Computation of integrals

Alberto ramos <alberto.ramos@maths.tcd.ie>
Thu Nov 7 07:41:35 2019

1 A simple example

we are going to compute the area of the function $f(x) = \sin x$ between 0 and 1.5. In math notation this is:

$$\int_0^{1.5} \sin x \, \mathrm{d}x \,. \tag{1}$$

We are going to examine this by computing the Lower and Upper sums for some partitions. First we need to be able to generate such partitions. This is done by the function `generate_partition`.

Code (Julia)

```julia
using PyPlot
using Printf

function generate_partition(n::Int64)

    P = sort(rand(n-1)*1.5)
    push!(P,1.5)
    pushfirst!(P,0.0)

    return P
end

println("Example partition: ", map( x -> @sprintf("%.3f", x), generate_partition(8)))
```

```
Example partition: ["0.000", "0.134", "0.271", "0.328", "0.455", "0.585", "1.179", "1.199", "1.500"]
```

Now we determine The lower and upper sums of $\sin(x)$ for a given partition

Code (Julia)

```julia
  function L(P::Array{Float64,1})

      sum::Float64 = 0.0
      for i in 1:length(P)-1
          sum = sum + sin(P[i])*(P[i+1] - P[i])
      end

      return sum
  end

  function U(P::Array{Float64,1})

      sum::Float64 = 0.0
      for i in 2:length(P)
          sum = sum + sin(P[i])*(P[i] - P[i-1])
      end
```

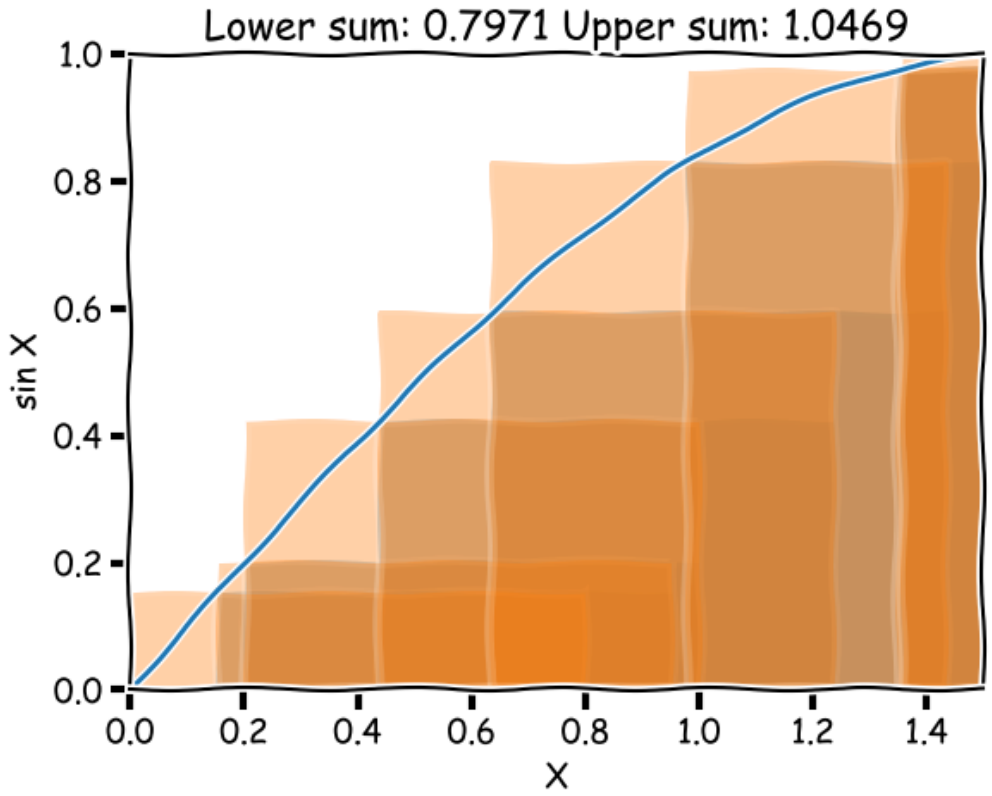
```
        return sum
    end

P = generate_partition(8)
println("Example with a partition of 8 elements", P)
println("  o Lower sums: ", L(P))
println("  o Upper sums: ", U(P))
```

```
Example with a partition of 8 elements[0.0, 0.153708, 0.202928, 0.438109, 0.634563, 0.978134, 1.
  o Lower sums: 0.7971446093651167
  o Upper sums: 1.0468901477179275
```

2 Exploring the Lower and Upper sums

Code (Julia)

```
xkcd()

function plot_prepare(P)
    xlim(0.0,1.5)
    ylim(0.0,1.0)
    xlabel("X")
    ylabel("sin X")
    tit = "Lower sum: " * @sprintf("%.4f",L(P)) * " Upper sum: " * @sprintf("%.4f",U(P))
    title(tit)
end

function plot_L(P::Array{Float64,1})
    bar(P[1:end-1],map(sin,P[1:end-1]), align="edge", alpha=0.2)
end

function plot_U(P::Array{Float64,1})
    bar(P[1:end-1],map(sin,P[2:end]), align="edge", alpha=0.2)
end

function plot_func()
    plot(0.0:0.01:1.5, map(sin, 0.0:0.01:1.5))
end


plot_prepare(P)
plot_L(P)
plot_U(P)
plot_func()
```

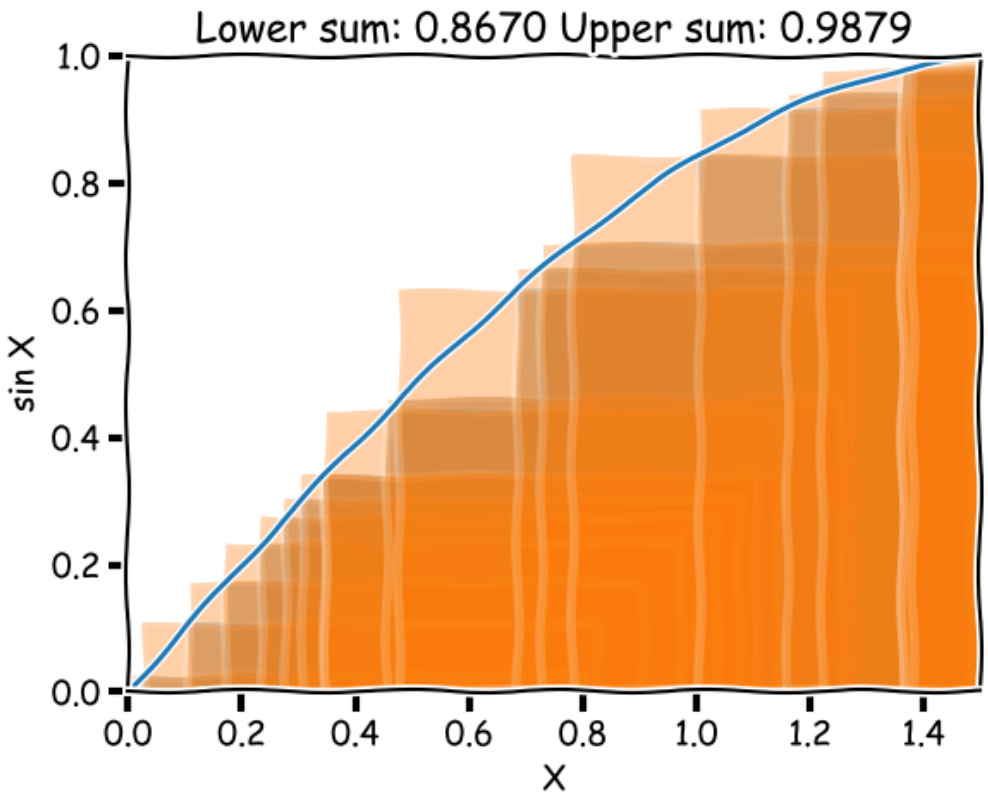



```
1-element Array{PyCall.PyObject,1}:
 PyObject <matplotlib.lines.Line2D object at 0x838093c50>
```

Now try again with a partition with many points (20)

Code (Julia)

```
P2 = generate_partition(20)
plot_prepare(P2)
plot_L(P2)
plot_U(P2)
plot_func()
```

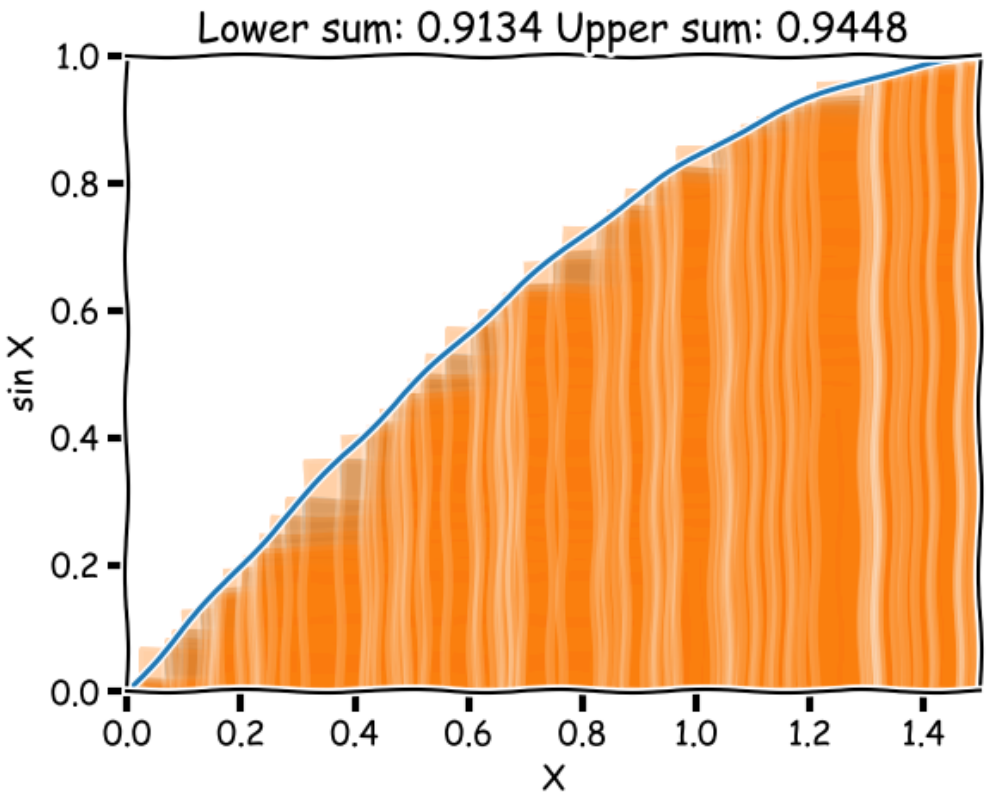



```
1-element Array{PyCall.PyObject,1}:
 PyObject <matplotlib.lines.Line2D object at 0x837ade668>
```

Or even many, many points (100)

Code (Julia)

```
P3 = generate_partition(100)
plot_prepare(P3)
plot_L(P3)
plot_U(P3)
plot_func()
```




```
1-element Array{PyCall.PyObject,1}:
 PyObject <matplotlib.lines.Line2D object at 0x838cd0dd8>
```

3 Numerical analysis

Finally one can produce many results

Code (Julia)

```julia
ns = 8
for n = 1:16
    P = generate_partition(ns)
    println("Partition size: $(@sprintf("%6d",ns)), Lower Sum: ", @sprintf("%.6f",L(P)), " Upper Su
    ns = ns * 2
end
```

```
Partition size:      8, Lower Sum: 0.804624 Upper Sum: 1.042651
Partition size:     16, Lower Sum: 0.825399 Upper Sum: 1.023551
Partition size:     32, Lower Sum: 0.885790 Upper Sum: 0.971179
Partition size:     64, Lower Sum: 0.908903 Upper Sum: 0.949090
Partition size:    128, Lower Sum: 0.917327 Upper Sum: 0.940750
Partition size:    256, Lower Sum: 0.923876 Upper Sum: 0.934623
Partition size:    512, Lower Sum: 0.926433 Upper Sum: 0.932085
Partition size:   1024, Lower Sum: 0.927734 Upper Sum: 0.930790
Partition size:   2048, Lower Sum: 0.928514 Upper Sum: 0.930011
Partition size:   4096, Lower Sum: 0.928899 Upper Sum: 0.929627
Partition size:   8192, Lower Sum: 0.929077 Upper Sum: 0.929448
Partition size:  16384, Lower Sum: 0.929171 Upper Sum: 0.929355
```

```
Partition size:  32768, Lower Sum: 0.929217 Upper Sum: 0.929309
Partition size:  65536, Lower Sum: 0.929240 Upper Sum: 0.929286
Partition size: 131072, Lower Sum: 0.929251 Upper Sum: 0.929274
Partition size: 262144, Lower Sum: 0.929257 Upper Sum: 0.929269
```