syntheticbaseloadvaria

December 21, 2017

1 Synthetic Base Load Varia

File: syntheticbaseloadvaria.ipynb. Created 2017-12-21

In [1]:
```
import numpy as np
```

In [2]:
```
# From engineering-economics-utilities.ipynb

def AgivenP(K, r, n):
    '''
    Given a present outlay of capital (debt) K, an interest rate
    or hurdle rate if r, and a period of n years to repay the
    debt, returns the annual payments. 5% interest means r = 0.05.
    '''
    numer = r * (1 + r) ** n
    denom = (1 + r) ** n - 1
    return K * numer / denom
```

1.1 Now, working with the Randolph etc. spreadsheet

LCOE for Fig 9.22 e4s2.xls

This has some very useful information, I think the LCOE calculations are problematic.

In [11]:
```
# Begin with the LCOE material, rows 2-7

# Note that Randolph follows the sensible path of figuring an
# annual cost of operation and calls this LCOE. I guess that's ok
# but I do think it better just to say this is the annual cost...

# Pulverized coal-steam:

capitalCost = 2300 # This is in $/kW
# Leaving this out. Will figure LCOE for a kW
#plantSize = 100 # 100 megawatts
#plantCost = capitalCost*plantSize*1000 # 2300 per kW times 1000 to get cost per mega
r = 0.07 # interest rate of 7% by default
r = 0.16763 # assumed by Randolph et al.
```

```
n = 25
annualCapitalRecoveryCost = AgivenP(capitalCost,r,n)
print(annualCapitalRecoveryCost)
# This value, 393.72540351603493, is slightly higher than Randolph,
# which is capitalCost*r = 385.549. Notice that Randolph
# does not provide a value for n. The results will be sensitive
# to both r and n, but we can visit that later.
```

```
393.72540351603493
```

```
In [37]: # OK, let's do this (the annual fixed costs) for each of the four cases.
         capCost={'Coal':2300,'GasExp':950,'GasChp':950,'Nuclear':4500}

         annualFixed = {}
         r = 0.16
         n = 25
         for technology in capCost.keys():
             capitalCost = capCost[technology]
             annualFixedCost = AgivenP(capitalCost,r,n)
             print('{} has annual fixed cost: {}'.format(technology,annualFixedCost))
             annualFixed[technology] = annualFixedCost
         print(annualFixed)
```

```
Coal has annual fixed cost: 377.22901515427895
GasExp has annual fixed cost: 155.81198452024566
GasChp has annual fixed cost: 155.81198452024566
Nuclear has annual fixed cost: 738.056768780111
{'Coal': 377.22901515427895, 'GasExp': 155.81198452024566, 'GasChp': 155.81198452024566, 'Nucl
```

1.2 Now let's do the variable cost.

Here the spreadsheet is very useful, especially columns E, F, and G. Again, I'm uneasy with the spreadsheet calculations, so I will do it my way with comments.

```
In [16]: heatRate = 8750 # this is for coal in Btu/kWh. We are assuming 1 kW per hour.
         capacityFactor = 0.85 # Given in the spreadsheet. Is this just a place holder or does
         # have additional meaning, e.g., coal has to be down 15% of the year?
         fuelPrice = 2.5 # This is in dollars per million Btu
         annualFuelCost = 8760*heatRate*capacityFactor*(fuelPrice/1000000) # Dividing by 1,000
         # Note that Randolph multiplies this all by 1.5. Why?
         # storage costs?
         print(annualFuelCost)
```

```
162.88125000000002
```

```
In [18]: variableOandMCost = 0.004 # in dollars per kWh
         annualOandMCost = variableOandMCost * capacityFactor * 8760
         print(annualOandMCost)
```

```
29.784
```

```
In [19]: print(annualFuelCost + annualOandMCost)
```

```
192.66525000000001
```

```
In [20]: # This is what Randolph gets, but what is the 1.5 about?
         print(annualFuelCost* 1.5 + annualOandMCost)
```

```
274.105875
```

1.3 Well, now let's do the variable component for all four cases.

```
In [38]: heatRateDict = {'Coal':8750,'GasExp':6600,'GasChp':6600,'Nuclear':10500}
         fuelPriceDict  = {'Coal':2.5,'GasExp':6,'GasChp':3,'Nuclear':0.6}
         capacityFactor = {'Coal':0.85,'GasExp':0.9,'GasChp':0.85,'Nuclear':0.85}
         variableOandMCost = 0.004 # in dollars per kWh
         annualVariableCost = {}
         for technology in capCost.keys():
             annualOandMCost = variableOandMCost * capacityFactor[technology] * 8760
             heatRate = heatRateDict[technology]
             fuelPrice = fuelPriceDict[technology]
             annualFuelCost = 8760*heatRate*capacityFactor[technology]*(fuelPrice/1000000)
             totalAnnualVariableCost = annualOandMCost + annualFuelCost
             print('{} has total annual variable cost of {}'.format(technology,totalAnnualVaria
             annualVariableCost[technology] = totalAnnualVariableCost
```

```
Coal has total annual variable cost of 192.66525000000001
GasExp has total annual variable cost of 343.74240000000003
GasChp has total annual variable cost of 177.2148
Nuclear has total annual variable cost of 76.6938
```

```
In [39]: totalAnnualCost = {}
         for technology in capCost.keys():
             total = annualVariableCost[technology] + annualFixed[technology]
             totalAnnualCost[technology] = total
             print('{} has total annual cost of {}'.format(technology,total))
```

```
Coal has total annual cost of 569.894265154279
GasExp has total annual cost of 499.55438452024566
GasChp has total annual cost of 333.02678452024566
Nuclear has total annual cost of 814.750568780111
```

```
In [41]: # Now add the 1.5 factor
         totalAnnualCost = {}
         for technology in capCost.keys():
             annualOandMCost = variableOandMCost * capacityFactor[technology] * 8760
             heatRate = heatRateDict[technology]
             fuelPrice = fuelPriceDict[technology]
             annualFuelCost = 8760*heatRate*capacityFactor[technology]*(fuelPrice/1000000)* 1.5
             totalAnnualVariableCost = annualOandMCost + annualFuelCost
             #print('{} has total annual variable cost of {}'.format(technology,totalAnnualVar
             annualVariableCost[technology] = totalAnnualVariableCost

             total = annualVariableCost[technology] + annualFixed[technology]
             totalAnnualCost[technology] = total
             print('{} has total annual cost of {}'.format(technology,total))
```

```
Coal has total annual cost of 651.334890154279
GasExp has total annual cost of 655.6575845202457
GasChp has total annual cost of 406.7421845202457
Nuclear has total annual cost of 838.205468780111
```

1.4 And these last numbers are close to Randolph

They differ mainly in the interest rate assumed and using the AgivenP function.

```
In [44]: # so I will use the last version now to calculate LCOE i $/kWh
         for tech in totalAnnualCost.keys():
             lcoe = totalAnnualCost[tech]/(capacityFactor[tech]*8760)
             print('{} has LCOE in $/kWh of {}'.format(tech,lcoe))
```

```
Coal has LCOE in $/kWh of 0.08747446819154969
GasExp has LCOE in $/kWh of 0.08316306247085815
GasChp has LCOE in $/kWh of 0.054625595557379225
Nuclear has LCOE in $/kWh of 0.11257124211390156
```

```
In [32]: !ls
```

```
syntheticbaseloadvaria.ipynb syntheticbaseloadvaria.pdf
```

```
In [33]: !jupyter nbconvert --to pdf syntheticbaseloadvaria.ipynb
```

```
[NbConvertApp] Converting notebook syntheticbaseloadvaria.ipynb to pdf
[NbConvertApp] Writing 30860 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
```

```
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 36803 bytes to syntheticbaseloadvaria.pdf
```

In [34]:
```
!ls
```

```
syntheticbaseloadvaria.ipynb syntheticbaseloadvaria.pdf
```

In [ ]: