MAT 300/500 Supplementary Notes

(Note: These notes are meant to supplement the lectures. Full lecture notes can only be obtained by attending class lectures and taking notes.)

Lecture 1

Main Points:

- **Syllabus - grading, homework and quiz policies.**
- **Introduction to Project Part I - Bezier Curves, Nested Linear Interpolation**
- **Review of Linear Algebra**
- **Introduction to Vector Spaces of Polynomials and bases**

Refer to project specification for full details. In class we discuss the basic steps and requirements without full justification, which will come later. The idea is to get going quickly on the project.

The projects are all about different methods for rendering curves in 2D. The curves that we deal with in this course are either parametric polynomial or parametric piecewise polynomial. We also discuss some implicit curves briefly.

- A 2D parametric polynomial curve is:
$$\gamma(t) = (p_1(t), p_2(t))$$
where $p_1(t)$ and $p_2(t)$ are polynomials.
- A 3D parametric polynomial curve is:
$$\gamma(t) = (p_1(t), p_2(t), p_3(t))$$
where $p_1(t)$, $p_2(t)$ and $p_3(t)$ are polynomials.
- A 2D parametric piecewise polynomial curve is:
$$\gamma(t) = (f_1(t), f_2(t))$$
where $f_1(t)$, and $f_2(t)$ are piecewise polynomials.
- An 2D implicit curve is given by an equation $f(x, y) = 0$, such as the unit circle: $x^2 + y^2 - 1 = 0$.

Algorithms for rendering curves can use explicit polynomial formulas like $\gamma(t) = (p_1(t), p_2(t))$, or they can use other methods, as we see below, such as Nested Linear Interpolation.

In the project, you will design an interface (or use one of the frameworks) which brings up a window and allows the user to click on points to select *input points*, which are either control points (which influence the shape of the curve) or interpolation points (which the curve must pass through). The points are typically labelled P_0, $P_1, \ldots, P_d$, in the order in which they are clicked by the user. In the first part of the project, for each of 3 subparts, you use the control points to generate a Bezier curve. The method is different for each subpart, but the curve should be the same if you work with high enough resolution.

Nested Linear Interpolation (Refer to Project Part I, subpart 1)

- t-values should be chosen and one point generated on the curve for each t. It is up to you to choose enough t-values to the make the curve look smooth.
- the basic interval for the t-values is $[0, 1]$ (optionally you can extend the curve to fill the screen with $t < 0$ and $t > 1$.)
- basic linear interpolation between points: $(1-t)P_0 + tP_1$. (This is an affine sum since the coefficients $1-t$ and t add up to 1.)
- nested linear interpolation for three points:

$$\gamma_{[P_0,P_1,P_2]}(t) = (1-t)\left[(1-t)P_0 + tP_1\right] + t\left[(1-t)P_1 + tP_2\right]$$

- recursive form for Nested Linear Interpolation:

$$\gamma(t) = \gamma_{[P_0,P_1\ldots P_d]}(t) = (1-t)\gamma_{[P_0,P_1\ldots P_{d-1}]}(t) + t\gamma_{[P_1\ldots P_d]}(t).$$

- Bezier point recursion for stages $k = 1, \ldots, d$, with base case $P_i^0 = P_i$ (control points)

$$P_i^k = (1-t)P_i^{k-1} + tP_{i+1}^{k-1}.$$

- Array of Bezier Points for a degree three Bezier curve with 4 control points P_0, P_1, P_2 and P_3: (and some chosen value of t)

$$\begin{array}{cccc}
P_0^0 & & & \\
 & P_0^1 & & \\
P_1^0 & & P_0^2 & \\
 & P_1^1 & & P_0^3 = \gamma(t) \\
P_2^0 & & P_1^2 & \\
 & P_2^1 & & \\
P_3^0 & & &
\end{array}$$

- Note: for any triangle of 3 points in the Bezier point diagram:

$$\begin{array}{cc}
P & \\
 & R \\
Q &
\end{array}$$

  R is obtained as: $R = (1-t)P + tQ$.

- Note: when computing $\gamma(t)$ it is important to use arrays based on Bezier points instead of recursive functions (even though the recursive form is nice to write down.)
- the **polyline** is the collection of line segments joining the control points together in order from P_0 to P_d.
- the **shells** consist of the polyline together with the line segments joining the Bezier points for each stage (superscript) in order from P_0^k to P_{d-k}^k.
- Note: This is the ONLY subpart of the Project Part I which uses the shells. The other parts do not use shells.

BB-(Bernestein-Bezier) form (Refer to Project Part I, subpart 2)

- BB-form:

$$\gamma(t) = \gamma_{[P_0, P_1 \ldots P_d]}(t) = \sum_{i=0}^{d} B_i^d(t) P_i.$$

- Bernstein polynomials:

$$B_i^d(t) = \binom{d}{i}(1-t)^{d-i}t^i, \quad i = 0, \ldots, d,$$

  where the binomial coefficient $\binom{d}{i}$ is defined as:

$$\binom{d}{i} = \frac{d!}{(d-i)!i!}.$$

- Note that the BB-form is an affine sum, since the coefficients $B_i^d(t)$ all add up to 1 (to be proved later)
- Binomial coefficients should be computed with Pascal's Identity: (not factorials)

$$\binom{d}{i} = \binom{d-1}{i-1} + \binom{d-1}{i},$$

  with base cases: $\binom{d}{0} = \binom{d}{d} = 1$.

Important Points to Review from Linear Algebra:

- linear independence (for a finite set of vectors)
  - a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is linearly *independent* if the vector equation:

$$c_1 v_1 + \cdots + c_n v_n = 0_V$$

    is only true when all the coefficients are zero. (The coefficients are real number scalars, and 0_V represents the zero vector for the vector space V.)
  - a set of n column vectors in R^n is linearly independent if and only if the $n \times n$ matrix of column vectors is nonsingular, which means it has nonzero determinant.
- linear dependence (for a finite set of vectors)
  - a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is linearly *dependent* if it is not linearly independent.
  - equivalently, a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is linearly *dependent* if the vector equation

$$c_1 v_1 + \cdots + c_n v_n = 0_V$$

    has a nontrivial solution, which means that some coefficients can be found which are not all zero and which make the equation true.
  - equivalently, a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is linearly *dependent* if at least one of the vectors can be written as a linear combination of the others.
- spanning property (for a finite set of vectors), and the $Span(S)$.
  - a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is said to *span* V if any vector in V can be written as a linear combination of $v_1, \ldots, v_n$.
  - The set $Span(S)$ for $S = \{v_1, \ldots, v_n\}$ in a vector space V means the set of linear combinations of vectors in S.
  - The set $Span(S)$ is always a vector sybspace of V.
- basis and dimension

- a set of n vectors $\{v_1, \ldots, v_n\}$ in a vector space V is a basis of V if it is both linearly independent and spans V. (Note: Any basis must have the same number of vectors.)
- The dimension of a vector space V is the number of vectors in a basis.

- coordinate vectors with respect to a basis
  - If $B = \{v_1, \ldots, v_n\}$ is a basis of a vector space V and v is any vector in V, then we can write
  $$v = c_1 v_1 + \cdots + c_n v_n$$
  for some real scalar coefficients. The column vector
  $$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$
  is then called the *coordinate vector* for v with respect to the basis B.

- change of basis matrix
  - If B_1 and B_2 are bases of a vector space V, and a vector v in V has coordinate vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, then these column vectors are related by the equation
  $$A\mathbf{x}_1 = \mathbf{x}_2$$
  where A is an invertible matrix called the *change of basis matrix* from B_1 to B_2. Also, A^{-1} is then the change of basis matrix from B_2 to B_1.

- determinant of a square matrix (with cofactors)
  - the 2×2 determinant formula:
  $$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$
  - the 3×3 determinant formula:
  $$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}$$

- determinant of a triangular matrix
  - A matrix is called *upper triangular* if the entries below the main diagonal are zero, and *lower triangular* if the entries above the main diagonal are zero. A matrix which is upper or lower triangular is called a *triangular* matrix. A matrix which is both upper and lower triangular is called a *diagonal* matrix.
  - The determinant of a triangular matrix is equal the product of the diagonal entries.
  - a 3×3 upper triangular determinant:
  $$\begin{vmatrix} a_1 & b_1 & c_1 \\ 0 & b_2 & c_2 \\ 0 & 0 & c_3 \end{vmatrix} = a_1 b_2 c_3.$$

- Inverse of a square matrix
  - A square matrix A is invertible if and only if the determinant of A is nonzero:
  $$A^{-1} \text{ exists } \iff det(A) \neq 0.$$

- 2×2 matrix inversion formula:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

- Gaussian elimination method for 3×3 (or higher) matrix inversion. The arrow means apply Gaussian elimination to entire matrix, reducing A to the identity I, and reducing I to A^{-1}.

$$(A|I) \longrightarrow \left(I|A^{-1}\right).$$

- subspace of a vector space:

  A subset U of a vector space V is called a vector subspace of V if it satisfies the axioms of a vector space with the same addition and scalar multiplication inherited from V.

- criterion for a subset to be a subspace:

  A subset U of a vector space V is a subspace of V if and only if U is closed under vector addition and scalar multiplication. (Equivalently, if u and v are in U, then $u + v$ must also be in U, and for any scalar c, $c \cdot u$ must also be in U.)

Important Theorem from Linear Algebra:

If V is a vector space of dimension n, and $S = \{v_1, \ldots, v_n\}$ is a set of n distinct vectors in V, then S is linearly independent if and only if S is a spanning set. We use this in most proofs of new bases for P_d or for vector spaces of splines.