Fall, 2013 CIS 515

Fundamentals of Linear Algebra and Optimization
Jean Gallier

Project 5

November 27, 2013; Due December 10, 2013

Problem B1(60 pts). (1) Implement the Gram-Schmidt orthonormalization procedure and the modified Gram-Schmidt procedure. You may use the pseudo-code showed in (2).

(2) Implement the method to compute the QR decomposition of an invertible matrix. You may use the following pseudo-code:

function $qr(A$: matrix): $[Q, R]$ pair of matrices
  begin
    $n = \dim(A)$;
    $R = 0$; (the zero matrix)
    $Q1(:, 1) = A(:, 1)$;
    $R(1, 1) = \mathrm{sqrt}(Q1(:, 1)^\top \cdot Q1(:, 1))$;
    $Q(:, 1) = Q1(:, 1)/R(1, 1)$;
    for $k := 1$ **to** $n - 1$ **do**
      $w = A(:, k + 1)$;
      for $i := 1$ **to** k **do**
        $R(i, k + 1) = A(:, k + 1)^\top \cdot Q(:, i)$;
        $w = w - R(i, k + 1)Q(:, i)$;
      endfor;
    $Q1(:, k + 1) = w$;
    $R(k + 1, k + 1) = \mathrm{sqrt}(Q1(:, k + 1)^\top \cdot Q1(:, k + 1))$;
    $Q(:, k + 1) = Q1(:, k + 1)/R(k + 1, k + 1)$;
    endfor;
  end

Test it on various matrices.

(3) Given any invertible matrix A, define the sequences A_k, Q_k, R_k as follows:

$$\begin{aligned} A_1 &= A \\ Q_k R_k &= A_k \\ A_{k+1} &= R_k Q_k \end{aligned}$$

for all $k \geq 1$, where in the second equation, $Q_k R_k$ is the QR decomposition of A_k given by part (2).

Run the above procedure for various values of k (50, 100, ...) and various real matrices A, in particular some symmetric matrices; also run the `Matlab` command `eig` on A_k, and compare the diagonal entries of A_k with the eigenvalues given by $\texttt{eig}(A_k)$.

What do you observe? How do you explain this behavior?

Problem B2(40 pts). Refer to Problem B4 of HW6. Write a `Matlab` program to compute a logarithm B of a rotation matrix $R \in \mathbf{SO}(3)$, namely a skew-symmetric matrix B such that $e^B = R$, using the method described in Problem B4. In particular, compute a log B of R such that the angle θ associated with B satisfies the condition $0 \leq \theta \leq \pi$. If $\mathrm{tr}(R) \neq -1$, then $\theta < \pi$ and the matrix B associated with θ is uniquely determined. Otherwise $\theta = \pi$ and B is determined up to sign.

Problem B3(210 pts). Refer to Problem B5 of HW6. Similitudes can be used to describe certain deformations (or flows) of a deformable body $\mathcal{B}_t$ in 3D. Given some initial shape $\mathcal{B}$ in $\mathbb{R}^3$ (for example, a sphere, a cube, etc.), a deformation of $\mathcal{B}$ is given by a piecewise differentiable curve

$$\mathcal{D} \colon [0, T] \to \mathbf{SIM}(3),$$

where each $\mathcal{D}(t)$ is a similitude (for some $T > 0$). The deformed body $\mathcal{B}_t$ at time t is given by

$$\mathcal{B}_t = \mathcal{D}(t)(\mathcal{B}),$$

where $\mathcal{D}(t) \in \mathbf{SIM}(3)$ is a similitude.

The surjectivity of the exponential map $\exp \colon \mathfrak{sim}(3) \to \mathbf{SIM}(3)$ implies that there is a map $\log \colon \mathbf{SIM}(3) \to \mathfrak{sim}(3)$, although it is multivalued. The exponential map and the log "function" allows us to work in the simpler (noncurved) Euclidean space $\mathfrak{sim}(3)$ (which has dimension 7).

For instance, given two similitudes $A_1, A_2 \in \mathbf{SIM}(3)$ specifying the shape of $\mathcal{B}$ at two different times, we can compute $\log(A_1)$ and $\log(A_2)$, which are just elements of the Euclidean space $\mathfrak{sim}(3)$, form the linear interpolant $(1-t)\log(A_1) + t\log(A_2)$, and then apply the exponential map to get an interpolating deformation

$$t \mapsto e^{(1-t)\log(A_1) + t\log(A_2)}, \quad t \in [0, 1].$$

Also, given a sequence of "snapshots" of the deformable body $\mathcal{B}$, say $A_0, A_1, \ldots, A_m$, where each is A_i is a similitude, we can try to find an interpolating deformation (a curve in $\mathbf{SIM}(3)$) by finding a simpler curve $t \mapsto C(t)$ in $\mathfrak{sim}(3)$ (say, a B-spline) interpolating $\log A_1, \log A_1, \ldots, \log A_m$. Then, the curve $t \mapsto e^{C(t)}$ yields a deformation in $\mathbf{SIM}(3)$ interpolating $A_0, A_1, \ldots, A_m$.

(1) **(60 pts).** Write a program interpolating between two deformations, using the formulae found in Problems B4 and B5 of HW6 (not the built-in `Matlab` functions!).

(2) **(150 pts).** Write a program using your cubic spline interpolation program from the first project, to interpolate a sequence of deformations given by similitudes $A_0, A_1, \ldots, A_m$ in $\mathbf{SIM}(3)$. Use the formulae found in Problems B4 and B5 of HW6 (not the built-in `Matlab` functions!).

TOTAL: 410 points.