Betrachtung eines Butterworth-TP --- 6.Mai 2007 Ingenieurbüro Baumann, Dorsten

```
reset():ta:=time():DIGITS:=32:w:=2*PI*f:
```

die Eingangsdaten

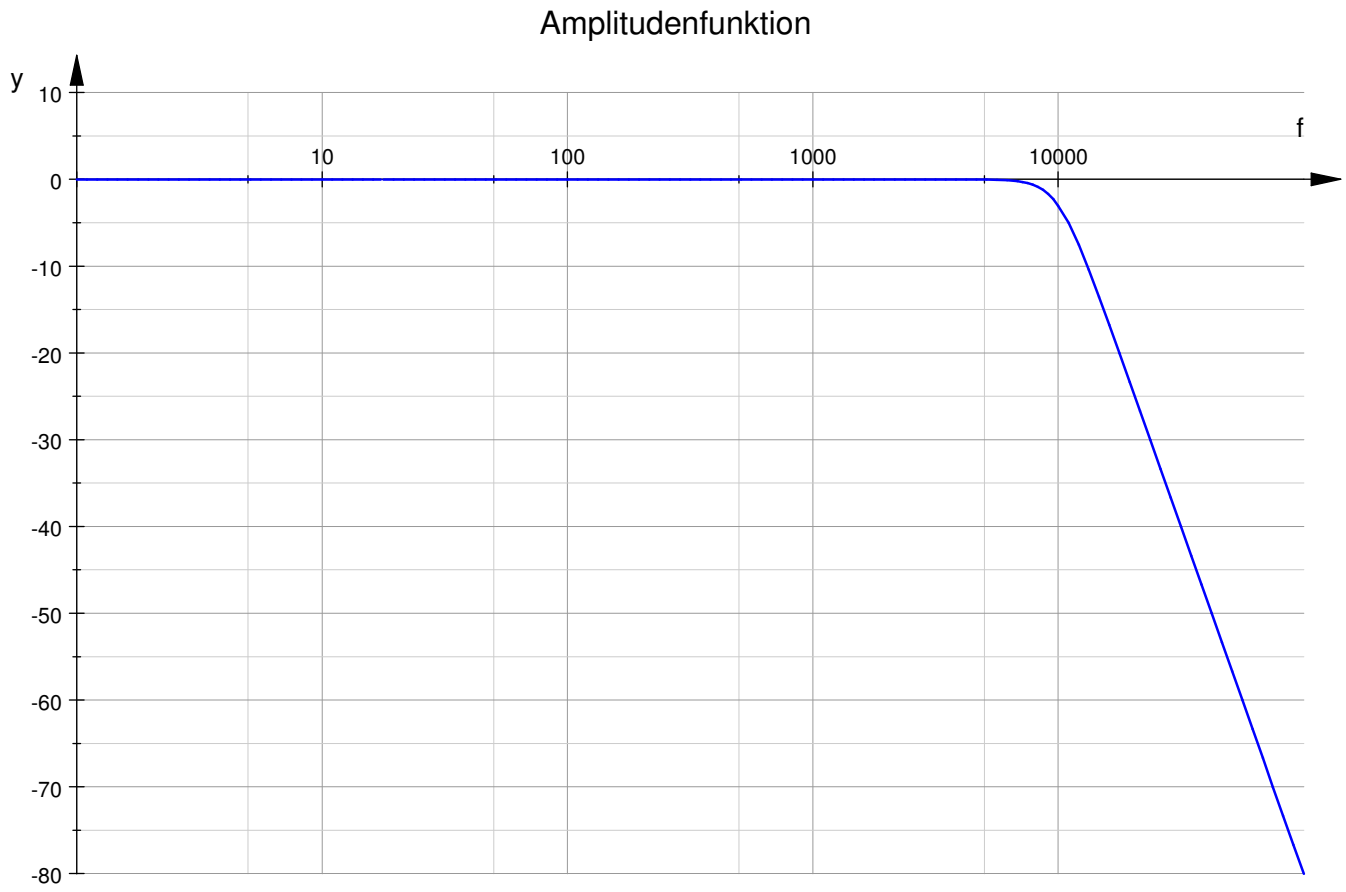
```
n:=4:fg:=10e3:ue2:=1:
```

die Berechnungen

```
wg:=2*PI*fg:
wiwg:=[cos((2*(i-1)+1)*PI/2/n) $ i=1..n]:
sigwg :=[-sin((2*(i-1)+1)*PI/2/n) $ i=1..n]:
prod:=(f)->product(-sigwg[i]+I*(w/wg-wiwg[i]), i=1..n):
U2U0:=(f)->1/(1+1/ue2)/prod(f):
U2U0dB:=(f)->20*log(10,abs(U2U0(f))):
Winkel:=(f)->-180/PI*sum(arctan((w/wg-wiwg[i])/-sigwg[i]), i=1..n):
Winkel:=(f)->180/PI*arg(U2U0(f)):
tg:=(f)->-1/wg*sum(sigwg[i]/(sigwg[i]^2+(w/wg-wiwg[i])^2), i=1..n):
tg1:=(f)->-diff(Winkel(f),f)/360:
```

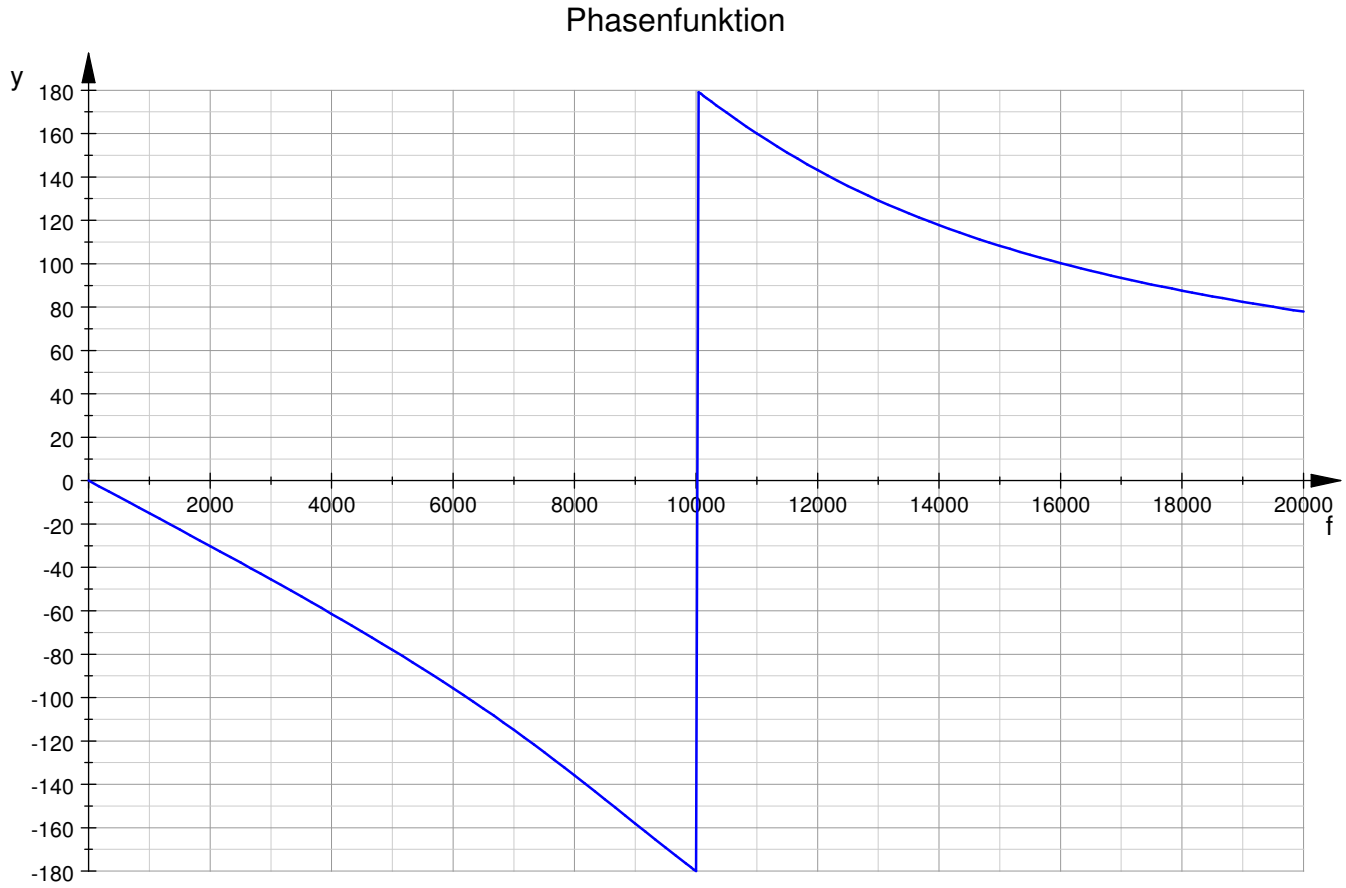
Betrag der Übertragungsfunktion, doppelt logarithmisch

```
plotfunc2d(U2U0dB(f)+6.01, f=1..10*fg, LegendVisible=FALSE,
CoordinateType=LogLin,
            GridVisible=TRUE, SubgridVisible=TRUE,
  Height=120*unit::mm, Width=180*unit::mm,
  Header="Amplitudenfunktion", YMax=10):
```



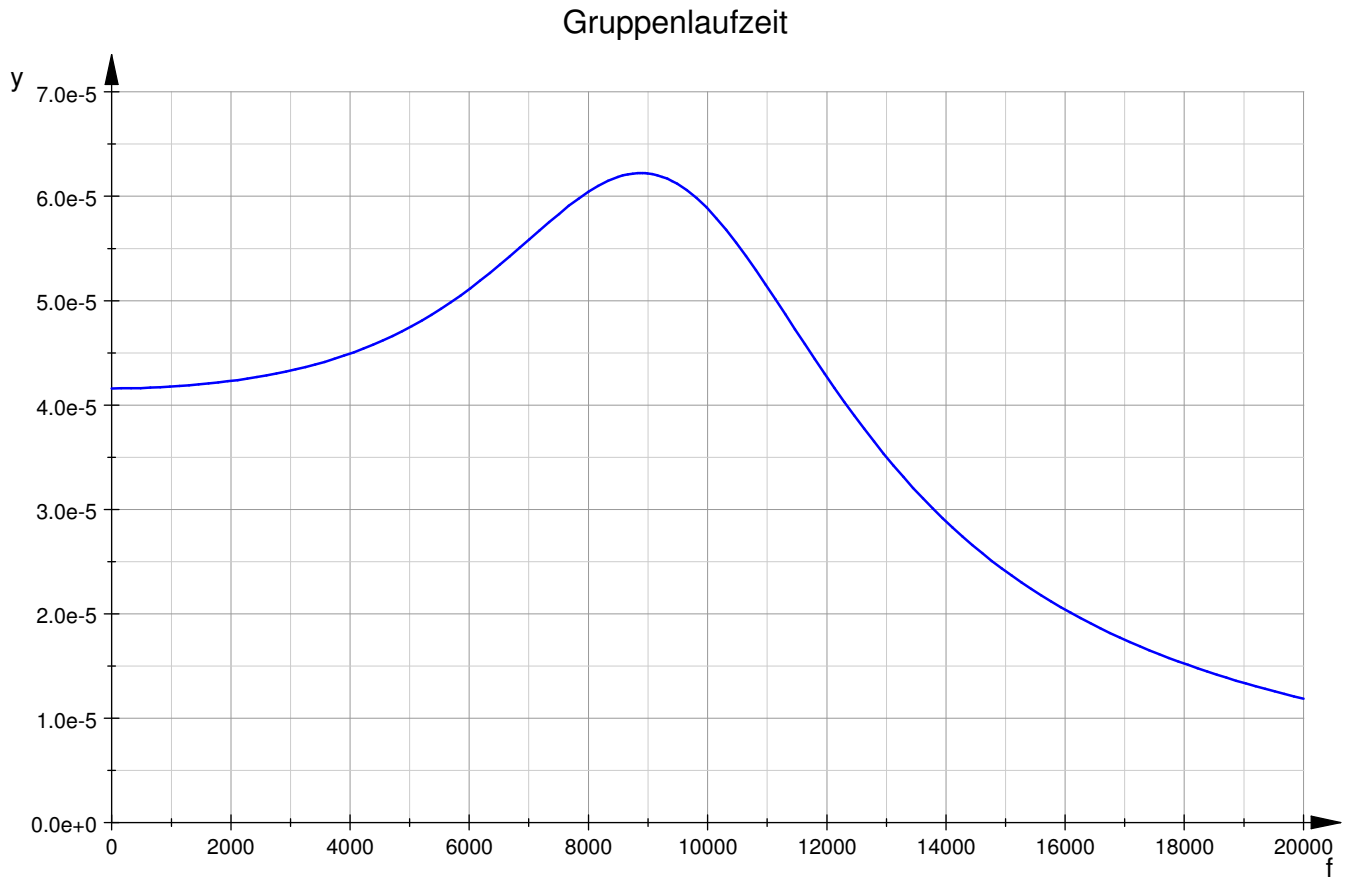
die Phasenverschiebung des Filters

```
plotfunc2d(Winkel(f), f=0..2*fg, LegendVisible=FALSE,
           GridVisible=TRUE, SubgridVisible=TRUE,
  Height=120*unit::mm, Width=180*unit::mm, Header="Phasenfunktion"):
```



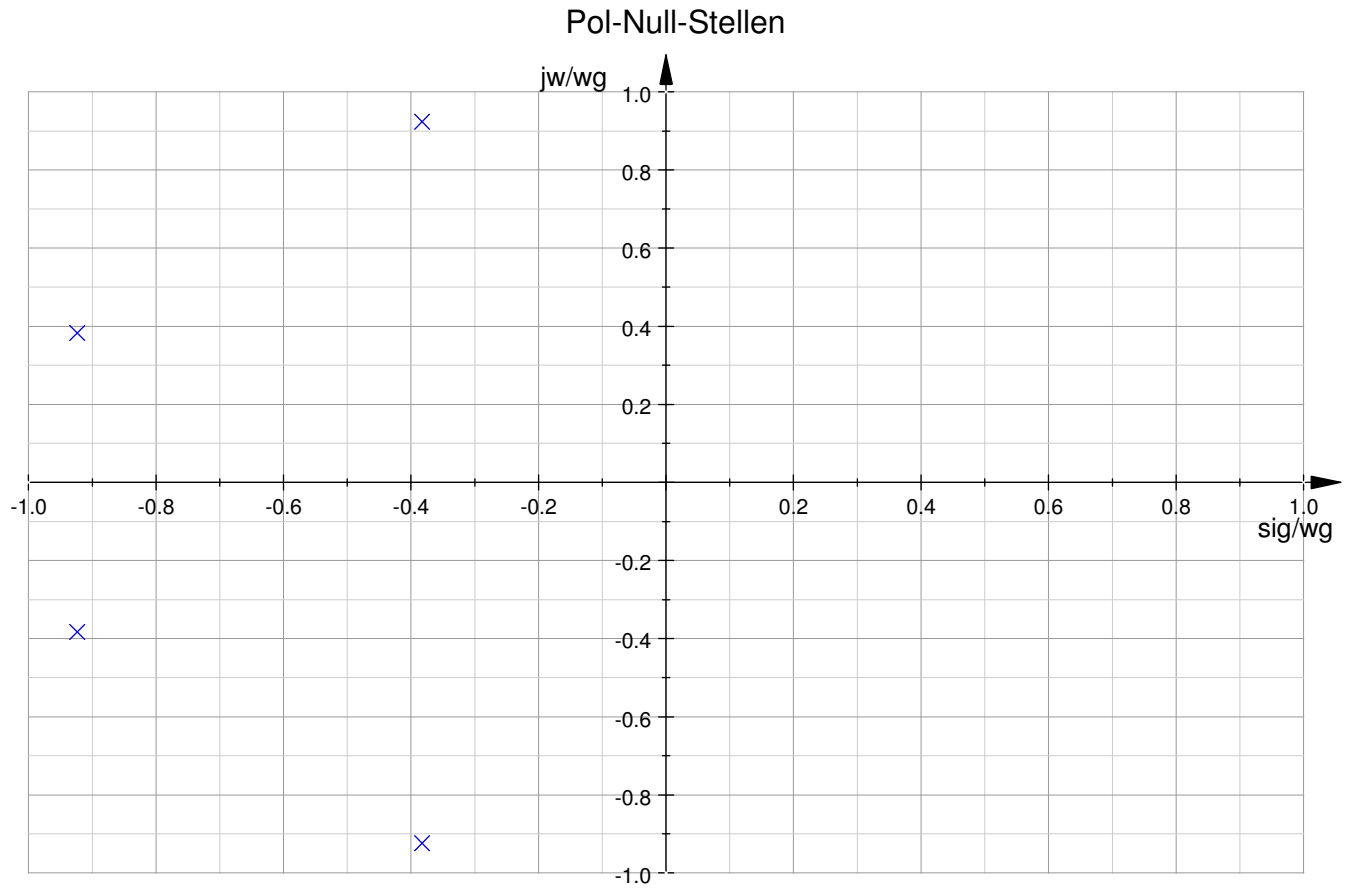
Berechnete Gruppenlaufzeit aus Formel tg(f)

```
plotfunc2d(tg1(f), f=0..2*fg, LegendVisible=FALSE,
           GridVisible=TRUE, SubgridVisible=TRUE,
  Height=120*unit::mm, Width=180*unit::mm,Header="Gruppenlaufzeit",
  YRange=0..7e-5):
```



