Degree of an AGCD of two Bernstein polynomials

This note describes the programs

`o_roots_Univariate.m` and `o_gcd_Univariate_2Polys.m`

1. The first file is used in the computation of a polynomial's roots and corresponding multiplicities, where the given polynomial is in Bernstein form.
2. The second file is used in the computation of the Greatest Common Divisor (GCD) of two polynomials in Bernstein form.

The programs are executed by typing

```
o_roots_Univariate ( ex_num, emin, emax, mean_method, bool_alpha_theta,
low_rank_approx_method, apf_method, sylvester_build_method, rank_revealing_metric,
deconvolution_method_hx, deconvolution_method_wx, )
o_gcd_2Polys_Univariate ( ex_num, emin, emax, mean_method, bool_alpha_theta,
low_rank_approx_method, apf_method, sylvester_build_method, rank_revealing_metric )
```

where

ex_num : (String) A string typically containing an integer, which defines the example to be run.

emin : (Float) Minimum signal : noise ratio

emax : (Float) Maximum signal : noise ratio

mean_method : (String) Method used to compute the mean of the entries of the partitions of the Sylvester subresultant matrix

- `None` : No mean method used
- `Geometric Mean My Method` : Fast method
- `Geometric Mean Matlab Method` : Standard Matlab method
- `Arithmetic Mean` :

bool_alpha_theta : (Boolean)

- `true` : Preprocess polynomials
- `false` : Exclude preprocessing

low_rank_approx_method : (String)

- `None` :
- `Standard SNTLN` :
- `Standard STLN` :
- `Root Specific SNTLN` :

apf_method : (String)

- `None` :
- `Standard Linear APF` :
- `Standard NonLinear APF` :

Sylvester_Build_Method : (String)

T : The matrix $T_k\left(f(x), g(x)\right)$

DT : The matrix $D^{-1}_{m+n-k} T_k\left(f(x), g(x)\right)$

DTQ : The matrix $D^{-1}_{m+n-k} T_k\left(f(x), g(x)\right)\hat{Q}$

TQ : The matrix $T_k\left(f(x), g(x)\right)\hat{Q}$

DTQ Rearranged : The matrix $S_k\left(f(x), g(x)\right) = D^{-1}_{m+n-k} T_k\left(f(x), g(x)\right)\hat{Q}$ where the entries are computed in a rearranged form.

DTQ Denominator Removed : $\tilde{S}\left(f(x), g(x).\right)$

rank_revealing_metric (String)

Singular Values : Compute the degree of the GCD using minimum singular values of the set of Sylvester subresultant matrices.

Max/Min Singular Values : Compute the degree of the GCD using the ratio of maximum to minimum singular values of each Sylvester subresultant matrix.

R1 Row Norms : Compute the degree of the GCD using the norm of the rows of the matrix R from the QR decomposition of each Sylvester subresultant matrix.

R1 Row Diagonals : Compute the degree of the GCD using the diagonals of the matrix R obtained by the QR decomposition of each Sylvester subresultant matrix.

Residuals : Compute the degree of the GCD using the minimum residual obtained by removing the optimal column of each of the Sylvester subresultant matrices.

deconvolution_method_hx (String)

Separate :

Batch :

Batch STLN :

Batch Constrained :

Batch Constrained STLN :

deconvolution_method_wx (String)

Separate :

Batch :

1 Examples

Examples of executing the programs are

```
    o_gcd_Univariate_2Polys ( '1', 1e-10, 1e-12, 'Geometric Mean Matlab Method', true, 'None',
'None', 'DTQ', 'Minimum Singular Values' )
    o_roots_Univariate ( '1', 1e-12, 1e-10, 'Geometric Mean Matlab Method', true, 'None', 'None',
'DTQ', 'Minimum Singular Values', 'Batch Constrained', 'Batch' )
```

2 Points of Interest

2.1 Other Variables

Other less frequently changed variables are found in the file `SetGlobalVariables.m`

SEED (Int)Used in random number generation

PLOT_GRAPHS : (Boolean)

1. true : Plot graphs
2. false : Don't plot graphs

BOOL_LOG : (Boolean) Whether to use logs in the computation of the geometric mean

1. true
2. false

GCD_COEFFICIENT_METHOD (String)

ux and vx

ux

MAX_ERROR_SNTLN (Float)

MAX_ITERATIONS_SNTLN (Int)

MAX_ERROR_APF (Float)

MAX_ITERATIONS_APF (Int)

GET_HX_METHOD (String) Used in the computation of the set of polynomials $h_i(x)$ in the Tobey and Horowitz factorisation algorithm.

From Deconvolutions : Deconvolve the set of polynomials $f_i(x)$

From ux : The polynomials $h_i(x)$ are a by-product of the GCD computation.

DECONVOLVE_METHOD_HX_FX (String) The deconvolution method used to obtain $h_i(x)$ from $f_i(x)$

Batch

Batch With STLN

Batch Constrained

Batch Constrained With STLN

DECONVOLVE_METHOD_WX_HX (String)

Separate

Batch

MAX_ERROR_DECONVOLUTIONS (Float)

MAX_ITERATIONS_DECONVOLUTIONS (Int)

PREPROC_DECONVOLUTIONS (Boolean)

2.2 Limits

The code makes frequent use of variables `t_limits` and `k_limits`.

`t_limits` **:** In the computation of the factorisation of $\hat{f}_0(x)$, many GCD computations are required to generate the sequence $\hat{f}_i(x) = GCD\left(\hat{f}_{i-1}(x), \hat{f}'_{i-1}(x)\right)$. The degree of $GCD\left(\hat{f}_i(x), \hat{f}'_i(x)\right)$ is bound by the number of distinct roots of $\hat{f}_i(x)$, and the number of distinct roots is always less than or equal to the number of distinct roots of $\hat{f}_{i-1}(x)$.

`k_limits` **:** This variable defines the range of Sylvester subresultant matrices $S_k\left(\hat{f}_i(x), \hat{f}'_i(x)\right)$ considered in the computation of the degree of the GCD. By default this range is set between 1 and $\min(m, n)$, however `limits_t` can also be used since it is known that all subresultant matrices outside this range are known to be singular.

2.3 Computing the Degree of the GCD

There are several methods considered for the computation of the degree of the GCD. A global variable `SETTINGS.RANK_REVEALING_METRIC` defined in the file `SetGlobalVariables.m` determines which method is used.

`Singular Values` **:**

`R1 Row Norms` **:**

`R1 Row Diagonals` **:**

`Residuals` **:**

2.4 The Deconvolution Problem in the Polynomial Factorisation Algorithm

The deconvolution functions can be tested away from the polynomial factorisation algorithm using the file `o_Deconvolution.m`. The deconvolution methods considered are:

`Separate` **:**

`Batch` **:**

`Batch With STLN` **:**

`Batch Constrained` **:**

`Batch Constrained With STLN` **:**

3 Code Flow

[illegible]