NA08 Izvrednjavanje funkcija

Ivan Slapničar

29. listopada 2018.

1 Izvrednjavanje funkcija

Računalo može izvoditi samo četiri osnovne operacije, +, -, * i / pa se sve ostale funkcije računaju pomoću polinoma (npr. Taylorova formula uz ocjenu ostatka ili bolje formule).

Neka je zadan polinom *stupnja* n:

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{n-1}x^{n-1} + a_n x^n, \quad a_n \neq 0.$$

1.1 Brzina

Direktno računanje vrijednosti $p_n(x)$ treba $O(n^2)$ operacija.

Uz **pamćenje potencija** imamo sljedeći algoritam:

```
In [1]: using Polynomials
```

```
In [2]: function mypolyval(p::Poly,x::Number)
            s=p[0]
            t=one(typeof(p[0]))
            for i=1:length(p)-1
                t*=x
                s+=p[i]*t
            end
            s
            end
```

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

```
In [3]: p=Poly([1,2,3,4,5])
```

```
Out[3]: Poly(1 + 2*x + 3*x^2 + 4*x^3 + 5*x^4)
```

```
In [4]: mypolyval(p,3)
```

Out[4]: 547

In [5]: mypolyval(p,π)

Out[5]: 647.962560401659

Funkcija `mypolyval()` koristi $2n$ množenje i n zbrajanja.

In [10]:
```
srand(123)
pbig=Poly(rand(1000));
```

In [11]:
```
@time mypolyval(pbig,1.5)
```

```
  0.000008 seconds (5 allocations: 176 bytes)
```

Out[11]: 1.0876081198598302e176

In [12]:
```
@time polyval(pbig,1.5)
```

```
  0.000011 seconds (5 allocations: 176 bytes)
```

Out[12]: 1.0876081198598281e176

Hornerova shema (Horner, 1819, Newton 1669) treba n množenja i n zbrajanja:

In [13]:
```
function myhorner(p::Poly,x::Number)
    s=p[end]
    for i=length(p)-2:-1:0
        # s*=x
        # s+=p[i]
        s=s*x+p[i]
    end
    s
end
```

Out[13]: myhorner (generic function with 1 method)

In [14]: myhorner(p,3)

Out[14]: 547

In [16]:
```
@time myhorner(pbig,1.5)
```

```
  0.000010 seconds (5 allocations: 176 bytes)
```

Out[16]: 1.0876081198598281e176

Hornerova shema je **optimalna** u smislu da je općenito za izvrednjavanje polinoma $p_n(x)$ potrebno barem n množenja.

(Mogući su, naravno, posebni slučajevi, kao x^{100}.)

1.2 Točnost

Neka je $\hat q$ vrijednost $p_n(x)$ izračunata u aritmetici s točnošću stroja ε. Tada vrijedi ocjena (vidi Accuracy and Stability of Numerical Algorithms, str. 105):

$$\big|\, p_n(x) - \hat q\, \big| \leq \frac{2n\varepsilon}{1-2n\varepsilon} \sum_{i=0}^n |a_i||x|^i.$$

In [17]:
```
p=Poly([1,2,3,4,5])
myhorner(p,sqrt(2))
```

Out[17]: 41.142135623730965

In [18]:
```
pb=Poly(map(BigInt,[1,2,3,4,5]))
```

Out[18]: Poly(1 + 2*x + 3*x^2 + 4*x^3 + 5*x^4)

In [19]:
```
myhorner(pb,sqrt(map(BigFloat,2)))
```

Out[19]:
4.114213562373095048801688724209698078569671875376948073176679737990732478462071e+01

In [20]:
```
myhorner(p,sqrt(200000))
```

Out[20]: 2.0035837177182715e11

In [21]:
```
myhorner(pb,sqrt(map(BigFloat,200000)))
```

Out[21]:
2.003583717718271573513413463506664716901809931137851879604936332676245815123154e+11

In [22]:
```
r=[1,sqrt(2),3,4,5,6,sqrt(50)]
p=poly(r)
```

```
Out[22]: Poly(-3600.0000000000005 + 10074.701294725885*x - 10926.667524715478*x^2
+ 5983.714713523981*x^3 - 1813.4835482706842*x^4 + 308.2203461105329*x^5
- 27.48528137423857*x^6 + 1.0*x^7)
```

```
In [23]: pb=poly(map(BigFloat,r))
```

```
Out[23]:
Poly(-3.600000000000000379131566326426658519951798233677472418044762658695390200591646e+03
+ 1.007470129472588617359567977839658265635786507617888371518728718445601089115371e+04*x
- 1.092666752471547708236282815742121598471939907533833099921843218238493022909097e+04*x^2
+ 5.983714713523981214300865042833422625843619822133098010478145789559079048558488e+03*x^3
- 1.813483548270684199169749849270990311652746621526228579007918029208923371697892e+03*x^4
+ 3.082203461105328434184026381162611967403808614536830831056798962741538616683101e+02*x^5
- 2.748528137423857065080312622740166261792182922363281250000000000000000000000000e+01*x^6
+ 1.000000000000000000000000000000000000000000000000000000000000000000000000000000*x^7)
```

```
In [24]: myhorner(p,sqrt(2)+0.1)
```

```
Out[24]: -16.501829900900248
```

```
In [25]: myhorner(pb,sqrt(map(BigFloat,2))+0.1)
```

```
Out[25]: -1.650182990089441570965221108407569790203119747704640347484
               765272697823211215861e+01
```

```
In [26]: myhorner(p,-sqrt(10000))
```

```
Out[26]: -1.307549271826299e14
```

```
In [27]: myhorner(pb,-sqrt(map(BigFloat,10000)))
```

```
Out[27]: -1.307549271826298681134778698255153254173615274698734585218
               439050325472949865002e+14
```