Some Matlab calculations

To supplement lecture notes

Anindya Chatterjee

November 12, 2017

1 The Newton-Raphson method for solving simultaneous equations

Here is the basic code. You can experiment with the increment used for finite difference derivatives, the termination criterion, and the allowed number of iterations.

```
function x=newton(fname,x)

f0=feval(fname,x);
n=length(x);
count=0;

while (norm(f0) > 1e-10*max(1,norm(x)))*(count<60)
    epsil=1e-5;
    E=eye(n)*epsil;

    D=E; % initialization; will be overwritten
    for k=1:n
        temp=feval(fname,x+E(:,k));
        D(:,k)=(temp-f0)/epsil;
    end

    x=x-D\f0;
    f0=feval(fname,x);
    count=count+1;
end

if count >=60, x=inf; end
```

Now suppose we want to solve the following 3 simultaneous equations:

$$\begin{aligned} x+y+z^2-2 &= 0, \\ \sin(xy)-z &= 0, \\ \tan(z+1)-z-xyz &= 0. \end{aligned}$$

We write corresponding code in a new m-file (called, say, `junk.m`). Matlab window commands and results follow (there are many solutions).

```
function q=junk(x)
y=x(2); z=x(3); x=x(1);
q=[x+y+z^2-2; sin(x*y)-z; tan(z+1)-z-x*y*z];
```

```
Now Matlab command and results:

>> newton('junk',randn(3,1))
ans =
   3.709228340533302
  -1.745835279114973
  -0.191329398111399
>> newton('junk',randn(3,1))
ans =
   5.785608219099861
  -3.789804530610969
  -0.064778943274406
>> newton('junk',randn(3,1))
ans =
   2.813668771259672
  -0.971395509386618
  -0.397148257112533
>> newton('junk',randn(3,1))
ans =
   2.813668771260789
  -0.971395509386705
  -0.397148257112137
```

---

2 Elementary arc-length based continuation

Now suppose we have two equations in three unknowns (or two equations in two unknows, but with a free parameter). The set of solutions lies on a curve (or curves). For example, consider

$$\begin{aligned} x+y+z^2-2 &= 0, \\ \sin(xy)-z &= 0, \end{aligned}$$

where we can (if we wish) think of z as a parameter. For illustration, we note that for $z=0$ above, one solution is $x=0$ and $y=2$. Is there a solution that passes through this point, (0,2,0)?

We create a new m-file, `junk1.m` as below. Matlab commands and results follow. Required Matlab code follow later.

```
function q=junk1(x)
global gp
z=gp;
y=x(2); x=x(1);
q=[x+y+z^2-2; sin(x*y)-z];
```

```
Matlab commands and results follow.
```

```
>> globalsolution=[0;2;0]
globalsolution =
     0
     2
     0
>> gp=0.02;
>> newton('junk1',[0;2])
ans =
   0.010053210954130
   1.989546789045870
>> globalsolution=[globalsolution,[ans;gp]]
globalsolution =
                   0   0.010053210954130
   2.000000000000000   1.989546789045870
                   0   0.020000000000000
>> run_continuation('junk1',2);
>> globalsolution
globalsolution =
                   0   0.010053210954130   0.020044163194762   0.029978951297443
   2.000000000000000   1.989546789045870   1.978384137560396   1.966549371291992
                   0   0.020000000000000   0.039644662249140   0.058920942037366
```

Refer to class notes, or discuss with your classmates, if you do not recall what the above means. One clue: the bottom row contains the free-parameter values (in this case, z). Now only the background Matlab code is needed.

```
function run_continuation(fname,N)

global globalsolution continuationfname
continuationfname=fname;

for k=1:N
    Xg=2*globalsolution(:,end)-globalsolution(:,end-1);
    Xg=newton('intermed',Xg);
    globalsolution=[globalsolution,Xg];
end
```

```
function z=intermed(X)

global globalsolution gp continuationfname
[m,n]=size(globalsolution);
nx=m-1;
ds=norm(globalsolution(:,n-1)-globalsolution(:,n));

x=X(1:nx);
gp=X(m);
z=feval(continuationfname,x);
z=[z; norm(globalsolution(:,end)-X)-ds];
```
