

## 報告書：2013 年度

### 「年金財政シミュレーションツールの開発に向けた研究」

#### 1 はじめに

公的年金制度は、5年ごとに、直近の社会・経済情勢等を踏まえて設定した基礎数値を使用し、制度内容に沿って将来の財政見通しを作成して、長期的な財政の持続可能性の確認が行われている。そのための基礎数値やシミュレーションプログラムは、厚生労働省のホームページに公開されており、厚生労働省が想定した前提条件のもとでの財政検証結果の再現が可能なものとなっている。年金財政シミュレーションツールの開発にあっては、この公開されているプログラムについて、これを一般に実行できるようにし、その出力結果を見ながらそのプログラムの改訂を行い、より年金財政の研究に適したシミュレーションツールを開発することとした。

そこで、開発・研究方針として、始めに「厚生労働省のプログラムの実行環境の整備」を行うこととした。これは、厚生労働省のプログラムが、プログラミング言語として用いているのが FORTRANとC++であり、計算機はSUN(オラクル社)を用いることを前提に作成されている。そのために一般の研究者がこのプログラムを厚生労働省と同じ構成で実行しようとしても、この実行環境と同じものを用意することは難しい。そこで、Windows上でも実行できるようにプログラムの変更し実行する計算機がSUNである必要性をなくし、そのつぎに「使用言語の統一」を行う。これは、プログラミング言語として、FORTRANとC++を厚生労働省のプログラムでは使用しているが、現在ではFORTRANは一般的なプログラムには使用することは少ない。また、プログラムの変更や可読性に欠けている部分があるので、FORTRANで書かれたプログラムをC++に変更する。このことより、Windows上でよく使用されるVisual Studio C++でコンパイルすることが可能となるので、研究者がパラメータを変更したり、プログラムを改変して実行することが容易となる。3番目に行うこと「実行用のユーザーインターフェースの作成」である。厚生労働省の実行プログラムは5つのフォルダーの中にありこれを順番に実行することで、全体の計算が終了する形式となっている。その実行に際して、整合性を持ったいろいろなパラメータを設定する必要があるので、これを一貫して実行するためのユーザーインターフェースの作成を行い、それをシミュレーションツールとして実装することが必要となる。

また、厚生労働省の実行プログラムでは、最終的なファイルはExcelのCSVファイルが3つ生成されるが、2つは調整率の表であり、検証結果として重要なのは1つのファイルだけである。このファイルは、ExcelのCSVファイルとして保存されているが、この形式では分かりにくく必要な項目を捜すのに非常に手間がかかる。そこで、財政検証レポートに掲載されているような表の一部を作成するような「出力ファイルのExcelファイルへの変換」をするソフトウェアの開発をした。

そして、シミュレーションツールを作成していく上で、年金システムの知識を深めるための勉強は必要であるので、また委員全体の知識の共有するため「年金財政の勉強会」を機会を設けて行うことにした。

以上のことを実行するために定例会を月に1回開催し、その開発の進捗状況について報告し、また議論することにした。

## 2 プログラム

プログラムを改訂をするにあたっては、まず始めに厚生労働省の財政検証プログラムを実行できる環境を作成した。つぎに、厚生年金のバックデータを展開して使用しようとしても直ちに使える訳ではないので、その実行を可能になるようにして、その計算結果とデータに含まれている結果との比較検討をした。しかし、その比較においてはかなり違ったものになり、そのためコンパイルするパラメータやその原因を調べることに時間を費やしたが、誤差精度を $10^{-5}$ 程度にして環境が整ったものとした。その後、C++に変換してその精度がFortranの場合と比較して $10^{-10}$ 程度に収めるようにした。ここまでは、バックデータにおけるプログラムと1行単位でなるべく変更しないようにしたが、計算時間がFortranの場合より増加してしまったので、その削減をするためにプログラムの書き換えをすることにした。また、メモリーの使用量もFortranの場合は大きいのでこれの削減化も試行した。実際の工程における作業内容はつぎようである。

