

forceMat (generic function with 1 method)

```

1 #####calculate force matrix#####
2 function forceMat(parPosTmp::Matrix{Float64},Rs::Float64)
3   parPos=deepcopy(parPosTmp);
4   δh=0.0001;
5   dimension=3;
6   dimension2=2;
7   rndDigNo=12;
8   pNo=length(parPos[:,1]);
9   kMat=Matrix{Float64}(undef,pNo*dimension2,pNo*dimension2);
10  for iTmp in 1:pNo
11    for jTmp in 1:dimension2
12      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]+2*δh;
13      f1=round.(forcePackHigh(Rs,parPos); digits=rndDigNo);
14      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]-2*δh;
15      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]+δh;
16      f2=round.(forcePackHigh(Rs,parPos); digits=rndDigNo);
17      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]-δh;
18      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]-2*δh;
19      f3=round.(forcePackHigh(Rs,parPos); digits=rndDigNo);
20      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]+2*δh;
21      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]-δh;
22      f4=round.(forcePackHigh(Rs,parPos); digits=rndDigNo);
23      parPos[iTmp,jTmp]=parPos[iTmp,jTmp]+δh;
24      kk1=(-f1.+8*f2.-8*f4.+f3)/12/δh;
25      if dimension2==dimension
26        kk2=copy(kk1);
27      else
28        kk2=Array{Float64}(undef,pNo*dimension2);
29        countTmp=1;
30        for kTmp in 1:pNo*dimension
31          if kTmp%3==0
32            continue
33          else
34            kk2[countTmp]=kk1[kTmp];
35            countTmp+=1;
36          end
37        end
38      end
39      kMat[:,(iTmp-1)*dimension2+jTmp]=kk2;
40    end
41    println("particle "*string(iTmp)*" done")
42  end
43  return kMat
44 end

```

