
csm Documentation

Release 0.1.0

George Scott and Katherine Dykes

June 19, 2013

CONTENTS

1	Introduction	1
2	Installation	2
3	Tutorial	3
4	Module Documentation	4
4.1	Cost and Scaling Model Utilities	4
4.2	Turbine Capital Costs (TCC) Module	4
4.3	Balance of Station Costs (BOS) Module	8
4.4	Operations and Maintenance (O&M) Module	9
4.5	Annual Energy Production (AEP)	10
4.6	csmPowerCurve Module	10
4.7	Drivetrain Efficiency Module	11
4.8	Finance	12
5	Theory	13

INTRODUCTION

The set of models contained in this software allow for the determination of wind plant cost of energy including models for turbine capital costs, balance of station costs, operations and maintenance costs, and annual energy production. It is a direct implementation of the model described in [Fingersh2006] with modifications as described in [Maples2010].

INSTALLATION

prerequisites

NumPy, SciPy

Download either `csm.py-0.1.0.tar.gz` or `csm.py-0.1.0.zip`, and uncompress/unpack it.

Install HubNacelleCST with the following command.

```
$ python setup.py install
```

To check if installation was successful run the unit tests for the NREL 5MW model

```
$ python test/test_csm.py
```

An 'OK' signifies that all the tests passed.

To access an HTML version of this documentation which contains further details and links to the source code, open `docs/index.html`.

TUTORIAL

To be done...

MODULE DOCUMENTATION

To be done...

4.1 Cost and Scaling Model Utilities

Cost escalators are used to adjust for input factor prices trends for materials and labor associated with turbine manufacturing, plant development and operation.

4.2 Turbine Capital Costs (TCC) Module

The NREL Cost and Scaling Model turbine module calculates turbine component costs and masses (either onshore or offshore). This excludes the foundation which is considered part of the balance of station even for offshore plants.

4.2.1 csmTurbine

This class calculates the the component masses and costs for a wind turbine using the NREL Cost and Scaling Model.

class `csm.src.csmTurbine.csmTurbine`

Methods

<code>__init__()</code>	Initialize properties for operations and maintenance costs.
<code>compute([hubHeight, machineRating, ...])</code>	Computes the turbine component masses and costs using the NREL Cost and Scaling Model method.
<code>getMass()</code>	Provides the overall turbine mass.
<code>getCost()</code>	Provides the turbine capital costs for a single turbine.

There are several sub-modules in the overall TCC module for each of the major sub-systems including the blades, hub, nacelle and tower.

4.2.2 csmBlades

This class calculates wind turbine blade mass and cost based on the NREL Cost and Scaling Model.

class `csm.src.csmBlades.csmBlades`

Methods

<code>__init__()</code>	Initialize the parameters for the wind turbine blade
<code>compute([diam, advanced, curr_yr, curr_mon, ...])</code>	Compute mass and cost for a single wind turbine blade by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(diam[, advanced])</code>	Compute mass for a single wind turbine blade using NREL cost and scaling model
<code>computeCost(diam[, advanced, curr_yr, curr_mon])</code>	Compute cost for a single wind turbine blade using NREL cost and scaling model
<code>getMass()</code>	Provides the mass for the wind turbine blade.
<code>getCost()</code>	Provides the cost for the wind turbine blade.

Example

The following is an example use of the individual `csmBlades` module.

Output should be:

Conventional blade design: Blades 276129.6 K\$ 25614.4 kg

Advanced blade design: Blades 251227.0 K\$ 17650.7 kg

4.2.3 csmHub

This class calculates wind turbine hub system mass and cost based on the NREL Cost and Scaling Model.

```
class csm.src.csmHub.csmHub
```

Methods

<code>__init__()</code>	Initialize the parameters for csm Hub
<code>compute(blademass[, diam, nblades, curr_yr, ...])</code>	Compute mass and cost for the hub system by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(blademass[, diam, nblades])</code>	Compute mass for the hub system based on NREL Cost and Scaling Model
<code>computeCost([diam, curr_yr, curr_mon, ...])</code>	Compute cost for the hub system by the NREL Cost and Scaling Model
<code>getMass()</code>	Provides the overall hub system mass.
<code>getCost()</code>	Provides the overall hub system cost.
<code>getHubComponentMasses()</code>	Provides the detail on hub system component masses.
<code>getHubComponentCosts()</code>	Provides the overall hub system cost.

