Solar Design Algorithm

The document below defines the calculations used to design and evaluate a PV system in preparation for creating electrical drawings.

The secondary documents are automatically created from this source:

- A printable PDF document describing the algorithm, with no computer code (SDA_standard.pdf).
- Key computer code used in FSEC's online express drawing creation application (SDA.js).
- A printable PDF document describing the algorithm and it's related computer code (SDA.pdf).

Note: For each section, the symbols are pre-pended by a section name to assist with their use in the computer code, in the form of "section.symbol".

Modules, source circuits, and array

Calculation summary:

Description	Symbol	Calculation
Maximum Power (W)	inverter.dc_voltage_nominal	inverter.mppt_max
Maximum Power Voltage (V)	source.vmp	module.pmp / source.max_power * inverter.dc_voltage_nominal
Maximum Power Current (A)	source.imp	source.max_power / inverter.dc_voltage_nominal
Open-Circuit Voltage (V)	source.voc	1 * array.largest_string
Short-Circuit Current (A)	source.isc	0.6
Maximum Circuit Current (A)	source.i_max	optimizer.max_output_current
Maximum Power (W)	source.max_power	module.pmp * array.largest_string
Source Circuit Maximum Current (A), Isc x 1.25	source.Isc_adjusted	module.isc * 1.25
Maximum system voltage	array.max_sys_voltage	inverter.dc_voltage_nominal
Minimum array voltage (module temp. correction factor)	array.min_voltage	array.smallest_string * module.vmp * (1 + module.tc_vpmax_percent / 1
Maximum Power (W)	array.pmp	array.num_of_modules * module.pmp
Enter Maximum Number of Parallel Source Circuits per Output Circuit (1-2)	array.circuits_per_MPPT	Math.ceil(array.num_of_strings / inverter.mppt_channels)
PV Output Circuit Maximum Current per MPPT (A)	array.combined_isc	source.isc * array.circuits_per_MPPT
Total PV Output Circuit Maximum Current (A)	array.total_isc	optimizer.max_output_current * array.num_of_strings
Maximum module voltage	module.max_voltage	module.voc * (1 + module.tc_voc_percent / 100 * (array.min_temp - 25)
MPPT maximum operating voltage (V)	inverter.mppt_max	

```
inverter.dc_voltage_nominal = inverter.mppt_max;
source.vmp = module.pmp / source.max_power * inverter.dc_voltage_nominal;
source.imp = source.max_power / inverter.dc_voltage_nominal;
source.voc = 1 * array.largest_string;
source.isc = 0.6; //amps
source.i_max = optimizer.max_output_current;
source.max_power = module.pmp * array.largest_string;
source.Isc_adjusted = module.isc * 1.25;
array.max_sys_voltage = inverter.dc_voltage_nominal;
array.min_voltage = inverter.dc_voltage_nominal;
array.pmp = array.num_of_modules * module.pmp;
array.circuits_per_MPPT = Math.ceil( array.num_of_strings / inverter.mppt_channels );
array.combined_isc = source.i_max * array.circuits_per_MPPT;
array.total_isc = optimizer.max_output_current * array.num_of_strings;
module.max_voltage = module.voc * ( 1 + module.tc_voc_percent / 100 * ( array.min_temp - 25));
```

Inverter

If max_ac_ocpd is not provided by the manufacturer, it is calculated as follows:

AC_OCPD_max = max_ac_output_current * 1.25

```
inverter.AC_OCPD_max = sf.if( sf.not( inverter.max_ac_ocpd ), inverter.max_ac_output_current * 1.25, inverter.max_ac_ocpd );
```

The nominal_ac_output_power is selected from fields based on the user selected grid voltage. As an example, if the user selects 240 VAC, then:

nominal_ac_output_power = nominal_ac_output_power_240
max_ac_output_current = max_ac_ouput_current_240

```
inverter.nominal_ac_output_power = inverter['nominal_ac_output_power_'+inverter.grid_voltage];
inverter.max_ac_output_current = inverter['max_ac_output_current_'+inverter.grid_voltage];
```

