Advances in Applied Mathematics 应用数学进展, **2023, 12(2), 634-644**
Published Online February 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.122067

求解不等式约束非线性优化问题的信赖域SQP方法

刘文杰，范喜东

云南财经大学统计与数学学院，云南 昆明

收稿日期：2023年1月21日；录用日期：2023年2月20日；发布日期：2023年2月27日

摘 要

针对不等式约束非线性优化问题求解上已经有了很多优化的方法，例如序列二次规划(**SQP**)方法，而本文则是在**SQP**优化方法的基础上研究并建立的一个将信赖域方法与其结合的方法——信赖域**SQP**方法，然后给出相应的实例对该算法进行数值实验模拟，并将其与原来的**SQP**方法进行比较。最后得出在适当的条件下，信赖域**SQP**方法相较于原有的**SQP**方法具有更好的数值效果，数值的结果也一定程度上验证了这个算法的可行性。

关键词

非线性优化问题，**SQP**优化方法，信赖域**SQP**优化方法

Trust Region SQP Method for Solving Inequality Constrained Nonlinear Optimization Problems

Wenjie Liu, Xidong Fan

School of Mathematics and Statistics, Yunnan University of Finance and Economics, Kunming Yunnan

文章引用: 刘文杰, 范喜东. 求解不等式约束非线性优化问题的信赖域SQP方法[J]. 应用数学进展, 2023, 12(2): 634-644. DOI: 10.12677/aam.2023.122067

Received: Jan. 21st, 2023; accepted: Feb. 20th, 2023; published: Feb. 27th, 2023

Abstract

There have been many optimization methods for solving inequality constrained nonlinear optimization problems, such as sequential quadratic programming (SQP) method. In this paper, a method combining confidence region method with SQP method is studied and established on the basis of SQP optimization method-confidence region SQP method. Then the corresponding examples are given to simulate the algorithm and compare it with the original SQP method. Finally, under appropriate conditions, the confidence region SQP method has better numerical results than the original SQP method, and the numerical results also verify the feasibility of this algorithm to a certain extent.

Keywords

Nonlinear Optimization Problems, SQP Optimization Method, SQP Optimization Method of Trust Region

1. 引言

随着现代化生产的发展和科学技术水平的进步, 最优化理论与方法日益受到人们的重视, 它已广泛应用于化工、航空、机械、建筑等许多工程技术部门, 也应用于生产、管理、军事、决策等方面. 最优化方法也已经成为一种重要的决策手段. 早在17世纪, 牛顿和莱布尼茨发明微积分的时代, 已经提出了函数的极值问题, 后来又出现了拉格朗日乘子法、柯西的最速下降法. 但是直到20世纪30年代最优化理论与方法才得到迅速的发展, 并不断完善逐步成为一门系统的学科. 1939年Hitchcock等人首先研究和应用了线性规划, 1947年Dantzig提出了单纯形法来求解线性规划问题, 为线性规划的理论和算法奠定了基础. 1951年Kunh等人完成了非线性规划的基础性工作. 到20世纪70年代最优化理论和方法在应用的深度上进一步得到了发展.

最优化计算方法一般分为两类: 一类是无约束优化问题, 一类是约束优化问题. 无约束最优化

计算方法不仅本身适应于大多数的实际应用, 而且与无约束优化问题有着紧密的联系. 一方面, 约束优化计算方法能够推广应用与无约束优化问题, 一方面, 约束优化问题能够转化成无约束优化问题来处理. 因此无约束优化的计算方法也可适应于约束优化问题, 这也是最优化问题的处理手段之一. 而约束优化问题常常也分为两类, 即等式约束和不等式约束问题. 而本文则主要针对不等式约束优化问题展开研究.

考虑如下不等式约束优化问题

$$\min \ f(x) \tag{1a}$$

$$\text{subject to} \ \ c(x) \leq 0 \tag{1b}$$

其中$f(x) : \mathbb{R}^n \to \mathbb{R}$和$c(x) : \mathbb{R}^n \to \mathbb{R}^m$是连续可微函数. 而序列二次规划方法则是求解问题(1)比较有效的一类算法. 该方法早期是由Powell等人提出, 由于它具有超线性收敛的一些良好性质, 吸引了许多学者对其进行研究和探讨, 使得SQP算法成为求解非线性约束优化问题最有效的算法之一. 随着SQP算法的日益完善, Levenberg与Marguart首次提出了信赖域SQP算法. 该算法主要是用来求解无约束优化问题的, 该算法具有很好的收敛性质和鲁棒性. 该算法不需要Hessian 矩阵$\nabla^2_{xx}\mathcal{L}_k p$正定; 可以在Hessian和Jacobi阵奇异的情形下控制迭代步的质量; 提供了一种强制全局收敛的机制; 它们其中的一些嵌套了IQP算法, 而另一些则与EQP算法结合. 近年来, 有许多学者也逐步在研究信赖域SQP的算法, 2003年Yamashite等人 [1]提出一种新的无惩罚函数的信赖域SQP 方法来解决非线性约束优化问题, 并证明了该方法的全局收敛性. 2011年Ridzal [2] 等人研究了一种带偏微分的等式约束条件的方程可以使用信赖域SQP方法来求解, 并用数值结果说明的这一方法的可行性. 2013年Heinkenschloss [3] 等人开发并分析了一种用于求解光滑等式约束优化问题的信赖域SQP 方法, 该方法允许线性系统的不精确和迭代解. 2014年孙中波等人 [4]讨论了信赖域SQP方法解决不等式约束优化问题, 并采用高阶校正的方法来克服算法产生的Maratos效应现象. 在适当的条件下, 证明了算法的全局收敛性和超线性收敛性. 2018年屠卿瑞等人 [5]提出了一种基于SQP信赖域算法的光伏单峰MPPT方法, 提高了光伏发电的效率. 2019 年zhang等人 [6]研究了信赖域SQP方法在非线性二阶锥规划中的应用以及在数学领域的应用. 同年Fletcher 等人 [7] 证明了一般非线性规划中信赖域SQP 算法的全局收敛性. 2020年孙泽等人 [8]首次研究提出了一种具有超线性收敛性质的信赖域SQP 方法, 并在双足机器人上实现, 在适当的条件下, 分析并验证了信赖域SQP 方法的超线性收敛性质.而本文通过给出相应的数值例子对SQP优化算法和信赖域SQP优化算法进行数值实验比较, 并分析两种优化方法对数值实验结果的影响, 最后总结归纳出具有更高效的优化方法.

本文的结构安排如下:第二节对最优化方法进行了介绍; 第三节通过给出的数值例子对该问题进行数值模拟并进行分析; 第四节对本文的工作做一个简单的总结和展望.

2. 最优化方法概述

本小节将对序列二次规划法及信赖域SQP方法的理论知识、基本思想及算法框架进行一个简单的回顾.

2.1. SQP方法

序列二次规划法(SQP, Sequential Quadratic Programming)算法 [9]是将复杂的非线性优化问题转换为较简单的二次规划问题来求解的算法. 而二次规划问题则是指目标函数为二次函数, 约束函数为线性函数的最优化问题. 二次规划问题是最简单的非线性优化问题, 有很多成熟的快速求解的方法. 而针对SQP算法的相关算例分析及收敛性分析可详见文献 [10].

2.1.1. 基本思想

非线性约束优化:

$$min \quad f(x)$$

$$s.t. \quad C_j(x) \geq 0 \quad for j = 0, 1, 2, 3 \cdots m$$

我们知道这个问题的拉格朗日函数$L(x,\mu) = f(x) - \sum_{j=1}^{m} \mu_j C_j(x)$, 现在需要做的就是在第$k$步的$x_k$和$\mu_k$的基础上, 找到一个方向, 使得$x_{k+1}$和$\mu_{k+1}$满足下面的$KKT$条件：

$$\begin{aligned}
&\nabla_x L(x,\mu) = 0\\
&C_j(x) \geq 0 \quad for j = 0, 1, 2, 3 \cdots m\\
&\mu \geq 0\\
&\mu_j C_j(x) = 0 \quad for j = 0, 1, 2, 3 \cdots m
\end{aligned}$$

离散后的写法为:

$$\nabla_x L(x_{k+1}, \mu_{k+1}) \approx \nabla_x L(x_k, \mu_k) + \nabla_x^2 L(x_k, \mu_k)\delta_x + \nabla_{x\mu}^2 L(x_k, \mu_k)\delta_\mu = 0 \tag{2}$$

$$C_j(x_k + \delta_x) \approx C_j(x_k) + \delta_x^T \nabla_x C_j(x_k \geq 0 \quad for j = 0, 1, 2, 3 \cdots m \tag{3}$$

$$\mu_{k+1} \geq 0 \tag{4}$$

$$[C_j(x_k + \delta_x^T \nabla_x C_j(x_k](\mu_{k+1})_j = 0 \quad for j = 0, 1, 2, 3 \cdots m \tag{5}$$

其中对朗格朗日函数的一阶导数和二阶导数为:

$$\nabla_x L(x_k, \mu_k) = \nabla_x f(x_k) - \sum_{j=1}^{m} (\mu_k)_j \nabla_x C_j(x_k) = g_k - A_k^T \mu_k \tag{6}$$

$$\nabla_x^2 L(x_k, \mu_k) = \nabla_x^2 f(x_k) - \sum_{j=1}^{m} (\mu_k)_j \nabla_x^2 C_j(x_k) = Y_k \tag{7}$$

因此上述KKT条件可以改写为:

$$Y_k \delta_x + g_k - A_k^T \mu_{k+1} = 0 \tag{8}$$

$$A_k \delta_x \geq -C_k \tag{9}$$

$$\mu_{k+1} \geq 0 \tag{10}$$

$$(\mu_{k+1})_j (A_k \delta_x + C_k)_j = 0 \quad for j = 0, 1, 2, 3 \cdots m \tag{11}$$

其中$C_k[C_1(x_k), C_2(x_k), \cdots C_m(x_k)]^T$.

2.1.2. SQP框架

假设在迭代(x^k, μ^k)处定义了二次规划:

$$\begin{aligned} min \quad & \frac{1}{2}\delta^T Y_k \delta + \delta^T g_k p \\ s.t. \quad & A_k \delta \geq -C_k \end{aligned} \tag{12}$$

我们可以通过求解这个二次规划问题, 得到δ, 再根据:

$$\widehat{\mu}_{k+1} = (A_k A_k^T)^{-1} A_k (Y_k \delta_x + g_k) \tag{13}$$

求得μ, 这样就得到了x和μ, 然后继续迭代求解, 直到满足终止条件即可.

2.1.3. SQP算法步骤

具体的迭代算法步骤如下:

[1]给定初始点(x_0, μ_0), 收敛精度ε, 置$k = 0$;
[2]计算Y_k, A_k, g_k, C_k的值;
[3]求解式(12)中的二次规划问题, 用式(13)计算朗格朗日乘子μ_{k+1}的值;
[4]迭代$x_{k+1} = x_k + \delta$, 如果收敛性满足, 则停止计算,
得到近似解(x_{k+1}, μ_{k+1});
[5]否则令$k = k + 1$, 转[2].

2.2. 信赖域SQP方法

2.2.1. 基本思想

考虑如下不等式约束优化问题

$$\min_p \quad f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \tag{14a}$$

$$\text{subject to} \;\; \nabla c_i(x_k)^T p + c_i(x_k) = 0, \;\; i \in \varepsilon, \tag{14b}$$

$$\nabla c_i(x_k)^T p + c_i(x_k) \geqslant 0, \;\; i \in \mathcal{I} \tag{14c}$$

$$\|p\| \leq \Delta_k \tag{14d}$$

其中∇f_k是目标函数$f(x)$的梯度；$\nabla_{xx}^2 \mathcal{L}_k p$是拉格朗日函数的Hessian矩阵；$\nabla c_i(x_k)$是约束函数的梯度.

但是这种方法可能会产生信赖域范围在约束条件之外的情况发生，会破坏我们使用信赖域约束最初的目的, 进一步, 这样的方法会影响算法的收敛性质.

另一种更加合理的看法是, 我们没有理由要每一步都精确满足满足线性约束, 而应当是在每一步改善可行性, 并仅在信赖域约束满足的前提下要求精确性.于是我们可以利用松弛法进行来满足以上猜想.