Example

The following is an example use of the individual `csmHub` module.

The output should be:

Hub System Components:

Hub 124903.9 K\$ 22519.7 kg Pitch Mech 24379.7 K\$ 10313.9 kg Nose Cone 10509.0 K\$ 1810.5 kg

Hub total 379152.6 K\$ 34644.2 kg

4.2.4 csmNacelle

This class calculates wind turbine hub system mass and cost based on the NREL Cost and Scaling Model.

class `csm.src.csmNacelle.csmNacelle`

Methods

<code>__init__()</code>	Initialize the parameters for the nacelle
<code>compute(RotorDiam, MachineRating, RotorMass, ...)</code>	Compute mass and cost for a full wind turbine Nacelle by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(RotorDiam, MachineRating, ..., ...)</code>	Compute mass for a full wind turbine Nacelle using the NREL Cost and Scaling Model
<code>computeCost(RotorDiam, MachineRating, ..., ...)</code>	Compute cost for a full wind turbine Nacelle using the NREL Cost and Scaling Model
<code>getMass()</code>	Provides the mass for the overall wind turbine nacelle assembly.
<code>getCost()</code>	Provides the cost for the overall wind turbine nacelle assembly.
<code>getNacelleComponentMasses()</code>	Provides the mass for the overall wind turbine nacelle components.
<code>getNacelleComponentCosts()</code>	Provides the cost for the overall wind turbine nacelle components.

Example

The following is an example use of the individual `csmNacelle` module.

Output should be:

Nacelle Total 3249404.5229 K\$ 223315.6984 kg

Within the nacelle, there are additional sub-modules for the low-speed shaft, the gearbox and the generator.

4.2.5 LowSpdShaft

This class calculates wind turbine low speed shaft mass and cost based on the NREL Cost and Scaling Model.

class `csm.src.csmNacelle.LowSpdShaft`

Methods

<code>__init__()</code>	Initialize the parameters for low speed shaft
<code>compute(RotorDiam, RotorMass, RotorTorque, ...)</code>	Compute mass and cost for a low speed shaft by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(RotorDiam, RotorMass, RotorTorque)</code>	Compute mass for a low speed shaft using the NREL Cost and Scaling Model
<code>computeCost(RotorDiam, curr_yr, curr_mon)</code>	Compute cost for a low speed shaft using the NREL Cost and Scaling Model
<code>getMass()</code>	Provides the mass for the low speed shaft.
<code>getCost()</code>	Provides the cost for the wind turbine low speed shaft.

4.2.6 GearBox

This class calculates wind turbine gearbox mass and cost based on the NREL Cost and Scaling Model.

```
class csm.src.csmNacelle.GearBox
```

Methods

<code>__init__()</code>	Initialize the parameters for gearbox
<code>compute(iDsgn, MachineRating, RotorTorque, ...)</code>	Compute mass and cost for a gearbox by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(iDsgn, RotorTorque)</code>	Compute mass for a gearbox using the cost and scaling model
<code>computeCost(iDsgn, MachineRating, curr_yr, ...)</code>	Compute cost for a gearbox using the NREL Cost and Scaling Model
<code>getMass()</code>	Provides the mass for the gearbox.
<code>getCost()</code>	Provides the cost for the wind turbine gearbox.

4.2.7 Generator

This class calculates wind turbine generator mass and cost based on the NREL Cost and Scaling Model.

```
class csm.src.csmNacelle.Generator
```

Methods

<code>__init__()</code>	Initialize the parameters for the generator
<code>compute(iDsgn, MachineRating, RotorTorque, ...)</code>	Compute mass and cost for a generator by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(iDsgn, MachineRating, RotorTorque)</code>	Compute mass for a generator using the NREL Cost and Scaling Model
<code>computeCost(iDsgn, MachineRating, curr_yr, ...)</code>	Compute mass and cost for a generator using the NREL Cost and Scaling Model
<code>getMass()</code>	Provides the mass for the generator.
<code>getCost()</code>	Provides the cost for the wind turbine generator.

4.2.8 csmTower

This class calculates wind turbine tower mass and cost based on the NREL Cost and Scaling Model.

```
class csm.src.csmTower.csmTower
```

Methods

<code>__init__()</code>	Initialize the parameters for the wind turbine tower
<code>compute(RotorDiam, HubHeight[, curr_yr, ...])</code>	Compute mass and cost for a wind turbine tower by calling <code>computeMass</code> and <code>computeCost</code>
<code>computeMass(RotorDiam, HubHeight)</code>	Compute mass for a wind turbine tower using the NREL Cost and Scaling Model
<code>computeCost(curr_yr, curr_mon[, towerMass])</code>	Compute cost for a wind turbine tower by calling <code>computeMass</code> and <code>computeCost</code>
<code>getMass()</code>	Provides the mass for the wind turbine tower.
<code>getCost()</code>	Provides the cost for the wind turbine tower.

Example

The following is an example use of the individual `csmTower` module.

Output should be:

Tower Mass: 444384.157764 Tower Cost: 1009500.23594

Examples The following is an example use of the overall `csmTurbine` module for calculating the onshore and offshore wind turbine costs using the NREL 5 MW reference turbine [\[1\]](#):

Output should be:

Onshore configuration 5 MW turbine: Turbine total: 5487334.85 \$K 774049.19 kg

Offshore configuration 5 MW turbine: Turbine total: 6064385.11 \$K 774049.19 kg

References

4.3 Balance of Station Costs (BOS) Module

The NREL Cost and Scaling Model balance of station module calculates the installation, assembly and other non-turbine capital costs associated with the development of a wind plant (either onshore or offshore). The foundation is considered part of the balance of station even for offshore plants.

4.3.1 csmBOS

class `csm.src.csmBOS.csmBOS`

Methods

<code>__init__()</code>	Initialize properties for balance of plant costs including default inputs
<code>compute(seaDepth, machineRating, hubHt, ...)</code>	Computes the balance of station costs for a wind plant using the NREL Cost and Scaling Model method.
<code>getCost()</code>	Provides the overall balance of station costs for the plant.
<code>getDetailedCosts()</code>	Provides the detailed balance of station costs for the plant.

4.3.2 csmFoundation

class `csm.src.csmFoundation.csmFoundation`

Methods

<code>__init__()</code>	Initialize properties for balance of plant foundation cost.
<code>compute(MachineRating, HubHt, RotorDiam, ...)</code>	Computes the balance of station foundation costs for a wind plant using the NREL Cost and Scaling Model method.
<code>getCost()</code>	Provides the overall balance of station costs for the plant.

Examples The following is an example use of `csmBOS` for calculating the onshore and offshore wind plant Balance of Station Costs using the NREL 5 MW reference turbine [1]:

The output should appear as: BOS cost onshore: 3084560.970 BOS cost offshore: 7613488.697

The following is an example use of `csmBOS` for calculating the onshore and offshore wind plant Balance of Station Foundation Costs using the NREL 5 MW reference turbine [1]:

Onshore foundation cost: Foundation cost: 120779.05392 Offshore foundation cost: Foundation cost: 2125550.66079

References

4.4 Operations and Maintenance (O&M) Module

The NREL Cost and Scaling Model operations and maintenance module calculates the operations and maintenance, land lease costs, levelized replacement costs for operating a wind plant (either onshore or offshore).

4.4.1 csmOM

class `csm.src.csmOM.csmOM`

Methods

<code>__init__()</code>	Initialize properties for operations and maintenance costs.
<code>compute(aep, seaDepth, machineRating, year, ...)</code>	Computes the operations and maintenance costs for a wind plant using the NREL Cost and Scaling Model method.
<code>getOMCost()</code>	Provides the operations and maintenance costs for the plant.
<code>getLLC()</code>	Provides the land lease costs costs for the plant.
<code>getLRC()</code>	Provides the levelized replacement costs costs for the plant.