Array checks

The maximum array voltage is must not exceed the maximum system voltage allowed by the module.

```
error_check.array_test_1 = array.max_sys_voltage > module.max_system_v;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.array_test_1 ){ report_error( 'Maximum system voltage exceeds the modules max system voltage.' );}
```

The maximum array voltage is must not exceed the maximum system voltage allowed by the building code.

```
error_check.array_test_2 = array.max_sys_voltage > array.code_limit_max_voltage;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.array_test_2){ report_error( 'Maximum system voltage exceeds the maximum voltage allows by code.' );}
```

The maximum array voltage must not exceed the maximum system voltage allowed by the inverter.

```
error_check.array_test_3 = array.max_sys_voltage > inverter.vmax;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.array_test_3){ report_error( 'Maximum system voltage exceeds the inverter maximum voltage rating' );}
```

The minimum array voltage must be greater than the inverter minimum operating voltage.

```
error_check.array_test_4 = array.min_voltage <= inverter.voltage_range_min;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.array_test_4){ report_error( 'Minimum Array Vmp is less than the inverter minimum operating voltage.' );}
```

The total array power must be less than 10,000W.

```
error_check.power_check_array = array.pmp > 10000;
// If error check is true, flag system design failure, and report notice to user.
if( error_check.power_check_array ){ report_error( 'Array total power exceeds 10kW' );}
```

The DC array can be oversized relative to the inverter, but the total DC power can not exceed 135% of the inverters AC output power.

```
error_check.current_check_inverter = array.max_power > ( inverter.max_ac_output_current * interconnection.grid_voltage * 1.35 );
// If error check is true, flag system design failure, and report notice to user.
if( error_check.current_check_inverter ){ report_error( 'PV output circuit maximum current exceeds the inverter maximum dc
current per MPPT input.' );}
```

Array source checks

The largest number of optimizers per branch must not exceed the maximum number allowed by the manufacturer.

error_check.optimizer_micro_branch_too_many_modules = array.largest_string > optimizer.max_optis_per_string;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.optimizer_micro_branch_too_many_modules){ report_error('The system has too many inverters per array source circuit.');}

error_check.optimizer_micro_branch_too_few_modules = array.smallest_string < optimizer.min_optis_per_string;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.optimizer_micro_branch_too_few_modules){ report_error('The system has too many inverters per array source circuit.');}

The total nominal module power output for each branch must not exceed the manufacturer's limit.

error_check.optimizer_micro_branch_too_much_power = source.max_power > optimizer.max_power_per_string;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.optimizer_micro_branch_too_much_power){ report_error('The array source power limit has exceeded the manufacturer's limit.');}

Module - Optimizer checks

The module(s) power and voltage must be within the inverter manufacturer's limits.

```
error_check.module_power_too_high = module.pmp > optimizer.rated_max_power;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.module_power_too_high ){ report_error( 'Optimizer is undersized for module.' );}

error_check.module_voltage_too_low = module.vmp < optimizer.mppt_op_range_min ;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.module_voltage_too_low ){ report_error( 'Module does not meet minimum optimizer operating voltage.' );}
error_check.module_voltage_too_high = module.vmp > optimizer.mppt_op_range_max ;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.module_voltage_too_high ){ report_error( 'Module exceeds optimizer operating voltage range.' );}
```

The module's operating voltage must be less than the inverter maximum operating voltage.

```
error_check.module_voltage = module.vmp > inverter.vmax;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.module_voltage ){ report_error( 'Module voltage exceeds inverter maximum.' );}
```

The modules maximum voltage, and the lowest temperature, can not exceed the optimizer's limit.
error_check.module_max_voltage = module.max_voltage > optimizer.max_input_voltage ;
// If error check is true, flag system design failure, and report notice to user.
if(error_check.module_max_voltage){ report_error('Module maximum voltage exceeds the maximum allowed by the optimizer.');}

Conductor and conduit schedule

For string inverters, these are the circuit names:

- Exposed source circuit wiring: DC wires exposed on the roof.
- PV DC source circuits: DC wires in conduit.
- Inverter AC output circuit: AC circuits between the inverter and panel OCPD.

```
var circuit_names = [
  'exposed source circuit wiring',
  'pv dc source circuits',
  'inverter ac output circuit',
];
circuit_names.forEach(function(circuit_name){
  circuits[circuit_name] = {};
});
```

The array temperature adder is found in NEC table 310.15(B)(3)(c), or Table 1 in appendix, with module.array_offset_from_roof as "Distance Above Roof to Bottom of Conduit (in)".

```
// Use Table 1, lookup: module.array_offset_from_roof, return the first column.
circuits['exposed source circuit wiring'].temp_adder = sf.lookup( module.array_offset_from_roof, tables[1] );
```

The maximum current and voltage for the array DC circuits are equal to source.isc and source.voc.

```
circuits['exposed source circuit wiring'].max_current = array.combined_isc;
circuits['exposed source circuit wiring'].max_voltage = source.voc;
circuits['pv dc source circuits'].max_current         = array.combined_isc;
circuits['pv dc source circuits'].max_voltage         = source.voc;
```

The number of DC current carrying conductors is equal to twice the number of strings in the array (array.num_of_strings * 2).
Total conductors adds one more for the ground.

```
circuits['exposed source circuit wiring'].total_cc_conductors = ( array.num_of_strings * 2 );
circuits['exposed source circuit wiring'].total_conductors    = ( array.num_of_strings * 2 ) + 1;
circuits['pv dc source circuits'].total_cc_conductors         = ( array.num_of_strings * 2 );
circuits['pv dc source circuits'].total_conductors            = ( array.num_of_strings * 2 ) + 1;
```

The AC grid voltage is defined by system specifications (user input).

```
circuits['inverter ac output circuit'].max_voltage = inverter.grid_voltage;
```

The maximum AC output is defined by the inverter manufacturer specifications.

```
circuits['inverter ac output circuit'].max_current = inverter.max_ac_output_current;
```

AC conductors numbers are defined by the grid voltage.

```
var conductors_options = {
  '120V': ['ground','neutral','L1'],
  '240V': ['ground','neutral','L1','L2'],
  '208V': ['ground','neutral','L1','L2'],
  '277V': ['ground','neutral','L1'],
  '480V Wye': ['ground','neutral','L1','L2','L3'],
  '480V Delta': ['ground','L1','L2','L3'],
};
inverter.conductors = conductors_options[inverter.grid_voltage+'V'];
inverter.num_conductors = inverter.conductors.length;

circuits['inverter ac output circuit'].total_cc_conductors = inverter.num_conductors - 1;
circuits['inverter ac output circuit'].total_conductors = inverter.num_conductors;
```

For each circuit, calculate the following.

```
circuit_names.forEach(function(circuit_name, i){
  var circuit = circuits[circuit_name];
  circuit.id = i;

  circuit.power_type = sf.index( ['DC', 'DC', 'AC'], circuit.id );
  // If temperature adder is not defined, set it to 0 for use in further calculations.
  circuit.temp_adder = sf.if( circuit.temp_adder, circuit.temp_adder, 0 );
```

Select circuit details based on code requirements and best practices.

Exposed source circuit wiring:

- Conductor: 'DC+/DC-, EGC'
- Location: 'Free air'
- Material: 'CU'
- Type: 'PV Wire, bare'
- Volt rating: 600
- Wet temp rating: 90
- Conduit type: '-'

PV DC source circuits:

- Conductor: 'DC+/DC-, EGC'
- Location: 'Conduit/Exterior'
- Material: 'CU'
- Type: 'THWN-2'
- Volt rating: 600
- Wet temp rating: 90

- Conduit type: 'Metallic'

Inverter ac output circuit:

