Degree of an AGCD of three univariate polynomials in Bernstein form

---

This note describes the function

`o_gcd_Univariate_3Polys.m`

The program is executed by typing

```
o_gcd_Univariate_3Polys ( ex_num, ex_num_variant emin, emax, mean_method,
bool_alpha_theta, low_rank_approx_method, apf_method, sylvester_matrix_variant,
n_equations, rank_revealing_metric )
```

where

Variable	Type	Description
`ex_num`	(String)	A string typically containing an integer. This determines the set of polynomials used in the GCD finding problem.
`ex_num_variant`	(String)	Determines the ordering of the example polynomials $a(x)$, $b(x)$ and $c(x)$ from the example file are assigned to $f(x)$, $g(x)$ and $h(x)$. All three unique polynomial orderings can be considered. 'a' assigns $a(x) \rightarrow f(x)$, $b(x) \rightarrow g(x)$ and $c(x) \rightarrow h(x)$. 'b' assigns $a(x) \rightarrow g(x)$, $b(x) \rightarrow f(x)$ and $c(x) \rightarrow h(x)$. 'c' assigns $a(x) \rightarrow h(x)$, $b(x) \rightarrow g(x)$ and $c(x) \rightarrow f(x)$.
`emin`	(Float)	Minimum level of additive componentwise noise.
`emax`	(Float)	Maximum componentwise noise
`mean_method`	(String)	Method used to compute the mean of the entries of the partitions of the Sylvester subresultant matrix **None :** No mean is computed and coefficients of polynomials $f(x)$, $g(x)$ and $h(x)$ are not normalised. **Geometric Mean My Method :** My method for computing the geometric mean using a reduced algorithm. **Geometric Mean Matlab Method :** Standard Matlab method for computing the geometric mean of a set of values. **Arithmetic Mean :**
`bool_alpha_theta`	(Boolean)	Determines whether the subresultant matrices are preprocessed **true :** Preprocess polynomials **false :** Exclude preprocessing

`low_rank_approx_method`	(String)	Method used for computing the low rank approximation of the tth subresultant matrix
		
		`None :`
		
		`Standard SNTLN :`
		
		`Standard STLN :`
		
		`Root Specific SNTLN :`

`apf_method`	(String)	`None :`
		
		`Standard Linear APF :`
		
		`Standard NonLinear APF :`

`Sylvester_Matrix_Variant`	(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 :` $\tilde{S}\left(f(x), g(x).\right)$
`n_equations`	(String)	Determines the shape of the three-polynomial subresultant matrices. **2** : The subresultant matrices have a 2×3 partitioned structure. **3** : The subresultant matrices have a 3×3 partitioned structure.

`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.

0.1 Starting Points

A set of experiments files consider alternative combinations of input variables.

`Experiment0_Standard( ex_num, ex_num_variant, bool_preproc)`	This experiment is a blank canvas, the user can change any variable. Note that the `low_rank_approx_method` and `apf_method` should remain set to *None* since the relevant functions have not been extended to the three-polynomial problem. These arguments are included for several reasons. The input arguments of the three-polynomial problem closely match those of the two polynomial problem. The code has been developed such that the relevant functions can easily be added in the future.
`Experiment1SylvesterVariants_3Polys( ex_num, ex_num_variant, bool_preproc)`	This experiment considers the alternate variants of the (2×3) partitioned, three-polynomial subresultant matrices $\hat{S}_k(f,g,h)$. The degree of the Approximate Greatest Common Divisor (AGCD) is computed using each of the four variant $\hat{T}_k(f,g,h)$, $\hat{D}_k^{-1}\hat{T}_k(f,g,h)$, $T_k(f,g,h)\tilde{Q}_k$ and $\hat{D}^{-1}T_k(f,g,h)\tilde{Q}_k$. Typically best results are obtained when the subresultant matrix variant $\hat{D}^{-1}T_k(f,g,h)\tilde{Q}_k$ is considered.

`Experiment2Preprocessing_3Polys(ex_num, ex_num_variant)`	This experiment considers the computation of the GCD of three univariate polynomials by Sylvester subresultant matrix based methods, where the matrices may or may not be preprocessed. It is typically shown that preprocessing yields improved results.
`Experiment3ReorderPolys.m`	This experiment considers the alternative orderings of the three polynomials from the example file. Ordering 'a' assigns $a(x) \rightarrow f(x)$, $b(x) \rightarrow g(x)$ and $c(x) \rightarrow h(x)$. 'b' assigns $a(x) \rightarrow g(x)$, $b(x) \rightarrow f(x)$ and $c(x) \rightarrow h(x)$. 'c' assigns $a(x) \rightarrow h(x)$, $b(x) \rightarrow g(x)$ and $c(x) \rightarrow f(x)$. This is equivalent to keeping the three polynomials $f(x)$, $g(x)$ and $h(x)$ constant and considering the three orderings of the polynomials in the subresultant matrices $\hat{S}_k(f,g,h)$, $\hat{S}_k(g,f,h)$ and $\hat{S}_k(h,g,f)$.

0.2 Global Variables

Other less frequently altered variables are found in the file `SetGlobalVariables_GCD_3Polys.m`

Variable	Type	Description
`EX_NUM`	(String)	Example Number
`EX_NUM_VARIANT`	(String)	'a', 'b' or 'c'
`EMIN`	(Float)	
`EMAX`	(Float)	
`SEED`	(Int)	Used in random number generation
`PLOT_GRAPHS`	(Boolean)	1. True : Plot graphs 2. False : Don't plot graphs
`PLOT_GRAPHS_GCD_DEGREE`	(Boolean)	Plot graphs related to the computation of the degree of the Greatest Common Divisor (GCD). For instance, setting this variable to true would plot the minimum singular values associated with the set of subresultant matrices if the `rank_revealing_metric` was set to 'Minimum Singular Values'.
`PLOT_GRAPHS_PREPROCESSING`	(Boolean)	Plot graphs related to preprocessing of the subresultant matrices. For instance, setting this variable to true would plot the coefficients of the unprocessed and preprocessed polynomials $f(x)$ and $\lambda_1 \tilde{f}_1(\omega)$, $g(x)$ and $\mu_1 \tilde{g}_1(x)$, and $h(x)$ and $\rho_1 \tilde{h}_1(\omega)$.
`PLOT_GRAPHS _LOW_RANK_APPROXIMATION`	(Boolean)	Plot graphs associated with the computation of the low rank approximation of the t-th subresultant matrix. Since the low rank approximation methods have not been implemented, this is best set to *false*.

BOOL_LOG	(Boolean)	Whether to use logs in the computation of the geometric mean and computation of the entries in the subresultant matrices. Typically this is set to false as the conversion to logs and back again seems to introduce more error than leaving the problem in the power basis. 1. True : Use logs 2. False : Don't use logs
BOOL_ALPHA_THETA	(Boolean)	Determines whether the subresultant matrices are preprocessed or not.

SCALING_METHOD	(String)	The scaling method determines which polynomials are scaled in the three-polynomial subresultant matrices. We may only wish to scale $f(x)$ and $g(x)$ by optimised values λ and μ while scaling $h(x)$ by $\rho = 1$. Theoretically, any combination of scaling should give the same result, but in practice this may not be the case. Further study is required and code correctness must be checked.
		`lambda_mu_rho`
		`lambda_mu`
		`mu_rho`
		`lambda_rho`

MEAN_METHOD	(String)	Method used to compute the mean of the entries of the partitions of the Sylvester subresultant matrix `None` **:** No mean is computed and coefficients of polynomials $f(x)$, $g(x)$ and $h(x)$ are not normalised. `Geometric Mean My Method` **:** My method for computing the geometric mean using a reduced algorithm. `Geometric Mean Matlab Method` **:** Standard Matlab method for computing the geometric mean of a set of values. `Arithmetic Mean` **:**
RANK_REVEALING_METRIC		

`GCD_COEFFICIENT_METHOD`	(String)	Coefficients of the GCD $d(x)$ can be approximated in two ways: `ux and vx` : $$\begin{bmatrix} C_t(u(x)) \\ C_t(v(x)) \\ C_t(w(x)) \end{bmatrix} \mathbf{d}_t = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \\ \mathbf{h} \end{bmatrix} \quad (1)$$ `ux` $$C_t(u(x))\mathbf{d}_t = \mathbf{f} \quad (2)$$
`MAX_ERROR_SNTLN`	(Float)	Level at which the low rank approximation method terminates.
`MAX_ITERATIONS_SNTLN`	(Int)	Maximum number of iterations of low rank approximation method.
`MAX_ERROR_APF`	(Float)	Level at which the APF method terminates.
`MAX_ITERATIONS_APF`	(Int)	Maximum number of iterations of APF method.