

```

1 include("util.jl")
2 include("../src/CGPO.jl")
3
4 function PerformCGFor(
5     A::AbstractMatrix,
6     b::AbstractVecOrMat;
7     epsilon=1e-2,
8     exact::Bool=true,
9     partial_ortho=nothing,
10 )
11     cg = CGPO(A, b)
12
13     if exact
14
15     else
16         if partial_ortho === nothing
17             cg |> TurnOffReorthogonalize!
18         else
19             StorageLimit!(cg, partial_ortho)
20         end
21     end
22
23     ẋ = A\b
24     ê = ẋ - cg.x
25     êAê = dot(ê, A*ê)
26     E = Vector{Float64}()
27     push!(E, 1)
28     RelErr = 1
29     while RelErr > epsilon
30         cg()
31         e = ẋ - cg.x
32         RelErr = (dot(e, A*e)/êAê)|>sqrt
33         push!(E, RelErr)
34     end
35     return E
36 end
37
38 E = PerformCGFor(Diagonal(rand(10)), rand(10))
39
40 function PerformExperiment1()
41     N = 256
42     A = GetNastyPSDMatrix(N, 0.9)
43     b = rand(N)
44     A = convert(Matrix{Float16}, A)
45     b = convert(Vector{Float16}, b)
46     # TODO: Make the plot distinguishable without colors.
47
48     #

```

```

=====
49  # The exact computations
50  #
=====

51
52  RelErr = PerformCGFor(A, b, epsilon=1e-3, exact=true)
53  k = length(RelErr)
54  fig1 = plot(
55      log10.(RelErr),
56      label="Relative Energy (exact)",
57      legend=:bottomleft
58  )
59
60  #
=====

61  # No-Orthogonalizations
62  #
=====

63
64  RelErr = PerformCGFor(A, b, exact=false, epsilon=1e-3)
65  k = length(RelErr)
66  plot!(
67      fig1,
68      log10.(RelErr),
69      label="Relative Energy (floats)",
70      linestyle=:dash
71  )
72
73  #
=====

74  # Theoretical Bounds
75  #
=====

76  ErrorsBound = [TheoreticalErrorBound(A, idx) for idx in 1: k]
77  plot!(
78      fig1,
79      log10.(ErrorsBound),
80      label="Theoretical Bound (exact)",
81      xlabel="iteration count",
82      ylabel="relative error energy norm.",
83      linestyle=:dot
84  )
85
86  #

```

```

=====
87  # Floating Points Partially Orthogonalized
88  #
=====

89  RelErr = PerformCGFor(A, b, exact=false, epsilon=1e-3,
90  partial_ortho=div(N, 8))
91  k = length(RelErr)
92  plot!(
93      fig1,
94      log10.(RelErr),
95      label="Relative Energy (partial)",
96      legend=:bottomleft,
97      linestyle=:dashdot
98  )
99  display(fig1)
100  SaveFigToCurrentScriptDir(fig1, "fig1.png")
101
102
103  return end
104
105  PerformExperiment1()
106
107

```