

```

# abfragen.ps1
# 17.02.2026
# Dialogabfrage des Netzes
# Parameter ist obligatorisch und ein entwerteter und
# geklammerter Vektor mit Komma als Dezimaltrenner
# -----
Set-StrictMode -Version Latest
. ./Tools/netzFktn.ps1
. ./Tools/alphaHolen.ps1

# Aufruf ohne Params oder mit mehr als einem
# -----

if($args.count -ne 1) {
    Write-Host ('-----')
    Write-Host ('Der Aufruf erwartet genau einen entwerteten')
    Write-Host ('String mit einem nicht leeren, geklammerten und')
    Write-Host ('wortlückentoleranten Eingabevektor, dessen')
    Write-Host ('Komponenten ein Komma als Dezimaltrenner haben')
    Write-Host ('und durch Wortlücken voneinander getrennt sind!')
    Write-Host ('Beispiel mit 2 Komponenten:')
    Write-Host ('    ./abfragen "( 0,9  0,5027 )"    oder')
    Write-Host ('    ./abfragen "'( 0,9  0,5027 )'"')
    Write-Host ('-----')
    return
}

# Aufruf mit genau einem Param: Stimmt sein Aufbau?
# -----
$P = $args[0]
if(! ($P -match '^\s*\([^()]+\)\s*$')) {
    throw 'Argument ist falsch geklammert'
}
$P = $P.trim() # entklammern
$P = $P -replace '^\(', ''
$P = $P -replace '\)$', ''
$P = $P.trim()
$toks = $P -split '\s+' # in Komponenten zerlegen
$anzP = $toks.count

# Matrizen-Info auswerten
# -----
$matrizen = leseMatKonfig
$anzK = $matrizen[0,1]

# Argumentenzahl prüfen
# -----
if($anzP -ne $anzK) {
    throw "Vektor hat $anzP braucht $anzK Komponenten"
}

# Typ und Wert der Komponenten prüfen und Vektor bilden
# -----
$ev = @()
foreach($t in $toks) {
    if(! [double]::TryParse($t, [ref]$null)) {
        throw "Komponente $t muss Double sein"
    }
    $t = [double]($t.replace(',','.'))
    if(($t -le 0) -or ($t -ge 1)) {

```

```

        throw "Komponente $t hat falschen Wert"
    }
    $ev += $t
}

# Eingabepropagierung: Gewichte einlesen
# -----
$anzM = $matrizen.getLength(0)
$gew = [object[]]::new($anzM)
foreach($i in 0..($anzM-1)) {
    $gew[$i] = leseGewMat $i $matrizen[$i,0] $matrizen[$i,1]
}

# Eingabepropagierung: Eingabevektor ist ev
# -----
$vec = $ev
foreach($m in 0..($anzM-1)) {
    $vec = sigmV (multMV $gew[$m] $vec)
}

# Ausgabe
# -----
Write-Host 'Komponenten des Ausgabevektors:'
Write-Host $vec

```