Introduction

EMSO is the acronym for Environment for Modeling, Simulation, and Optimization. It is a dynamic simulator for general processes.

This is the EMSO quick reference card, for detailed documentation please consult the EMSO User's Guide.

You can find the latest version of this document, the EMSO User's Guide, and the software itself at `http://www.enq.ufrgs.br/alsoc`.

EMSO Types – `types.mso`

Name	Base	Default	Lower	Upper	Unit
Real	Built-In	1	-∞	∞	$none$
coefficient	Real	1	-50	50	$none$
Constant	Real	10	-5e3	5e3	$none$
positive	Real	1.0	-1e-6	∞	$none$
negative	Real	-1.0	-∞	1e-6	$none$
fraction	Real	0.5	-1e-6	1.00001	$none$
percent	Real	50	0	100	$none$
control_signal	Real	1	-1e9	1e9	$none$
efficiency	Real	0.5	0	1	$none$
pressure	Real	1	1e-30	5e7	atm
press_delta	pressure	0.01	-5e6	5e7	atm
head_mass	Real	50	-1e6	1e6	kJ/kg
head	Real	50	-1e6	1e6	$kJ/kmol$
temperature	Real	300	27	5273	K
temp_delta	temperature	30	-1e3	1e3	K
time_h	positive	1	-1e-6	1e4	h
time_min	time_h	1	-1e-6	∞	min
time_sec	time_h	1	-1e-6	∞	s
frequency	positive	1	-1e-6	100	$1/s$
angle	Real	0	-7	7	rad
area	positive	1	-1e-6	1e6	m^2
length	positive	1	-1e-6	5e6	m
length_delta	length	1	-1e3	∞	m
volume	positive	10	-1e-6	1e3	m^3
volume_mol	positive	10	-1e-6	1e6	m^3/mol
volume_mass	positive	10	-1e-6	1e30	m^3/kg
current	positive	10	-1e-6	1e12	A
charge	Real	10	-1e12	1e12	C
capacitance	positive	10	-1e-6	1e12	F
indutance	positive	10	-1e-6	1e12	$V*s/A$
voltage	positive	100	-1e-6	1e9	V
resistance	positive	50	-1e-6	800	ohm
potency	Real	10	-1e3	1e3	kW
currency	Real	0	-∞	∞	US$
mass	positive	2.5	-1e-6	1e6	kg
mol	positive	2.5	-1e-6	1e6	$kmol$
molweight	Real	75	1	1e8	$kg/kmol$
molweight_inv	Real	1	0	1e30	$kmol/kg$
dens_mol	Real	1	1e-30	5e3	$kmol/m^3$
dens_mass	Real	1e3	1e-30	5e5	kg/m^3
conc_mol	dens_mol	1e-3	1e-30	5e3	$kmol/m^3$
inv_conc_mol	Real	0.05	2e-4	1e30	$m^3/kmol$
conc_mass	dens_mass	1e3	1e-30	5e3	kg/m^3
inv_conc_mass	Real	1e-3	2e-6	1e30	m^3/kg
reaction_mol	Real	10	-1e6	1e6	$kmol/h/m^3$
reaction_mass	Real	1e3	-1e3	1e9	$kg/h/m^3$
cp_mass	Real	1.0	0.1	10.0	$kJ/kg/K$
cp_mol	Real	100	1	1e3	$J/mol/K$
cv_mol	Real	100	1	1e3	$J/mol/K$
enth_mass	Real	500	-1e4	1e4	kJ/kg
enth_mol	Real	500	-1e9	1e9	J/mol
entr_mol	Real	0	-1e4	1e4	$kJ/kmol/K$
entr_mass	Real	0	-1e4	1e4	$kJ/kg/K$
heat_reaction	Real	1e3	-1e8	1e8	$kJ/kmol$
heat_rate	Real	1e3	-1e11	1e11	J/s
heat_flux	Real	1	-1e5	1e5	kW/m^2
heat_trans_coeff	positive	1	-1e-6	1e3	$kW/m^2/K$
energy	Real	1e4	-1e11	1e11	kJ
energy_mass	Real	1e4	-1e15	1e15	kJ/kg
energy_mol	Real	1e4	-1e15	1e15	$kJ/kmol$
power	Real	10	-1e8	1e8	kW
flow_mass	positive	1e3	-1e-6	1e10	kg/h