### 1. 厚生労働省のプログラムについて

厚生労働省で配布している財政検証バックデータ(s0223-9i.zip)は、プログラムファイルとそのデータファイルが入っている。実行するプログラムファイルは、被保険者推計、国民保険、基礎年金、厚生年金/給付費推計、厚生年金/収支計算の5つのフォルダーに入っており、データは、rev21フォルダーに入っている。そこで、始めにこのプログラムの実行をできるようにした。この作業においては、実際に稼働させるには、プログラム言語の基礎専門知識が要求されるものであった。

- 厚生労働省のプログラムにおいては、ZIPファイルを展開してデータフォルダーが作成されるが、その配置が実際にプログラムで実行されるときに指定と違っているものがあつた。また、計算結果を保存するためのフォルダーが作成されていないので、これを作成する必要があつた。
- 初期値の設定について、プログラム内で初期化をする場合に、配列数が違っている場合があつたが実行上の問題点ではなかつた。
- IF文において、計算が不安定になる場合があり、これについては強制的に閾値を定めて判定するようにした。(厚生年金/給付費推計)。これは、いろいろなプラットフォーム上(計算機やコンパイラ)で実行するとき計算の違いを生じたため、その原因を調べている際に分かつたものであり、コンパイル方法の違いにより生じていると思われる。

- プログラム実行する際に、必要とされるパラメータについては、その多くはデータファイルを読むの使用されるパラメータであった。そのため、プログラムの中で使用されるファイル名とデータにあるファイル名と付き合わせて、そのパラメータを推定した。このことは、それ以外のパラメータを設定では対応するデータがないので、実行できなくなることを意味する。

## 2. C++の統一化

FORTRANからC++への変換は、厚生年金の計算における給付推計と収支計算のみがFORTRANになっていたのをこれを対象に変換を施した。コマンドなどの変換は、Perlのスクリプトを書いて変換したが、配列についてはそのままでは動かないので変換する際には、以下の点に考慮してスクリプトを書きながら行った。

- FORTRANにおけるCOMMON文で配列データの受け渡しをしているが、C++では対応する配列データのみを持つクラスをCOMMON文のブロック名に対応して作成して利用することとした。
- FORTRANでの配列の要素番号がマイナスでもプラスでも任意にとることができる。厚生労働省のプログラムでは、それを利用してプログラムの可読性をあげている。しかし、C++の配列は0からスタートするように設定されているので、配列要素番号について対応させる必要があり、いろいろな方法が考えられたが、最も簡単な配列が0からになるよう一定の数を付加する方法を採用した。
- プログラムにおいてFORTRANとC++の内容の対応が容易になるよう、変数名は一致させその行単位でのコマンドもできるだけ一致させるようにした。
- C++では配列の初期化が必ず必要となるが、変数をすべてStaticにして初期化することで対応することにした。

## 3. 使用メモリの削減と高速化

厚生労働省のプログラムを実行するには、8GバイトのメモリとIntelCPU Core2-3GHz程度を搭載したPCが必要である。これは、厚生年金/収支計算においては、大きい配列をいくつか使用しており、そのままであると実行するのに8Gバイトのメモリが必要になる。仮想メモリを使用することも考えられるが実行上の計算時間を考えれば、この程度の性能を持つPCの使用が望ましい。しかし、この仕様のPCはあまり一般的なPCとは言えないので、使用メモリの削減とその高速化を行った。その反面、先に述べたC++とFORTRANとの内容の対応を失うことになったが、それは、コメント文でできるだけ対応することにした。

- 収支計算で使用される配列で可読性のためデータ保存されたものや一時的に計算して保存するために利用されていた配列を洗い出して、使用しないようにして削減化を果たした。その結果、実メモリが4GバイトのPCでも計算ができるようになった。しかし、配列の使い回しをして削減をしている部分もある

ので、内容がその利用する場所によって意味が違ってくるので注意する必要がある。

- FORTRANで配列計算をした場合と C++で配列計算をした場合では、配列の要素の計算順序によって計算速度が違う。そこで、計算順序を変更できるものについては対応して高速を図った。しかし、プログラムの計算手順上対応するのが、難しい場合もあり、その場合には計算が遅くなっている部分がある。
- 高速化で一番顕著に効果があったのは、厚生年金/収支計算においてスライド調整率を決める際のプログラムある。この中で、調整終了年度については順序探索をし、最終年度の調整率の決定には2分法を利用していた。これを改訂して、終了年度の探索には2分法を、また最終年度における調整率の決定にはセカント法を用いることにして計算を高速化することができた。(参考：資料セカント法)

## 4. 計算精度について

このように作成したプログラムについて、コンパイラによる違いや元の厚生労働省のプログラムとの計算制度について調べてみた。つぎの表は、計算に使用したOSとプログラミング言語名である。

- Linux (Ubuntu-12.04) Gnu g++(ver 4.6.3), g95(ver 4.1.2)
- Linux (Ubuntu-12.04) Intel C++(ver 13.1.1), Fortran(ver 13.1.1)
- Max OSX-10.8 Gnu g++(ver 4.2.1), g95(ver 4.2.4)
- Max OSX-10.8 Intel C++(ver 13.0.2), g95(ver 4.2.4)
- Windows 7 Visial Studio 2012 C++

## 結果

各プログラムごと (Fortran 同士, C++同士) での差は、最終結果の

"/rev21/emp/rslt/a-rev/5500-400-XXXX-1000-1830-12-02-1105-05.csv"

において相対誤差の最大値が  $10^{-12}$ 以下となった。

なお、FORTRAN用いたものとC++を用いたものでの差は、最終結果のCSVファイルにおいて計算誤差は、3 つ要素以外では、 $10^{-10}$ 以下になった。

## 財政検証プログラムにおける資料での差

以上は一般の PC で計算した結果であるが、「財政検証プログラム」にある資料

” 財政検証バックデータ/3 試算結果/厚生年金/

厚生年金-出生 XX 位-死亡中位-経済 YY 位.xls”

と計算結果

"/rev21/emp/rslt/a-rev/5500-400-XXXX-1000-1830-12-02-1105-05.csv"

との違いを「収入合計」, 「保険料収入」, 「支出合計」, 「給付金」, 「積立金」の5つの項目に対して、2010年度から 2105年度まで個々の要素について相対誤差  $(|a_i - b_i|/b_i)$  の最大なものを調べた。

表1： 財政検証バックデータとの比較（相対精度，死亡は中位）

ファイル	収入合計	保険料収入	支出合計	給付金	積立金
出中-経中	4.2E-07	9.9E-14	1.1E-06	8.6E-07	9.7E-07
出中-経低	4.1E-07	2.9E-14	9.8E-07	8.2E-07	7.2E-07
出中-経高	4.8E-07	3.8E-14	9.7E-07	8.7E-07	1.3E-06
出高-経中	4.3E-07	9.7E-14	1.1E-06	8.6E-07	1.0E-06
出高-経低	4.2E-07	9.2E-14	1.0E-06	8.3E-07	7.5E-07
出高-経高	4.6E-07	3.3E-14	9.9E-07	8.7E-07	1.3E-06
出低-経中	3.4E-07	3.4E-14	1.0E-06	8.5E-07	7.4E-07
出低-経低	3.4E-07	3.7E-14	9.0E-07	7.8E-07	8.1E-07
出低-経高	5.8E-07	3.2E-14	9.9E-07	9.1E-07	1.5E-06

出中は出生中位，経中は経済中位を示す。

結果は，保険料収入は非常に精度良く求められているが，他の数値はその半分程度の精度を保持して求めることができた。しかし，これらの結果から，計算精度はほぼ  $10^{-6}$  程度とよい。

## 5. 厚生年金・給付費推計における計算の効率化

厚生年金・収支計算がセカント法を利用できるようになり高速化され，各計算の所要時間(Apple社 iMacにおける)は，被保険者推計0.6秒，国民保険1.4秒，基礎年金2.8秒，厚生年金・給付費推計61.2秒，厚生年金・収支計算10.2秒となった。そこで，最も時間がかかっている給付費推計のプログラムを変更して高速化を図ることにした。はじめに，FortranからC++に変更したために遅くなった可能性のある計算順の変更を試みて，つぎにプログラムの変更を行った。そしてさらに，OpenMPを利用して計算の並列化を行い，さらに効率化を高めることを実施した。

- 実行頻度の高いサブルーチンにおいて，Fortranでの  $A=A+B, A=A*B$  の形式を，Cでの  $A+=B, A*=B$  に変更した。（すべてではない）。これは，変更した方がCPUのレジスタの使い方にもよるが，配列が大きい場合に効果が期待できる。
- また，FortranとC++におけるDo文とfor文の順序変更についても，いくつかのサブルーチンにおいて行った。基本的には，プログラム上問題のないと考える限り，次元の高位のものから計算するように順序を変更した。
- 被保険者データの作成のプログラム(simbzw)は，すべての制度について計算が実行されるので，この高速化が重要である。そこで，このサブルーチンにおける計算の順序などを変更して高速化を図った。
- OpenMPの利用（並列化計算）にあたっては，実行頻度の高いサブルーチンで，並列化が容易なものに絞って並列化を試みることにした。

## 結果

高速化とOpenMPの効果を見るために，経済状況中位，出生率中位で，「厚生年金の給付費推計」のみの所要時間を計測し比較することを行った。対象とする

プログラムは、Fortran(表中:Fort)とそれをC++にしたもの(表中:C++),そしてC++を高速化したもの(表中:C++Rev), OpenMPを使用して高速化したもの(表中:OMP)である。また、それらを実行するプラットフォームとして、Apple社のiMac Late2012(3.4GHz-Intel Core i7u), BookPro Late2013(2.4GHz-IntelCore i5), BookAir Mid2012(1.7GHz-Intel Core i5)を用いた。なお、これらの計算機のCPUにおけるコア数は、4, 2, 2で、BookAirはHyper-Threadingの機能がない。表は、各計算を5回行いし、その結果の中位3つ(real timeにおける)を次の表に示す。表における real, user, sysは、MacOSやUnixマシンなどで使用されるtime命令よる所要時間(秒)を示し、realは実際の所要時間、userはCPUの使用時間、sysはシステムの使用時間である。

表2. 各プログラムによる所要時間 (秒)

	iMac(3.4GHz)			BookPro(2.4GHz)			BookAir(1.7GHz)		
	real	user	sys	real	user	sys	real	user	sys
Fort	55.965	53.011	1.904	65.046	62.956	2.045	82.578	79.177	2.952
	55.941	53.075	1.947	65.208	63.062	2.092	82.696	79.126	2.928
	56.100	52.951	1.918	65.413	63.256	2.122	82.729	79.329	2.959
C++	61.188	54.637	5.839	74.365	69.533	4.741	95.004	84.851	9.735
	61.166	54.558	5.871	73.782	69.129	4.607	93.902	83.999	9.282
	61.070	54.281	5.865	73.717	68.949	4.674	94.512	84.303	9.503
C++R	46.638	39.844	5.887	54.435	49.579	4.764	69.126	59.657	9.213
	46.384	39.641	5.909	54.408	49.492	4.854	70.212	60.542	9.403
	46.343	39.527	5.853	53.990	48.965	4.968	69.991	60.333	9.416
OMP	36.551	61.655	6.960	44.993	69.836	5.185	59.854	87.699	10.349
	36.291	61.772	6.933	45.650	70.916	5.401	59.653	87.488	10.423
	36.280	61.594	6.927	45.212	70.366	5.217	59.856	87.714	10.415

ここでOMPの場合、所要時間(real)が使用時間(user)より小さいが、使用時間はCPUにおける計算に使用したスレッドすべての所要時間であり、使用時間との差が、並列化したことによる時間短縮となっている。ただし、並列化のための時間が使用時間(user)に含まれており、それがC++Rにおける使用時間(user)との差である。これらの結果からC++を書き換えた効果があり、さらにコア数が2でもOpenMPを使用した効果を確認することができた。また、5つのプログラムの総実行時間は1分以内(iMacの場合)となり、シミュレーションの容易さが増大した。

## 3 ユーザーインターフェース

プログラムを実行して、その結果を利用して研究する際に、その環境を整備する必要がある。そこで、プログラムがC++になったこともありマイクロソフトのVisual StudioでコンパイルしWindows上で実行できるので、マイクロソフトのVBを使用してその環境

# 年金財政シミュレーションツールの開発に向けた研究報告書

を整えることにした。さらには、人口と死亡率のデータを新たに読み込めるようにして、シミュレーションを可能なようにした。

このインターフェースは、様々なパラメータを変更して実行する部分と、その最終的な結果から、必要とされる表を作成する部分の2つに分けて作成したが、その実行上は、1つの表示上で行うこととした（参照：定例会1月資料2）。

## 1. パラメータ設定ツール

パラメータを設定し実行する環境においては、5つの実行ファイルに一貫したパラメータを設定する必要があるのでこの選択を容易にして、また実行するようにした。

- インターフェースのプログラムは Windows の VBA を使用して作成した。パラメータ設定画面では、人口推計（出生率、死亡率）、経済状況 について高位、中位、低位のどれかを選択できるようにした。（図1）
- 人口データや死亡率データの変更には、新たに作成されてデータを読み込めるように入力表を作成した（入力用の右側）。
- 計算した結果を表にするものを選択するようにした。  
その表は、被保険者数・受給者の見通しでは「公的年金被保険者数（被保険者種別毎）、厚生年金の被保険者数・受給者数」、財政見通しでは「基礎年金、国民年金、厚生年金」、また「所得代替率の見通し」である。
- 計算結果をExcelファイルを出力するが、そのファイル名を入力する項目も設けた。
- プログラムの実行時には、計算時間が長くどれくらい時間がかかるか分からないため、その5つのプログラム（1. 被保険者推計、2. 国民年金、3. 厚生年金（給付費推計）、4. 基礎年金、5. 厚生年金（収支計算））のどれが実行しているかを表示させた。（図2）
- このプログラムは、出力ファイルを作成するために、収支計算後にExcelのマクロファイルと呼んでいる。

## 2. 出力ファイル

財政検証のプログラムでは最終的には、

`/rev21/emp/rslt/a-rev/5500-400-XXXX-1000-1830-12-02-1105-05.csv`

に必要なデータが出力されるが、その項目は多岐にわたっている。また、それ以外でも実行時に出力されるファイルには重要な情報も含まれている。

- 財政検証のバックデータには、各パラメータにおける計算結果のExcelファイルが納められている。そこで、ファイルから必要な項目を取り出して、財政検証レポートに掲載されている表の内で上記で選択されたものを1つの Excel ファイルの中で Sheet として作成をした。（図3）
- 最終出力ファイルと計算途中で出力ファイルから、この Excel ファイルを作成することを Excel マクロを使用して、作成することにした。このマクロファイルは、「財政シミュレーション結果表編集プログラム\_Ver6.xlsm」であり、ユーザーインターフェースから呼ばれて、Excel表を作成する。

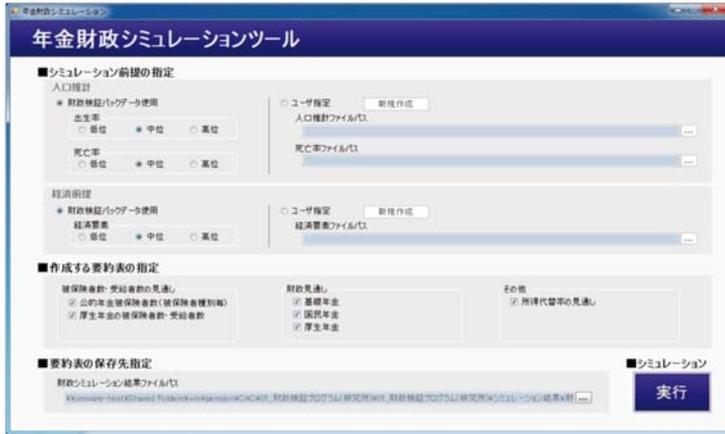


図1：ユーザーインターフェース



図2：シミュレーションの実行状況

年度	公的年金 被保険者計	第1号 被保険者				第3号被保険者			公的年金被保険者 数の減少率 ①	①に寿命の伸び等 を勘案して設定し た一定率(0.3%)を 加えた率 ②
		合計	厚生年金	共済組合	合計	厚生年金	共済組合			
2009	68.9	19.8	38.9	34.4	4.4	10.3	9.0	1.3		
2010	68.2	19.1	38.9	34.5	4.4	10.1	8.9	1.3	-0.3	
2011	67.5	18.4	39.1	34.8	4.3	10.0	8.8	1.2	-0.7	
2012	66.9	17.9	39.1	34.8	4.3	9.9	8.7	1.2	-1.0	
2013	66.3	17.6	39.0	34.7	4.2	9.8	8.6	1.1	-1.1	
2014	65.8	17.4	38.8	34.6	4.2	9.6	8.5	1.1	-1.0	
2015	65.4	17.2	38.7	34.6	4.1	9.5	8.4	1.1	-0.9	
2020	63.5	16.7	37.8	34.0	3.9	8.9	8.0	0.9	-0.6	
2025	61.5	16.3	36.9	33.2	3.7	8.3	7.5	0.9	-0.6	
2030	58.6	15.4	35.5	32.1	3.5	7.7	6.9	0.8	-0.9	
2040	49.7	12.6	30.6	27.6	2.9	6.5	5.8	0.7	-1.7	
2050	42.8	10.9	26.2	23.7	2.5	5.6	5.0	0.6	-1.5	
2060	37.1	9.5	22.8	20.6	2.2	4.8	4.3	0.5	-1.5	
2070	31.5	8.0	19.4	17.5	1.8	4.1	3.7	0.4	-1.6	
2080	27.4	7.0	16.8	15.2	1.6	3.6	3.2	0.4	-1.3	
2090	24.0	6.2	14.7	13.3	1.4	3.1	2.8	0.3	-1.3	
2100	21.0	5.4	12.9	11.7	1.2	2.8	2.5	0.3	-1.4	
2105	19.7	5.0	12.1	10.9	1.1	2.6	2.3	0.3	-1.3	

図3：出力のExcelファイル

## 4 マニュアル

作成したプログラムをコンパイル実行する上での手順を以下に示す。

### データの展開

1. pen-w1401.zipをunzip(展開)する。展開するとpensionという名のフォルダーが作成される。
2. このフォルダーを利用者の「ドキュメント」フォルダーの中に移動させる。このフォルダーの中身は、rev21¥, exec¥, シミュレーション結果¥, source¥である。rev21¥, exec¥これが「ドキュメント¥pension¥」の位置に配置されていないと、Visual Studioでのコンパイルができなくなる。また、シミュレーション結果 ¥ は出力を入れるためのフォルダーであり、位置を変更することができない。

3. 検証プログラムにおけるデータフォルダーは ¥rev21となっているが、このデータの配置位置を

`$(HomePath)¥Documents¥pension¥rev21`

に変更してある。正確には(..¥rev21)で実行ファイルのある上の階層にrev21がないと実行できない。

4. rev21¥は、入力用のデータと出力の計算結果が入る場所になっている。その構造は、検証プログラムとおおよそ同じである。それは、財政検証のプログラムと同じデータ配置であると、読み出すフォルダーの位置が違っていて実行できないので、変更してある。

5. exec¥は、Visual Studioでコンパイルされた実行ファイルが自動的に配置される。また、このフォルダーには、実行中に読み込まれるVersionファイルもおいてある。それらは国民保険計算用にnatVersion, 基礎年金計算用に basVersion, basVersion cutファイルの3つであり、変更しないようにする必要がある。

コンソールからプログラムを起動せずに、このフォルダー内の実行ファイルをダブルクリックで起動すると、プログラムが終了した時点でコンソール画面も消える。計算の途中経過も残したい場合には、コマンドプロンプトから実行する必要がある。入力パラメータ用に、input-waku.txt(被保険者推計), input-nat.txt, input-empu.txt(厚生年金の給付費推計), input-bas.txt(基礎年金), input-empa.txt(厚生年金の収支計算)を用意してある。(input-nat.txt, input-bas.txt は, " y" の文字が入っているだけである)

コマンドプロンプト実行には、始めに

```
cd ドキュメント ¥pension¥exec
```

を入力してフォルダーに移動する。その後、

```
waku.exe < input-waku.txt
```

とすれば、コマンドプロンプトから実行することができる。

## Visual Studio でのコンパイル

1. 展開してできたフォルダー(pension)の Projects の中の

waku, nat, bas, empu, empa

を

```
ドキュメント ¥Visual Studio 2012¥Projects¥
```

のフォルダーへ移動する。

なお、各フォルダーは、「財政検証プログラム」の

waku:	被保険者推計
nat:	国民保険
bas:	基礎年金
empu:	厚生年金 ¥ 給付費推計
empa:	厚生年金 ¥ 収支計算

に対応している。

2. Projectjsフォルダーに入れた, waku, nat, bas, empu, empaは各々が1つのプロジェクトになっているので, これらを順番に開いて実行する. なお, 各プロジェクトは, プラットフォームは, アクティブ(x64)で, プリプロセッサにsprint用に\_CRT\_SECURE\_NO\_WARNINGSが定義してあり, もし, 厳密にVisual Studioの書式に従う場合はsprintf変更する必要がある. また, 出力先として\$(HomePath)¥Documents¥pension¥exec¥がVisual Studio上に登録されている.
3. 起動して, ファイル > プロジェクトを開く > waku > waku.slt を起動する.
4. プロジェクトが開いたら,  
ビルド > ソリューションのリビルド  
として, ソースファイルからコンパイルをすべてし直す. なお, 使用されない変数の warning が出るが, 無視して構わない.
5. コンパイルが終了したら  
ドキュメント ¥pension¥exec¥waku.exe  
をダブルクリックして, プログラムを実行する. なお, なるべくソースを変更しないようにしているので, デバックにはいまは対応していない(実際は, rev21¥ の位置だけの問題である).
6. 1~3. を必ず waku, nat, empu, bas, empa の順に実行する.

### ユーザーインターフェースの利用

プログラムを利用する際に, コンパイルした実行ファイルを動かすときにパラメータを入力する必要がある. このパラメータは, 使用するモデルによって変更しなければならない. その利便性を考えて, ユーザーインターフェースを作成した. その利用法の概略は以下の通りであるが, 実際の提供時にはコンパイル済みの実行ファイルを添付した形で提供することも検討中である. なお, 詳細のマニュアルの原案を定例会1月資料2に示す.

1. 前述のデータの展開をまず行う.
2. execフォルダーの中にある「年金財政シミュレーション ver4.exe」を実行すれば, ユーザーインターフェースが立ち上がる.
3. 必要なパラメータ人口, 経済状況において「低位, 中位, 高位」を設定する.
4. 必要とする出力ファイルの項目を設定する.
5. 出力ファイルであるが, これはファイル名だけ変更すること.
6. 「OK」をクリックすればシミュレーションが実行される. その際に, 右側に実行中のプログラムが表示される.
7. 終了すると出力を表示するかを問い合わせるウィンドウが開くので, 「OK」をすれば, 出力がExcelで見ることができる. チェックした表はすべてそのファイルのSheetとして表示される.

## 5 年金財政の勉強と研究会での発表会

シミュレーションを開発する上でも必要であるので、年金のシステムについて知識を深めるために、研究会での時間を割いて勉強会を開いた。

- 年金制度の仕組み，マクロ経済スライドについて（江郷）  
年金システムの概略とスライド調整について解説をして頂いた（定例会6月資料2）。
- スライド調整の設定法（江郷）  
スライド調整の方法について解説をして頂いた。また，スライド調整が及ぼす世代への効果についてExcelでの計算も示しながら再度お話を伺った。（定例会11月資料1, 2, 3）
- スライド調整の計算法  
スライド調整最終年度における調整率に計算に利用されている2分法と，導入を考えているセカント法の説明を矢部委員にお願いして，その解説を受けた。なお，このセカント法を利用することによりプログラムの高速化が可能となった（資料セカント法）。
- 確率的公的年金財政モデルによる基本ポートフォリオの検討（稲垣）  
公的年金の積立金の運用に対するリスクに関する論文の紹介があった（定例会9月資料4）。
- 公的年金のリスク管理に対するポートフォリオモデル（浦谷）  
研究している公的年金をふまえたポートフォリオモデルの構築の話とそのモデルを用いたリスク管理についての解説があった（定例会12月資料2）。
- 人口推計の前提について（藤澤）  
人口推計をする際に用いられるモデルやその改良についての紹介があった（定例会12月資料3）。
- 日本保険・年金リスク学会での発表  
2013年11月9日（於：明治大学）でのアブストラクトと発表資料（資料：定例会11月資料2, 3）

## 6 今後の課題

西岡委員より，すでに評議会での報告のように今後はつぎのようなことを目指して検討を行っていく。（以下，報告書から省略抜粋）

研究所のテーマ検討委員会からは以下のような要請があり，優先順位を上げて，試算が対応できるような検討を進める。また，社会保障審議会年金部会においても，平成26年財政検証に際して，同様のオプション試算が実施されるとの方針が示されており，その試算の前提などにも配慮しつつ，本研究における試算を行っていく。

## 1. 年金支給開始年齢の変更を行った場合の試算

その際、支給開始年齢の引上げが、労働力人口の変化に及ぼす影響も研究し、雇用と整合性のある試算を行う。（別途で行われる、年金支給開始年齢の変更研究と共同研究を進める）

## 2. インフレシナリオに基づいた年金財政試算

仮に、アベノミクスの効果により2%のインフレが定着した場合に、マクロ経済スライドによる0.9%のマイナス分によって生じる高齢者の暮らしにどのような影響があるのかなどを考慮し、これに対して、何らかの調整が考えられないか。（国民に対し、マクロ経済スライド調整への合理性をアピールする）年金改定率に用いられる消費者物価指数の変動に対して、輸入物価の変動など高齢者の生活にどのように影響しているかを分析する。高齢者用の物価指数の検討を行う。

## 3. 女性の就業率の向上や外国人労働者の受け入れ拡大などの労働人口の変動シナリオに基づいた年金財政試算

女性の就業率が向上した場合や、非正規雇用の適用拡大が進んだ場合、一定所得層の外国人労働者が入ってきた場合など、様々な前提をおいた場合の労働力人口の見通しを作成し、年金制度の支え手を増やすことによる財政影響を定量的に示す。による調整について、現行の法体系のままでは、比例部分のスライド調整が終了したあと、長きにわたって基礎年金のスライド調整が行われることとなる。しかし、老後の生活基盤たる基礎年金がそのような水準になってよいのかといった議論が必要である。例えば、マクロ経済スライドによるスライド調整は、基礎年金も比例部分も同じ期間だけ調整を行うとした場合には年金財政がどのようになるのかを検証し、新たな調整方法の検討を行う。

## 4. 情報発信のための方針

平成26年度において、今年度の研究成果も含めて、本研究所から年金財政試算の結果等を情報発信していくこととなるが、その際、安易な前提をおいて、無謀な計算を行うことができないよう、本研究所としての指針・考え方を整理していく作業を行う。

## 5. その他

- 研究会の委員からの意見として出てきた、国民年金の納付率と年金財政の影響を評価し、さらに、生活保護にどのような影響がでるのかといった検証、若い年齢のときの保険料は低めに設定するとした場合の影響、感度分析としての死亡率が人口推計の仮定以上に改善した場合の影響試算、被保険者推計と脱退力との整合性を確保するための検証などは、具体的な計算手法を固めた上で、できる範囲で行っていく。
- さらに、年金財政シミュレーションからの派生で出てくる世代間の拋出と給付の関係や、マイクロシミュレーションを活用した財政試算、防貧対策としての年金制度の目的変数としてどのようなものがありうるのかの検討といった課題についても、並行して議論を行い、本研究における財政シミュレーションの幅を広げていくこととする。