



**HAL**  
open science

# Economic consistency of salvage value definitions

Pierre Haessig

► **To cite this version:**

| Pierre Haessig. Economic consistency of salvage value definitions. 2024. hal-04097092v2

**HAL Id: hal-04097092**

**<https://hal.science/hal-04097092v2>**

Preprint submitted on 29 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Economic consistency of salvage value definitions

Pierre Haessig, April 2023, updated July 2024 with references from the field of *Engineering Economics*.

## Abstract

This short report analyzes the definition of the *salvage value* of a component within a wider project investment analysis. The analysis shows that the classical() *definition is \*close but not exactly consistent* from an economic point of view. For an exact economic consistency of the salvage value, we derive an alternate definition using a more complex formula, which we call the *economically consistent salvage value*. This formula is equivalent to the classical definition when the discount factor is zero. The key takeaway of our analysis is that **using the classical definition creates an overestimation of the annualized project cost of up to 30%** in the worst cases (see section 3), even for a small discount factor!

(\*) classical definition in the field on energy system optimization at least

One year after the 2023 version of this report, we found references from the *Engineering Economics* literature showing that the formula we derived was in fact already known under the term “implied” (Thuesen 1984) or “imputed” (Jones 1990) salvage value. Notice that we didn't rewrite the entire document to reflect the preexistence of these references, but comparisons are done in Appendix 6.2 to verify the equivalence between ours and literature definitions.

Still, all the analyses (graphics, series development and sensitivity to interest rate and lifetime), presented in this document are, to the best of our knowledge, original. Another contribution is the generalization in the case of component replacements (section 4.1).

## Context and definitions

When computing the present value of a project like a microgrid, there is generally a mismatch between the economic lifetime (or horizon of analysis) of the project and the lifetime of each component used in the project. In order to smooth out this mismatch, a salvage value is subtracted from the project cost for each component having some remaining lifetime at the end of the project.

A key parameter in this analysis is the real discount factor  $i$  which can be 0 (no discount), positive (the interest rate for borrowing capital is higher than inflation) or negative (inflation is higher than the interest rate). The discount factor is assumed constant over the years in this analysis.

For simplicity, we start by analyzing a project with one single component. The investment cost of the component is  $C_c$ . The component lifetime is  $T_c$ . We also assume that the project

horizon is shorter than the component lifetime ( $T_p \leq T_c$ ) so that there is no replacement cost. We consider zero operation and maintenance (O&M) costs in this analysis since they are just extras which are independent to this discussion. Therefore, from the component point of view, considering its own lifetime has the economic analysis horizon, the Net Present Cost of the component is just the investment:

$$NPC_c = C_c$$

Then, the annualized cost of the component over its lifetime  $T_c$  is:

$$C_{ann,c} = NPC_c \times CRF(i, T_c)$$

where  $CRF$  is the Capital Recovery Factor (HOMER Software, 2023) which transforms a Net Present Value of Costs into corresponding annuities:

$$CRF(i, T) = \frac{i(1+i)^T}{(1+i)^T - 1}$$

Now we turn to the economic analysis of the project which includes this component. The Net Present Cost of the project over its lifetime  $T_p$  is:

$$NPC_p = C_c - \frac{1}{(1+i)^{T_p}} S$$

where  $S$  is the *nominal* salvage value of the component, which gets discounted since the salvage happens at the end of year  $T_p$ . The corresponding annualized cost is:

$$C_{ann,p} = NPC_p \times CRF(i, T_p)$$

(notice that annuities are computed over the project lifetime, i.e.  $T_p$ )

The key subject of this report is: *what should be value of the salvage value?*

## Properties of different salvage values definitions

### Possible definitions of the salvage value

There are two possible definitions for the (nominal) salvage value:

a. the *classical*(\*) definition which is proportional to the relative remaining lifetime of the component at the end of the project:

$$S_a = \frac{T_c - T_p}{T_c} C_c$$

b. the *economically consistent* definition which we propose

$$S_b = \frac{(1+i)^{T_c} - (1+i)^{T_p}}{(1+i)^{T_c} - 1} C_c$$

(\*) definition e.g. used in the HOMER Pro microgrid design software, see (Lambert 2005) or (HOMER 3.16 documentation)

Remark: the two definitions are quite close, because

$$S_b \sim S_a + O(i)$$

so the economically consistent definition falls back to classical one for small discount rate  $i$ , with a difference which is linear in  $i$  (see illustration below and appendix).

Now we need to specify what we call "economically consistent" ...

## Economic consistency of salvage value

We say that a definition of the salvage value is economically consistent if it yields an **annualized project cost which is equal to the annualized cost of the component**. That is, there is no financial difference between the analysis of the component alone versus the same component embedded into the wider scope of the project.

This means that the *economically consistent* definition is the solution of the following equation:

$$\underbrace{C_{ann,p}}_{\text{depends on } S} = C_{ann,c}$$

Proof: see Appendix.

Notice that this formula was already known in the *Engineering Economics* literature under the term "implied" (Thuesen 1984) or "imputed" (Jones 1990) salvage value. It was similarly used in the context of comparing alternative investments with different lifetimes. The "implied" comes from the fact this salvage value emerges when equating annualized cost computations with different lifetimes. See Appendix for a proof of equivalence.

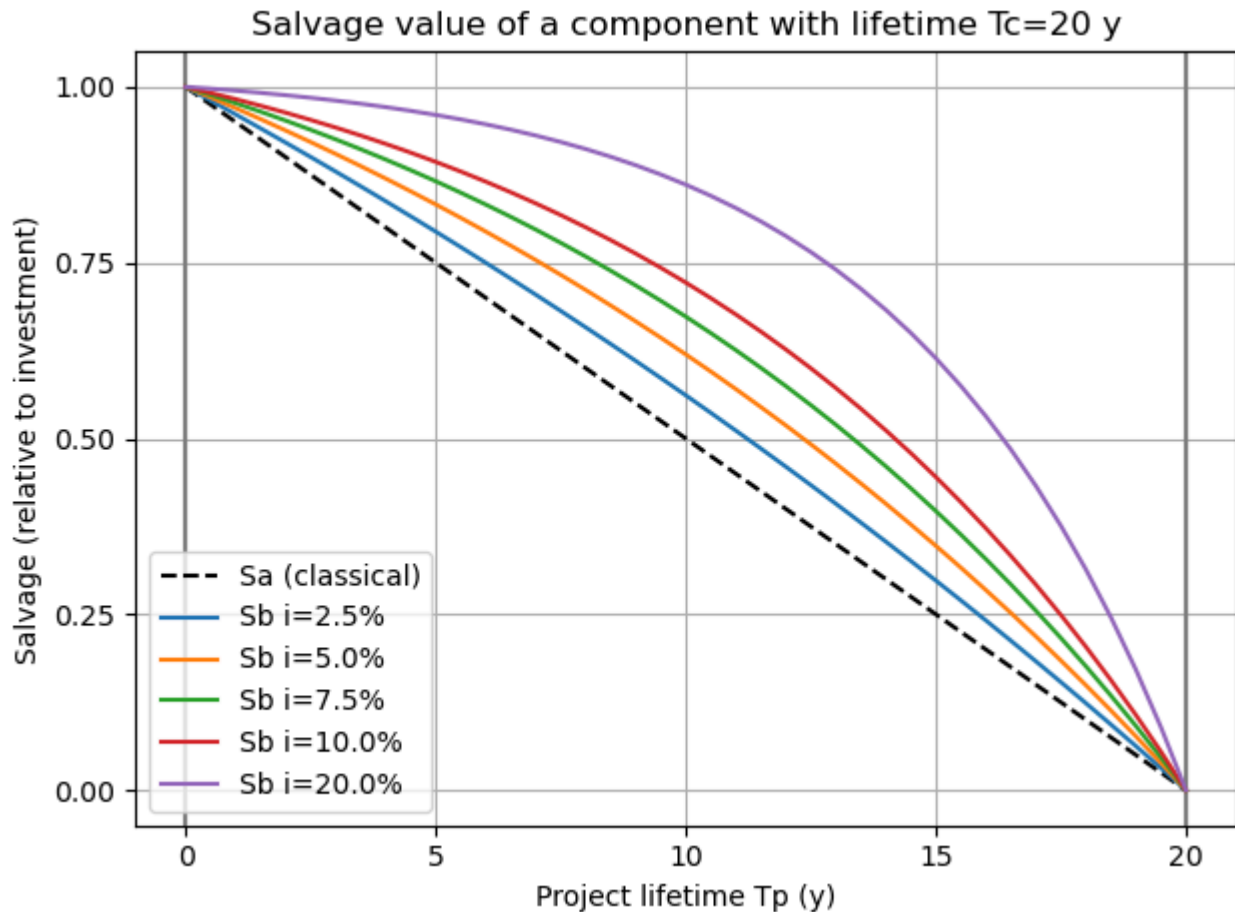
## Illustration of the two definitions of the salvage value

and see appendix for the proof of the analysis of the difference between the two definitions for small discount rates.

Key takeaway: for a positive discount rate, the **consistent salvage value is greater than the classical (linear) salvage value**, except in the trivial cases when the component is new ( $T_p \ll T_c$ ) or to be scrapped ( $T_p = T_c$ ).

*Remark: the following code cells which produce the graphs can only be run once the cells in the appendix are run.*

```
In [71]: i_list = [0.025, 0.050, 0.075, 0.10, 0.20] # list of discount rates
plot_salvage_Tp(i_list, Tc=20);
```



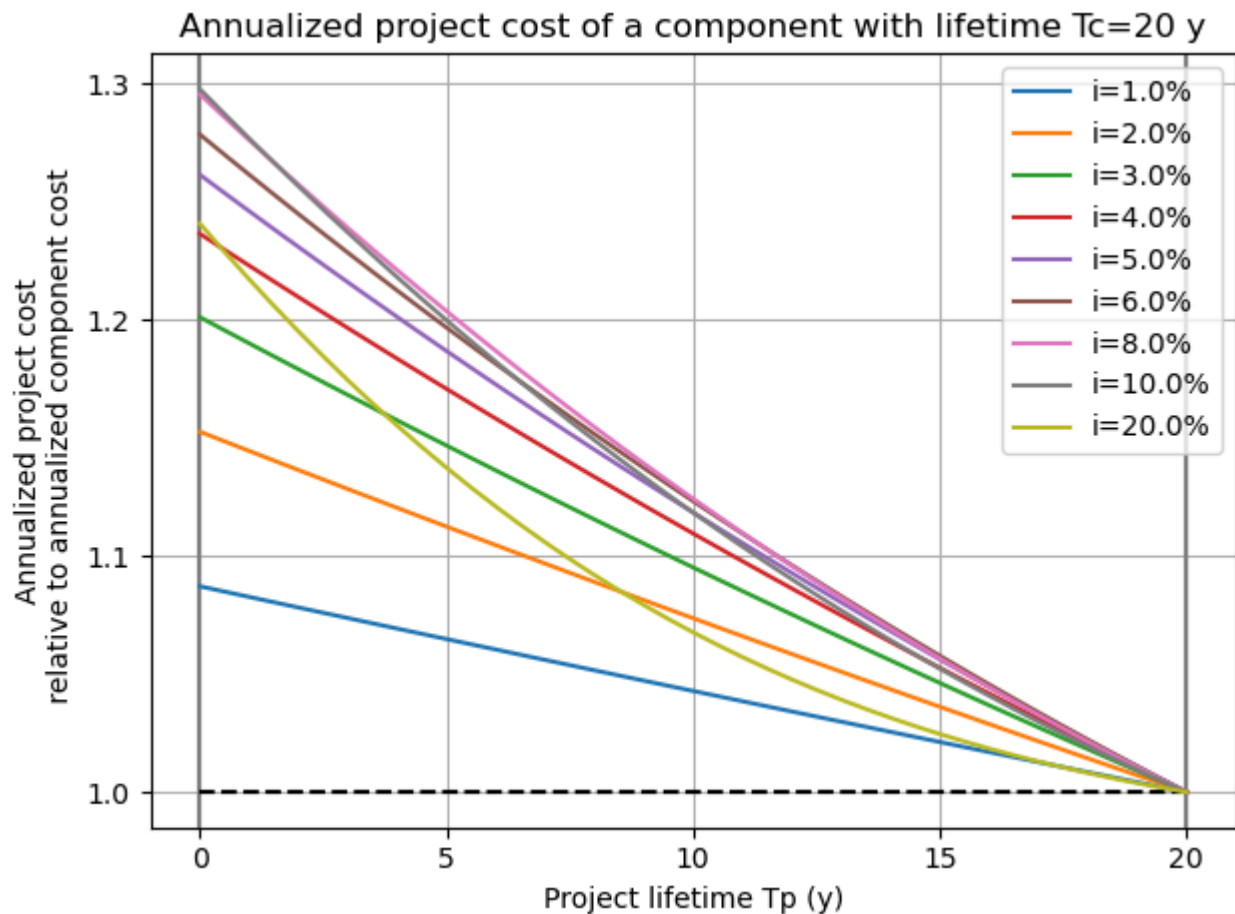
## Illustration of the economic inconsistency of the classical definition

Choosing the classical salvage value when computing the annualized project cost creates a positive bias (cost is overestimated compared to the annualized cost of the component alone).

The shape of this effect is more difficult to analyze compared to the shape of the salvage value. From the following plot we observe that:

- the overestimation of annualized project cost is worst (strongest) for short projects ( $T_p/T_c \rightarrow 0$ )
  - it can reach **+20% to +30%** for the case below
- this overestimation decrease almost linearly with project horizon (and is of course zero for  $T_p = T_c$  since salvage is zero)
- the effect of the discount rate is *complex*:
  - no overestimation when discount is zero (since  $S_a = S_b$  in that case)
  - the overestimation grows quickly with discount rate, as long as it is "small enough"
  - beyond a certain threshold of discount rate, the overestimation is near constant or even decreasing

```
In [72]: #i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many
plot_Cann_Tp(i_list, Tc=20);
```

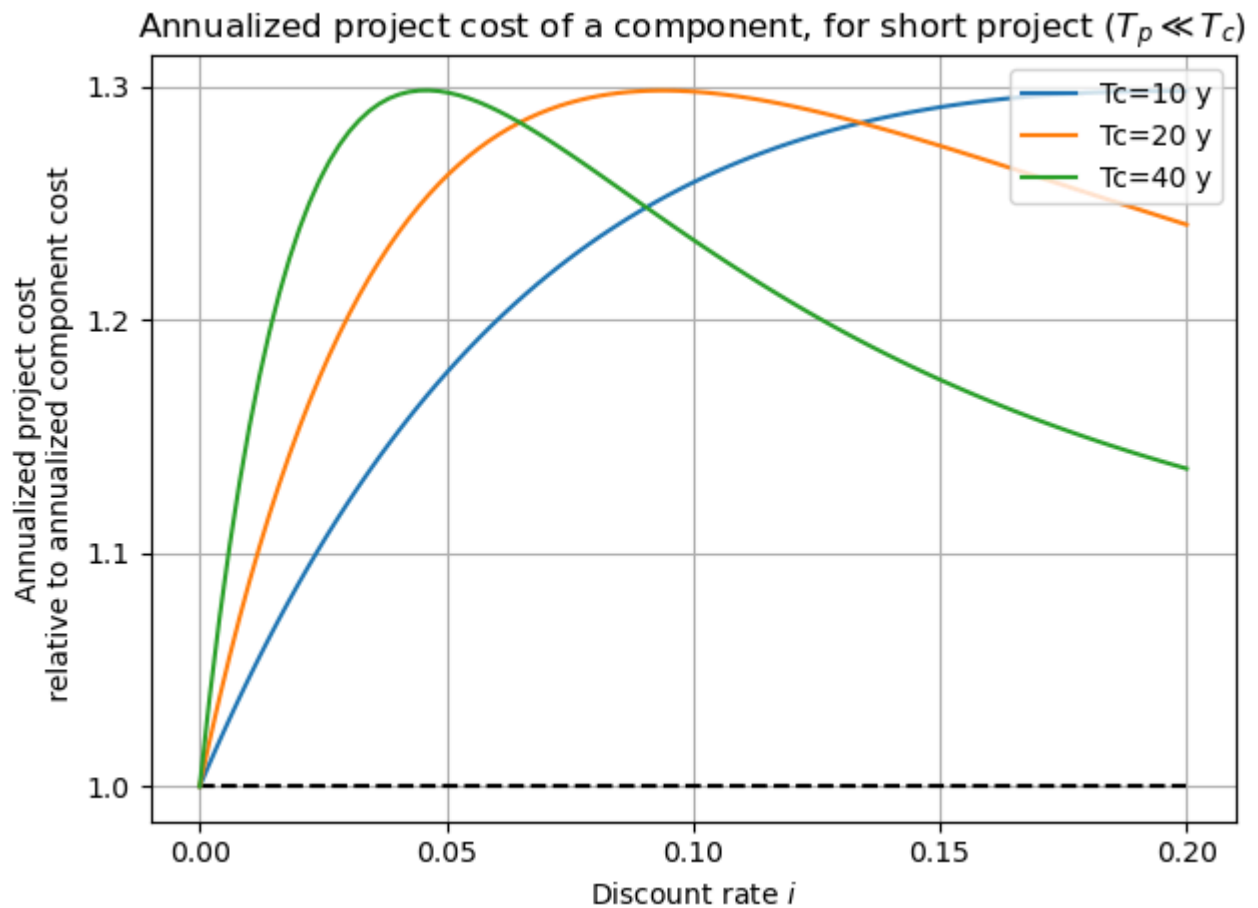


Now looking only at the start of the curves above, that is for short projects ( $T_p/T_c \rightarrow 0$ ), we plot the effect of the discount rate, for different component lifetime. It appears that:

- the worst case overestimation is consistently about 1.30, i.e. +30%
- this worst case overestimation is attained for a discount rate slightly below  $2/T_c$

See the appendix for a mixed analytical and numerical proof that the worst case is attained for  $i \approx 1.79/T_c$  (when  $T_c \gg 1.79$ ). The worst case overestimation is indeed independant of  $T_c$  and  $i$  and is numerically evaluated at 1.2984... i.e. +29.84%.

```
In [73]: Tc_list=[10, 20, 40]
plot_Cann_Tp0_i(Tc_list, i_max=0.20);
```



## Further use of the economically consistent salvage value

In the above discussion, we have only studied the case a of project with an horizon  $T_p$  shorter than the component lifetime  $T_c$ . However, the economic consistency of the salvage value definition  $S_b$  also applies to more complex cases. In particular, we analyze two useful cases:

1. The project horizon is longer than the component lifetime ( $T_p > T_c$ ) so that there is a replacement cost that needs to be accounted for
2. The project is (virtually) sold at mid-project term and bought back immediately after.

In both cases, we show (see appendix for mathematical proves) that the annualized project cost is unchanged and equal to the annualized component cost

## Project with replacements of the component

### with one replacement

With one replacement (occurring at the end of year  $T_c$ ), the Net Present Cost of the project over its lifetime  $T_p$  is:

$$NPC_p = C_c + \frac{1}{(1+i)^{T_c}} R_c - \frac{1}{(1+i)^{T_p}} S$$

To prove the economic consistency (annualized project cost = annualized component cost) we must of course assume that  $R_c$ , the nominal replacement cost of the component, is equal to the initial investment cost:  $R_c = C_c$ .

For the salvage value  $S$ , we use the above definitions except that the remaining life should be counted for the second component, which has only been used for time  $T_p - T_c$  and which end of life should occur at  $2.T_c$ . This means that in the salvage value definitions become:

The classical salvage value definition becomes:

$$S_a = \frac{2.T_c - T_p}{T_c} C_c$$

For the economically consistent definition, there is an ambiguity about which term of the numerator to update:

- should  $T_c$  become  $2.T_c$ , that is the component end of life happen twice later ?
- should  $T_p$  become  $T_p - T_c$ , that is update the usage duration of the component?

As proved in the appendix, correct update of the definition is to update the usage duration ( $T_p$  becomes  $T_p - T_c$ ):

$$S_b = \frac{(1+i)^{T_c} - (1+i)^{T_p-T_c}}{(1+i)^{T_c} - 1} C_c$$

## With more than one replacement

If the component is replaced  $n$  times, the salvage value should be updated by replacing  $T_p$  by  $T_p - n.T_c$ . In the appendix, we have the proof for two replacements ( $n = 2$ ) and we guess that it could be proved for the general case  $n \geq 1$ .

## Project is sold at mid-term and bought back immediately after

This case is useful when optimizing *reinvestment* within the project lifetime. Indeed in some context, it can be easier to model the complete sale of the project which is immediately bought after (possibly with different capacities) than to model the cost of the capacity change.

To prove the economic consistency (annualized project cost = annualized component cost) we must of course assume here that the project is bought back *unchanged* (*no capacity expansion*) even if the interest of this formulation is to allow capacity adjustment. For simplicity, we do not consider replacement (i.e.  $T_p \leq T_c$ ).

We introduce  $T_{mid} \leq T_p$  the year of selling the project at mid project. There are **two salvage values**, one for the first sale at mid project ( $S_{mid}$ ) and then the salvage at the real end of the project ( $S_{final}$ ) but considering only duration of that second sub-project for the aging of the component. With these notations, the Net Present Cost of a project, with mid-term sale and immediate buy back, over its lifetime  $T_p$ , is:

$$NPC_p = C_c - \frac{1}{(1+i)^{T_{mid}}} S_{mid} + \frac{1}{(1+i)^{T_{mid}}} C_c - \frac{1}{(1+i)^{T_p}} S_{final}$$



The two economically consistent salvage values are the same as the case of a project without replacement, but with reduced project durations:

- the mid project salvage uses  $T_{mid}$  as project duration
- the final salvage value uses  $T_p - T_{mid}$  as project duration

$$S_{b,mid} = \frac{(1+i)^{T_c} - (1+i)^{T_{mid}}}{(1+i)^{T_c} - 1} C_c$$

$$S_{b,final} = \frac{(1+i)^{T_c} - (1+i)^{T_p - T_{mid}}}{(1+i)^{T_c} - 1} C_c$$

Proof: see Appendix

### Economic inconsistency of the classical definition with mid-term sale

To study the practical importance of using the above formulae for salvage, we compute the annualized project cost when using the classical salvage values. Since there are two salvage events, this means using the following salvage values:

$$S_{a,mid} = \frac{T_c - T_{mid}}{T_c} C_c$$

$$S_{a,final} = \frac{T_c - (T_p - T_{mid})}{T_c} C_c$$

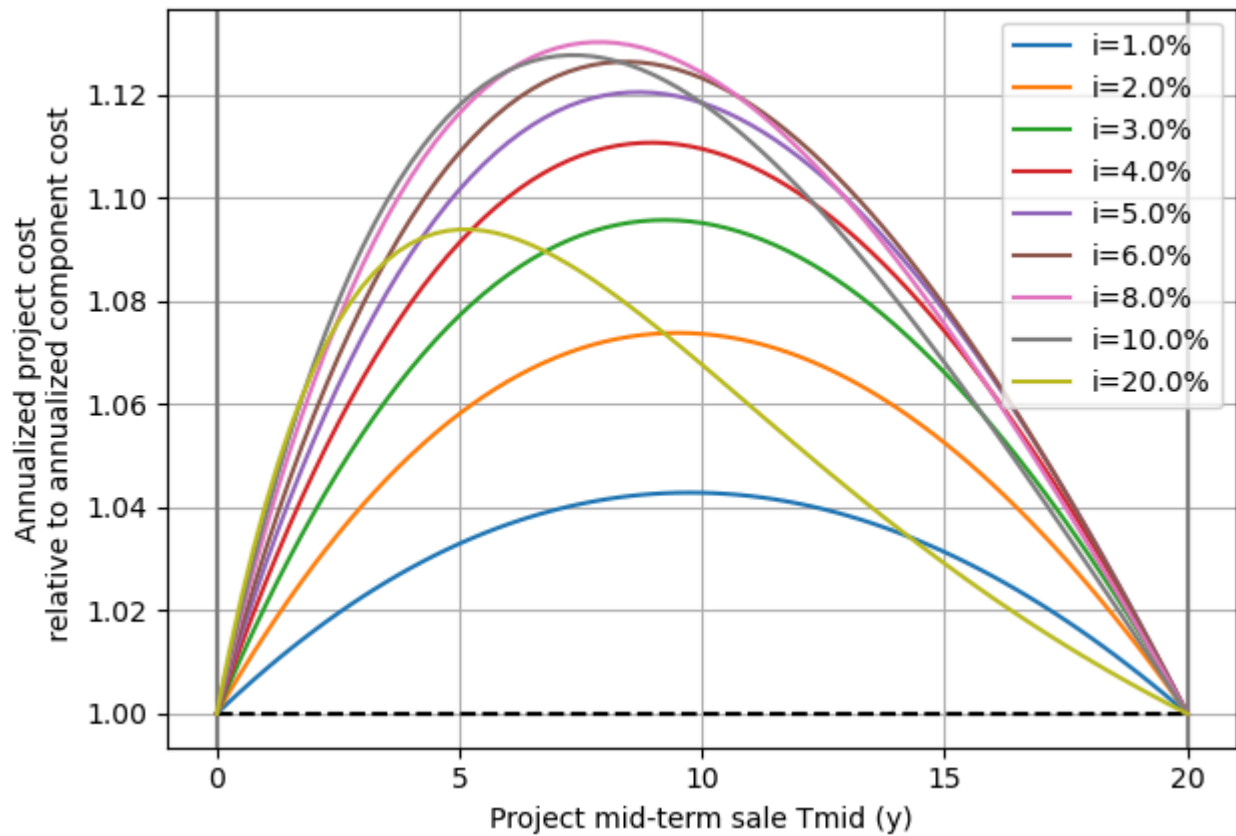
Like for the first graph of the economic inconsistency of the classical definition without mid-term sale, the effect is not so easy to analyze. We observe that:

- the overestimation of annualized project cost is worst (strongest) when the mid-term sale happens at half-project life for "small enough" discount rates
  - **it can reach +13%**
  - the maximum is reached for mid-term sale happening before half-project in the case of "high" discount rates
- this overestimation is zero for  $T_{mid} = 0$  or  $T_p$
- the effect of the discount rate is *complex*, like for the case without mid-term sale:
  - no overestimation when discount is zero (since  $S_a = S_b$  in that case)
  - the overestimation grows quickly with discount rate, as long as it is "small enough" (already +4% for half-project sale when  $i = 1\%$  only)
  - beyond a certain threshold of discount rate, the overestimation is near constant or even decreasing

Also, repeating the plot for different values of the component lifetime  $T_c$ , it seems that the worst case overestimation is attained for a discount rate slightly below  $2/T_c$ . This is similar (at least approximately) to the case without mid-term sale.

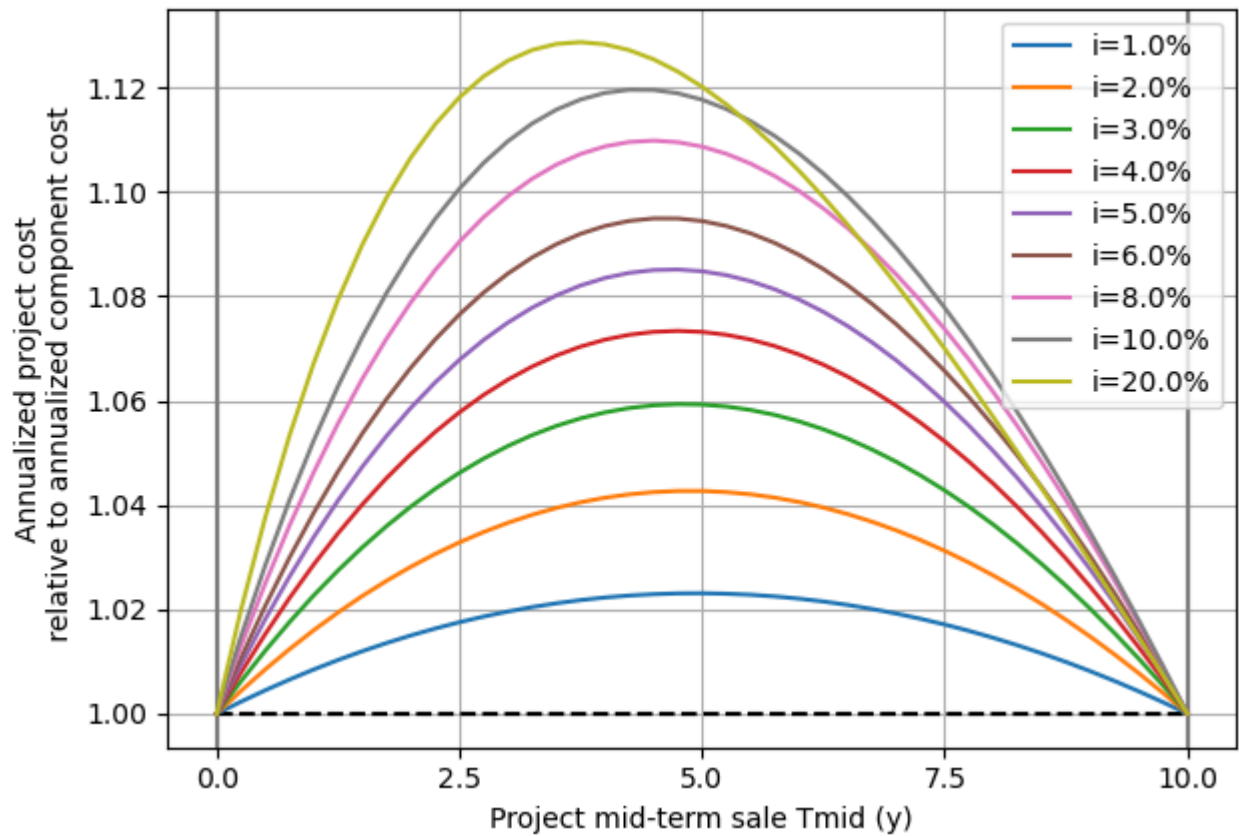
```
In [74]: #i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many
plot_Cann_Tmid(i_list, Tc=20);
```

Annualized project cost of a component with lifetime  $T_c=20$  y when project is sold at mid-term and immediately bought back



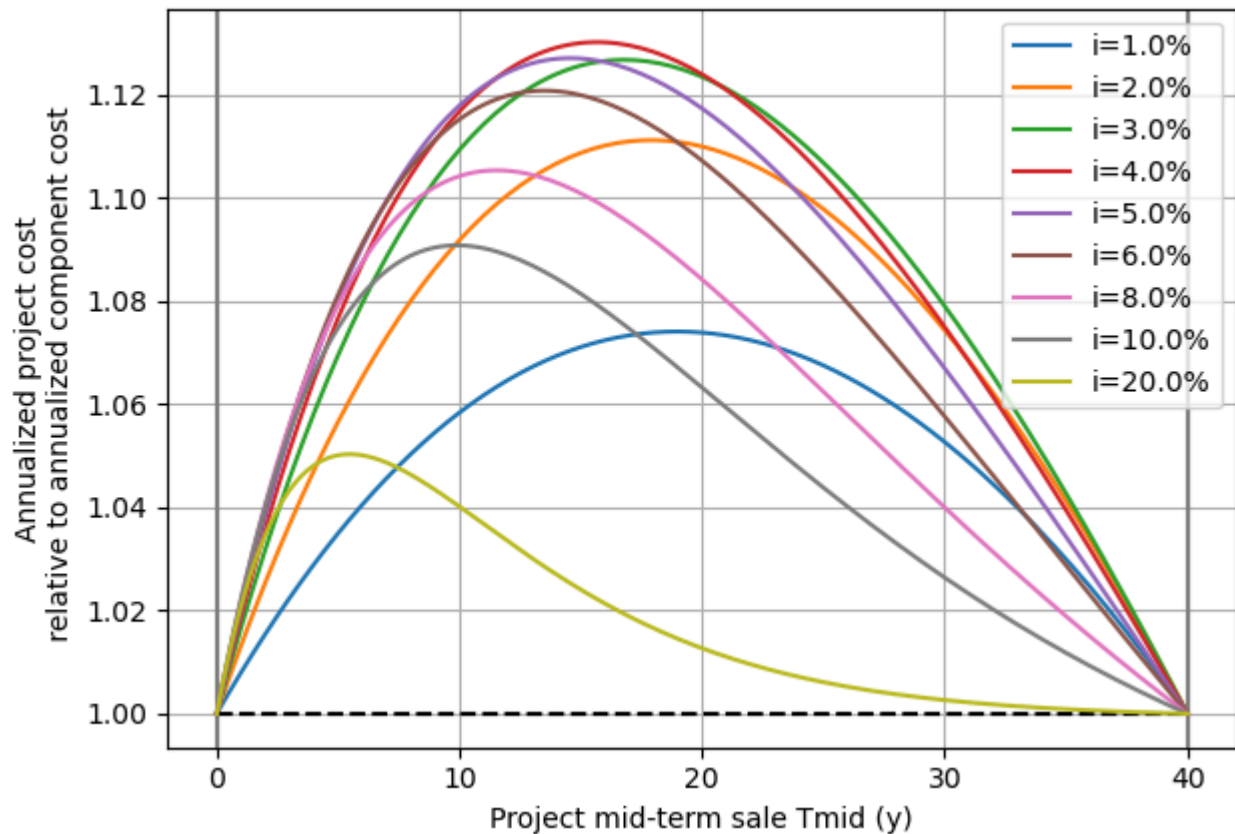
```
In [75]: #i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many
plot_Cann_Tmid(i_list, Tc=10);
```

Annualized project cost of a component with lifetime  $T_c=10$  y when project is sold at mid-term and immediately bought back



```
In [76]: #i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates  
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many  
plot_Cann_Tmid(i_list, Tc=40);
```

Annualized project cost of a component with lifetime  $T_c=40$  y when project is sold at mid-term and immediately bought back



## Salvage value when Operation and Maintenance costs are time-varying

Throughout this document, we haven't discussed Operation and Maintenance (O&M) costs, because when they are constant (and this was our implicit assumption, as done in HOMER), there can be simply added/subtracted from the annualized cost computation as needed.

