

ภาคผนวก ก.

MATPOWER

A MATLAB™ Power System Simulation Package

ข้อกำหนดในการเขียนไฟล์ของข้อมูลที่ใช้ในการคำนวณของโปรแกรม MATPOWER นั้น จะขอแนะนำเสนอเป็นภาษาอังกฤษ เพื่อความสะดวกถูกต้องไว้ให้มากที่สุด

1. Data File Format

The data file used by MATPOWER are simply MATLAB M-file which define and return the variables baseMVA , bus , branch , gen , areas , and gencost. The baseMVA variables is a scalar and the rest are matrices. Each row in the matrix corresponds to a single bus , branch , or generator. The columns are similar to the columns in the standard I E E E a n d P T I f o r m a t .

1 . 1 B u s D a t a F o r m a t :

- | | |
|-----|--|
| 1 | bus number |
| 2 | bus type |
| | PQ bus = 1 |
| | PV bus = 2 |
| | Reference bus = 3 |
| 3 | Pd , real power demand (MW) |
| 4 | Qd , reactive power demand (MVA _r) |
| 5 | Gs , shunt conductance (Mw) at V = 1.0 p.u. |
| 6 | Bs , shunt susceptance (MVA _r) at V = 1.0 p.u. |
| 7 | area number |
| 8 | V _m , voltage magnitude (p.u.) |
| 9 | V _a , voltage angle (degree) |
| (-) | (bus name) |
| 10 | baseKV , base voltage (kv) |
| 11 | zone , loss zone |
| (+) | 12 maxV _m , maximum voltage magnitude (p.u.) |
| (+) | 13 minV _m , minimum voltage magnitude (p.u.) |

1 . 2 G e n e r a t o r D a t a F o r m a t :

- 1 bus number
- (-) (machine identifier)
- 2 Pg , real power output (MW)
- 3 Qg , reactive power output (MVAr)
- 4 Qmax , maximum reactive power output (MVAr)
- 5 Qmin , minimum reactive power output (MVAr)
- 6 Vg , voltage magnitude setpoint (p.u.)
- (-) (remote controlled bus index)
- 7 mBase , total MVA base of this machine, defaults to baseMVA
- (-) (machine impedance, p.u. on mBase)
- (-) (step up transformer impedance, p.u. on mBase)
- (-) (step up transformer off nominal turns ratio)
- 8 status, > 0 - machine in service
 <= 0 - machine out of service
- (-) (% of total VAR's to come from this gen in order to hold V at remote bus
 controlled by several generators)
- 9 Pmax, maximum real power output (MW)
- 10 Pmin, minimum real power output (MW)

1.3 Branch Data Format :

- 1 f, from bus number
- 2 t, to bus number
- (-) (circuit identifier)
- 3 r, resistance (p.u.)
- 4 x, reactance (p.u.)
- 5 b, total line charging susceptance (p.u.)
- 6 rateA, MVA rating A (long term rating)
- 7 rateB, MVA rating B (short term rating)
- 8 rateC, MVA rating C (emergency rating)
- 9 ratio, transformer off nominal turns ratio (= 0 for lines)
 (taps at 'from' bus, impedance at 'to' bus, i.e. ratio = Vf / Vt)

- 10 angle, transformer phase shift angle (degrees)
- (-) (Gf, shunt conductance at from bus p.u.)
- (-) (Bf, shunt susceptance at from bus p.u.)
- (-) (Gt, shunt conductance at to bus p.u.)
- (-) (Bt, shunt susceptance at to bus p.u.)
- 11 initial branch status, 1 - in service, 0 - out of service

1.4 Area Data Format :

- 1 i, area number
- 2 price_ref_bus, reference bus for that area

1.5 Generator Cost Data Format

NOTE: If gen has n rows, then the first n rows of gencost contain the cost for active power produced by the corresponding generators. If gencost has 2*n rows then rows n+1 to 2*n contain the reactive power costs in the same format.

- 1 model, 1 - piecewise linear, 2 - polynomial
- 2 startup, startup cost in US dollars
- 3 shutdown, shutdown cost in US dollars
- 4 n, number of cost coefficients to follow for polynomial cost function, or number of data points for piecewise linear
- 5 and following, cost data defining total cost function

For polynomial cost :

$$c_2, c_1, c_0$$

where the polynomial is $c_0 + c_1 * P + c_2 * P^2$

For piecewise linear cost:

$$x_0, y_0, x_1, y_1, x_2, y_2, \dots$$

where $x_0 < x_1 < x_2 < \dots$ and the points $(x_0, y_0), (x_1, y_1),$

$(x_2, y_2), \dots$ are the end- and break-points of the cost function.

2. MATPOWER Option

MPOPTION Used to set and retrieve a MATPOWER options vector.

opt = mption
 returns the default options vector

opt = mption(name1, value1, name2, value2, ...)
 returns the default options vector with new values for up to 7 options, name# is the name of an option, and value# is the new value. Example: options = mption('PF_ALG', 2, 'PF_TOL', 1e-4)

opt = mption(opt, name1, value1, name2, value2, ...)
 same as above except it uses the options vector opt as a base instead of the default options vector.

The currently defined options are as follows:

Idx	NAME, default	description [options]
---	-----	-----
power flow options		
1	PF_ALG, 1	power flow algorithm [1 - Newton's method] [2 - Fast-Decoupled (XB version)] [3 - Fast-Decoupled (BX version)] [4 - Gauss Seidel]
2	PF_TOL, 1e-8	termination tolerance on per unit P & Q mismatch
3	PF_MAX_IT, 10	maximum number of iterations for Newton's method
4	PF_MAX_IT_FD, 30	maximum number of iterations for fast decoupled method
5	PF_MAX_IT_GS, 1000	maximum number of iterations for Gauss-Seidel method
6	ENFORCE_Q_LIMS, 0	enforce gen reactive power limits, at expense of V [0 or 1]
10	PF_DC, 0	use DC power flow formulation, for power flow and OPF [0 - use AC formulation & corresponding algorithm opts] [1 - use DC formulation, ignore AC algorithm options]

OPF options

11 - OPF_ALG, 0	algorithm to use for OPF
[0 - choose best default solver available in the]
[following order, 500, 520 then 100/200]
[Otherwise the first digit specifies the problem]
[formulation and the second specifies the solver,]
[as follows, (see the User's Manual for more details)]
[100 - standard formulation (old), constr]
[120 - standard formulation (old), dense LP]
[140 - standard formulation (old), sparse LP (relaxed)]
[160 - standard formulation (old), sparse LP (full)]
[200 - CCV formulation (old), constr]
[220 - CCV formulation (old), dense LP]
[240 - CCV formulation (old), sparse LP (relaxed)]
[260 - CCV formulation (old), sparse LP (full)]
[500 - generalized formulation, MINOS]
[520 - generalized formulation, fmincon]
[See the User's Manual for details on the formulations.]
12 - OPF_ALG_POLY, 100	default OPF algorithm for use with polynomial cost functions (used only if no solver available for generalized formulation)
13 - OPF_ALG_PWL, 200	default OPF algorithm for use with piece-wise linear cost functions (used only if no solver available for generalized formulation)
14 - OPF_POLY2PWL_PTS, 10	number of evaluation points to use when converting from polynomial to piece-wise linear costs
16 - OPF_VIOLATION, 5e-6	constraint violation tolerance
17 - CONSTR_TOL_X, 1e-4	termination tol on x for copf & fmincopf
18 - CONSTR_TOL_F, 1e-4	termination tol on F for copf & fmincopf
19 - CONSTR_MAX_IT, 0	max number of iterations for copf & fmincopf
[0 => 2*nb + 150]
20 - LPC_TOL_GRAD, 3e-3	termination tolerance on gradient for lpopf

[(same options for OUT_LINE_LIM, OUT_PG_LIM, OUT_QG_LIM)]

- 40 - OUT_LINE_LIM, 1 control output of line limit info
 - 41 - OUT_PG_LIM, 1 control output of gen P limit info
 - 42 - OUT_QG_LIM, 1 control output of gen Q limit info
 - 43 - OUT_RAW, 0 print raw data for Perl database interface code
- [0 or 1]

other options

- 51 - SPARSE_QP, 1 pass sparse matrices to QP and LP solvers if possible
- [0 or 1]

MINOPF options

- 61 - MNS_FEASTOL, 0 (1E-3) primal feasibility tolerance, set to value of OPF_VIOLATION by default
- 62 - MNS_ROWTOL, 0 (1E-3) row tolerance set to value of OPF_VIOLATION by default
- 63 - MNS_XTOL, 0 (1E-3) x tolerance set to value of CONSTR_TOL_X by default
- 64 - MNS_MAJDAMP, 0 (0.5) major damping parameter
- 65 - MNS_MINDAMP, 0 (2.0) minor damping parameter
- 66 - MNS_PENALTY_PARM, 0 (1.0) penalty parameter
- 67 - MNS_MAJOR_IT, 0 (200) major iterations
- 68 - MNS_MINOR_IT, 0 (2500) minor iterations
- 69 - MNS_MAX_IT, 0 (2500) iterations limit
- 70 - MNS_VERBOSITY, -1
 - [-1 - controlled by VERBOSE flag (0 or 1 below)]
 - [0 - print nothing]
 - [1 - print only termination status message]
 - [2 - print termination status and screen progress]
 - [3 - print screen progress, report file (usually fort.9)]
- 71 - MNS_CORE, 1200 * nb + 5000
- 72 - MNS_SUPBASIC_LIM, 0 (2*ng) superbasics limit
- 73 - MNS_MULT_PRICE, 0 (30) multiple price

3. Running an Optimal Power Flow

To run optimal power flow an 118 – bus system whose data is in case118.m , with the default algorithm option , at the MATLAB prompt , type :

```
>> runopf('case118')
```

4. Summary of the Files

runopf.m	runs an optimal power flow
case118.m	IEEE 118 bus system
opf_form.m	return code for formulation given OPF algorithm code
opf_slvr.m	return code for solver given OPF algorithm code
opf.m	top – level OPF solver routine
poly2pwl.m	creates piecewise linear approximation to polynomial cost function
ppcost.m	splits gencost into real and reactive power costs
totcost.m	computes total cost for given dispatch