

2007 年度卒業論文

シンプレックス法学習支援ソフトの提案

文教大学情報学部経営情報学科 学籍番号 A4P21176

根本ゼミナール 10 期生

和田 学

シンプレックス法学習支援ソフトの提案

和田 学

概要

本論文は、シンプレックス法と呼ばれる線形計画問題に対する解法の学習について、学習理解をソフト導入によって支援する研究を記したものである。

手法中の計算過程において、数値の流れがイメージしにくかったり、手法通りに導いた情報から解答が読み取れなかったりといった問題に対して、学習者は苦手意識を持つようだ。シンプレックス法は線形計画問題の最適解を求めるひとつの手法だが、その学習の一環としてシンプレックス法を紙面上で実行すると非常に手間が掛かり、それが学習者の躓き易い学習障壁となっている。このシンプレックス法の学習障壁に対する解消へのアプローチとして、支援ソフトの導入を試みた。機械的な作業である作図や計算行為にかかる労力をソフトが代行し、学習者の負担を緩和することで、学習の障壁を少しでも薄くしたい。本論文では支援ソフトのプログラムの骨格を考える過程と、実際の試作プログラムのソースを記す。今後の研究活動としては、支援ソフトを公開しているサイトを通じて、利用者との対話的に支援ソフトの改良をしていきたい。

概要

もくじ

第1章 はじめに

第2章 シンプレックス法の学習障壁

- 2-1 数理計画問題の紹介
- 2-2 シンプレックス法の紹介
- 2-3 学習者の悩み
- 2-4 現状の問題点

第3章 ソフトが支援できる範囲

- 3-1 コンピュータの特性
- 3-2 支援ソフトが行える作業
- 3-3 支援ソフト外で学習者に努力してもらえる部分

第4章 支援ソフトの制作

- 4-1 制作方針
- 4-2 開発環境
- 4-3 プログラムの流れ

第5章 ソフト制作の結果と、今後の研究活動

謝辞

参考文献

付録

シンプレックス法学習支援ソフトの提案

和田 学 著

第1章 はじめに

数理計画問題を解く手法を学習するとき、手法を手計算で実践することがある。実際に扱うときはコンピュータに実行させる程の複雑な手法ならば、手計算による作業は非常に手間の掛かるものとなるだろう。

シンプレックス法は線形計画問題の最適解を求める上で有効な手法であるが、これもコンピュータ上で実用されるほど、人の手で実行するには手間が掛かる。手法中で扱う計算は単純な加算と乗算の繰り返しであるが、計算量が多い。また、手法の実行対象の問題によっては作成する表が多くなり、広い紙面を要することがある。私自身も学習した経験があるが、作業の煩わしさのために学習に対する集中を欠くことがあった。もし回りくどい手間を省けたなら、その分だけ学習に集中できるようになり、より早い理解が期待できるはずである。

人の手による計算や作図の手間を省く方法としては、コンピュータソフトによる計算の代行が有効である。単純な処理を繰り返す事においては、精度も速度も人よりコンピュータの方が信頼できるからである。事実、シンプレックス法はコンピュータにプログラムできる手法なので、全ての計算をコンピュータソフトに代行させることは可能である。しかし、人の手で書いていたものをそのまま代行させるのなら、計算過程が紙面上で実践する手書きの形式で出力されるものが学習ソフトとして望ましい。

そういった既存のソフトが無いか探してみたが、シンプレックス法を実行するソフトは多々あっても、殆どは結果のみを出力するだけで、学習面で役に立つと思えるものは見つけれなかった。最適解を求める手法そのものは計算ソフトにツールとしてプログラムされていることが多くても計算過程を見る事ができないのは、計算結果を重要視して作られたソフトにとって、計算過程をわかりやすく出力する必要がないからであろう。そもそも計算ソフトの利点は、利用者が手法を知らなくても最適解を求めることができるという点にもある。

そこで、私はシンプレックス法の学習に役立つ学習支援ソフトを研究することにした。

研究の流れは、学習の妨げになっている具体的な問題を見つけ、それらからコンピュータソフトが支援するべきものを選別し、問題に対応する支援ソフトの機能を考え、それらを実装したものを作成する。

研究結果として、シンプレックス法学習支援ソフト“JESICA”を作成した。プログラムソースは付録に記す。主な機能は、シンプレックス表の表計算の代行である。表計算は

クリックのみの簡易操作で進める事ができ、利用者にはソフト利用環境（パソコン）に対して詳しい知識を持っている必要は無い。シンプレックス法そのものを理解するには、学習者は教科書や参考書を手に取らなければならないが、学習中に手法を実践する際に計算の手間を省くことが出来る。

本論文の構成は、研究の流れの詳細を追ったものである。本章では研究の流れを記した。第2章ではシンプレックス法の学習者の悩みを学習の問題点としてまとめ、支援すべき対象を考えるための足掛かりを示す。第3章ではコンピュータソフトの特性を思慮した上で、支援ソフトの支援方法を定める。第4章では、ソフト制作の計画に触れ、実装する機能やプログラムの流れを記す。第5章では、制作された実際の支援ソフトの効果を検証する方法について述べる。最終章である第6章では、研究の反省点と、今後の研究活動について記して本論文を終える。

第2章 シンプレックス法の学習障壁

シンプレックス法の学習を支援するにあたり、具体的にどのような点に対して支援してゆくのかを決定する足掛かりとして、学習における現状の問題点を示したい。まず、数理計画問題とシンプレックス法の概念を紹介してから、シンプレックス法の学習において支援ソフトを扱わない学習状況を見つめ、そこから学習の妨げとなっている問題点を学習障壁としてまとめる。学習障壁の解消を目指すことが支援ソフトの課題となる。

2-1 数理計画問題と線形計画法の紹介

数理計画の説明として、[1]によると

与えられた制約条件のもとで、ある一つの目的関数を最大あるいは最小にするという最適化手法

と述べられている。これを受けて、数理計画問題とは、実際問題を定式化することで最適解もしくはそれに近似する解を求める問題と言える。定式化とは、問題を数式化したもので、題中の目的を目的式、目的に対する条件を制約式（条件式）として記述する。

ある数理計画問題が、1次式の等式及び不等式による制約条件の下で、1次式の目的関数を最大化あるいは最小化する問題であるとき、それを線形計画問題という。また、線形計画問題の最適解を求める手法を線形計画法という。

2-2 シンプレックス法の紹介

シンプレックス法について[2]による解説を簡潔にまとめる。

シンプレックス法とは線形計画法のひとつで、制約式によって座標上で超多面体となる実行可能解の、ある頂点から目的関数値を出発し、目的関数の方向に従って実行可能解の頂

点を移動しながら最適解を求める手法である。実用上では高速に線形計画問題の最適解を求めることができ、多くの数理計画ソフトにプログラムされている。

2-3 学習者の悩み

数理計画問題の解法を理解しようとするなら、教科書に目を通しただけで応用までできてしまう人は少ないであろう。多くの人は、ペンとノート等を使い、教科書の演習問題を自分でこなすことで理解するものだ。

シンプレックス法についても同様の方法で学習できるが、手法の手順も表計算量も多い。このとき、学習者にはどのような問題意識が生まれるのかを、実際にシンプレックス法を勉強している学生や、学習経験のある学生、指導をしている教員を対象に、具体的な問題意識を聞き込むことにした。主な学習障壁を表2-1で列挙した。

2-4 現状の問題点

表2-1からシンプレックス法の学習における問題点をまとめてみると、手法の手順や計算量の多さから招く問題は、演習に時間を取られてしまったり、計算ミスが多くなることで手順が正しく実行できているか学習者がわからなくなったり、集中力を欠くといったものだった。ソフトの導入の有無にかかわらず、数学に弱い学習者にとってはシンプレックス法そのものの理解がしにくいといった旨の意見も目立つ。表計算をこなしてゆけば計算力はつくかもしれないが、手法の学習を手早く済ませたいなら、計算を自分の手で行うのはなるべく省きたい。

表2-1

シンプレックス法における学習の障壁	
<ul style="list-style-type: none">● 数値の動きがイメージできない● 計算過程でミスをしやすい● 用語が覚えにくい● 定式化ができない● 標準化ができない● 手法の実行結果から解答が読み取れない● シンプレックス表を描くのが面倒くさい	<ul style="list-style-type: none">● 表の計算回数を事前に知りたい● 計算に用いる紙面が大きい● 計算を間違えたとき、その箇所を探しづらい● 多次元の問題では退行現象に気づきにくい● ピボットが選びにくい● 手法の手順を忘れてしまう

第3章 ソフトが支援できる範囲

コンピュータソフトによる支援を行うなら、コンピュータに行わせるべき処理と、そうでない処理とを分けたうえで、コンピュータならではの支援を行いたい。支援ソフトは、どのような学習障壁に支援をしてゆけるのかを考える。

3-1 コンピュータの特性

コンピュータは計算機なので、数理計算や単純な繰り返し作業が得意だ。コンピュータソフトに行わせることができるのも単純な作業だけである。

単純な作業とは、決まった法則と形式に基づいた処理を言うものとし、逆に複雑な作業は毎回の課題に特別な処理をしなくてはいけないものとする。コンピュータに行えるのは、単純な計算の積み重ねに過ぎないので、プログラム上のアルゴリズムは全て単純な作業である。

3-2 支援ソフトが行える作業

単純な作業の組み合わせで実現できることでも、現代のコンピュータ程の計算速度があれば幅広い。四則演算をはじめ、方程式をグラフに描画したり、予め記憶されている情報を素早く読み込んだりできる。支援ソフトでこれらの作業を利用するなら、表3-1のような支援方法を考えることが出来る。支援対象は、表2-1で列挙した学習障壁の諸項目である。また、支援方法から期待できる支援効果も表3-2に示す。

表3-1

学習障壁に対する支援方法案	
障壁	支援方法
● 数値の動きがイメージできない	計算過程を表示する 二次元問題はグラフも表示される
● 計算過程でミスをしやすい	全ての計算を代行する
● 用語が覚えにくい	用語をまとめたものを直ぐに参照できるようにする
● 定式化ができない	既存の数理モデルへのデータ入力で定式化を代行する
● 標準化ができない	定式を自動で標準化する
● 手法の実行結果から解答が読み取れない	シンプレックス表の結果から読み取れる情報を解説したレポートを出力する
● シンプレックス表を描くのが面倒くさい	表の作成はソフトが代行する
● 表の計算回数を事前に知りたい	※なし
● 計算に用いる紙面が大きい	編集中のシンプレックス表とその前後だけを表示する
● 計算を間違えたとき、その箇所を探しづらい	ピボットの選択が正規の手順と異なっていたら、そのシンプレックス表に印をつけておく
● 多次元の問題では退行現象に気づきにくい	記録された過去のシンプレックス表と同じ表が出現した場合にソフトが退行現象を感知する
● ピボットが選びにくい	ピボットの選択を任意で制限できる

