Trabalho 1

- Disciplina: Física Teórica e Computacional
- Nome: Arthur Araujo Galdino dos Santos

GitHub: https://github.com/araujoarthur/TeoricaEComputacional/tree/master/Trabalho1

[8]
```
for n ∈ 0:100
  println("n:",n, " => ",n+2, "*", n+1, "* C[",n+2,"] - C[",n,"]")
end
```

```
n:0 => 2*1* C[2] - C[0]
n:1 => 3*2* C[3] - C[1]
n:2 => 4*3* C[4] - C[2]
n:3 => 5*4* C[5] - C[3]
n:4 => 6*5* C[6] - C[4]
n:5 => 7*6* C[7] - C[5]
n:6 => 8*7* C[8] - C[6]
n:7 => 9*8* C[9] - C[7]
n:8 => 10*9* C[10] - C[8]
n:9 => 11*10* C[11] - C[9]
n:10 => 12*11* C[12] - C[10]
n:11 => 13*12* C[13] - C[11]
n:12 => 14*13* C[14] - C[12]
n:13 => 15*14* C[15] - C[13]
```

[31]
```
for n ∈ 0:10
  recurrenceRelation = string("n:", n, " => C[",n+2,"] = C[",n,"] / ", n+2 , "*", n+1)
  println(recurrenceRelation)
end
```

```
n:0 => C[2] = C[0] / 2*1
n:1 => C[3] = C[1] / 3*2
n:2 => C[4] = C[2] / 4*3
n:3 => C[5] = C[3] / 5*4
n:4 => C[6] = C[4] / 6*5
n:5 => C[7] = C[5] / 7*6
n:6 => C[8] = C[6] / 8*7
n:7 => C[9] = C[7] / 9*8
n:8 => C[10] = C[8] / 10*9
```

```
n:9 => C[11] = C[9] / 11*10
n:10 => C[12] = C[10] / 12*11
```

[40]
```
#Work Smarter, Not Harder kkkkkkkkkkkkkkkkkkkk

storedC = []
for n ∈ 0:10

  if n >= 2
    enhancedRelation = string(storedC[n-1]," / ", n+2 , "*",
n+1)
    println("[ENC] n:", n, " => C[",n+2,"] =
",enhancedRelation)
    push!(storedC, enhancedRelation)
  else
    recurrenceRelation = string( "C[",n,"] / ", n+2 , "*", n+1)
    println("[REC] n:", n, " =>  C[",n+2,"]
=",recurrenceRelation);
    push!(storedC,recurrenceRelation)
  end

end
```

```
[REC] n:0 =>  C[2] =C[0] / 2*1
[REC] n:1 =>  C[3] =C[1] / 3*2
[ENC] n:2 => C[4] = C[0] / 2*1 / 4*3
[ENC] n:3 => C[5] = C[1] / 3*2 / 5*4
[ENC] n:4 => C[6] = C[0] / 2*1 / 4*3 / 6*5
[ENC] n:5 => C[7] = C[1] / 3*2 / 5*4 / 7*6
[ENC] n:6 => C[8] = C[0] / 2*1 / 4*3 / 6*5 / 8*7
[ENC] n:7 => C[9] = C[1] / 3*2 / 5*4 / 7*6 / 9*8
[ENC] n:8 => C[10] = C[0] / 2*1 / 4*3 / 6*5 / 8*7 / 10*9
[ENC] n:9 => C[11] = C[1] / 3*2 / 5*4 / 7*6 / 9*8 / 11*10
[ENC] n:10 => C[12] = C[0] / 2*1 / 4*3 / 6*5 / 8*7 / 10*9 / 12*11
```

[42]
```
using Pkg
```

[44]
```
Pkg.add("Plots")
```

```
 Resolving package versions...
  Updating `C:\Users\arthu\.julia\environments\v1.4\Project.toml`
[no changes]
  Updating `C:\Users\arthu\.julia\environments\v1.4\Manifest.toml`
[no changes]
```

[45]
```
Pkg.add("GR")
```

```
  Resolving package versions...
   Updating `C:\Users\arthu\.julia\environments\v1.4\Project.toml`
  [28b8d3ca] + GR v0.50.1
   Updating `C:\Users\arthu\.julia\environments\v1.4\Manifest.toml`
  [no changes]
```

[47]
```
using Plots
```

```
┌ Info: Precompiling Plots [91a5bcdd-55d7-5caf-9e0b-520d859cae80]
└ @ Base loading.jl:1260
```

[48]
```
gr()
```

```
Plots.GRBackend()
```

[93]
```
c0 = 1
c1 = 1
_solution(x) = (c0 * cosh(x)) + (c1 * sinh(x))
```

```
_solution (generic function with 1 method)
```

[81]
```
?plot
```

```
search: plot plot! plotly plot3d Plots plot3d! plotlyjs plotattr
plotarea
```

The main plot command. Use `plot` to create a new plot object, and `plot!` to add to an existing one:

```
    plot(args...; kw...)                # creates a new plot window, an
    plot!(args...; kw...)               # adds to the `current`
    plot!(plotobj, args...; kw...)      # adds to the plot `plotobj`
```

There are lots of ways to pass in data, and lots of keyword arguments... just try it and it will likely work as expected. When you pass in matrices, it splits by columns. To see the list of available attributes, use the `plotattr([attr])` function, where `attr` is the symbol `:Series:`, `:Subplot:`, `:Plot` or `:Axis`. Pass any attribute to `plotattr` as a

[104]
```
c0 = 1
c1 = 1
```

```
plot(_solution, -2*pi:2*pi, label = "-2π ≤ x ≤ 2π", line=
(:black, 0.5, 3, :dot), title = "Grafico da Solução - Itens
3.A, 3.B, 3.C", ylabel ="y(x)", xlabel="x")
plot!(_solution, -pi:pi, line=(:red, 0.5, 3, :dash), label = "-
π ≤ x ≤ π")
plot!(_solution, -1:1, line=(:green, 0.5, 6, :solid), label =
"-1 ≤ x ≤ 1",)
savefig("PLOT - A B C.png")
```

[102]
```
c1 = 2
c0 = 2*c1
plot(_solution, -2*pi:2*pi, label = "-2π ≤ x ≤ 2π", line=
(:black, 0.5, 3, :dot), title = "Grafico da Solução - Itens
Item 3.D.I", ylabel ="y(x)", xlabel="x")
savefig("PLOT - D I.png")
```

[103]
```
c0 = 2
c1 = 2*c0
plot(_solution, -2*pi:2*pi, label = "-2π ≤ x ≤ 2π", line=
(:black, 0.5, 3, :dot), title = "Grafico da Solução - Item
3.D.II", ylabel ="y(x)", xlabel="x")
savefig("PLOT - D II.png")
```

[ ]