If interested (Jones 1990) generalizes the formula in the case of time-varying Operation and Maintenance costs.

## References

- T. Lambert, P. Gilman, and P. Lilienthal, "Micropower system modeling with HOMER," in *Integration of Alternative Sources of Energy* (F. A. Farret and M. G. Simões, eds.), John Wiley & Sons, Dec. 2005.  
<https://www.homerenergy.com/documents/MicropowerSystemModelingWithHOMER.pdf>
- HOMER Software, "Capital Recovery Factor", in *HOMER Pro 3.16 documentation*, 2024.  
[https://support.ul-renewables.com/homer-manuals-pro/capital\\_recovery\\_factor.html](https://support.ul-renewables.com/homer-manuals-pro/capital_recovery_factor.html)
- HOMER Software, "Salvage value", in *HOMER Pro 3.16 documentation*, 2024.  
[https://support.ul-renewables.com/homer-manuals-pro/salvage\\_value.html](https://support.ul-renewables.com/homer-manuals-pro/salvage_value.html)
- P. C. Jones, W. J. Hopp, and J. L. Zydiak, "Generalized Imputed Salvage Values," *The Engineering Economist*, vol. 35, no. 3, pp. 215–229, Jan. 1990, doi: [10.1080/00137919008903018](https://doi.org/10.1080/00137919008903018).

- G. J. Thuesen and W. J. Fabrycky, Engineering economy, 6th ed. Prentice-Hall, 1984.  
Available: [https://archive.org/details/engineeringecono0000thue\\_t6g6/](https://archive.org/details/engineeringecono0000thue_t6g6/)

## Appendix

Remark: the code cells above can only be run once the cells in this appendix are run.

```
In [1]: import sympy
from sympy import symbols, series, simplify, log

import numpy as np
import scipy.optimize as opt
import matplotlib.pyplot as plt
```

```
In [2]: def CRF(i,T):
    'Capital Recovery Factor'
    a = (1+i)**T
    return i*a/(a-1)
```

In the following, we assume that the investment cost  $C_c = 1$  without loss of generality. Only, the salvage value discussed in the following is fact a *relative* salvage value (with true value  $S = S_{relative} \cdot C_c$ ).

### Proof of the economic consistency of salvage value definition b

```
In [3]: i, S = symbols('i S', real=True)
Tc, Tp = symbols('T_c T_p', positive=True, integer=True)
# integer Tc, Tp needed for (1/x)**T simplification into 1/x**(-T),
# because the alternative assumption that x=1+i>=0 cannot be specified in SymPy
# The most robust solution would be to introduce a symbol for 1+i, assumed posi
# or 1/(1+i) like Jones,
```

Annualized cost of the component over its lifetime:

```
In [4]: Cann_c = CRF(i, Tc)*1
Cann_c
```

```
Out[4]: 
$$\frac{i(i+1)^{T_c}}{(i+1)^{T_c} - 1}$$

```

```
In [5]: NPCp = 1 - S/(1+i)**Tp
NPCp
```

```
Out[5]: 
$$-S(i+1)^{-T_p} + 1$$

```

```
In [6]: Cann_p = CRF(i, Tp) * NPCp
Cann_p
```

```
Out[6]: 
$$\frac{i(i+1)^{T_p} \left( -S(i+1)^{-T_p} + 1 \right)}{(i+1)^{T_p} - 1}$$

```

We inject the proposed definition into the project annualized cost to show that it indeed gets equal to the component annualized cost:

```
In [7]: Sb = ((1+i)**Tc - (1+i)**Tp)/((1+i)**Tc - 1) * 1
Sb
```

```
Out[7]: 
$$\frac{(i+1)^{T_c} - (i+1)^{T_p}}{(i+1)^{T_c} - 1}$$

```

```
In [8]: simplify(Cann_p.subs({S:Sb})) # == Cann_c
```

```
Out[8]: 
$$\frac{i(i+1)^{T_c}}{(i+1)^{T_c} - 1}$$

```

```
In [9]: simplify(Cann_p.subs({S:Sb}) - Cann_c) # == 0
```

```
Out[9]: 0
```

## Comparison with “imputed” or “implied” salvage value

Comparison with formulae from the Engineering Economics literature.

“Imputed” salvage value formula in (Jones 1990), using their notations:

$$ISV = \left[ (p - s_n \delta^n) \frac{1 - \delta}{\delta(1 - \delta^n)} \right] \left[ \frac{\delta(1 - \delta^{n-n^*})}{(1 - \delta)} \right] + s_n \delta^{n-n^*}$$

with:

- capital cost investment  $p$
- discount factor  $\delta = 1/(1 + i)$
- component lifetime  $n$  (asset economic life in the article).  $T_c$  in our notation.
- project study period  $n^*$  ( $n^* \geq n$ ).  $T_p$  in our notation.
- component salvage value at the end of its economic life  $s_n$

Since here we consider *technical* and not *economic* component lifetime, we are in the special case  $s_n = 0$ , so the formula simplifies to

$$ISV = \left[ p \frac{1 - \delta}{\delta(1 - \delta^n)} \right] \left[ \frac{\delta(1 - \delta^{n-n^*})}{(1 - \delta)} \right]$$

and several other factors cancel out:

$$ISV = p \frac{1 - \delta^{n-n^*}}{1 - \delta^n}$$

Implementation note: due to the way SymPy handles powers for complex numbers, the necessary simplifications needs some strange ways to specify things. In particular, I would need to specify that  $i \geq -1$  so that  $1 + i \geq 0$ , but this is not possible with SymPy's per-

symbol `assumption` mechanism. See discussion

<https://groups.google.com/g/sympy/c/9HDZLOPHMH4/m/r14AILbEAQAJ>

```
In [10]: a = symbols('a')
x = symbols('x', positive=True)
simplify(a**x * (1/a)**x) # subtly different from simplify(a**x * (1/(a**x))) w
```

```
Out[10]: ax (1/a)x
```

```
In [11]: simplify(a**x * (1/(a**x)))
```

```
Out[11]: 1
```

Simplification OK if both are positive

```
In [12]: a = symbols('a', positive=True)
x = symbols('x', positive=True)
simplify(a**x * (1/a)**x)
```

```
Out[12]: 1
```

Alternative: assume x is an integer

```
In [13]: a = symbols('a')
x = symbols('x', integer=True)
simplify(a**x * (1/a)**x)
```

```
Out[13]: 1
```

```
In [14]: δ = 1/(1+i)
ISV = (1-δ**(Tc-Tp))/(1-δ**Tc)
ISV
```

```
Out[14]: 
$$\frac{1 - (i + 1)^{-T_c + T_p}}{1 - (i + 1)^{-T_c}}$$

```

```
In [15]: Sb
```

```
Out[15]: 
$$\frac{(i + 1)^{T_c} - (i + 1)^{T_p}}{(i + 1)^{T_c} - 1}$$

```

```
In [16]: simplify(Sb - ISV) # == 0
```

```
Out[16]: 0
```

So (Jones 1990) imputed salvage value is indeed the same as our "consistent" one.

Now, (Jones 1990) cites the textbook (Thuesen 1984), where I found a similar discussion in *Ch 7 Decision Making Among Alternatives*, specifically in *section 7.10 Comparison of Alternatives with Unequal Service Lives*. There is introduced the project study period  $n^*$  ( $T_p$  in our notation) with  $n^* \neq n$  (with  $n$  being the alternative's life,  $T_c$  in our notation). The book states there are several methods:

Because alternatives being compared must be judged over the same study period, various assumptions are utilized to places alternatives with unequal lives within the same study period.

Authors classifies these methods depending on the situation whether  $n^* > n$  or  $n^* < n$  for one or any of the alternatives under study. They start the discussion with the case *Alternative's life longer than study period* ( $n^* < n$  for at least one alternative). Then the economic analysis of that long lived asset should include an "implied salvage value or unused value" to "be imputed at the end of the study period". Thuesen & Fabrycky state that two estimations can be used:

1. actual market value of the partially used asset (how much can it be sold?)
2. compute annual equivalent amount (AE) of each alternative over its service life (the life of each asset being different)

Approach 2 "is widely applied because of its ease of calculation" and works for "any study period less than the life of the shortest lived alternative" (extreme case:  $n^* = 1$  year). Indeed, the study period duration doesn't enter the computation.

However, authors go on to underline that Approach 2 embodies an implied salvage value at the end of the study period. Notation  $F_{n^*}$ , which is solution of the annual cost equivalence equation:

$$\underbrace{-p \cdot (A/P, i, n) + F_n(A/F, i, n)}_{\text{AE over actual service life } n} = \underbrace{-p \cdot (A/P, i, n^*) + F_{n^*}(A/F, i, n^*)}_{\text{AE over study period } n^*}$$

with  $F_n$  the component salvage value at the end of its economic life ( $s_n$  in Jones 1990) and using Jones 1990 notation for  $p$ , the capital cost investment.

(Thuesen 1984) uses the following notations for the conversion to one-time payments to annual payments (defined in chapter 3):

- $P$  present payment,  $F$  future payment,  $A$  annual recurring payment
- $(A/P, i, n)$ : Equal-payment-series capital-recovery factor, at annual interest rate  $i$ , over  $n$  years (p 42-43)
  - This is the CRF in this document.
- $(A/F, i, n)$ : Equal-payment-series sinking-fund factor (p 41):  $= CRF \cdot \frac{1}{(1+i)^n}$

As with (Jones 1990), we only focus on the special case where there is zero salvage value at the end of asset life:  $F_n = 0$ , so

$$F_{n^*} = -p \frac{(A/P, i, n) - (A/P, i, n^*)}{(A/F, i, n^*)}$$

```
In [17]: S_T84 = -(CRF(i, Tc) - CRF(i, Tp)) / (CRF(i, Tp) / (1+i)**Tp)
S_T84 = simplify(S_T84)
S_T84
```



$$\text{Out[17]: } \frac{(i+1)^{T_c} - (i+1)^{T_p}}{(i+1)^{T_c} - 1}$$

In [18]: `simplify(S_T84 - Sb)`

Out[18]: 0

Thuesen discussion goes on with an alternative formula (again rewritten in the special case of  $F_n = 0$ )

$$F_{n^*} = p \cdot (A/P, i, n) \cdot (P/A, i, n - n^*)$$

that is using our notation:

$$S/p = \frac{CRF(i, T_c)}{CRF(i, T_c - T_p)}$$

and Thuesen comment this term as "the single-payment equivalent of the loss of capital that will be incurred after the study period"

In [19]: `S_T84_bis = CRF(i, Tc)/CRF(i, Tc-Tp)`  
`S_T84_bis = simplify(S_T84_bis)`  
`S_T84_bis`

$$\text{Out[19]: } \frac{(i+1)^{T_p} \left( (i+1)^{T_c - T_p} - 1 \right)}{(i+1)^{T_c} - 1}$$

In [20]: `simplify(S_T84_bis - Sb)`

Out[20]: 0

Finally, (Thuesen 1984, p 214) comments that the implied salvage value should be compared with an estimate of the actual salvage value. If there is a significant difference, then *the Annual Equivalent method should not be used*. This is where (Jones 1990) disagrees.

## Behavior of the economically consistent salvage value (definition b) for small discount rate

Observation: for small discount rate  $i$ ,  $S_b$  tends linearly to  $S_a$ :

$$S_b \sim S_a + \frac{T_p \cdot (T_c - T_p)}{2 \cdot T_c} i + O(i^2)$$

The corrective term is zero, as expected for:

- $T_p = 0$  (when  $S = 1$  since the component is new)
- $T_p = T_c$  (when  $S = 0$  since the component is at end of life)

Series expansion of  $S_b$  around  $i = 0$ :

```
In [21]: Sb_ser = series(Sb, i, n=2)
simplify(Sb_ser)
```

```
Out[21]: 
$$\frac{T_c (2 + O(i^2)) + T_p i (T_c - T_p) - 2T_p}{2T_c}$$

```

```
In [22]: SbTc_ser = series(Sb*Tc, i, n=2)
simplify(SbTc_ser)
```

```
Out[22]: 
$$-T_p + \frac{T_p i (T_c - T_p)}{2} + T_c + O(i^2)$$

```

limit behavior of the difference with the classical definition  $S_a$ :

```
In [23]: Sa = (Tc-Tp)/Tc
```

```
In [24]: simplify((Sb_ser - Sa))
```

```
Out[24]: 
$$\frac{-T_p^2 i + T_c T_p i + O(i^2)}{2T_c}$$

```

```
In [25]: simplify((SbTc_ser - Sa*Tc))
```

```
Out[25]: 
$$\frac{T_p i (T_c - T_p)}{2} + O(i^2)$$

```

## Decrease of the economically consistent salvage value (definition b) for small project lifetime

Since the classical definition of salvage value is linear in project lifetime  $T_p$ , its derivative is constant equal to  $-1/T_c$ .

Compared to this, the economically consistent salvage value (definition b) is nonlinear in  $T_p$ . Its derivative for small project values is less negative:

```
In [26]: Sb.diff(Tp).subs({Tp:0})
```

```
Out[26]: 
$$-\frac{\log(i+1)}{(i+1)^{T_c} - 1}$$

```

```
In [27]: simplify(series(Sb.diff(Tp).subs({Tp:0}), i, n=3))
```

```
Out[27]: 
$$-\frac{1}{T_c} + \frac{i}{2} - \frac{i^2}{4} - \frac{T_c i^2}{12} + O(i^3)$$

```

## Economic inconsistency of the classical definition

Injecting now the classical salvage value definition into the annual project cost:

```
In [28]: Cann_p_rel = Cann_p.subs({S:Sa})/Cann_c
Cann_p_rel = simplify(Cann_p_rel)
Cann_p_rel
```

$$\text{Out[28]: } \frac{(i+1)^{-T_c} \left( (i+1)^{T_c} - 1 \right) \left( T_c(i+1)^{T_p} - T_c + T_p \right)}{T_c \left( (i+1)^{T_p} - 1 \right)}$$

## Inconsistency for short projects

Focus on short projects (we need to take the limit  $T_p \rightarrow 0$ , since the denominator is zero at  $T_p = 0$ ):

```
In [29]: Cann_p_rel_Tp0 = Cann_p_rel.limit(Tp, 0)
Cann_p_rel_Tp0
```

$$\text{Out[29]: } \frac{(i+1)^{-T_c} (T_c \log(i+1) + 1) \left( (i+1)^{T_c} - 1 \right)}{T_c \log(i+1)}$$

To analyze the effect of discount rate (which is not monotonic as visualized with `plot_Cann_Tp0_i`), we turn to a series decomposition:

- linear effect is positive:  $+T_c/2 \times i$ , which explains the increase in overestimation for small discount rate
- quadratic effect is negative  $-T_c/4 \times i^2$  which explains the decrease when the discount rate gets "big enough"

```
In [30]: Cann_p_rel_Tp0_ser = simplify(Cann_p_rel_Tp0.series(i, 0, n=3))
Cann_p_rel_Tp0_ser
```

$$\text{Out[30]: } 1 + \frac{T_c i}{2} - \frac{T_c i^2}{4} - \frac{T_c^2 i^2}{3} + O(i^3)$$

```
In [31]: Cann_p_rel_Tp0_ser.diff(i)
```

$$\text{Out[31]: } \frac{T_c}{2} - \frac{T_c i}{2} - \frac{2T_c^2 i}{3} + O(i^2)$$

```
In [32]: sympy.solve(Cann_p_rel_Tp0_ser.diff(i), i)[0]
```

$$\text{Out[32]: } \frac{3(T_c + O(i^2))}{T_c(4T_c + 3)}$$

And this series is maximal (zero of the derivative) at:

$$i = \frac{1}{1 + 4/3 \cdot T_c}$$

```
In [33]: i_worst = 1/(1+4*Tc/3)
```

```
In [34]: simplify(Cann_p_rel_Tp0_ser.diff(i).subs({i:i_worst})) # = 0 + O(...)
```

$$\text{Out[34]: } O\left(\frac{1}{T_c^2}; T_c \rightarrow \infty\right)$$

However, the maximal value is not a constant:

```
In [35]: simplify(Cann_p_rel_Tp0_ser.subs({i:i_worst}))
```

```
Out[35]: 
$$\frac{12 + 19T_c + O\left(\frac{1}{T_c^2}; T_c \rightarrow \infty\right)}{4 \cdot (4T_c + 3)}$$

```

For further analysis, we simplify the expression `Cann_p_rel_Tp0` by recognizing a function  $x \mapsto C(x)$  evaluated at  $x = (1 + i)^{T_c}$ :

```
In [36]: x = symbols('x', positive=True)
Cx = (1+log(x))*(x-1)/(x*log(x))
Cx
```

```
Out[36]: 
$$\frac{(x - 1)(\log(x) + 1)}{x \log(x)}$$

```

Check that the expression in  $x$  is indeed equivalent to `Cann_p_rel_Tp0` for  $x = (1 + i)^{T_c}$ :

```
In [37]: simplify(Cx.subs({x:(1+i)**Tc}) - Cann_p_rel_Tp0)
```

```
Out[37]: 
$$\frac{1}{\log\left((i + 1)^{T_c}\right)} - \frac{(i + 1)^{-T_c}}{\log\left((i + 1)^{T_c}\right)} - \frac{1}{T_c \log(i + 1)} + \frac{(i + 1)^{-T_c}}{T_c \log(i + 1)}$$

```

Remark: SymPy doesn't see the equality between logs at the denominator, because it needs a *positivity* assumption:

```
In [38]: simplify(log(x**2) - 2*log(x))
```

```
Out[38]: 0
```

```
In [39]: a = symbols('a') # without positive=True
simplify(log(a**2) - 2*log(a))
```

```
Out[39]: 
$$-2\log(a) + \log(a^2)$$

```

Now looking at the derivative:

```
In [40]: simplify(Cx.diff(x))
```

```
Out[40]: 
$$\frac{-x + \log(x)^2 + \log(x) + 1}{x^2 \log(x)^2}$$

```

```
In [41]: num = Cx.diff(x)*x**2*log(x)**2
num = simplify(num)
num
```

```
Out[41]: 
$$-x + \log(x)^2 + \log(x) + 1$$

```

There is no analytical root that SymPy can find:

```
In [42]: # sympy.solve(num, x) # NotImplementedError
```

Only numerical root finding works:  $x^* \approx 6.0091$

```
In [43]: opt.root_scalar(lambda x: 1 - x + np.log(x) + (np.log(x))**2, bracket = (5,10))
```

```
Out[43]:      converged: True
          flag: converged
          function_calls: 8
          iterations: 7
          root: 6.009142941081862
          method: brentq
```

And this is coherent with the numerical maximization of  $C(x)$ :

```
In [44]: Cx_fun = lambda x: (1+np.log(x))*(x-1)/(x*np.log(x))
          Cx_fun(6.01-0.5), Cx_fun(6.01), Cx_fun(6.01+0.5)
```

```
Out[44]: (1.2981372299707084, 1.298425606777053, 1.298198390999316)
```

```
In [45]: Cx_minus_fun = lambda x: -Cx_fun(x)
          res = opt.minimize_scalar(Cx_minus_fun, [2,6,10], bounds=[2,10])
          Cx_max = -res.fun # 1.2984
          Cx_argmax = res.x # 6.0091
          print(res)
```

```
message: Solution found.
success: True
status: 0
      fun: -1.298425607525638
       x: 6.009143977032298
      nit: 12
     nfev: 12
```

```
In [46]: np.log(Cx_argmax)
```

```
Out[46]: 1.793282305296451
```

Therefore, the worst case economic inconsistency happens for pairs  $(i, T_c)$  which satisfies:

$$(1 + i)^{T_c} = x^* \approx 6.01$$

and this equation can be solved for the discount factor  $i$ :

$$i = \exp\left(\frac{\log(x^*)}{T_c}\right) - 1$$

and since  $\log(x^*) \approx 1.79 \ll T_c$  for most component lifetime values, we can linearize exp:

$$i \approx \frac{\log(x^*)}{T_c} \approx \frac{1.79}{T_c}$$

This observation is coherent with the graph obtained with `plot_Cann_Tp0_i`: the worst case discount factor is about *twice the inverse lifetime*.

## Proof of the economic consistency with replacements

## with one replacement

That is  $T_p \in [T_c, 2.T_c]$  and assuming that replacement cost is equal to the initial investment.

NPC becomes:

```
In [47]: NPCp_1rep = 1 + 1/(1+i)**Tc - S/(1+i)**Tp
NPCp_1rep
```

```
Out[47]: -S(i + 1)-Tp + 1 + (i + 1)-Tc
```

```
In [48]: Cann_p_1rep = CRF(i, Tp) * NPCp_1rep
Cann_p_1rep
```

```
Out[48]: 
$$\frac{i(i+1)^{T_p} \left( -S(i+1)^{-T_p} + 1 + (i+1)^{-T_c} \right)}{(i+1)^{T_p} - 1}$$

```

As mentioned in the section *Project with one replacement cost*, there is an ambiguity about which term of the numerator to update in the salvage value definition. So we create both possible expressions and inject them in the project annualized cost.

We show that replacing  $T_c$  becomes  $2.T_c$  (i.e. component end of life happen twice later) is incorrect. The correct update of the definition is to replace  $T_p$  becomes  $T_p - T_c$  (i.e. the usage duration of the component).

```
In [49]: Sb_1rep_incorrect = ((1+i)**(2*Tc) - (1+i)**Tp)/((1+i)**Tc - 1) * 1
Sb_1rep_incorrect
```

```
Out[49]: 
$$\frac{(i+1)^{2T_c} - (i+1)^{T_p}}{(i+1)^{T_c} - 1}$$

```

```
In [50]: simplify(Cann_p_1rep.subs({S:Sb_1rep_incorrect})) # != Cann_c
```

```
Out[50]: 
$$\frac{i(i+1)^{-T_c} \left( (i+1)^{T_c} \left( -(i+1)^{2T_c} + (i+1)^{T_p} \right) + (i+1)^{T_p} \left( (i+1)^{T_c} - 1 \right) + (i+1)^{T_c+T_p} \left( (i+1)^{T_c} - (i+1)^{T_p} \right) \right)}{\left( (i+1)^{T_c} - 1 \right) \left( (i+1)^{T_p} - 1 \right)}$$

```

```
In [51]: simplify(Cann_p_1rep.subs({S:Sb_1rep_incorrect}) - Cann_c) # != 0
```

```
Out[51]: 
$$\frac{i(i+1)^{-T_c} \left( -(i+1)^{2T_c} + (i+1)^{T_p} \right)}{(i+1)^{T_p} - 1}$$

```

```
In [52]: Sb_1rep = ((1+i)**Tc - (1+i)**(Tp-Tc))/((1+i)**Tc - 1) * 1
Sb_1rep
```

```
Out[52]: 
$$\frac{(i+1)^{T_c} - (i+1)^{-T_c+T_p}}{(i+1)^{T_c} - 1}$$

```

```
In [53]: simplify(Cann_p_1rep.subs({S:Sb_1rep})) # == Cann_c
```

$$\text{Out[53]: } \frac{i(i+1)^{-T_c} \left( (i+1)^{T_c} \left( -(i+1)^{T_c} + (i+1)^{-T_c+T_p} \right) + (i+1)^{T_p} \left( (i+1)^{T_c} - 1 \right) + (i+1)^{T_c+T_p} \left( (i+1)^T \right)}{\left( (i+1)^{T_c} - 1 \right) \left( (i+1)^{T_p} - 1 \right)}$$

In [54]: `simplify(Cann_p_1rep.subs({S:Sb_1rep}) - Cann_c) # == 0`

Out[54]: 0

## With two replacements

That is  $T_p \in [2.T_c, 3.T_c]$  and again assuming that replacement cost is equal to the initial investment.

NPC becomes:

In [55]: `NPCp_2rep = 1 + 1/(1+i)**Tc + 1/(1+i)**(2*Tc) - S/(1+i)**Tp`  
`NPCp_2rep`

$$\text{Out[55]: } -S(i+1)^{-T_p} + 1 + (i+1)^{-T_c} + (i+1)^{-2T_c}$$

In [56]: `Cann_p_2rep = CRF(i, Tp) * NPCp_2rep`  
`Cann_p_2rep`

$$\text{Out[56]: } \frac{i(i+1)^{T_p} \left( -S(i+1)^{-T_p} + 1 + (i+1)^{-T_c} + (i+1)^{-2T_c} \right)}{(i+1)^{T_p} - 1}$$

The definition of salvage value is updated by replacing  $T_p$  by  $T_p - 2.T_c$  (i.e. the usage duration of the last component).