Name	Base	Default	Lower	Upper	Unit
flow_mass_delta	flow_mass	1e3	-1e10	1e10	kg/h
flow_vol	positive	1	-1e-6	1e12	m^3/h
flow_vol_delta	flow_vol	1	-1e12	1e12	m^3/h
flow_mol	positive	10	-1e-6	1e8	$kmol/h$
flow_mol_delta	flow_mol	10	-1e8	1e8	$kmol/h$
flux_mol	positive	1	-1e-6	1e4	$kmol/s/m^2$
flux_mol_delta	flux_mol	1	-1e4	1e4	$kmol/s/m^2$
flux_mass	positive	1	-1e-6	1e6	$kg/s/m^2$
flux_mass_delta	flux_mass	1	-1e6	1e6	$kg/s/m^2$
flux_vol	positive	1	-1e-6	1e4	$m^3/s/m^2$
flux_vol_delta	flux_vol	1	-1e4	1e4	$m^3/s/m^2$
vel_angular	Real	1e3	-1e5	1e5	rpm
rotation	Real	5e2	-1e4	1e4	rad/s
velocity	Real	1	-1e5	1e5	m/s
velocity_delta	velocity	0	-1e3	-1e3	m/s
acceleration	Real	9.81	-1e3	1e3	m/s^2
fricfactor	Real	0.05	1e-5	2e3	$none$
moment_inertia	Real	100	1e-3	1e4	$kg*m^2$
hookes_const	Real	1e5	1	1e10	N/m
conductivity	Real	1.0	1e-3	500	$W/m/K$
diffusivity	positive	1e-3	-1e-6	1	cm^2/s
fugacity	positive	0.5	-1e-6	10	$none$
viscosity	Real	1	1e-30	1e5	cP
vol_mol	volume_mol	10	-1e-6	1e6	m^3/mol
vol_mass	volume_mass	10	-1e-6	1e6	m^3/kg
spec_surface_vol	positive	1e5	-1e-6	1e15	m^2/m^3
spec_surface_mass	positive	100	-1e-6	1e12	m^2/kg
surf_tens	positive	0.05	-1e-6	1	N/m
act_coeff	positive	1	-1e-6	30	$none$
ph	Real	7	-5	20	$none$

Built-In Functions

Function	Meaning
abs(Z)	Returns the magnitude or absolute value of Z
max(Z)	Returns the maximum value of Z
min(Z)	Returns the minimum value of Z
sign(Z)	Returns the signal of Z (-1 if Z $<$ 0 e 1 if Z$>$ 0
round(Z)	Returns the small integer value of Z
sinh(Z)	Returns the hyperbolic sine of Z
cosh(Z)	Returns the hyperbolic cosine of Z
tanh(Z)	Returns the hyperbolic tangent of Z
coth(Z)	Returns the hyperbolic cotangent of Z
sin(Z)	Returns the sine of Z
cos(Z)	Returns the cosine of Z
tan(Z)	Returns the tangent of Z
asin(Z)	Returns the angle whose sine is Z
acos(Z)	Returns the angle whose cosine is Z
diff(Z)	Returns the derivative of Z with respect to time
atan(Z)	Returns the angle whose tangent is Z
sum(Z)	Returns the sum of components of a vector or matrix Z
prod(Z)	Returns the product of a vector or matrix Z
sumt(Z)	Returns the transpose sum of components of a vector or matrix Z
prodt(Z)	Returns the transpose product of components of a vector or matrix Z
transp(Z)	Returns the transpose of a matrix Z
exp(Z)	Returns the exponential function, e raised to the power Z
log(Z)	Returns the base 10 logarithm of Z
ln(Z)	Returns the natural logarithm (base e) of Z
sqrt(Z)	Returns the square root of Z

Units Of Measurement (UOM)

Fundamental Units

m	length in meters
kg	mass in kilogram
s	time in seconds
K	temperature in Kelvin
A	eletric current in Ampere
mol	the amount of substance in mole
cd	the luminous intensity in Candela
rad	angle measure in radian
US$	money in dollar (USA)