※シンプレックス表を更新する計算回数は事前に計算できない

表 3-2

支援方法案から支援効果が得られる理由	
支援方法案	支援効果
計算過程を表示する	シンプレックス表の計算を代行するにあたり、表の出力形式は手書きに近い見た目が望ましい。表中に計算過程を示し、また数値は分数表記にも対応する。学習者は目で式を追うだけで計算過程がわかる。
二次元問題はグラフも表示される	実行可能領域の端点を移動して最適解を求めるのがシンプレックス法である。その様子をグラフを交えて見る事で、解答までの流れをイメージできるようになる。
全ての計算を代行する	計算量の多い手法なので、そこに掛ける手間が省かれた分の労力と時間を学習に向けられる。
用語をまとめたものを直ぐに参照できるようにする	シンプレックス表や、実行結果などで専門用語が表示される事がある。文字の意味を解説してくれる辞書機能を持たせることで、参考書を見る手間が少し省ける。
既存の数理モデルへのデータ入力で定式化を代行する	問題を表現したものが定式なので、表現の仕方は学習者に依存するが、式を全て手打ちで入力するのも手間なので、入力を支援するツールとして実装。支援ソフトが出力した定式は定式化の解答例として利用できる。
定式を自動で標準化する	標準化の方法は参考書で学習するべきだが、支援ソフトの出力する標準形は標準化の解答例として利用できる。
シンプレックス表の結果から読み取れる情報を解説したレポートを出力する	シンプレックス表が最適解を示すか、その途中であるかに関わらず、表の各セルのデータが示す情報を解説したレポートを出力できるようにし、表の見方の参考となる。
表の作成はソフトが代行する	作図の手間が省ける。
編集中のシンプレックス表とその前後だけを表示する	紙面上で計算するのとは異なり、パソコンのディスプレイ上では必要な部分だけをスクロールして表示することができる。
ピボットの選択が正規の手順と異なっていたら、そのシンプレックス表に印をつけておく	学習の参考ため、ピボットは任意で自由に選択できる。その際、正規の手順とは異なった表を分かるようにしておくことで、誤った手順からの解答の派生を見る事ができる。
記録された過去のシンプレックス表と同じ表が出現した場合にソフトが退行現象を感知する	多次元問題においては、人の目では退行現象が分かり辛いときがある。ソフトが代わって感知してくれる。
ピボットの選択を任意で制限できる	学習の参考として、ピボットの選択に対する制限は任意で指定できる。非基底変数の全てを選択できるし、正規手順のみの選択もできる。

3-3 支援ソフト外で学習者に努力してもらえる部分

全ての機能を実装できたなら理想ではあるが、中には学習者自身の努力で補える部分もある。計算の速度面はコンピュータが代替すべきであるが、辞書機能やレポート機能などは教科書を片手に利用すれば、学習者側で解決できる。

表 3-2 から、学習者側で対処できる機能を表 3-3 に示す。これらは、制作活動において優先順位の低い実装項目となる。

表 3-3

実装しなくても良い機能	
支援方法案	学習者側の対処
用語をまとめたものを直ぐに参照できるようにする	支援ソフトは演習の手助けに、教科書は用語などの解説に利用する。
シンプレックス表の結果から読み取れる情報を解説したレポートを出力する	
既存の数理モデルへのデータ入力で定式化を代行する	例題や定式は学習者が既存のものを見つけて支援ソフトに入力する。

第 4 章 支援ソフトの制作

支援ソフトを制作するにあたり、開発環境やプログラムの流れ、インターフェイスのコンセプトなどを決める。

4-1 制作方針

支援ソフトは、なるべく多くの人が扱いやすいものでありたい。複雑な操作を必要とせず、また外観も難しくせず、利用者が馴染みやすいものが良い。

例えば、シンプレックス表の出力の仕方などは、なるべく手書きの形式に近いものを描画したいし、二次元の線形問題を読み込んだときは見やすいグラフを描画したい。利用者の声を受けてバージョンアップが図れるよう、ソースには将来性を持たせて制作しておくべきである。

4-2 開発環境

インターフェイスを綺麗に仕上げたいので、描画命令に強い言語を利用する。

プログラミング言語はHSP (Hot Soup Processor) を扱う。ONION software というグループが開発した、Windows 上で動作するスクリプト言語の一種で、マルチメディアを手軽に扱うことができる高水準言語である。BASIC 言語をベースにした簡潔な文法で、プログラミング初心者でもすぐに使い始めることができ、直感的にプログラムソースを記述できるのが強みだ。

ソフト中で扱う画像は自作する。描画ソフトは、私がい慣れた Photoshop を利用することにした。Adobe Systems 社が開発した画像編集ソフトで、その性能の良さから世界中に幅広く利用されている。

4-3 プログラムの流れ

ソフトにプログラムするソースの大まかな流れを図 4-1 に示す。図 4-1 中で示した各動作に実装する機能を表 4-2 で対応させた。これはそのままソフトの仕様書として、制作を進めることになる。

図 4-1

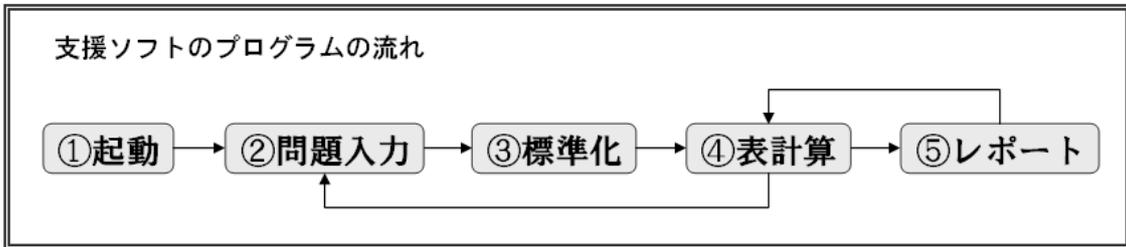


表 4-2

各動作に対応する機能・操作		
①	起動	【ソフトを起動】 <ul style="list-style-type: none"> ● 各種オプション設定を読み込む
②	問題入力	【問題を定式で入力】 <ul style="list-style-type: none"> ● 既存の数理モデルへのデータ入力で定式化を代行する ● 例題を選択して入力
③	標準化	【定式を標準化, 式の係数を元にシンプレックス表を作成】 <ul style="list-style-type: none"> ● 定式を自動で標準化する ● 表の作成はソフトが代行する
④	表計算	【ピボットをクリックで選択していただくの簡単計算】 <ul style="list-style-type: none"> ● 表の作成はソフトが代行する ● 全ての計算を代行する ● 計算過程を表示する ● 二次元問題はグラフも表示される ● 編集集中のシンプレックス表とその前後だけを表示する ● ピボットの選択を任意で制限できる ● ピボットの選択が正規手順と異なると印がつく ● 退行現象を感知する
⑤	レポート	【シンプレックス表の情報解説】 <ul style="list-style-type: none"> ● シンプレックス表の結果から読み取れる情報を解説したレポートを出力する
○	全般	【殆どの動作で利用可能】 <ul style="list-style-type: none"> ● 辞書機能を右クリックでポップアップ参照 ● 画面の印刷 ● ウィンドウの大きさを調整できる

第5章 ソフト制作の結果と、今後の研究活動

本来なら学習者に試用してもらい、学習効果を検証したかったが、支援ソフトは制作できたものの、試用の結果までをまとめることはできなかった。支援ソフトのプログラムソースは付録に記す。

支援ソフトに期待している効果は、シンプレックス法の学習者が手法を実践する段階で修得を諦めてしまうことを防ぐことである。数字を扱うことに苦手意識を持っている学習者が、本ソフトを利用することで少しでも学習環境が向上してほしい。

今後も制作を進めていくつもりである。現状ではまだソフトのインターフェイスに見辛い部分が多い。学習者の計算の代行をするわけだから、できる限り手書きの様なデザインのフォーマットを実現してゆく。

ソフトを配布するサイトを通して、利用者からの意見を聞きながら、より学習の役に立つソフトにしていきたい。

支援ソフト公開URL

<http://www.bunkyo.ac.jp/~nemoto/lecture/seminar2/2007/wada/>

謝辞

本研究活動にあたり、終始に渡り懇切な御指導を賜りました根本俊男教授へ深謝致します。また、根本研究室の皆様には研究発表や研究成果へ数々の御意見を頂き、誠に有難う御座いました。関係者方々、心より感謝の意を表します。

参考文献

[1] 坂和正敏；数理計画法の基礎，森北出版（1999）

[2] フリー百科事典 ウィキペディア シンプレックス法

<http://ja.wikipedia.org/wiki/%E3%82%B7%E3%83%B3%E3%83%97%E3%83%AC%E3%83%83%E3%82%AF%E3%82%B9%E6%B3%95>