Examples The following is an example use of `csmOM` for calculating the onshore and offshore wind plant Operations and Maintenance Costs:

Output should be:

OM costs onshore 144053.487 LevRep 58699.223 Lease 22225.395

OM costs offshore 401819.023 LevRep 91048.387 Lease 22225.395

4.5 Annual Energy Production (AEP)

The annual energy production model uses methods as described in the references for determining a wind plant annual energy production from a small number of site and turbine design input parameters. Within the AEP module are sub-modules for determining the wind turbine power curve and drivetrain efficiency.

4.5.1 csmAEP

The NREL Cost and Scaling Model AEP calculation based on a Weibull distribution of wind speed and input assumptions regarding plant losses.

class `csm.src.csmAEP.csmAEP`

Methods

<code>__init__()</code>	
<code>compute(powercurve[, ratedPower, hubHt, ...])</code>	Computes annual energy production from a wind plant
<code>getAEP()</code>	Returns annual energy production for the project.
<code>getCapacityFactor()</code>	Returns project capacity factor.

Example The following is an example use of `csmAEP` to calculate the annual energy production for a fictitious plant using the NREL 5 MW Reference Turbine [\[1\]](#):

The output should appear as: AEP: 18744306.322 MWh (for Rated Power: 5000.0) CapFactor: 42.795 %

References

4.6 csmPowerCurve Module

This module determines a wind turbine power curve which accounts for drivetrain efficiency. It is based on the NREL Cost and Scaling Model and requires a small number of input parameters as described below.

4.6.1 csmPowerCurve

The NREL Cost and Scaling Model Power Curve module.

class `csm.src.csmPowerCurve.csmPowerCurve`

Methods

<code>__init__()</code>	Initialize properties for power curve
<code>compute(drivetrain[, hubHt, ratedPower, ...])</code>	Computes the ideal power curve for a turbine after drivetrain losses using the NREL Cost and Scaling Model method.
<code>getRatedWindSpeed()</code>	Provides the minimum wind speed required for a turbine to operate at rated power after drivetrain losses are deducted.
<code>getRatedRotorSpeed()</code>	Provides the rotor speed for a turbine operating at rated power after drivetrain losses are deducted.
<code>getMaxEfficiency()</code>	Provides the maximum drivetrain efficiency for a power curve.
<code>getPowerCurve()</code>	Gives an array of power output as a function of input wind speed for a wind turbine accounting for drivetrain losses.
<code>idealPowerCurve(Wind, ITP, kTorque, ...)</code>	Determine the ITP (idealized turbine power) array

Example The following is an example use of `csMPowerCurve` to the a power curve based on the NREL 5 MW Reference Turbine [1]:

The output should appear as: Rated Speed: 11.750 mps Rated RPM: 12.126 rpm

Along with the below plot.

References

4.7 Drivetrain Efficiency Module

This module describes an interface for drivetrain efficiency analysis as well as a model based on the NREL Cost and Scaling Model which implements the interface. The methods are necessary to determine the actual output power of a wind turbine to the interconnection point after drivetrain losses for the drivetrain (gearbox, generator, etc) are taken into account. The simplified model is based on the analysis done in reference [1].

4.7.1 DrivetrainEfficiencyModel

class `csM.src.csMDriveEfficiency.DrivetrainEfficiencyModel`

Methods

<code>getMaxEfficiency()</code>	Return maximum efficiency of the generator
<code>getDrivetrainEfficiency(outputPower, ...)</code>	Returns efficiency of drivetrain operating at a ratio of outputPower to ratedHubPower

4.7.2 csMDrivetrainModel

class `csM.src.csMDriveEfficiency.csMDriveEfficiency`

Methods

<code>__init__(drivetrainDesign)</code>	Set efficiency calculation coefficients based on drivetrain configuration type.
<code>getMaxEfficiency()</code>	Uses quadratic formula to calculate maximum drivetrain efficiency with coefficients based on drivetrain configuration.
<code>getDrivetrainEfficiency(outputPower, ...)</code>	Uses quadratic formula to calculate drivetrain efficiency with coefficients based on drivetrain configuration.

