Lin.Alg. Project 2 3114

Prof. J.L. King

Recall that a ***polynomial*** P is a function of the form

$$P(x) \;=\; a_N x^N + a_{N-1}x^{N-1} + \cdots + a_1 x + a_0$$

where the ***coefficients*** a_j are real (more generally, complex) numbers. If $a_N \neq 0$ then the ***degree*** of P is N.

As an abbreviation, say that P is an **"N-pol"** if it is a polynomial whose degree is *at most* N. Non-zero constant polynomials have degree 0 and –this is a reasonable convention– the constant 0 polynomial is defined to have degree $-\infty$. (Think what happens when you multiply polynomials, and you'll see the reason for this convention.)

FACT 1. *Suppose* $\widehat{x} := (x_0, x_1, \ldots, x_N)$ *is a tuple of* $N+1$ *different real* (or complex) *numbers. Suppose* $\widehat{y} := (y_0, y_1, \ldots, y_N)$ *is a tuple of real* (or complex) *numbers, not necessarily distinct. THEN, there are many polynomials* h *such that*

$$\text{For } j = 0, \ldots, N\text{:} \quad h(x_j) = y_j. \tag{2}$$

Furthermore, there is a <u>unique</u> N*-pol with this property. Call this* N*-pol* $P_{\widehat{x},\widehat{y}}$.

Project, programming. Write a Maple (preferred) or Matlab program called `polyinter` which takes two $(N+1)$-tuples $\widehat{x}$ and $\widehat{y}$ as input, eg,

$$\texttt{polyinter}\big([2, -8.94, Pi, 3], [0, 0, 17, 19]\big)$$

and computes the unique polynomial P of degree at most N fulfilling (2).

Your program must work for any N (including $N = 0$) and will compute N from the length of the tuples. If the tuples have unequal lengths, or the x-tuple has a repeated value, then your program must generate a meaningful error message.

Your program should operate as in the algorithm for Project 1: The coefficients a_j are the $N+1$ unknowns in a system of $N+1$ linear equations that you cook up. Your program makes the appropriate augmented matrix, **Arr**, then will apply `rref` to produce a matrix **R**. From the rightmost column of **R**, compute the desired polynomial **P** and return this value.

Your program should bind 3 "sticky" (global) variables.

Arr is bound to the array describing the augmented system whose solution is the coefficient list of P.

Coefvec is bound to the coefficient list of P

P is the Maple (or Matlab) representation of the interpolation polynomial.

Note. I will email out a sample run of such a program, along with a few helper Maple programs that you are free to use. Please put "header comments" in *your* program(s) analogous to the that I email to you.

Project, essay. Now fix a value of N.

[1] Show that that the set of N-pols, call it $\mathbb{V}$, is a *vectorspace*, under the usual notions of adding polynomials and multiplying by a constant.

[2] Now fix $\widehat{x}$. Let $\mathbb{Y}$ denote the *vectorspace* of $(N+1)$-tuples

$$\widehat{y} = (y_0, y_1, \ldots, y_N)$$

with the usual notions of addition and sv-multiplication.

Look at the map $T\colon \mathbb{Y} \to \mathbb{V}$ defined by

$$T(\widehat{y}) := P_{\widehat{x},\widehat{y}}$$

Using the above "uniqueness" fact, show that T is a linear transformation. (That is, verify the two axioms of linearity.)

[3] Let $\mathbf{e}_0, \ldots, \mathbf{e}_N$ be the "standard basis" for $\mathbb{Y}$, in that

$$\mathbf{e}_0 = (1, 0, 0, \ldots, 0) \quad (N \text{ zeros})$$

Can you establish the above "Fact" by yourself? That is, can you convince me that there do exist polynomials

$$P_{\widehat{x},\mathbf{e}_0},\ \ldots,\ P_{\widehat{x},\mathbf{e}_N}$$

by directly giving me a formula for each one?

[4] Read your email for an update.

Filename: `Classwork/LinearAlg/LinC1998g/P2-project.LinC1998g.ams.tex`
As of: *Sunday 06Sep2015* Typeset: *6September2015*