

Worksheet 4 (week5) MCC3, MATLAB Hans G. Feichtinger, Edinburgh, Feb. 6th

1. It is well known (from algebra courses) that it is enough to know the coefficients of a polynomial of order n (or degree $n-1$) from the values of the corresponding polynomial $p(x) = p_{\mathbf{a}}(x) = a_1x^{n-1} + \dots + a_{n-1}x + a_n$ (following the MATLAB conventions). We can check this in two steps:
 - a) verify that (for some random polynomial of degree 5, say, and some random numbers, both of them ideally complex, but a “real-valued” experiment should make you confident as well) the Vandermonde matrix `vand(a)` is providing exactly such a matrix, and
 - b) verify that `V=vand(A)` is in fact invertible (by looking at `rref(V)` or determining `det(V)`). Recall that the VANDERMONDE matrix is the linear matrix which describes the mapping from the space of
2. It is clear that a linear system $\mathbf{A} * \mathbf{x} = \mathbf{b}$ has a unique solution for every $\mathbf{b}$ if and only if $\mathbf{A}$ is invertible. But not all invertible matrices are “equal” (think geometrically: acute angles produce imprecise information about the intersection point). In order to check this choose 4 points in the interval $[-1, 1]$ (not too close to each other) and built the corresponding Vandermonde matrix `V4`. The corresponding condition number `cond(V4)` should not to big. Then add one more point (try to put it relatively close to one of the existing points or not so close, like halfway between the existing points), and check the influence on the corresponding 5–point Vandermonde matrix `V5`. Clearly enough it will be large (in fact `norm(inv(V5))` will be large, if the minimal distance is small, meaning that in this case small changes of the values of the polynomial may mean large changes in the coefficients!
3. The unit roots of order n occur as entries in the FFT-matrix, which we can obtain via the call `F = fft(eye(n))`. Convince yourself that this is the case (by plotting the first few columns of F). Check that `F == F'` (for some small n , e.g. $n = 5$). The command `plot(F(:,2), 'k*'); axis square; shg;` might be convincing (here $n = 12$ or $n = 36$ would look interesting).
4. 4
5. 5
6. 6