Final exam

You should be able to solve this exam using only the course notes and previous assignments, but you are welcome to consult any resource you wish except for people outside the course staff. Provide attribution (a citation or link) for any ideas you get. Your final write-up should be your own.

You should do problem 1, and any four out of the remaining five problems. Please indicate which four of the remaining five problems you want graded (we will *not* grade all five and take the best).

1 Snacks

1. (1 pt) Complete the course evaluation, if you have not already done so!
2. (1 pt) Give an example of a continuous function f with a zero in $[0, 1]$ where bisection starting from the interval $[0, 1]$ will fail. Explain.
3. (1 pt) Give an example 1D optimization problem where Newton iteration converges linearly (not quadratically) to a minimum. Explain.
4. (1 pt) Given `F = cholesky(A)` for spd $A \in \mathbb{R}^{n\times n}$, give a Julia code fragment to evaluate $e_n^T A^{-1} e_n = (A^{-1})_{nn}$ in $O(1)$ time.
5. (1 pt) Using Newton iteration, solve the simultaneous equations $x^2 + xy^2 = 9$ and $3x^2y - y^3 = 4$. You may use the initial guess $(x, y) = (1, 1)$. Report your results to at least ten digits.

Answer

```julia
let
    A = [11.5   6.0    1.0   6.5    3.0;
          6.0  16.5    8.0   4.75   7.5;
          1.0   8.0   14.5   7.5    5.5;
          6.5   4.75   7.5  18.0    5.25;
          3.0   7.5    5.5   5.25  13.0]
    F = cholesky(A)
    invAnn_ref = inv(A)[end,end]
    invAnn_fast = 0.0       # TODO: Replace
    abs(invAnn_ref-invAnn_fast)/abs(invAnn_ref)
end
```

```
let
    xy = [1.0; 1.0]
    # TODO: Fill in Newton iteration for 1.5
    xy
end
```

2 Interesting iterations

Consider the function $w(s)$ defined by the scalar equation

$$we^w = s.$$

1. (2 point) Argue that the equation has a unique solution for any $s > 0$.
2. (2 points) By manipulating the equations and some clever uses of Taylor's theorem with remainder, we get upper and lower bounds of $\log((1 + \sqrt{1 + 4s})/2) \leq w \leq \log(1 + s)$. For $s \ll 1$, both these expressions have large relative errors. Rewrite each expression accurately using the `log1p`, which evaluates $\log(1 + z)$ accurately when $z \ll 1$.
3. (2 points) Consider the iteration $w_{k+1} = s \exp(-w_k)$. Derive an error iteration and analyze its convergence. For what range of s values does the iteration converge?
4. (2 points) Consider the fixed point iteration $w_{k+1} = G(w_k)$ where $G(w) = s(1 + w)/(s + e^w)$. The solution to $we^w = s$ is a fixed point of G (you do not need to show this). Argue that this iteration converges quadratically from good enough initial guesses. In fact, it converges for any initial guess in $[0, \infty)$; you do not have to prove this.
5. (2 points) Complete the function below to evaluate $w(s)$ and $w'(s)$ for a given s. Report values of $w(1)$ and $w'(1)$ to at least ten digits.

Answer

```
function bracket_w(s)
    # TODO: Rewrite this to avoid error for small s
    log((1+sqrt(1+4*s))/2), log(1+s)
end

function compute_w(s)
    l, u = bracket_w(s)
```

```
    w = (l+u)/2
    for k = 1:10
        wprev = w
        w = s*(1+w)/(s+exp(w))
        if abs(wprev-w) < 1e-12*w
            break
        end
    end
    # TODO: Rewrite to also return w'
    dw = 0.0
    w, dw
end
```

We provide sanity checks for the bracketing behavior at small s and for the correctness of the derivative calculation.

3 Quirky quadratics

Consider the almost-quadratic optimization problem of minimizing (or finding a stationary point of)

$$\phi(x) = \frac{1}{2}x^T Hx - x^T d + g(c^T x).$$

Here $H \in \mathbb{R}^{n\times n}$, $x, c, d \in \mathbb{R}^n$, and $g : \mathbb{R} \to \mathbb{R}$.

1. (2 points) Write an expression for $\nabla\phi$.
2. (2 points) Show that $x = H^{-1}(d + \gamma c)$ for some γ
3. (2 points) Show $\gamma + g'(\alpha + \gamma\beta) = 0$ for some α and β.
4. (4 points) Using part 3, complete the Newton solver below to compute γ and hence to compute a stationary point of ϕ. For the given test case, report the computed γ to at least ten digits. For full credit, you should not factor H more than once, and you should use a minimal number of linear solves with H.

Answer

```
function p3newton(γ, H, c, d, dg, Hg;
                  rtol=1e-12, monitor=(γ, r)->nothing)
```

```
    # TODO: Newton to solve for γ to resid < rtol (also form x)
    γ, x
end

let
    H = [1.5  6.0   1.0  6.5   3.0;
         6.0  6.5   8.0  4.75  7.5;
         1.0  8.0   4.5  7.5   5.5;
         6.5  4.75  7.5  8.0   5.25;
         3.0  7.5   5.5  5.25  3.0]
    c = [10.0; 1.5; 8.5; 2.5; 8.5]
    d = [ 1.5; 8.0; 6.5; 4.0; 1.5]

    g(x) = -x^3
    dg(x) = -3*x^2
    Hg(x) = -6*x

    rnorms = []
    γ, x = p3newton(0.0, H, c, d, dg, Hg,
                    monitor=(x,r)->push!(rnorms, abs(r)))
    # RECOMMENDED CHECKS: Form the residual + plot quadratic conv
end
```

4 Block GS

Consider the linear system $Lz = h$ with block structure

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

where $A \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{m \times m}$ are invertible. The block Gauss-Seidel iteration is

$$\begin{aligned} Ax^{k+1} &= f - By^k \\ Dy^{k+1} &= g - Cx^{k+1}. \end{aligned}$$

1. (2 points) Write the splitting $L = M - N$ associated with this iteration.
2. (2 points) Write $R = M^{-1}N$ in terms of the component matrices A, B, C, D.

3. (3 points) Argue that $\rho(R) = \rho(D^{-1}CA^{-1}B)$ where ρ denotes the spectral radius. You may use without argument that the eigenvalues of a block triangular matrix are equal to the eigenvalues of its diagonal blocks.
4. (3 points) Now consider the *nonlinear* block Gauss-Seidel iteration $Ax^{k+1} = u(y^k)$ and $Dy^{k+1} = w(x^{k+1})$ where $u : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $w : \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfy $\|u'\|_2 \leq M_u$ and $\|w'\|_2 \leq M_w$ everywhere. For convenience, we can also write $y^{k+1} = D^{-1}w(A^{-1}u(y^k))$ (we can similarly write an iteration satisfied by the x^{k+1} and x^k alone). Argue that the y_k iteration converges if $M_u M_w < \sigma_{min}(D)\sigma_{\min}(A)$.

Answer

5 Lolling linkages

We consider an optimization problem inspired by the equilibrium behavior of a chain of four unit-length rigid beams connected by pivot joints, where the position of the end beams is constrained. Leaving aside the physical model, we have the constrained optimization problem

$$\text{minimize } 3\sin(\theta_1) + \sin(\theta_2) \text{ s.t. } \cos(\theta_1) + \cos(\theta_2) = 2 - \delta.$$

1. (2 points) Argue from Taylor expansion of $\cos(\theta) = \sqrt{1 - \sin(\theta)^2}$ that $\cos(\theta) \approx 1 - \frac{1}{2}\sin(\theta)^2$. Hence for small δ, the constraint is approximately $\sin(\theta_1)^2 + \sin(\theta_2)^2 = 2\delta$. For what values of $\sin(\theta_1)$ and $\sin(\theta_2)$ satisfying this constraint do we minimize the objective?
2. (4 points) Starting from an initial guess derived from the estimate in the previous problem, write a Newton iteration to find the optimum angles. What are the angles for $\delta = 10^{-2}$, $\delta = 10^{-1}$, and $\delta = 0.25$? Give a semilog convergence plot for the $\delta = 0.25$ case.
3. (4 points) Assuming the above constrained optimization problem is satisfied, write a Newton solver to find δ given the additional equation $y = \sin(\theta_1) + \sin(\theta_2)$.

Answer

```
function linkage_θ(δ; rtol=1e-10, monitor=(θ, μ, rnorm)->nothing)
    # TODO: Fill in Newton iteration to compute angles for given δ
```

```
    #   Return θ vector and a residual norm for the tester
end

function linkage_δ(y; rtol=1e-10, monitor=(θ, μ, δ, rnorm)->nothing)
    # TODO: Fill in Newton iteration to compute δ for given y
    #   Return δ and a residual norm for the tester
end
```

We provide a consistency check for part 3.

```
let
    δref = 0.01
    θ, rnormθ = linkage_θ(δref)
    δ, rnormδ = linkage_δ(sin(θ[1]) + sin(θ[2]))
    relerr_δ = (δref-δ)/δref
md"""
Relative error in recovering δ: $(relerr_δ)
"""
end
```

We also provide the code to report the numbers and plots that we want!

```
let
    resids = []
    monitor(θ, μ, rnorm) = push!(resids, rnorm)
    θ01, rnorm01 = linkage_θ(0.01)
    θ10, rnorm10 = linkage_θ(0.10)
    θ25, rnorm25 = linkage_θ(0.25, monitor=monitor)
    p1 = plot(resids, xlabel="k", ylabel="resid norm", yscale=:log10)
md"""
$p1

Converged angles:

- δ=0.01: $(θ01[1]), $(θ01[2])
- δ=0.10: $(θ10[1]), $(θ10[2])
- δ=0.25: $(θ25[1]), $(θ25[2])
"""
end
```

6 Double trouble

Consider the matrix-valued function $G(s) = A + sB$ where $G : [0, 1] \to \mathbb{R}^{n\times n}$. We give a specific case below. In our example, as s moves from zero to one, a pair of real eigenvalues "collide" and become a complex conjugate pair. We are interested in locating this. Your task is to complete several analysis codes that can help.

1. (3 points) Write a bisection routine to approximately compute the s where we go from all real eigenvalues to some complex eigenvalues. Resolve s to at least three digits (bracketing interval of length 10^{-3}).
2. (3 points) Complete the pseudo-arclength continuation code, trace $\lambda(\gamma)$ vs $s(\gamma)$ for $(v(\gamma), \lambda(\gamma)s(\gamma))$ given by $G(s)v = \lambda v$ and $v^T v = 1$. Plot the resulting curve. What is the maximum value of s you see?
3. (4 points) Write a Gauss-Newton iteration to solve the nonlinear equations $(G(s) - \lambda I)^2 V = 0$ and $V_0^T V = I$, which characterize a point at which we have a double eigenvalue. The unknowns in this problem are s, λ, and $V \in \mathbb{R}^{n\times 2}$; note that we have $2n + 4$ equations in $2n+2$ unknowns. Nevertheless, this overdetermined system is solvable, and Gauss-Newton should converge to it quadratically. Demonstrate quadratic convergence with a convergence plot. Use an initial guess of $s = 0.25$, $\lambda = 3$, and $V = V_0$ computed by the previous equation.

Answer

```
function p6bisection(G)
    g(s) = all(isreal.(eigvals(G(s))))
    a, b = 0.0, 1.0
    a, b
end

function p6continuation(G, A, B, λ0)
    n = size(A)[1]

    # Provided code to get starting point
    FA = eigen(A)
    k = findmin(abs.(FA.values.-λ0))[2]
    v0 = FA.vectors[:,k]
    λ0 = FA.values[k]
```

```
    s0 = 0.0

    # Packed starting point and initial reference direction
    vλs = [v0; λ0; s0]
    tprev = [zeros(n+1); 1.0]

    # Function whose zeros are of interest
    R(v, λ, s) = [G(s)*v-λ*v; v'*v-1]
    R(vλs) = R(vλs[1:n], vλs[n+1], vλs[n+2])

    # Set up storage for results
    srecord = Vector{Float64}([])
    λrecord = Vector{Float64}([])

    for k = 0:100

        # TODO: Compute unit tangent vector t with dot(t, tprev) > 0

        # TODO: Starting from predictor vλs + h*t, correct back to curve
        #   with three Newton steps (take h = 5e-2)

        # Stop if we've doubled back and s is going negative
        if vλs[end] < 0
            break
        end

        # Record s and λ
        push!(λrecord, vλs[n+1])
        push!(srecord, vλs[n+2])

    end

    srecord, λrecord
end

function p6gnsolver(G, A, B, V0, s0, λ0; rtol=1e-10,
                    monitor=(V, s, λ, rnorm)->nothing)
    n = size(V0)[1]
```

```
    V = copy(V0)
    s = s0
    λ = λ0

    # TODO: Run G-N to find the double eigenvalue and an associated
    #       invariant subspace basis V
    V, s, λ
end

function p6subspace(G, s, λ)
    F = G(s)-λ*I
    V0 = qr(randn(4,2)).Q[:,1:2]
    for k = 1:10
        V0 = qr(F\V0).Q[:,1:2]
    end
    V0
end

let
    A = [ 3.0  6.0  2.0  1.0;
          6.0  3.0  2.5  6.5;
          2.0  2.5  8.0  4.0;
          1.0  6.5  4.0  4.0]
    B = [ 1.5  3.0  1.0  5.0;
          5.0  9.0  8.0  8.0;
          9.0  9.0  0.5  7.0;
          2.0  7.0  9.0  8.0]
    G(s) = A+s*B

    # Estimate transition point via bisection on g
    sa, sb = p6bisection(G)

    # Pseudo-arclength continuation
    srecord, λrecord = p6continuation(G, A, B, 2.0)
    smax = maximum(srecord)
    p1 = plot(srecord, λrecord, xlabel="s", ylabel="λ",
              title="Results of continuation")
```

```
    # Estimate starting point with a shift-invert subspace iteration step
    s0 = 0.25
    λ0 = 3.0
    V0 = p6subspace(G, s0, λ0)
    resids = []
    Vc, sc, λc = p6gnsolver(G, A, B, V0, s0, λ0,
                            monitor=(V,s,λ,rnorm)->push!(resids, rnorm))
    p2 = plot(resids, yscale=:log10)

md"""
- Bisection final interval: [$sa, $sb]
- Maximum $s$ in continuation: $smax
- Gauss-Newton converged to s = $sc, λ = $λc

$p1

$p2
"""
end
```