HW2.6

February 21, 2019

2.6 Homework

Any polynomial of degree n has the form

$$P(x) = P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

where the a_i's, are the coefficients of P_n and $a_n \neq 0$.

Consider the Horner's Method to evaluate $P(x)$ and $P'(x)$ at specified values as follows:

Theorem (Horner's Method)

Let
$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0.$$

Define $b_n = a_n$ and

$$b_k = a_k + b_{k+1}x_0, \qquad k = n-1, n-2, \ldots, 1, 0.$$

Then $b_0 = P(x_0)$. Moreover, if

$$Q(x) = b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2 x + b_1,$$

then
$$P(x) = (x - x_0)Q(x) + b_0.$$

Moreover,

$$P'(x) = Q(x) + (x - x)Q'(x) \qquad \text{and} \qquad P'(x_0) = Q(x_0).$$

Exercise 1:

We assume that all a_i's are non zeros.

1. Determine how many flops (algebraic operations) we will need to evaluate $P(x)$ at x_0 **directly**.
2. Determine how many flops we will need to evaluate $P(x)$ at x_0 by using Horner's Method.

3. Consider the problem to find x such that

$$P(x) = 0.$$

When the Newton's method is being used to find an approximate zero of a polynomial, $P(x)$ and $P'(x)$ can be evaluated. Based on the before items 1. and 2. compute how many flops we will need to apply 10 iterations of Newton Method by evaluating $P(x)$ and $P'(x)$ **(a)** directly and **(b)** via Horner's Method. Here you can assume that the calculation of the coefficients of P' is free.

4. Implement the Newton Method efficiently by using Horner Method to evaluate $P(x)$ and $P'(x)$ creating a function in JULIA as follows:

5. Use it to find the zeros of $x^5 - x^4 + 2x^3 - 3x^2 + x - 4$ with a tolerance 10^{-8}.

6. Use it to find the critical points of $x^4 + 2x^3 - 3x^2 + x - 4$ with a tolerance 10^{-8}.

Given the coefficients of a polynomial coefs=[list] the Horner evaluation at x is:

```
In [1]: function horner(coefs, x)
            s = coefs[end]
            for k in length(coefs)-1:-1:1
                s = coefs[k] + x * s  # b_k = a_k + x_0*b_{k+1}
            end
            return s
        end
```

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

The coefficient of $Q(x)$ such that at some given point $\bar{x}$, $Q(\bar{x}) = P'(\bar{x})$ are:

```
In [2]: function hornerVect(coefs, x)
            coefsnew = zeros(coefs)
            coefsnew[end] = coefs[end]
            for k in length(coefs)-1:-1:1
                coefsnew[k] = coefs[k] + x * coefsnew[k+1]  # b_k = a_k + x_0*b_{k+1}
            end
            return coefsnew[2:length(coefsnew)]
        end
```

```
Out[2]: hornerVect (generic function with 1 method)
```

Your Newton function could be something like this: Given a tolerance (tol=1e-6) and maximum of iterations (maxiter=10)

```
In [3]: using Printf
        function yourNewtonHorner(coefs, x0; tol=1e-6, maxiter=10)
            k=0
            x = x0
            px=horner(coefs,x0)
            derivcoef=hornerVect(coefs, x0)
```

```
        qx=horner(derivcoef,x0)
        xold, xnew = x, Inf
        @printf "%4s %22s  %22s\n" "k" "x" "p(x)"
        @printf "%4d %22.15e  %22.15e\n" k x px
        while abs(xnew - xold) > tol*(1 + abs(xold)) && k < maxiter
            k += 1
            xnew = x - px/qx
            xold = x
            x = xnew
            px=horner(coefs,xnew)
            derivcoef=hornerVect(coefs, xnew)
            qx=horner(derivcoef,xnew)
            @printf "%4d %22.15e %22.15e\n" k x px
        end
        return x
    end
```

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

(Re-write the above code if needed!!!)