Calculus, finite differences
Interpolation, Splines, NURBS
CMSC 828 D

Least Squares, SVD, Pseudoinverse
- $Ax=b$ where $A$ is $m \times n$, $x$ is $n \times 1$, and $b$ is $m \times 1$.
- $A=USV^T$ where $U$ is $m \times m$, $S$ is $m \times n$, and $V$ is $n \times n$.
- $USV^T x=b$.
- If $A$ has rank $r$, then $r$ singular values are significant.
- $x = \sum \frac{s_i}{\sigma_i} v_i$, $\sigma_i > \epsilon$, $\sigma_i \leq \epsilon$.
- Pseudoinverse $A^+ = V\text{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0) U^T$.

Regularization
- Pseudoinverse provides one means of regularization.
- Another is to solve $(A+\epsilon I)x=b$.
- Solution of the regular problem requires minimizing $\|Ax-b\|^2 + \epsilon \|x\|^2$.
- This corresponds to minimizing $\|Ax-b\|^2 + \epsilon \|x\|^2$.
- Philosophy: pay a “penalty” of $\epsilon \|x\|^2$ to ensure solution does not blow up.
- In practice we may know that the data has an uncertainty of a certain magnitude so it makes sense to optimize with this constraint.

Well Posed problems
- Hadamard postulated that for a problem to be “well posed”:
  1. Solution must exist.
  2. It must be unique.
  3. Small changes to the input data should cause small changes to the solution.
- Many problems in science and computer vision result in “ill-posed” problems.
  - Numerically it is common to have condition 3 violated.
- Recall from the SVD $x = \sum \frac{s_i}{\sigma_i} v_i$, $\sigma_i > \epsilon$, $\sigma_i \leq \epsilon$.
- If $s_i$ are close to zero small changes in the “data” vector $b$ cause big changes in $x$.
- Converting ill-posed problem to well-posed one is called regularization.

Outline
- Gradients/derivatives
  - needed in detecting features in images
  - Derivatives are large where changes occur
  - essential for optimization
- Interpolation
  - Calculating values of a function at a given point based on known values at other points
  - Determine error of approximation
  - Polynomials, splines
- Multiple dimensions

Derivative
- In 1-D $\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$.
- Taylor series: for a continuous function
  - $f(x+h) = f(x) + k_0 h + k_1 h^2 + \cdots$
  - $f(x-h) = f(x) - k_0 h + k_1 h^2 + \cdots$
- Geometric interpretation: Approximate smooth curve by values of tangent, curvature, etc.
Remarks
• Mean value theorem:
  - $f(b)-f(a)=(b-a)\frac{df}{dx}|_c \quad a<c<b$
  - There is at least one point between $a$ and $b$ on the curve where the slope matches that of the straight line joining the two points.
• $\frac{df}{dx}=0$
  - Represents a minimum, maximum or saddle point of the curve $y=f(x)$
  - $\frac{d^2f}{dx^2} > 0$ minimum, $\frac{d^2f}{dx^2} < 0$ maximum
  - $\frac{d^2f}{dx^2} = 0$ saddle point

Finite differences
• Approximate derivatives at points by using values of a function known at certain neighboring points.
• Truncate Taylor series and obtain an expression for the derivatives.
• Forward differences: use value at the point and forward $x$
  - $\frac{df}{dx} = h^{-1} \left[ f(x+h) - f(x) \right] + O\left(h^2\right)$
  - $\frac{d^2f}{dx^2} = -\frac{f(x+h) - 2f(x) + f(x-h)}{2h^2} + O\left(h^2\right)$

Finite Differences
• Central differences
  - Higher order approximation
  - $2 \frac{df}{dx} = \frac{f(x+h)-f(x-h)}{2h} + O\left(h^2\right)$
  - $\frac{df}{dx} = \frac{f(x)+f(x-h)}{2h} + O\left(h^2\right)$
  - However we need data on both sides
  - Not possible for data on the edge of an image
  - Not possible in time dependent problems (we have data at current time and previous one)

Approximation
• Order of the approximation $O(h)$, $O(h^2)$
• Sidedness, one sided, central etc.
• Points around point where derivative is calculated that are involved are called the “stencil” of the approximation.
• Second derivative
  - $\frac{df}{dx} = h^{-1} \left[ f(x+h) - f(x-h) \right] + O\left(h^2\right)$
  - $\frac{d^2f}{dx^2} = \frac{f(x+h)-2f(x)+f(x-h)}{2h^2} + O\left(h^2\right)$
  - One sided difference of $O(h^2)$

Polynomial interpolation
• Instead of playing with Taylor series we can obtain fits using polynomial expansions.
  - 3 points fit a quadratic $ax^2+bx+c$
  - Can calculate the 1st and 2nd derivatives
  - 4 points fit a cubic, etc.
• Given $x_0, x_1, x_2, x_3$ and values $f_0, f_1, f_2, f_3$
  - $\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$
  - Vandermonde system – fast algorithms for solution.
• If more data than degree .. Can get a least squares solution.
• Matlab functions polyfit, polyval

Remarks
• Can use the fitted polynomial to calculate derivatives.
• If equation is solved analytically this provides expressions for the derivatives.
• Equation can become quite ill conditioned
  - especially if equations are not normalized.
  - $ax^2+bx+c$ can also be written as $a(x-h)^2+b(x-h)+c$.
  - Find the polynomial through $x_0, h, x_0^2+h$
  - $\begin{bmatrix} 1 & h & h^2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}$
  - $a_0=f_0, a_1=\frac{f_2-f_0}{2h}, a_2=\frac{f_1-f_2-f_0}{2h}$
  - Gives the expected values of the derivatives.
Polynomial interpolation
- Results from Algebra
  - Polynomial of degree $n$ through $n+1$ points is unique
  - Polynomials of degree less than $x^*$ is an $n$ dimensional space.
  - $1, x, x^2, \ldots, x^n$ form a basis.
  - Any other polynomial can be represented as a combination of these basis elements.
  - Other sets of independent polynomials can also form bases.
- To fit a polynomial through $x_0, \ldots, x_n$ with values $f_0, \ldots, f_n$
  - Use Lagrangian basis $l_k$
    \[ l_k = \prod_{i=0, i \neq k}^{n} \frac{x-x_i}{x_k-x_i}, \quad k=0, \ldots, n \]
  - Formula
    \[ p(x) = a_0l_0 + a_1l_1 + \ldots + a_nl_n \]
  - Then $a_i$ is $f_i$
  - Many polynomial bases: Chebyshev, Legendre, Laguerre ...
  - Bernstein, Bookstein ...

Spline interpolation
- Piecewise polynomial approximation
  - E.g. interpolation in a table
  - Given $x_0, x_1, \ldots, x_i$, evaluate $f$ at a point $x$ such that
    \[ \frac{x-x_0}{x_i-x_0} \begin{cases} \frac{x-x}{x_i-x} & s_i \leq x \leq x_i \\ 0 & \text{otherwise} \end{cases} \]
  - Construct approximations of this type on each subinterval
  - This method uses Lagrangian interpolants
- Endpoints are called breakpoints
- For higher polynomial degree we need more conditions
  - E.g. specify values at points inside the interval $[x_k < x < x_{k+1}]$
  - Specifying function and derivative values at the end points $x_0, x_n$ leads to cubic Hermite interpolation

Increasing $n$
- As $n$ increases we can increase the polynomial degree.
- However the function in between is very poorly interpolated.
- Becomes ill-posed.
- For large $n$ interpolant blows up.
  - Idea: Taylor series provides good local approximations
    - Use local approximations
    - Splines

Cubic Spline
- Splines – name given to a flexible piece of wood used by draftsmen to draw curves through points.
  - Bend wood piece so that it passes through known points and draw a line through $x$. 
  - Most commonly used interpolant used is the cubic spline
    - Provides continuity of the function, 1s and 2s derivatives at the breakpoints.
    - Given $n+1$ points we have $n$ intervals $[x_i, f_i]$, $i=1, \ldots, n+1$
    - Each polynomial has four unknown coefficients
      - Specifying function values provides 2 equations
      - Two derivative continuity equations provides two more
    - Left with two free conditions. Usually chosen so that second derivatives are zero at ends

Interpolating along a curve
- Curve can be given as $x(s)$ and $y(s)$
- Given $x,y,z$
- Can fit splines for $x$ and $y$
- Can compute tangents, curvature and normal based on this fit
- Things like intensity vary along the curve. Can also fit $l(s)$

Two and more dimensions
- Gradient $\nabla f = \frac{df}{dx} \hat{i} + \frac{df}{dy} \hat{j}$
- Directional derivative in the direction of a vector $n$
  \[ \nabla f \cdot n = \frac{df}{dx} e_x \cdot n + \frac{df}{dy} e_y \cdot n = \frac{df}{dn} \]
- Geometric interpretation
  - $\nabla f$ is normal to the surface $f(x)=c$
  - $n = \nabla f / |\nabla f|$
- Taylor series
  \[ f(x+h) = f(x) + h \frac{df}{dx} + \frac{h^2}{2} \frac{df}{dx^2} + O(h^2) \]
  \[ f(x+h) = f(x) + h \frac{dy}{dx} + \frac{h^2}{2} \frac{dy}{dx^2} + O(h^2) \]
Finite differences
• Follows a similar pattern. One dimensional partial derivatives are calculated the same way.
• Multiple dimensional operators are computed using multidimensional stencils.

\[
\nabla f = \nabla_x f \cdot \nabla_y f = \frac{f_{i+1,j} - f_{i-1,j}}{2h_x} - \frac{f_{i,j+1} - f_{i,j-1}}{2h_y}
\]

Interpolation
• Polynomial interpolation in multiple dimensions
• Pascals triangle
• Least squares
• Move to a local coordinate system

Tensor product splines
• Splines form a local basis.
• Take products of one dimensional basis functions to make a basis in the higher dimension.

Interpolation
• Polynomial interpolation in multiple dimensions
• Pascals triangle
• Least squares
• Move to a local coordinate system

NURBS
• Used for precisely specifying n-d data.
• October 3 Tapas Kanungo, NURBS: Non-Uniform Rational B-Splines

Derivative of a matrix
Suppose \( f(x) \) is a scalar-valued function of \( n \) variables \( x_j, j = 1,2...,n \), which we represent as the vector \( x \). Then the derivative or gradient of \( f \) with respect to this vector is computed component by component, i.e.,

\[
\nabla f(x) = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}
\]

If we have an \( n \times n \) vector-valued function \( f \) (note the use of bolds), of a \( d \)-dimensional vector \( x \), we calculate the derivatives and represent them as the Jacobian matrix

\[
J(x) = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_d} \end{bmatrix}. \quad (13)
\]

If this matrix is square, its determinant (Sect. 2.2.5) is called simply the Jacobian or sometimes the Jacobian determinant.

Jacobian and Hessian
We first recall the use of second derivatives of a scalar function of a scalar \( x \) in writing a Taylor series (or Taylor expansion) about a point:

\[
f(x) = f(x_0) + \frac{df}{dx} \Big|_{x=x_0} (x-x_0) + \frac{1}{2} \frac{d^2f}{dx^2} \Big|_{x=x_0} (x-x_0)^2 + O((x-x_0)^3). \tag{20}
\]

Analogously, if our scalar-valued \( f \) is a function of a vector \( x \), we can expand \( f(x) \) in a Taylor series around a point \( x_0 \):

\[
f(x) = f(x_0) + \sum_{i=1}^n \frac{\partial f}{\partial x_i} \Big|_{x=x_0} (x_i-x_0) + \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_{x=x_0} (x_i-x_0)(x_j-x_0) + O((x-x_0)^3). \tag{21}
\]

where \( H \) is the Hessian matrix, the matrix of second-order derivatives of \( f(x) \), here