Economic consistency of salvage value definitions

Pierre Haessig

Economic consistency of salvage value definitions

Pierre Haessig, April 2023

Abstract

This short report analyzes the definition of the *salvage value* of a component within an wider project investment analysis. The analysis shows that the classical definition is *close but not exacltly consistent* from an economic point of view. For an exact economic consistency of the salvage value, we derive an alternate defintition using a more complex formula. This formula is equivalent to the classical defintion when the discount factor is zero up to second order factors.

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 substracted from the project cost for each component having some remaining lifetime at the end of the project.

A key parameter in this analysis os 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 independant 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 (citation...) 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

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

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.

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.

In [59]:
```
i_list = [0.025, 0.050, 0.075, 0.10, 0.20] # list of discount rates
plot_salvage_Tp(i_list, Tc=20);
```

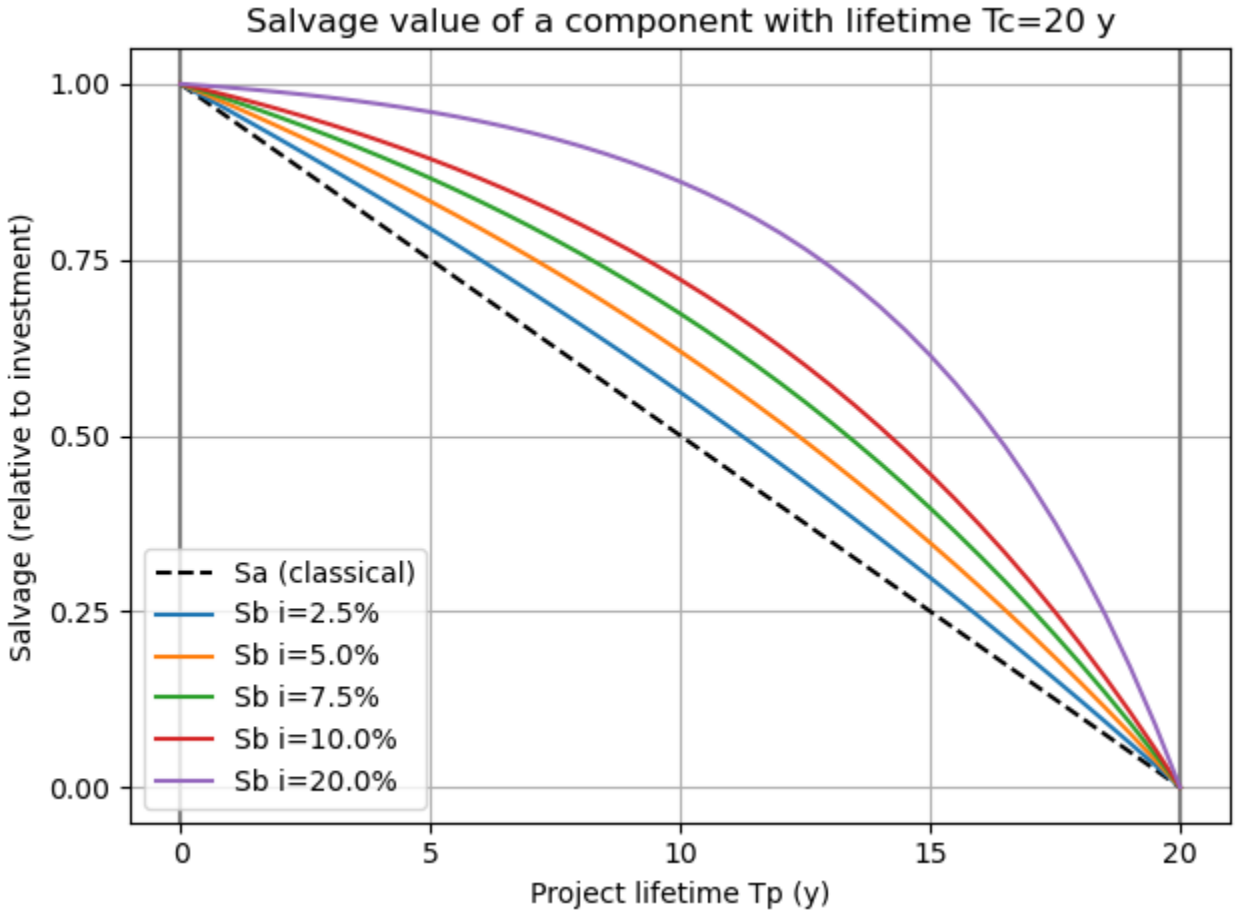


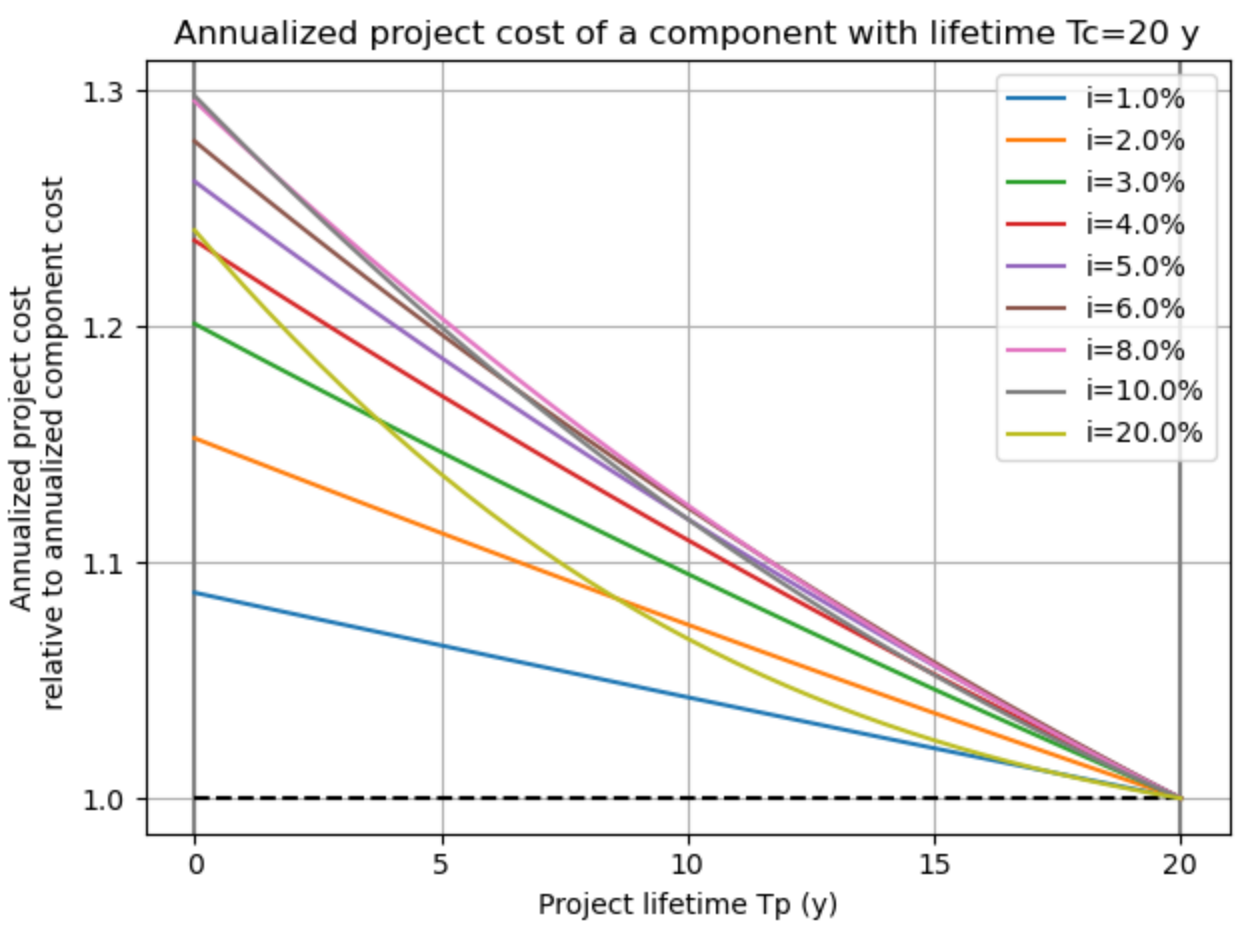
Illustration of the economic inconsistency of the classical definition

Choosing the classical salvage value when computing the annualized project cost creates a positive bias (cost is overestimated compared to the annulized 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 ($T_p/T_c \to 0$)
  - 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 [60]:
```
#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 discount
plot_Cann_Tp(i_list, Tc=20);
```

