In [1]:
```
using Symbolics
@variables kappa
```

Out[1]:
```
1-element Vector{Num}:
 kappa
```

In [2]:
```
c=5
w=10
A=[1-w*kappa -c*kappa; 1 0]
```

Out[2]:
```
2×2 Matrix{Num}:
 1 - 10kappa  -5kappa
           1        0
```

In [3]:
```
fAs="fA(kappa)="*string(Symbolics.toexpr(A))
eval(Meta.parse(fAs))
```

Out[3]:
```
fA (generic function with 1 method)
```

In [4]:
```
fA(kappa)
```

Out[4]:
```
2×2 Matrix{Num}:
 1 - 10kappa  -5kappa
           1        0
```

In [5]:
```
fA(0.1)
```

Out[5]:
```
2×2 Matrix{Float64}:
 0.0  -0.5
 1.0   0.0
```

In [6]:
```
k1=(w+2*c-2*sqrt(w*c+c^2))/w^2
```

Out[6]:
```
0.026794919243112253
```

In [7]:
```
k2=(w+2*c+2*sqrt(w*c+c^2))/w^2
```

Out[7]:
```
0.3732050807568878
```

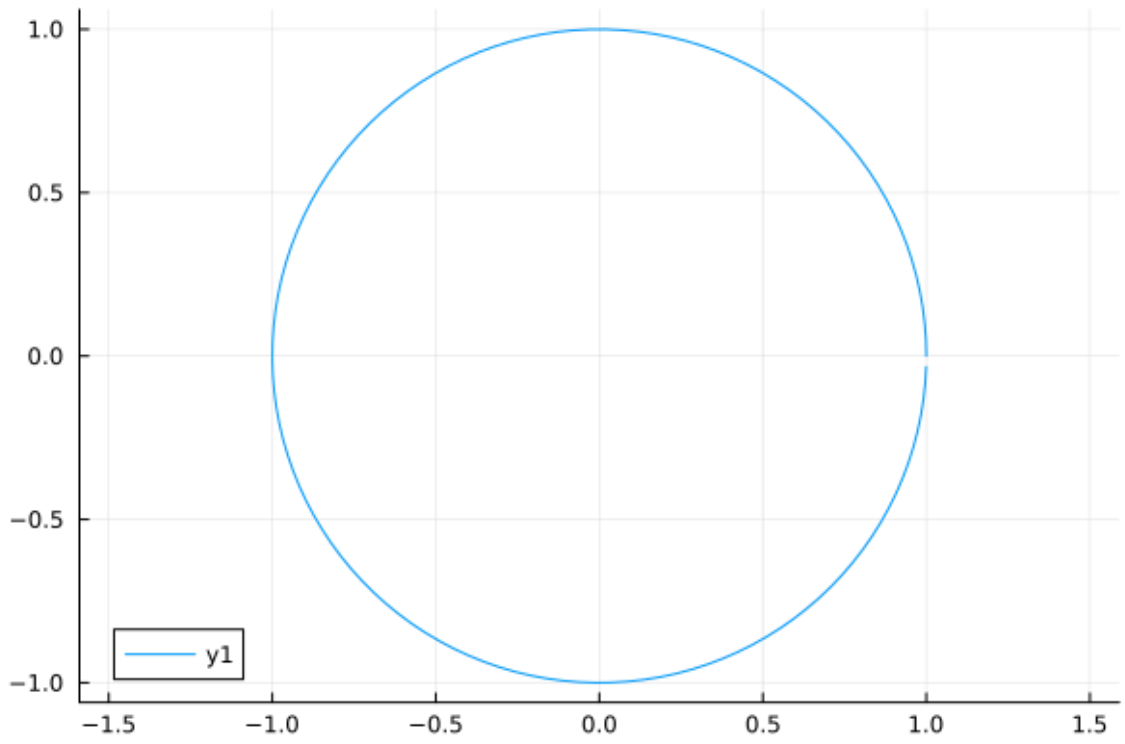
In [8]:
```
circ=[[cos(x),sin(x)] for x=0:0.05:2*pi]
```

Out[8]:
```
126-element Vector{Vector{Float64}}:
 [1.0, 0.0]
 [0.9987502603949663, 0.04997916927067833]
 [0.9950041652780258, 0.09983341664682815]
 [0.9887710779360422, 0.14943813247359924]
 [0.9800665778412416, 0.19866933079506122]
 [0.9689124217106447, 0.24740395925452294]
 [0.955336489125606, 0.2955202066613396]
 [0.9393727128473789, 0.3428978074554514]
 [0.9210609940028851, 0.3894183423086505]
 [0.9004471023526769, 0.43496553411123023]
 [0.8775825618903728, 0.479425538604203]
 [0.8525245220595057, 0.5226872289306592]
 [0.8253356149096782, 0.5646424733950355]
 ⋮
 [0.8347127848391598, -0.5506855425976376]
 [0.8611924171615208, -0.5082790774992584]
 [0.8855195169413194, -0.4646021794137566]
 [0.9076332790984135, -0.4197640178398589]
 [0.9274784307440359, -0.373876664830236]
 [0.9450053693342275, -0.32705481486974064]
 [0.960170286650366, -0.27941549819892586]
 [0.9729352782968976, -0.23107778829939138]
 [0.9832684384425847, -0.18216250427209502]
 [0.991143939568469, -0.13279190885251674]
 [0.9965420970232175, -0.0830894028174964]
 [0.9994494182244994, -0.03317921654755682]
```

In [9]:
```
using Plots
```

In [10]:
```
plot(first.(circ),last.(circ),aspect_ratio=1)
```

Out[10]:



In [11]:
```
A1=fA(0.1)
```

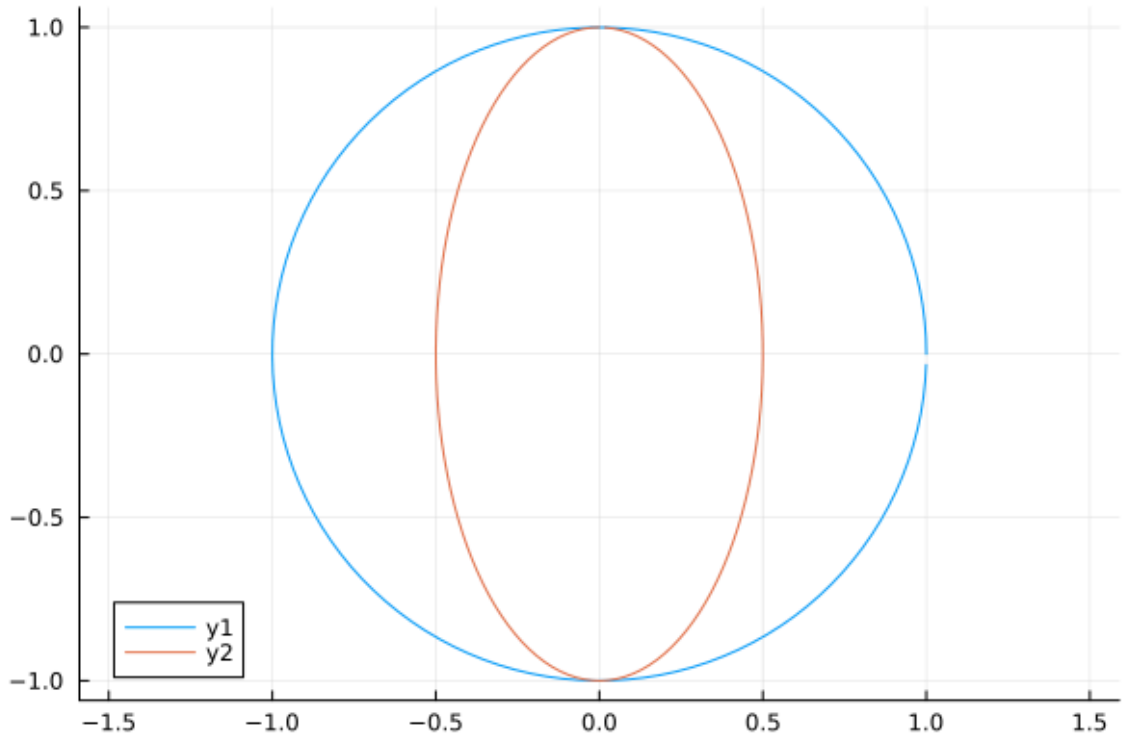
Out[11]:
```
2×2 Matrix{Float64}:
 0.0  -0.5
 1.0   0.0
```

