MATH661 HW10 - Nonlinear operators

Posted: 11/27/23
Due: 12/06/23, 11:59PM

The simplest nonlinear operator is a scalar function $f(x)$, and a basic problem is to find the null set of f, those values x for which $f(x)=0$, known as the roots of f.

Consider $f(x)=p_8(x)=\sum_{i=0}^{8} a_i\,x^i$, an eigth degree polynomial with $\boldsymbol{a}=[\ a_0\ \ \ldots\ \ a_8\ ]$

$$\boldsymbol{a}=[\ 40320\ \ -109584\ \ 118124\ \ -67284\ \ 22449\ \ -4536\ \ 546\ \ -36\ \ 1\ ].$$

Define f, f' evaluation using Horner's algorithm. The above polynomial has roots 1,2,..,8, with large derivative $p_8'(x)$ values at the roots.

```
∴ a=[40320, -109584, 118124, -67284, 22449, -4536, 546, -36, 1];
```

```
∴ n=length(a)-1; ap=a[2:n+1] .* range(1,n);
```

```
∴ function poly(x,a)
    n=length(a); p=a[n]
    for k=n-1:-1:1
      p = p*x + a[k]
    end
    return p
  end;
```

```
∴ poly.(1:8,Ref(a))'
```

$$[\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ ] \tag{1}$$

```
∴ poly.(1:8,Ref(ap))'
```

$$[\ -5040\ \ 720\ \ -240\ \ 144\ \ -144\ \ 240\ \ -720\ \ 5040\ ] \tag{2}$$

1 Track 1

Implement each of the following methods to find a root of p_8.

1. Seek a root $r\in[5.5,6.5]$ using the bisection algorithm (see course webpage).

   Solution. Use course webpage bisect function

```
∴ function bisect(f,a,b,ε)
    if (a>b) a,b=b,a end
    fa=f(a); fb=f(b)
    δ=b-a; c=(a+b)/2
    while ((δ>ε) && (fa*fb<=0) && abs(fa)>ε)
      δ=δ/2; c=a+δ; fc=f(c)
      if (fa*fc<=0)
        b,fb=c,fc
      else
        a,fa=c,fc
      end
    end
    return c
  end;
```

```
∴ function p8(x)
    global a,neval
    neval = neval +1
    poly(x,a)
  end;
```

```
∴ neval=0; [bisect(p8,5.5,6.5,1.0e-4) neval]
```

$$[\ 5.99993896484375 \quad 16.0\ ] \tag{3}$$

```
∴ [bisect(p8,5.5,6.5,1.0e-4) neval]
```

$$[\ 5.99993896484375 \quad 32.0\ ] \tag{4}$$

2. Seek r by the secant method.

Solution. The implementation below uses only one new function evaluation per secant iteration.

```
∴ function secant(f,x0,x1,ε,nmax)
    f0=f(x0); f1=f(x1); δ=abs(x1-x0); n=0;
    while ((δ>ε)  && abs(f1)>ε && (n<=nmax))
      df=(f1-f0)/(x1-x0); x0=x1
      x1=x1-f1/df; n=n+1; δ=abs(x1-x0)
      f0=f1; f1=f(x1)
    end
    return x1
  end;
```

```
∴ neval=0; [secant(p8,5.5,6.5,1.0e-3,100) neval]
```

$$[\ 5.999999329036827 \quad 7.0\ ] \tag{5}$$

```
∴ neval=0; [secant(p8,5.5,6.5,1.0e-5,100) neval]
```

$$[\ 6.000000000082597 \quad 8.0\ ] \tag{6}$$

```
∴
```

3. Seek r by Newton's method.

Solution. Note the sensitivity of Newton's method to the intial guess.

```
∴ function newton(f,df,x0,ε,nmax)
    δ=2*ε; n=0; x1=x0; f0=f(x0)
    while ((δ>ε) && abs(f0)>ε && (n<=nmax))
      f0=f(x0); df0=df(x0);
      x1=x0-f0/df0; δ=abs(x1-x0)
      x0=x1; n=n+1
    end
    return x1
  end;
```

```
∴ function dp8(x)
    global ap,neval
    neval = neval +1
    poly(x,ap)
  end;
```

```
∴ neval=0; [newton(p8,dp8,6.0,1.0e-3,100) neval]
```

$$[\ 6.0 \quad 1.0\ ] \tag{7}$$

```
∴ neval=0; [newton(p8,dp8,6.5,1.0e-4,100) neval]
```

$$[\ 8.00000000000131 \quad 21.0\ ] \tag{8}$$

```
∴ neval=0; [newton(p8,dp8,5.5,1.0e-4,100) neval]
```

$$[\ 4.999999999991451 \quad 13.0\ ] \tag{9}$$

```
∴ neval=0; [newton(p8,dp8,6.1,1.0e-8,100) neval]
```

$$[\ 6.0000000000097025 \quad 9.0\ ] \tag{10}$$

```
∴ neval=0; [newton(p8,dp8,6.25,1.0e-3,100) neval]
```

$$[\ 6.000000000188763 \quad 7.0\ ] \tag{11}$$

```
∴ neval=0; [newton(p8,dp8,6.375,1.0e-3,100) neval]
```

$$[\ 6.000000000053686 \quad 9.0\ ] \tag{12}$$

```
∴ neval=0; [newton(p8,dp8,6.49,1.0e-3,100) neval]
```

$$[\ 7.0000000000054055 \quad 19.0\ ] \tag{13}$$

```
∴
```

4. Seek r by Steffensen's method.

Solution. Note the threshold for the step size used in derivaive evaluation.

```
∴ function steffensen(f,x0,ε,nmax)
    δ=2*ε; n=0; x1=x0; f0=f(x0); hmax=1.0e-3
    while ((δ>ε) && abs(f0)>ε && (n<=nmax))
      f0=f(x0); h=min(abs(f0),hmax); df0=(f(x0+h)-f0)/h;
      x1=x0-f0/df0; δ=abs(x1-x0)
      x0=x1; n=n+1
    end
    return x1
  end;
```

```
∴ neval=0; [steffensen(p8,6.1,1.0e-5,100) neval]
```

$$[\ 5.999999999991587 \quad 9.0\ ] \tag{14}$$

```
∴ neval=0; [steffensen(p8,5.6,1.0e-5,100) neval]
```

$$[\ 8.000000003220128 \quad 19.0\ ] \tag{15}$$

```
∴ neval=0; [steffensen(p8,5.8,1.0e-5,100) neval]
```

$$[\ 6.000000007443956 \quad 9.0\ ] \tag{16}$$

```
∴
```

5. Change $a_7 = -36 - 10^{-3}$ and repeat the above.

Solution. Small changes in polynomial coefficients can induce large changes in root values.

```
∴ a1=1.0*copy(a); a1[8]=a1[8]-1.0e-3
```

```
∴ function p81(x)
    global a1,neval
    neval = neval +1
    poly(x,a1)
  end;
```

```
∴ r=steffensen(p81,5.6,1.0e-5,100)
```

4.574836091800582

```
∴ p81(r)
```

$-3.8562575355172157\,e-10$

```
∴
```

2 Track 2

1. Prove that Steffensen's method is of second order.

Solution. Note: Observe the introduction of intermediate notation and overall organization to reduce the analytical calculation effort. Careful identification of the structure of analytical computations is a skill set that requires consistent nurturing. The solution also contains computer-assisted symbolic computations. Open the folds to study the relevant *Mathematica* code. Install the *Mathematica* TeXmacs plugin to enable execution. The *Mathematica* results can be directly copied and pasted into the TeXmacs document. The ease with which both numerical (Julia, Python, Octave, Matlab) and symbolic (Maxima, Mathematica) computations can be integrated with publication quality typesetting is one of the most attractive features of using TeXmacs as a platform for research in the mathematical sciences.

Re-express the iteration

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n + f(x_n)) - f(x_n)} = x_n - g(x) = x_n - \frac{a(x_n)}{b(x_n)},$$

in terms of errors $e_n = x_n - r$,

$$e_{n+1} = e_n - g(x_n).$$

Carry out a Taylor series expansion around $x = r$,

$$g(x_n) = g(r) + g'(r)e_n + \frac{1}{2}g''(r)e_n^2 + \ldots.$$

The iteration is second order if $g(r) = 0$, $g'(r) = 1$. Verify using $f(r) = a(r) = b(r) = 0$, applying l'Hôpital's rule, and assuming $f'(r) \neq 0$,

- $$\lim_{x\to r} g(x) = \lim_{x\to r} \frac{a(x)}{b(x)} = \lim_{x\to r} \frac{a'(x)}{b'(x)} = \lim_{x\to r} \frac{2\,f(x)\,f'(x)}{(f'(x)+1)\,f'(f(x)+x) - f'(x)},$$

$$a'(r) = 0, b'(r) = (f'(r)+1)\,f'(f(r)+r) - f'(r) = [f'(r)]^2 \Rightarrow$$

$$\lim_{x\to r} \frac{2\,f(x)\,f'(x)}{(f'(x)+1)\,f'(f(x)+x) - f'(x)} = \frac{0}{[f'(r)]^2} = 0.$$

- $$g'(x) = \left(\frac{a(x)}{b(x)}\right)' = \frac{a'b - ab'}{b^2} = \frac{c(x)}{d(x)}$$

$$c(x) = f(x)\,(2\,f\,(f(x)+x)\,f'(x) - f(x)\,(f'\,(f(x)+x) + f'(x)\,(f'\,(f(x)+x)+1)))$$

$$c(r) = d(r) = 0 \Rightarrow \lim_{x\to r} \frac{c(x)}{d(x)} = \lim_{x\to r} \frac{c'(x)}{d'(x)}$$

$$c'(r) = 0, d'(r) = 0 \Rightarrow \lim_{x\to r} \frac{c'(x)}{d'(x)} = \lim_{x\to r} \frac{c''(x)}{d''(x)}$$

$$c''(r) = 2\,f'(r)^4, d''(r) = 2\,f'(r)^4 \Rightarrow g'(r) = 1. \checkmark$$

2. Implement Steffensen's and find $r \in [5.5,\ 6.5]$, a root of a perturbed $\tilde{p}_8(t)$, where $\tilde{a}_2 = a_2 + \varepsilon_k$, $\varepsilon_k = 2^{-k}$, $k \in \{15, 14, \ldots, 10\}$. Comment on what you observe.

   Solution. See above.

3. Apply the vector-valued version of Newton's method

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{J}_n^{-1}\,\boldsymbol{f}(\boldsymbol{x}_n), \tag{17}$$

   where $\boldsymbol{J}_n$ is the Jacobian

$$\boldsymbol{J}_n = \boldsymbol{f}'(\boldsymbol{x}_n) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_n),$$

   to find a root of

$$\begin{cases} uv - w^2 = 1 \\ uvw - u^2 + v^2 = 2 \\ e^u - e^v + w = 3 \end{cases}.$$

Implementation notes:

- As usual, $\boldsymbol{J}^{-1}$ is implemented as linear system solve

$$\boldsymbol{J}\boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}_n) \Rightarrow \boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{s}$$

- An initial guess is typically obtained by linearization of the system.

Solution. Define $\boldsymbol{f}$, $\boldsymbol{J}$

- $$\boldsymbol{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} u\,v - w^2 - 1 \\ u\,v\,w - u^2 + v^2 - 2 \\ e^u - e^v + w - 3 \end{bmatrix}, \boldsymbol{J} = \begin{bmatrix} v & u & -2\,w \\ v\,w - 2\,u & u\,w + 2\,v & u\,v \\ e^u \log(e) & -e^v \log(e) & 1 \end{bmatrix}.$$

Mathematica definitions to carry out symbolic calculations

```
In[1]:= x={u,v,w}
```

$\{u, v, w\}$

```
In[2]:= f={u v-w^2-1,u v w-u^2+v^2-2,Exp[u]-Exp[v]+w-3}
```

$\{u\,v - w^2 - 1, -u^2 + u\,v\,w + v^2 - 2, e^u - e^v + w - 3\}$

```
In[6]:= J=D[f,{x}]
```

$$\begin{pmatrix} v & u & -2\,w \\ v\,w - 2\,u & u\,w + 2\,v & u\,v \\ e^u & -e^v & 1 \end{pmatrix}$$

```
In[8]:=
```

Julia definitions to carry out numerical computations

```
∴ f(u,v,w)=[u*v-w^2-1; u*v*w-u^2+v^2-2; exp(u)-exp(v)+w-3]
```

f

```
∴ J(u,v,w)=[v u -2*w; v*w-2*u u*w+2*v u*v; exp(u) -exp(v) 1]
```

J

```
∴ function f(x)
    (u,v,w)=x
    return f(u,v,w)
  end
```

f

```
∴ function J(x)
    (u,v,w)=x
    return J(u,v,w)
  end
```

J

```
∴ f(1,1,1)
```

$$\begin{bmatrix} -1.0 \\ -1.0 \\ -2.0 \end{bmatrix} \tag{18}$$

```
∴ f([1,1,1])
```

$$\begin{bmatrix} -1.0 \\ -1.0 \\ -2.0 \end{bmatrix} \tag{19}$$

```
∴ J(1,1,1)
```

$$\begin{bmatrix} 1.0 & 1.0 & -2.0 \\ -1.0 & 3.0 & 1.0 \\ 2.718281828459045 & -2.718281828459045 & 1.0 \end{bmatrix} \tag{20}$$

```
∴ J([1,1,1])
```

$$\begin{bmatrix} 1.0 & 1.0 & -2.0 \\ -1.0 & 3.0 & 1.0 \\ 2.718281828459045 & -2.718281828459045 & 1.0 \end{bmatrix} \tag{21}$$

```
∴
```

Linearize around $\boldsymbol{\xi} = \mathbf{3}/\mathbf{2}$,

- $$\boldsymbol{f}(\boldsymbol{x}) \cong \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{J}(\boldsymbol{\xi})(\boldsymbol{x} - \boldsymbol{\xi}) = \begin{bmatrix} -1. \\ 1.375 \\ -1.5 \end{bmatrix} + \begin{bmatrix} 1.5 & 1.5 & -3. \\ -0.75 & 5.25 & 2.25 \\ 4.48169 & -4.48169 & 1 \end{bmatrix} \begin{bmatrix} u - 1.5 \\ v - 1.5 \\ w - 1.5 \end{bmatrix}$$

Set $\boldsymbol{f}(\boldsymbol{r}) = \mathbf{0}$, and solve the linearized system to find an initial approximation

- $$\mathbf{0} = \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{J}(\boldsymbol{\xi})\boldsymbol{s}, \boldsymbol{s} = \boldsymbol{x}_0 - \boldsymbol{\xi} \Rightarrow \boldsymbol{x}_0 = \boldsymbol{s} + \boldsymbol{\xi} = \begin{bmatrix} 1.776648321339999 \\ 1.3856873536466667 \\ 1.2478345041599996 \end{bmatrix}$$

Carry out a few Newton iterations, checking if $\|\boldsymbol{s}\|$ decreases indicating convergence.

- $$\boldsymbol{J}_0\, \boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}_0) \Rightarrow \boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{s} = \begin{bmatrix} 1.7772496118205008 \\ 1.4240395897813467 \\ 1.2373202082484331 \end{bmatrix}, \|\boldsymbol{s}\| = 0.0397719.$$

- $$\boldsymbol{J}_1\, \boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}_1) \Rightarrow \boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{s} = \begin{bmatrix} 1.7776719798768659 \\ 1.4239605716049761 \\ 1.2374711571394028 \end{bmatrix}, \|\boldsymbol{s}\| = 0.000455.$$

- $$\boldsymbol{J}_2\, \boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}_2) \Rightarrow \boldsymbol{x}_3 = \boldsymbol{x}_2 + \boldsymbol{s} = \begin{bmatrix} 1.7776719180107414 \\ 1.4239605978884882 \\ 1.2374711177317046 \end{bmatrix}, \|\boldsymbol{s}\| = 7.792\, e - 8.$$

- After 3 iterations find

$$\boldsymbol{r} \cong \boldsymbol{x}_3 = \begin{bmatrix} 1.7776719180107414 \\ 1.4239605978884882 \\ 1.2374711177317046 \end{bmatrix}, \boldsymbol{f}(\boldsymbol{x}_3) = \begin{bmatrix} -3.552713678800501\, e - 15 \\ -3.1086244689504383\, e - 15 \\ 9.769962616701378\, e - 15 \end{bmatrix} = \mathcal{O}(\epsilon)$$