Derived Units

Acceleration of Gravity		
ga	9.80665*m/s^2	standard acceleration of gravity
Angle		
arcs	4.8481368111e-6*rad	arcsecond
arcmin	2.90888208666e-4*rad	arcminute
grad	1.57079632679e-2*rad	grad
deg	1.74532925199e-2*rad	degree
Area		
acre	4046.87260987*m^2	acre
a	100*m^2	are
ha	10000*m^2	hectare
b	1e-28*m^2	barn
Eletric		
Wb	kg*m^2/A/s^2	weber
T	kg/A/s^2	tesla
S	A^2*s^3/kg/m^2	siemens
mho	A^2*s^3/kg/m^2	mho
Fdy	96487*A*s	faraday
F	A^2*s^4/kg/m^2	farad
ohm	kg*m^2/A^2/s^3	ohm
C	A*s	unit of relative current for batteries
V	kg*m^2/A/s^3	volt
Energy		
J	kg*m^2/s^2	joule
kJ	1e3*J	kilojoule
MJ	1e6*J	megajoule
GJ	1e9*J	gigajoule
eV	1.60217733e-19*J	electronvolt
MeV	1e6*eV	megaelectronvolt
therm	105506000*J	therm
Btu	1055.05585262*J	British thermal unit
cal	4.1868*J	calorie
kcal	1e3*cal	kilo calorie
erg	1e-7*J	erg
Force		
N	kg*m/s^2	newton
pdl	0.138254954376*N	poundal
lbf	4.44822161526*N	pounds of force
kip	4448.22161526*N	kip
gf	0.00980665*N	gram force
kgf	1e3*gf	kilogram force
dyn	0.00001*N	dyne
Length		
cm	1e-2*m	centimeter
mm	0.1*cm	millimeter
fermi	1e-15*m	fermi
$\AA$	1e-10*m	angstrom
μ	1e-6*m	micro
mil	2.54e-5*m	mil
ftUS	0.304800609601*m	international foot
fath	1.82880365761*m	fathom
rd	5.02921005842*m	rod
chain	20.1168402337*m	chain
miUS	1609.34721869*m	US statute miles
nmi	1852*m	nautical mile
mi	1609.344*m	International Mile
km	1000*m	Kilometer
au	1.495979e11*m	Astronomical Unit
lyr	9.46052840488e15*m	light year
pc	3.08567818585e16*m	parsec
Mpc	3.08567818585e22*m	megaparsec
in	0.0254*m	inch
ft	0.3048*m	foot
yd	0.9144*m	yard
Mass		
u	1.6605402e-27*kg	atomic mass unit
grain	0.00006479891*kg	grain
ct	0.0002*kg	carat
ozt	0.0311034768*kg	troy ounce
t	1000*kg	tonne
tonUK	1016.0469088*kg	ton (UK)
ton	907.18474*kg	ton
lbt	0.3732417216*kg	troy pound

slug	14.5939029372*kg	slug
oz	0.028349523125*kg	ounce
lb	0.45359237*kg	pound
g	kg/1000	gram
kmol	1e3*mol	kilomole
lbmol	453.59237*mol	pound mole
Money		
R$	US$/3.05	Brazilian money (Real)
Power		
W	$kg * m^2/s^3$	watt
kW	1e3*W	Kilowatt
MW	1e6*W	megawatt
hp	745.699871582*W	horsepower
Pressure		
Pa	kg/m/s^2 pascal	
kPa	1e3*Pa	Kilopascal
MPa	1e3*kPa	megapascal
inH2O	248.84*Pa	inch of water column
inHg	3386.38815789*Pa	inch of mercury
mmHg	133.322368421*Pa	millimeter of mercury
torr	133.322368421*Pa	torr
psi	6894.75729317*Pa	pound per square inch
bar	1e5*Pa	bar
atm	101325*Pa	atmosphere
Radiation		
R	0.000258*A*s/kg	R
Ci	3.7e10/s	curie
Bq	1/s	becquerel
Sv	m^2/s^2	sievert
rem	0.01*m^2/s^2	rem
Gy	m^2/s^2	gray
Temperature		
degR	K/1.8	degree Rankine
Time		
Hz	1/s	hertz
min	60*s	minute
rpm	1/min	revolution per minute
h	60*min	hour
d	24*h	day
yr	31556925.9744*s	year
Velocity		
c	299792458*m/s	velocity (or "speed") of light
knot	0.514444444444*m/s	knot
mph	0.44704*m/s	mile per hour
kph	0.277777777778*m/s	kilometer per hour
Viscosity		
St	0.0001*m^2/s	stoke
P	0.1*kg/m/s	poise
cP	0.001*kg/m/s	centipoise
Volume		
st	m^3	Stere
fbm	0.002359737216*m^3	board foot
pk	0.0088097675*m^3	peck
bu	0.03523907*m^3	bushel
bbl	0.158987291928*m^3	barrel
trp	4.92892159375e-6*m^3	teaspoon
tbsp	1.47867647813e-5*m^3	tablespoon
ozUK	2.8413075e-5*m^3	fluid ounce (UK)
ozfl	2.95735295625e-5*m^3	fluid ounce
cu	2.365882365e-4*m^3	US Cup
l	1e-3*m^3	liter
ml	1e-3*l	milliliter
pt	0.000473176473*m^3	pint
qt	0.000946352946*m^3	quart
gal	0.00378541178*m^3	gallon
galC	0.00454609*m^3	imperial gallon
galUK	0.004546092*m^3	gallon (UK)

Modeling Language

Conventions:

- Every command enclosed by [ ] is optional.
- The code `a | b | c` means a or b or c.
- Every command between < > is an emphasis in that command.
- `FlowSheet` accepts all commands of a `Model`
- `Optimization` and `Estimation` accept all commands of a `FlowSheet`