Example The following is an example use of `csmDriveEfficiency` to determine the maximum drivetrain efficiency for a wind turbine with a 3-staged gearbox and induction generator drivetrain configuration:

The output should appear as: Max Efficiency: 0.902 Output power: 3422.000 Rated power: 5543.000 Efficiency: 0.894

References

4.8 Finance

This module uses the NREL Cost and Scaling Model to determine a wind plant project's cost of energy and levelized cost of energy. It takes as input all the aggregate costs and financial parameters for the project.

4.8.1 csmFinance

class `csm.src.csmFinance.csmFinance`

Methods

<code>__init__()</code>	Initialize properties for <code>csmFinance</code>
<code>compute(ratedpwr, tcc, om, llc, lrc, bos, ...)</code>	Computes a wind plant cost of energy and levelized cost of energy using the NREL Cost and Scaling Model method.
<code>getCOE()</code>	Returns the wind plant cost of energy
<code>getLCOE()</code>	Returns the wind plant levelized cost of energy

Example The following is an example use of `csmDriveEfficiency` to determine the maximum drivetrain efficiency for a wind turbine with a 3-staged gearbox and induction generator drivetrain configuration:

The output should appear as: Onshore: LCOE: 0.052776 COE: 0.065765

Offshore: LCOE: 0.102919 COE: 0.119694

THEORY

The theory for the models in this software are based directly on the work described in references [Fingersh2006], [WindPACT], [Sunderland1993], [Malcolm2002], and [Maples2010]. This section provides an overview of the development of the physical and cost models for the major hub and drivetrain components.

The NREL Cost and Scaling Model [Fingersh2006] provides a simple cost and sizing tool to estimate wind turbine component masses and costs based on a small number of input parameters such as rotor diameter, hub height and rated power. The model was developed over several results following the Wind Partnerships for Advanced Component Technology (WindPACT) work that occurred between roughly 2002 to 2005 [WindPACT]. The original form of the cost model was based on an earlier model from 1993 out of the University of Sunderland (the Sunderland Model) [Sunderland1993]. The Sunderland Model created a set of wind turbine models to estimate the mass and cost of all major wind turbine components including: blade, hub system [hub, pitch system, and nose cone], nacelle [low speed shaft, main bearings, gearbox, high speed shaft/mechanical brake, generator, variable speed electronics, electrical cabling, mainframe [bedplate, platforms and railings, base hardware, and crane], HVAC system, controls, and nacelle cover], and tower. The Sunderland model was based on a set of semi-empirical models for each component which estimated design loads at the rotor, propagated these loads through the entire system, and used the loads to estimate the size of each component calibrated to data on actual turbines in the field during that time. Cost estimates were then made on a per weight basis using a multiplier again based on field data or industry sources.

To arrive at the NREL Cost and Scaling Model, the WindPACT studies began in many cases with the Sunderland model and updated the results with new coefficients or, in some cases, with entirely new cost equations based on curve fits of key design parameters (rotor diameter, etc) to the results of detailed design studies [Malcolm2002]. In addition, the WindPACT work established estimates of costs associated with balance of station and operations and maintenance for a fictitious wind plant in North Dakota which led to an overall cost of energy model for a wind plant. The key cost of energy equation for a wind plant is given in the NREL Cost and Scaling Model [Fingersh2006] as:

$$COE = (FCR * (BOS + TCC)) / AEP + (LLC + LRC + (1 - tr) * OM) / AEP$$

where COE in this equation is a simple estimate of a wind plant cost of energy, FCR is the fixed charge rate for the project, BOS are the total balance of station costs for the project, TCC are the total turbine capital costs for the project, AEP is the annual energy production for the project, LLC are the annual land-lease costs, LRC is the levelized replacement cost for major part replacement, tr is the tax rate, and OM are the annual operations and maintenance costs which are tax deductible.

While the NREL Cost and Scaling Model improved the overall cost estimation for larger turbines on the order of 1 MW+, it abstracted away from the engineering analysis foundations of the original Sunderland model. This is depicted in the below figure where it can be seen that the engineering-analysis has been replaced by a series of curve fits which relate a small number of design parameters to mass and cost estimates for major wind turbine components as well as over all plant energy production and costs.

The resulting NREL Cost and Scaling Model allows for a variety of interesting analyses including scaling of conventional technology from under a MW to 5 MW+, assessing impact of trends in input factors for materials and labor on

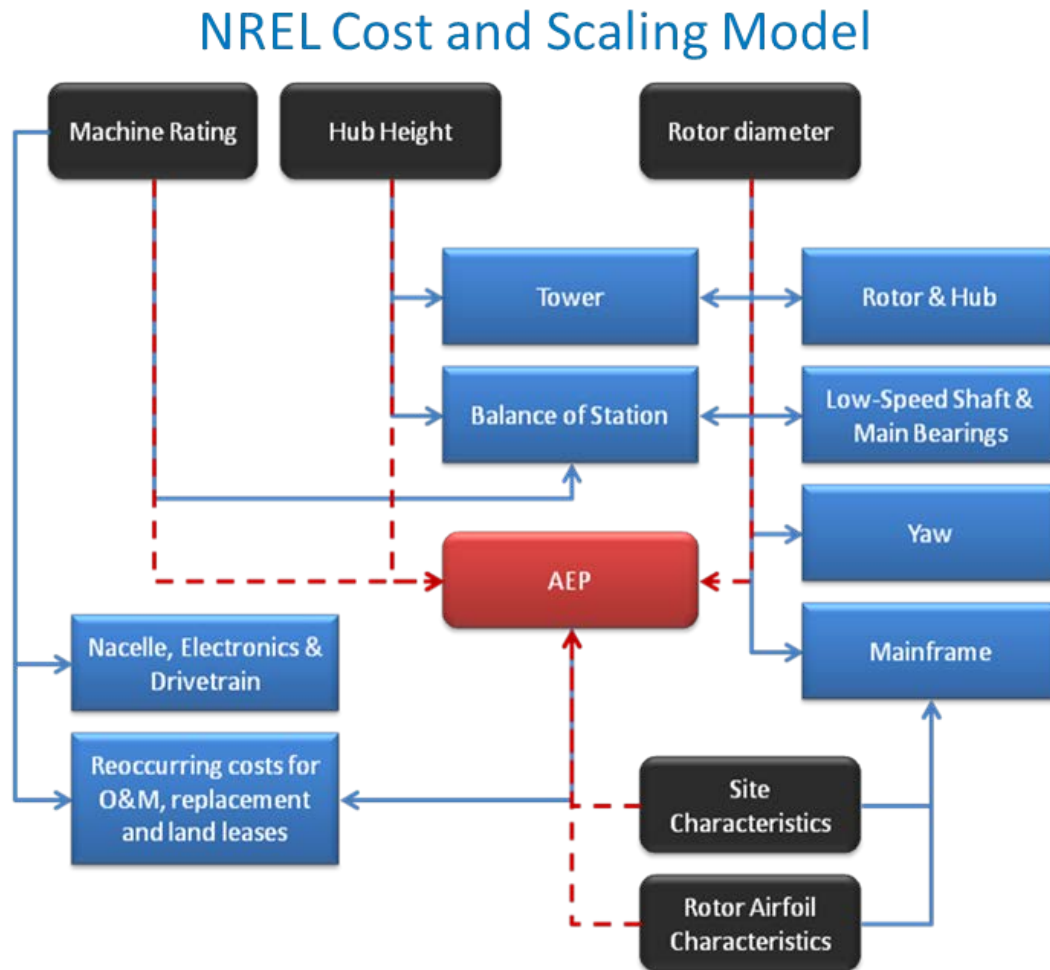


Figure 5.1: NREL Cost and Scaling Model Key Input-Output Relationships.

wind plant cost of energy, etc. However, it does not preserve the underlying engineering relationships of the original Sunderland model and thus loses some fidelity of assessing how design changes may impact system costs.

This model directly implements the full wind plant NREL Cost and Scaling Model as described in [Fingersh2006] with modifications for the drivetrain as described in [Maples2010].