V_{\text{basekV}}$), respectively. For current this is equivalent to an MVA value at a 1 p.u. voltage.

[†] Not included in version 1 case format. The voltage angle difference is taken to be unbounded below if $\text{ANGMIN} \leq -360$ and unbounded above if $\text{ANGMAX} \geq 360$. If both parameters are zero, the voltage angle difference is unconstrained.

[‡] Included in power flow and OPF output, ignored on input.

[§] Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .

Table B-4: Generator Cost Data[†] (`mpc.gencost`)

| name | column | description |
|----------|--------|---|
| MODEL | 1 | cost model, 1 = piecewise linear, 2 = polynomial |
| STARTUP | 2 | startup cost in US dollars* |
| SHUTDOWN | 3 | shutdown cost in US dollars* |
| NCOST | 4 | number $N = n + 1$ of data points defining an n -segment piecewise linear cost function, or of coefficients defining an n -th order polynomial cost function |
| COST | 5 | parameters defining total cost function $f(p)$ begin in this column, units of f and p are \$/hr and MW (or MVA _r), respectively (MODEL = 1) \Rightarrow $p_1, f_1, p_2, f_2, \dots, p_N, f_N$ where $p_1 < p_2 < \dots < p_N$ and the cost $f(p)$ is defined by the coordinates $(p_1, f_1), (p_2, f_2), \dots, (p_N, f_N)$ of the end/break-points of the piecewise linear cost (MODEL = 2) \Rightarrow c_n, \dots, c_1, c_0 N coefficients of n -th order polynomial cost function, starting with highest order, where cost is $f(p) = c_n p^n + \dots + c_1 p + c_0$ |

[†] If `gen` has n_g rows, then the first n_g rows of `gencost` contain the costs for active power produced by the corresponding generators. If `gencost` has $2n_g$ rows, then rows $n_g + 1$ through $2n_g$ contain the reactive power costs in the same format.

* Not currently used by any MATPOWER functions.

Table B-5: DC Line Data* (`mpc.dcline`)

| name | column | description |
|-----------------------|--------|--|
| F_BUS | 1 | “from” bus number |
| T_BUS | 2 | “to” bus number |
| BR_STATUS | 3 | initial branch status, 1 = in-service, 0 = out-of-service |
| PF [†] | 4 | real power flow at “from” bus end (MW), “from” → “to” |
| PT [†] | 5 | real power flow at “to” bus end (MW), “from” → “to” |
| QF [†] | 6 | reactive power injected into “from” bus (MVar) |
| QT [†] | 7 | reactive power injected into “to” bus (MVar) |
| VF | 8 | voltage magnitude setpoint at “from” bus (p.u.) |
| VT | 9 | voltage magnitude setpoint at “to” bus (p.u.) |
| PMIN | 10 | if positive (negative), lower limit on PF (PT) |
| PMAX | 11 | if positive (negative), upper limit on PF (PT) |
| QMINF | 12 | lower limit on reactive power injection into “from” bus (MVar) |
| QMAXF | 13 | upper limit on reactive power injection into “from” bus (MVar) |
| QMINT | 14 | lower limit on reactive power injection into “to” bus (MVar) |
| QMAXT | 15 | upper limit on reactive power injection into “to” bus (MVar) |
| LOSS0 | 16 | coefficient l_0 of constant term of linear loss function (MW) |
| LOSS1 | 17 | coefficient l_1 of linear term of linear loss function (MW/MW) ($p_{\text{loss}} = l_0 + l_1 p_f$, where p_f is the flow at the “from” end) |
| MU_PMIN [‡] | 18 | Kuhn-Tucker multiplier on lower flow limit at “from” bus (u/MW) |
| MU_PMAX [‡] | 19 | Kuhn-Tucker multiplier on upper flow limit at “from” bus (u/MW) |
| MU_QMINF [‡] | 20 | Kuhn-Tucker multiplier on lower VAr limit at “from” bus (u/MVar) |
| MU_QMAXF [‡] | 21 | Kuhn-Tucker multiplier on upper VAr limit at “from” bus (u/MVar) |
| MU_QMINT [‡] | 22 | Kuhn-Tucker multiplier on lower VAr limit at “to” bus (u/MVar) |
| MU_QMAXT [‡] | 23 | Kuhn-Tucker multiplier on upper VAr limit at “to” bus (u/MVar) |

* Requires explicit use of `toggle.dcline`.

[†] Output column, value updated by power flow or OPF (except PF in case of simple power flow).

[‡] Included in OPF output, typically not included (or ignored) in input matrix. Here we assume the objective function has units u .