Notation

Unit 1: Intro

- x, y, z are data inputs/outputs
- A is a matrix (I for identity), b is the right hand side (y is used when the right hand side is the data)
- $i = 1, m$ subscript enumerates data (and thus rows of a matrix A)
- f is function of the data
- $\hat{x}$, $\hat{y}$, $\hat{z}$, $\hat{f}$, $\hat{\varphi}$ are inference/approximation of same variables or functions
- c represents unknown parameters to characterize functions
- $k = 1, n$ subscript enumerates c (and thus columns of a matrix A)
- a_k is column of A
- Σ_k is the sum over all k, $\Pi_{i \neq k}$ is the product over all i not equal to k
- <u>Quadratic Formula slide</u>: uses standard notation for the quadratic formula
- ϕ are basis functions
- θ are pose parameters, φ represents all vertex positions of the cloth mesh
- S are the skinned vertex positions of the body mesh, D is the displacement from the body mesh to the cloth mesh
- u, v are the 2D texture space coordinate system, n is the (unit) normal direction
- I is 2D RGB image data, ψ interpolates RGB values and converts them to a 3D displacement

Unit 2: Linear Systems

- R^n is an n dimensional Cartesian space (e.g. R^1, R^2, R^3)
- a_{ik} is the element in row i and column k of A
- A^T is the transpose of matrix A, and A^{-1} is its inverse
- $\det A$ is the determinant of A
- ∃ is ”there exists”, and ∀ is ”for all”
- $\hat{e}_i$ are the standard basis vectors, with a 1 in the i-th entry (and 0’s elsewhere)
- Gaussian Elimination slides m_{ik} special column, M_{ik}, L_{ik} elimination matrices
- I_{mxm} is a size mxm identity matrix
- U upper triangular matrix, L lower triangular matrix
- $\hat{c}$ transformed version of c
- P permutation matrix (with it own special notation)

Unit 3: Understanding Matrices

- λ eigenvalue (scalar)
- v eigenvector, u right eigenvector (both column vectors)
- α is a scalar
- $i = \sqrt{-1}$ when dealing with complex numbers
- $*$ superscript indicates a complex conjugate (for imaginary numbers)
- $\hat{b}, \tilde{b}, \hat{c}$ perturbed or transformed b, c
- $\hat{A}^{-1}, \hat{I}$ approximate versions of A^{-1}, I
- U, V orthogonal (for SVD)
- u_k, v_k are columns of U, V
- Σ diagonal (not necessarily square, potentially has zeros on the diagonal)
- σ_k singular values (diagonal entries of Σ)

Unit 4: Special Matrices

- v, u column vectors
- $u \cdot v$ or $< u, v >$ is the inner product (or dot product) between u and v
- $< u, v >_A$ is the A weighted inner product
- Λ is a diagonal matrix of eigenvalues
- l_{ik} is an element of L
- $\hat{A}$ is an approximation of A

Unit 5: Iterative Solvers

- q superscript, integer for sequences/iterations (iterative solvers)
- ϵ small number
- t time
- X, V position and velocity
- r, e residual and error (column vectors)
- $\hat{r}, \hat{e}$ are transformed versions of r, e
- s search direction
- α, β are scalars
- $\bar{S}$ column vector (potential search direction)

Unit 6: Local Approximations

- p is an integer for sequences, polynomial degree, order of accuracy
- $p!$ is p factorial
- h scalar (relatively small)
- f' and f'' one derivative and two derivatives
- $f^{(p)}$ parenthesis (integer) indicates taking p derivatives
- ϕ basis functions
- w weighting function

Unit 7: Curse of Dimensionality

- A, V area and volume
- r radius
- N integer, number of sample points
- $\vec{x}$ vector of data input to a function

Unit 8: Least Squares

- <u>False Statements (first slide)</u>: a, b scalars
- $D, \widehat{D}$ diagonal matrices

Unit 9: Basic Optimization

- F system of functions (output is a vector not a scalar)
- ∂ partial derivative
- J Jacobian matrix of all first partial derivatives
- F' is the Jacobian of F
- ∇f gradient of scalar function f (Jacobian transposed)
- H matrix of all second partial derivatives of scalar function f (Jacobian of the gradient transposed)
- c^* critical point (special value of c)
- $\tilde{A}$ matrix
- $\tilde{b}$, $\tilde{c}$ vectors

Unit 10: Solving Least Squares

- $\hat{\Sigma}$ diagonal invertible matrix (no zeros on the diagonal)
- I_{nxn} stresses the size of the identity as nxn
- $\hat{b}_r, \hat{b}_z$ sub-vectors of $\hat{b}$ of shorter length (r for range, z for zero)
- $\hat{Q}$ orthogonal matrix
- $Q, \tilde{Q}$ are tall matrices with orthonormal columns (subsets of an orthogonal matrix)
- q_k column of Q
- R upper triangular matrix
- r_{ik} entry of R
- <u>Householder slides</u>: $\hat{v}$ normal vector, H householder matrix, a column vector

Unit 11: Zero Singular Values

- c_r, c_z sub-vectors of $\hat{c}$ of shorter length (range and zero abbreviations)
- A^+ pseudo-inverse of A
- T matrix (for similarity transforms)
- Q^q is orthogonal and R^q is upper triangular
- <u>Power Method Slides</u>: A^q and λ^q are A and λ raised to the q power

Unit 12: Regularization

- ϵ is a small positive number
- c^* is an initial guess for c
- r used in its geometric series capacity (a scalar)
- D is a diagonal matrix with all positive diagonal entries
- a_k is a column of A
- Θ is the angle between two vectors
- θ are pose parameters, φ represents all vertex positions of the face mesh
- C^* are 2D curves (vertices connected by line segments) drawn on the image
- C are 3D curves embedded on the 3D geometry, and subsequently projected into the 2D image space

Unit 13: Optimization

- f briefly is allowed to be either vector valued (or stay scalar valued)
- $\hat{f}$ is a (scalar) cost function for optimization
- F is a system of functions (the gradient in the case of optimization)
- $\hat{g}$ is a vector valued function of constraints
- η is a column vector of scalar Lagrange multipliers

Unit 14: Nonlinear Systems

- c^* is a point to linearize about
- d is for the standard derivative
- t is an arbitrary (scalar) variable
- dc is a vanishingly small differential (of c)
- Δ finite size difference
- α, β are scalars with $\beta \in [0,1)$
- g scalar function (that determines the line search parameter α)

Unit 15: Root Finding

- $\hat{g}$ is a modified g
- t is search parameter in 1D, replacing α
- t^* is the converged solution
- e is the error
- g' is the derivative of g
- $\hat{t}$ is a particular t
- $C \geq 0$ is a scalar
- p integer (power)
- t_L, t_R interval bounds
- t_M interval midpoint

Unit 16: 1D Optimization

- t_{min}, t_{M1}, t_{M2} more t values
- δ scalar (interval size)
- $\lambda \in (0, .5)$ is a scalar
- $\tau \in (0,1)$ is a scalar
- H_F is a 3rd order tensor of 2nd derivatives of F
- $OMG_{\hat{f}}$ is a 3rd order tensor of 3rd derivatives of $\hat{f}$

Unit 17: Computing Derivatives

- H is the Heaviside function
- $\hat{f}$ is a scalar function to be minimized
- $\hat{g}$ is a vector-valued function of constraints ($\hat{g}_i$ is a component of $\hat{g}$)
- $\hat{e}_i$ is the i-th standard basis vector
- $\hat{n}$ is a (possibly) high-dimensional unit normal
- $\epsilon > 0$, b are scalars
- e, $\log$ are the usual exponential and logarithmic functions
- C_1, C_2, C_3 are different sets of parameters
- f_1, f_2, f_3 are different functions
- X_1, X_2, X_3, X_4 are the data as it is processed through the pipeline
- X_{target} is the desired final result as the data is processed through the pipeline

Unit 18: Avoiding Derivatives

- $\widehat{m}$ is the integer length of the column vector output of $f(x, y, c)$
- $\tilde{f}(c)$ is a column vector of size $m * \widehat{m}$ that stacks the $\widehat{m}$ outputs of $f(x_i, y_i, c)$ for each of the m data points (x_i, y_i)
- $\hat{e}_k$ is the standard basis vector

Unit 19: Descent Methods

- (covered in other units)

Unit 20: Momentum Methods

- t is time
- t_o, t_f initial and final time
- Δt time step size
- k_1, k_2, k_3, k_4 intermediate function approximations in RK methods
- $\hat{c}$ intermediate states for TVD RK methods
- λ is a scalar, and represents an eigenvalue
- $X(t), V(t), A(t), F(t), M$ position, velocity, acceleration, force, mass
- v is the velocity of state c in parameter space
- $\alpha, \beta, \hat{\beta}$ are scalars