- Conductor: 'L1/L2, N, EGC'
- Location: 'Conduit/Interior'
- Material: 'CU'
- Type: 'THWN-2'
- Volt rating: 600
- Wet temp rating: 90
- Conduit type: 'Metallic'

```
circuit.conductor = sf.index( ['DC+/DC-, EGC', 'DC+/DC-, EGC', 'L1/L2, N, EGC'], circuit.id );
circuit.location = sf.index( ['Free air', 'Conduit/Exterior', 'Conduit/Interior'], circuit.id );
circuit.material = 'CU';
circuit.type = sf.index( ['PV Wire, bare', 'THWN-2', 'THWN-2'], circuit.id );
circuit.volt_rating = 600;
circuit.wet_temp_rating = 90;
circuit.conduit_type = sf.index( ['-', 'Metallic', 'Metallic'], circuit.id );
```

The array maximum temperature of the array is equal to the 2% maximum temperature at the install location, or nearest weather station.
For a state wide design, the largest maximum temperature for the state is used.
Rooftop array circuits also have a temperature adjustment defined above.

```
  circuit.max_conductor_temp = array.max_temp + circuit.temp_adder;
  // Use Table 2, lookup: circuit.max_conductor_temp, return the first column.
  circuit.temp_correction_factor = sf.lookup( circuit.max_conductor_temp, tables[2] );
  // Use Table 3, lookup: circuit.total_cc_conductors, return the first column.
  circuit.conductors_adj_factor = sf.lookup( circuit.total_cc_conductors , tables[3] );
```

There are three options to calculate the minimum required current:

1. circuit.max_current * 1.25;
2. circuit.max_current / (circuit.temp_correction_factor * circuit.conductors_adj_factor);
3. circuit.max_current * 1.25 * 1.25;

```
  circuit.min_req_cond_current_1 = circuit.max_current * 1.25;
  circuit.min_req_cond_current_2 = circuit.max_current / ( circuit.temp_correction_factor * circuit.conductors_adj_factor );
  circuit.min_req_cond_current_3 = circuit.max_current * 1.25 * 1.25;
```

For AC circuits, the maximum of 1 and 2 is used. For DC circuits, the maximum of 2 and 3 is used.

```
  circuit.min_req_cond_current    = sf.max( circuit.min_req_cond_current_1, circuit.min_req_cond_current_2 );
  circuit.min_req_OCPD_current_DC = sf.max( circuit.min_req_cond_current_2, circuit.min_req_cond_current_3  );
  circuit.min_req_OCPD_current = sf.if( circuit.power_type === 'DC', circuit.min_req_OCPD_current_DC,
circuit.min_req_cond_current_1);
```

For strings per MPP tracker of 2 or less, or for inverters with built in OCPD, additional DC OCPD is not required. The AC circuits do require OCPD at the panel.

```
  circuit.OCPD_required = sf.index( [false, false, true ], circuit.id );
  circuit.ocpd_type = sf.index( ['NA', 'PV Fuse', 'Circuit Breaker'], circuit.id );
```

Choose the OCPD that is greater or equal to the minimum required current.

```
  // Use Table 9, lookup: circuit.min_req_OCPD_current, find the next highest or matching value, return the index column.
  circuit.OCPD = sf.lookup( circuit.min_req_OCPD_current, tables[9], 0, true, true);
  if( circuit_name === 'inverter ac output circuit' ){ inverter.OCPD = circuit.OCPD; }
```

Choose the conductor with a current rating that is greater than the OCPD rating from NEC table 310.15(B)(16).
NEC chapter 9 table 8 provides more details on the conductor. For DC circuits, 10 AWG wire is used as a best practice.

```
  circuit.min_req_cond_current = sf.if( circuit.OCPD_required, circuit.OCPD, circuit.min_req_OCPD_current );

  // Use Table 4, lookup: circuit.min_req_cond_current, find the next highest value, return the index column.
  circuit.conductor_current = sf.lookup( circuit.min_req_cond_current, tables[4], 0, true);
  // Use Table 4, lookup: circuit.conductor_current, return the first column.
  circuit.conductor_size_min = sf.lookup( circuit.conductor_current, tables[4] );
  if( circuit_name === 'exposed source circuit wiring' ){
    circuit.conductor_size_min = '10';
  }
  if( circuit_name === 'pv dc source circuits' ){
    circuit.conductor_size_min = '10';
  }
  if( circuit_name === 'inverter ac output circuit' ){
    if( circuit.OCPD === 15){
      circuit.conductor_size_min = '14';
    } else if( circuit.OCPD === 20){
      circuit.conductor_size_min = '12';
    } else if( circuit.OCPD === 25){
      circuit.conductor_size_min = '10';
    } else if( circuit.OCPD === 30){
      circuit.conductor_size_min = '10';
    }
  }
  // Use Table 5, lookup: circuit.conductor_size_min, return the first column.
  circuit.conductor_current = sf.lookup( circuit.conductor_size_min, tables[5], 1);
  // Use Table 6, lookup: circuit.conductor_size_min, return the first column.
  circuit.conductor_strands = sf.lookup( circuit.conductor_size_min, tables[6], 1 );
  // Use Table 6, lookup: circuit.conductor_size_min, return the second column.
  circuit.conductor_diameter = sf.lookup( circuit.conductor_size_min, tables[6], 2 );
  circuit.min_req_conduit_area_40 = circuit.total_conductors * ( 0.25 * PI() * math.pow(circuit.conductor_diameter, 2) );
```

The NEC article 352 and 358 tables are used to find a conduit with a sufficent 40% fill rate to hold the total conductor size for all the conductors.

```
  // Use Table 7, lookup: circuit.min_req_conduit_area_40, find the next highest value, return the first column.
  circuit.min_conduit_size_PVC_80 = sf.lookup( circuit.min_req_conduit_area_40, tables[7], 1, true );
  // Use Table 8, lookup: circuit.min_req_conduit_area_40, find the next highest value, return the first column.
  circuit.min_conduit_size_EMT = sf.lookup( circuit.min_req_conduit_area_40, tables[8], 1, true );
  circuit.min_conduit_size = circuit.min_conduit_size_EMT;
  if( circuit.conduit_type === '-' ){ circuit.min_conduit_size = '-'; }









  ///////
  // cleanup for display
  if( ! circuit.OCPD_required ){
    circuit.ocpd_type = '-';
    circuit.OCPD = '-';
  }
  circuit.conductor_size_min = circuit.conductor_size_min + ', ' + circuit.conductor_size_min;
  //////

});
```

Interconnection

At least one of the following checks must not fail:

- The sum of 125 percent of the inverter(s) output circuit current and the rating of the overcurrent device protecting the busbar exceeded the ampacity of the busbar.
- The sum of 125 percent of the inverter(s) output circuit current and the rating of the overcurrent device protecting the busbar exceeded 120 percent of the ampacity of the busbar.
- The sum of the ampere ratings of all overcurrent devices on panelboards exceeded the ampacity of the busbar.

```
interconnection.inverter_output_cur_sum = interconnection.inverter_output_cur_sum || inverter.max_ac_output_current;
interconnection.inverter_ocpd_dev_sum = interconnection.inverter_ocpd_dev_sum || inverter.OCPD;

interconnection.check_1 = ( ( interconnection.inverter_output_cur_sum * 1.25 ) + interconnection.supply_ocpd_rating ) >
interconnection.bussbar_rating;
interconnection.check_2 = ( interconnection.inverter_output_cur_sum * 1.25 ) + interconnection.supply_ocpd_rating >
interconnection.bussbar_rating * 1.2;
interconnection.check_3 = ( interconnection.inverter_ocpd_dev_sum + interconnection.load_breaker_total ) >
interconnection.bussbar_rating;

error_check.interconnection_bus_pass = sf.and( interconnection.check_1, interconnection.check_2, interconnection.check_3 );
// If error check is true, flag system design failure, and report notice to user.
if( error_check.interconnection_bus_pass ){ report_error( 'The busbar is not compliant.' );}
```

The panel's main OCPD must not exceed the bussbar rating.

```
error_check.interconnection_check_4 = interconnection.supply_ocpd_rating > interconnection.bussbar_rating;
// If error check is true, flag system design failure, and report notice to user.
if( error_check.interconnection_check_4 ){ report_error( 'The rating of the overcurrent device protecting the busbar exceeds the
rating of the busbar. ' );}
```