Updated on March 6, 2022

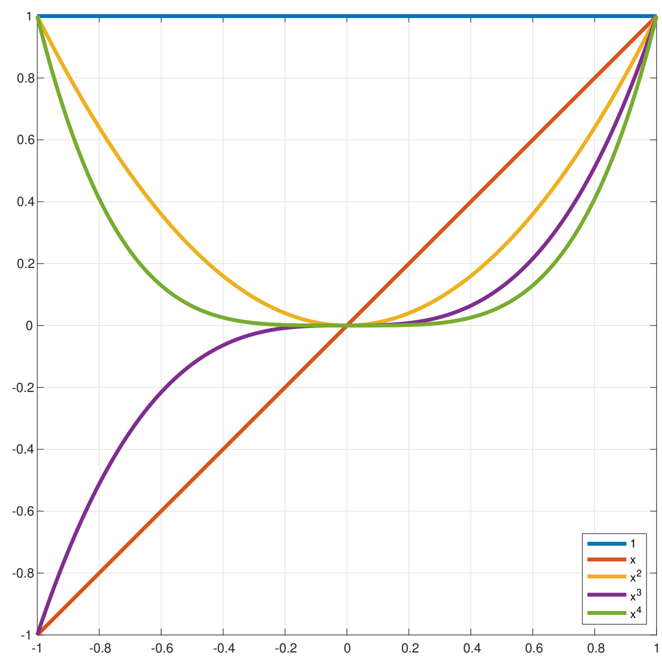
Hints for Homework 7

4. (Adapted from **FNC** 3.3.3.)

   Just to clarify what it means to plot a column of a matrix, consider the Vandermonde matrix

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix},$$

   where $-1 = x_1 < x_2 < \cdots < x_m = 1$ are equally spaced. So each column of V is a set of evenly distributed samples of a monomial function x^j, $j = 0, \ldots, n-1$. That is, the first column is a discrete representation of the constant function $f_1(x) = 1$, the second column $f_2(x) = x$, and the general j^{th} column $f_j(x) = x^{j-1}$. Thus if every column of V is plotted against $\mathbf{x} = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}$, one obtains something like



   Now, thin QR factorization of V will find the orthonormal set of vectors whose span coincides with the span of columns of V and put them in $\widehat{Q}$ (with $\widehat{R}$ being the *recovery formula*). When the columns of $\widehat{Q}$ are plotted against $\mathbf{x}$, you will see a visualization of so-called *orthogonal polynomials.*

5. (Gram-Schmidt in MATLAB; **LM** 12.6–2)

- The symbol $\mathcal{O}$ in the third line of the problem stands for the zero matrix, with suitable dimensions according to the context.
- For a summary of the Gram-Schmidt procedure, see "Appendix: Gram-Schmidt Orthogonalization" at the end of Lecture 22. For your convenience, the main algorithm (mathematical) is included below:

$$
\boxed{
\begin{aligned}
\mathbf{q}_1 &= \frac{\mathbf{a}_1}{r_{11}}, \\
\mathbf{q}_2 &= \frac{\mathbf{a}_2 - r_{12}\mathbf{q}_1}{r_{22}}, \\
\mathbf{q}_3 &= \frac{\mathbf{a}_3 - r_{13}\mathbf{q}_1 - r_{23}\mathbf{q}_2}{r_{33}}, \qquad \text{where} \quad r_{ij} = \begin{cases} \mathbf{q}_i^{\mathrm{T}}\mathbf{a}_j, & \text{if } i \neq j \\ \pm\left\|\mathbf{a}_j - \displaystyle\sum_{k=1}^{j-1} r_{kj}\mathbf{q}_k\right\|_2, & \text{if } i = j. \end{cases} \\
&\ \vdots \\
\mathbf{q}_n &= \frac{\mathbf{a}_n - \sum_{i=1}^{n-1} r_{in}\mathbf{q}_i}{r_{nn}},
\end{aligned}
}
$$

  Here, $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ are the columns of A; $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ the columns of Q; r_{ij} the (i, j)-element of R.
- Your goal is to translate the mathematical procedure inside the box to a MATLAB code. Begin the function with

```
function [Q,R] = GramSchmidt(A)
```

  so that it calculates both Q and R. Keep in mind that the Gram-Schmidt procedure yields a **thin QR factorization** of A.