
This note describes the programs

`o_roots.m` and `o_gcd.m`

The first is for calculating the roots of a polynomial in the Bernstein basis. The second is for calculating the GCD of two arbitrary polynomials. Both of these files use `o1.m` to calculate a GCD and quotient polynomials. Bernstein polynomials $f(y)$ and $g(y)$. The modified matrix $S(f,g)$.

The programs are executed by typing

`o_roots(n,ec_min,ec_max,BOOL_SNTLN,BOOL_APF,BOOL_DENOM,BOOL_PREPROC,BOOL_LOG)` and
`o_gcd(n,ec_min,ec_max,BOOL_SNTLN,BOOL_APF,BOOL_DENOM,BOOL_PREPROC,BOOL_LOG)`

where

`n` is an integer that defines the polynomials $f(y)$ in the program `Root_examples.m` or polynomials $f(y)$ and $g(y)$ in the program `GCD_examples.m` for `o_roots` and `o_gcd.m` respectively.

`ec_min` and `ec_max` are the minimum and maximum ratios:

$$\frac{\text{noise level}}{\text{signal level}}$$

measured in the componentwise sense.

`BOOL_SNTLN` is a boolean value, whether structured perturbations are added to the Sylvester matrix. `BOOL_APF` is a boolean value, whether structured perturbations are added to the Approximate polynomial factorisation.

The differences in the programs are:

- The program o_gcd.m takes two polynomials $f(y)$ and $g(y)$ and calculates the GCD of their perturbed forms.
- The program o_roots.m takes only one exact polynomial $\hat{f}(y)$ adds noise to its coefficients such that $f(y)$ is the noisy form, calculates it's derivative $f'(y)$ and performs a gcd calculation to obtain $d_1(y)$. A new gcd calculation is performed on d(y) and its derivative. This is performed iteratively until the GCD calculation output is a scalar. The set of gcd values $d_1, d_2, \ldots d_n$ are used to compute the roots and corresponding multiplicities of $\hat{f}(y)$

Examples of executing the programs are
`o1(5,1e-8), o2(15,1e-7), o3(21,1e-5)` and `o4(30,1e-7)`.

□

The programs produce the following output: