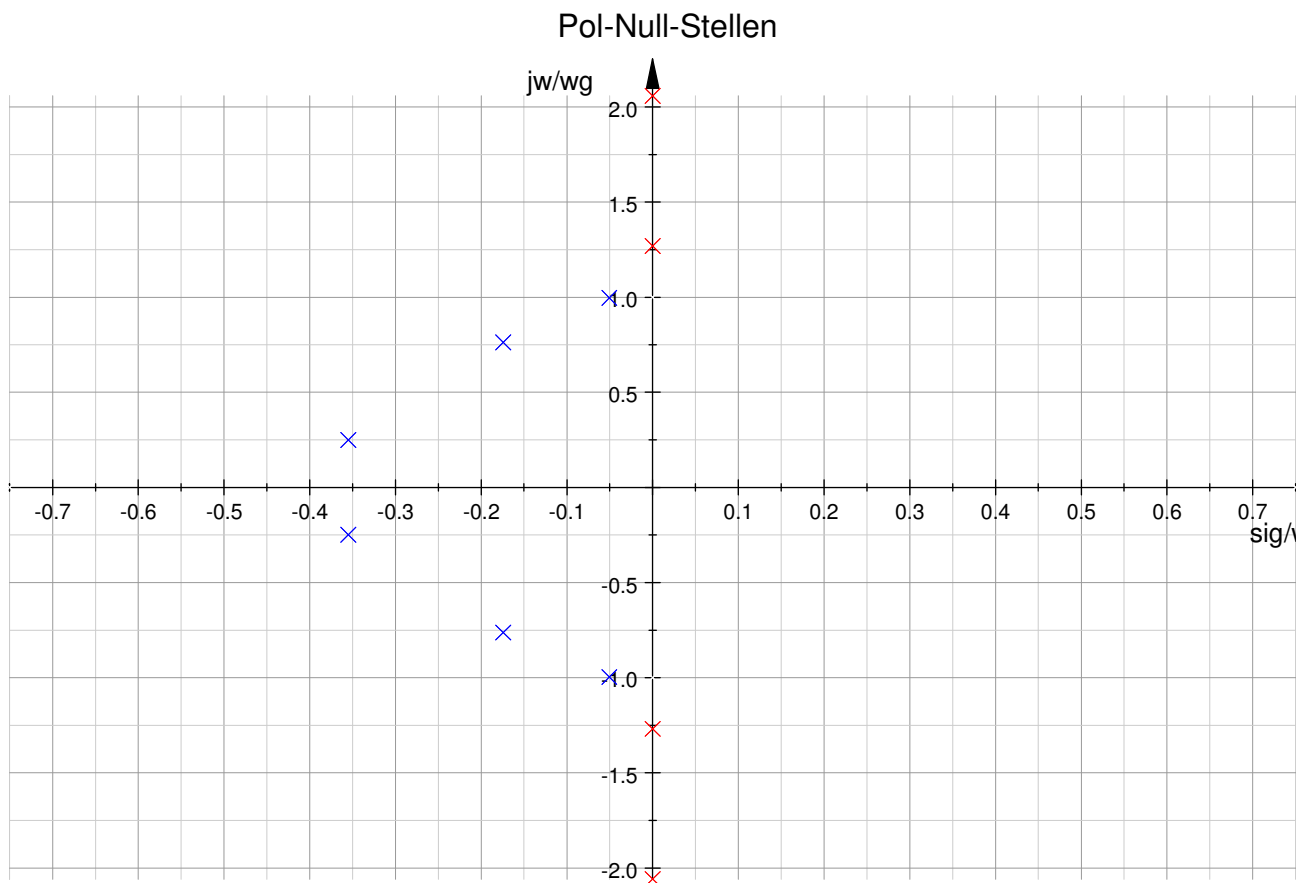


Betrachtung eines Cauer-TP aus Tabellendaten --- 6.Mai 2007 Ingenieurbüro Baumann, Dorsten  
es soll sich um einen CAUER-TP (mathematisch.) 6. Grades handeln mit  $ah=1$  dB und  $as=65$  dB

- `reset():ta:=time():DIGITS:=8:w:=2*PI*f:`
- `nZ:=4:nN:=6:fg:=10e3:ue2:=1:wg:=2*PI*fg:`
- `sigwgN=[-0.35495,-0.35495,-0.17433,-0.17433,-0.05047,-0.05047]:wiwgN=[0.24874,-0.24874,0.76244,-0.76244,0.99619,-0.99619]:sigwgZ=[0,0,0,0]:wiwgZ=[-2.05830,2.05830,-1.26883,1.26883]:`

Lage der Pol-Null-Stellen, rot -> Nullstellen, blau -> Polstellen

- `ListeN:=[]:for i from 1 to nN do  
ListeN:=ListeN.[[op(sigwgN,i),op(wiwgN,i),RGB::Blue]]:  
end_for:  
ListeZ:=[]:for i from 1 to nZ do  
ListeZ:=ListeZ.[[op(sigwgZ,i),op(wiwgZ,i),RGB::Red]]:  
end_for:`
- `Breite:=3/4:Liste:=ListeZ.ListeN.[[Breite,0,RGB::White]].[[0,1,RGB::White]].[[-Breite,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")):`

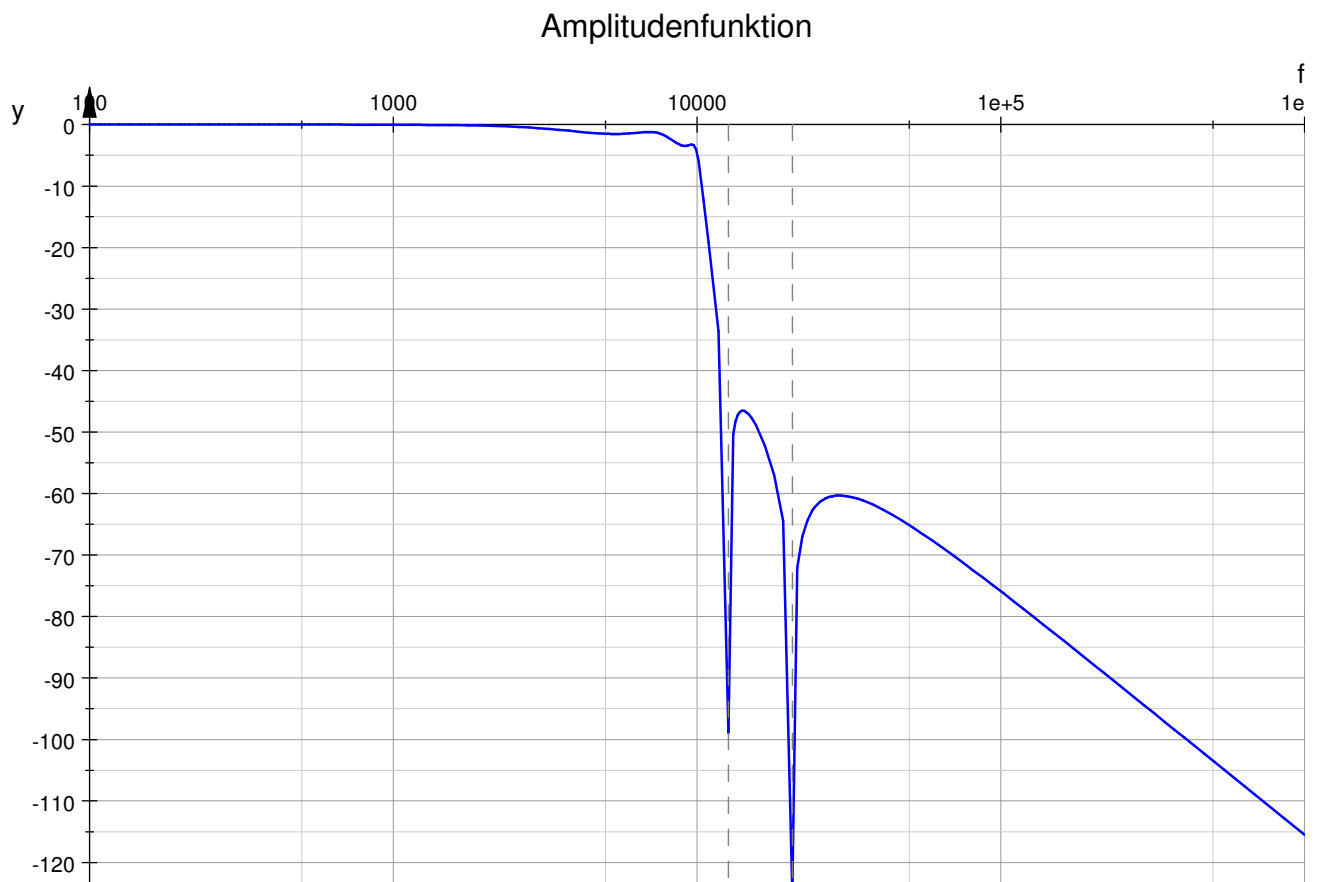


- `delete i:prodN:=(f)->product(I*w/wg-sigwgN[i]-I*wiwgN[i],`

- ```
i=1..nN):prodZ:=(f)->product(I*w/wg-sigwgZ[i]-I*wiwgZ[i], i=1..nZ):
```
- `f:=0:a0:=float(Re(prodN(f)/prodZ(f))):`
  - `U2U0:=(f)->a0/(1+1/ue2)*prodZ(f)/prodN(f):`
  - `U2U0dB:=(f)->20*log(10,abs(U2U0(f))):`
  - `Winkel:=(f)->180/PI*arg(U2U0(f)):`
  - `tg:=(f)->-diff(Winkel(f),f)/360:`

#### Betrag der Übertragungsfunktion doppelt Logarithmisch

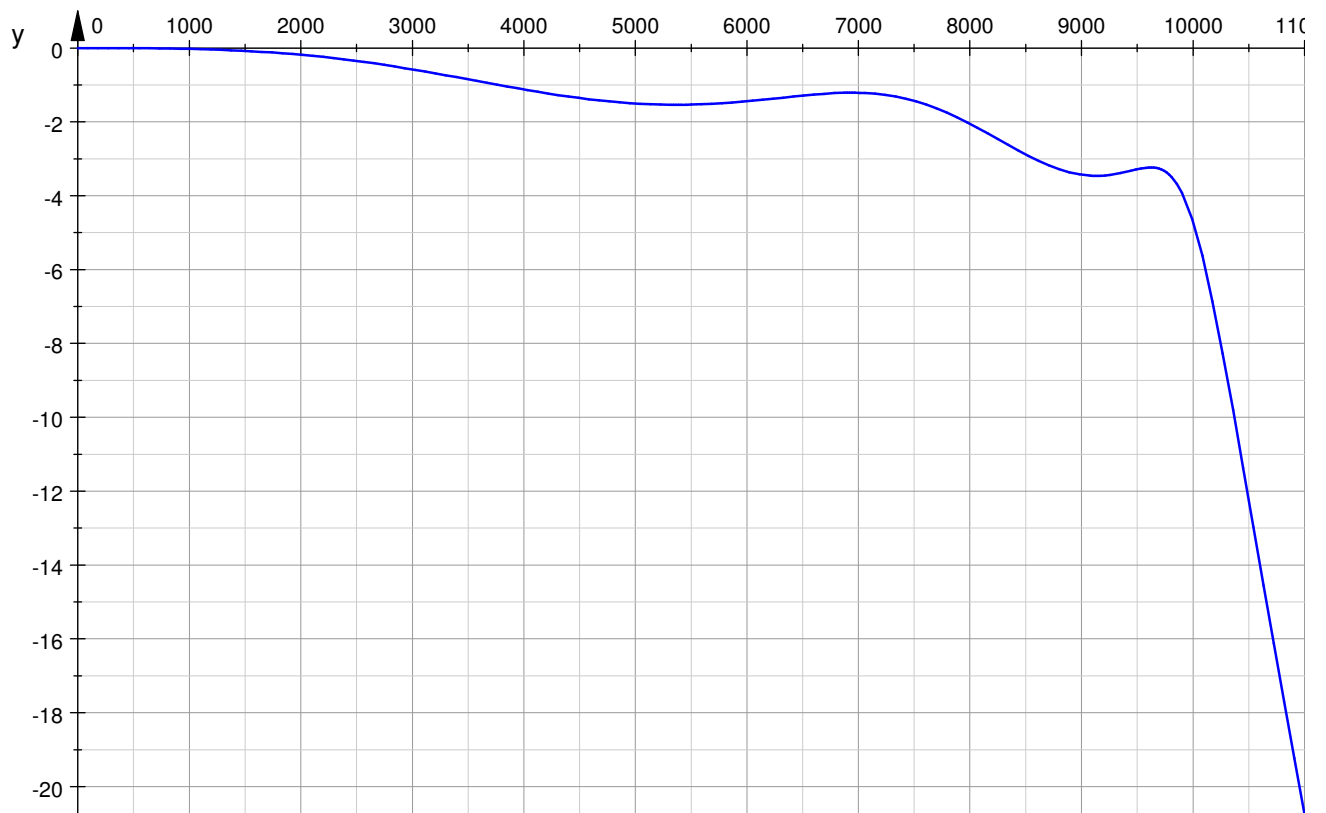
- `delete f:plotfunc2d(U2U0dB(f)+6.02, f=100..100*fg, LegendVisible=FALSE, CoordinateType=LogLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Amplitudenfunktion"):`