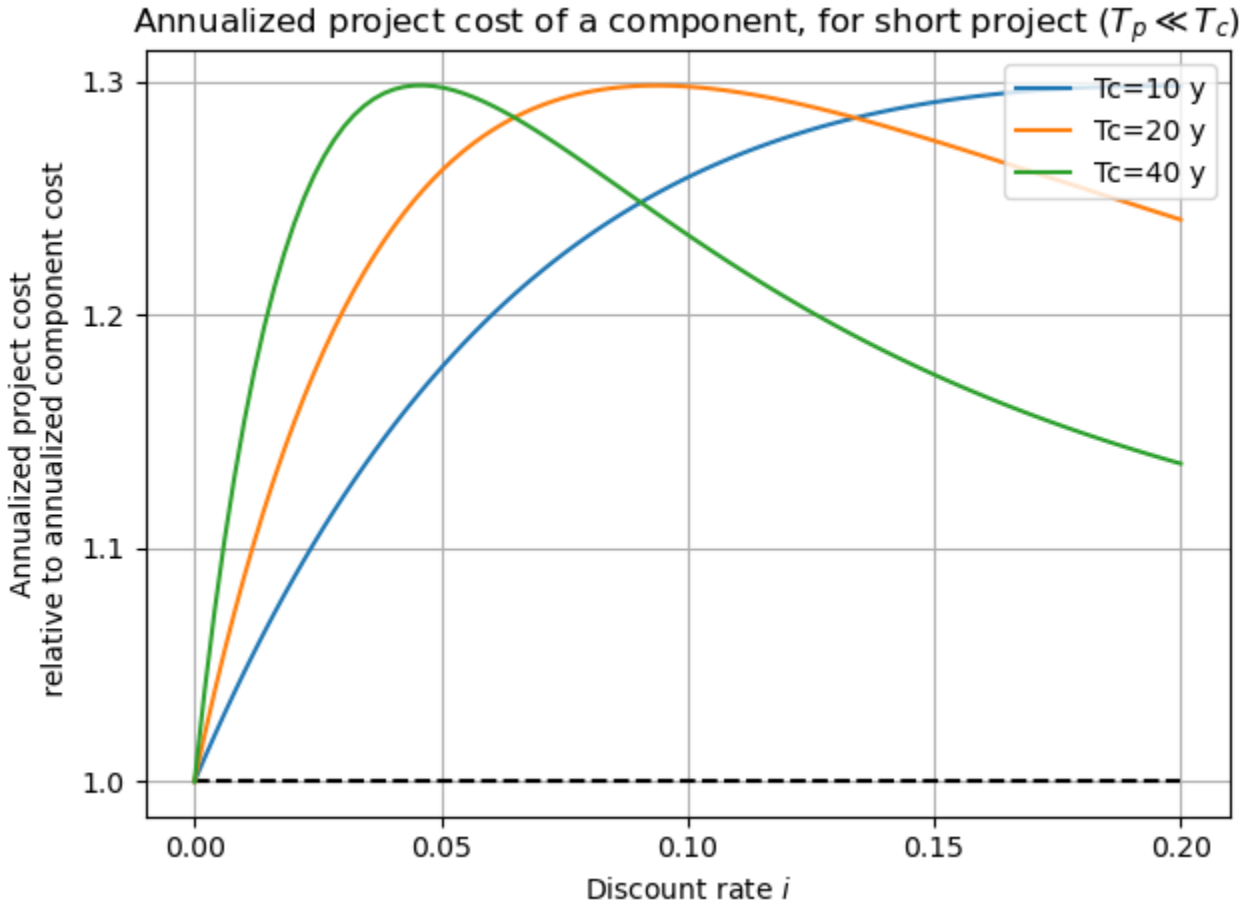


Now looking only at the start of the curves above, that is for short projects ($T_p/T_c \to 0$), we plot the effect of the discount rate, for different component lifetime. It appears that:

- 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 [61]:
```
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 prooves) that the annualized project cost in unchanged and equal to the annualized component cost

Project with replacements of the component

with one replacement

With one replacement (occuring 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 proove the economic consistency (annualized project cost = annualized component cost) we must of course assume that R_c, the nomial replacement cost of the component, is equal to the initial investement 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 prooved 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 prooved 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 proove 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 formulaes 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:

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

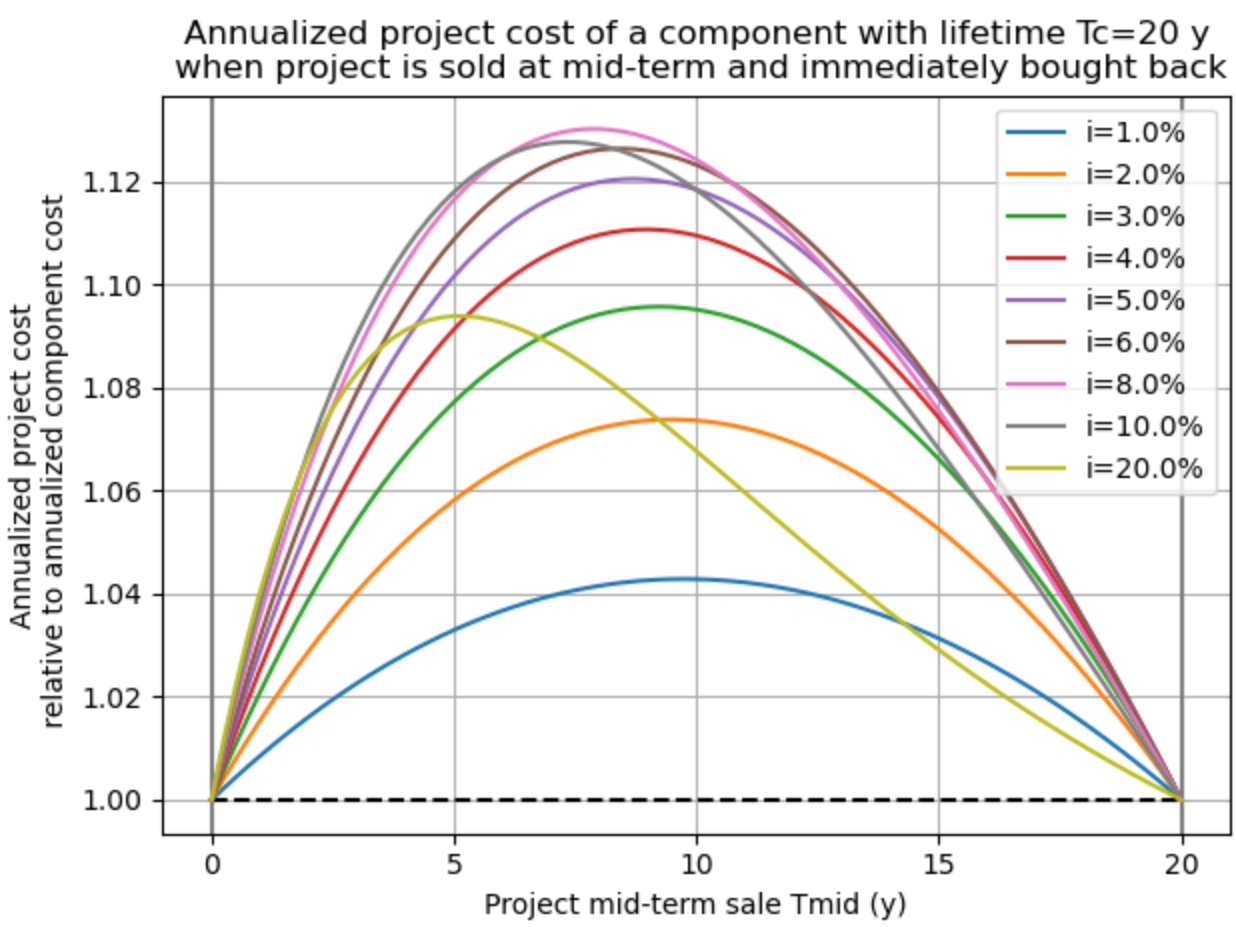
$$S_{a,final} = \frac{T_c - (T_p - T_{mid})}{T_c} C_c$$

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:

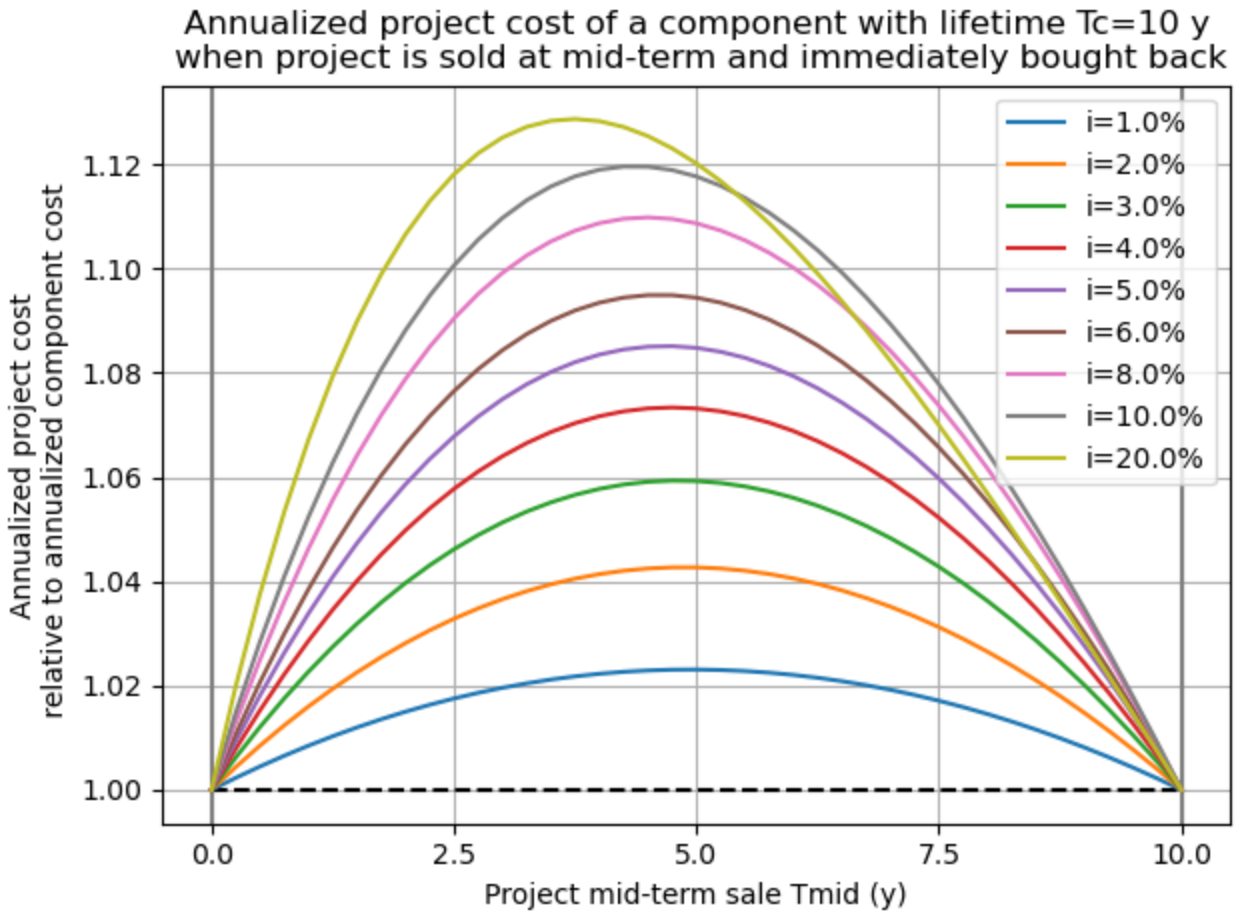
- the overestimation of annualized project cost is worst (strongest) when the mid-term sale happens at half-project life *for "small enough" discount rates*
  - **it can reach +13%**
  - the maximum is reached for mid-term sale happening before half-project in the case of "high" discount rates
- this 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 [80]:
```
#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 discount
plot_Cann_Tmid(i_list, Tc=20);
```

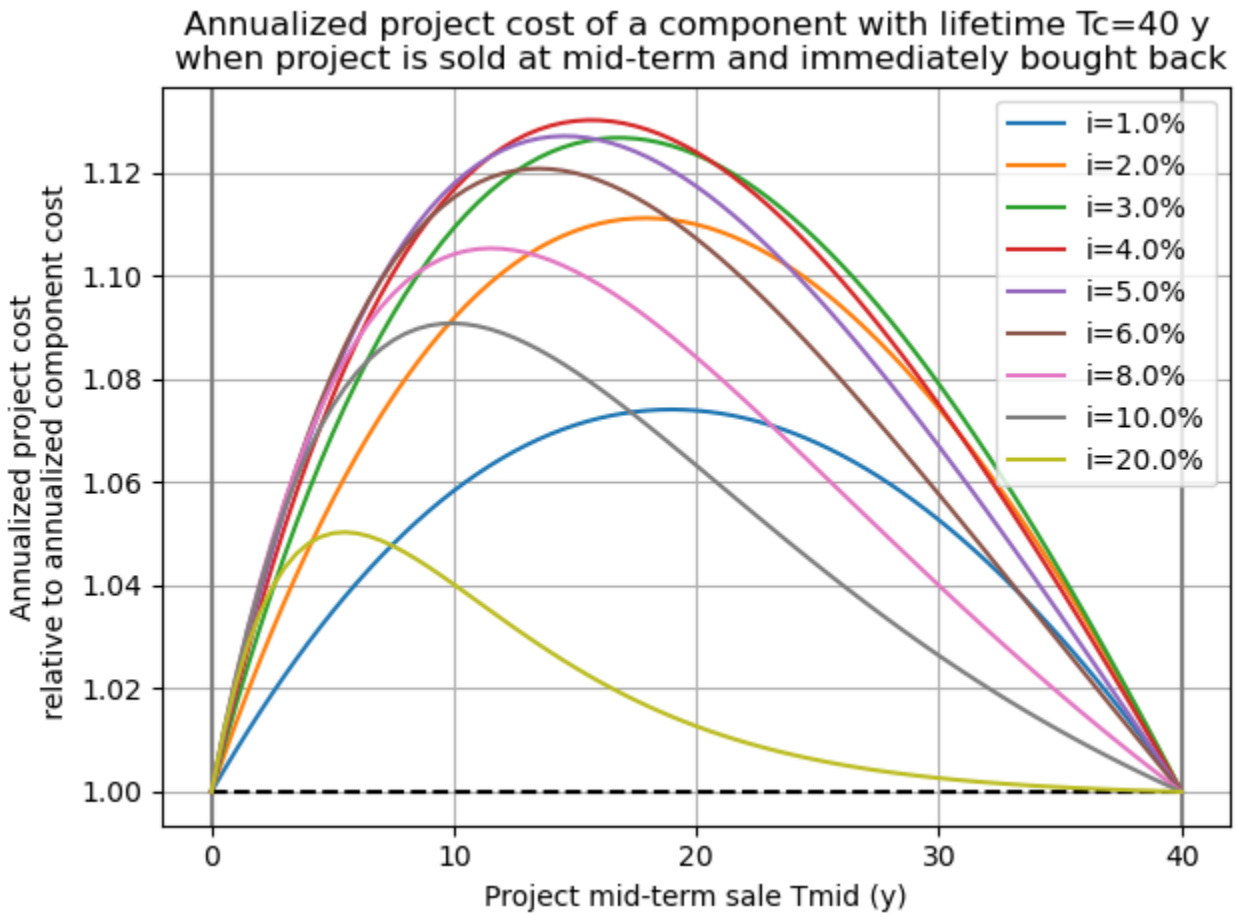


In [81]:
```
#i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many discount
plot_Cann_Tmid(i_list, Tc=10);
```