在迭代点 x_k, 我们通过求解以下子问题计算SQP算法的迭代步:

$$\min_{p} \quad f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \tag{15a}$$

$$\text{subject to} \quad A_k p + c_k = r_k \tag{15b}$$

$$\|p\|_2 \leq \Delta_k \tag{15c}$$

我们的目标是选取最小的使得约束相容的r_k，即求解以下子问题

$$\min_{p} \ \|A_k p + c_k\|_2^2 \tag{16a}$$

$$\text{subject to} \ \|v\|_2 \leq 0.8\Delta_k \tag{16b}$$

记此子问题的解为v_k 我们定义

$$r_k = A_k v_k + c_k \tag{17}$$

之后求解信赖域子问题得定义新迭代点$x_{k+1} = x_k + p_k$,再用最小二乘公式得新的乘子估计λ_{k+1} 然后再利用以下等式

$$p^B = -A_k^T [A_k A_k^T]^{-1} c_k \tag{18}$$

得到牛顿步p^B, 即目标函数的无约束极小点.

适用于此方法的一个价值函数为非光滑的$\mathcal{L}_2$函数，即$\phi_2(x;\mu) = f(x) + \mu\|c(x)\|_2$我们将其近似为(实际上也是从信赖域子问题得到的)

$$q_\mu(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p + \mu m(p) \tag{19}$$

其中

$$m(p) = \|c_k + A_k p\|_2 \tag{20}$$

其中惩罚因子的选取满足

$$\mu \geqslant \frac{\nabla f_k^T p_k + (\frac{\sigma}{2}) p_k^T \nabla_{xx}^2 \mathcal{L}_k p_k}{(1-\rho)\|c_k\|_1} \tag{21}$$

为评估p_k的可接受程度, 我们将计算比值

$$\rho = \frac{ared_k}{pred_k} = \frac{\phi_2(x_k,\mu) - \phi_2(x_k + p_k,\mu)}{q_\mu(0) - q_\mu(p_k)} \tag{22}$$

2.2.2. 信赖域SQP算法步骤

具体的迭代算法步骤如下:

[1]初始化. 任选精度ϵ和η, γ∈(0,1); 初始值x_0,信赖域算子$\delta_0 > 0$;

[2]for k=0,1,2... 分别计算出$f_k, c_k, \nabla f_k, A_k$;

[3]利用$\hat{\lambda}_k = (A_k A_k^T)^{-1} A_k \nabla f_k$ 得到$\hat{\lambda}_k$; 如果$\|\nabla f_k - A_k^T \hat{\lambda}_k\|_\infty < \epsilon$

并且$\|c_k\|_\infty < \epsilon$;停止运行并输出此时的$x_k$;否则转步骤四;

[4]利用(16-22)等式分别计算出$v_k; r_k; \nabla^2_{xx}\mathcal{L}_k; \mu_k; \rho_k$

[5]利用上文提到的最小二乘法公式(18) 算出p_k

若$\rho_k > \eta$,则$x_{k+1} = x_k + p_k; \Delta_{k+1} = 2\Delta_k$;

否则令$x_{k+1} = x_k$;找到满足$\Delta_{k+1} \le \gamma\|p_k\|$;再带入以上循环中直到满足条件.

3. 数值实验

在前面的章节中我们已经介绍了最优化理论中的两种优化计算方法, 本节我们将给出了几个实验例子来验证序列二次规划法和信赖域SQP优化方法的优劣性.

3.1. 数值例子及结果

在下面实验中所使用符号的含义:

x:非线性约束优化问题的自变量;

f:非线性约束优化问题的函数值;

cpu:非线性约束优化问题的迭代时间, 单位为t/s;

T:非线性约束优化问题的迭代次数k;

现在我们考虑如下几个非线性优化问题:

Table 1. Numerical results of trust region SQP algorithm
表 1. 信赖域SQP算法数值结果

x_1	x_2	f(x)	cpu	T
5.2396	3.7460	−79.8078	0.5216	11

Table 2. Numerical results of SQP algorithm
表 2. SQP算法数值结果

x_1	x_2	f(x)	cpu	T
5.2393	3.7455	−79.8063	0.8592	29

例子1:

$$\begin{aligned}&\min f(x)=x_1^2+x_2^2-16x_1-10x_2\\&\text{subject to}\quad -x_1^2+6x_1-4x_2+11\geqslant 0\\&\qquad\qquad\quad x_1^2-3x_2-e^{x_1-3}+1\geqslant 0\\&\qquad\qquad\quad x_1\geqslant 0\\&\qquad\qquad\quad x_2\geqslant 0\end{aligned}$$

我们针对上式不等式约束优化问题使用辅助计算工具MATLAB进行数值实验, 我们取初始值$x=(4,4)^T$, 代码是在MATLAB7.1环境下运行的. CPU是奔腾(R)2.19GHZ 内存. 实验结果如下表1 和表2所示.

例子2:

$$\begin{aligned}&\min f(x)=(x_1-3)^2+(x_2-1)^2\\&\text{subject to}\quad -x_1^2+x_2\geqslant 0\\&\qquad\qquad\quad 2x_1+x_2-3=0\end{aligned}$$

我们针对上式不等式约束优化例子2问题使用辅助计算工具MATLAB进行数值实验, 上式问题可知精确解为$x=(1,1), f=4$, 这里初始值我们取$x=(0,0)^T$, 代码是在MATLAB7.1 环境下运行的. CPU 是奔腾(R)2.19GHZ 内存. 实验结果如下表3和表4所示.

Table 3. Numerical results of trust region SQP algorithm
表 3. 信赖域SQP算法数值结果

x_1	x_2	f(x)	cpu	T
5.2396	3.7460	−79.8078	0.5216	11

Table 4. Numerical results of SQP algorithm
表 4. SQP算法数值结果

x_1	x_2	f(x)	cpu	T
5.2393	3.7455	−79.8063	0.8592	29

例子3:

$$\begin{aligned}&\min f(x)=1000(x_2-x_1^2)^2+(1-x_1)^2\\&\text{subject to}\quad 3x_1^2-5x_2\leq 0\\&\qquad\qquad\quad 3x_1-x_2-2\leq 0\end{aligned}$$

我们针对上式不等式约束优化例子3问题使用辅助计算工具MATLAB进行数值实验, 上式问题可知精确解为$x = (1, 1), f = 0$, 这里初始值我们取$x = (0, 0)^T$, 代码是在MATLAB7.1 环境下运行的. CPU 是奔腾(R)2.19GHZ 内存. 实验结果如下表5 和表6所示.

Table 5. Numerical results of trust region SQP algorithm
表 5. 信赖域SQP算法数值结果

x_1	x_2	f(x)	cpu	T
1	1	5.8129e−21	0.120058	44

Table 6. Numerical results of SQP algorithm
表 6. SQP算法数值结果

x_1	x_2	f(x)	cpu	T
1	1	1.9737e−9	0.719165	48

例子4:

$$
\begin{aligned}
\min f(x) &= x_1^2 + x_2^2 - 4x_1 + 4 \\
\text{subject to} \quad & -x_1^2 + x_2 - 1 \geqslant 0 \\
& x_1 - x_2 + 2 \geqslant 0 \\
& x_1 \geqslant 0 \\
& x_2 \geqslant 0
\end{aligned}
$$

我们针对上式不等式约束优化例子4问题使用辅助计算工具MATLAB进行数值实验, 我们取初始值$x = (0, 1)^T$, 代码是在MATLAB7.1环境下运行的. CPU是奔腾(R)2.19GHZ 内存. 实验结果如下表7 和表8所示.

Table 7. Numerical results of trust region SQP algorithm
表 7. 信赖域SQP算法数值结果

x_1	x_2	f(x)	cpu	T
0.5536	1.3064	3.7989	0.07509	7

3.2. 章末小结

针对不等式非线性优化问题的求解我们通过以上四个例子的数值结果进行分析, 例子1的数值结果中当信赖域SQP方法求得的解和原始SQP方法求得的解误差极小时, 原始方法所求得的解所花的计算时间比信赖域SQP方法的时间长. 而例子2和例子3中的数值结果表明, 在已经知道精确解的情况下, 我们所使用的原始SQP方法和信赖域SQP方法都能够求得所需要的解, 在确保解正确的情况下我们分析两种方法所花的计算时间发现, 信赖域SQP方法在达到同样计算结果的情况下计算时

Table 8. Numerical results of SQP algorithm
表 8. SQP算法数值结果

x_1	x_2	f(x)	cpu	T
0.5536	1.3064	3.7989	0.51732	7

间远远短于原始的SQP方法. 例子1、例子2及例子3的数值结果中我们还可以发现信赖域SQP方法较于SQP方法迭代次数少. 最后在例子4的数值实验结果中, 当信赖域SQP方法和SQP方法所求得的解相同, 函数值相同, 并且迭代次数也相同的情况下, 我们对比信赖域SQP方法和原始SQP方法的计算所需时间, 可以发现信赖域SQP方法较于原始SQP方法更快. 综上几个例子的数值实验结果分析得知, 信赖域SQP方法在一定条件下, 是有效并且高效率的算法.

4. 总结与展望

在论文中我们简单介绍了最优化的历史以及发展背景, 然后概述了求解非线性不等式约束优化问题最常用的一种SQP方法, 并建立研究了本文的主要思想: 使用信赖域方法与SQP 方法相结合的算法来求解非线性不等式约束优化问题. 最后在第三章中通过四个非线性不等式约束优化问题的数值例子对两种方法进行试验, 并呈现跟我们预想一致的数值结果. 通过这次实验结果分析得知, 在可以达到相同的数值解的情况下, 信赖域SQP方法较原始的SQP优化方法更有优势, 具有精度高, 计算效率快的特点, 因此在某些复杂的非线性不等式约束优化问题上我们可以采用信赖域SQP优化方法来求解, 会大大地节省运算时间.

本文在使用信赖域SQP优化方法求解非线性不等式约束优化问题上考虑到了一些问题, 如果在一些较为复杂的特殊方程中使用信赖域SQP 优化方法来求解, 它是否还能够有同样的优势. 因此, 接下来的工作我们会继续对信赖域SQP优化方法进行相关方面的研究.

参考文献

[1] Yamashita, H. and Yabe, H. (2007) A Globally Convergent Trust-Region SQP Method without a Penalty Function for Nonlinearly Constrained Optimization. Cooperative Research Report 168 “OPTIMIZATION: Modeling and Algorithms 17”, The Institute of Statistical Mathematics, Tokyo.

[2] Ridzal, D., Aguilo, M. and Heinkenschloss, M. (2011) Numerical Study of a Matrix-Free Trust-Region SQP Method for Equality Constrained Optimization. Office of Scientific and Technical Information Technical Reports.

[3] Heinkenschloss, M. and Ridzal, D. (2013) A Matrix-Free Trust-Region SQP Method for Equality Constrained Optimization. *SIAM Journal on Optimization*, **24**, 1507-1541. https://doi.org/10.1137/130921738

[4] 孙中波, 段复建, 许春玲, 等. 不等式约束优化超线性收敛的信赖域-SQP算法[J]. 2014, 37(5): 878-890.

[5] 屠卿瑞, 李一泉, 曾耿晖, 等. 一种基于SQP信赖域算法的光伏单峰MPPT方法[P]. 中国专利, CN107272815B. 2017-10-20.

[6] Zhang, X., Liu, Z. and Liu, S. (2012) A Trust Region SQP-Filter Method for Nonlinear Second-Order Cone Programming. *Computers and Mathematics with Applications: An International Journal*, **63**, 1569-1576.

[7] Fletcher, R., Gould, N., Leyffer, S., *et al.* (2002) Global Convergence of a Trust-Region SQP-Filter Algorithm for General Nonlinear Programming. *SIAM Journal on Optimization*, **13**, 635-659.

[8] Sun, Z., Zhang, B., Sun, Y., *et al.* (2020) A Novel Superlinearly Convergent Trust Region-Sequential Quadratic Programming Approach for Optimal Gait of Bipedal Robots via Nonlinear Model Predictive Control. *Journal of Intelligent and Robotic Systems*, **100**, 401-416.

[9] Nocedal, J., Wright, S.J., Mikosch, T.V., *et al.* (1999) Numerical Optimization. Springer, Berlin. https://doi.org/10.1007/b98874

[10] Zhang, H.P. and Ye, L.-Q. (2009) A Feasible SQP Descent Method for Inequality Constrained Optimization Problems and Its Convergence. *Chinese Quarterly Journal of Mathematics*, No. 3, 469-474.