付録 “JESICA” プログラムソース

```
////////////////////////////////////
://// " JESICA " //////////////////////////////////
:////////////////////////////////////
;_____Program Start
;_____System Init

;      System Value
PROGRAMNAME="JESICA ( simplex method support soft )"
FONTSTYLE=" M S   ゴシック "
FONTSIZE=12
EXTENSION="jpd"           ;Personal Extension (Jessica Problem Data)
VERSION="1.00"           ;Version
SYSTEMWAIT=30           ;System Wait Time
SYSTEMTIME=0           ;System Count Time
SYSTEMVIEW=0           ;System View Mode [ 0=Input , 1=Table , 2=Result ]
sdim FNAME, 1024       ;System File Name
sdim ENAME, 1024       ;System Edit Name
sdim SOM, 1024         ;System Operation Message
sdim SSTR, 128         ;System String
ddim SDV, 8           ;System Double Value
dim DBG, 64           ;System Debug Stat
DEBUGWINDOW=0         ;System Debug Window
SOMnom=0             ;System Operation Message Number

;      Key Cord : getkey
KeyNum=16
dim KEYcord, KeyNum
dim KEYsign, KeyNum
dim KEYpast, KeyNum
n=0:KEYcord.n=37       ;00. [←]
n+ :KEYcord.n=38       ;01. [↑]
n+ :KEYcord.n=39       ;02. [→]
n+ :KEYcord.n=40       ;03. [↓]
n+ :KEYcord.n=01       ;04. [LeftCrick]
n+ :KEYcord.n=02       ;05. [RightCrick]
n+ :KEYcord.n=09       ;06. [Tab]
n+ :KEYcord.n=16       ;07. [Shift]
n+ :KEYcord.n=17       ;08. [Ctrl]
n+ :KEYcord.n=18       ;09. [Alt]
n+ :KEYcord.n=32       ;10. [Space]
```

```

;          Memory
n=0:ID_MAIN=n          ;Buffer ID
n+ :ID_STORE=n        ;Buffer ID
n+ :ID_SYSTEM=n       ;Buffer ID
n+ :ID_TITLE=n        ;Buffer ID
n+ :ID_BACK=n         ;Buffer ID
n+ :ID_GRAPH=n        ;Buffer ID
n+ :ID_GRAPH_S=n     ;Buffer ID
wx=800:wy=600        ;Window Size
notesel EDTnote      ;Edit Note Selection

;          Data
sdim DATinput        ;Data Input
sdim DATcompile      ;Data Compile Strings
DATcmp=0            ;Data Compile
DATaim=0            ;Data Aim [0=Max,1=Min]
DATov=0             ;Data Ordered Variable

;          Effect
effect=64
n=effect
dim EFFcnd,n
dim EFFtyp,n
dim EFFcnt,n
dim EFFx,n
dim EFFy,n

;          Object
n=32
OBJnum=0           ;Object Number
dim OBJid,n        ;Object ID
dim OBJcnd,n       ;Object Condition

;          Etc.
CELL=32
ACC=100
sdim ATTENTION,1024

goto *BOOT

```

```

;_____Function (C)

```

```

#defcfunc JessicaComment int fjc_n

```

```

SOMnum=fjc_n

```

```
SSTR=""
```

```
if fjc_n=001:SSTR=" 新規作成：現在未実装の機能です。"  
if fjc_n=002:SSTR=" 読み込み：現在未実装の機能です。"  
if fjc_n=003:SSTR=" 上書保存：現在未実装の機能です。"  
if fjc_n=004:SSTR="      保存：現在未実装の機能です。"  
if fjc_n=005:SSTR="      終了：プログラムを終了します。"  
if fjc_n=006:SSTR="      次へ：入力された式をシンプレックス表にし、ピボット選択フェーズに移行します。"  
if fjc_n=007:SSTR="      戻る：入力フェーズに戻ります。編集集中のシンプレックス表は消去されます。"  
if fjc_n=010:SSTR=" オプション：シンプレックス表の編集について、各種設定の変更をします。"
```

```
if fjc_n=101:SSTR=" 現在、入力モードです。目的式と制約式を入力して下さい。入力規則は説明書またはヘルプを参照して下さい。"
```

```
if fjc_n=201:SSTR=" 現在、テーブル操作モードです。クリックしてピボットを選択して下さい。"
```

```
return SSTR
```

```
#defcfunc FigureCheck str ffc_str
```

```
ffc_n=0  
cnvstow ffc_c, ffc_str  
if (ffc_c =000) :ffc_n=1 :Null  
if (ffc_c>=048)&(ffc_c<=057):ffc_n=2 :Number  
if (ffc_c>=065)&(ffc_c<=090):ffc_n=3 :Alphabet (Capital)  
if (ffc_c>=097)&(ffc_c<=122):ffc_n=3 :Alphabet (Small)  
if (ffc_c =043) :ffc_n=4 :Formula (+)  
if (ffc_c =045) :ffc_n=4 :Formula (-)  
if (ffc_c =042) :ffc_n=5 :Formula (*)  
if (ffc_c =047) :ffc_n=5 :Formula (/)  
if (ffc_c =060) :ffc_n=6 :Formula (<)  
if (ffc_c =062) :ffc_n=6 :Formula (>)  
if (ffc_c =061) :ffc_n=7 :Formula (=)  
:if (ffc_c>=033)&(ffc_c<=041):ffc_n=3 :Alphabet (Special) [!, ", #, $, %, &, ', (, )]  
:if (ffc_c =044) :ffc_n=3 :Alphabet (Special) [, ]  
:if (ffc_c =046) :ffc_n=3 :Alphabet (Special) [., ]  
if (ffc_c>=058)&(ffc_c<=059):ffc_n=3 :Alphabet (Special) [:, ;]  
if (ffc_c>=063)&(ffc_c<=064):ffc_n=3 :Alphabet (Special) [?, @]  
if (ffc_c>=123)&(ffc_c<=126):ffc_n=3 :Alphabet (Special) [{, |, }, ~]  
return ffc_n
```

```
#defcfunc CalFractPledge int cfp_m, int cfp_d
```

```
cfp_a=cfp_m:if cfp_a<0:cfp_a=-cfp_a  
cfp_b=cfp_d:if cfp_b<0:cfp_b=-cfp_b  
if (cfp_a=0)|(cfp_b=0):return 1  
if cfp_a<dfp_b{  
    cfp_c=cfp_a  
    cfp_a=cfp_b
```

```

        cfp_b=cfp_c
    }
repeat -1:if (0=cnt¥1000):await
    cfp_c=cfp_a¥cfp_b
    if cfp_c=0:break
    cfp_a=cfp_b
    cfp_b=cfp_c

loop
return cfp_b

#defcfunc StrFract int sff_m, int sff_d
    if sff_d=0:return "(ERROR)"
    sff_p=CalFractPledge(sff_m, sff_d)
    sff_pm=sff_m/sff_p
    sff_pd=sff_d/sff_p
    if sff_pd= 1{
        if sff_pm<0:sff="" +sff_pm:return sff
        sff="" +sff_pm:return sff
    }
    if sff_pd=-1{
        sff_pm=-sff_pm
        if sff_pm<0:sff="" +sff_pm:return sff
        sff="" +sff_pm:return sff
    }
    sff="" (" +sff_pm+ "/" +sff_pd+" )"
return sff

; _____ Function

```

```

#deffunc DebugTable
    SSTR=""
    repeat DATfnc:n=cnt
        repeat DATvar:m=cnt
            SSTR+=" "+StrFract (DATtable_m. n. m, DATtable_d. n. m)+" , "
        loop
        SSTR+="¥n"
    loop
    dialog SSTR
    return

#deffunc DebugDisp
    SSTR=""
    repeat TBLrow:n=cnt
        repeat TBLcol:m=cnt
            SSTR+=" "+TBLdisplay. n. m+" , "
        loop
        SSTR+="¥n"

```

```

loop
dialog SSTR
return

#deffunc Draw
redraw 1
await SYSTEMWAIT
redraw 0
return

#deffunc KeyIn
getkey a, 27 : if a=1:end
KEYwheel=mousew
csrpx=csrx
csrpy=csry
csrx=mousex
csry=mousey
repeat KeyNum:n=cnt
getkey key, KEYcord.n
KEYpast.n=KEYsign.n
if key=0:KEYsign.n=0:else:KEYsign.n+
loop
return

#deffunc ColorOrder int fcl_o
fcl_j=fcl_o¥13
fcl_i=0:if fcl_j=fcl_i:COLr=000:COLg=000:COLb=128:return
fcl_i+ :if fcl_j=fcl_i:COLr=000:COLg=064:COLb=000:return
fcl_i+ :if fcl_j=fcl_i:COLr=128:COLg=064:COLb=000:return
fcl_i+ :if fcl_j=fcl_i:COLr=000:COLg=128:COLb=128:return
fcl_i+ :if fcl_j=fcl_i:COLr=096:COLg=000:COLb=160:return
fcl_i+ :if fcl_j=fcl_i:COLr=128:COLg=064:COLb=064:return
fcl_i+ :if fcl_j=fcl_i:COLr=128:COLg=128:COLb=064:return
fcl_i+ :if fcl_j=fcl_i:COLr=064:COLg=064:COLb=064:return
fcl_i+ :if fcl_j=fcl_i:COLr=000:COLg=000:COLb=255:return
fcl_i+ :if fcl_j=fcl_i:COLr=255:COLg=128:COLb=000:return
fcl_i+ :if fcl_j=fcl_i:COLr=000:COLg=128:COLb=128:return
fcl_i+ :if fcl_j=fcl_i:COLr=128:COLg=000:COLb=128:return
fcl_i+ :if fcl_j=fcl_i:COLr=255:COLg=000:COLb=128:return
return

#deffunc LineBox int flb_x1, int flb_y1, int flb_x2, int flb_y2
line flb_x1, flb_y1, flb_x2, flb_y1
line flb_x2, flb_y1, flb_x2, flb_y2
line flb_x2, flb_y2, flb_x1, flb_y2
line flb_x1, flb_y2, flb_x1, flb_y1
return

```

```

#deffunc ObjectInit
    foi_j=0
    notesel EDTnote
    noteload "object.txt"
    noteget SSTR, SYSTEMVIEW
    repeat 32:foi_i=cnt
        getstr OBJid.foi_i, SSTR, foi_j, ':foi_j+=strsize
        if strsize=0:break
    loop
    OBJnum=foi_i
    return

#deffunc ObjectAction int foa_n
    if foa_n=05{
        dialog " 終了しますか？ ", 2, PROGRAMNAME
        if stat=6:end
    }
    if foa_n=06{
        SYSTEMVIEW=1
    }
    if foa_n=07{
        SYSTEMVIEW=0
    }
    if foa_n=10{
        SYSTEMVIEW=5
    }
    return

#deffunc TableRead int ftr_o
;       Table Data
notesel DATtable_a
noteget SSTR, (ftr_o*2+0)
ftr_n=0
ftr_i=0
repeat -1:if 0=cnt¥1000:await
    getstr SSTR.1, SSTR, ftr_i, ':ftr_i+=strsize:if strsize=0:break
    ftr_m=0
    ftr_j=0
    repeat -1:if 0=cnt¥1000:await
        getstr SSTR.2, SSTR.1, ftr_j, ':ftr_j+=strsize:if strsize=0:break
        DATtable_m.ftr_n.ftr_m=int(SSTR.2)
        ftr_m+
    loop
    ftr_n+
loop

```

```

noteget SSTR, (ftr_o*2+1)
ftr_n=0
ftr_i=0
repeat -1:if 0=cnt¥1000:await
    getstr SSTR.1,SSTR,ftr_i,'':ftr_i+=strsize:if strsize=0:break
    ftr_m=0
    ftr_j=0
    repeat -1:if 0=cnt¥1000:await
        getstr SSTR.2,SSTR.1,ftr_j,'':ftr_j+=strsize:if strsize=0:break
        DATtable_d.ftr_n.ftr_m=int(SSTR.2)
        ftr_m+
    loop
    ftr_n+
loop
;      Table Display
notesel TBLdispldx
noteget SSTR,ftr_o
ftr_n=0
ftr_i=0
repeat -1:if 0=cnt¥1000:await
    getstr SSTR.1,SSTR,ftr_i,'':ftr_i+=strsize
    ftr_m=0
    ftr_j=0
    repeat -1:if 0=cnt¥1000:await
        getstr SSTR.2,SSTR.1,ftr_j,'':ftr_j+=strsize
        TBLdisplay.ftr_n.ftr_m=SSTR.2
        ftr_m+
        if ftr_m>TBLcol:break
    loop
    ftr_n+
    if ftr_n>TBLrow:break
loop
;      Table Range ( width , Heght )
repeat TBLcol:ftr_m=cnt
    TBLw.ftr_m=0
    repeat TBLrow:ftr_n=cnt
        ftr_a=strlen(TBLdisplay.ftr_n.ftr_m)
        if TBLw.ftr_m<ftr_a:TBLw.ftr_m=ftr_a
        loop
        TBLw.ftr_m+=2
        if TBLw.ftr_m<8:TBLw.ftr_m=8
    loop
repeat DATfnc+1:ftr_n=cnt

```

```

        TBLh.ftr_n=2

loop
;      Pibot
Pfn=-1
Pvar=-1
if ftr_o<CLASS{
        notesel TBLpibot
        noteget SSTR, ftr_o*2+0:Pfn=int(SSTR)
        noteget SSTR, ftr_o*2+1:Pvar=int(SSTR)
}
;      Constant
repeat DATvar-1:m=cnt
        DATbase_m.m=0
        DATbase_d.m=1
        ftr_hit=0:i=-1
        repeat DATfnc:n=cnt
                SDV=double(DATtable_m.n.m)/double(DATtable_d.n.m)
                if SDV!=0.0:ftr_hit+=i=n
        loop
        if ftr_hit=1{
                j=DATvar-1
                DATbase_m.m=DATtable_m.i.j
                DATbase_d.m=DATtable_d.i.j
        }
loop
return

#deffunc TableSave int fts_o
        fts_order0=fts_o
        fts_order1=fts_o*2
        fts_order2=fts_o*2+1
        fts_mode=1
;      -- Pibot Log
        dim GRPbs_m, fts_o
        dim GRPbs_d, fts_o
        dim GRPpx_m, fts_o
        dim GRPpx_d, fts_o
        dim GRPpy_m, fts_o
        dim GRPpy_d, fts_o

;      -- Table
        notesel DATtable_a
        noteadd ""
        noteadd ""

```

```

SSTR=""
repeat DATfnc:n=cnt
    repeat DATvar:m=cnt
        if m>0:SSTR+=","
        SSTR+=DATtable_m.n.m
    loop
    SSTR+=":"
loop
noteadd SSTR,fts_order1,fts_mode
SSTR=""
repeat DATfnc:n=cnt
    repeat DATvar:m=cnt
        if m>0:SSTR+=","
        SSTR+=DATtable_d.n.m
    loop
    SSTR+=":"
loop
noteadd SSTR,fts_order2,fts_mode
;      -- Display
notesel TBLdispldx
noteadd ""
SSTR=""
repeat TBLrow:n=cnt
    repeat TBLcol:m=cnt
        if m>0:SSTR+=","
        SSTR+=TBLdisplay.n.m
    loop
    SSTR+=":"
loop
noteadd SSTR,fts_order0,fts_mode
;      -- Range List
notesel TBLrange
noteadd ""
noteadd ""
cellx=0:celly=0
repeat TBLrow:celly+=TBLh.cnt:loop
repeat TBLcol:cellx+=TBLw.cnt:loop
noteadd str(cellx),fts_order1,fts_mode
noteadd str(celly),fts_order2,fts_mode
;      -- Pivot List
if fts_o>0{
    fts_order1=(fts_o-1)*2
    fts_order2=(fts_o-1)*2+1
}

```

```

        notesel TBLpivot
noteadd ""
noteadd ""
        noteadd str(Tfnc), fts_order1, fts_mode
        noteadd str(Tvar), fts_order2, fts_mode
    }
    return
#deffunc TableDisplay
;         Table Display
repeat TBLrow:n=cnt
repeat TBLcol:m=cnt
    SSTR=""
    if n=0{
        if m=0:SSTR=" 基底変数 "
        if (m>0)&(m<DATvar):a=m-1:SSTR=DATvarname. a
        if m=DATvar:SSTR=" 定数項 "
        if m=DATvar+1:SSTR=" 增加限界 "
    }
    if n>0{
        if m=0{
            a=n-1
            repeat DATvar:j=cnt
                if DATtable_m. a. j=1{
                    hit=1
                    repeat DATfnc:i=cnt
                        if i=a:continue
                        if DATtable_m. i. j!=0:hit=0:break
                    loop
                    if hit=1:SSTR=DATvarname. j:break
                }
            loop
        }
        if m>0:a=n-1:b=m-1:if b<DATvar:s=DATtable_m. a. b:t=DATtable_d. a. b:SSTR=""+StrFract(s, t)
    }
    TBLdisplay. n. m=SSTR
loop
loop
;         Table Range ( width , Hegt )
repeat TBLcol:m=cnt
    TBLw. m=0
    repeat TBLrow:n=cnt
        len=strlen(TBLdisplay. n. m)
        if TBLw. m<len:TBLw. m=len

```

```

        loop
        TBLw. m+=2
        if TBLw. m<8:TBLw. m=8

loop
repeat DATfnc+1:n=cnt
        TBLh. n=2

loop
return

#deffunc CameraRangeReset
CMRrx=0
CMRry=0
notesel TBLrange
repeat CLASS+1:n=cnt*2:m=n+1:o=cnt
        noteget SSTR, n:x=int(SSTR)*fw
        noteget SSTR, m:y=int(SSTR)*fh
        if CMRrx<x:CMRrx=x
        CMRry+=y+CELL

loop
if CMRrx<0:CMRrx=0
if CMRry<0:CMRry=0
return

;_____BOOT

*BOOT

;      Buffering
buffer ID_SYSTEM      :picload "system.bmp"
buffer ID_TITLE       :picload "title.bmp"
buffer ID_BACK        :picload "back.bmp"
buffer ID_STORE, wx, wy
buffer ID_GRAPH, wx, wy
buffer ID_GRAPH_S, wx, wy

;      Window Open
screen ID_MAIN, wx, wy, 4, (ginfo_dispx-wx)/2, (ginfo_dispy-wy)/2, wx, wy:gsel ID_MAIN
title " "+PROGRAMNAME+" --- ver "+VERSION
redraw 0
SOM=JessicaComment(0)
f=12
font FONTSTYLE, f, 16
color 0, 0, 0:boxf 0, 0, wx, wy
x=(wx-360)/2
y=(wy-120)/2
pos x, y:gmode 0, 360, 120:gcopy ID_TITLE, 0, 0
y+=120

```

```

color 255,255,255
SSTR="Please Click."
:w=strlen(SSTR)*f/2:x=(wx-w)/2:y+=f+4:pos x,y:mes SSTR
color 128,128,128
pos 16,16:mes "Copyright: 和田学 (HanleName = M A N A )¥n 文教大学経営情報学科 ¥n 根本ゼミナール 10 期生 ¥n 卒業研
究制作ソフト"
Draw

repeat 300
KeyIn
if KEYsign.4=1:break
await SYSTEMWAIT
loop

;_____MAIN_Input

*MAIN_INPUT_INIT
SYSTEMVIEW=0
font FONTSTYLE, FONTSIZE, 16

clobj
await 0
KeyIn
objmode 2,0
objsize wx,wy-64
pos 0,64
mesbox DATinput,,,0
ObjectInit

*MAIN_INPUT
KeyIn

SOM=JessicaComment(101)
gosub *OBJECT_BASE

if SYSTEMVIEW=1:goto *MAIN_INPUT_COMPILE

Draw
goto *MAIN_INPUT

*MAIN_INPUT_COMPILE
index=0
compile=0
PHASE=0
DATerr=0
DATov=0

```

```

sdim DATcompile
sdim DATstandard
sdim LISTvar

; Compile
repeat -1
    notesel DATinput
    if noteinfo(0)=0:DATerr=001:break
    if noteinfo(0)<=index:break
    noteget SSTR, index
        ; Delete Space Code
        SSTR.1=""
        repeat -1:i=cnt
            SSTR.2=strmid(SSTR, i, 1)
            hit=0
            if SSTR.2="" :break
            if SSTR.2=" " :hit+
            if SSTR.2="¥t" :hit+
            if hit=0:SSTR.1+=SSTR.2
        loop
        SSTR=SSTR.1

    if PHASE=0{
        ;Seek "Aim"
        DATerr=002
        if (SSTR="max") | (SSTR="Max") | (SSTR="MAX") | (SSTR="max.") | (SSTR="Max.") | (SSTR="MAX.") | (SSTR="maximize") |
(SSTR="Maximise") | (SSTR="MAXIMISE") :DATerr=0:DATaim=0
        if (SSTR="min") | (SSTR="Min") | (SSTR="MIN") | (SSTR="min.") | (SSTR="Min.") | (SSTR="MIN.") | (SSTR="minimize") |
(SSTR="Minimise") | (SSTR="MINIMISE") :DATerr=0:DATaim=1
        if DATerr=0{
            notesel DATcompile
            if DATaim=0:noteadd "Maximize"
            if DATaim=1:noteadd "Minimize"
            PHASE+
            DATerr=011
        }
    }else:if PHASE=1{
        ;Seek "Aim Function"
        if SSTR!="" {
            DATerr=0
            FIGact=0
            SSTR.1=""
            SSTR.2=""
            SSTR.3=""
            repeat -1:i=cnt

```

```

;FIGact=0 [Not Found ]
;FIGact=1 [Null ]
;FIGact=2 [Number ]Coefficient
;FIGact=3 [Alphabet ]Variable
;FIGact=4 [Formula +, - ]
;FIGact=5 [Formula *, / ]
;FIGact=6 [Formula >, < ]
;FIGact=7 [Formula = ]

if
FIGact=00:ROT. 0=1012:ROT. 1=1019:ROT. 2=0001:ROT. 3=0002:ROT. 4=0001:ROT. 5=1013:ROT. 6=1013:ROT. 7=1013
if
FIGact=01:ROT. 0=1012:ROT. 1=1013:ROT. 2=0001:ROT. 3=0002:ROT. 4=1013:ROT. 5=1013:ROT. 6=1013:ROT. 7=1013
if
FIGact=02:ROT. 0=1012:ROT. 1=0099:ROT. 2=0002:ROT. 3=0002:ROT. 4=0001:ROT. 5=1013:ROT. 6=1013:ROT. 7=1013

SSTR. 2=strmid(SSTR, i, 1)
k=FigureCheck(SSTR. 2)
pastFIGact=FIGact
SSTR. 1+=SSTR. 2
if 0=ROT. k/1000:FIGact=ROT. k≠1000:else:DATerr=ROT. k≠1000
if pastFIGact=2{
if (FIGact=1)|(FIGact=99){
hit=0
notesel LISTvar
repeat noteinfo(0):n=cnt
noteget SSTR. 4, n
if SSTR. 3=SSTR. 4:hit=1:break
loop
if hit=0:noteadd SSTR. 3
SSTR. 3=""
}
}
if FIGact=02:SSTR. 3+=SSTR. 2
if FIGact=99:break
if DATerr>0:break
loop
if DATerr=0{
notesel DATcompile
noteadd SSTR. 1
PHASE+
DATerr=021
}
}

```

```

}else:if PHASE=2{ :Seek "subject to"
    if (SSTR="s. t.") | (SSTR="s. t") | (SSTR="st") | (SSTR="ST") | (SSTR="subjectto") | (SSTR="Subjectto") |
(SSTR="SubjectTo"):DATerr=0
    if DATerr=0{
        notesel DATcompile
        noteadd "subject to"
        PHASE+
        DATerr=031
    }
}else:if PHASE=3{ :Seek "Subject"
    if SSTR!=""{
        DATerr=0
        DATres=0
        FIGact=0
        SSTR. 1=""
        SSTR. 2=""
        SSTR. 3=""
        repeat -1:i=cnt
            ;FIGact=0 [Not Found ]
            ;FIGact=1 [Null ]
            ;FIGact=2 [Number ]Coefficient
            ;FIGact=3 [Alphabet ]Variable
            ;FIGact=4 [Formula +, - ]
            ;FIGact=5 [Formula *, / ]
            ;FIGact=6 [Formula >, < ]
            ;FIGact=7 [Formula = ]
            if
FIGact=00:ROT. 0=1032:ROT. 1=1039:ROT. 2=0001:ROT. 3=0002:ROT. 4=0001:ROT. 5=1033:ROT. 6=1033:ROT. 7=1033
                if
FIGact=01:ROT. 0=1032:ROT. 1=1033:ROT. 2=0001:ROT. 3=0002:ROT. 4=1033:ROT. 5=1033:ROT. 6=1033:ROT. 7=1033
                    if
FIGact=02:ROT. 0=1032:ROT. 1=1033:ROT. 2=0002:ROT. 3=0002:ROT. 4=0001:ROT. 5=1033:ROT. 6=0003:ROT. 7=0004
                        if
FIGact=03:ROT. 0=1032:ROT. 1=1033:ROT. 2=0005:ROT. 3=1033:ROT. 4=0005:ROT. 5=1033:ROT. 6=1033:ROT. 7=0004
                            if
FIGact=04:ROT. 0=1032:ROT. 1=1033:ROT. 2=0005:ROT. 3=1033:ROT. 4=0005:ROT. 5=1033:ROT. 6=1033:ROT. 7=1033
                                if
FIGact=05:ROT. 0=1032:ROT. 1=0099:ROT. 2=0005:ROT. 3=1033:ROT. 4=1033:ROT. 5=1033:ROT. 6=1033:ROT. 7=1033
                                    SSTR. 2=strmid(SSTR, i, 1)
                                    k=FigureCheck(SSTR. 2)
                                    pastFIGact=FIGact
                                    if 0=ROT. k/1000:FIGact=ROT. k¥1000:else:DATerr=ROT. k¥1000

```

```

        if pastFIGact=2{
            if (FIGact=1)|(FIGact=3)|(FIGact=4){
                hit=0
                notesel LISTvar
                repeat noteinfo(0):n=cnt
                    noteget SSTR.4,n
                    if SSTR.3=SSTR.4:hit=1:break
                loop
                if hit=0:noteadd SSTR.3
                SSTR.3=""
            }
        }
        if FIGact=02:SSTR.3+=SSTR.2
        if FIGact=03:DATres+
        if FIGact=04:if DATres>0:SSTR.2=""
        SSTR.1+=SSTR.2
        if FIGact=99:break
        if DATerr>0:break
    loop
    if DATerr=0{
        notesel DATcompile
        noteadd SSTR.1
    }
}

index+
await 0

loop
notesel EDTnote
if DATerr>0{
    if DATerr=001:SSTR="Compile error "+DATerr+"%n 入力ボックスの内容が空値です。"
    if DATerr=002:SSTR="Compile error "+DATerr+"%n 目的関数の方向性が見つかりません。 %n 行頭には max（最大
化問題）か min（最小化問題）を入力して下さい。"
    if DATerr=011:SSTR="Compile error "+DATerr+"%n 目的関数が見つかりません。"
    if DATerr=012:SSTR="Compile error "+DATerr+"%n 目的関数の記述が不正です。 %n 使用できない文字が含まれて
います。"
    if DATerr=013:SSTR="Compile error "+DATerr+"%n 目的関数の記述が不正です。 %n 入力規則に違反しています。"
    if DATerr=014:SSTR="Compile error "+DATerr+"%n 目的関数の記述が不正です。 %n 同一名の変数が重複して定義
されています。"
    if DATerr=019:SSTR="Compile error "+DATerr+"%n システムエラーです。 %n 目的関数のチェック中に内部変数が
不正な値を取りました。 %n 製作元にコンパイルデータを送信して下さい。"
    if DATerr=021:SSTR="Compile error "+DATerr+"%n 制約関数の定義位置が見つかりません。 %n 制約関数を入力す

```

る前の行に subject to を入力して下さい。 ”

```
if DATerr=031:SSTR="Compile error "+DATerr+"%n 制約関数が見つかりません。 ”
```

```
if DATerr=032:SSTR="Compile error "+DATerr+"%n 制約関数の記述が不正です。 %n 使用できない文字が含まれて  
います。 ”
```

```
if DATerr=033:SSTR="Compile error "+DATerr+"%n 制約関数の記述が不正です。 %n 入力規則に違反しています。 ”
```

```
if DATerr=034:SSTR="Compile error "+DATerr+"%n 制約関数の記述が不正です。 %n 同一名の変数が重複して定義  
されています。 ”
```

```
if DATerr=039:SSTR="Compile error "+DATerr+"%n システムエラーです。 %n 制約関数のチェック中に内部変数が  
不正な値を取りました。 %n 製作元にコンパイルデータを送信して下さい。 ”
```

```
if DATerr=999:SSTR="Compile error "+DATerr+"%n 致命的なエラーです。 コンパイルは正しく終了できませんで  
したが、何処で問題が発生したのかわかりません。 ”
```

```
dialog SSTR, 1, PROGRAMNAME
```

```
SYSTEMVIEW=0
```

```
goto *MAIN_INPUT_INIT
```

```
}
```

```
; Table Init
```

```
notesel DATcompile :DATfnc=noteinfo(0)-2 :TBLrow=DATfnc+1
```

```
notesel LISTvar :DATvar=noteinfo(0)+DATfnc+1:TBLcol=DATvar+2
```

```
dim DATrestric, DATfnc :Data Restriction
```

```
dim DATtable_m, DATfnc, DATvar :Data Table Value Molecule
```

```
dim DATtable_d, DATfnc, DATvar :Data Table Value Denominator
```

```
dim DATbase_m, DATvar :Data Base Var. Molecule
```

```
dim DATbase_d, DATvar :Data Base var. Denominator
```

```
dim TBLw, TBLcol :Table Stat Width
```

```
dim TBLh, TBLrow :Table Stat Height
```

```
sdim DATvarname, DATvar :Data Var. Name
```

```
sdim DATtable_i :Data Table Value Index
```

```
sdim TBLdisplay, TBLrow, TBLcol :Table Stat Display
```

```
sdim TBLdispldx :Table Stat Display Index
```

```
sdim TBLpibot :Table Pibot List
```

```
sdim TBLrange :Table Range List
```

```
ORDER=0 :Table Index Order
```

```
CLASS=0 :Table Class Number
```

```
; Variable Value Name Read
```

```
repeat (DATvar-1):n=cnt:m=n-1:o=n-noteinfo(0)
```

```
if n=0:SSTR="(Z)":if DATaim=1:SSTR="(-Z)"
```

```
if (n>0)&(n<noteinfo(0)+1):noteget SSTR, m
```

```
if n>noteinfo(0):SSTR="(S"+o+)"
```

```
DATvarname.n=SSTR
```

```

loop

;      Coefficient Read
;      Molecule Reset
notesel DATcompile
repeat DATfnc:n=cnt
  if n=DATfnc-1:o=1:else:o=cnt+3
  noteget SSTR,o
  repeat (DATvar-1):m=cnt
    FIGact=0
    SSTR.1=""
    SSTR.2=""
    SSTR.3=""
    SSTR.4=""
    repeat -1:i=cnt
      if
FIGact=00:ROT.0=1051:ROT.1=1051:ROT.2=0001:ROT.3=0002:ROT.4=0001:ROT.5=1051:ROT.6=1051:ROT.7=1051
      if
FIGact=01:ROT.0=1051:ROT.1=1051:ROT.2=0001:ROT.3=0002:ROT.4=1051:ROT.5=1051:ROT.6=1051:ROT.7=1051
      if
FIGact=02:ROT.0=1051:ROT.1=0099:ROT.2=0002:ROT.3=0002:ROT.4=0001:ROT.5=1051:ROT.6=0003:ROT.7=0003
      if
FIGact=03:ROT.0=1051:ROT.1=1051:ROT.2=0004:ROT.3=1051:ROT.4=0004:ROT.5=1051:ROT.6=1051:ROT.7=1051
      if
FIGact=04:ROT.0=1051:ROT.1=0099:ROT.2=0004:ROT.3=1051:ROT.4=1051:ROT.5=1051:ROT.6=1051:ROT.7=1051

      SSTR.2=strmid(SSTR,i,1)
      k=FigureCheck(SSTR.2)
      pastFIGact=FIGact
      FIGact=ROT.k¥1000
      if pastFIGact=2{
        if (FIGact=1)|(FIGact=3)|(FIGact=99){
          if SSTR.3=DATvarname.m{
            a=int(SSTR.4)
            if SSTR.4="" :a=1
            if SSTR.4="+" :a=1
            if SSTR.4="-" :a=-1
            if (o=1)&(DATAim=0) :a=a*(-1)
            DATtable_m.n.m+=a
          }
          SSTR.3=""
        }
      }
    }
  }
}

```

```

        if FIGact=01 {
            if pastFIGact!=01:SSTR. 4=""
            SSTR. 4+=SSTR. 2
        }
        if FIGact=02:SSTR. 3+=SSTR. 2
    if FIGact=03 {
        if SSTR. 2="":DATrestric. n=1
        if SSTR. 2="<":DATrestric. n=2
        if SSTR. 2=">":DATrestric. n=3
    }
    if m=0 {
        if FIGact=04 {
            if pastFIGact!=4:SSTR. 4=""
            SSTR. 4+=SSTR. 2
        }
        if FIGact=99 {
            if pastFIGact=4:r=DATvar-1:DATtable_m. n. r=int (SSTR. 4)
        }
    }
    if FIGact=099:break

        loop
    loop
loop
;      Denominator Reset
repeat DATfnc:n=cnt
repeat DATvar:m=cnt
    DATtable_d. n. m=1
loop
loop
;      Standardization
repeat DATfnc:n=cnt
    m=n+DATvar-DATfnc
    if n=DATfnc-1:m=0
    d=1:if DATrestric. n=3:d=-1
    if DATrestric. n=1:d=0
    DATtable_m. n. m=d
    if d=-1 {
        repeat DATvar:m=cnt
            DATtable_m. n. m=-DATtable_m. n. m
        loop
    }
loop

```

```

;      Write Standardly
notesel DATstandard
repeat DATfnc+1:n=cnt
    SSTR=""
    if n=0{
        SSTR="Max. ¥t"+DATvarname. 0
    }else{
        if n=1:SSTR="s. t. ¥t":else:SSTR="¥t"
        i=n-1:k=0
        repeat DATvar-1:j=cnt
            if DATtable_m. i. j=0:continue
            if k=0{
                if DATtable_m. i. j<0:SSTR. 1="-":else:SSTR. 1=""
            }else{
                if DATtable_m. i. j<0:SSTR. 1="-":else:SSTR. 1="+
            }
            t=abs(DATtable_m. i. j)
            if t>1:SSTR. 1+=str(t)
            SSTR+=""+SSTR. 1+DATvarname. j+" "
            k+
        loop
        j+
        SSTR+="="
        if DATtable_m. i. j<0:SSTR+="-":else:SSTR+=" "
        SSTR+=str(abs(DATtable_m. i. j))
    }
    noteadd SSTR
loop
SSTR="¥t"
repeat DATvar-2:n=cnt+1
    if n>1:SSTR+=", "
    SSTR+=""+DATvarname. n
loop
SSTR+="≥0"
noteadd SSTR
INFOLINE=noteinfo(0)

;      Graph
GRAPH=0
if 2=DATvar-DATfnc-1{
    ;      2 Dimention Graph
    ;      -- Graph Range

```

```

GRAPH=1
d=DATfnc-1
dim GRPxm, n
dim GRPxd, n
dim GRPym, n
dim GRPyd, n
MaxX=0
MaxY=0
repeat DATfnc-1:n=cnt      :Crossing Position
    i=1:j=2:k=DATvar-1
    a=DATtable_m. n. i
    b=DATtable_m. n. j
    c=DATtable_m. n. k
    repeat DATfnc+1:m=cnt
        if m=n:continue
        if m<DATfnc{
            i=1:j=2:k=DATvar-1
            s=DATtable_m. m. i
            t=DATtable_m. m. j
            u=DATtable_m. m. k
        }
        if m=DATfnc-1:s=1:t=0:u=0
        if m=DATfnc-0:s=0:t=1:u=0
        GRPxm. n=u*b-t*c
        GRPxd. n=s*b-t*a
        GRPym. n=u*a-s*c
        GRPyd. n=t*a-s*b
        x=0:y=0
        if GRPxd. n!=0:x=GRPxm. n/GRPxd. n
        if GRPyd. n!=0:y=GRPym. n/GRPyd. n
        if MaxX<x:MaxX=x
        if MaxY<y:MaxY=y
    loop
loop
MaxX+
MaxY+
;      -- Graph Range (Ratio to WindowSize)
sx=wx-CELL*8
sy=wy-CELL*4
mx=MaxX*11/10
my=MaxY*11/10

```

```

p=CalFractPledge (sx, sy)
rx=sx/p
ry=sy/p
SDV. 0=double (rx)/double (mx)
SDV. 1=double (mx)*SDV. 0
SDV. 2=double (my)*SDV. 0
SDV. 3=double (ry)
BASIS=0: if SDV. 2>SDV. 3: BASIS=1
MAG=0. 0: MEASURE=0. 0
if BASIS=0: MAG=double (sx)/SDV. 1
if BASIS=1: MAG=double (sy)/SDV. 2
GRPwx=int (SDV. 1*MAG)+CELL*2: if BASIS=0: MEASURE=double (GRPwx-CELL*2)/double (MaxX)
GRPwy=int (SDV. 2*MAG)+CELL*2: if BASIS=1: MEASURE=double (GRPwy-CELL*2)/double (MaxY)

;      -- Draw Graph
;      -- -- Background
gsel ID_GRAPH
color 255, 255, 255
boxf 0, 0, GRPwx, GRPwy

;      -- -- Grid
color 224, 224, 224
repeat GRPwx:x=cnt
    x=int (double (x)*MEASURE)+CELL
    if x>GRPwx:break
    line x, 0, x, GRPwy
loop
repeat GRPwy:y=cnt
    y=-int (double (y)*MEASURE)+GRPwy-CELL
    if y<0:break
    line 0, y, GRPwx, y
loop

;      -- -- Function
font FONTSTYLE, 11, 16
notesel DATcompile
repeat DATfnc-1:n=cnt
    ColorOrder n
    color COLr, COLg, COLb
    i=1:j=2:k=DATvar-1
    a=DATtable_m. n. i
    b=DATtable_m. n. j
    c=DATtable_m. n. k

```

```

    if b!=0{
        mx1=0 :dx1=1
        my1=c-a*mx1 :dy1=b
        mx2=MaxX*11/10+1 :dx2=1
        my2=c-a*mx2 :dy2=b
        x1=int(double(mx1)*MEASURE/double(dx1))
        y1=int(double(my1)*MEASURE/double(dy1))
        x2=int(double(mx2)*MEASURE/double(dx2))
        y2=int(double(my2)*MEASURE/double(dy2))
        x1+=CELL
        x2+=CELL
        y1=-y1+GRPwy-CELL
        y2=-y2+GRPwy-CELL
    }else{
        mx1=c :dx1=a
        mx2=c :dx2=a
        x1=int(double(mx1)*MEASURE/double(dx1))
        x2=int(double(mx2)*MEASURE/double(dx2))
        x1+=CELL
        x2+=CELL
        y1=0
        y2=GRPwy
    }
    noteget SSTR, n+3
    pos CELL+16, 8+n*(FONTSIZE+6) :mes "■:"+SSTR
    s=0:d=0
    if DATrestrict.n=2:d=1
    if DATrestrict.n=3:d=-1
    cr=(255-COLr)/7
    cg=(255-COLg)/7
    cb=(255-COLb)/7
    repeat 7:t=cnt
        s+=d*t
        color COLr+cr*t, COLg+cg*t, COLb+cb*t
        if a<=b:line x1, y1+s, x2, y2+s
        if a> b:line x1-s, y1, x2-s, y2
    loop
loop
; -- -- Stem
font FONTSTYLE, 10, 16
color 255, 255, 255
boxf 0, GRPwy-CELL, GRPwx, GRPwy

```

```

boxf 0,0,CELL,GRPwy
color 32,32,32
line CELL,0,CELL,GRPwy-CELL                                :Stem X
line CELL,GRPwy-CELL,GRPwx,GRPwy-CELL                    :Stem Y
gx=int(double(GRPwx)/MEASURE)/10:if gx<1:gx=1
gy=int(double(GRPwy)/MEASURE)/10:if gy<1:gy=1
r=4
repeat GRPwx:x=cnt:n=cnt
    if 0!=x%gx:continue
    x=int(double(x)*MEASURE)+CELL
    if x>GRPwx:break
    line x,GRPwy-CELL-r,x,GRPwy-CELL+r
    w=strlen(str(n))*5/2
    pos x-w,GRPwy-CELL+r+2:mes ""+n
loop
repeat GRPwy:y=cnt:n=cnt
    if 0!=y%gy:continue
    y=-int(double(y)*MEASURE)+GRPwy-CELL
    if y<0:break
    line CELL-r,y,CELL+r,y
    w=strlen(str(n))*5
    pos CELL-w-r-2,y-5:mes ""+n
loop
color 255,0,0
pos 4,8:mes DATvarname.2
pos GRPwx-strlen(DATvarname.1)*5-4,GRPwy-15:mes DATvarname.1

:          -- End Making Graph
color 0,255,255
LineBox 0,0,GRPwx-1,GRPwy-1

gsel ID_MAIN

:clrobj
:pos 0,0
:gmode 0,GRPwx,GRPwy
:gcopy ID_GRAPH,0,0
:Draw
:stop
}

:          Make Table Display
TableDisplay

```

TableSave ORDER

SYSTEMVIEW=2

*MAIN_TABLE_INIT

clrobj

await 0

KeyIn

ObjectInit

; Option

dim OPTION,8,8

OPTION.0.0=1 ;Pibot Option -- Aim less 0

OPTION.0.1=1 ;Pibot Option -- Base than 0

; Camera

CMRpx=0:CMRpy=0 ;Camera Position

CMRfx=0:CMRfy=0 ;Camera Focus

CMRrx=0:CMRry=0 ;Camera Range

CameraRangeReset

; KeySpeed

WheelSpeed=CELL

; Font Size

font FONTSTYLE, FONTSIZE, 16

fw=FONTSIZE/2

fh=FONTSIZE

FADE=0.0 ;Font color fade

; ClickFlag

hit=0

*MAIN_TABLE

KeyIn

SOM=JessicaComment(201)

if SYSTEMVIEW=0:goto *MAIN_INPUT_INIT

; Camera Action

if (csry<=80)&(csry>64) :CMRfy-=4

if (csry>=wy-16) :CMRfy+=4

```

if (KEYsign.7=0)&(KEYwheel>0):CMRfy-=WheelSpeed
if (KEYsign.7=0)&(KEYwheel<0):CMRfy+=WheelSpeed
if (KEYsign.7>0)&(KEYwheel>0):CMRfx-=WheelSpeed
if (KEYsign.7>0)&(KEYwheel<0):CMRfx+=WheelSpeed
  if CMRfy<0:CMRfy=0:else:if CMRfy>CMRry:CMRfy=CMRry
  if CMRfx<0:CMRfx=0:else:if CMRfx>CMRrx:CMRfx=CMRrx
CMRpy+=(CMRfy-CMRpy)/2

;          View Stat
dx= 64-CMRpx
dy=128-CMRpy
dy+=INFOLINE*fh+CELL*2
FADE+=0.40:if FADE>=6.28:FADE-=6.28

;          Background
TEXTURE=1024
gmode 0,TEXTURE,TEXTURE
pos 0,64
gcopy ID_BACK,0,0

;          Head Information
if CMRdy<(INFOLINE*fh+CELL*2){
  color 255,244,224
  boxf 0,128-CMRpy,wx,128-CMRpy+INFOLINE*fh
  color 64,64,32
  pos dx,128-CMRpy
  mes DATstandard
}

;          Table
VIEW=-1
repeat CLASS+1:ORDER=cnt
  ;          Seek In Camera
  if VIEW<0{
    notesel TBLrange
    noteget SSTR,ORDER*2+1
    y=dy+int(SSTR)*fh+CELL
    py=CMRpy-(160*ORDER)-CELL*5-INFOLINE*fh
    if y<py:dy=y:continue
    VIEW=ORDER
  }

;          Table Order Read

```

TableRead ORDER

; Attention

SSTR=""

t=0

repeat DATvar-1:m=cnt

if m=0:continue

if DATbase_m.m<0:t+:break

loop

if t>0{

SSTR+=" 基底変数 "

t=0

repeat DATvar-1:m=cnt

if m=0:continue

if DATbase_m.m<0{

if t>0:SSTR+=" , "

SSTR+=" "+DATvarname.m

t+

}

loop

SSTR+=" の値が負数です。¥n非不制約に違反しています。 "

}

ATTENTION=SSTR

; Cursor

; -- Cursor Graph

if GRAPH=1{

w=24:h=w:x=dx-w:y=dy:f=0

if hit=0{

if (csrx>x)&(csrx<=x+w)&(csry>y)&(csry<=y+h):f+

if (KEYsign.4=1)&(f=1){

hit+

f+

GRPodr=ORDER+1

SYSTEMVIEW=4

}

}

pos x,y

gmode 2,w,h

gcopy ID_SYSTEM,f*w,183

gcopy ID_SYSTEM,8*w,0

}

if ATTENTION!=""{

```

w=24:h=w:x=dx-w:y=dy+h:f=0
if hit=0{
    if (csrx>x)&(csrx<=x+w)&(csry>y)&(csry<=y+h):f+
    if (KEYsign.4=1)&(f=1):dialog ATTENTION,0,PROGRAMNAME
}
pos x,y
gmode 2,w,h
gcopy ID_SYSTEM,f*w,183
gcopy ID_SYSTEM,9*w,0
}
; -- Cursor Table
if hit=0{
    Tvar=0
    cellx=0
    celly=0
    repeat DATvar:m=cnt
        if m<=1:cellx+=TBLw.m:continue
        t=0
        i=DATfnc-1:j=m-1
        if (DATtable_m.i.j<0)|(OPTION.0.0=0):t+
        if DATbase_m.j!=0:t=0
        if t>0{
            celly=0
            repeat TBLrow:n=cnt
                if (n>0)&(n<DATfnc){
                    t=1:p=n-1:q=m-1
                    if (DATtable_m.p.q<0)&(OPTION.0.1=1):t=0
                    if DATtable_m.p.q=0:t=0
                    if t>0{
                        x=dx+cellx*fw:w=TBLw.m*fw:y=dy+celly*fh:h=TBLh.n*fh
                        if (csrx>x)&(csrx<=x+w)&(csry>y)&(csry<=y+h){
                            hit=1
                            Tfnc=n-1 ;Target Function
                            Tvar=m-1 ;Target Variable
                            Todr=ORDER ;Target Order
                            break
                        }
                    }
                }
            }
            celly+=TBLh.n
        }
    }
}
if hit>0:break

```

```

        cellx+=TBLw.m
    loop
    if hit>0{
        repeat DATfnc-1:n=cnt:m=DATvar-1:i=n+1:j=DATvar+1
            a=DATtable_m.n.m
            b=DATtable_d.n.m
            c=DATtable_m.n.Tvar
            d=DATtable_d.n.Tvar
            s=a*d
            t=b*c
            TBLdisplay.i.j=""+StrFract(a,b)+"/"+StrFract(c,d)+"=" +StrFract(s,t)
        loop
    }
    m=TBLcol-1
    TBLw.m=0
    repeat TBLrow:n=cnt
        len=strlen(TBLdisplay.n.m)
        if TBLw.m<len:TBLw.m=len
    loop
    TBLw.m+=2
    if TBLw.m<8:TBLw.m=8

;           Decide Pibot
    if (KEYsign.4=1)&(hit>0):SYSTEMVIEW=3
}

;           -- Table Frame
cellx=0
celly=0
repeat TBLcol:cellx+=TBLw.cnt:loop
repeat TBLrow:celly+=TBLh.cnt:loop
color 250,240,220
boxf dx,dy,dx+cellx*fw,dy+celly*fh

color 240,200,180
boxf dx,dy,dx+TBLw.0*fw,dy+celly*fh
boxf dx,dy,dx+cellx*fw,dy+TBLh.0*fh

x=0:y=0
repeat DATfnc:y+=TBLh.cnt:loop
repeat DATvar:x+=TBLw.cnt:loop
boxf dx+x*fw,dy,dx+cellx*fw,dy+celly*fh
boxf dx,dy+y*fh,dx+cellx*fw,dy+celly*fh

```

```

;          Draw Table
;          -- Table No.
color 0, 0, 0
pos dx, dy-fh:mes "Table No."+(ORDER+1)
color 255, 255, 255
boxf dx, dy+celly*fh, dx+cellx*fw, dy+celly*fh+fh+4
pos dx, dy+celly*fh+4
SSTR=""
repeat DATvar-1:m=cnt
    if m>0:SSTR+=" , "
    SSTR+=""+DATvarname. m+"="+StrFract (DATbase_m. m, DATbase_d. m)
loop
color 0, 64, 64
mes SSTR

;          -- Cell Info.
celly=0
repeat TBLrow:n=cnt
    cellx=0
    repeat TBLcol:m=cnt
        SSTR=TBLdisplay. n. m+" "
        len=strlen(SSTR)
        color 24, 24, 32
        if (m=Tvar+1)&(Tvar>0)&(Todr=ORDER):color 0, 128, 0
        if m=DATvar:color 0, 0, 255
        x=dx+cellx*fw+(TBLw. m-len)*fw/2
        y=dy+celly*fh+(TBLh. n-1)*fh/2
        pos x, y
        if m<=DATvar:mes SSTR:else:if strlen(SSTR)>0{
            if n=0:mes SSTR:else{
                color 0, 0, 255
                type=0
                i=n-1:j=Tvar-1
                if (DATtable_d. i. j=1)|(DATtable_d. i. j=-1):type+
                repeat strlen(SSTR):i=cnt
                    SSTR. 1=strmid(SSTR, i, 1)
                    repeat 1
                        if type=0:if SSTR. 1="/" :color 0,
0, 255:type+:break
                        if type=1:if SSTR. 1="/" :color 24, 24,
32:type+:break
                        if type=2:

```

```

color 0, 128, 0:type+:break

if type=3:if SSTR. 1="" :color 24, 24,
32:type+:break

if type=4:

color 24, 24, 32:type+:break

loop
if (type=5)&(Tfnc=n-1):color 255, 0, 0
pos x+fw*i, y:mes SSTR. 1

loop
}
}
if n=DATfnc{
color 0, 0, 255
line cellx*fw+dx, dy, cellx*fw+dx, (celly+TBLh. n)*fh+dy
}
cellx+=TBLw. m

loop
color 0, 0, 255
line dx, dy+celly*fh, dx+cellx*fw, dy+celly*fh
if n=DATfnc:line dx, dy+celly*fh+1, dx+cellx*fw, dy+celly*fh+1
celly+=TBLh. n

loop

; -- Table Out Frame
color 0, 0, 255
LineBox dx, dy, dx+cellx*fw, dy+celly*fh
LineBox dx-1, dy-1, dx+cellx*fw+1, dy+celly*fh+1

; -- Cursor Mark
r=sin(FADE)*30+220
cellx=0
repeat DATvar:m=cnt
celly=0
i=DATfnc-1:j=m-1
repeat DATfnc+1:n=cnt
if m<=1:break
if (DATtable_m. i. j)>=0)&(OPTION. 0. 0=1):break
if (DATbase_m. j!=0):break
if (n>0)&(n<DATfnc){
t=1:p=n-1:q=m-1
if (DATtable_m. p. q<0)&(OPTION. 0. 1=1):t=0
if DATtable_m. p. q=0:t=0
if t>0{

```

```

        cr=250:cg=r:cb=r
        if (n=Tfnc+1)&(m=Tvar+1)&(Todr=ORDER):cr=255:cg=0:cb=0
        if (n=Pfnc+1)&(m=Pvar+1)&(Pfnc>=0):cr=255:cg=0:cb=255
        x=dx+cellx*fw:w=TBLw.m*fw:y=dy+celly*fh:h=TBLh.n*fh
        a=1:color cr,cg,cb:LineBox x+a,y+a,x+w-a,y+h-a:cr=cr/2:cg=cg/2:cb=cb/2
        a=2:color cr,cg,cb:LineBox x+a,y+a,x+w-a,y+h-a
    }
}
celly+=TBLh.n
if (n=DATfnc)&(m=Tvar+1)&(Todr=ORDER){
    x=dx+cellx*fw:w=TBLw.m*fw:y=dy:h=celly*fh
    color 0,128,0:a=3:LineBox x+a,y+a,x+w-a,y+h-a
    color 0,255,0:a=4:LineBox x+a,y+a,x+w-a,y+h-a
    color 0,128,0:a=5:LineBox x+a,y+a,x+w-a,y+h-a
}
loop
cellx+=TBLw.m
loop

;          Display Position Y - Shift
celly=0
repeat TBLrow:celly+=TBLh.cnt:loop
    dy+=celly*fh+CELL*2
    if wy<dy:break
loop

;          Navi
color 0,0,0
pos 0,64:mes "VIEW="+ (VIEW+1)
pos 0,80:mes "L. 0. =" + (ORDER+1)

;          Object
gosub *OBJECT_BASE
Draw

if SYSTEMVIEW=3:goto *MAIN_TABLE_SHIFT
if SYSTEMVIEW=4:goto *MAIN_GRAPH_VIEW
if SYSTEMVIEW=5:goto *MAIN_OPTION
goto *MAIN_TABLE

*MAIN_TABLE_SHIFT
;          Init
TableRead Todr

```

ORDER=Todr+1

CLASS=Todr+1

; Pivot

PIBOT_M=DATtable_m.tfnc.tvar

PIBOT_D=DATtable_d.tfnc.tvar

repeat DATvar:m=cnt:n=tfnc

if DATtable_m.n.m=0:continue

a=DATtable_m.n.m*PIBOT_D

b=DATtable_d.n.m*PIBOT_M

c=CalFractPledge(a,b)

a=a/c

b=b/c

if b<0:b=-b:a=-a

DATtable_m.n.m=a

DATtable_d.n.m=b

loop

; Shift

repeat DATfnc:n=cnt

if n=tfnc:continue

ratio_m=DATtable_m.n.tvar

ratio_d=DATtable_d.n.tvar

repeat DATvar:m=cnt

if DATtable_m.tfnc.m=0:continue

a=DATtable_m.n.m*DATtable_d.tfnc.m*ratio_d

b=DATtable_m.tfnc.m*ratio_m*DATtable_d.n.m

c=DATtable_d.n.m*DATtable_d.tfnc.m*ratio_d

d=a-b

if d=0{

DATtable_m.n.m=0

DATtable_d.n.m=1

}else{

p=CalFractPledge(d,c)

a=d/p

b=c/p

if b<0:b=-b:a=-a

DATtable_m.n.m=a

DATtable_d.n.m=b

}

loop

loop

```

;          Make Table Display
TableDisplay
TableSave ORDER
CameraRangeReset

SYSTEMVIEW=2
goto *MAIN_TABLE
*MAIN_GRAPH_VIEW
;          Store
gsel ID_STORE
color 0,0,0
boxf 0,0,wx,wy
pos 0,0
gmode 3,wx,wy,128
gcopy ID_MAIN
pos (wx-GRPwx)/2,(wy-GRPwy)/2
gmode 2,GRPwx,GRPwy
gcopy ID_GRAPH,0,0
;          Main
gsel ID_MAIN
pos 0,0
gmode 0,wx,wy
gcopy ID_STORE
SHIFTx=(wx-GRPwx)/2+CELL-8
SHIFTy=(wy-GRPwy)/2+GRPwy-CELL-8

;          Stat
MOVESHEET=30
MOVETIME=0
MOVETYPE=0
x=0
y=0
repeat GRPOdr:o=cnt
    TableRead o
    GRPbs_m.o=DATbase_m.0
    GRPbs_d.o=DATbase_d.0
    GRPpx_m.o=DATbase_m.1
    GRPpx_d.o=DATbase_d.1
    GRPpy_m.o=DATbase_m.2
    GRPpy_d.o=DATbase_d.2
    if o=0{
        n=DATfnc-1
        vx=DATtable_m.n.1

```

```

                vy=DATtable_m.n.2
            }
        loop

*MAIN_GRAPH_WAIT
    KeyIn
    MOVETIME+
    if MOVETIME>MOVESHEET:MOVETIME=0:MOVETYPE+
    if MOVETYPE>=GRPodr-1:MOVETYPE=0

;        Background
gmode 0,wx,wy
pos 0,0:gcopy ID_STORE,0,0

;        Cursor
repeat GRPodr:o=cnt
    color 0,0,0
    px=x
    py=y
    x= int(double(GRPpx_m.o)*MEASURE/double(GRPpx_d.o))+SHIFTx
    y=-int(double(GRPpy_m.o)*MEASURE/double(GRPpy_d.o))+SHIFTy
    if (o>0)&(o=MOVETYPE+1){
        p=(x-px)*MOVETIME/MOVESHEET+px
        q=(y-py)*MOVETIME/MOVESHEET+py
        t=1
        pos p,q
        gmode 4,16,16,128
        gcopy ID_SYSTEM,t*16,223
    }
    if o=GRPodr-1{
        color 255,0,0
        SDV=atan(vx,vy)
        p=x+8
        q=y+8
        s=p-sin(SDV)*CELL
        t=q+cos(SDV)*CELL
        line p,q,s,t
        p=s
        q=t
        SDV+=2.80
        s=p-sin(SDV)*8
        t=q+cos(SDV)*8
        line p,q,s,t
    }
}

```

```
SDV=-5.60
s=p-sin(SDV)*8
t=q+cos(SDV)*8
line p,q,s,t
```

```
}
```

```
pos x,y
```

```
t=2:if o=GRPodr-1:t=3
```

```
gmode 2,16,16
```

```
gcopy ID_SYSTEM,t*16,223
```

```
gcopy ID_SYSTEM,0,223
```

```
;
```

```
SSTR="" + DATvarname.1 + "" + StrFract (GRPpx_m.o, GRPpx_d.o) + "¥n" + DATvarname.2 + "" + StrFract (GRPpy_m.o, GRPpy_d.o) + "¥n" + DATvarname.0 + "" + StrFract (GRPbs_m.o, GRPbs_d.o)
```

```
;color 32,32,32
```

```
;pos x+17,y-3*fh+1:mes SSTR
```

```
;color 128,128,255:if o=GRPodr-1;color 255,0,0
```

```
;pos x+16,y-3*fh+0:mes SSTR
```

```
loop
```

```
; Return Table
```

```
hit=0
```

```
if KEYsign.4=1:hit+
```

```
if hit>0:SYSTEMVIEW=2:goto *MAIN_TABLE
```

```
Draw
```

```
await STSTEMWAIT
```

```
goto *MAIN_GRAPH_WAIT
```

```
*MAIN_OPTION
```

```
; Store
```

```
gsel ID_STORE
```

```
color 0,0,0
```

```
boxf 0,0,wx,wy
```

```
pos 0,0
```

```
gmode 3,wx,wy,128
```

```
gcopy ID_MAIN
```

```
; Main
```

```
gsel ID_MAIN
```

```
pos 0,0
```

```
gmode 0,wx,wy
```

```
gcopy ID_STORE
```

```
w=320:h=240
```

```
SHIFTx=(wx-w)/2
```

```

SHIFTy=(wy-h)/2
x1=SHIFTx
y1=SHIFTy
x2=x1+w
y2=y1+h
color 160,160,160:boxf x1,y1,x2,y2
color 224,224,224:line x1,y1,x2,y1
color 064,064,064:line x2,y1,x2,y2
color 064,064,064:line x2,y2,x1,y2
color 224,224,224:line x1,y2,x1,y1
SHIFTx+=16
SHIFTy+=16

color 224,224,224:boxf SHIFTx+030,SHIFTy+015,SHIFTx+232,SHIFTy+048

color 000,000,000
pos SHIFTx,SHIFTy:mes "ピボット選択制限オプション"
pos SHIFTx+032,SHIFTy+018:mes "目的制約が負数なら選択可 ... "
pos SHIFTx+032,SHIFTy+034:mes "基底変数が正数なら選択可 ... "

objmode 2,0
objsize 16,16
pos SHIFTx+200,SHIFTy+016:checkbox "",OPTION.0.0
pos SHIFTx+200,SHIFTy+032:checkbox "",OPTION.0.1

x1=SHIFTx
y1=y2-32
x2=x2-16
y2=y2-32
color 224,224,224:line x1,y1+0,x2,y2+0
color 064,064,064:line x1,y1-1,x2,y2-1
objsize 64,24:pos (wx-64)/2,y1+4
button goto "決定",*MAIN_OPTION_END

Draw

*MAIN_OPTION_EDIT
  await STSTEMWAIT
  goto *MAIN_OPTION_EDIT
*MAIN_OPTION_END
  clrobj
  SYSTEMVIEW=2
  goto *MAIN_TABLE

```

_____ Sub Routine

*OBJECT_BASE

_____ Back Ground

r=wx/CELL:if wx≠CELL>0:r+

gmode 0,CELL,CELL

repeat 2:m=cnt

repeat r:n=cnt

pos n*CELL,m*CELL

gcopy ID_SYSTEM,m*CELL*2,CELL

loop

loop

pos 0,CELL:gcopy ID_SYSTEM,1*CELL,CELL

pos wx-CELL,CELL:gcopy ID_SYSTEM,3*CELL,CELL

_____ Object

x=8:y=4:w=24:h=w:gmode 2,w,h:color 128,64,64

repeat OBJnum:n=cnt:z=w*n

if OBJid.n=0:continue

hit=0

if (csrx>=x+z)&(csrx<x+w+z)&(csry>=y)&(csry<y+h) {

hit+

SOM=JessicaComment(int(OBJid.n))

if (KEYsign.4>1)&(OBJcnd.n=2):hit+

if (KEYsign.4=1):hit+

if (KEYsign.4=0)&(OBJcnd.n=2):ObjectAction int(OBJid.n)

}

pos x+z,y:gcopy ID_SYSTEM,w*hit,183

pos x+z,y:gcopy ID_SYSTEM,w*OBJid.n,0

OBJcnd.n=hit

loop

_____ JessicaComment

pos 8,CELL+10 :color 64,32,32:mes SOM

return

_____ Program End