NA11 Prirodni kubicni splajn

Ivan Slapničar

6. studenog 2018.

1 Prirodni kubični splajn

Neka je zadana funkcija $f(x)$ na intervalu $[a,b]$.

Odaberimo $n+1$ točku

$$a \equiv x_0 < x_1 < x_2 < \cdots < x_n \equiv b$$

i izračunajmo vrijednosti

$$y_i = f(x_i), \quad i=0,1,\ldots,n.$$

Na intervalu $[x_{i-1},x_i]$ funkciju f aproksimiramo kubičnim polinomom C_i, tako da je na intervalu $[a,b]$ funkcija f aproksimirana funkcijom

$$C(x)=C_i(x), \quad x\in[x_{i-1},x_i]$$

Od funkcije $C(x)$ tražimo

- *neprekidnost*,
- *neprekidnost prve derivacije* i
- *neprekidnost druge derivacije*.

Dakle,

$$\begin{aligned}
C_i(x_{i-1})&=y_{i-1}, \quad i=1,\ldots,n,\\
C_i(x_i)&=y_i \quad i=1,\ldots,n,\\
C_i'(x_i)&=C_{i+1}'(x_i), \quad i=1,\ldots,n-1,\\
C_i'(x_i)&=C_{i+1}'(x_i), \quad i=1,\ldots,n-1,
\end{aligned}$$

pa imamo sustav od $4n-2$ jednadžbe i $4n$ nepoznanica (svaki od n polinoma ima 4 koeficijenta).

Vrijede sljedeće tvrdnje:

$$C_i(x) = y_{i-1} - s_{i-1}\frac{h_i^2}{6} + b_i(x - x_{i-1}) + \frac{s_{i-1}}{6h_i}(x_i - x)^3 + \frac{s_i}{6h_i}(x - x_{i-1})^3,$$

gdje je

$$\begin{aligned}
b_i &= d_i - (s_i - s_{i-1})\frac{h_i}{6}, \\
d_i &= \frac{y_i - y_{i-1}}{h_i}, \\
h_i &= x_i - x_{i-1},
\end{aligned}$$

a brojevi s_i, $i = 0, 1, \ldots, n$, zadovoljavaju sustav jednadžbi

$$s_{i-1}h_i + 2s_i(h_i + h_{i+1}) + s_{i+1}h_{i+1} = 6(d_{i+1} - d_i), \quad i = 1, \ldots, n-1.$$

Ako zadamo s_0 i s_n, sustav će imati jedinstveno rješenje.

Najčešće sz zadani *prirodni uvjeti*:

$$s_0 = 0, \quad s_n = 0.$$

U tom slučaju, $s_1, \ldots, s_{n-1}$ su rješenja sustava

$$\begin{bmatrix}
2(h_1+h_2) & h_2 & 0 & \cdots & 0 & 0 \\
h_2 & 2(h_2+h_3) & h_3 & \cdots & 0 & 0 \\
0 & h_3 & 2(h_3+h_4) & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\
0 & 0 & 0 & \cdots & h_{n-1} & 2(h_{n-1}+h_n)
\end{bmatrix}
\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{bmatrix}
=
\begin{bmatrix} 6(d_2 - d_1) \\ 6(d_3 - d_2) \\ 6(d_4 - d_3) \\ \vdots \\ 6(d_{n-1} - d_{n-2} \\ 6(d_n - d_{n-1}) \end{bmatrix}.$$

Dokaz se nalazi u udžbeniku Numerička matematika, str. 29.

Matrica sustava je *tridijagonalna* i *pozitivno definitna* pa se sustav može riješiti metodom Choleskog (bez pivotiranja) u $O(n)$ operacija.

Vrijede *ocjene pogreške*:

$$\begin{aligned}
\max |f(x) - C(x)| &\le \frac{5}{384} \max h_i^4 \\
\max |f'(x) - C'(x)| &\le \frac{1}{24} \max h_i^3 \\
\max |f''(x) - C''(x)| &\le \frac{3}{8} \max h_i^2.
\end{aligned}$$

Ocjene se mogu promatrati i na svakom intervalu posebno.

1.1 Primjer - Interpolacija slučajnih točaka

In [1]:
```
using Polynomials
using Gadfly
include("Vandermonde.jl")
```

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

In [2]:
```
# Broj intervala
srand(123)
n=5
x=sort(rand(n+1))
y=rand(n+1)
h=x[2:end]-x[1:end-1]
d=(y[2:end]-y[1:end-1])./h
H=SymTridiagonal(2*(h[1:end-1]+h[2:end]),h[2:end-1])
```

Out[2]:
```
4×4 SymTridiagonal{Float64}:
 0.698622  0.267102    ⋅          ⋅
 0.267102  0.557011   0.0114039   ⋅
  ⋅        0.0114039  0.211786   0.094489
  ⋅         ⋅         0.094489   0.533113
```

In [3]:
```
b1=6*(d[2:end]-d[1:end-1])
s=H\b1
s=[0;s;0]
```

Out[3]:
```
6-element Array{Float64,1}:
    0.0
  155.934
 -243.764
  456.445
  -67.1929
    0.0
```

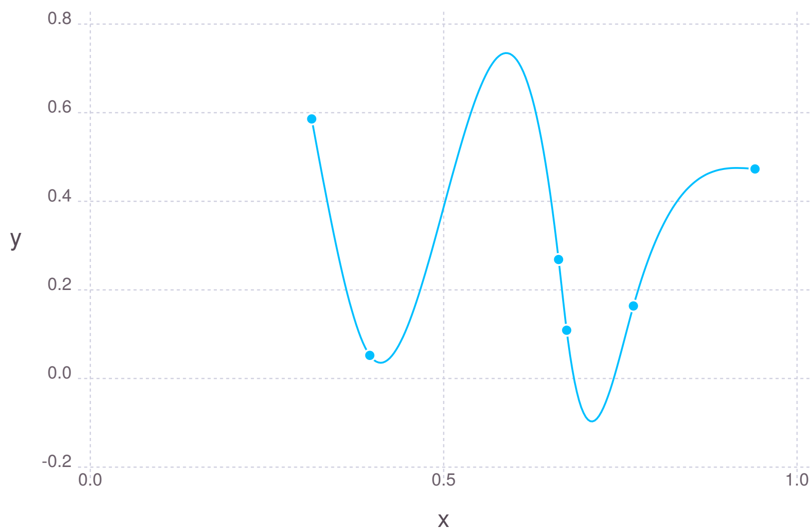
In [4]:
```
# Definirajmo polinome
b=d-(s[2:end]-s[1:end-1]).*h/6
C=Array{Any}(n)
C=[xx -> y[i]-s[i]*h[i]^2/6+b[i]*(xx-x[i])+s[i]*(x[i+1]-xx)^3/(6*h[i])
    +s[i+1]*(xx-x[i])^3/(6*h[i]) for i=1:n]
```

Out[4]:
```
5-element Array{##2#4{Int64},1}:
 #2
 #2
 #2
 #2
 #2
```

In [5]:
```
# Definirajmo točke za crtanje
lsize=200
xx=linspace(x[1],x[end],lsize)
ySpline=Array{Float64}(lsize)
for i=1:lsize
    for k=1:n
        if xx[i]<=x[k+1]
            ySpline[i]=C[k](xx[i])
            break
        end
    end
end
```

In [6]:
```
# Crtanje
plot(layer(x=x,y=y,Geom.point),layer(x=xx,y=ySpline,Geom.line))
```

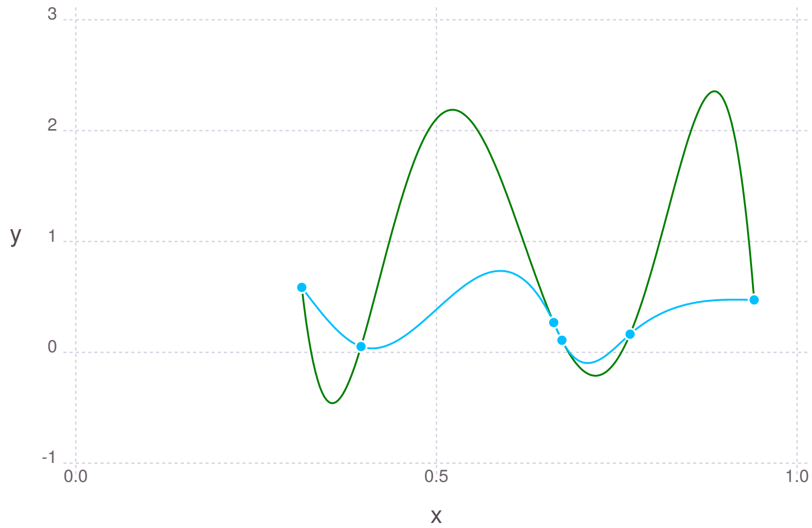
Out[6]:



Usporedimo splajn s interpolacijskim polinomom:

In [7]:
```
A=Vandermonde(x)
p=Poly(A\y)
yPoly=polyval(p,xx)
plot(layer(x=x,y=y,Geom.point),layer(x=xx,y=ySpline,Geom.line),
    layer(x=xx,y=yPoly,Geom.line,Theme(default_color=colorant"green")))
```

Out[7]:



Usporedimo splajn s interpolacijskim polinomom kada mijenjamo jednu točku:

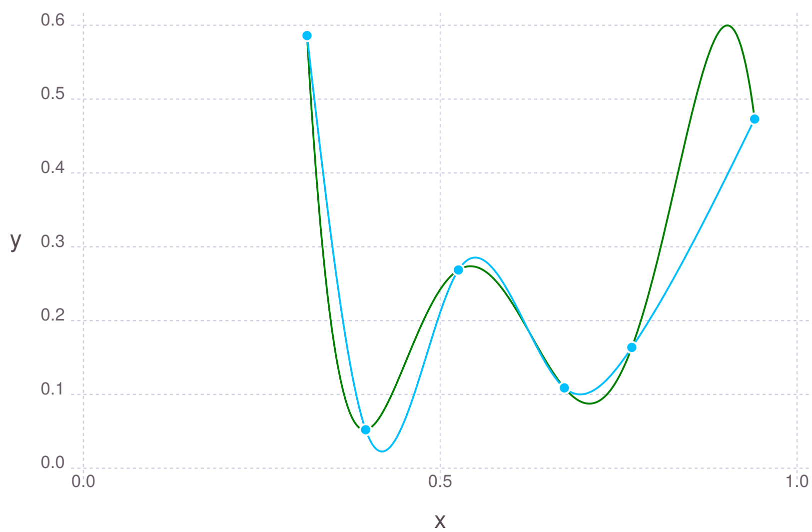
```
In [8]: using Interact
```

```
INFO: Interact.jl: using new nbwidgetsextension protocol
```

```
In [9]: # Traje duze!
        point=3
        xc=deepcopy(x)

        ySpline=Array{Float64}(lsize)
        yPoly=Array{Float64}(lsize)
        C=Array{Any}(n)

        @manipulate for xp=xc[point-1]:0.01:xc[point+1]
            xc[point]=xp
            h=xc[2:end]-xc[1:end-1]
            d=(y[2:end]-y[1:end-1])./h
            H=SymTridiagonal(2*(h[1:end-1]+h[2:end]),h[2:end-1])
            b1=6*(d[2:end]-d[1:end-1])
            s=H\b1
            s=[0;s;0]
            b=d-(s[2:end]-s[1:end-1]).*h/6
            C=[xx->y[i]-s[i]*h[i]^2/6+b[i]*(xx-xc[i])+s[i]*(xc[i+1]-xx)^3/(6*h[i])
                           +s[i+1]*(xx-xc[i])^3/(6*h[i]) for i=1:n]
            for i=1:lsize
                for k=1:n
```

```
                if xx[i]<=xc[k+1]
                    ySpline[i]=C[k](xx[i])
                    break
                end
            end
        end
        A=Vandermonde(xc)
        p=Poly(A\y)
        yPoly=polyval(p,xx)
        plot(layer(x=xc,y=y,Geom.point),layer(x=xx,y=ySpline,Geom.line),
             layer(x=xx,y=yPoly,Geom.line,Theme(default_color=colorant"green")))
    end
```

Out[9]:

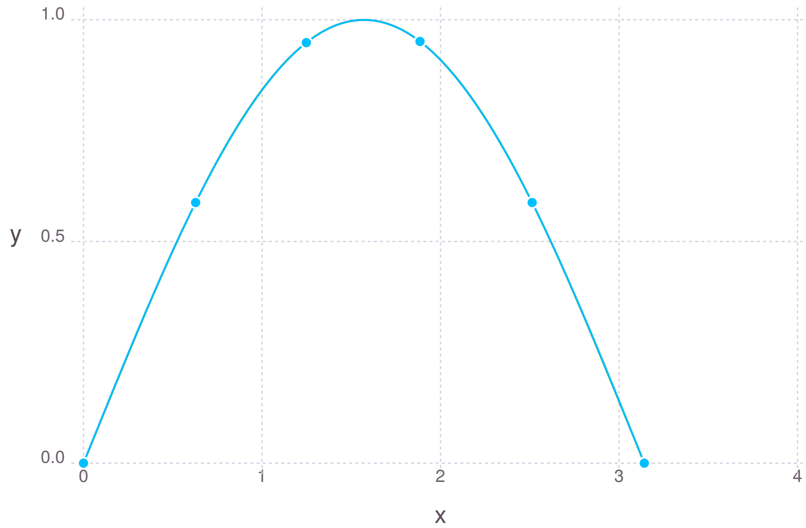


1.2 Primjer - Interpolacija funkcije $\sin(x)$

In [10]:
```
n=5
a=0
b=pi
f(x)=sin.(x)

x=collect(linspace(a,b,n+1))
y=f(x)
lsize=200
xx=collect(linspace(a,b,lsize))
```

```julia
ySpline=Array{Float64}(lsize)
yPoly=Array{Float64}(lsize)
yFun=f(xx)

point=3
xc=deepcopy(x)
yc=deepcopy(y)
@manipulate for xp=xc[point-1]:0.01:xc[point+1]
    # Splajn
    xc[point]=xp
    yc[point]=sin(xp)
    h=xc[2:end]-xc[1:end-1]
    d=(yc[2:end]-yc[1:end-1])./h
    H=SymTridiagonal(2*(h[1:end-1]+h[2:end]),h[2:end-1])
    b1=6*(d[2:end]-d[1:end-1])
    s=H\b1
    s=[0;s;0]
    b=d-(s[2:end]-s[1:end-1]).*h/6
    C=Array{Any}(n)
    C=[xx-> yc[i]-s[i]*h[i]^2/6+b[i]*(xx-xc[i])+s[i]*(xc[i+1]-xx)^3/(6*h[i])
                +s[i+1]*(xx-xc[i])^3/(6*h[i]) for i=1:n]
    for i=1:lsize
        for k=1:n
            if xx[i]<=xc[k+1]
                ySpline[i]=C[k](xx[i])
                break
            end
        end
    end

    # Polinom
    A=Vandermonde(xc)
    p=Poly(A\yc)
    yPoly=polyval(p,xx)

    plot(layer(x=xc,y=yc,Geom.point),layer(x=xx,y=ySpline,Geom.line),
         layer(x=xx,y=yPoly,Geom.line,Theme(default_color=colorant"green")))

    # Pogreske
    #=
    norm(ySpline[2:end-1]-yFun[2:end-1],Inf),
    norm((ySpline[2:end-1]-yFun[2:end-1])./yFun[2:end-1],Inf),
    norm(yPoly[2:end-1]-yFun[2:end-1],Inf),
    norm((yPoly[2:end-1]-yFun[2:end-1])./yFun[2:end-1],Inf)
    =#
end
```

Out[10]:



1.3 Primjer - Interpolacija funkcije $f(x) = 1 - |x-1|$, $x \in [0,2]$

In [11]:
```
n=10
a=0
b=2
f(x)=1-abs.(x-1)

# Ravnomjerno raspoređene točke
x=collect(linspace(a,b,n+1))
y=f(x)
lsize=200
xx=collect(linspace(a,b,lsize))
ySpline=Array{Float64}(lsize)
yPoly=Array{Float64}(lsize)
yFun=f(xx)

point=3
xc=deepcopy(x)
yc=deepcopy(y)
@manipulate for xp=xc[point-1]:0.01:xc[point+1]
    # Splajn
    xc[point]=xp
    yc[point]=sin(xp)
    h=xc[2:end]-xc[1:end-1]
    d=(yc[2:end]-yc[1:end-1])./h
    H=SymTridiagonal(2*(h[1:end-1]+h[2:end]),h[2:end-1])
```

```
    b1=6*(d[2:end]-d[1:end-1])
    s=H\b1
    s=[0;s;0]
    b=d-(s[2:end]-s[1:end-1]).*h/6
    C=Array{Any}(n)
    C=[xx-> yc[i]-s[i]*h[i]^2/6+b[i]*(xx-xc[i])+s[i]*(xc[i+1]-xx)^3/(6*h[i])
              +s[i+1]*(xx-xc[i])^3/(6*h[i]) for i=1:n]
    for i=1:lsize
        for k=1:n
            if xx[i]<=xc[k+1]
                ySpline[i]=C[k](xx[i])
                break
            end
        end
    end

    # Polinom
    A=Vandermonde(xc)
    p=Poly(A\yc)
    yPoly=polyval(p,xx)

    plot(layer(x=xc,y=yc,Geom.point),layer(x=xx,y=ySpline,Geom.line),
         layer(x=xx,y=yPoly,Geom.line,Theme(default_color=colorant"green")))
end
```

Out[11]: