



Cairo University

A TRUST REGION APPROACH WITH MULTIVARIATE PADÉ MODEL FOR THE OPTIMIZATION OF REGULAR AND FRACTIONAL ORDER CIRCUITS

By

Shaimaa Ebid Kamel Ebid

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Engineering Mathematics

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

A TRUST REGION APPROACH WITH MULTIVARIATE
PADÉ MODEL FOR THE OPTIMIZATION OF REGULAR
AND FRACTIONAL ORDER CIRCUITS

By
Shaimaa Ebid Kamel Ebid

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Engineering Mathematics

Under the Supervision of

Prof. Dr. Hany L. Abdel-Malek

Professor of Engineering Mathematics
Engineering Mathematics Department
Faculty of Engineering, Cairo University

Dr. Ahmed S. A. Mohamed

Assistant Professor
Engineering Mathematics Department
Faculty of Engineering, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

A TRUST REGION APPROACH WITH MULTIVARIATE
PADÉ MODEL FOR THE OPTIMIZATION OF REGULAR
AND FRACTIONAL ORDER CIRCUITS

By
Shaimaa Ebid Kamel Ebid

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Engineering Mathematics

Approved by the
Examining Committee

Prof. Dr. Hany L. Abdel-Malek, Thesis Main Advisor

Prof. Dr. Mohamed F. Abu-Elyazeed, Internal Examiner

Prof. Dr. Hassanein H. Amer, External Examiner
Electronics and Electrical communications Department
Faculty of Engineering, American University in Cairo (AUC), Cairo, Egypt

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016

Engineer's Name: Shaimaa Ebid Kamel Ebid
Date of Birth: 26/09/1990
Nationality: Egyptian
E-mail: Eng_shaimaaebid@yahoo.com
Phone: 012-85429070
Address: Giza
Registration Date: 1/10/2013
Awarding Date:/....../2016
Degree: Master of Science
Department: Engineering Mathematics and Physics



Supervisors:

Prof. Dr. Hany L. Abdel-Malek
Dr. Ahmed S. A. Mohamed

Examiners:

Prof. Dr. Hany L. Abdel-Malek (Thesis main advisor)
Prof. Dr. Mohamed F. Abu-Elyazeed (Internal examiner)
Prof. Dr. Hassanein H. Amer (External examiner)
Professor at Electronics and Electrical communications
Department
Faculty of Engineering, American University in Cairo
(AUC), Cairo, Egypt.

Title of Thesis:

A Trust Region Approach with Multivariate Padé Model for the Optimization of
Regular and Fractional Order Circuits

Key Words:

Padé model; Trust region optimization; Sensitivity analysis; Yield optimization;
Fractional order circuits

Summary:

A new model based on Padé approximation for trust region optimization is proposed, tested and compared with the commonly used quadratic model. Sensitivity of output voltage is derived using a matrix approach. This approach allows getting the expressions for the output voltage sensitivity to fractional order powers. Yield is maximized using the proposed derivative free optimization method (DFO) with a reduced computational effort. A starting minimax point using gradient optimization technique reduces the required number of yield evaluations to achieve the optimum.

Acknowledgments

All praise is due to Allah.

I would like to express my deep gratefulness to my supervisors Prof. Hany L. Abdel-Malek and Dr. Ahmed S. A. Mohamed for their guidance, supervision and excellent advice throughout the work of this thesis.

I want to express as well my appreciation to all my colleagues for their encouragement. Special thanks to Eng. Wafaa Saber, Eng. Mohamed Fouda, Eng. Ahmed El-Sayed and Eng. Mahmoud Ayad.

Table of Contents

ACKNOWLEDGMENTS.....	I
TABLE OF CONTENTS.....	II
LIST OF TABLES.....	IV
LIST OF FIGURES.....	V
ABSTRACT	X
CHAPTER 1 : INTRODUCTION	1
1.1. DERIVATIVE FREE OPTIMIZATION METHODS	1
1.2. MOTIVATION.....	2
1.3. ORGANIZATION OF THE THESIS.....	3
CHAPTER 2 : TRUST REGION OPTIMIZATION.....	4
2.1. INTRODUCTION.....	4
2.2. AN OUTLINE OF THE PROPOSED METHOD.....	5
CHAPTER 3 : THE PROPOSED MULTIVARIATE PADÉ MODEL	9
3.1. LINEAR MODEL.....	9
3.2. QUADRATIC MODEL BY INTERPOLATION.....	10
3.2.1. Powell’s quadratic interpolation methods	10
3.2.2. Conn and Toint quadratic interpolation methods	11
3.3. QUADRATIC MODEL BY FITTING	12
3.4. PADÉ APPROXIMATION	14
3.4.1. Univariate Padé approximation	14
3.4.1.1. Padé approximation of e^x	15
3.4.2. Multivariate Padé approximation	16
3.5. THE PROPOSED MELTIVARIATE PADÉ MODEL.....	18
3.6. THE TRUST REGION SUB-PROBLEM.....	20
3.6.1. FINDING A STEP SIZE.....	23
3.7. THE COMPLETE ALGORITHM.....	24
3.8. NUMERICAL RESULTS FOR THE PROPOSED METHOD.....	26
3.8.1. Polynomial type function examples	26
3.8.1.1. A 2D quadratic numerical example	26
3.8.1.2. A 2D Rosen Brock banana numerical example	34
3.8.1.3. A 2D Beale numerical example.....	39
3.8.1.4. A 2D cube numerical example	42
3.8.1.5. A 4D Powell sigular numerical example	46
3.8.1.6. A 3D box numerical example.....	48
3.8.2. Rational type function examples	51
3.8.2.1. Example (1)	51
3.8.2.2. Example (2)	55
3.8.2.3. Example (3)	59
3.8.3. Comparison between birational and quadratic models for rational functions	63

CHAPTER 4 : SENSITIVITY ANALYSIS OF CIRCUITS WITH FRACTIONAL ORDER ELEMENTS	66
4.1. THE FRACTIONAL ORDER ELEMENTS.....	66
4.1.1. Fractional derivative.....	66
4.1.2. Fractional elements.....	67
4.2. SENSITIVITY ANALYSIS.....	68
4.2.1. Illustrative example	72
CHAPTER 5 : YIELD ESTIMATION AND MAXIMIZATION.....	73
5.1. YIELD ESTIMATION.....	73
5.2. THE GEOMETRICAL APPROACH.....	75
5.2.1. The simplicial approximation method.....	76
5.2.2. The quadratic approximation method.....	76
5.2.3. The ellipsoidal technique for design centering.....	78
5.3. THE STATISTICAL APPROACH.....	79
CHAPTER 6 : PRACTICAL APPLICATIONS	80
6.1. LOW-PASS FILTER DESIGN EXAMPLES.....	80
6.1.1. Passive filter example.....	80
6.1.2. Active filter example	82
6.2. BAND-PASS FILTER DESIGN EXAMPLES.....	86
6.2.1. Single stage LC circuit	86
6.2.2. T-shape circuit.....	89
6.2.3. π -shape circuit.....	92
CHAPTER 7 : DISCUSSION AND CONCLUSION.....	100
REFERENCES	101

List of Tables

Table 3.1: Results of $f_1(\mathbf{x})$	27
Table 3.2: Results of $f_2(\mathbf{x})$	27
Table 3.3: Results of the 2D Rosen Brock banana example.....	34
Table 3.4: Results of the 2D Beale example.....	39
Table 3.5: Results of the 2D Cube example.....	42
Table 3.6: Results of the 4D Powell Singular example.....	46
Table 3.7: Results of the 3D box example.....	48
Table 3.8: Results of the illustrative example (1).....	51
Table 3.9: Results of the illustrative example (2).....	55
Table 3.10: Results of the illustrative example (3).....	59
Table 3.11: Results of the illustrative example (2) for quadratic model.....	63
Table 3.12: Results of the illustrative example (3) for quadratic model.....	64
Table 4.1: $(\partial y_{ij} / \partial \varphi)$ for different parameters.....	71
Table 4.2: Sensitivity based on calculated and difference formula for the illustrative example.....	72
Table 4.3: The sensitivity of absolute output voltage and the sensitivity of phase output voltage with respect of φ for illustrative example.....	72
Table 6.1: Results for regular and fractional order elements in LC filter.....	81
Table 6.2: Rate of change in objective function for low pass LC filter.....	82
Table 6.3: Results for active filter.....	83
Table 6.4: Yield results for active filter.....	85
Table 6.5: Yield results for active filter with tolerance 15%.....	86
Table 6.6: Results for regular and fractional order elements in single stage LC.....	87
Table 6.7: Rate of change in objective function in single stage LC filter.....	88
Table 6.8: Results for regular and fractional order elements in T-shape circuit.....	90
Table 6.9: Rate of change in objective function in T-shape circuit.....	91
Table 6.10: Yield results for T-shape filter.....	91
Table 6.11: Results for regular and fractional order elements in π -shape circuit	92
Table 6.12: Rate of change in objective function in π -shape circuit.....	94
Table 6.13: Percentage of tolerance in parameters in π -shape circuit.....	95
Table 6.14: Yield results for π -shape filter.....	99

List of Figures

Figure 2.1: The general flow chart for trust region optimization	8
Figure 3.1: The complete flow graph.....	25
Figure 3.2: The result of the first simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$	28
Figure 3.3: The result of the first simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$	28
Figure 3.4: The result of the first simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	29
Figure 3.5: The result of the first simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	29
Figure 3.6: The result of the first simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -3 \\ 4 \end{bmatrix}$	30
Figure 3.7: The result of the first simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -3 \\ 4 \end{bmatrix}$	30
Figure 3.8: The result of the second simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$	31
Figure 3.9: The result of the second simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$	31
Figure 3.10: The result of the second simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.5 \end{bmatrix}$	32
Figure 3.11: The result of the second simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.5 \end{bmatrix}$	32
Figure 3.12: The result of the second simple quadratic function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -1.5 \\ -2 \end{bmatrix}$	33
Figure 3.13: The result of the second simple quadratic function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -1.5 \\ -2 \end{bmatrix}$	33
Figure 3.14: The result of the 2D Rosen Brock banana function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$	35
Figure 3.15: The result of the 2D Rosen Brock banana function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$	35
Figure 3.16: The result of the 2D Rosen Brock banana function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.2 \end{bmatrix}$, with $\varepsilon = 0.0001$	36
Figure 3.17: The result of the 2D Rosen Brock banana function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.2 \end{bmatrix}$, with $\varepsilon = 0.0001$	36

Figure 3.18: The result of the 2D Rosen Brock banana function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.2 \end{bmatrix}$, with $\varepsilon = 0.001$	37
Figure 3.19: The result of the 2D Rosen Brock banana function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0.25 \\ 0.2 \end{bmatrix}$, with $\varepsilon = 0.001$	37
Figure 3.20: The result of the 2D Rosen Brock banana function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	38
Figure 3.21: The result of the 2D Rosen Brock banana function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	38
Figure 3.22: The result of the 2D Beale function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	40
Figure 3.23: The result of the 2D Beale function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	40
Figure 3.24: The result of the 2D Beale function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -1.2 \\ -1 \end{bmatrix}$	41
Figure 3.25: The result of the 2D Beale function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -1.2 \\ -1 \end{bmatrix}$	41
Figure 3.26: The result of the 2D Beale function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ with $\varepsilon = 0.001$	43
Figure 3.27: The result of the 2D Beale function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ with $\varepsilon = 0.001$	43
Figure 3.28: The result of the 2D Beale function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ with $\varepsilon = 0.0001$	44
Figure 3.29: The result of the 2D Beale function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ with $\varepsilon = 0.0001$	44
Figure 3.30: The result of the 2D Beale function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$	45
Figure 3.31: The result of the 2D Beale function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$	45
Figure 3.32: The result of the 4D Powell Singular function versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 3 \\ -1 \\ 0 \\ 1 \end{bmatrix}$	47
Figure 3.33: The result of the 4D Powell Singular function versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 3 \\ -1 \\ 0 \\ 1 \end{bmatrix}$	47

Figure 3.34: The result of the 3D box versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 1 \\ 9 \\ 0 \end{bmatrix}$	49
Figure 3.35: The result of the 3D box versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 1 \\ 9 \\ 0 \end{bmatrix}$	49
Figure 3.36: The result of the 3D box versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$	50
Figure 3.37: The result of the 3D box versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$	50
Figure 3.38: The result of example (1) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	52
Figure 3.39: The result of example (1) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	52
Figure 3.40: The result of example (1) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	53
Figure 3.41: The result of example (1) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	53
Figure 3.42: The result of example (1) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	54
Figure 3.43: The result of example (1) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	54
Figure 3.44: The result of example (2) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	56
Figure 3.45: The result of example (2) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	56
Figure 3.46: The result of example (2) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	57
Figure 3.47: The result of example (2) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	57
Figure 3.48: The result of example (2) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$	58
Figure 3.49: The result of example (2) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$	58
Figure 3.50: The result of example (3) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	60

Figure 3.51: The result of example (3) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	60
Figure 3.52: The result of example (3) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix}$	61
Figure 3.53: The result of example (3) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix}$	61
Figure 3.54: The result of example (3) versus number of iterations (k) starting with $x_{ini} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$	62
Figure 3.55: The result of example (3) versus number of function evaluations (N) starting with $x_{ini} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$	62
Figure 3.56: The result of example (2) versus number of iterations (k) in case of the rational model and quadratic model starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	63
Figure 3.57: The result of example (2) versus number of function evaluations (N) in case of the rational model and quadratic model starting with $x_{ini} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$	64
Figure 3.58: The result of example (3) versus number of iterations (k) in case of the rational model and quadratic model starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	65
Figure 3.59: The result of example (3) versus number of function evaluations (N) in case of the rational model and quadratic model starting with $x_{ini} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$	65
Figure 4.1: Illustration of circuit elements considered. (a) A passive element with admittance y_{ij} . (b) A voltage controlled current source with a transconductance g_m	69
Figure 4.2: Low-pass LC filter.....	72
Figure 5.1: A two dimensional representation of 7 Latin hypercube samples [61].....	75
Figure 6.1: Low-pass filter response for regular and fractional order elements.....	81
Figure 6.2: Active filter.....	82
Figure 6.3: Active filter using integer order elements [35].....	83
Figure 6.4: Low-pass filter response for active filter.....	84
Figure 6.5: Low-pass filter response after relaxation of the constraints in active filter.....	84
Figure 6.6: The response of active filter for 100 random numbers at x_{final}	85
Figure 6.7: Single stage LC circuit.....	86
Figure 6.8: Band-pass filter response for regular and fractional order elements in single stage LC.....	87
Figure 6.9: Band-pass filter response for fractional order elements after relaxation of the constraints in the case of the single stage LC filter.....	88
Figure 6.10: T-shape circuit.....	89
Figure 6.11: Band-pass filter response for regular and fractional order elements in case of the T-shape circuit.....	89
Figure 6.12: Band-pass filter response with relaxed constraints for T-shape circuit...	90
Figure 6.13: π -shape circuit.....	92
Figure 6.14: Band-pass filter response for regular and for fractional order elements in the case of π -shape circuit.....	93
Figure 6.15: Band-pass filter response with relaxed constraints for π -shape circuit...	94
Figure 6.16: Band-pass filter response with relaxed constraints after adding maximum tolerance of α_1 for π -shape circuit.....	96

Figure 6.17: Band-pass filter response with relaxed constraints after adding maximum tolerance of β_1 for π -shape circuit.....	97
Figure 6.18: Band-pass filter response with relaxed constraints after adding maximum tolerance of β_4 for π -shape circuit.....	97
Figure 6.19: Band-pass filter response with relaxed constraints after adding maximum tolerance for all parameters for π -shape circuit.....	98

Abstract

Optimization is very important to find optimal nominal values of the designable system parameters. The system is required to satisfy the design specifications as well as not to be too sensitive to parameter variations. Parameter variations can be due to noise or unavoidable statistical fluctuations in the fabrication process. Usually the objective function defined by the system specifications is computationally expensive. Since the optimization process requires a significant number of function evaluations, it is recommended to represent the objective function by building up a model that approximates the objective function within a certain trust region. Many models are used among them linear and quadratic models. In this thesis, the objective function is approximated by building rational models called multivariate Padé model over a sequence of trust regions. The multivariate Padé model is constructed by using data points of $O(n)$, where n is the number of design parameters. The proposed approach is tested by applying it to several bench mark problems.

In case of the fractional order circuits, the sensitivities are derived based on a matrix approach. An adjoint matrix approach is used in the derivation of the sensitivities in this thesis. The sensitivity with respect to the fractional derivative orders α and β are also derived. The use of fractional order elements instead of regular integer elements enhances a better circuit performance. Optimal design using the derived sensitivity can be obtained using efficient gradient optimization techniques.

The yield is defined as the probability that a design satisfies the specifications. It is difficult process to calculate as yield is represented as a statistical function. Therefore, it is desired to obtain a good starting point for yield optimization process. The proposed method with gradient optimization technique is used to obtain this starting point.

The sensitivity analysis of circuits with fractional order elements is illustrated by applying it to practical circuits of active and passive filters with different topologies. The yield is calculated and optimized by the proposed derivative free method.

Chapter 1 Introduction

Engineering systems performance is optimized by the adjustment of a set of designable parameters such that predefined design specifications are satisfied. The optimal engineering design is selected by comparing different acceptable designs. The solution of this optimization problem is the optimal design. The objective function is computed through excessive simulations in many engineering systems. The computational cost of the objective function evaluations could be very high. Therefore, one of the design objectives is to decrease the number of function evaluations.

1.1. Derivative Free Optimization Methods

These methods use only function values and can be classified to: direct search methods [38], heuristic methods [29, 31], and derivative free trust region methods [20, 32, 42-45]. The direct search method and the heuristic methods require many function evaluations to improve the current iterate when close to the optimal point. The most important used methods are the derivative free trust region methods. Trust region is used in optimization to denote the objective function of a region within a space. These methods construct model to approximate the objective function in the neighborhood of a current iterate. This neighborhood is called the trust region. The approximating model can present the objective function well within this trust region without any derivatives calculated. The model is optimized and from region to region until the solution is obtained. There exist many such models such as linear, quadratic, etc..... In this thesis, Padé approximation is introduced for the first time for trust region optimization. Powell was the first one that solved constrained optimization problems using trust region. This method depends on the construction of the cheaper model to approximate the objective function in the neighborhood of a current trust region. The objective function and the constraints are described by linear multivariate interpolation model in his proposal. Powell then described unconstrained optimization problems by a quadratic multivariate interpolation model [43-45]. Conn and Toint [20] used the same algorithm with different criterion in selecting the points and modifying the model at each iteration. Marazzi and Nocedal [32] developed an algorithm that adds a geometric constraint to the trust region. The model is generated without calculating the derivatives by evaluating the function at a certain number of points. It is easy to maximize yield by using derivative free optimization since it needs function values only without using the derivatives.