#### Ausschnittsvergrößerung aus dem Betrag der Übertragungsfunktion, einfach logarithmisch

- `plotfunc2d(U2U0dB(f)+6.02, f=0..1.1*fg, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Vergrößerung Amplitudenfunktion"):`

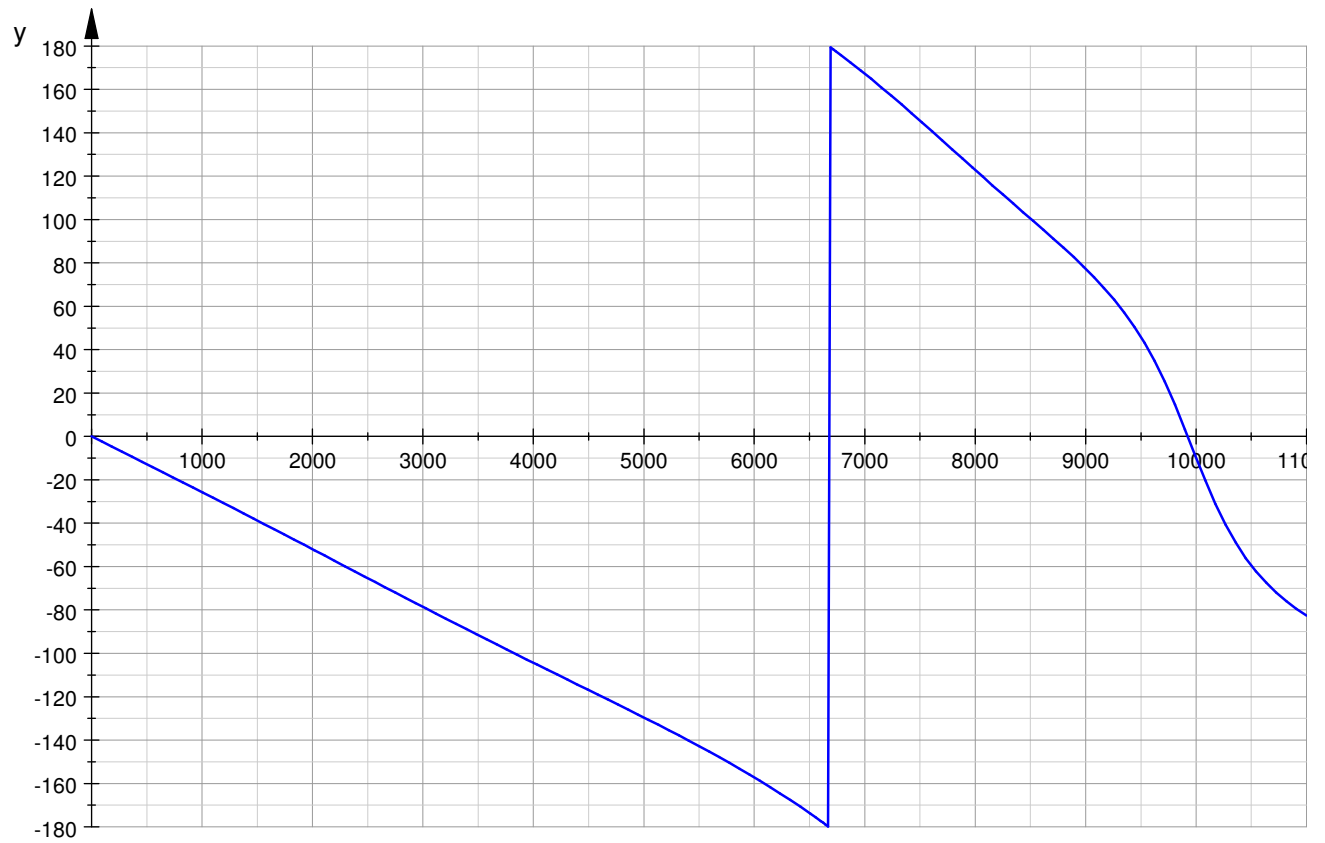
## Vergößerung Amplitudenfunktion



### die Phasenverschiebung des Filters

- `plotfunc2d(Winkel(f), f=0..1.1*fg, LegendVisible=FALSE, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Phasenfunktion"):`

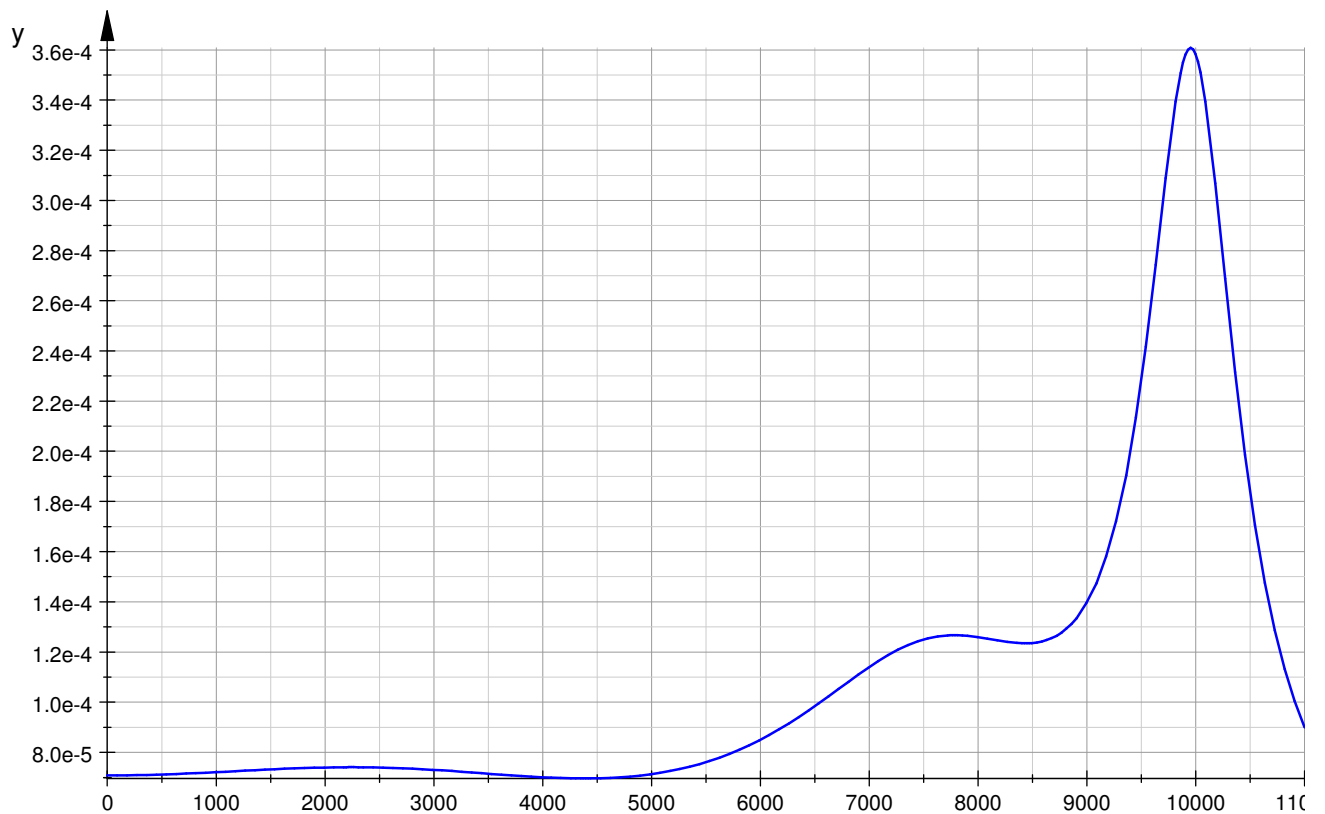
## Phasenfunktion



### Berechnete Gruppenlaufzeit

- `plotfunc2d(tg(f), f=0..1.1*fg, LegendVisible=FALSE, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Gruppenlaufzeit"):`

## Gruppenlaufzeit



die Grunddämpfung durch  $\ddot{u}^2$

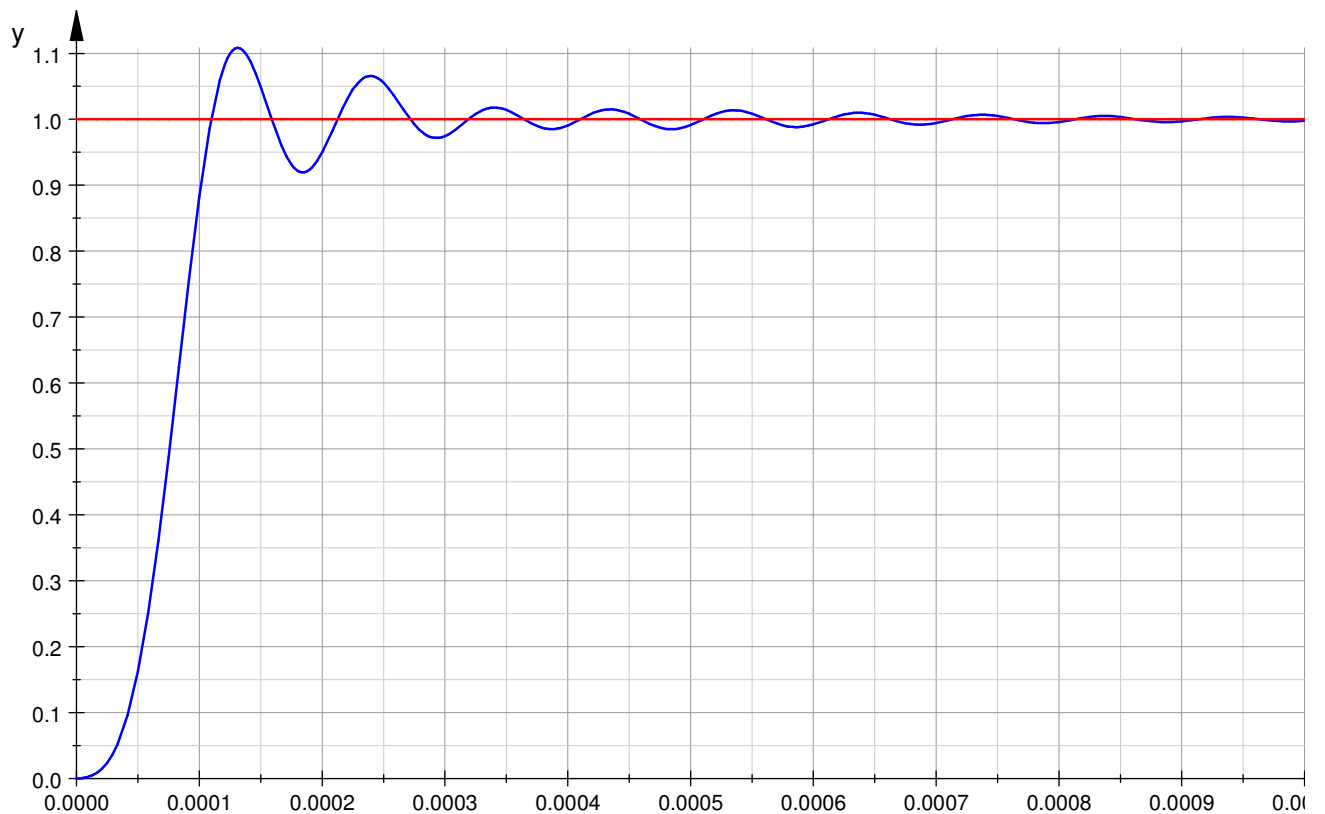
- $1/(1+1/\ddot{u}^2)$ ;

$$\frac{1}{2}$$

Sprungantwort des Filters  $u_a(t)=\text{invlaplace}(2/p*T(p))$

- `delete i:prodpZ:=(f)->product(p/wg-sigwgZ[i]-I*wiwgZ[i], i=1..nZ):prodpN:=(f)->product(p/wg-sigwgN[i]-I*wiwgN[i], i=1..nN):`
- `ua:=(t)->Re(transform::invlaplace(a0/(1+ue2)*2/p*prodpZ(p)/prodpN(p),p,t)):`
- `plotfunc2d(ua(t), 1, t=0..10/fg, LegendVisible=FALSE, CoordinateType=LinLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Sprungantwort"):`

## Sprungantwort



### Ausschnittsvergrößerung der Sprungantwort

Suchbereich definieren

- `anf:=0.5e-5:ende:=1.5e-4:`

Überschwingen in % bei t in us

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

10.826286

131.17211

to für  $ua(t)=1/2$  in us

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

75.792688

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;
```

```
62.405794
```

```
8.0120766
```

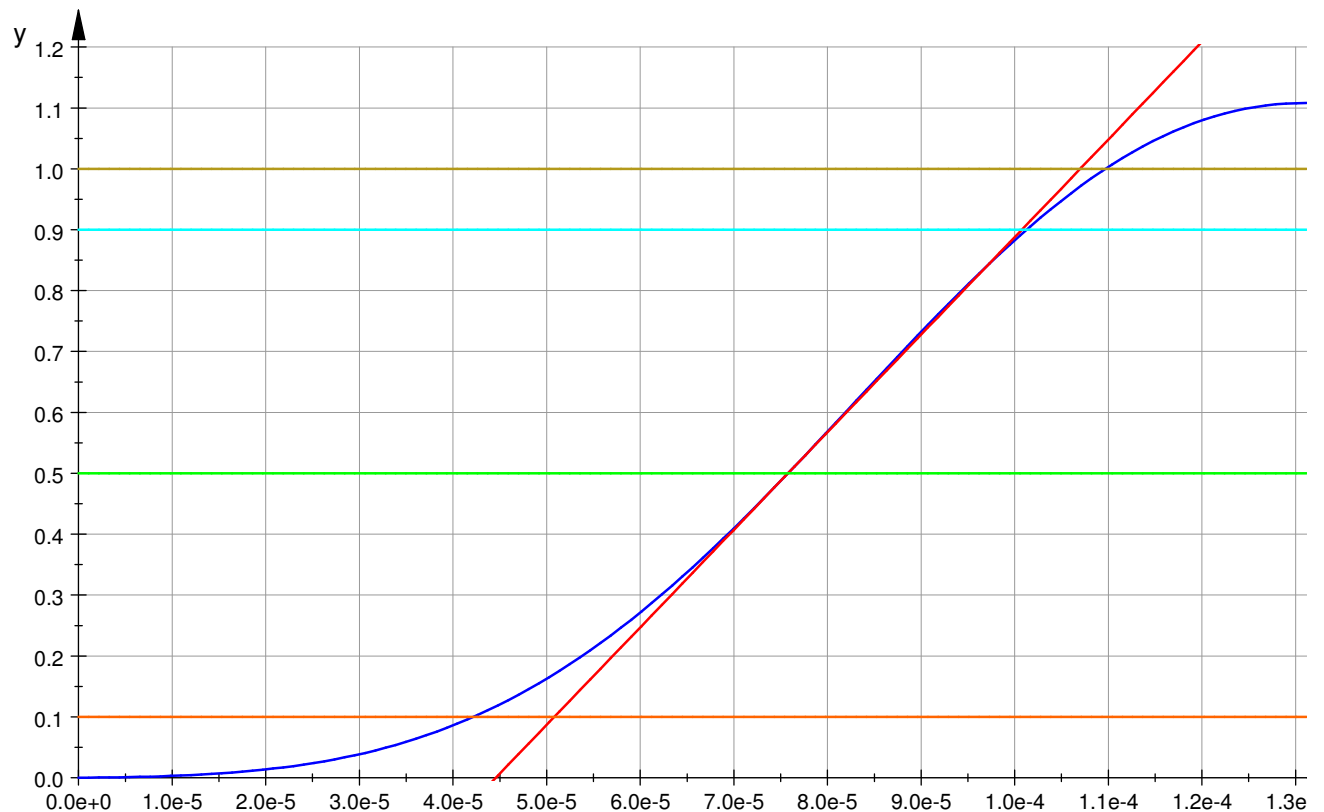
tr, Rise-Time in us

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

```
59.041904
```

- ```
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):
```

Vergrößerung Sprungantwort

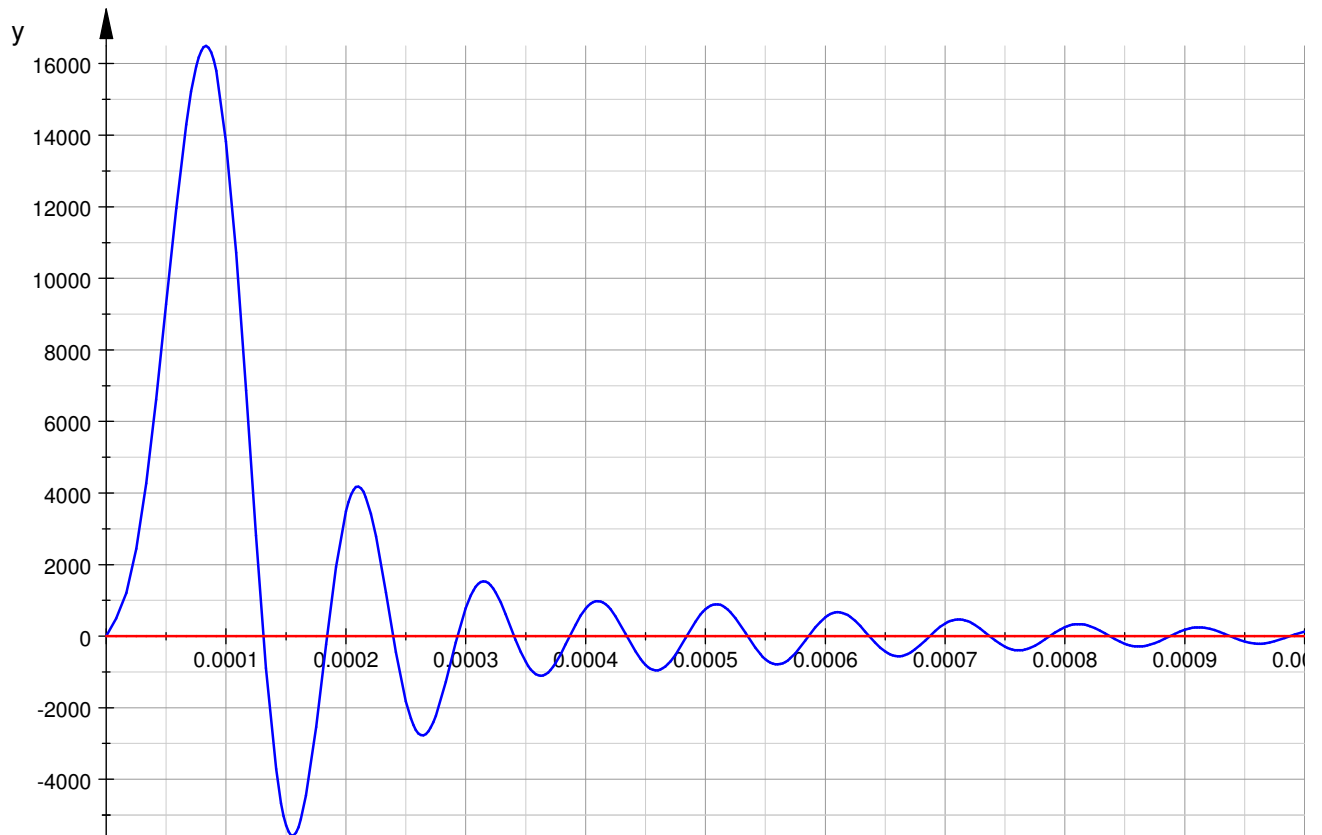


Impulsantwort des Filters  $ua(t)=\text{invlaplace}(T(p))$

- ```
ua:=(t)-  
>Re(transform::invlaplace(a0/(1+ue2)*2*prodpZ(p)/prodpN(p),p,t)):
```
- ```
plotfunc2d(ua(t), 1, t=0..10/fg, LegendVisible=FALSE,
```

```
CoordinateType=LinLin,  
    GridVisible=TRUE, SubgridVisible=TRUE,  
    Height=120*unit::mm, Width=180*unit::mm, Header="Impulsantwort"):
```

### Impulsantwort



CPU-Zeit in Sekunden und in Minuten

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

9.109

0.15181667

•