

経済数学における概念の数式表現のための 知識データベース構築

白田 由香利[†]

要旨

本稿では、学部レベルの経済数学で必要とされる、(a) 連立方程式の経済学への応用、(b) 経済学における導関数の用法、(c) 経済学における多変数関数の計算の3分野について、その分野の問題の解法に必要とされる用語及び概念の数式すべてを、Prologのルールとして表現し、学部レベルの経済数学問題の解法に必要な知識ベースを構築した。これにより、問題として与えられたデータ以外の式を導出し、必要な数の方程式を揃えることが可能となり、与えられた経済数学問題に対して、経済の領域知識を用いて正しい問題解決過程を求めることが可能となる。

1. 始めに

本稿では、従来からデータベースの分野と人工知能の境界領域の研究として注目を集めている知識データベースの構築を、経済数学に出現する概念及び用語の数式表現を対象として行なった結果を報告する。我々は経済数学に関するITS(Intelligent Tutoring System)の研究開発を行なっているが、このITSを構築する場合、経済数学に関する知識データベースの構築が必要となってくる。まず我々が想定するITSに求められる機能であるが、以下の2つがある。

(1) 与えられた経済数学問題に対して、経済の領域知識を用いて正しい問題解決過程を求める。

(2) 問題解決過程において言葉やグラフ、アニメーションなどを用いて学習者の質問に答える。

本論文では上記(1)を、証明問題を除く決定問題に限定した場合を考える*。つまり与えられた経済数学問題の中心的部分には、必ず連立方程式が在る。そして問題解決のためには、まず必要な連立方程式をセットアップする必要がある。与えられた文章中には、連立方程式を作成するために必要なデータや条件が記述されているが、一部のデータや条件は経済の基本的知識として記述が省力されている場合がある。学習者がそうした経済数学の基本的知識に気がつかない場合あるいは元々知らない場合、ITSは学習者支援のために学習者に代わってその知識を

[†] 学習院大学 経済学部, 〒171-8588東京都豊島区目白 1-5-1。

E-mail: yukari.shirota@gakushuin.ac.jp.

* 上記(2)については、既に発表したe-Math Interaction Agentの論文を参照して頂きたい(文献8参照)。

もつ必要がある。この経済数学の基本的知識をどのように知識ベース化するかが本論文の主題である。

我々は学部レベルの経済数学で必要とされる，(a) 連立方程式の経済学への応用，(b) 経済学における導関数の用法，(c) 経済学における多変数関数の計算の3分野について，その内容を調べ，その分野の問題の解法に必要な用語及び概念の数式すべてを，Prologのルールとして表現し，経済数学問題の解法に必要な知識ベースを構築した。参考とした経済数学のテキストは，Dowlingの経済数学の入門書であり，広く普及しているテキストである¹。また，必要となる用語及び概念を網羅するために，GREの問題集²の用語集，及び他のDowlingの著作となるテキスト^{3,4}，及び広く利用されている経済数学テキスト⁵を参照した。

ITSとしての一般的，解決可能な問題の範囲についての議論は，人工知能のCAIの分野で広く研究されている。この分野のサーベイとしては，例えば，大槻らの論文が参考となるであろう⁶。本論文の主題は，そうしたCAIシステムの限界的能力を対象とするものではない。また，限定分野に対してわずかの初期知識から新しい知識を発見するというような，発見システム[†]の開発でもなく，本稿の目的は，経済数学向けe-Learningシステムにおいていかに数学知識を表現するかという問題に対して我々の採った現実的アプローチを説明するものである。データベースにおける知識獲得についての汎用技術は，西尾の論文などを参考として頂きたい⁷。

2. 知識の分類

本節では，経済数学問題を解く場合必要となる数学表現知識が3種類あることを示し，そのうちのどの部分を知識ベース化するのを示す。

一般的に数学問題では，その問題（文章題）中に与えられるデータの他，数学公式およびその分野に特化する知識（セオリー）がある。我々が研究開発したe-Math Interaction Agentにおいては，前者は，メタレベル記述ファイルという問題定義ファイルで与えられる⁸。こちらは個々の文章題に特化したデータである。他方，後者にあたる知識は一般性が高くなる。経済数

-
- 1 Edward T. Dowling: Mathematics for Economists, McGraw-Hill, 1980. 訳：大住栄治，川島康男 訳，例題で学ぶ入門経済数学 [上]，シーエービー出版，1995。
 - 2 Georgene Gallagher et al.: Best Test Preparation for the Gre in Economics (Gre Economics Test), Research & Education Association, 1989.
 - 3 Edward T. Dowling: Schaum's Outline of Theory and Problems of Calculus for Business, Economics, and the Social Sciences (Schaum's Outlines), McGraw-Hill, 1990.
 - 4 Edward T. Dowling: Schaum's Outline of Theory and Problems of Mathematical Methods for Business and Economics (Schaum's Outlines), McGraw-Hill, 1992.
 - 5 Ian Jacques: Mathematics for Economics and Business (4th ed.), Prentice Hall, 2003.
 - 6 大槻説平 他: 知識科学の展開，大須賀節雄 他 編，第5章「CAIにおける知識獲得」，オーム社，1996。
 - 7 西尾章治郎 他: 知識科学の展開，大須賀節雄 他 編，第4章「データベースにおける知識獲得」，オーム社，1996。
 - 8 白田由香利: 「経済最適化問題を例とする数学Web教材自動作成システム」，学習院大学経済論集，第41巻，第1号（通巻122号），2004年4月，pp.1-15。
- † 例えば，DST (Discovery System for Trigonometric Function) では，三角関数を対象として，補助線を適当にひきながら，三角形の角及び辺の間にある重要な関係式を導出していく (Murata, T., Mizutani, M., and Shimura, M.: "A Discovery System for Trigonometric Functions," Proc. of AAAI, pp.645-650, 1994.)

学の場合，経済に特化する知識が後者のセオリーに当たる。この他，数学問題の解法プラン，あるいは問題解決過程の知識が必要となる。例えば，最大値を求める数学的手法がこれに相当する。一般に解法プランはアルゴリズムとして定義される。我々の e-Math Interaction Agent では，解法プランは予め問題のタイプごとに定義する。

まとめると以下の3種類に知識は分類される。

- (1) 経済に関する知識
- (2) 個々の問題に特化して与えられるデータ
- (3) 問題タイプに対する解法プラン

問題に対して連立方程式を立てる際，(1)及び(2)の知識及びデータが必要となる。その導出を Prolog で行っている様子を図1に示す。ここでは経済知識として，profit, revenue, cost の知識表現が定義されている。またこの問題として与えられたデータとして average cost, price の式が与えられている。知識は rule，与えられたデータは data で定義されている。両者とも，第1引数である変数を定義するための引数セット及び，その変数間の関係性を定義している。例えば revenue は引数 price 及び quantity で定義されていることがルールとして表現されている。

この問題の未知数を profit とした場合，set() を起動することで，Prolog でユニフィケーションを行い，結果の連立方程式 $R(C), R(P,Q), C(AC,Q), P(Q), AC(Q)$ を求めて List に入れている。data (quantity) はそれ自体他の変数で定義されていないので，つまりそれを定義する引数が空リストなので，ここで検索が止まる。この検索エンジンでは，このように独立変数として残したい変数を data として追加しておく必要がある。

必要な連立方程式が一旦求められた後は，数式処理システムによって記号により方程式を解

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help
4 ?- listing(rule).

rule(profit, 'Pi', 'Pi:=R-C', [revenue, cost]).
rule(revenue, 'R', 'R:=P*Q', [price, quantity]).
rule(cost, 'C', 'C:=AC*Q', [averageCost, quantity]).

Yes
5 ?- listing(data).

data(averageCost, 'AC', 'AC:=2000/Q+10', [quantity]).
data(price, 'P', 'P:=210-10*Q', [quantity]).
data(quantity, quantity, 'Q=Q', []).

Yes
6 ?- set(profit,List).

List = ['Pi:=R-C', 'R:=P*Q', 'C:=AC*Q', 'P:=210-10*Q', 'AC:=2000/Q+10', 'Q=Q']

Yes
7 ?- set(revenue,List1).

List1 = ['R:=P*Q', 'P:=210-10*Q', 'Q=Q']
    
```

図1：経済知識とデータから方程式を求める

くことが可能である。従来のITSの多くは開発システム中にPrologの数式処理システムを内包することが求められたが、現在のように数式処理システムが容易に起動可能となった状況では、必要に応じて数式処理システムのsolve関数を呼び出すほうがシステム構築は効率良く行えると考えられる。MapleT.A.というシステムでは、学習者が入力した数式が、教師が想定した数式と、syntactical equalであるか否かのチェックを行うため、Mapleを起動して計算をするような機能がある。このように必要な数式処理はその都度数式処理システムを起動する、という方式が実現可能な時代となったと言える。

上記の問題においても、Mapleシステムのsolve関数によりprofitをquantity Qの関数として求めることが可能である。Prologシステムの役割は、経済知識を用いて必要な数の連立方程式を正しく求めることである。

3. 方程式セットアップのための要件

与えられた経済問題に対して連立方程式を立てるためには、必要な数の方程式を知識ベースから発見し、未知数を指定の変数で表現可能とする必要がある。そのための関数としてset()を作成した。関数setは、指定の未知数（unknown）を、指定の引数リスト（paramList）のみで表現可能な連立方程式に展開する。結果の連立方程式はresultListに入る。

- set(unknown, paramList, resultList)

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help
% math8 compiled 0.00 sec, 5,672 bytes

Yes
3 ?- listing(rule).

rule(profit, 'Pi', 'Pi:=R-C', [revenue, cost]).
rule(revenue, 'R', 'R:=P*Q', [price, quantity]).
rule(revenue, 'R', 'R:=P1*Q1+P2*Q2', [price1, price2, quantity1, quantity2]).
rule(cost, 'C', 'C:=AC*Q', [averageCost, quantity]).

Yes
4 ?- listing(data).

data(price1, 'P1', 'P1:=56-4*Q1', [quantity1]).
data(price2, 'P2', 'P2:=48-2*Q2', [quantity2]).
data(cost, 'C', 'C:=Q1^2+5*Q1*Q2+Q2^2', [quantity1, quantity2]).
data(quantity1, 'Q1', 'Q1=Q1', []).
data(quantity2, 'Q2', 'Q2=Q2', []).

Yes
5 ?- set(price1, [quantity1, quantity2], List).

List = ['P1:=56-4*Q1']

Yes
6 ?- set(profit, [quantity1, quantity2], List).

List = ['C:=Q1^2+5*Q1*Q2+Q2^2', 'P2:=48-2*Q2', 'P1:=56-4*Q1', 'R:=P1*Q1+P2*Q2', 'Pi:=R-C']

Yes
7 ?- expand(profit, [quantity, quantity2], List).

No
8 ?- █

```

図2：R(Q)及びR(Q1,Q2)のルール定義

但し、未知数は引数リストの全ての変数を含む必要がある[‡]。この条件が満たされないと関心のある知識表現にマッチしない場合があるからである。

収入 revenue の計算の場合、商品が2種類ある場合、 $R=P_1*Q_1+P_2*Q_2$ と表現される。経済知識ベースを構築する場合、関心のある引数リストごとに同じ revenue に対して複数の数式表現を格納しておく必要がある。図2のルールでは revenue に対して2つのルールが定義されている。Profit を quantity1 及び2 で展開しようとした場合、図2のようにデータを与えると、問合せは

set(profit, [quantity1,quantity2], List)

となり、結果として、以下のように5本の方程式が得られる。

$$\begin{aligned} \Pi &= R - C \\ R &= P_1 \times Q_1 + P_2 \times Q_2 \\ P_1 &= 56 - 4 \times Q_1, \quad P_2 = 48 - 2 \times Q_2 \\ C &= Q_1^2 + 5 \times Q_1 Q_2 + Q_2^2 \end{aligned}$$

これにより (Q1,Q2) の式が求められる。図2の最後に示された set による問合せは、profit を [quantity, quantity2] という間違っ引数セットで展開しようとするとう失敗する、ことを示している。

経済数学用知識ベース構築にあたり注意すべき点は、必要な使用頻度の高い数式表現を変数名が衝突しないように、複数の引数セットで表現することである。以下の章では、それぞれの分野ごとにどのような知識表現をルール化したか、また実際の経済数学問題との対応を、例をあげて示す。知識ベース化された経済知識は35個でありそれらは付録として添付したので、参照して頂きたい。

4. 連立方程式の経済学への応用

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help

Yes
4 ?- set(totalDemand,[income,interestRate],List).
List = ['I:=41-80*i', 'C:=34+0.4*Y', 'Yd:=C+I']

Yes
5 ?- set(moneyDemand,[income,interestRate],List).
List = ['Mt:=0.4*Y', 'Mw:=65-124*i', 'Md:=Mw+Mt']

Yes
6 ?- set('MPC',[consumption,income],List).
List = ['C:=34+0.4*Y', 'b:=diff(C,Y)']

Yes
7 ?-
    
```

図3：国民所得決定問題で連立方程式を立てる

[‡] 図2のバージョンのset()では、まだこの条件が実現されていない。

```

math8a.pl - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

%% National Income Determination
data(consumption, 'C', 'C:=34+0.4*Y', [income]).
data(investment, 'I', 'I:=41-80*i', [interestRate]).
data(interestRate, 'i', 'i=i', []).
data(moneySupply, 'Ms', 'Ms:=325', []).
data(speculativeDemandForMoney, 'Mw', 'Mw:=65-124*i', [interestRate]).
data(transaction-precautionaryDemandForMoney, 'Mt', 'Mt:=0.4*Y', [income]).
data(income, 'Y', 'Y=Y', []).

%% National Income Determination
rule(totalDemand, 'Yd', 'Yd:=C+I', [consumption, investment]).
rule(totalDemand, 'Yd', 'Yd:=C+I+G', [consumption, investment, governmentExpenditures]).
rule(totalDemand, 'Yd', 'Yd:=C+I+G+(X-Z)',
    [consumption, investment, governmentExpenditures, exports, imports]).
rule(totalSupply, 'Ys', 'Ys=Y', [income]).
rule('MPC', 'b', 'b:=diff(C,Y)', [consumption, income]). %%marginal propensity to consume|
rule(moneyDemand, 'Md', 'Md:=Mw+Mt',
    [speculativeDemandForMoney, transaction-precautionaryDemandForMoney]).

```

図 4 : 図 3 の問題のために与えられたデータ

```

SWI-Prolog Thread 2 (2796) Interactor
File Edit Settings Run
[1] 1 ?- group(income.Rules).

Rules = [totalDemand, totalSupply, savings, 'MPC', 'MPS', disposableIncome, consumption, moneyDemand]

Yes
[1] 2 ?- listing(data).

data(consumption, 'C', 'C:=0.9*Ydis+70', [disposableIncome]).
data(governmentExpenditures, 'G', 'G:=20', []).
data(investment, 'I', 'I:=35', []).
data(tax, 'T', 'T:=0.2*Y+25', [income]).
data(income, 'Y', [], []).

Yes
[1] 3 ?- set(consumption.[income],List).

List = ['T:=0.2*Y+25', 'Ydis:=Y-T', 'C:=0.9*Ydis+70']

Yes
[1] 4 ?- set(totalDemand.[investment,governmentExpenditures,income],List).

List = ['G:=20', 'I:=35', 'T:=0.2*Y+25', 'Ydis:=Y-T', 'C:=0.9*Ydis+70', 'Yd:=C+I+G']

Yes

```

図 5 : 総需要を I, G, Y で展開した例

まず連立方程式を立てるための経済知識としてどのようなルールを定義したかを説明する。経済数学の連立方程式に関する問題の主たるテーマとして国民所得決定問題があり、ルールとして図4に示すような2セクター、3セクター、輸入輸出まで入れた総需要の3モデルを定義した。データとして図4のdataが与えられた場合、CとIしか定義されていないため、モデルは2セクターモデルが選択される。その他のモデルが選択された場合、必要な変数が見つからないためエラーとなりその検索は終了する。例えば、Gなどはdataとして定義されていないためエラーとなり、そこで検索が止まる。

図3の例では、総需要、及び貨幣需要の変数を、引数 [国民所得 Y, 利率 i] で展開した様子を示した。結果は正しく2セクターモデル $Y_d=C+I$ が選択された様子を示している。また貨幣需要も正しく展開されたことが示されている（図3参照）。

図3の3番目の例はMPCを引数C及びYで展開しようとした問題である。結果として問題として与えられたCの方程式 $C(Y)$ 及び、MPCの定義式が検索により得られた。MPCの知識ルールでは記号bを使って関係を定義しているので、このような結果となる。Prologのユニフィケーションでは、単に定義された関係式を返り値として返すだけなので、自由な記号を使って式が表現可能である。ここでdiffはMapleの表現で微分及び偏微分を意味する。Diffでは第1引数を第2引数により微分する。このset関数の返り値である方程式をMapleに渡せば、具体的な計算がそのまま実行できるように、このようなdiff表現を用いた。

図5の例は、総需要をI,G,Yで展開した例を示した。ここでは、与えられたCが間接的に税Tを含むため、 $Y_{dis}=Y - T$ の式が結果に含まれる。この展開においても総需要は正しく3セクターモデルを選択していることが確認された。

5 . 経済学における導関数の用法

```

%% Total, Marginal, Average
%% Elasticity
rule(profit, 'Pi', 'Pi:=R-C', [revenue, cost]).
rule(revenue, 'R', 'R:=P*Q', [price, quantity]).
rule(revenue, 'R', 'R:=P1*Q1+P2*Q2', [price1, price2, quantity1, quantity2]).
rule(cost, 'C', 'C:=AC*Q', [averageCost, quantity]).
rule('MC', 'MC', 'MC:=diff(C,Q)', [cost, quantity]). %%marginal cost
rule('MR', 'MR', 'MR:=diff(R,Q)', [revenue, quantity]). %%marginal revenue
rule('MC', 'MC', 'MC:=diff(C,Q)', [cost, quantity]). %%marginal cost
%%point of elasticity of demand
rule(elasticity, 'e', 'e:=diff(Qd,P)*(P/Qd)', [quantityDemanded, price]).

data(quantityDemanded, 'Qd', 'Qd:=650-5*P-P^2', [price]).
data(price, 'P', 'P:=P', []).
data(quantity, 'Q', 'Q:=Q', []).
data(price1, 'P1', 'P1:=56-4*Q1', [quantity1]).
data(price2, 'P2', 'P2:=48-2*Q2', [quantity2]).
data(cost, 'C', 'C:=Q1^2+5*Q1*Q2+Q2^2', [quantity1, quantity2]).
data(quantity1, 'Q1', 'Q1:=Q1', []).
data(quantity2, 'Q2', 'Q2:=Q2', []).
    
```

図6：導関数に関する経済知識と与えられたデータ

次に経済における導関数の用法の分野を説明する。この分野では主に最適化問題と価格弾力性の定義を行った（図6参照）。式はMapleを使うことにより、ある変数について解くことが可能である。例えば需要関数をデータとして与える際、式としては陰関数の形で表現することも可能であり、変数の表現形式は解が定まる限り問わないからである。また式がPで与えられていた場合、逆関数Qを求めるなどの操作も、Mapleに任せるべき作業であり、経済知識としては、関係式のみを定義すればよい。

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help
7 ?- listing(data).

data(cost, 'TC', 'TC:=Q^3+5*Q^2+60*Q', [quantity]).
data(quantity, 'Q', [], []).

Yes
8 ?- set(averageCost, [quantity], List).

List = ['TC:=Q^3+5*Q^2+60*Q', 'AC:=C*Q']

```

図7 a : 平均費用の最適化問題

```

Maple 9.5 - Untitled (2) - [Server 2]
> eqs:={TC=Q^3-5*Q^2+60*Q, AC=TC/Q};
      eqs := { AC = TC/Q, TC = Q^3 - 5 Q^2 + 60 Q }
> sols:=solve(eqs);
      sols := [AC = Q^2 - 5 Q + 60, TC = (Q^2 - 5 Q + 60) Q, Q = Q]
> assign(sols);
> deriv1:=diff(AC, Q);
      deriv1 := 2 Q - 5
> solve(deriv1=0, Q);
      5
      2
> subs(Q=5/2, AC);
      215
      4

```

図7 b : Mapleによって式を解く様子

次に、平均費用を最小化する点を求める最適化問題を解く場合を考える。文章題で総費用関数が与えられるとする。あとひとつ必要な式は平均費用の定義式である。これはルールとして定義されている。問合せ `set(averageCost, [quantity],List)` によって、ACがQの関数として展開される（図7 a 参照）。得られた2式は、AC(Q)を計算するためにMapleに渡される。図7bにMapleで処理される計算の過程を示した。この命令の与え方は人間が対話的に入力する方式であるが、同様の過程を予め定義されている解法プランに従いシステムが行うようにシステムを構成することが可能である（実際には2階微分を計算して、得られた極小値が最小値であることを証明する必要があるが、ここでは省略している）。

需要の所得弾力性に関する式の展開例を次に見てみよう（図8参照）。

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help
9 ?- set(elasticity, [quantityDemanded, price], List).
List = ['Qd:=650-5*P-P^2', 'e:=diff(Qd,P)*(P/Qd)']
Yes
10 ?-
    
```

図8：需要の所得弾力性に関する展開

はじめに需要の式をデータとして与える。問合せとして `set(elasticity,...)` を与えると、結果として図8の2式が返る。第1式は与えられた需要の式であり、第2式は、所得弾力性の定義式である。

6. 経済学における多変数関数の計算

```

SWI-Prolog (Multi-threaded, version 5.4.5)
File Edit Settings Run Debug Help
Yes
16 ?- listing(data).
data(equilibriumIncome, 'Y', 'Y:=C+I0+G0+X0-Z', [consumption, import]).
data(tax, 'T', 'T:=T0+t*Y', [income]).
data(consumption, 'C', 'C:=C0+b*Ydis', [disposableIncome]).
data(import, 'Z', 'Z:=Z0+z*Ydis', [disposableIncome]).
data(income, 'Y', [], []).
data(autonomousG, 'G0', [], []).
Yes
17 ?- set(equilibriumIncome, [income], L).
L = ['Z:=Z0+z*Ydis', 'T:=T0+t*Y', 'Ydis:=Y-T', 'C:=C0+b*Ydis', 'Y:=C+I0+G0+X0-Z']
Yes
18 ?- set(governmentMultiplier, [income, autonomousG], L).
L = ['Z:=Z0+z*Ydis', 'T:=T0+t*Y', 'Ydis:=Y-T', 'C:=C0+b*Ydis', 'Y:=C+I0+G0+X0-Z', 'dYdG:=diff(Y*,G0)']
Yes
19 ?-
    
```

図9：国民所得決定問題の政府乗数の展開例

```

SWI-Prolog Thread 2 (3264) Interactor
File Edit Settings Run
SWI-Prolog console for thread 2

1 ?- listing(data).

data(quantityDemanded1, 'Qd1', 'Qd1:=4850-5*P1+1.5*P2+0.1*Y', [price1, price2, income]).
data(income, 'Y', 'Y:=10000', []).
data(price1, 'P1', 'P1:=200', []).
data(price2, 'P2', 'P2:=100', []).

Yes
2 ?- set(cross-elasticityOfDemand, [income, price1, price2], List).

List = ['Y:=10000', 'P2:=100', 'P1:=200', 'Qd1:=4850-5*P1+1.5*P2+0.1*Y', 'E12:=diff(Qd1,P2)*(P2/Qd1)']

Yes
3 ?- █

```

図 10：需要の交差弾力性の展開例

本節では経済における多変数関数の計算に関する知識の定義を説明する。

多変数関数の問題の多くは、前章で述べた最適化問題の多変数化である。Maple では diff で偏微分も表現できるので、多変数化によって加える必要のある知識は、revenue などの変数のみであり、微分に関する定義に変更をする必要はない。また図 9 に示したような国民所得決定問題における各種の乗数の定義をルール化した（付録参照）。

多変数関数に関する問題としてラグランジェの未定乗数法がある。ラグランジェ式を知識として入れるべきか考えた結果、この定義式は解法プランに属するものであるとし、経済知識ルールには入れないこととした。あくまでも知識を、経済知識ルール、与えられたデータ、解法プランの 3 つに類別し、それぞれの独立性を高めたほうがシステムの寿命が長くなるからである。

需要の交差弾力性、価格弾力性などの定義も経済知識ルールとして入れた。図 10 では需要関数式および、所得、2財の価格式を与え、その需要の交差弾力性の式を求める展開を行った。変数が 5 個、式が 5 本得られたので、これは Maple の solve で解くことが可能である。

7. まとめ

本稿では、学部レベルの経済数学で必要とされる、(a) 連立方程式の経済学への応用、(b) 経済学における導関数の用法、(c) 経済学における多変数関数の計算の 3 分野について、その分野の問題の解法に必要なとされる用語及び概念の数式すべてを、Prolog のルールとして表現し、経済数学問題の解法に必要な知識ベースを構築した。こうした知識データベース及び Prolog エンジンによって、問題として与えられたデータ以外の式を導出し、必要な数の方程式を揃えるこ

とが可能となる。方程式を解くことは Maple の solve 関数を呼び出すことで可能となる。これにより、与えられた経済数学問題に対して、経済の領域知識を用いて正しい問題解決過程を求めることが可能となる。

定義した経済知識ルールは35個であった。この35ルールの拡張として、国民所得決定問題の各種の乗数定義を追加するなどが考えられるが、この35のルールの組合せで正しく知識が表現できていることを確認できた。その確認のため、代表的な経済数学問題をデータとして与え、式の展開が正しく行えることを確認した。この知識ルールは入れ替える必要はない。

数学用 Web 教材システム生成システムを研究開発してきた経験から得られた重要なことは、必要な知識を3つのグループ、(1) 経済知識、(2) 文章題で与えられたデータ、(3) 問題タイプに対する解法プラン、に分けることであった。これらを独立化することにより夫々の知識の独立性を向上させ、そのデータベースを独立に拡充することが可能となる。(2) のデータに関して、ドリル問題を多数データベース化するという目的のため、独立化することは重要である。経済知識データベース及び、解法プランのデータベースの充実ももちろん今後一層行われるべきものとする。

Prolog による知識データベースへの問合せ機能、及び Maple による方程式の解法機能を適宜起動することで、既存の数学に関する ITS 及び e-Learning システムの機能はさらに向上し学習者の学習を支援するものとする。どのような組み合わせが有効であるのか、今後とも研究を重ねていきたい。

謝辞 本研究の一部は、平成17年度科研費基盤研究(C)(2)「マルチメディア教育支援システム eMath における教育用データベースの構築」(課題番号：15606014、代表：白田由香利)による。ここに記して謝意を表します。

付録

%% Optimization

%% Total, Average, Marginal

- rule(profit, 'Pi', 'Pi:=R-C', [revenue, cost]).
- rule(revenue, 'R', 'R:=P*Q', [price, quantity]).
- rule(revenue, 'R', 'R:=P1*Q1+P2*Q2', [price1, price2, quantity1, quantity2]).
- rule(cost, 'C', 'C:=AC*Q', [averageCost, quantity]).
- rule('MC', 'MC', 'MC:=diff(C,Q)', [cost, quantity]).
- rule('MR', 'MR', 'MR:=diff(R,Q)', [revenue, quantity]).
- rule('MC', 'MC', 'MC:=diff(C,Q)', [cost, quantity]).
- rule('MPk', 'MPk', 'MPk:=diff(Q,K)', [quantity, capital]).
- rule('MPPk', 'MPPk', 'MPPk:=diff(Q,K)', [quantity, capital]).
- rule('MPl', 'MPl', 'MPl:=diff(Q,L)', [quantity, labor]).
- rule('MPPl', 'MPPl', 'MPPl:=diff(Q,L)', [quantity, labor]).
- rule('MRP', 'MRP', 'MRP:=MR*MP', ['MR', 'MP']).

- rule('MRTS', 'MRTS', 'MRTS:=diff(K,L)', [capital, labor]).
- %% Elasticity
- rule(price-elasticityOfDemand, 'Ep', 'Ep:=diff(Qd,P)*(P/Qd)', [quantityDemanded, price]).
- rule(price-elasticityOfSupply, 'Es', 'Es:=diff(Qs,P)*(P/Qs)', [quantitySupplied, price]).
- rule(income-elasticityOfDemand, 'Ey', 'Ey:=diff(Qd,Y)*(Y/Qd)', [income, quantityDemanded]).
- rule(cross-elasticityOfDemand, 'E12', 'E12:=diff(Qd1,P2)*(P2/Qd1)', [quantityDemanded1, price2]).
- %% National Income Determination
- rule(totalDemand, 'Yd', 'Yd:=C+I', [consumption, investment]).
- rule(totalDemand, 'Yd', 'Yd:=C+I+G', [consumption, investment, governmentExpenditures]).
- rule(totalDemand, 'Yd', 'Yd:=C+I+G+(X-Z)', [consumption, investment, governmentExpenditures, exports, imports]).
- rule(governmentMultiplier, 'dYdG', 'dYdG:=diff(Y*,G0)', [equilibriumIncome, autonomousG]).
- rule(importMultiplier, 'dYdZ', 'dYdZ:=diff(Y*,Z0)', [equilibriumIncome, autonomousZ]).
- rule(exportMultiplier, 'dYdX', 'dYdX:=diff(Y*,X0)', [equilibriumIncome, autonomousX]).
- rule(taxationMultiplier, 'dYdT', 'dYdT:=diff(Y*,T0)', [equilibriumIncome, autonomousT]).
- rule(totalSupply, 'Ys', 'Ys:=Y', [income]).
- rule(savings, 'S', 'S:=Y-C', [income, consumption]).
- rule('MPC', 'b', 'b:=diff(C,Y)', [consumption, income]).
- rule('APC', 'APC', 'APC:=C/Y', [consumption, income]).
- rule('MPS', 'a', 'a:=diff(S,Y)', [savings, income]).
- rule('APS', 'APS', 'APS:=S/Y', [savings, income]).
- rule('MPS', 'a', 'a:=1-b', ['MPC']).
- rule(disposableIncome, 'Ydis', 'Ydis:=Y-T', [income, tax]).
- rule(consumption, 'C', 'C:=a*Y+b', [autonomousConsumption, 'MPC', income]).
- rule(consumption, 'C', 'C:=a*Ydis+b', [autonomousConsumption, 'MPC', disposableIncome]).
- rule(moneyDemand, 'Md', 'Md:=Mw+Mt', [speculativeDemandForMoney, transaction-precautionary-DemandForMoney]).