```
#i_list = [0.025, 0.050, 0.075, 0.100, 0.200] # list of discount rates
```

In [75]:
```
i_list = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.20] # list of many discount
plot_Cann_Tmid(i_list, Tc=40);
```



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 documentation*, 2023. https://www.homerenergy.com/products/pro/docs/latest/capital_recovery_factor.html

Appendix

Remark: the code cells above can only be run once the appendix is 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 $S = S_{relative}.C_c$).

Proof of the economic consistency of salvage value definition b

In [3]:
```
i, S = symbols('i S')
Tc, Tp = symbols('T_c T_p', positive=True)
```

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)
```

Out[9]:
$$0$$

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:

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

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 expension of S_b around $i = 0$:

In [10]:
```
Sb_ser = series(Sb, i, n=2)
simplify(Sb_ser)
```

Out[10]:
$$\frac{T_c\left(2 + O\left(i^2\right)\right) + T_p i\left(T_c - T_p\right) - 2T_p}{2T_c}$$

In [11]:
```
SbTc_ser = series(Sb*Tc, i, n=2)
simplify(SbTc_ser)
```

Out[11]:
$$-T_p + \frac{T_p i\left(T_c - T_p\right)}{2} + T_c + O\left(i^2\right)$$

limit behavior of the difference with the classical definition S_a:

In [12]:
```
Sa = (Tc-Tp)/Tc
```

In [13]:
```
simplify((Sb_ser - Sa))
```

Out[13]:
$$\frac{-T_p^2 i + T_c T_p i + O\left(i^2\right)}{2T_c}$$

In [14]:
```
simplify((SbTc_ser - Sa*Tc))
```

Out[14]:
$$\frac{T_p i\left(T_c - T_p\right)}{2} + O\left(i^2\right)$$

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 [15]:
```
Sb.diff(Tp).subs({Tp:0})
```

Out[15]:
$$-\frac{\log\left(i+1\right)}{\left(i+1\right)^{T_c} - 1}$$

In [16]:
```
simplify(series(Sb.diff(Tp).subs({Tp:0}), i, n=3))
```

Out[16]:
$$-\frac{1}{T_c} + \frac{i}{2} - \frac{i^2}{4} - \frac{T_c i^2}{12} + O\left(i^3\right)$$

Economic inconsistency of the classical definition

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

In [17]:
```
Cann_p_rel = Cann_p.subs({S:Sa})/Cann_c
Cann_p_rel = simplify(Cann_p_rel)
Cann_p_rel
```

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

Inconistency for short projects

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

In [18]:
```
Cann_p_rel_Tp0 = Cann_p_rel.limit(Tp, 0)
Cann_p_rel_Tp0
```

Out[18]:
$$\frac{(i+1)^{-T_c}\left(T_c\log(i+1)+1\right)\left((i+1)^{T_c}-1\right)}{T_c\log(i+1)}$$