In [12]:
```
circ1=(x->A1*x).(circ)
```

Out[12]:
```
126-element Vector{Vector{Float64}}:
 [0.0, 1.0]
 [-0.024989584635339165, 0.9987502603949663]
 [-0.04991670832341408, 0.9950041652780258]
 [-0.07471906623679962, 0.9887710779360422]
 [-0.09933466539753061, 0.9800665778412416]
 [-0.12370197962726147, 0.9689124217106447]
 [-0.1477601033306698, 0.955336489125606]
 [-0.1714489037277257, 0.9393727128473789]
 [-0.19470917115432526, 0.9210609940028851]
 [-0.21748276705561512, 0.9004471023526769]
 [-0.2397127693021015, 0.8775825618903728]
 [-0.2613436144653296, 0.8525245220595057]
 [-0.28232123669751774, 0.8253356149096782]
 ⋮
 [0.2753427712988188, 0.8347127848391598]
 [0.2541395387496292, 0.8611924171615208]
 [0.2323010897068783, 0.8855195169413194]
 [0.20988200891992945, 0.9076332790984135]
 [0.186938332415118, 0.9274784307440359]
 [0.16352740743487032, 0.9450053693342275]
 [0.13970774909946293, 0.960170286650366]
 [0.11553889414969569, 0.9729352782968976]
 [0.09108125213604751, 0.9832684384425847]
 [0.06639595442625837, 0.991143939568469]
 [0.0415447014087482, 0.9965420970232175]
 [0.01658960827377841, 0.9994494182244994]
```

In [13]:
```
plot(first.(circ),last.(circ),aspect_ratio=1)
plot!(first.(circ1),last.(circ1))
```

Out[13]:



In [14]:
```
A1
```

Out[14]:
```
2×2 Matrix{Float64}:
 0.0  -0.5
 1.0   0.0
```

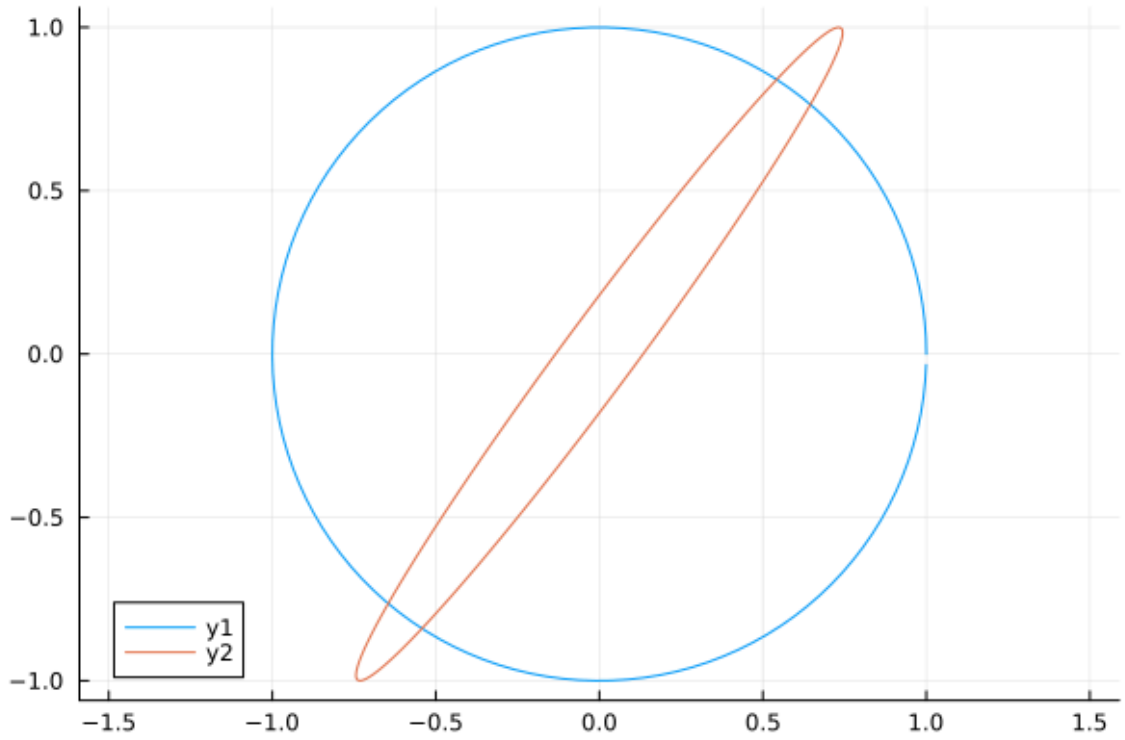
In [15]:
```
using LinearAlgebra
eigvecs(A1)
```

Out[15]:
```
2×2 Matrix{ComplexF64}:
      0.0-0.57735im       0.0+0.57735im
 0.816497-0.0im      0.816497+0.0im
```

In [16]:
```
circk1=(x->fA(k1)*x).(circ);
```

In [17]:
```
plot(first.(circ),last.(circ),aspect_ratio=1)
plot!(first.(circk1),last.(circk1))
```

Out[17]:



In [18]:
```
eigvals(fA(k1))
```

Out[18]:
```
2-element Vector{Float64}:
 0.36602539247562665
 0.36602541509325087
```

In [19]:
```
eigvecs(fA(k1))
```

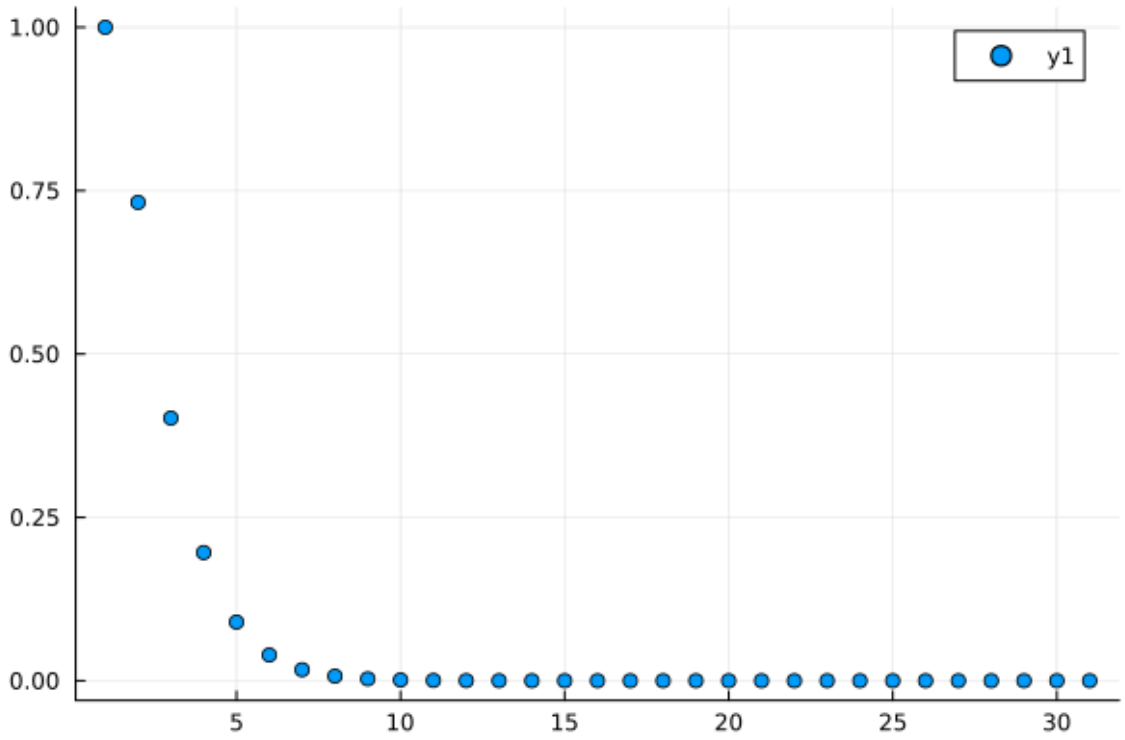
Out[19]:
```
2×2 Matrix{Float64}:
 0.343724  0.343724
 0.939071  0.939071
```

Now let's compute the solution...

In [20]:
```
kappa=k1
v0=1
a0=0
v1=v0+c*a0-w*kappa*v0
myA=fA(kappa)
Xn=[v1,v0]
vs=[v0,v1]
for n=2:30
    Xn=myA*Xn
    push!(vs,Xn[1])
end
```

In [21]:
```
scatter(vs)
```

Out[21]:



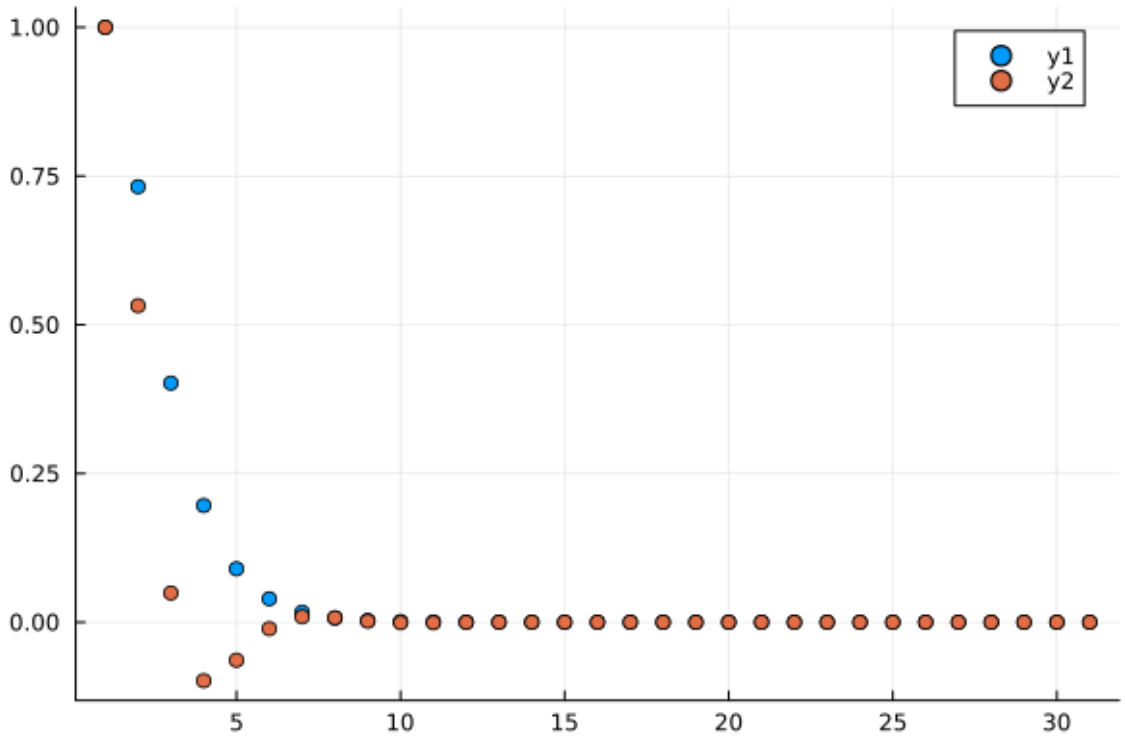
In [22]:

```
kappa=k1+0.02
v0=1
a0=0
v1=v0+c*a0-w*kappa*v0
myA=fA(kappa)
Xn=[v1,v0]
vs2=[v0,v1]
for n=2:30
    Xn=myA*Xn
    push!(vs2,Xn[1])
end
```

In [23]:

```
scatter(vs)
scatter!(vs2)
```

Out[23]:



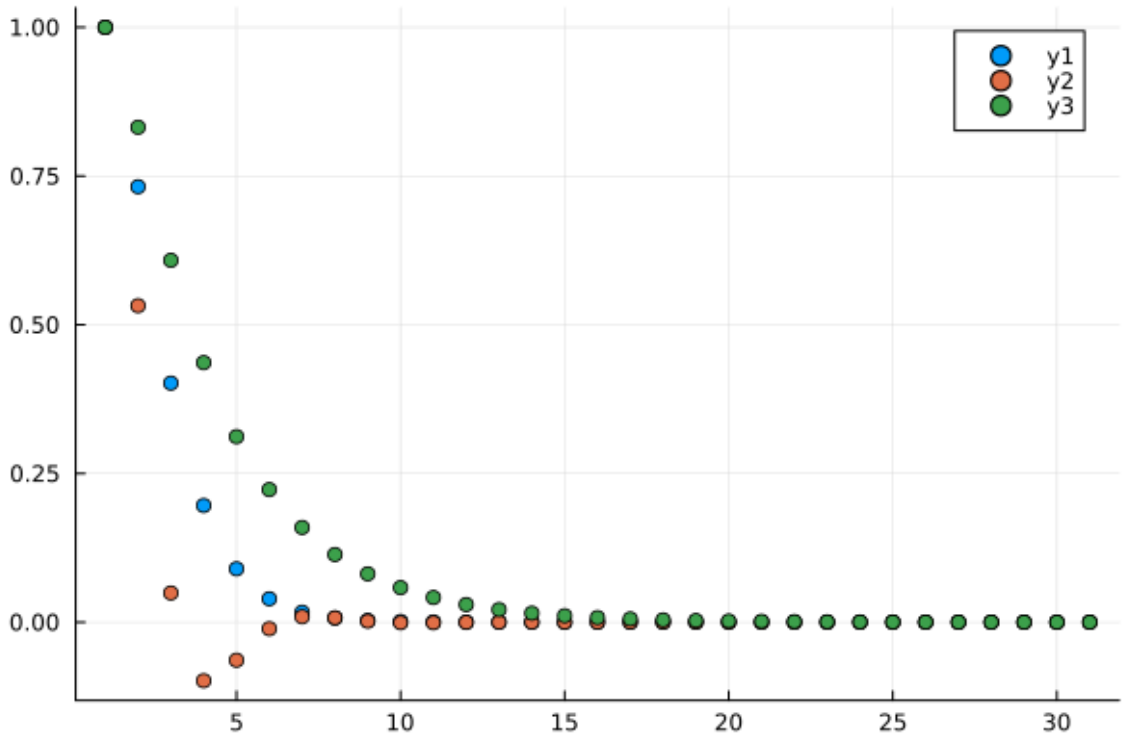
In [24]:

```
kappa=k1-0.01
v0=1
a0=0
v1=v0+c*a0-w*kappa*v0
myA=fA(kappa)
Xn=[v1,v0]
vsm1=[v0,v1]
for n=2:30
    Xn=myA*Xn
    push!(vsm1,Xn[1])
end
```

