

厚生労働科学研究費補助金(政策科学推進研究事業)
分担研究報告書

公的年金財政検証プログラムの Python への移植

研究分担者 川出真清 日本大学経済学部教授

研究代表者 佐藤格 国立社会保障・人口問題研究所 社会保障基礎理論研究部第1室長

研究分担者 八塩裕之 京都産業大学経済学部教授

研究分担者 金田陸幸 神戸学院大学経済学部准教授

研究要旨

目的 本研究は財政検証の計算ロジックを Python で再実装し、可読性・検証可能性・将来の分析基盤（データサイエンス環境、マイクロシミュレーション連携など）との接続を目的とする。特に、動学的マイクロシミュレーション（Dynamic Microsimulation）とのシームレスな接続を目指して、制度側からのプログラミング開発を行う。

方法 財政検証のプログラムの構造を解析するとともに、C を用いた公式な計算結果とともに、R を用いた先行研究、Python を用いた本研究の対照を行う。計算結果については、個々のセルの値を丁寧に突き合わせるマイクロ検証と計数の集計値でかつ代表的な変数となる収支変数に関するマクロ検証の両面から作業を進める。マイクロ計数からは制度的な実装の整合性を確認し、マクロ計数からは制度全体の統合状況や誤差の全体から見た誤差の発生状況を把握する。

結果 Python を用いた財政検証の再現において、年金財政の議論で重視されやすい主要な政策変数は実質的に一致している。また終了年度やスライド改定率に紐づく特定の行・列に多少の乖離はあるものの、試算の中核となる水準指標は大きくずれていない。

「総報酬」や「保険料」といった上流の計算結果は参照データと実質的に一致しており、給付推計の入力となる収入経路のロジックは高い精度で揃っていることが確認できる。「収入計」については、全体平均で約 2%、終期の 2120 年時点で最大約 6.5% の乖離にとどまっている。

長期収支の現状を見ると、最新のプロット結果では収入計の乖離率が平均 1.98%、最大 6.48%である。終期における収支差や積立金の乖離も 15%前後まで縮小しており、実用上のマクロトレンドの再現性は極めて高い。

考察 本研究における最大の成果は、財政検証を構成する六つの主要計算ブロックがすべて移植され、標準的な経済シナリオであれば Python だけで一貫してシミュレーションを実行できる段階に到達したことである。

既に記した通り、主要な政策変数は実質的に一致するような計算結果が得られている。したがって、実用上のマクロトレンドの再現性は極めて高いと言える。ただし、すべてのセルや系列が参照データと完全に一致したわけではなく、許容誤差に依存したターゲット等は技術負債として厳密に管理していく必要がある。

結論 今後の法改正や制度変更に伴うロジック改修において、プログラミング支援型 AI などを活用しやすくなり、修正の効率と安全性が飛躍的に向上するモデルの構築を行った結果、中長期的な最大の展望としての動学的マイクロシミュレーションとの連携の実現可能性が高まった。これにより、短時間労働者への適用拡大などのミクロな制度変更が、マクロな年金収支やマクロ経済スライドの調整期間に与える影響を、従来のような簡便な補正ではなく、ボトムアップで正確に算出できる画期的な道が開ける。今後はミクロモデルとマクロモデルを完全に統合した、次世代の統合年金収支計算シミュレータを実現することが中核的な研究課題となる。今後、年金制度の適用拡大の対象者など、個人レベルの多様な属性やキャリア軌跡を精緻にシミュレートしたミクロデータを、本 Python マクロモデルのインプットとして動的に接続することで、より解像度が高く機動的な「次世代統合シミュレーションモデル」の構築へとつながる研究課題に拡張してゆく必要がある。

A.研究目的

本研究は財政検証の計算ロジックを Python で再実装し、可読性・検証可能性・将来の分析基盤（データサイエンス環境、マイクロシミュレーション連携など）との接続を目的とする。特に、動学的マイクロシミュレーション（Dynamic Microsimulation）とのシームレスな接続を目指して、制度側からのプログラミング開発を行う。

また、本研究は3つの問いを目的として構成される。第一に「なぜ Python による再実装が必要か」という点である。これは、旧来の C / C++ からなるレガシー実行系の保守負荷を下げ、単一言語で一連の試算を透過的に扱えるようにするためである。

第二に「計算結果は信頼できるか」という検証の問いである。これに対しては、高橋他(2023)による R をラッパーとして介して実行した従来の公式モデルの出力（ゴールデンデータ）と直接突き合わせる手法を採り、実装漏れを特定・解消したうえで、実務上許容できる水準に達しているかを確認する。第三に「将来何ができるようになるか」という展望の問いである。Python のエコシステムを活用することで、パラメータ変更を伴う反復試算や高度な可視化、さらには将来的なミクロデータとの接続など、分析の拡張余地を確保することが狙いである。

B.研究方法

財政検証のプログラムの構造を解析するとともに、C を用いた公式な計算結果とともに、R を用いた先行研究、Python を用いた本研究の対照を行う。まず外枠を置き、厚生・国年・基礎の各勘定を順に埋め、最後に厚生全体の収支表（01shushi 等）に集約する。

計算結果については、個々のセルの値を丁寧に突き合わせるマイクロ検証と計数の集計値でかつ代表的な変数となる収支変数に関するマクロ検証の両面から作業を進める。マイクロ計数からは制度的な実装の整合性を確認し、マクロ計数からは制度全体の統合状況や誤差の全体から見た誤差の発生状況を把握するアプローチとなっている。（倫理面への配慮）
該当なし。

C. 研究結果

Python を用いた財政検証の再現において、年金財政の議論で重視されやすい主要な政策変数は実質的に一致している。また終了年度やスライド改定率に紐づく特定の行・列に多少の乖離はあるものの、試算の中核となる水準指標は大きくずれていない。

「総報酬」や「保険料」といった上流の計算結果は参照データと実質的に一致しており、給付推計の入力となる収入経路のロジックは高い精度で揃っていることが確認できる。「収入計」については、全体平均で約 2%、終期の 2120 年時点で最大約 6.5% の乖離にとどまっている。

長期収支の現状を見ると、最新のプロット結果では収入計の乖離率が平均 1.98%、最大 6.48% である。終期における収支差や積立金の乖離も 15% 前後まで縮小しており、

実用上のマクロトレンドの再現性は極めて高いと言える。

D. 考察

本研究における最大の成果は、財政検証を構成する六つの主要計算ブロックがすべて移植され、標準的な経済シナリオであれば Python だけで一貫してシミュレーションを実行できる段階に到達したことである。計算の信頼性については、マイクロレベルでの全セル比較 (compare) によって実装漏れやデータの読み込みズレを徹底的に潰しつつ、マクロレベル (01shushi_sum プロット等) で長期収支の動態を俯瞰的に確認する、という双眼的なアプローチが有効に機能している。既に記した通り、主要な政策変数は実質的に一致するような計算結果が得られている。したがって、実用上のマクロトレンドの再現性は極めて高いと言える。ただし、すべてのセルや系列が参照データと完全に一致したわけではなく、許容誤差に依存したターゲット等は技術負債として厳密に管理していく必要がある。

E. 結論

今後の法改正や制度変更に伴うロジック改修において、プログラミング支援型 AI などを活用しやすくなり、修正の効率と安全性が飛躍的に向上するモデルの構築を行った結果、中長期的な最大の展望としての動学的マイクロシミュレーションとの連携の実現可能性が高まった。これにより、短時間労働者への適用拡大などのマイクロな制度変更が、マクロな年金収支やマクロ経済スライドの調整期間に与える影響を、従来のような簡便な補正ではなく、ボトムアップ

で正確に算出できる画期的な道が開ける。
今後は、両モデル間で効率的に受け渡し可能なデータフォーマット (Parquet 等) を整備し、マイクロモデルとマクロモデルを完全に統合した、次世代の統合年金収支計算シミュレータを実現することが中核的な研究課題となる。今後、年金制度の適用拡大の対象者など、個人レベルの多様な属性やキャリア軌跡を精緻にシミュレートしたマイクロデータを、本 Python マクロモデルのインプットとして動的に接続することで、より解像度が高く機動的な「次世代統合シミュレーションモデル」の構築へとつながる研究課題に拡張してゆく必要がある。

F.健康危険情報

該当なし

G.研究発表

なし

H.知的財産権の出願・登録状況

該当なし