Module Interface Specification for ROC: Software estimating the radius of convergence of a power series

John M Ernsthausen

December 25, 2020

1 Revision History

Date	Version	Notes
22 November 2020	1.0	First submission
24 December 2020	2.0	Second submission

2 Symbols, Abbreviations and Acronyms

Symbols, abbreviations, and acronyms applicable to ROC are enumerated in Section 1 of the Software Requirements Document (SRS) (Ernsthausen, 2020).

Contents

3 Introduction

The following document details the Module Interface Specifications for the implemented modules in library software ROC estimating the radius of convergence of a power series. It is intended to ease navigation through the program for design and maintenance purposes.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at the project's GitHub repository.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by ROC.

Data Type	**Notation**	**Description**
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)
real vector	$\mathbb{R}^d$	the real d-dimensional vector space

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding Module	
Behaviour-Hiding Module	In Out Params Real Pole Complex Pair Poles Top Line Analysis ROC
Software Decision Module	Solver

Table 1: Module Hierarchy

4

6 MIS of In

The secrets of this module are input acquisition via software (R1), validate input format (R2), validate input type (R3), inputs should satisfy the assumptions (R4), and inputs should be scaled to prevent overflow/underflow (R5). This module handles the data structure for input coefficients and the scaling of the input coefficients, how the values are input, and how the values are verified.

This module is an invarient for the inputs of the modules ROC, Real Pole, Complex Pair Poles, and Top Line Analysis. This module has no way of knowing that the user satisfied the assumptions of the invarient. For example, this module cannot tell the scale on the coefficients. However poorly scaled coefficients may lead to overflow/underflow which would cause ROC to fail.

6.1 Module

In

6.2 Uses

None

6.3 Syntax

Name	**In**	**Out**	**Exceptions**
coeffs	$\mathbb{R}^N$	$\mathbb{R}^N$	
scale	$\mathbb{R}$	$\mathbb{R}$	SCALENEG

6.4 Semantics

The calling sequence should be consistent across the modules ROC, Real Pole, Complex Pair Poles, and Top Line Analysis.

exception: exec := $scale < 0 \Rightarrow$ SCALENEG

6.4.1 Environment Variables

None

6.4.2 Assumptions

The coefficients must be ordered according to the convention from lowest order to highest order.

6.4.3 Access Routine Semantics

None

7 MIS of Out

The secrets of this module are output via software (R6), validate output format (R7), validate output type (R8), This module handles the data structure for output coefficients, how the values are output, and how the values are verified.

7.1 Module

Out

7.2 Uses

None

7.3 Syntax

Name	In	Out	Exceptions
R_c		$\mathbb{R}$	
μ		$\mathbb{R}$	
fitting error		$\mathbb{R}$	
truncation error		$\mathbb{R}$	

7.4 Semantics

The calling sequence should be consistent among ROC, Real Pole, Complex Pair Poles, and Top Line Analysis.

7.4.1 Environment Variables

None

7.4.2 Assumptions

This module is an invarient for the outputs of the modules ROC, Real Pole, Complex Pair Poles, and Top Line Analysis. The calling sequence should be consistent among all these modules.

7.4.3 Access Routine Semantics

None

8 MIS of Params

The secrets of this module are parameter acquisition (R9), format (R10), type (R11), distribution (R12), and constraints (R13).

8.1 Module

Params

8.2 Uses

None

8.2.1 Exported Constants

Name	In	Out	Constraints
MINTERMS	$\mathbb{N}$		must be greater or equal to 10
3TAnUses	$\mathbb{N}$		must be less than the size of the coefficient vector minus MINTERMS
6TAnUses	$\mathbb{N}$		must be less than the size of the coefficient vector minus MINTERMS
TLAnUses	$\mathbb{N}$		must be less than the size of the coefficient vector minus MINTERMS
TLAkStart	$\mathbb{N}$		must be greater or equal to 10
TOL	$\mathbb{R}$		must be positive

8.2.2 Assumptions

These parameters will not change. The author is thinking of these parameters in a header file.

9 MIS of Solver

Algorithm to find the distance to the nearest real pole (R14), nearest complex conjugate pair of poles (R15), and nearest pole in hard to resolve case (R16). This solver is required to solve the optimization problem

$$\beta = \min_{x \in \mathbb{R}^2} \| x\|_2^2 \quad \text{such that} \quad Wx = b. \tag{1}$$

Here W is an over determined matrix and b is in its range.

9.1 Module

Solver

9.2 Uses

None

9.3 Syntax

The syntax depends on the algorithm used.

9.4 Semantics

The semantics depends on the algorithm used.

9.4.1 Environment Variables

None

9.4.2 Assumptions

Efficiently solves the optimization presented in the Instance Modules for Real Pole, Complex Pair Poles, and Top Line Analysis.

9.4.3 Access Routine Semantics

None

10 MIS of Real Pole

Find the distance to the nearest real pole (R14).

10.1 Module

Real Pole

10.2 Uses

In Section 6, Out Section 7, Params Section 8, Solver Section 9

10.3 Syntax

Name	In	Out	Exceptions
threeterm	$\mathbb{R}^N$, $\mathbb{R}$	$\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$	RCEXCEPTION, MUEXCEPTION

10.4 Semantics

coeff is coefficients of truncated power series, h is scaling, R_c is radius of convergence, fError is fitting error, and tError is truncation error.

$\text{threeterm}(\text{coeff}, h, R_c, \mu, \text{fError}, \text{tError})$

exception: RCEXCEPTION := $(R_c = h/\beta(1)) == \text{NaN}$

exception: MUEXCEPTION := $(\mu = \beta(2)/\beta(1)) == \text{NaN}$

NaN is Not A Number after the division is computed.

10.4.1 Environment Variables

None

10.4.2 Assumptions

None

10.4.3 Access Routine Semantics

None

11 MIS of Complex Pair Poles

Find the distance to the nearest complex conjugate pair of poles (R15).

11.1 Module

Complex Pair Poles

11.2 Uses

In Section 6, Out Section 7, Params Section 8, Solver Section 9

11.3 Syntax

Name	In	Out	Exceptions
sixterm	$\mathbb{R}^N$, $\mathbb{R}$	$\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$	RCEXCEPTION, MUEXCEPTION, BETA4EXCEPTION, COSEXCEPTION

11.4 Semantics

coeff is coefficients of truncated power series, h is scaling, R_c is radius of convergence, fError is fitting error, and tError is truncation error.

sixterm(coeff, h, R_c, μ, fError, tError)

exception: BETA4EXCEPTION := $\beta(4) < 0$

exception: RCEXCEPTION := $(R_c = h/\beta(1)) ==$ NaN

Set $s_1 = \beta(1)/\beta(2)$ and $s_2 = \beta(3)/\beta(4)$
exception: MUEXCEPTION := $s_1 ==$ NaN and $s_1 ==$ NaN
$(\mu = s_2) := s_1 ==$ NaN and $\neg(s_1 ==$ NaN$)$
$(\mu = s_1) := \neg(s_1 ==$ NaN$)$ and $s_1 ==$ NaN
$(\mu = (s_1 + s_2)/2) := \neg(s_1 ==$ NaN$)$ and $\neg(s_1 ==$ NaN$)$

exception: COSEXCEPTION := $\beta(2)/R_c < -1$ or $\beta(2)/R_c > 1$

NaN is Not A Number after the division is computed.

11.4.1 Environment Variables

None

11.4.2 Assumptions

None

11.4.3 Access Routine Semantics

None

12 MIS of Top Line Analysis

Find the distance to the nearest complicated pole (R16).

12.1 Module

Top Line Analysis

12.2 Uses

In Section 6, Out Section 7, Params Section 8, Solver Section 9

12.3 Syntax

Name	In	Out	Exceptions
topline	$\mathbb{R}^N$, $\mathbb{R}$	$\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$	

12.4 Semantics

coeff is coefficients of truncated power series, h is scaling, R_c is radius of convergence, fError is fitting error, and tError is truncation error.

$\text{topline}(\text{coeff}, h, R_c, \mu, \text{fError}, \text{tError})$

12.4.1 Environment Variables

None

12.4.2 Assumptions

None

12.4.3 Access Routine Semantics

None

13 MIS of ROC

Find the Radius of convergence R_c and order of singularity μ.

13.1 Module

ROC

13.2 Uses

In Section 6, Out Section 7, Real Pole Section 10, Complex Pair Poles Section 11, Top Line Analysis Section 12

13.3 Syntax

Name	In	Out	Exceptions
roc	$\mathbb{R}^N$, $\mathbb{R}$	$\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$	KSTARTEXCEPTION, NUSEEXCEPTION

13.4 Semantics

coeff is coefficients of truncated power series, h is scaling, R_c is radius of convergence, fError is fitting error, and tError is truncation error.

$\text{roc}(\text{coeff}, h, R_c, \mu, \text{fError}, \text{tError})$

Recall N is the number of coefficients.
exception: KSTARTEXCEPTION := Cannot construct linear system
exception: NUSEEXCEPTION := $N -$ TLAkStart $<$ TLAnUses
exception: NUSEEXCEPTION := $N -$ 3TAnUses $<$ MINTERMS
exception: NUSEEXCEPTION := $N -$ 6TAnUses $<$ MINTERMS

13.4.1 Environment Variables

None

13.4.2 Assumptions

None

13.4.3 Access Routine Semantics

None

References

J.M. Ernsthausen. Software requirements specification for ROC: Software estimating the radius of convergence of a power series. https://github.com/JohnErnsthausen/roc/blob/master/docs/SRS/SRS.pdf, 2020.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.