The FlowSheet Entity

```
FlowSheet  FlowSheetName [as base]
DEVICES
DevName as base [(Brief="brief")];
CONNECTIONS
DevName.varName<in> to DevName.varName<out>;
SET
ParName = <expression involving only parameters> | value ;
SPECIFY
["equation name"] DevName.varName = expression | value;
INITIAL
["equation name"] DevName.varName = expression | value;
OPTIONS
      Dynamic     = false | true;
      TimeStart   = value;
      TimeStep    = value;
      TimeEnd     = value;
      TimeUnit    = 'unit';
      SparseAlgebra = false | true;
      GuessFile   = "File Name";
      InitialFile = "File Name";
NLASolver(File="sundials" |"nlasolver",
      RelativeAccuracy=1e-3, AbsoluteAccuracy=1e-6, MaxIterations =100);
DAESolver(File="sundials"|"dassl"|"dasslc"|"mebdf",
      RelativeAccuracy=1e-3,AbsoluteAccuracy=1e-6,EventAccuracy =1e-2);
NLPSolver(File="complex"|"ipopt_emso",RelativeAccuracy=1e-3);
end
```

The Model Entity

```
Model ModelName [as base]
ATTRIBUTES
Icon     ="file name";
Pallete = false | true;
Brief    ="A shortcut information about the model";
Info     ="A detailed information about the model";
PARAMETERS
[outer] ParName;
[outer] ParName as base;
[outer] ParName as base(Brief="brief",Default=value,Lower=value,Upper=
    value,DisplayUnit='unit');
[outer] size as Integer(Brief="brief",Default=value,Lower=value,Upper=
    value);
[outer] ParName(size) as base(Brief="brief",Default=value,Lower=value,
    Upper=value,DisplayUnit='unit');
[outer] ParName(dim1,dim2,...) as base(Brief="brief",Default=value,Lower=
    value,Upper=value,DisplayUnit='unit');
ParName as Switcher(Brief="brief",Valid=["value1","value2",...],Default="
    value");
VARIABLES
[in | out] VarName;
[in | out] VarName as base;
[in | out] VarName as ModelBase [(Brief="brief")];
VarName as base(Brief="brief",Default=value,Lower=value,Upper=value,
    DisplayUnit='unit');
VarName(size) as base(Brief="brief",Default=value,Lower=value,Upper=value
    ,DisplayUnit='unit');
VarName(dim1,dim2,...) as base(Brief="brief",Default=value,Lower=value,
    Upper=value,DisplayUnit='unit');
CONNECTIONS
varName<in> to varName<out>;
EQUATIONS
["equation name"] expression = expression;
INITIAL
["equation name"] varName = expression | value;
SET
ParName = <expression involving only parameters> | value;
size = < a integer value >;
end
```

The Optimization Entity

```
Optimization name [as base]
MINIMIZE
      expression1;
      expression2;
MAXIMIZE
      expression3;
      expression4;
EQUATIONS
      expression5 < expression6;
      expression7 > expression8;
FREE
     variable1;
     variable2;
end
```

The Estimation Entity

```
Estimation EstimationName as FlowSheetName
ESTIMATE
#PARAMETER  START       LOWER      UPPER     UNIT
ParName  StartValue  LowerValue  UpperValue  'unit'
EXPERIMENTS
# DATA FILE          WEIGTH
"FileName1.dat"       WeigthValue1;
"FileName2.dat"       WeigthValue2;
. . .
OPTIONS
. . .
end
```

The experimental data – `FileName.dat`

```
  MEASURE   time        Param1      Param2   . . .   ParamN
[ UNIT      'timeUnit'  'unit1'     'unit2'  . . .     'unitN' ]
[ VARIANCE  timeVar     var1        var2     . . .   varN ]
  DATA      Timevalue   value1      value2   . . .   valueN
```

Simple Equations

```
      ["equation name"] expression = expression;
```

Expressions:

```
                algebraic operations:+, -, *, /, ^
                built-In functions: See list above
                plugin Method: PluginName.PluginMethod(arguments)
```

Conditional Equations:

```
      if LogicalExpression then
            ["equation name"] Equations;
      else
            ["equation name"] Equations;
      end

      switch Valid_Switcher_ParName
            case "validName":
            ["equation name"] Equations;
            [when LogicalExpression switchto "Any validName"];
            . . .
      end
```

LogicalExpressions:

```
  Relational operations:
      <, >, <=, >=
  Logical constants:
    true, false
  Logical operations:
     and, or, not, equal
```

Iterative equations:

```
      for index in [StartValue:EndValue]
            ["equation name"] Equations;
      end
```