Lage der Pol-Nullstellen in der komplexen Ebene

- `delete Liste:Liste:=[[sigwg[i],wiwg[i],RGB::Blue] $ i=1..n]:`
- `Liste:=Liste.[[1,0,RGB::White]].[[0,1,RGB::White]].[[-1,0,RGB::White]].[[0,-1,RGB::White]]:`
- `plot(plot::PointList2d(Liste, PointStyle=XCrosses, PointSize=2, Color=RGB::Blue, GridVisible=TRUE, SubgridVisible=TRUE, Scaling=Unconstrained, AxesTitles=["sig/wg","jw/wg"]), Height=120*unit::mm, Width=180*unit::mm, Header="Pol-Null-Stellen"):`



Pol-Nullstellen des Filters

```
for i from 1 to n do
     PolTab[i]:=float(sigwg[i]+I*wiwg[i])
end_for:
```

```
PolTab;
```

$$\begin{bmatrix} 1 & = -0.382683432365089771728459984030 4 + 0.923879532511286756128183189396 79 \cdot i \\ 2 & = -0.923879532511286756128183189396 79 + 0.382683432365089771728459984030 4 \cdot i \\ 3 & = -0.923879532511286756128183189396 79 - 0.382683432365089771728459984030 4 \cdot i \\ 4 & = -0.382683432365089771728459984030 4 - 0.923879532511286756128183189396 79 \cdot i \end{bmatrix}$$

die Pol-Nullstellen durch Suche aus dem Polynom (auch hier:Im u. Re vertauscht und Faktor -1)

```
delete i:Pol:=solve(prod(f)=0,f)/fg:
```

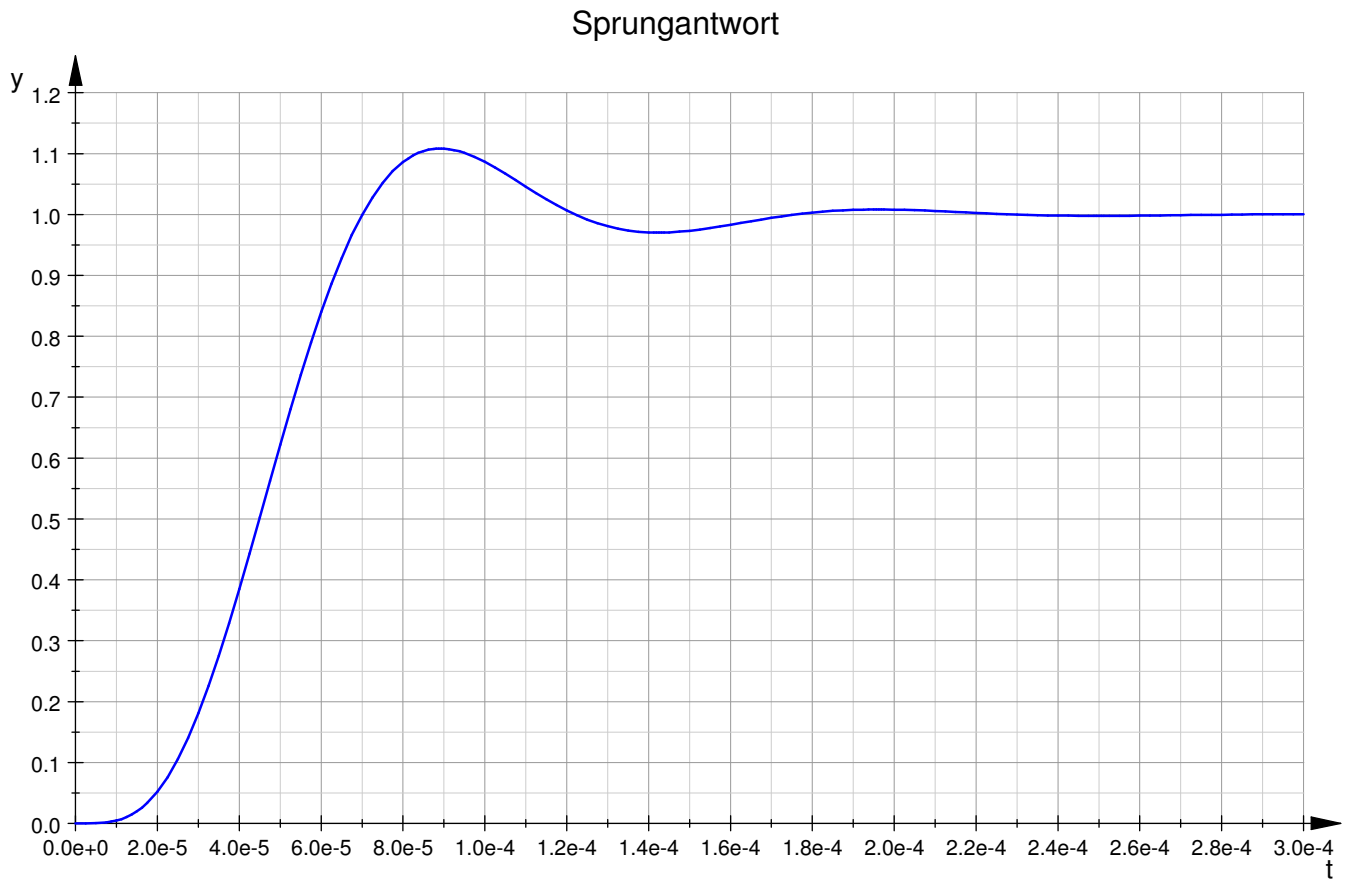
```
for i from 1 to n do
     PolTabN[i]:=-Im(op(Pol,i))-I*Re(op(Pol,i)):
end_for:
```

```
PolTabN;
```

$$\begin{bmatrix} 1 & = -0.923879532511286756128183189396 79 + 0.382683432365089771728459984030 4 \cdot i \\ 2 & = -0.923879532511286756128183189396 79 - 0.382683432365089771728459984030 4 \cdot i \\ 3 & = -0.382683432365089771728459984030 4 + 0.923879532511286756128183189396 79 \cdot i \\ 4 & = -0.382683432365089771728459984030 4 - 0.923879532511286756128183189396 79 \cdot i \end{bmatrix}$$

Sprungantwort des Filters ua(t)=invlaplace(2/p*T(p))

```
delete i:prodp:=(p)->product(p/wg-sigwg[i]-I*wiwg[i], i=1..n):
ua:=(t)->Re(transform::invlaplace(1/(1+1/ue2)*2/p/prodp(p),p,t)):
plotfunc2d(ua(t), t=0..3/fg, LegendVisible=FALSE,
CoordinateType=LinLin,
          GridVisible=TRUE, SubgridVisible=TRUE,
  Height=120*unit::mm, Width=180*unit::mm, Header="Sprungantwort",
  YMax=1.2):
```



Suchbereich an der Flanke definieren

```
anf:=0:ende:=2/fg:
```

Überschwingen in % bei t in us

```
maximum:=op(numeric::solve(diff(ua(t),t)=0,t=anf..ende,RestrictedSea
rch),1):
(ua(maximum)-1)*100;maximum/1e-6;
```

10.830150888537364280420386368286

89.091495433129426852445810912783

t0 für ua(t)=1/2 in us

- `tx:=op(numeric::solve(Re(ua(t))=1/2,t=anf..maximum, RestrictedSearch),1):tx/1e-6;`

  44.885890189741794003857925638927

die Einschwingzeit tau in us und die daraus resultierende Grenzfrequenz in kHz

- `m:=ua'(t):t:=tx:m:=float(m):delete t:yt:=t->1/2-m*(tx-t):`
- `tau:=op(solve(yt(t)=1,t),1)-op(solve(yt(t)=0,t),1):tau/1e-6;1/2/tau/1e3;`

  41.776687025127236519570794894073

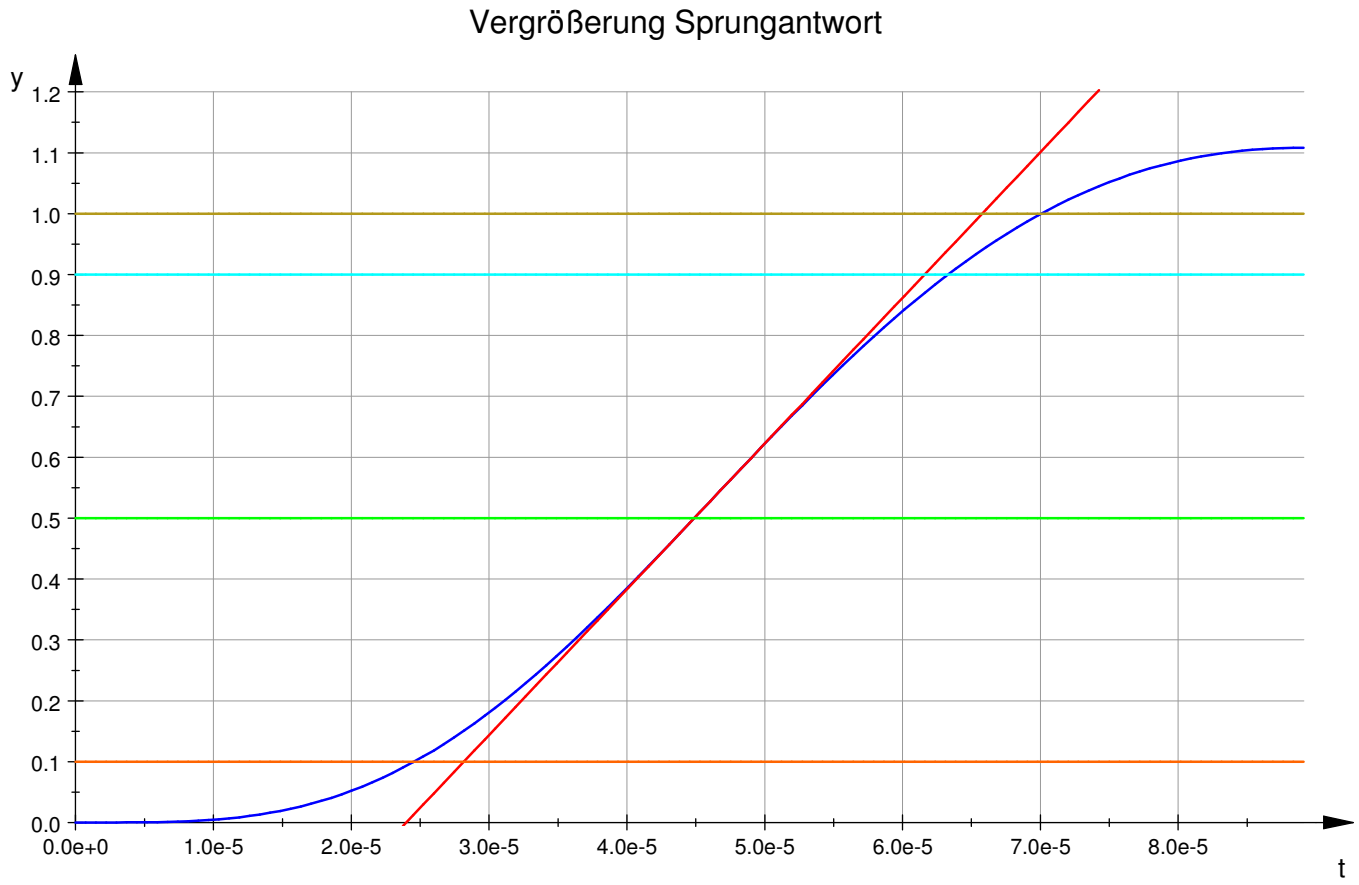
  11.968397582586365529226696635187

tr, Rise-Time in us

- `tr:=op(numeric::solve(ua(t)=9/10,t=anf..ende,RestrictedSearch),1)-op(numeric::solve(yt(t)=1/10,t=anf..ende,RestrictedSearch),1):tr/1e-6;`

  35.117834672408776683575152053648

- ```
  plotfunc2d(ua(t), yt(t), 1/2, 1, 1/10, 9/10, t=0..maximum,
  LegendVisible=FALSE, CoordinateType=LinLin,
              GridVisible=TRUE, SubgridVisible=FALSE,
    Height=120*unit::mm, Width=180*unit::mm, Header="Vergrößerung
    Sprungantwort", YRange=0..1.2):
  ```
  ```



CPU-Zeit in Sekunden und in Minuten

- `te:=time():float((te-ta)/1000);float((te-ta)/1000/60);`

  5.797

  0.096616666666666666666666666666667

-