In [57]: `Sb_2rep = ((1+i)**Tc - (1+i)**(Tp-2*Tc))/((1+i)**Tc - 1) * 1`  
`Sb_2rep`

$$\text{Out[57]: } \frac{(i+1)^{T_c} - (i+1)^{-2T_c+T_p}}{(i+1)^{T_c} - 1}$$

In [58]: `simplify(Cann_p_2rep.subs({S:Sb_2rep})) # == Cann_c`

$$\text{Out[58]: } \frac{i(i+1)^{T_c}}{(i+1)^{T_c} - 1}$$

In [59]: `simplify(Cann_p_2rep.subs({S:Sb_2rep}) - Cann_c) # == 0`

Out[59]: 0

## Proof of the economic consistency with a mid project sale and immediate buy back

We introduce  $T_{mid} < T_p$  the year of selling the project at mid project. There are two salvage values, one for the first sale at mid project ( $S_{mid}$ ) and then the usual salvage at the end of the

project.

NPC becomes:

```
In [60]: T_mid = symbols('T_mid', integer=True, positive=True)
S_mid, S_fin = symbols('S_mid S_fin')
```

```
In [61]: NPCp_1sale = 1 - S_mid/(1+i)**T_mid + 1/(1+i)**T_mid + - S_fin/(1+i)**Tp
NPCp_1sale
```

```
Out[61]: 
$$-S_{fin}(i+1)^{-T_p} - S_{mid}(i+1)^{-T_{mid}} + 1 + (i+1)^{-T_{mid}}$$

```

```
In [62]: Cann_p_1sale = CRF(i, Tp) * NPCp_1sale
Cann_p_1sale
```

```
Out[62]: 
$$\frac{i(i+1)^{T_p} \left( -S_{fin}(i+1)^{-T_p} - S_{mid}(i+1)^{-T_{mid}} + 1 + (i+1)^{-T_{mid}} \right)}{(i+1)^{T_p} - 1}$$

```

The definitions of the two salvage values are the same as the case of a project without replacement, but with reduced project duration:

- the mid project salvage uses  $T_{mid}$  as project duration
- the final salvage value uses  $T_p - T_{mid}$  as project duration

```
In [63]: Sb_mid = ((1+i)**Tc - (1+i)**(T_mid))/((1+i)**Tc - 1) * 1
Sb_mid
```

```
Out[63]: 
$$\frac{(i+1)^{T_c} - (i+1)^{T_{mid}}}{(i+1)^{T_c} - 1}$$

```

```
In [64]: Sb_fin = ((1+i)**Tc - (1+i)**(Tp-T_mid))/((1+i)**Tc - 1) * 1
Sb_fin
```

```
Out[64]: 
$$\frac{(i+1)^{T_c} - (i+1)^{-T_{mid}+T_p}}{(i+1)^{T_c} - 1}$$

```

```
In [65]: simplify(Cann_p_1sale.subs({S_fin:Sb_fin, S_mid:Sb_mid})) # == Cann_c
```

```
Out[65]: 
$$\frac{i(i+1)^{T_c}}{(i+1)^{T_c} - 1}$$

```

```
In [66]: simplify(Cann_p_1sale.subs({S_fin:Sb_fin, S_mid:Sb_mid}) - Cann_c) # == 0
```

```
Out[66]: 0
```

## Plot functions

```
In [67]: def plot_salvage_Tp(i_list, Tc):
    """Salvage value with respect to project lifetime,
    for a given component lifetime Tc, for a list of discount rates
    """
    Tp = np.linspace(0, Tc, num=2*Tc+1)
```



```

Sa = (Tc-Tp)/Tc

fig, ax = plt.subplots()

ax.axvline(0, color='gray')
ax.axvline(Tc, color='gray')
ax.plot(Tp, Sa, 'k--', label='Sa (classical)')

for i in i_list:
    Sb = ((1+i)**Tc - (1+i)**Tp)/((1+i)**Tc - 1)
    ax.plot(Tp, Sb, label=f'Sb i={i:.1%}')

ax.grid()
ax.xaxis.major_locator.set_params(nbins=5)
ax.yaxis.major_locator.set_params(nbins=5)
ax.legend(loc='lower left')
ax.set(
    title=f'Salvage value of a component with lifetime Tc={Tc:.0f} y',
    xlabel='Project lifetime Tp (y)',
    ylabel='Salvage (relative to investment)'
)
fig.tight_layout()
return fig, ax

```

```

In [68]: def plot_Cann_Tp(i_list, Tc):
    """Annualized project cost (relative to component's annualized cost),
    with respect to project lifetime,
    when using the classical salvage value definition,
    for a given component lifetime Tc, for a list of discount rates
    """
    Tp = np.linspace(0+1e-3, Tc, num=2*Tc+1)
    Sa = (Tc-Tp)/Tc

    fig, ax = plt.subplots()

    ax.axvline(0, color='gray')
    ax.axvline(Tc, color='gray')

    # reference = 1 (when using economically consistent salvage value)
    ax.hlines(1, 0, Tc, colors='k', linestyle='--', label='')

    for i in i_list:
        Cann_c = CRF(i, Tc)
        NPCp = (1 - Sa/(1+i)**Tp)
        Cann_p = CRF(i, Tp) * NPCp
        Cann_p_rel = Cann_p/Cann_c
        ax.plot(Tp, Cann_p_rel, label=f'i={i:.1%}')

    ax.grid()
    ax.xaxis.major_locator.set_params(nbins=5)
    ax.yaxis.major_locator.set_params(nbins=5)
    ax.legend(loc='upper right')
    ax.set(
        title=f'Annualized project cost of a component with lifetime Tc={Tc:.0f} y',
        xlabel='Project lifetime Tp (y)',
        ylabel='Annualized project cost\n relative to annualized component cost'
    )
    fig.tight_layout()
    return fig, ax

```

```

In [69]: def plot_Cann_Tp0_i(Tc_list, i_max):
    """Annualized project cost (relative to component's annualized cost),
    with respect to discount rate from 0 to i_max,
    for a list of component lifetime Tc_list,
    when using the classical salvage value definition,
    when the project lifetime is small (Tp/Tc ~ 0)
    """
    Tp = Tc_list[0]/1000

    i_lin = np.linspace(1e-6, i_max, num=100)

    fig, ax = plt.subplots()
    ax.hlines(1, i_lin[0], i_lin[-1], colors='k', linestyle='--', label='')

    for Tc in Tc_list:
        Sa = (Tc-Tp)/Tc
        Cann_c = CRF(i_lin, Tc)
        NPCp = (1 - Sa/(1+i_lin)**Tp)
        Cann_p = CRF(i_lin, Tp) * NPCp
        Cann_p_rel = Cann_p/Cann_c

        ax.plot(i_lin, Cann_p_rel, label=f'Tc={Tc:.0f} y')

    ax.grid()
    ax.xaxis.major_locator.set_params(nbins=5)
    ax.yaxis.major_locator.set_params(nbins=5)
    ax.legend(loc='upper right')
    ax.set(
        title=r'Annualized project cost of a component, for short project ($T_p
        xlabel='Discount rate $i$',
        ylabel='Annualized project cost\n relative to annualized component cost
        )
    fig.tight_layout()
    return fig, ax

```

```

In [70]: def plot_Cann_Tmid(i_list, Tc):
    """Annualized project cost (relative to component's annualized cost),
    with respect to mid-term project sale (with immediate buy back),
    when using the classical salvage value definition,
    with assumption Tp=Tc (no replacement),
    for a given component lifetime Tc, for a list of discount rates.
    """
    Tp = Tc # assumption of no replacement and no salvage if there were no mid-
    Tmid = np.linspace(0, Tc, num=4*Tc+1)

    # Classical salvage values:
    Sa_mid = (Tc-Tmid)/Tc
    Sa_fin = (Tc-(Tp-Tmid))/Tc

    fig, ax = plt.subplots()

    ax.axvline(0, color='gray')
    ax.axvline(Tc, color='gray')

    # reference = 1 (when using economically consistent salvage values)
    ax.hlines(1, 0, Tc, colors='k', linestyle='--', label='')

    for i in i_list:
        Cann_c = CRF(i, Tc) # Component alone, for reference

```

```

NPCp = (1 - Sa_mid/(1+i)**Tmid + 1/(1+i)**Tmid - Sa_fin/(1+i)**Tp)
Cann_p = CRF(i, Tp) * NPCp
Cann_p_rel = Cann_p/Cann_c
ax.plot(Tmid, Cann_p_rel, label=f'i={i:.1%}')

ax.grid()
ax.xaxis.major_locator.set_params(nbins=5)
ax.legend(loc='upper right')
ax.set(
    title=f'Annualized project cost of a component with lifetime Tc={Tc:.0f}
          \n when project is sold at mid-term and immediately bought back
    xlabel='Project mid-term sale Tmid (y)',
    ylabel='Annualized project cost\n relative to annualized component cost
)
fig.tight_layout()
return fig, ax

```