To analyze the effect of discount rate (which 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 explaines 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 [19]:
```
Cann_p_rel_Tp0_ser = simplify(Cann_p_rel_Tp0.series(i, 0, n=3))
Cann_p_rel_Tp0_ser
```

Out[19]:
$$1+\frac{T_c i}{2}-\frac{T_c i^2}{4}-\frac{T_c^2 i^2}{3}+O\left(i^3\right)$$

In [20]:
```
Cann_p_rel_Tp0_ser.diff(i)
```

Out[20]:
$$\frac{T_c}{2}-\frac{T_c i}{2}-\frac{2T_c^2 i}{3}+O\left(i^2\right)$$

In [21]:
```
sympy.solve(Cann_p_rel_Tp0_ser.diff(i), i)[0]
```

Out[21]:
$$\frac{3\left(T_c+O\left(i^2\right)\right)}{T_c\left(4T_c+3\right)}$$

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

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

In [22]:
```
i_worst = 1/(1+4*Tc/3)
```

In [23]:
```
simplify(Cann_p_rel_Tp0_ser.diff(i).subs({i:i_worst})) # = 0 + O(...)
```

Out[23]: $$O\left(\frac{1}{T_c^2}; T_c \to \infty\right)$$

However, the maximal value is not a constant:

In [24]:
```
simplify(Cann_p_rel_Tp0_ser.subs({i:i_worst}))
```

Out[24]: $$\frac{12 + 19T_c + O\left(\frac{1}{T_c^2}; T_c \to \infty\right)}{4 \cdot (4T_c + 3)}$$

To 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 [25]:
```
x = symbols('x', positive=True)
Cx = (1+log(x))*(x-1)/(x*log(x))
Cx
```

Out[25]: $$\frac{(x-1)\left(\log(x)+1\right)}{x\log(x)}$$

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

In [26]:
```
simplify(Cx.subs({x:(1+i)**Tc}) - Cann_p_rel_Tp0)
```

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

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

In [27]:
```
simplify(log(x**2) - 2*log(x))
```

Out[27]: 0

In [28]:
```
a = symbols('a') # without positive=True
simplify(log(a**2) - 2*log(a))
```

Out[28]: $-2\log(a) + \log\left(a^2\right)$

Now looking at the derivative:

In [29]:
```
simplify(Cx.diff(x))
```

Out[29]: $$\frac{-x + \log(x)^2 + \log(x) + 1}{x^2\log(x)^2}$$

In [30]:
```
num = Cx.diff(x)*x**2*log(x)**2
num = simplify(num)
num
```

Out[30]: $-x + \log(x)^2 + \log(x) + 1$

There is no analytical root that SymPy can find:

In [31]:
```
# sympy.solve(num, x) # NotImplementedError
```

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

In [32]:
```
opt.root_scalar(lambda x: 1 - x + log(x) + (log(x))**2, bracket = (5,10))
```

Out[32]:
```
      converged: True
           flag: 'converged'
 function_calls: 9
     iterations: 8
           root: 6.00914294108186
```

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

In [33]:
```
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[33]:
```
(1.2981372299707084, 1.298425606777053, 1.298198390999316)
```

In [34]:
```
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 [35]:
```
np.log(Cx_argmax)
```

Out[35]:
```
1.793282305296451
```

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

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

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 [36]:
```
NPCp_1rep = 1 + 1/(1+i)**Tc - S/(1+i)**Tp
NPCp_1rep
```

Out[36]: $-S(i+1)^{-T_p} + 1 + (i+1)^{-T_c}$

In [37]:
```
Cann_p_1rep = CRF(i, Tp) * NPCp_1rep
Cann_p_1rep
```

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

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 [38]:
```
Sb_1rep_incorrect = ((1+i)**(2*Tc) - (1+i)**Tp)/((1+i)**Tc - 1) * 1
Sb_1rep_incorrect
```

Out[38]: $$\frac{(i+1)^{2T_c} - (i+1)^{T_p}}{(i+1)^{T_c} - 1}$$

In [39]:
```
simplify(Cann_p_1rep.subs({S:Sb_1rep_incorrect}))  # != Cann_c
```

Out[39]: $$\frac{i(i+1)^{-T_c}\left((i+1)^{T_c}\left(-(i+1)^{2T_c} + (i+1)^{T_p}\right) + (i+1)^{T_p}\left((i+1)^{T_c} - 1\right) + (i+1)^{T_c+T_p}\left((i+1)^{T_c} - 1\right)\right)}{\left((i+1)^{T_c} - 1\right)\left((i+1)^{T_p} - 1\right)}$$

In [40]:
```
simplify(Cann_p_1rep.subs({S:Sb_1rep_incorrect}) - Cann_c) # != 0
```

Out[40]: $$\frac{i(i+1)^{-T_c}\left(-(i+1)^{2T_c} + (i+1)^{T_p}\right)}{(i+1)^{T_p} - 1}$$

In [41]:
```
Sb_1rep = ((1+i)**Tc - (1+i)**(Tp-Tc))/((1+i)**Tc - 1) * 1
Sb_1rep
```

Out[41]: $$\frac{(i+1)^{T_c} - (i+1)^{-T_c+T_p}}{(i+1)^{T_c} - 1}$$

In [42]:
```
simplify(Cann_p_1rep.subs({S:Sb_1rep}))  # == Cann_c
```

Out[42]:

$$\frac{i(i+1)^{T_c}}{(i+1)^{T_c}-1}$$

In [43]:
```
simplify(Cann_p_1rep.subs({S:Sb_1rep}) - Cann_c) # == 0
```

Out[43]: 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 [44]:
```
NPCp_2rep = 1 + 1/(1+i)**Tc + 1/(1+i)**(2*Tc) - S/(1+i)**Tp
NPCp_2rep
```

Out[44]: $-S(i+1)^{-T_p} + 1 + (i+1)^{-T_c} + (i+1)^{-2T_c}$

In [45]:
```
Cann_p_2rep = CRF(i, Tp) * NPCp_2rep
Cann_p_2rep
```

Out[45]: $$\frac{i(i+1)^{T_p}\left(-S(i+1)^{-T_p} + 1 + (i+1)^{-T_c} + (i+1)^{-2T_c}\right)}{(i+1)^{T_p}-1}$$

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 [46]:
```
Sb_2rep = ((1+i)**Tc - (1+i)**(Tp-2*Tc))/((1+i)**Tc - 1) * 1
Sb_2rep
```

Out[46]: $$\frac{(i+1)^{T_c} - (i+1)^{-2T_c+T_p}}{(i+1)^{T_c}-1}$$

In [47]:
```
simplify(Cann_p_2rep.subs({S:Sb_2rep}))  # == Cann_c
```

Out[47]: $$\frac{i(i+1)^{T_c}}{(i+1)^{T_c}-1}$$

In [48]:
```
simplify(Cann_p_2rep.subs({S:Sb_2rep}) - Cann_c) # == 0
```

Out[48]: 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 alvage at the end of the project.

NPC becomes:

In [62]:
```
T_mid, S_mid, S_fin = symbols('T_mid S_mid S_fin')
```

In [63]:
```
NPCp_1sale = 1 - S_mid/(1+i)**T_mid + 1/(1+i)**T_mid  +  - S_fin/(1+i)**Tp
NPCp_1sale
```

Out[63]: $-S_{fin}(i+1)^{-T_p} - S_{mid}(i+1)^{-T_{mid}} + 1 + (i+1)^{-T_{mid}}$

In [64]:
```
Cann_p_1sale = CRF(i, Tp) * NPCp_1sale
Cann_p_1sale
```

Out[64]: $$\frac{i(i+1)^{T_p}\left(-S_{fin}(i+1)^{-T_p} - S_{mid}(i+1)^{-T_{mid}} + 1 + (i+1)^{-T_{mid}}\right)}{(i+1)^{T_p} - 1}$$

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 [65]:
```
Sb_mid = ((1+i)**Tc - (1+i)**(T_mid))/((1+i)**Tc - 1) * 1
Sb_mid
```

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

In [66]:
```
Sb_fin = ((1+i)**Tc - (1+i)**(Tp-T_mid))/((1+i)**Tc - 1) * 1
Sb_fin
```

Out[66]: $$\frac{(i+1)^{T_c} - (i+1)^{-T_{mid}+T_p}}{(i+1)^{T_c} - 1}$$

In [67]:
```
simplify(Cann_p_1sale.subs({S_fin:Sb_fin, S_mid:Sb_mid}))  # == Cann_c
```

Out[67]: $$\frac{i(i+1)^{T_c}}{(i+1)^{T_c} - 1}$$

In [68]:
```
simplify(Cann_p_1sale.subs({S_fin:Sb_fin, S_mid:Sb_mid}) - Cann_c) # == 0
```

Out[68]: 0

Plot functions

In [56]:
```
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 [57]:

```
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', linestyles='--', 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 [58]:

```
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', linestyles='--', 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 ($T_p \ll T_c$
        xlabel='Discount rate $i$',
        ylabel='Annualized project cost\n relative to annualized component cost'
    )
    fig.tight_layout()
    return fig, ax
```

In [79]:

```
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-term sale
    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', linestyles='--', 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} y'+
              '\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
```