In [25]:

```
scatter(vs)
scatter!(vs2)
scatter!(vsm1)
```

Out[25]:



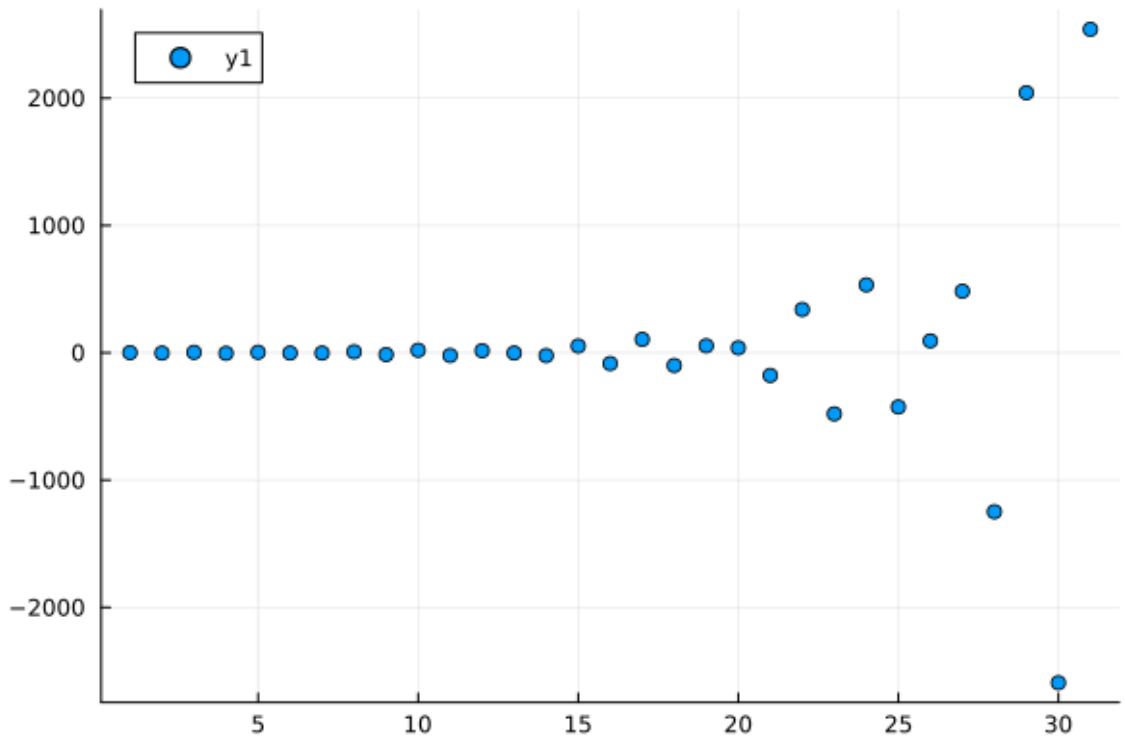
In [26]:

```
kappa=k1+0.3
v0=1
a0=0
v1=v0+c*a0-w*kappa*v0
myA=fA(kappa)
Xn=[v1,v0]
vs3=[v0,v1]
for n=2:30
    Xn=myA*Xn
    push!(vs3,Xn[1])
end
```

In [28]:

```
# velocities don't converge to zero
scatter(vs3)
```

Out[28]:



In [ ]: