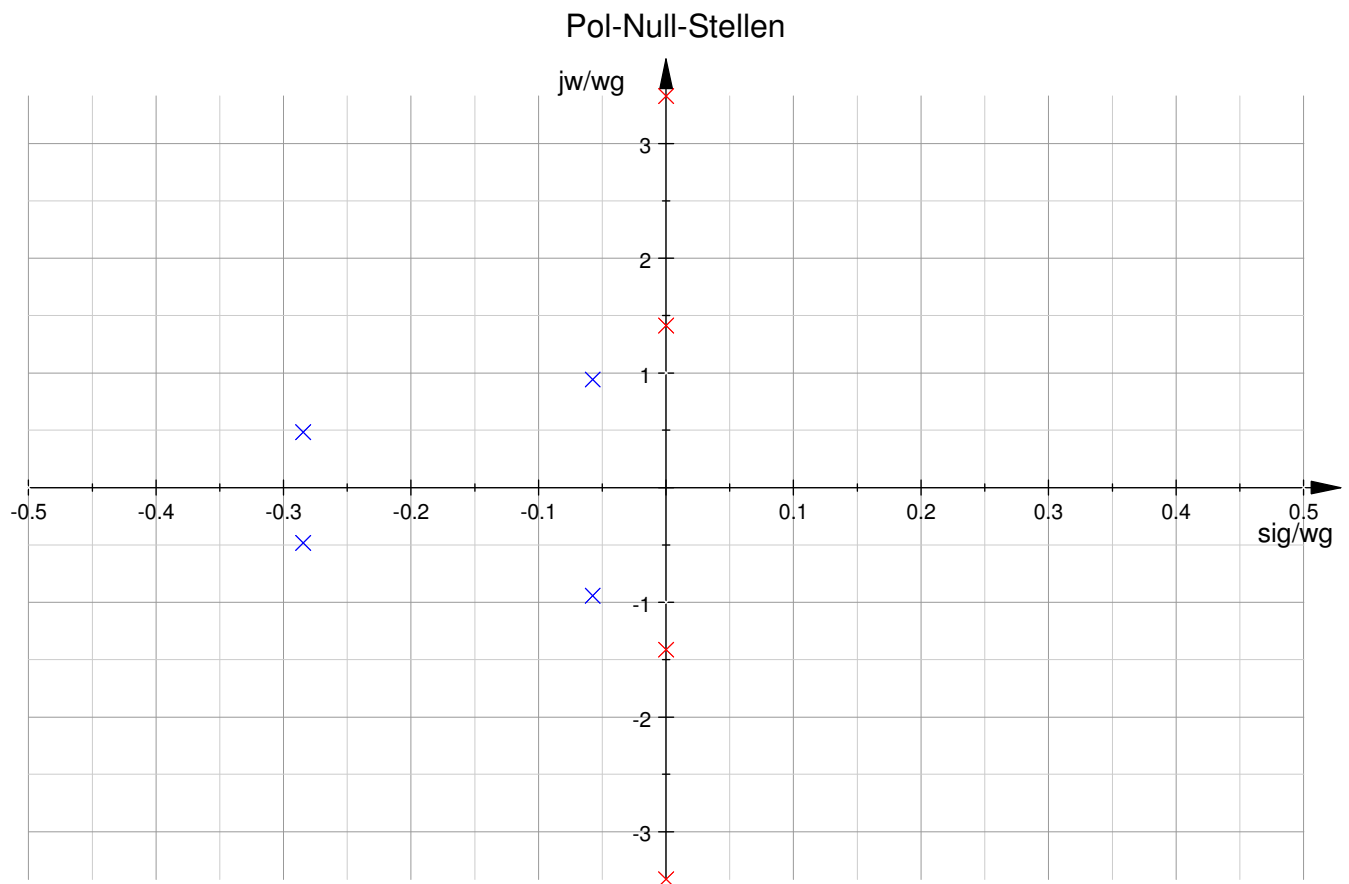


Analyse des mit o.g. Verfahren bestimmten Cauer-TP --- 20.Mai 2007 Ingenieurbüro Baumann, Dorsten

- `reset():ta:=time():DIGITS:=8:w:=2*PI*f:`
- `nZ:=4:nN:=4:fg:=10e3:ue2:=1:wg:=2*PI*fg:`
- `sigwgN:=[-0.05755,-0.05755,-0.28462,-0.28462]:`
- `wiwgN:=[0.94304,-0.94304,0.48267,-0.48267]:`
- `sigwgZ:=[0,0,0,0]:`
- `wiwgZ:=[-1.41241,1.41241,-3.41421,3.41421]:`

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

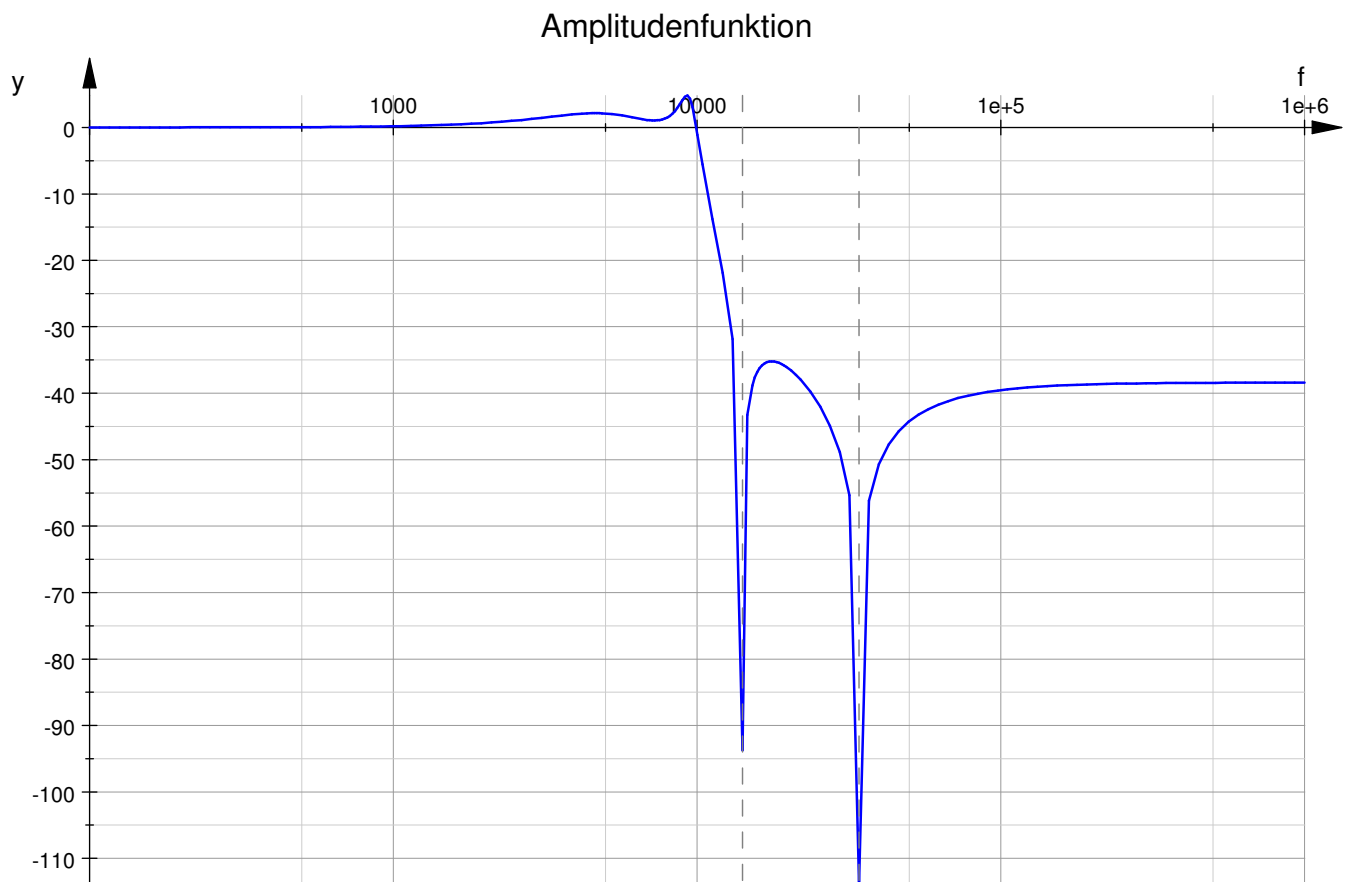
- `ListeN:=[]:for i from 1 to nN do`
`ListeN:=[op(sigwgN,i),op(wiwgN,i),RGB::Blue]:`
`end_for:`
- `ListeZ:=[]:for i from 1 to nZ do`
`ListeZ:=[op(sigwgZ,i),op(wiwgZ,i),RGB::Red]:`
`end_for:`
- `Breite:=1/2:Liste:=ListeZ.ListeN.[[Breite,0,RGB::White]].[[0,1,RGB::White]].[[-Breite,0,RGB::White]].[[0,-1,RGB::White]]:`
- `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

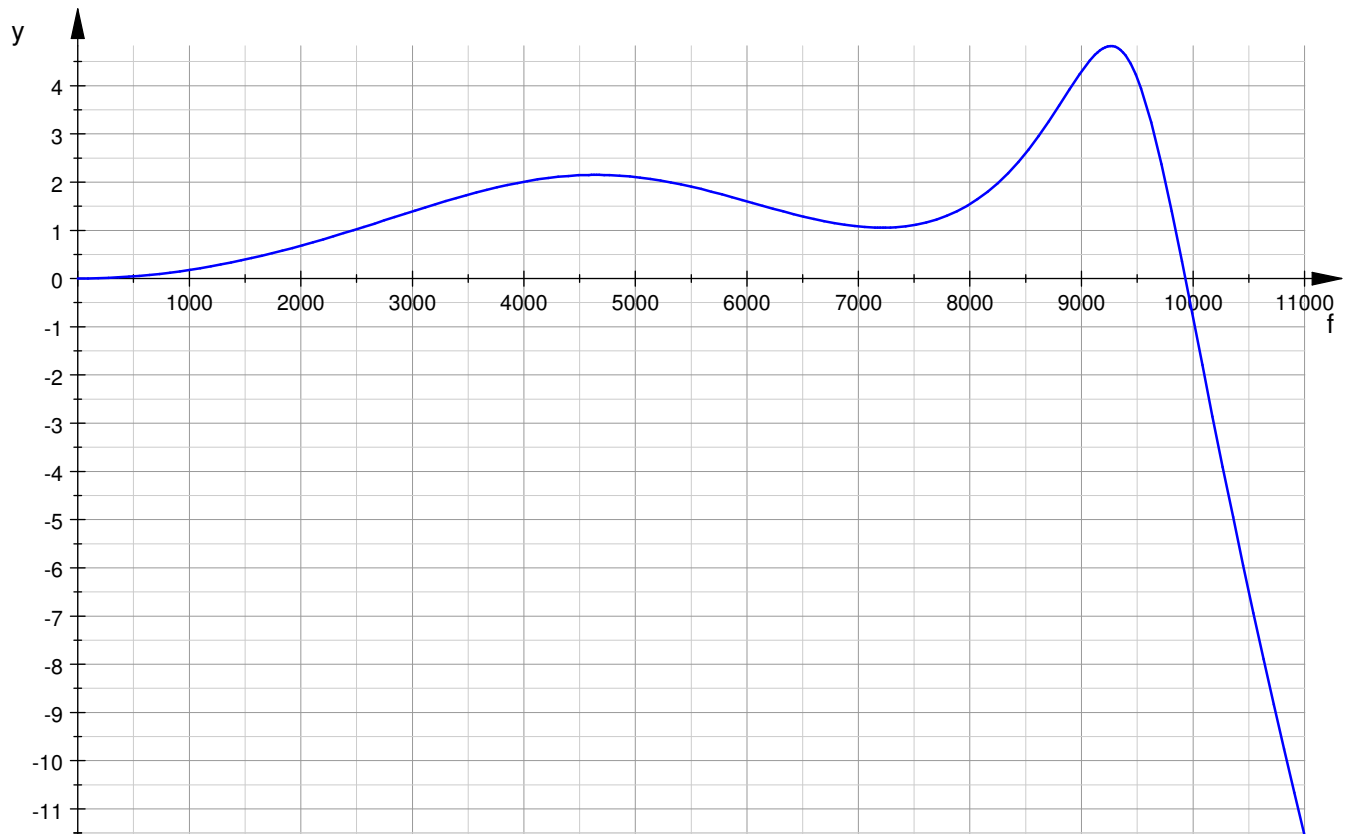
- `delete f:plotfunc2d(U2U0dB(f)+6.02, f=100..100*fg, LegendVisible=FALSE, CoordinateType=LogLin, GridVisible=TRUE, SubgridVisible=TRUE, Height=120*unit::mm, Width=180*unit::mm, Header="Amplitudenfunktion"):`



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"):`

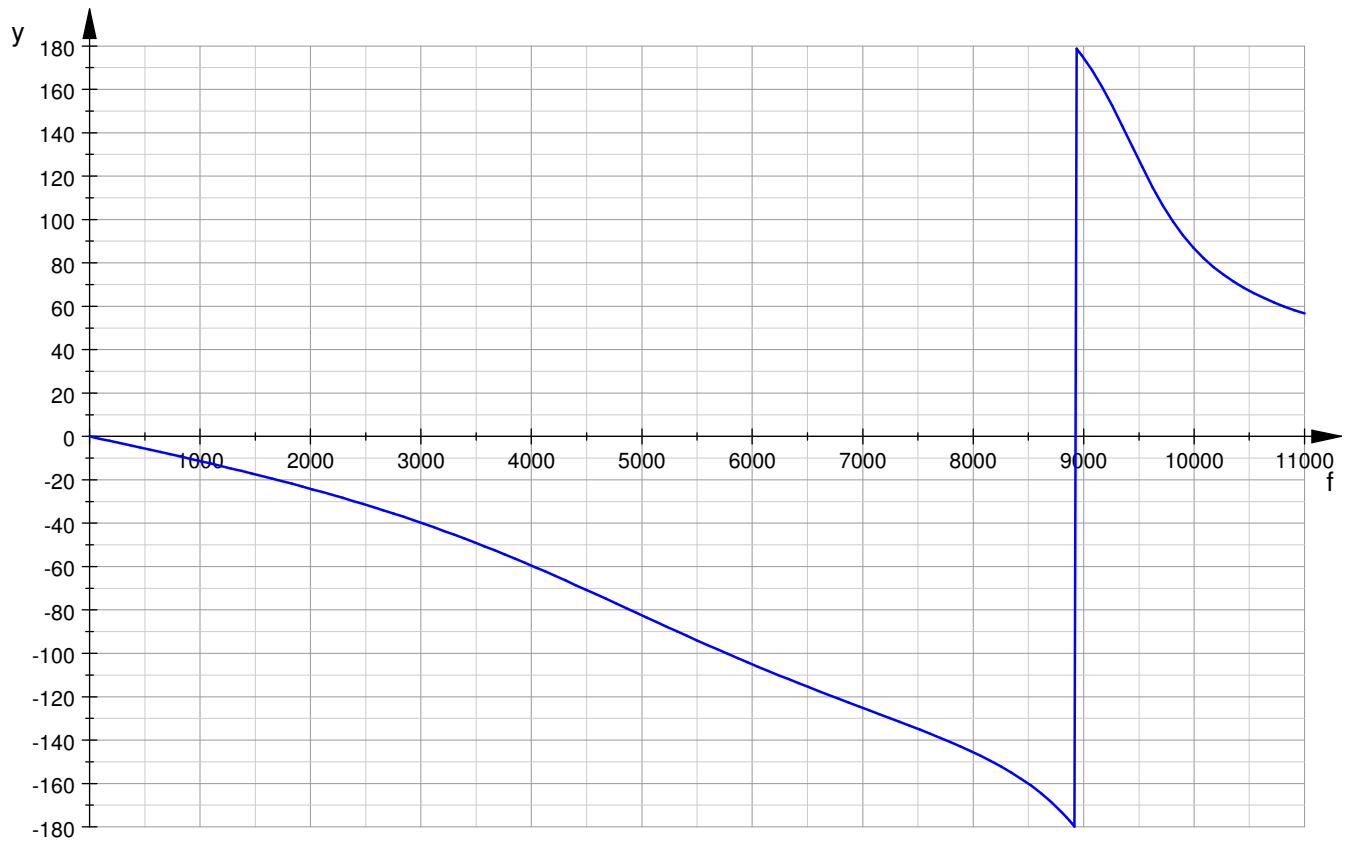
Vergrößerung Amplitudenfunktion



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"):
```

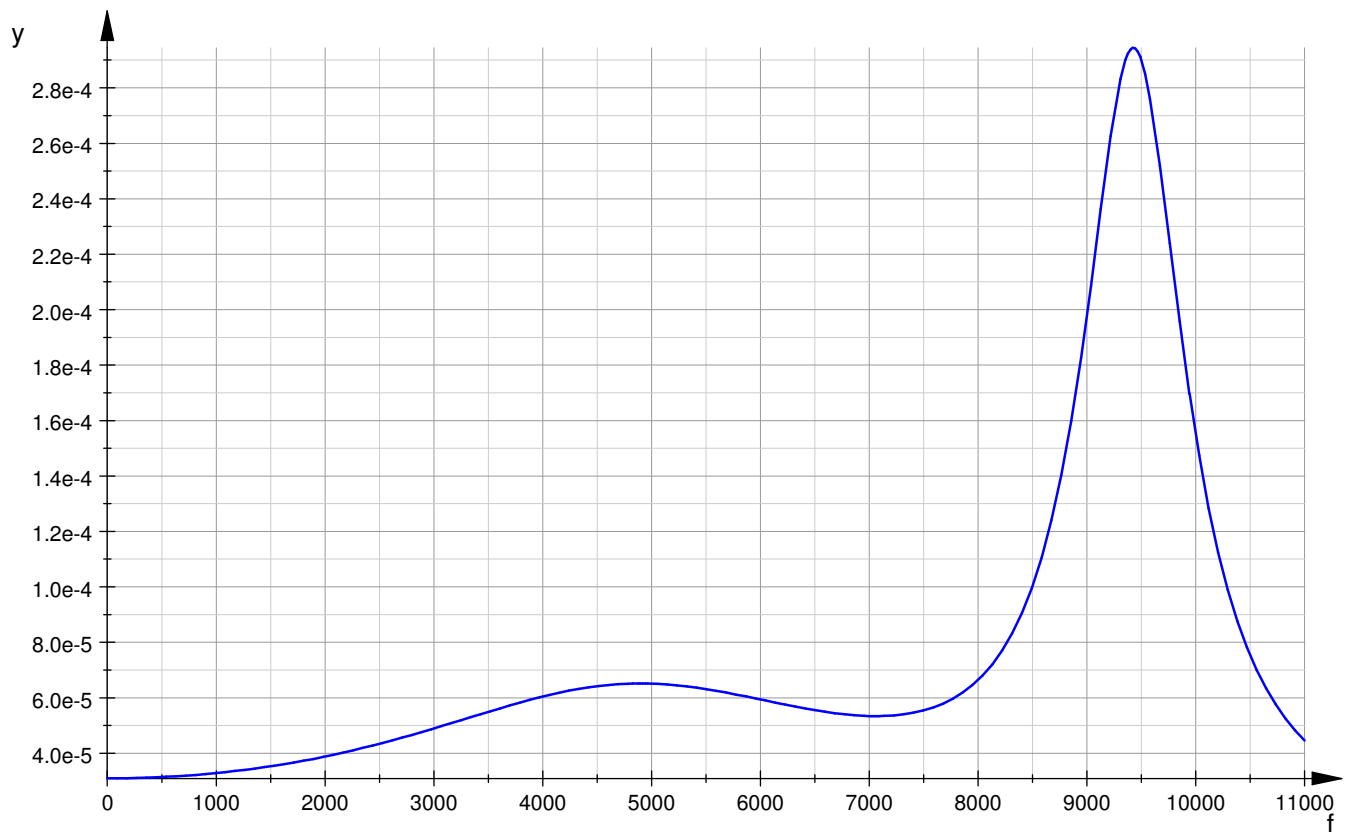
## Phasenfunktion



### 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

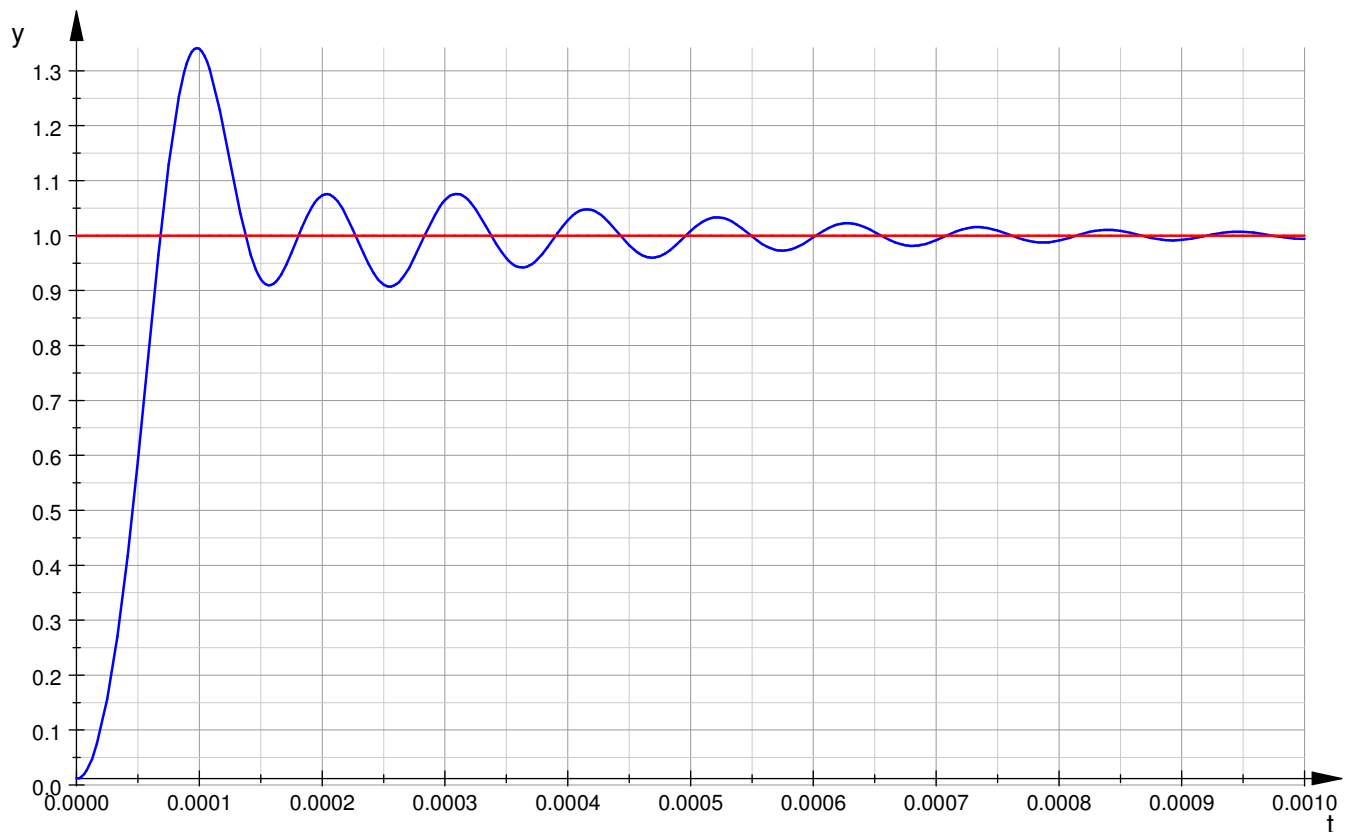
- $1 / (1 + 1/\ddot{u}^2)$;

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

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



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;`

34.132558

98.039113

Ausschnittsvergrößerung der Sprungantwort

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;`

45.803013

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;`

47.906017

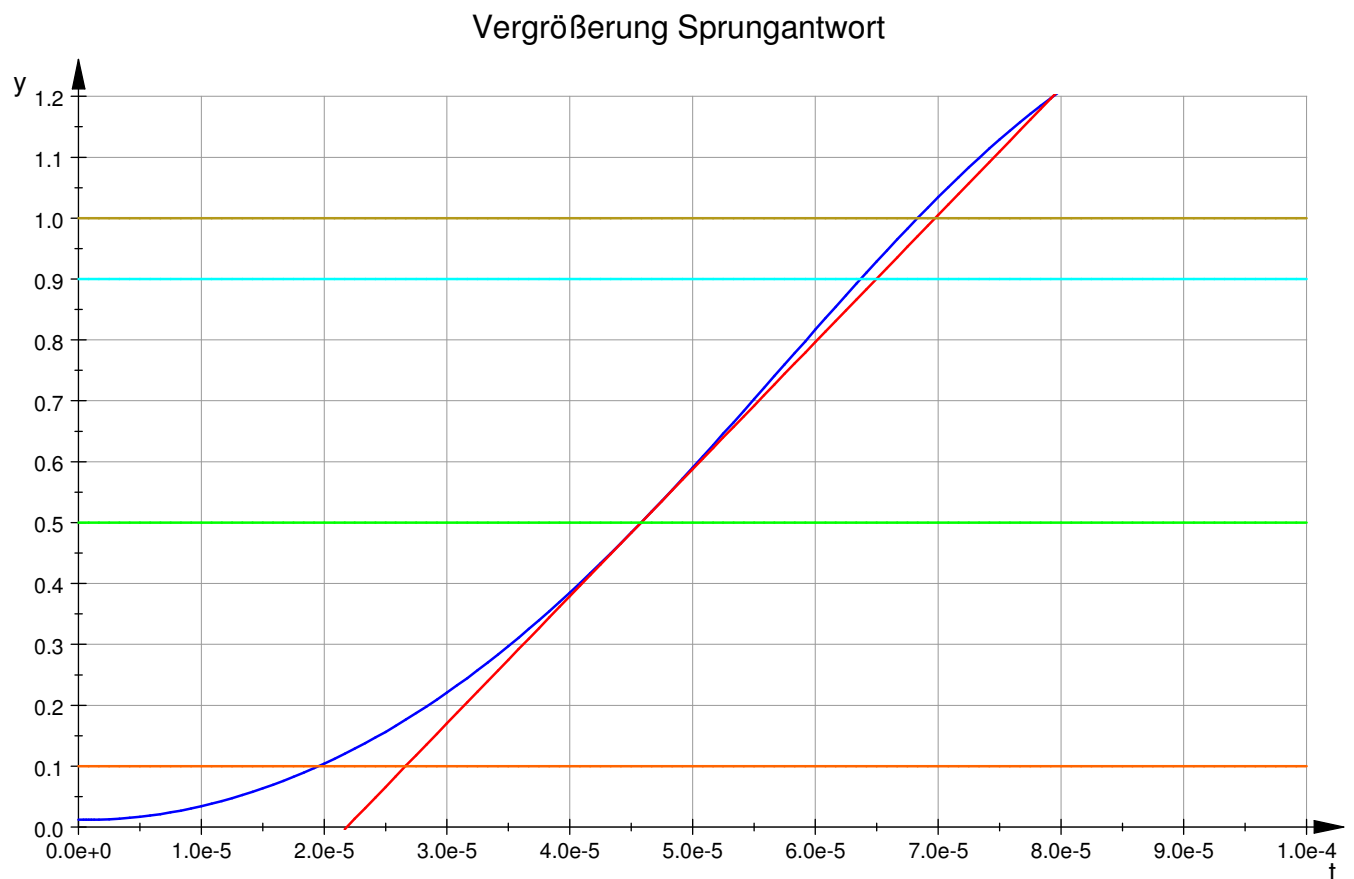
10.437102

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;`

44.134708

- `plotfunc2d(ua(t), yt(t), 1/2, 1, 1/10, 9/10, t=0..1e-4, 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);`

9.25

0.15416667

