Linear Algebra 2

Benchmarking

In [1]:

```
xs = 0:0.01:pi
```

Out[1]:

```
0.0:0.01:3.14
```

In [7]:

```
@time ys = sin.(xs)
```

```
  0.000016 seconds (2 allocations: 2.688 KiB)
```

Out[7]:

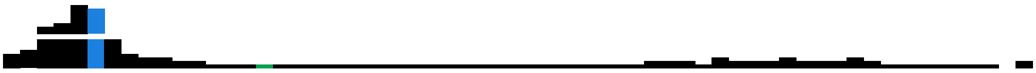
```
315-element Vector{Float64}:
 0.0
 0.009999833334166664
 0.01999866669333308
 0.02999550020249566
 0.03998933418663416
 0.04997916927067833
 0.059964006479444595
 0.06994284733753277
 0.0799146939691727
 0.08987854919801104
 0.09983341664682815
 0.10977830083717481
 0.11971220728891936
 ⋮
 0.11136118868664958
 0.10141798631660187
 0.09146464223243675
 0.08150215176026912
 0.07153151114084326
 0.06155371742991315
 0.05156976839853464
 0.04158066243329049
 0.031587398436453896
 0.02159097572609596
 0.011592393936158275
 0.0015926529164868282
```

In [4]:

```
using BenchmarkTools
```

In [8]:
```
@benchmark ys = sin.(xs)
```

Out[8]:
```
BenchmarkTools.Trial: 10000 samples with 8 evaluations.
 Range (min … max):  3.537 μs … 277.045 μs  ┊ GC (min … max): 0.0
0% … 95.36%
 Time  (median):     3.713 μs               ┊ GC (median):    0.0
0%
 Time  (mean ± σ):   4.035 μs ±   3.740 μs  ┊ GC (mean ± σ):  1.2
6% ±  1.35%
```



```
  3.54 μs          Histogram: frequency by time        5.5 μs <

 Memory estimate: 2.69 KiB, allocs estimate: 2.
```

Sparse Matrices

In [10]:
```
using LinearAlgebra
using SparseArrays
```

In [13]:
```
A = sprand(10, 10, 0.2)
```

Out[13]:
```
10×10 SparseMatrixCSC{Float64, Int64} with 18 stored entries:
  ⋅          ⋅         ⋅   0.00586887  …  0.937074   ⋅    ⋅
0.820368
 0.0285838   ⋅         ⋅    ⋅              ⋅         ⋅   0.382854
0.504618
 0.0176372  0.520039   ⋅    ⋅              ⋅         ⋅    ⋅
⋅
  ⋅          ⋅         ⋅    ⋅              ⋅         ⋅   0.251513
⋅
 0.488654    ⋅         ⋅   0.556611        ⋅         ⋅    ⋅
⋅
  ⋅          ⋅         ⋅    ⋅          …   ⋅         ⋅    ⋅
⋅
  ⋅          ⋅         ⋅    ⋅              ⋅         ⋅    ⋅
⋅
  ⋅          ⋅         ⋅    ⋅              ⋅         ⋅    ⋅
0.962364
  ⋅          ⋅         ⋅    ⋅              ⋅         ⋅    ⋅
⋅
 0.680894    ⋅         ⋅    ⋅              ⋅         ⋅    ⋅
⋅
```

In [16]:
```
pinv(Matrix(A))
```

Out[16]:
```
10×10 Matrix{Float64}:
  0.0577932     0.221219      0.164213     …  -0.231032     0.0
0.862024
 -0.0216246     1.07853       1.86149         -0.522485     0.0
0.128425
  1.62602e-16  -4.06114e-16  -1.34701e-16     -1.10676e-16  0.0
2.56886e-17
 -0.0507372    -0.19421      -0.144164         0.202825     0.0  -
0.75678
 -0.0829747    -0.317608     -0.235763         0.331697     0.0
0.870955
  0.016606     -0.917114      0.0471841    …   0.447837     0.0  -
0.133139
  1.05344      -0.789497     -0.038966        -0.468424     0.0
0.182963
  0.0           0.0           0.0              0.0          0.0
0.0
 -0.0254399     1.40499      -0.0722846       -0.686072     0.0
0.203964
  0.0160276     0.9032        0.0455406        0.53361      0.0  -
0.203577
```

In [17]:
```
spzeros(10, 10)
```

Out[17]:
```
10×10 SparseMatrixCSC{Float64, Int64} with 0 stored entries:
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
  .    .    .    .    .    .    .    .    .    .
```

In [19]:
```
sparse(Diagonal(1:10))
```

Out[19]:
```
10×10 SparseMatrixCSC{Int64, Int64} with 10 stored entries:
 1  .  .  .  .  .  .  .  .   .
 .  2  .  .  .  .  .  .  .   .
 .  .  3  .  .  .  .  .  .   .
 .  .  .  4  .  .  .  .  .   .
 .  .  .  .  5  .  .  .  .   .
 .  .  .  .  .  6  .  .  .   .
 .  .  .  .  .  .  7  .  .   .
 .  .  .  .  .  .  .  8  .   .
 .  .  .  .  .  .  .  .  9   .
 .  .  .  .  .  .  .  .  .  10
```

In [24]:

```
Bidiagonal(1:10, 1:9, :U)
```

Out[24]:

```
10×10 Bidiagonal{Int64, UnitRange{Int64}}:
 1  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅   ⋅
 ⋅  2  2  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅   ⋅
 ⋅  ⋅  3  3  ⋅  ⋅  ⋅  ⋅  ⋅   ⋅
 ⋅  ⋅  ⋅  4  4  ⋅  ⋅  ⋅  ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  5  5  ⋅  ⋅  ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  6  6  ⋅  ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  7  7  ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  8  8   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  9   9
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  10
```

In [29]:

```
Tridiagonal(2:10, 1:10, 3:11)
```

Out[29]:

```
10×10 Tridiagonal{Int64, UnitRange{Int64}}:
 1  3  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅   ⋅   ⋅
 2  2  4  ⋅  ⋅  ⋅  ⋅  ⋅   ⋅   ⋅
 ⋅  3  3  5  ⋅  ⋅  ⋅  ⋅   ⋅   ⋅
 ⋅  ⋅  4  4  6  ⋅  ⋅  ⋅   ⋅   ⋅
 ⋅  ⋅  ⋅  5  5  7  ⋅  ⋅   ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  6  6  8  ⋅   ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  7  7  9   ⋅   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  8  8  10   ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  9   9  11
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  10  10
```

In [32]:

```
sprandn(100, 100, 0.1)
```

Out[32]:

```
100×100 SparseMatrixCSC{Float64, Int64} with 1019 stored entries:
```