KinsolSolver

```
#
#
#
#
#
#
#
```

```
from casadi import *
from numpy import *
from pylab import *
```

We will investigate the working of rootfinder with the help of the parametrically exited Duffing equation.
Parameters

```
eps   = SX.sym("eps")
mu    = SX.sym("mu")
alpha = SX.sym("alpha")
k     = SX.sym("k")
sigma = SX.sym("sigma")
params = [eps,mu,alpha,k,sigma]
```

Variables

```
a     = SX.sym("a")
gamma = SX.sym("gamma")
```

Equations

```
res0 = mu*a+1.0/2*k*a*sin(gamma)
res1 = -sigma * a + 3.0/4*alpha*a**3+k*a*cos(gamma)
```

Numerical values

```
sigma_ = 0.1
alpha_ = 0.1
k_     = 0.2
params_ = [0.1,0.1,alpha_,k_,sigma_]
```

We create a Function instance

```
f=Function("f", [vertcat(a,gamma),vertcat(*params)],[vertcat(res0,res1)])
opts = {}
opts["strategy"] = "linesearch"
opts["abstol"] = 1e-14

opts["constraints"] = [2,-2]
s=rootfinder("s", "kinsol", f, opts)

x_ = s([1,-1], params_)
print "Solution = ", x_
```

```
Solution = [1.1547, -1.5708]
```

Compare with the analytic solution:

```
x = [sqrt(4.0/3*sigma_/alpha_),-0.5*pi]
print "Reference solution = ", x
```

```
Reference solution = [1.1547005383792515, -1.5707963267948966]
```

We show that the residual is indeed (close to) zero

```
residual = f(x_, params_)
print "residual = ", residual
```

```
residual = [4.16334e-15, 8.34363e-15]
```

```

for i in range(1):
  assert(abs(x_[i]-x[i])<1e-6)
```

Solver statistics

```
print s.stats()
```

```
{}
```