Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Modelo Mundial Latinoamericano: Recuperación y Modernización

Tesis de Licenciatura en Ciencias de la Computación

Alejandro Danós

Director: Rodrigo Castro
Codirector: Hugo Scolnik

Ciudad Autónoma de Buenos Aires, 2021

RESUMEN

El Modelo Mundial Latinoamericano (MML) es un modelo matemático-computacional que representa un sistema socioeconómico intersectorial a escala global. Fue desarrollado en 1972 por un grupo interdisciplinario de investigación en la Fundación Bariloche (Río Negro, Argentina). El proyecto debió abandonarse 4 años después por causa del golpe cívico-eclesiástico-militar sufrido en el país en 1976.

El MML se inserta en la disciplina de modelado global que tuvo su auge en las décadas de los 70 y 80 y en varios sentidos fue una experiencia sobresaliente. Sin embargo, la pieza fundamental del proyecto, el algoritmo del modelo, dejó de evolucionar y se hizo inaccesible.

En este trabajo nos embarcamos en la tarea de recuperar y poner operativa una versión del MML rescatada recientemente (en forma de impresión sobre formulario continuo datada en 1986) y a partir de allí aplicar estrategias de modernización.

En primer lugar digitalizamos la impresión del código con un enfoque documental y de preservación de patrimonio histórico. Luego aplicamos una variedad de técnicas de ingeniería de software obteniendo una nueva documentación sistemática en forma de diagramas de control y de flujo de datos, y una nueva versión del código reorganizado, modularizado y refactorizado. Desarrollamos también un prototipo de sitio web para experimentación con el modelo. Mediante una interfaz gráfica el usuario puede parametrizar y lanzar simulaciones de manera remota, con la posibilidad de analizar y comparar resultados entre diferentes escenarios de simulación. Finalmente, realizamos una traducción del modelo desde Fortran, un lenguaje de programación de propósito general, hacia Modelica, un lenguaje estandarizado de modelado matemático basado en ecuaciones y orientado a objetos, obteniendo una biblioteca de modelos operativos en el entorno OpenModelica.

La ambición última de este trabajo es sentar las bases de una nueva etapa en la evolución del MML, propiciando la incorporación interdisciplinaria de una gama amplia de investigadores, aportando sus conocimientos en forma de modelos matemáticos-computacionales apoyados en los últimos avances en cada disciplina.

Palabras clave: Modelo Mundial Latinoamericano, Modelos Globales, Arqueología Computacional, Digitalización, Refactorización, Ingeniería Inversa, Modelica.

RESUMEN EXTENDIDO

El Modelo Mundial Latinoamericano (MML) es un modelo matemático-computacional que representa un sistema socioeconómico intersectorial a escala global. Fue desarrollado en 1972 por un grupo interdisciplinario de investigación en la Fundación Bariloche (Río Negro, Argentina). El proyecto debió abandonarse 4 años después por causa del golpe cívico-eclesiástico-militar sufrido en el país en 1976.

El MML se inserta en la disciplina de modelado global que tuvo su auge en las décadas de los 70 y 80 y en varios sentidos fue una experiencia sobresaliente.

En el marco de la sociología de la ciencia marcó un hito demostrando la capacidad de nuestro país para producir conocimiento de manera interdisciplinaria en problemas que combinan ciencias exactas, naturales y sociales. Desde un punto de vista conceptual, representó una herramienta de posicionamiento ético-político frente a otros modelos de la época, al proponer un enfoque normativo que privilegia el desarrollo basado en la solidaridad y la equidad de acceso al bienestar. Desde un punto de vista técnico, se destacó por su planteo basado en técnicas avanzadas de optimización. Estas características se continúan evocando actualmente desde disciplinas como la historia de la ciencia, la epistemología, la filosofía y la ciencias políticas.

Sin embargo, la pieza fundamental del proyecto, el algoritmo del modelo, dejó de evolucionar y se hizo inaccesible.

Los motivos son multicausales. El tipo de modelado intersectorial perdió apoyo a nivel internacional luego de la década de los 80, con un cambio de enfoque hacia el estudio generalista de sistemas complejos y del cambio climático. A su vez, un hecho clave fue que por diversos motivos las versiones del código fuente del MML se extraviaron o se dañaron sus soportes de almacenamiento.

En este trabajo nos embarcamos en la tarea de recuperar y poner operativa una versión del MML rescatada recientemente (en forma de impresión sobre formulario continuo datada en 1986) y a partir de allí aplicar estrategias de modernización. Tomamos al algoritmo del MML como objeto central de estudio, dando una importancia particular a su interpretabilidad desde el punto de vista del sistema socioeconómico que representa.

Como primer paso digitalizamos la impresión del código, con un enfoque documental y de preservación de patrimonio histórico, para que pueda incorporarse en compilaciones historiográficas que rescatan aquella experiencia. En esta etapa privilegiamos la adherencia estricta con el código hallado. El resultado es un ejecutable capaz de correr simulaciones del MML por primera vez luego de 35 años.

Como segundo paso, aplicamos una variedad de técnicas de ingeniería de software como por ejemplo ingeniería inversa, análisis estático y dinámico de código, pruebas de regresión funcional parciales e integrales, entre otras. El resultado es doble: por un lado, obtuvimos una nueva documentación sistemática en forma de diagramas de control y de flujo de datos describiendo el funcionamiento a distintos niveles de detalle, y por otro lado, una nueva versión del código reorganizado, modularizado y refactorizado.

Desarrollamos también un prototipo de sitio web para experimentación con el modelo. Mediante una interfaz gráfica el usuario puede parametrizar y lanzar simulaciones de

manera remota, con la posibilidad de analizar y comparar resultados entre diferentes escenarios de simulación. La arquitectura fue diseñada de manera que pueda evolucionar de manera flexible junto a futuras versiones del modelo.

En tercer y último lugar avanzamos hacia una traducción del modelo desde Fortran, un lenguaje de programación de propósito general, hacia Modelica, un lenguaje estandarizado de modelado matemático basado en ecuaciones y orientado a objetos, con capacidad de expresar de manera combinada dinámicas híbridas (tiempo continuo, tiempo discreto y eventos discretos). El resultado es una biblioteca de modelos usable en cualquier herramienta compatible con Modelica, adoptando en nuestro caso el entorno de código abierto OpenModelica. Con esto buscamos acercar la experiencia de modelado computacional a la especificación matemática subyacente, alejándose de las particularidades de un lenguaje de programación particular.

La ambición última de este trabajo es sentar las bases de una nueva etapa en la evolución del MML, propiciando la incorporación interdisciplinaria de una gama amplia de investigadores, aportando sus conocimientos en forma de modelos matemáticos-computacionales apoyados en los últimos avances en cada disciplina.

Palabras clave: Modelo Mundial Latinoamericano, Modelos Globales, Arqueología Computacional, Digitalización, Refactorización, Ingeniería Inversa, Modelica.

Índice general

1. INTRODUCCIÓN

Los modelos globales son modelos matemático-computacionales utilizados para simular interacciones entre distintos aspectos de una sociedad [CJ15], como por ejemplo demografía, economía, alimentación, contaminación, educación, etc. El Modelo Mundial Latinoamericano (MML) es un modelo global desarrollado durante la primera mitad de la década del 70 en San Carlos de Bariloche, Argentina, bajo la gestión de la Fundación Bariloche [Assa].

El Proyecto MML fue muy innovador al momento de su desarrollo, con perspectivas de ser utilizado para muchos contextos diferentes de economías en el mundo. Sin embargo, luego del golpe de estado de 1976 varios investigadores de la Fundación Bariloche, incluyendo desarrolladores del MML, debieron exiliarse. El modelo sería luego aplicado por ejemplo en países como Brasil y en Egipto, y su uso mermaría con el tiempo.

A principios de los 2000 hubo una revitalización desde el punto de vista conceptual, con la reedición del libro original que acompañó al modelo, con motivo de su 30º aniversario.

Esta Tesis representa una segunda inyección de esfuerzos hacia la recuperación histórica, puesta en valor e inicio de modernización del MML.

El paso de las décadas sin utilizar el modelo influyeron en que se extravíen las versiones del código en tarjetas perforadas o que se desmagneticen las cintas magnéticas de *backup*. Esto produjo que, hasta este trabajo, no exista (hasta donde llega nuestro conocimiento) una versión del MML que pueda ser efectivamente utilizada, mejorada y/o aplicada por académicos o tomadores de decisiones en políticas públicas.

En el marco de una iniciativa para reconstruir el MML y reconsiderar su legado y enfoques particulares, es que se encuentra una copia impresa del código original (escrito en lenguaje Fortran 77 [Com15]) consistente en aproximadamente 70 hojas de formulario continuo impreso con matriz de puntos.

En base a esta versión se llevó a cabo la recuperación operativa que es uno de los motivos de esta Tesis. El trabajo incluye además una traducción a un lenguaje moderno de modelado basado en ecuaciones llamado Modelica [MLS17] y también el desarrollo de una interfaz web interactiva que ofrece la posibilidad de experimentar con el modelo vía Internet, configurando simulaciones y analizando sus resultados.

En este reporte introducimos al MML y exponemos los procesos de recuperación y traducción que incluyen la producción de nueva documentación sistemática y rigurosa sobre el funcionamiento del modelo. También presentamos brevemente el lenguaje Modelica y el funcionamiento del sitio web del MML.

1.1. El Modelo Mundial Latinoamericano

1.1.1. Contexto histórico

Probablemente el modelo global que llegó a ser más difundido sea World3 [MMRBI72], publicado a principios de los 70' por investigadores del grupo de System Dynamics de la Sloan School of Management dependiente del Massachusetts Institute of Technology (MIT). World3 fue utilizado, entre otras cosas, para dar un pretendido sustento científico a la predicción de una "catástrofe inminente", a ocurrir alrededor del año 2050, bajo

supuestos de "business as usual" (es decir, que todo continue igual). Dicha catástrofe sería causada por la acción combinada del crecimiento exponencial de la población y del nivel de consumo per cápita, resultando en una presión insostenible sobre los recursos naturales del planeta. Luego, los autores concluyen que para salvar al planeta debían implementarse políticas de control de crecimiento económico en los países ricos y de control poblacional en los países pobres.

El MML [HSC+76] surge en respuesta al modelo World3 con el objetivo de discutir varias de sus hipótesis y conclusiones. Los autores del MML discreparon con la suposición que los límites físicos sean los que predominen sobre el crecimiento poblacional. Consideraban que las políticas propuestas por World3 promovían el statu quo de desigualdad entre "hemisferio norte vs. hemisferio sur" y que la "catástrofe" en realidad ya estaba ocurriendo en varias regiones del mundo en las cuales las condiciones de vida eran todo menos dignas.

Desarrollado en la primera mitad de los años 70 en San Carlos de Bariloche, Argentina, el proyecto del MML estuvo compuesto por profesionales de varias disciplinas: matemática, demografía, economía, ecología, educación, salud, vivienda, urbanización y alimentación. Su aspecto interdisciplinario fue revolucionario para la época en la región y marcó el camino para proyectos futuros de similares características. La noción de *necesidades básicas* y la decisión de usar a la *Esperanza de Vida al Nacer* como variable principal a maximizar fueron influyentes en el estudio de la demografía en décadas siguientes.

El golpe de estado de 1976 obligó a varios investigadores de la Fundación Bariloche a interrumpir su trabajo, muchos optando por exiliarse. El MML, o parte de él, siguió siendo utilizado por ejemplo para aplicación de políticas públicas en Brasil [RLS79] y en Egipto.

En 2004 se publicó una nueva edición del libro [HSC+04] en conmemoración de su 30º aniversario. Sin embargo, el uso práctico del MML se discontinuó, principalmente por la imposibilidad de leer cintas magnéticas de la tecnología de la época.

1.1.2. Características principales del Modelo

Las principales fuentes de documentación que describen el concepto y funcionamiento del modelo son:

- El "Capítulo 4:El Modelo Matemático" del libro original [HSC+76] (y de su reedición en 2004 [HSC+04])
- El Handbook of the Latin American World Model [SFL+77] que presenta un enfoque más cercano a la implementación matemático-computacional.

En base a esta documentación previa y a nuestro propio estudio detallado del código, recopilamos la información que exponemos en esta sección, la cual cubre conceptos estructurales necesarios para comprender el desarrollo de esta tesis. Sin embargo, no pretende dar una explicación exhaustiva ni dar cuenta de un modelo matemático exhaustivo.

1.1.2.1. Los sectores económicos

La estructura del modelo queda definida esencialmente por un sistema económico a tiempo discreto que simula año a año un sistema productivo compuesto por cinco sectores económicos, descriptos de la siguiente manera en el Capítulo 4 de [HSC+76, HSC+04]:

Los sectores económicos mas importantes definidos en el modelo son aquellos con relación directa con el concepto de "necesidades básicas". Los sectores que se modelan son: **1) alimentación, 2) vivienda, 3) educación, 4) otros servicios y bienes de consumo y 5) bienes de Capital**.

Los sectores 1, 2 y 3 producen los bienes necesarios para la satisfacción de las necesidades básicas, el sector 5 (los bienes para el consumo futuro) y el sector 4 (todas las actividades económicas restantes).

Los sectores están verticalmente integrados, y sus limites definidos de manera tal que se eliminan las transacciones intermedias.

Tres de los cinco tipos de bienes producidos se pueden definir específicamente: alimentación, como calorías y proteínas; vivienda, como viviendas disponibles; y educación, como plazas en el sistema escolar básico (el que cubre los primeros doce años de educación formal).

"Otros servicios y bienes de consumo" y "bienes de Capital" no pueden ser especificados de esta manera, porque engloban un gran espectro de productos. Vestimenta, muebles y útiles del hogar, cuidado de la salud, transporte, comunicaciones, entretenimientos, servicios públicos y administrativos, y todas las actividades educacionales no contenidas en el sector 3, se incluyen en el sector 4, en tanto que el 5 abarca construcción de viviendas y planificación de infraestructura de las ciudades, edificios públicos, infraestructura de transporte, comunicaciones, y otros servicios básicos, maquinas y vehículos, etc.

Para todos los sectores se hace un seguimiento explícito del Capital, Mano de Obra y porción del producto sectorial bruto, referenciado como GNP en el código por sus siglas en inglés (Gross National Product). Esta información puede verse condensada en la Figura 1.1.

#	Sector	Producción anual	Capital	Mano de Obra	GNP
1	Alimentación	Proteínas y calorías	✓	✓	✓
2	Viviendas	Viviendas construidas	✓	✓	✓
3	Educación	Plazas en el sistema escolar básico	✓	✓	✓
4	Otros Bienes y Servicios	✗	✓	✓	✓
5	Bienes de Capital	✗	✓	✓	✓

Figura 1.1: Resumen de los 5 sectores económicos incluidos en el modelo.

1.1.2.2. Bloques regionales

El modelo divide al mundo en 4 bloques geográficos/económicos: países desarrollados, América Latina y el Caribe, África y, por último, Asia y Oceanía. En la Figura 1.2 se muestra la distribución de países entre bloques. Notar que algunos países, como Australia, Nueva Zelanda y Japón, pertenecen al bloque de países desarrollados y no al de Asia y Oceanía. Además, notar que incluye países no existentes en la actualidad, otros que sí existen pero cambiaron de nombre y también países faltantes que nacieron después de la publicación del modelo.

Países desarrollados	América Latina y el Caribe	África	Asia
1. Alemania Democrática	1. Argentina	1. Alto Volta	1. Afganistán
2. Alemania Federal	2. Bolivia	2. Angola	2. Arabia Saudita
3. Australia	3. Brasil	3. Argelia	3. Bangladesh
4. Austria	4. Chile	4. Burundi	4. Birmania
5. Bulgaria	5. Colombia	5. Camerún	5. Camboya
6. Bélgica	6. Costa Rica	6. Chad	6. China
7. Canadá	7. Cuba	7. Congo	7. Corea del Norte
8. Checoslovaquia	8. Ecuador	8. Costa de Marfil	8. Corea del Sur
9. Dinamarca	9. El Salvador	9. Dahomey	9. Filipinas
10. España	10. Guatemala	10. Egpito	10. Hong Kong
11. EEUU	11. Haití	11. Etiopía	11. India
12. Finlandia	12. Honduras	12. Ghana	12. Indonesia
13. Francia	13. Jamaica	13. Guinea	13. Irak
14. Grecia	14. México	14. Kenia	14. Irán
15. Hungría	15. Nicaragua	15. Liberia	15. Jordania
16. Irlanda	16. Panamá	16. Libia	16. Laos
17. Israel	17. Paraguay	17. Madagascar	17. Malasia
18. Italia	18. Perú	18. Malawi	18. Mongolia
19. Japón	19. República Dominicana	19. Mali	19. Nepal
20. Líbano	20. Trinidad y Tobago	20. Marruecos	20. Nueva Guinea
21. Noruega	21. Uruguay	21. Mauritania	21. Pakistán
22. Nueva Zelanda	22. Venezuela	22. Mozambique	22. Siria
23. Países Bajos		23. Niger	23. Singapur
24. Polonia		24. Nigeria	24. Sri Lanka
25. Portugal		25. RCA	25. Tailanda
26. Puerto Rico		26. Rhodesia	26. Taiwan
27. Reino Unido		27. Rwanda	27. Turquía
28. Rumania		28. Senegal	28. Vietnam del Norte
29. Suecia		29. Sierra Leona	29. Vietnam del Sur
30. Suiza		30. Somalía	30. Yemen
31. URSS		31. Sudáfrica	31. Yemen Democrático
32. Yugoslavia		32. Sudán	
		33. Tanzania	
		34. Togo	
		35. Túnez	
		36. Uganda	
		37. Zaire	
		38. Zambia	

Figura 1.2: Distribución de los países entre los 4 bloques.

1.1.2.3. Fases de simulación

Las fases son intervalos de tiempo de la simulación durante los cuales el modelo opera de modos claramente distinguibles de otros. Se las puede dividir entre **fases explícitas** y **fases implícitas**. Las explícitas son las mencionadas en las distintas fuentes de documentación disponibles y también se puede configurar su comienzo y fin mediante el ajuste de parámetros. Las fases implícitas no son mencionadas explícitamente en la documentación pero son muy relevantes ya que en ellas el modelo cambia su comportamiento considerablemente.

En los párrafos siguientes presentaremos información detallada de cada una de las

fases, cuya presentación condensada puede verse en las Figuras 1.3 y 1.4.

Fase	Tipo	Comienzo		Fin	
		Valor Estándar	¿Configurable?	Valor Estándar	**¿Configurable?**
Estimación 1960	Implícita	Pre-1960	✗	Instantánea	✗
Regresión	Implícita	1960	✗	1970	✗
Proyección	Explícita	1970	✗	1980	✓
Optimización	Explícita	1980	✓	Fin de la simulación	✗

Figura 1.3: Fases por las que transcurre el modelo durante la simulación.

Al comienzo de la simulación, se deben inicializar ciertas variables de los sectores, la demografía, costos, etc. Esta inicialización se hace en base a configuraciones de parámetros provistos por el usuario y por un conjunto de valores previamente almacenados[1].

Sin embargo, en la época no había suficiente información para inicializar todas las variables incluidas en el modelo y por eso algunas debieron ser estimadas en función de las otras. Esta estimación se hace para el año 1960, correspondiente al inicio de la simulación, y sus ecuaciones son distintas a las de la fase de proyección, que comenzaría oficialmente también en ese año. Es por esto que decimos que es una **fase implícita**, distinta a proyección, aunque dure sólo un instante de tiempo antes de 1960.

En el código se hace referencia implícita al intervalo entre 1960 y 1970, que es cuando se interpolan año a año los parámetros *alfa* de las funciones de producción de tipo Cobb-Douglas [CD28] utilizadas para calcular la producción sectorial anual (explicado en más detalle en la próxima sección). Este intervalo no es mencionado explícitamente en el código ni en la documentación original y tampoco es configurable, dado que son años que están fijos en el código. Sin embargo, su existencia es clara y por eso decimos que es una **fase implícita** y la llamamos fase de **regresión**.

La fase de **proyección** comienza en 1960, el primer año de la simulación, y termina en el año indicado por el párametro KPROJ, cuyo valor por defecto es 1980. Sin embargo, como mencionamos en el párrafo anterior, los 10 primeros años de esta fase corresponden a la fase de **regresión**, por lo que en este documento asumimos que en realidad comienza en el año 1970. El único comportamiento que diferencia a ambas fases es el cálculo de los alfa de Cobb-Douglas [CD28], que son interpolados en **regresión** y no son modificados en **proyección**, dando la posibilidad de pensar que alternativamente la primera es una *sub-fase* de la segunda.

Durante la **proyección**, y hasta el fin de la simulación, los parámetros alfa se mantienen constantes con los valores de 1970. Sin embargo, durante esta fase varias ecuaciones son proyecciones en base a valores de años anteriores, dándole el nombre a la fase. Se la menciona tanto en el código como en la documentación y por eso decimos que es una **fase explícita**.

La última fase es la de **optimización**. Comienza en un año definido por un parámetro (KPROJ) y continúa hasta el fin de la simulación. Varias de las ecuaciones que definen el comportamiento de los sectores durante esta fase son versiones más complejas que las ecuaciones homólogas durante la fase previa de proyección. Por ejemplo, a los cálculos de los subtotales sectoriales de Capital y Mano de Obra, que en fase de **proyección** se calculan en base a las proyecciones de sus respectivas proporciones, durante la fase

[1] Estos valores están declarados en estructuras DATA en la versión original en FORTRAN77, que son sentencias no ejecutables usadas como almacenamiento de datos.

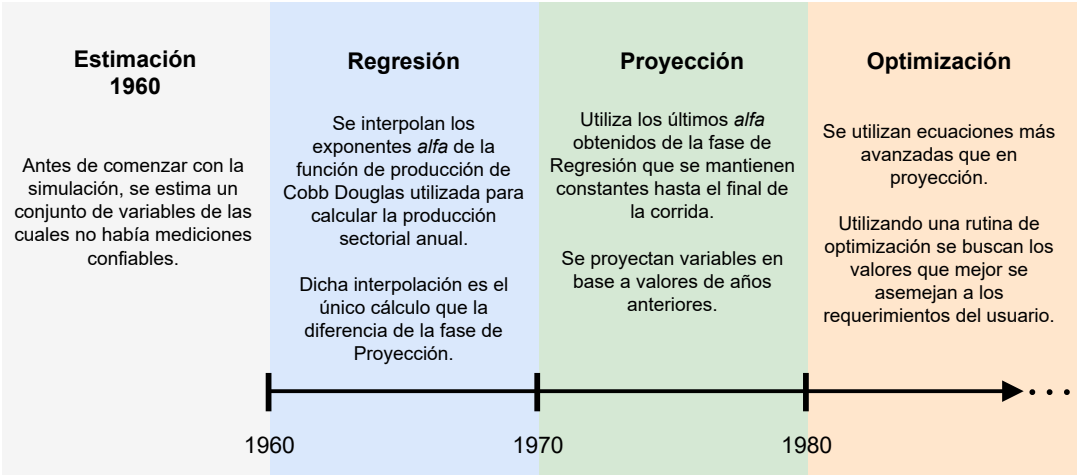


Figura 1.4: Línea de tiempo condensando la información de las fases del modelo.

de **optimización** se le asignan valores que maximicen la Esperanza de Vida al Nacer, y que son obtenidos optimizando una *función objetivo*. Otra diferencia es que durante esta fase se lleva a cabo la **ayuda económica** entre regiones para facilitar el desarrollo productivo, que consiste en distribuir un porcentaje bajo del Capital, modificable por el usuario, de los Países Desarrollados hacia de África y Asia. La fase de optimización se menciona explícitamente en la documentación original por lo que la denominamos una **fase explícita**.

1.1.2.4. La Función de Producción

En base a la Mano de Obra y Capital de cada uno de los sectores económicos se calcula el respectivo producto nacional bruto sectorial mediante el uso de una función de producción tipo Cobb-Douglas [CD28]. Esta función representa la relación tecnológica entre ambos factores de producción, que es ajustable por el parámetro alfa, y que en el MML se usa uno para cada sector económico. La forma usada es la siguiente:

$$Y_i = K_i^{\alpha_i} * L_i^{(1-\alpha_i)}$$

siendo i el i-ésimo sector, K el Capital, L la Mano de Obra, Y el producto nacional bruto y $0 < \alpha < 1$ el parámetro alfa.

En la fecha de desarrollo se contaba, para cada sector, solamente con los alfa para los años 1960 y 1970. Durante los años intermedios, pertenecientes a la fase de regresión, se hace una interpolación lineal y para los años siguientes se hace una extrapolación constante en base a los alfa de 1970. Esta extrapolación abarca el final de la fase de proyección y la totalidad de la fase de optimización hasta el fin de la simulación.

1.1.2.5. Ayuda económica

La simulación puede ser configurada para que haya ayuda económica desde el bloque de países desarrollados hacia África y Asia. Concretamente, durante la fase de optimización se le resta un porcentaje al producto bruto interno de los países desarrollados que es compartido, bajo un criterio en particular, entre África y Asia en forma de Capital. La

decisión de cuánto tomar de países desarrollados se define mediante unas instrucciones que tienen en cuenta que la ayuda debe ir creciendo gradualmente, que tiene que comenzar con un porcentaje dado como parámetro, que se tienen que cumplir ciertas condiciones para que haya ayuda y finalmente un porcentaje máximo de ayuda posible. Para facilitar la lectura no explicamos en detalle este procedimiento, que puede leerse en el código de la sección de cálculo de producto bruto interno GNP.

En cambio, la forma de repartir esta ayuda es simple y se puede ver en la siguiente ecuación:

$$K_i = \frac{P_i}{EV_i} * \frac{1}{(P_3/EV_3 + P_4/EV_4)} \tag{1.1}$$

En la ecuación (1.1), K_i refiere a la ayuda económica a asignar como Capital, i es el bloque de África o Asia, según corresponda, a P_3 y EV_3 la población y Esperanza de Vida al Nacer de África y P_4 y EV_4 lo propio de Asia. De esta forma, mientras mayor sea la Esperanza de Vida al Nacer de un bloque menor será el porcentaje de ayuda que le corresponda y en el caso de la población ocurre lo inverso.

1.1.2.6. Formulación matemática compacta como un problema de optimización

Un problema de optimización de una función f puede pensarse como

$$min(f(\vec{x})) \tag{1.2}$$

teniendo a $\vec{x}$ un punto inicial, X el conjunto de vectores de valores posibles que pueden tomar los parámetros y $\vec{x} \in X$.

En el caso del MML, el vector $\vec{x}$ representa los valores de Mano de Obra y Capital de los primeros 4 sectores

$$\vec{x} = (L_1, ..., L_4, K_1, ..., K_4) \tag{1.3}$$

con L_i y K_i la Mano de Obra y el Capital del sector i, respectivamente.

Partiendo de L_i y K_i se calcula el GNP_i de cada sector mediante una función de producción tipo Cobb-Douglas (ver Sección 1.1.2.4).

Para los objetivos se plantean 26 restricciones. Por ejemplo, la restricción CONS(1) busca que el Capital asignado a la inversión crezca, CONS(2) busca que el GNP asignado no decrezca más de 2 %, CONS(3) busca que el GNP asignado a consumo no caiga por debajo del 45 %, etc. En particular, la restricción CONS(26) busca que la Esperanza de Vida al Nacer crezca, siendo la más importante desde un punto de vista conceptual.

Para cada una de estas restricciones se proveen pesos por defecto para definir prioridades relativas entre ellas, que pueden ser modificados por el usuario. Cuando la optimización no logra satisfacer todas las restricciones en un año dado, elegirá cuales sacrificar utilizando el orden de prioridades. La restricción CONS(26) por defecto tiene la mayor prioridad por lo que en la ejecución estándar siempre se priorizará el crecimiento año a año de la Esperanza de Vida al Nacer. La definición de cada una de las prioridades, con sus respectivos pesos, se realiza en el módulo `CONTROL_VARIABLES`.

Luego, el modelo busca resolver un problema de optimización con restricciones de la función LELAPA, recurriendo a la función ZXPOWL[2], una implementación del método de minimización sin derivadas de Powell [CP77].

[2] Ambas funciones son introducidas en la Sección 1.1.3.1 y un flujo de control de LELAPA puede verse en la Sección 3.3.3.

El objetivo de la optimización de LELAPA es buscar valores para el Capital y la Mano de Obra que minimicen la expresión 1.4 y utilizarlos (en la misma función) para calcular los valores de las principales variables físicas de los sectores económicos modelados (e.g., alimentación, viviendas, educación).

En forma sintética se puede plantear que el modelo busca resolver, para cada año, el siguiente problema:

$$min \left\{ -(1 + QLIFE) * ALE - \sum_{i=1}^{26} WG(i) * min(0, CONS(i)) \right\} \tag{1.4}$$

en donde:

- QLIFE: 0 (si las necesidades básicas no fueron satisfechas) o una proporción del GNP asignado al Sector 4 "otros servicios y bienes de consumo" (en cualquier otro caso)
- ALE: Esperanza de Vida al Nacer
- WG(i): peso asignado a la restricción i. De no ser especificado por el usuario, se utiliza su respectivo valor por defecto.
- CONS(i): el valor de de la restricción i en la invocación actual de LELAPA

Entre estas variables, las únicas que no varían entre sucesivas invocaciones a LELAPA son los pesos $WG(i)$ definidos por el usuario antes de comenzar la simulación. Las otras son calculadas internamente en LELAPA y dependen, directa o indirectamente, del Capital y Mano de Obra recibidos como entradas en cada llamada a la función.

Con respecto a las necesidades básicas, en el modelo se asume que fueron satisfechas si se cumplen los 3 requisitos siguientes:(1) el porcentaje de matriculación llegó al máximo de 98 %, (2) las calorías por día por persona llegaron al máximo de 3000 y (3) hay por lo menos una vivienda por familia (en promedio).

Finalmente, una explicación más detallada del enfoque mediante optimización del Modelo Bariloche, resumido en esta sección, puede encontrarse en [SFL+77] (Secc. "Considerations on Mathematical Models and The Dynamics of The Bariloche Model", pag. 119) y la revisión metodológica por William Nordhaus en [Nor75].

1.1.2.7. Componentes principales y mecánica del modelo

En la Figura 1.5 presentamos una versión simplificada de las relaciones entre los distintos componentes del modelo con respecto al flujo de datos. Más adelante, en la Sección 3.3 veremos una versión que incluye las variables de dichas relaciones, además de diagramas internos de cada componente. Para facilitar su interpretación, dividimos los módulos entre los siguientes grupos:

- **Cálculo previo a Sectores:** Incluye los módulos con operaciones de las cuales dependen los sectores económicos.
- **Sectores (económicos):** Incluye cálculo de producción por sector. No incluye los cálculos de los sectores `Otros Bienes y Servicios` y `Bienes de Capital` porque en la implementación original no son calculados directamente, sino que son tenidos en cuenta sólo en las variables sectoriales de Capital, GNP y Mano de Obra.

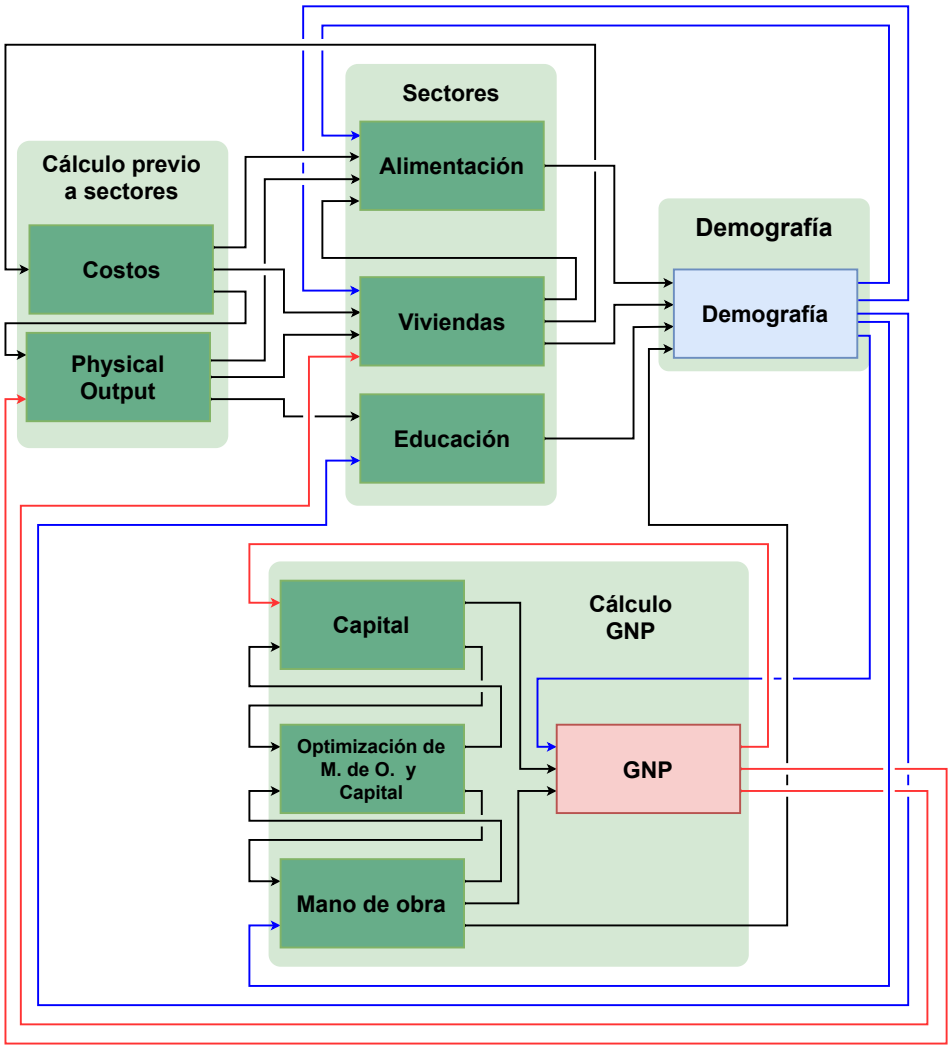


Figura 1.5: Flujo de datos de alto nivel simplificado

- **Demografía:** incluye un único módulo del mismo nombre que abarca el cálculo de poblaciones por rango etario, natalidad, mortalidad y Esperanza de Vida al Nacer, entre otras.
- **Cálculo de GNP:** Incluye los módulos necesarios para calcular año a año el GNP en base al Capital y la Mano de Obra.

Dicho diagrama no tiene en cuenta el paso del tiempo, por lo que puede ser un poco confuso cómo es que se lleva a cabo el pasaje del año N al N+1. Para ello podemos referirnos a la Figura 1.6, dedicada a describir a grandes rasgos el modelo matemático del MML. En ella, se resume qué variables son calculadas al comienzo del año y el proceso para calcular las del año siguiente. Este segundo diagrama está basado en la Figura 20 del "Capítulo 5: Modelo Matemático"del libro [HSC$^+$76, HSC$^+$04].

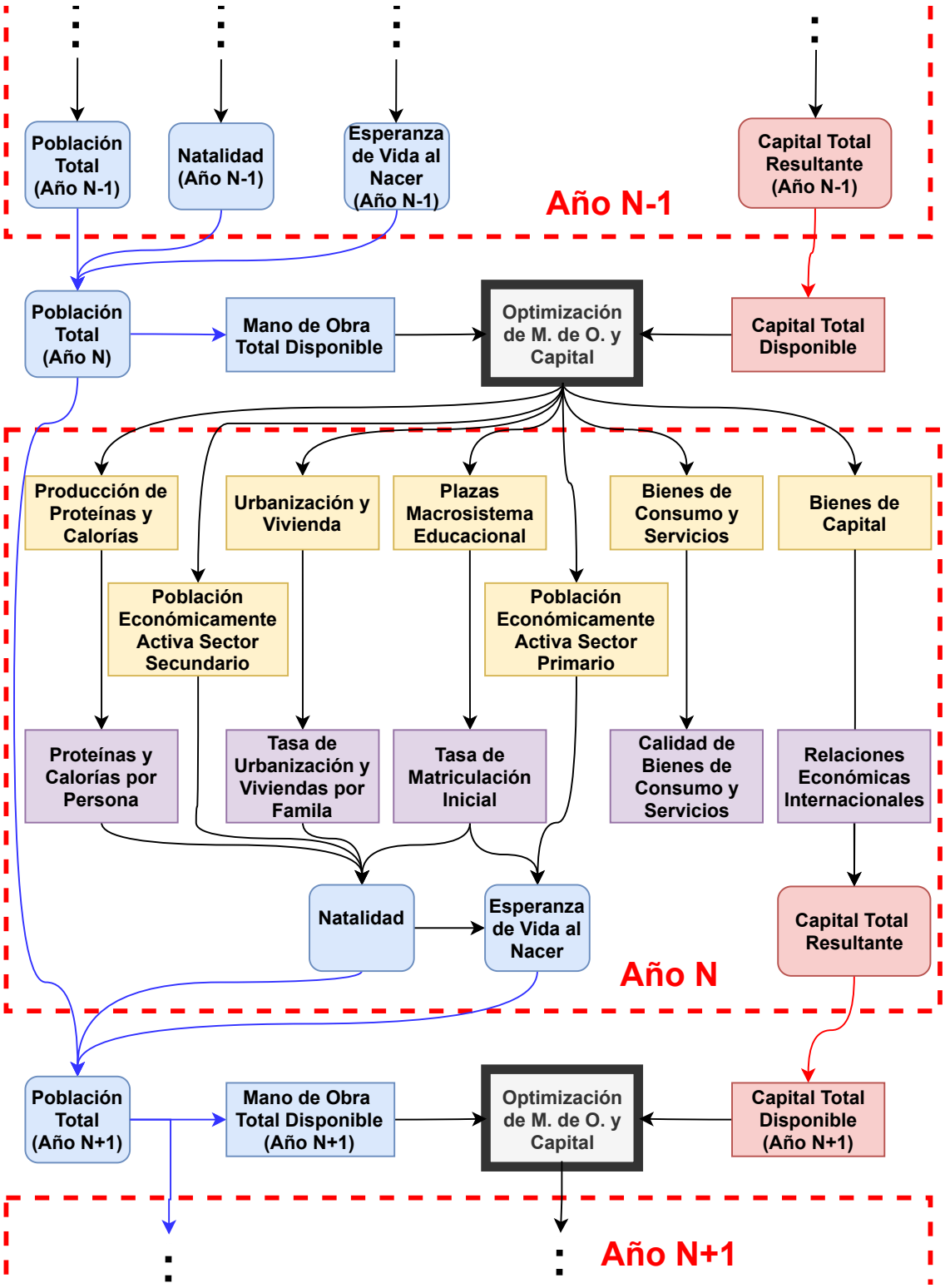


Figura 1.6: Modelo matemático mostrando el paso de tiempo N a N+1

1.1.3. Comentarios sobre la Implementación Técnica

El modelo fue implementado en FORTRAN 77 [Com15], un lenguaje de programación imperativo de propósito general utilizado principalmente en ámbitos científicos que requieren de cómputo numérico muy intensivo. En la versión que recuperamos en este trabajo, al iniciar el programa el usuario es recibido por un módulo *Monitor* que ofrece la posibilidad de configurar los parámetros de la corrida. Luego de esto, el *Monitor* comienza la simulación sin interrupciones, finalizando el hilo de ejecución con la escritura de un archivo de salida con los resultados. En él, primero se escriben los comentarios del usuario ingresados al *Monitor* y los parámetros modificados. Luego, por cada región y por cada categoría de variables (por ejemplo Demografía, Sectores Económicos, Capital, etc.) se presentan los resultados en tablas de valores y en gráficos que comparan variables dibujándolas usando caracteres ASCII para representar curvas. Un ejemplo de estas salidas puede verse mas adelante en las Figuras 2.1, 2.2 y 2.3.

1.1.3.1. Funciones

Con respecto al código, la Figura 1.7 resume todas las funciones y subrutinas presentes. Las más importantes son la subrutina MODEL, encargada de *orquestar* la simulación de principio a fin, y LELAPA, que incluye todas las instrucciones en la secuencia que va desde la Mano de Obra y el Capital de los primeros 4 sectores hasta el cálculo de la Esperanza de Vida al Nacer. Ambas funciones incluyen cálculos de variados contextos como de demografía, alimentación, educación, etc. Durante el flujo de ejecución normal, una vez que el *Monitor* finaliza con la configuración de los parámetros, se llama a la rutina MODEL y ésta invoca a todas las otras funciones de manera iterativa año a año, incluida LELAPA. En la Sección 3.3 estudiaremos y documentaremos en detalle el flujo de datos y el flujo de control del modelo.

Nombre	Tipo	Descripción
DICTIO	Subrutina	Descripción de las variables y funciones del modelo
ESCPLT	Subrutina	Escalado de gráficos
EXPV	Función	Fórmula de cálculo de esperanza de vida
FECUN	Subrutina	Tasas de fertilidad de mujeres
FNPF	Función	Fórmula de cálculo de personas por familia
FOODKP	Subrutina	Cálculo de calorías y proteínas provenientes de agricultura, pesca y ganadería
HIST	Subrutina	Generación de gráficos
LELAPA	Subrutina	Cómputos partiendo de M. de O. y Capital hasta el cálculo de E.V.
MML	Programa	Iniciar el programa, iniciar Monitor y llamar a Model
MODEL	Subrutina	Inicializar variables, orquestar simulación y llamar a escritor de resultados
MORFPA	Subrutina	Similar LELAPA partiendo de tierra cultivable hasta la diferencia de necesidades calóricas entre fases
PLTS	Subrutina	Auxiliar de graficación
POBLAC	Subrutina	Cálculos poblacionales
PRLPT	Subrutina	Escritura de resultados (incluyendo gráficos)
RNAT	Función	Fórmula de cálculo de tasa de natalidad
SOBREV	Subrutina	Tasas de supervivencia por edad y por sexo
ZXPOWL	Subrutina	Implementacón de minimización de Powell sin derivadas usado para optimizar MORFPA y LELAPA

Figura 1.7: Funciones y subrutinas de la implementación en FORTRAN77

1.1.3.2. Regiones y Sectores

Los 4 Bloques regionales y los 5 Sectores económicos son representados como *dimensiones* en las variables, como en los ejemplos a continuación. Al momento de acceder a los datos, se tiende a nombrar IB al iterador de Bloques regionales e IS al iterador de Sectores. Algunos ejemplos de esto son los siguientes:

- **Por región**. Las variables `POP(IB)`, `GNP(IB)` y `EXLIFE(IB)` corresponden a la población, producto bruto interno y Esperanza de Vida al Nacer del bloque `IB`, con `IB` $\in \{1, 2, 3, 4\}$.

- **Por sector**. Las variables `GNPXSX(IS)`, `COSTX(IS)` y `CLFX(IS)` corresponden al subtotal de producto bruto interno, costo de producción y promedio de salarios por sector con `IS` $\in \{1, 2, 3, 4, 5\}$.

- **Ambas**. Ídem para las variables `GNXS(IS,IB)`, `COST(IS,IB)` y `COSTLF(IS,IB)` con la misma interpretación para `IS` e `IB`.

1.1.3.3. Variables

El modelo incluye más de 130 variables y, por las limitaciones del Fortran de la época, sus nombres son de 6 caracteres o menos. Es por esto que los autores decidieron agrupar varias de ellas en categorías con reglas asociadas. La Figura 1.8 muestra las reglas elegidas en las cuales el asterisco representa el resto de la palabra. Por ejemplo, la primera regla abarca los nombres de variables que comienzan con K. Sin embargo, algunas reglas tienen excepciones y pueden confundir si no se analizan individualmente. Por ejemplo, la `X` del final de la variable `POAGSX` proviene de la abreviación *population* (PO), *age* (AG) y *sex* (SX); y no porque siga la regla `*X` de la tabla.

Otro aspecto a tener en cuenta es que el límite de 6 caracteres afecta directamente a algunas variables al ser aplicadas algunas reglas de la tabla. Por ejemplo, al aplicar la regla de `*X` sobre la variable `COSTLF(IB)` se formaría `COSTLFX` de 7 caracteres, uno más de lo permitido. En este caso, los creadores del modelo decidieron en cambio ponerle de nombre `CLFX`.

Finalmente, hay casos particulares donde el uso de la `X` puede confundir pero en realidad no rompe con ninguna regla, como es el caso de `GNPXS(IB)` y `GNPXSX(IS)`. Esta última combina dos reglas, la de `*XS`, por sector, y `*X`, versión no dependiente del bloque.

Con respecto a variables de uso general, hay dos variables cuyas interpretaciones dependen del contexto. Una es `X`, que cumple el rol tradicional de su par x usada en cálculo matemático como argumento de funciones. En el caso del MML, se usa como entrada en `MORFPA`, donde la primera posición representa la tierra cultivable, y en `LELAPA`, donde las 8 posiciones representan a los subtotales de Capital y Mano de Obra de los primeros 4 sectores. Con respecto a `W`, ésta cumple el rol de variable temporal y es usada en cálculos auxiliares del GNP, de población, estimaciones de 1960, etc. Por estos motivos, parte de los aportes de esta tesis es proveer una reingeniería parcial del modelo con un sistema de nombres mejorado, lo cual funciona también como paso previo a una traducción a un lenguaje de modelado matemático moderno como Modelica (ver Seccion 6).

Regla	Ejemplos	Excepciones	Descripción
K∗	KSTART, KSTOP, KPROJ	Variables de FOODKP	Variables/parámetros relacionados con años de simulación
∗D	CAPD, GNPD, RLFD	AID, CD, DECAID, REND ...	Proporciones (distribuciones) por sector (Capital-D, GNP-D, Labor Force-D)
∗XS	GNPXS, CAPXS	-	Subtotales por sector (GNP x Sector, Capital x Sector)
∗X	POPX, ENROLX, FERTX	CALMX, SPAMAX, POAGSX, ...	Versiones *no dependientes del bloque* (POP(IB), ENROL(IB), FERT(IB))
∗Y	POPY, ENROLY, FERTY	-	Similar a *X pero tiende a ser usado con variables del año anterior
∗2	COST2, GNP2	ALFA2	Variables del sector 2, viviendas
∗34	FR34, NQH34	-	Variables de los bloques 3 (África) y 4(Asia)

Figura 1.8: Reglas de nombres de variables.

1.1.3.4. Datos

Los datos necesarios para inicializar las variables del modelo se almacenan en un bloque `DATA`[3] categorizadas según si son de demografía, de viviendas, de educación, etc. Las subrutinas del modelo acceden a estas variables mediante el uso de `COMMONs`[4], la forma de compartir variables sin causar conflictos usada en el FORTRAN de la época.

1.1.4. Legado

Durante su primera década de vida, el MML fue estudiado en publicaciones por parte de los investigadores originales y también por terceros. Fue citado por primera vez en 1974 en el segundo simposio del International Institute for Applied Systems Analysis (IIASA) dedicado a los modelos globales [Bru74]. Este simposio fue dividido en dos partes, la primera fue dedicada al MML donde los autores presentaron en distintos segmentos la idea general del modelo, su funcionamiento, los módulos de demografía, urbanización y viviendas, alimentación, y demás. Fue la primera aparición oficial del modelo en el ámbito científico, por lo que también se le dio importancia a la discusión del mismo entre los asistentes del evento. La segunda parte estuvo enfocada en temas concretos de otros modelos globales.

Luego llegó la publicación en 1976 de "Catastrophe or New Society? A Latin American world model" [HSC+76], el libro en inglés asociado al MML escrito por sus autores. Los primeros capítulos describen las motivaciones por las cuáles decidieron desarrollar el modelo, los problemas que veían en los modelos globales de la época (especialmente World3 [MMRBI72], cuyo libro fue publicado en 1972 pero ya se había discutido anteriormente) y cómo pensaban diferenciarse de ellos. Los capítulos siguientes están dedicados al modelo en sí, comenzando con las suposiciones teóricas adoptadas, siguiendo con el modelo matemático y finalizando con varios capítulos enfocados en módulos internos como la demografía, alimentación, viviendas, etc. Acompañando al libro, se hicieron presentaciones en la Fundación Bariloche [Bar76] y en las Naciones Unidas [Nat76].

[3] Documentación de Oracle sobre los DATA
[4] Documentación de Oracle sobre los COMMON

En el momento de su publicación el país estaba pasando por una crisis política que culminó en un golpe de estado y el nuevo gobierno de facto era incompatible con las ideas progresistas que promovía el MML. El grupo de investigación se disolvió, muchos de ellos exiliados a países vecinos y a Europa, impidiendo la publicación de la versión en español de esta primera edición del libro además de la continuación y expansión del proyecto.

A pesar de las trabas políticas, los autores siguieron publicando resultados basados en el modelo y su investigación relacionada. En 1976 Scolnik, sub-director del proyecto original, colaboró en la publicación de un análisis en la aplicación de políticas basadas en objetivos usando al MML como ejemplo de ello [HSM75]. En 1977, varios de los investigadores originales publicaron en la UNESCO dos documentos relacionados al modelo. Por un lado, publicaron un reporte técnico sobre cómo fue implementado el modelo en Fortran [SFL$^+$77] y por el otro los resultados del experimento de adaptarlo a la situación real por ese entonces en Brasil [RLS77]. Este último sería luego la base para un reporte dedicado a la adaptación en un escenario nacional general [RLS79] incluido en el libro 'Models, Planning and Basic Needs" [CL79], dedicado a los modelos globales con énfasis en políticas sociales y necesidades básicas.

Por parte de terceros, el modelo fue estudiado en 1981 en Alemania por W.A. Johr [Joh81], economista político, quién primero resume los argumentos básicos en los que se basó el MML y luego expone sus críticas a éste, principalmente la factibilidad de la aplicación de sus políticas de estado y la estructura gubernamental necesaria. Sin embargo, reconoce el enfoque singular del modelo y su inspiración en resolver los problemas críticos del mundo en ese entonces. El modelo también fue incluido en dos recopilaciones de modelos globales en 1982, la primera en el libro "Groping in the Dark" [MRB82] por parte de los autores de World3, modelo que fue el principal estímulo en la creación del MML. El libro está dedicado a los modelos globales en general, introduciendo los conceptos teóricos, los modelos publicados hasta ese entonces y las respuestas a un cuestionario enviado a los respectivos autores de cada uno de ellos. Tiene un enfoque introspectivo sobre la ciencia de este tipo de modelos y es recomendado para investigadores y el público en general. La segunda recopilación pertenece a un artículo de Richardson [Ric82] que tiene un espíritu más conciso, enfocado más en recolectar los modelos conocidos que en hacer una exposición sobre la ciencia de modelos globales en general.

Varias agencias de las Naciones Unidas aplicaron el modelo con distintos fines: la UNIDO (Organización de las Naciones Unidas para el Desarrollo Industrial) para explorar las consecuencias de los LIMA Targets, la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) para entrenar planificadores, la ILO (Organización Internacional del Trabajo) para el *World Employment Programme* y ECLAC (Comisión Económica para América Latina y el Caribe, CEPAL) para planeamiento educativo. Además, fue influyente en la India donde se incluyeron las necesidades básicas en su Constitución Nacional.

Las citas al modelo mermaron hasta el comienzo del nuevo milenio, cuando en 2000 fue citado en el libro "Global Modeling: Origins, Assessment, and Alternative Futures" [Cha00] de Chadwick, de similares características a "Groping in The Dark" de 1982. También hace una recopilación de modelos globales como aquél pero incluye varios modelos nuevos. Con respecto al MML, menciona sus *ideas socialistas* en la producción de alimentos y cómo esto era incompatible con la dictadura naciente en ese entonces. Además, hace referencia a la dificultad de adquirir el código original y cómo esto imposibilitó un estudio exhaustivo del modelo. Este problema no fue únicamente de Chadwick y fue una de las mayores

motivaciones para el trabajo presentado en esta tesis.

Luego vendrían las rememoraciones del modelo por parte de los autores originales. Comenzando con Gallopin que en 2001 [Gal01] publicó un artículo recordando el MML y los acontecimientos en las décadas desde la publicación del libro de 1976. Este artículo sería incluido, acompañado por textos de similar espíritu por parte de Oteiza y Scolnik (co-director de esta tesis), en la primera parte de la reedición del libro del MML y primera versión en español [HSC+04].

Recientemente, en 2015 fue mencionado en 3 publicaciones de terceros. Por un lado se analizó al modelo desde el lado matemático-sociológico [QST15], por el otro se lo analizó desde una perspectiva epistemológica [Sau15] y también se lo incluyó en una nueva recopilación de modelos globales [CJ15]. Esta recopilación, llevada a cabo por Castro (director de esta tesis) y Jacovkis incluye también modelos que no tuvieron mucha difusión al momento de ser publicados. En 2018, Giri presentó su tesis doctoral [Gir18] donde lleva a cabo un estudio epistemológico, ontológico y teleológico de los modelos globales, además de hacer una recopilación histórica de los distintos modelos del siglo XX. En particular, analiza la repercusión política que tuvo el MML y su controversia con el modelo World3.

Recientemente en 2020 la revista *Pasado abierto* del Centro de Estudios Históricos (CEHis) de la Facultad de Humanidades de la Universidad Nacional de Mar del Plata (UNMDP) hizo una rememoración histórica del MML incluyendo la investigación llevada a cabo, su impacto internacional y su posterior ingreso a un período de total inactividad práctica y limitado a la evocación simbólica [Gro20].

1.1.5. Perduración del modelo

Como vimos en secciones anteriores, el MML es un modelo complejo y singular que supo estar a la altura de otros modelos globales de la época y se sobrepuso ante las críticas de científicos contemporáneos. Sin embargo, el golpe de estado en Argentina, sumado a las corrientes políticas mundiales emergentes en los 80' incompatibles con la idea de regulación de la producción y también la falta de copias de seguridad usables del código a largo plazo, hicieron que el MML gradualmente vaya perdiendo relevancia práctica, aunque no así conceptual, ya que tanto el modelo como la Fundación Bariloche continúan siendo considerados actualmente por la comunidad científica nacional como hitos de la ciencia interdisciplinaria argentina.

El objetivo primario de esta tesis es recuperar el modelo para que sea de libre acceso y pueda ser usado y analizado por quién lo desee. Buscamos contribuir a que el MML recupere un rol relevante en la discusión sobre modelos globales, como supo ser el caso durante la segunda mitad de la década del 70 y primera mitad de los 80.

El primer paso consistió en la digitalización y puesta en funcionamiento de una versión del modelo impresa de 1986, la única disponible al comienzo de este trabajo. Luego comenzaríamos con la traducción del modelo desde FORTRAN77 a Modelica, un lenguaje de modelado cuyos conceptos básicos resumimos en la Sección 1.2 y exponemos la traducción *per se* en la Sección 5. En Modelica, la especificación del comportamiento de los modelos queda liberada de conceptos implementativos [5], lo que facilita su interpretación y manipulación por investigadores que no necesariamente tengan una formación en programación. Esto abre más puertas de investigación del modelo que su implementación original

[5] Por ejemplo, no hay que lidiar con solicitudes y liberaciones de memoria, como es el caso de lenguajes de bajo nivel como C, C++, Fortran, etc.

en FORTRAN77.

Ambos procesos, el de recuperación y traducción, requirieron de una expansión de la documentación disponible del modelo. Aunque las fuentes de ese entonces explican muy bien los conceptos teóricos, necesitábamos información más detallada sobre la implementación y el funcionamiento bajo nivel del modelo. Mediante la aplicación de ingeniería inversa, detallada en la Sección 3, produjimos nueva documentación, como por ejemplo los diagramas de flujo de datos y de control presentados en la Sección 3.3. Esta información es esencial para asistir en la reutilización del modelo, en su completitud o partes de éste, por quién lo desee.

1.2. Conceptos preliminares sobre modelado avanzado con Modelica

En esta sección brindamos información introductoria sobre el lenguaje de modelado Modelica. Este fue seleccionado como candidato para reimplementar el MML luego de haber sido recuperado en FORTRAN, actividad que llevamos adelante hacia el final del presente trabajo. Por ello, su lectura puede ser diferida como se crea conveniente.

El objetivo de la traducción a Modelica es acercar al modelo a una de las tecnologías más modernas y avanzadas en la disciplina de modelado y simulación, que adhiere a la tendencia de utilizar *lenguajes de modelado* en reemplazo de *lenguajes de programación*.

A lo largo del capítulo procuramos poner en contexto las propiedades básicas del lenguaje con aspectos relacionados al MML y a la disciplina de los modelos globales, comentando brevemente una implementación en Modelica del modelo global World3.

1.2.1. ¿Qué es Modelica?

Modelica es un lenguaje de modelado orientado a objetos basado en ecuaciones, en donde las clases y objetos que se definen representan, esencialmente, ecuaciones matemáticas. Un conjunto de objetos define un sistema de ecuaciones, que en la mayoría de los casos tendrá como variable independiente el tiempo. La definición de un modelo en Modelica es posteriormente interpretada y traducida a algún lenguaje de programación tradicional, que será compilado (o interpretado, según el lenguaje) para ejecutar una simulación. Con este enfoque, el modelista expresa sus modelos de una manera mucho más cercana a la descripción del sistema bajo estudio, sin involucrarse en cuestiones de sintaxis o semántica propios de cualquier lenguaje de programación particular.

Un aspecto distintivo de Modelica es que el comportamiento de las variables puede definirse mediante el uso de ecuaciones diferenciales ordinarias (ODEs) respecto del tiempo, además de las instrucciones algorítmicas tradicionales. Al tratarse de ecuaciones, entonces, no importa el orden entre ellas y tampoco la posición de los términos a uno u otro lado de la igualdad. Es decir, no se describen *asignaciones* sino *igualdades*. Por ejemplo, el lado izquierdo de la Figura 1.9 muestra una ODE, siendo t el tiempo, y a su derecha como se expresaría en Modelica.

$$\frac{dx(t)}{dt} + nx(t)^3 = m\cos(x(t)) \qquad \texttt{der(x) + n*x^3 = m*cos(x)}$$

Figura 1.9: Cómo definir una ecuación diferencial ordinaria en Modelica. Forma analítica (izquierda) y declaración en Modelica (derecha). El operador `der()` presupone dependencia de la variable independiente `t` (tiempo).

Algunas de las aplicaciones usuales del lenguaje son en el modelado de sistemas ingenieriles como sistemas hidráulicos, eléctricos, electrónicos o mecánicos, con particular énfasis en sistemas que combinan estas disciplinas (sistemas *multidominio*) con sistemas digitales, dando lugar a sistemas ciber-físicos como por ejemplo los automóviles. En el caso de la industria automotriz, existen publicaciones de aplicaciones de Ford [TBD03] [SKT03] [NBT03], General Motors [TSGT08], Toyota [SM02], BMW [WvGC08] y Daimler [BSK+09] utilizando Modelica.

Varios productos de software implementan el estándar Modelica, habiendo alternativas pagas o de software libre. Entre las opciones comerciales están AMESim [SIE], Dymola [Sysb], CyModelica [Gro], Wolfram SystemModeler [Res], Simulation X [Gmb], MapleSim [Map] y CATIA Systems [Sysa]. Entre las versiones libres están jModelica.org [AB] y OpenModelica [Cona].

En esta tesis adoptamos OpenModelica para implementar una traducción (parcial) del MML y es la que usaremos para ilustrar algunos ejemplos sencillos. El lector que no requiera de aspectos técnicos sobre Modelica puede pasar directamente a la sección 1.2.3 donde se ilustra el uso del lenguaje en un sistema de poblaciones y en un modelo global intersectorial, para tener una impresión del potencial de Modelica por fuera de las aplicaciones clásicas en ingenierías.

1.2.2. Especificación del lenguaje

La especificación del lenguaje [MLS17] cae bajo la responsabilidad de la Modelica Association [MA], organización sin fines de lucro, que también provee la Modelica Standard Library (MSL) [Assb], biblioteca de libre acceso con varios modelos y funciones genéricas de uso general. Para el lector sin conocimientos previos del lenguaje recomendamos el Modelica Overview [Ott], donde se da un vistazo general de lo que puede hacerse con el lenguaje, y también el libro de Peter Fritzson [Fri14] para responder dudas sobre una gran variedad de temas específicos del lenguaje y sus aplicaciones.

El espíritu de esta sección es introducir al lector con el conocimiento básico necesario para comprender lo tratado en esta Tesis y, de quedar alguna duda, que pueda recurrir a las fuentes mencionadas en el párrafo anterior. Presentamos entonces al software usado en la Tesis y algunos conceptos clave del lenguaje (como clases, ecuaciones, algoritmos, conexiones, etc.)

1.2.2.1. OpenModelica

El ambiente de desarrollo elegido y que recomendamos es OpenModelica [Cona] de código abierto y libre. Puede bajarse para Linux, Mac y Windows desde la página oficial de OpenModelica y cualquier versión superior a la `1.13` debería ser compatible con la versión del MML desarrollada en esta Tesis.

El ambiente incluye al programa OMEdit capaz de crear, editar y simular modelos en Modelica. En la Figura 1.10 mostramos un modelo de ejemplo llamado `RollingWheel` que ya forma parte de la MSL. En la sub-ventana de la izquierda se encuentra el navegador de bibliotecas con la búsqueda, en la sub-ventana del centro se muestra su representación gráfica y a la derecha su documentación. Según la información provista en el modelo vemos que es un sistema mecánico rotacional, que representa la interacción entre una fuente de torque, un componente de inercia, una rueda, una masa y una fuerza de resistencia. Más

adelante veremos en detalle cómo se relacionan la *vista gráfica* y *las ecuaciones* de los modelos en general.

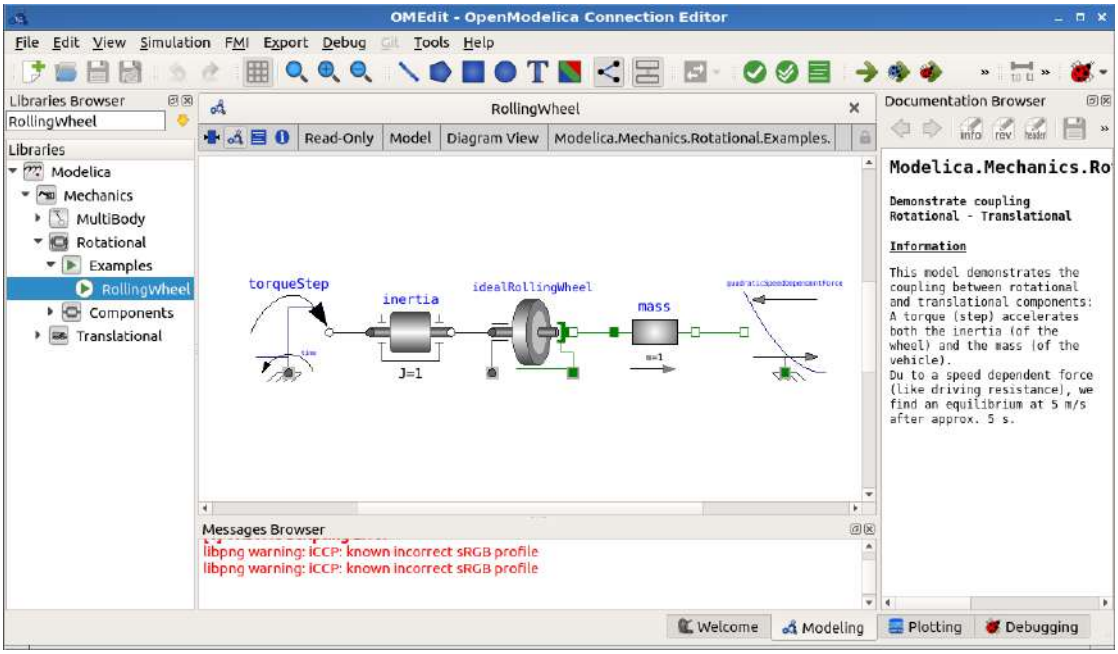


Figura 1.10: Interfaz gráfica de usuario de OMEdit. Cada componente conlleva asociado su propio sistema (parcial) de ecuaciones, que definen el sistema completo de ecuaciones.

OpenModelica también provee al programa OMC (Open Modelica Compiler), un alternativa de OMEdit para trabajar mediante línea de comandos, que facilitó el proceso de verificación al ofrecer la posibilidad de automatizar las secuencias de simulaciones.

1.2.2.2. Ecuaciones

Modelica describe el comportamiento de modelos principalmente mediante el uso de ecuaciones. Estas ecuaciones son más abstractas que las asignaciones en los lenguajes de programación tradicionales. Por ejemplo, en la Figura 1.11 definimos una ecuación en el lado izquierdo y mostramos 4 posibles asignaciones equivalentes del lado derecho.

En Modelica un conjunto de ecuaciones forman un sistema, similar al sistema de ecua-

$$a * b = 2c + d + 3$$

(a) Ecuación.

$$a \leftarrow \frac{2c + d + 3}{b}$$

$$b \leftarrow \frac{2c + d + 3}{a}$$

$$c \leftarrow \frac{a * b - d - 3}{2}$$

$$d \leftarrow a * b - 2c - 3$$

(b) Asignaciones equivalentes.

Figura 1.11: Ejemplo de equivalencia entre ecuaciones y asignaciones.

ciones usado en matemática y relacionado con reducción de filas de matrices. Como en matemática, el sistema debe tener igual cantidad de variables que de ecuaciones *únicas*[6] para que el sistema esté determinado. Si tomáramos como sistema a la ecuación del ejemplo, estaría sub-determinado al tener 1 ecuación y 4 incógnitas. En ese caso, deberíamos agregar 3 ecuaciones describiendo las variables restantes, siendo más simple la opción de que sean similares a asignaciones, como en el sistema de la Figura 1.12.

$$a = 3$$

$$b = 4$$

$$c = 5$$

$$a * b = 2c + d + 3$$

Figura 1.12: Sistema de ecuaciones determinado.

1.2.2.3. Tiempo continuo

Por defecto, aunque no se lo referencie directamente, el tiempo está siempre implícito en las ecuaciones. Así es el caso en el ejemplo anterior, donde las variables tendrán un valor constante de principio a fin en las simulaciones. Cambiemos la tercera ecuación de la Figura 1.12 anterior por $der(c) = 2$, equivalente a $\frac{dc}{dt} = 2$, para que c y d varíen pero a y b continúen siendo constantes. Para simular el nuevo comportamiento definimos un modelo llamado `CuatroVariables` que declara las 4 variables usadas, le asigna 0 a c como valor inicial y luego define las ecuaciones del sistema.

En la Figura 1.13 mostramos en el lado izquierdo la implementación del modelo acompañada en el lado derecho con el resultado de una simulación para $0 \leq t \leq 1$. En ella vemos cómo a y b son constantes mientras que c crece y d decrece con respecto al tiempo.

1.2.2.4. Tiempo discreto

El lenguaje es capaz de combinar variables de tiempo continuo y discreto, que pueden ser afectadas por eventos o definidas en base a relojes (llamados *clocks* en el lenguaje). Así será el caso de la versión en Modelica del MML, donde se disparan eventos para manejar el pasaje entre años, asemejándose a la versión original en Fortran que hacía la transición manualmente.

Veamos ahora que sucede si cambiamos la ecuación en la línea 8 de la Figura 1.13 por una similar pero en tiempo discreto con base temporal de 0.1 segundos, afectando a c pero también a d. Esta nueva ecuación afecta discretamente la pendiente de c por un factor de 2/10 para lograr una forma similar al de su versión en tiempo continuo.

La Figura 1.14 muestra el nuevo modelo, donde declaramos a c como variable discreta y usamos a `when` y a `sample` para marcar los tiempos de los eventos discretos. El primer argumento del `sample` indica el tiempo del comienzo de los eventos del `when` y el segundo el período de estos eventos. En este caso, el primer evento será disparado en tiempo 0.1 y luego le seguirán los tiempos 0.2, 0.3, 0.4, etc.

[6] Es decir, si una ecuación es combinación lineal de otras entonces no agrega nueva información al sistema y es ignorada.

```
model CuatroVariables
   Real a,b,c,d;
 initial equation
   c = 0;
 equation
   a = 3;
   b = 4;
   der(c) = 2;
   a*b = 2*c + d + 3;
end CuatroVariables;
```

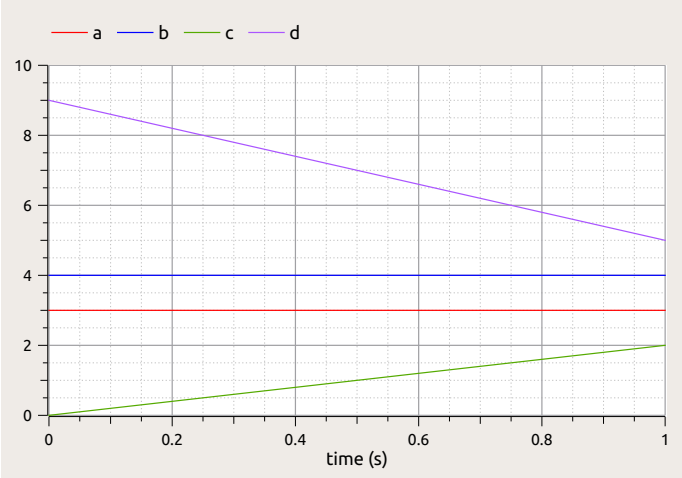


Figura 1.13: Implementación del modelo `CuatroVariables` (izquierda) y su simulación para $0 \leq t \leq 1$ (derecha).

```
model
    CuatroVariablesDiscreto
   Real a,b,d;
   discrete Real c;
 initial equation
   c = 0;
 equation
   a = 3;
   b = 4;
   a*b = 2*c + d + 3;
   when sample(0.1,0.1) then
     c = 2/10 + pre(c);
   end when;
end CuatroVariablesDiscreto;
```

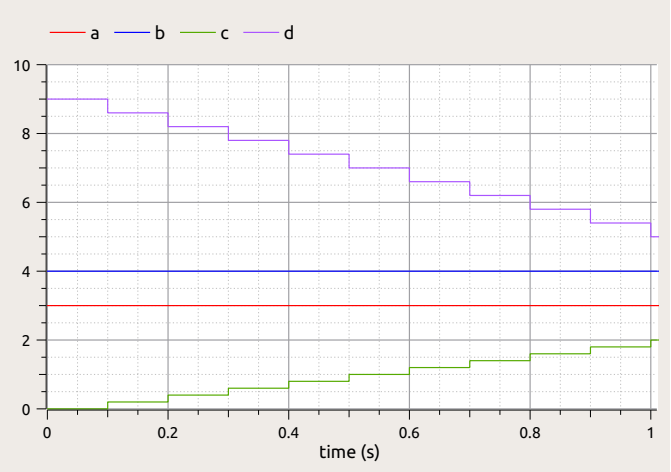


Figura 1.14: Implementación y simulación del modelo `CuatroVariablesDiscreto` con $0 \leq t \leq 1$.

Simulamos en $0 \leq t \leq 1$ y vemos como los valores iniciales y finales son iguales respectivamente que en el ejemplo continuo. Sin embargo, en este caso las variables dependientes del tiempo, c en forma directa y d en forma indirecta, presentan actualizaciones solo en los tiempos donde ocurren los eventos discretos modelados mediante la cláusula `when`.

1.2.2.5. Algorithm

La traducción del MML desde FORTRAN77 (paradigma imperativo) hacia Modelica (paradigma de modelado) requirió primero un pasaje conceptual y luego un pasaje de código desde instrucciones algorítmicas hacia ecuaciones diferenciales ordinarias. Este pasaje es muchas veces no trivial dado que consiste en proponer ecuaciones que pueden no parecer para nada similares a las instrucciones algorítmicas de partida. Este es un problema clásico en el cual "el modelo está mezclado con el simulador" y que los lenguajes *de modelado* (no de programación) como Modelica pretenden subsanar.

Dadas las restricciones de tiempo para esta tesis decidimos usar la opción `algorithm`

del lenguaje para las secciones de código en los que no se justificara la inversión de tiempo. Como su nombre lo indica, las secciones `algorithm` en Modelica dan la posibilidad de expresar comportamiento de manera algorítmica, similar al paradigma de programación imperativo, en lugar de especificar sistemas de ecuaciones como en las secciones `equation` vistas antes. La sintaxis de las instrucciones algorítmicas es `v:=expr;` donde `v` es una variable y `expr` es una expresión del lenguaje.

Tomando de base al sistema de ecuaciones presentado en la Figura 1.12, antes de los cambios referenciando al tiempo, el único cambio necesario para usar `algorithm` es en la última ecuación. Por cómo habíamos diseñado el sistema, esa ecuación era la única con información sobre la variable d y entonces podemos reemplazarla por su asignación correspondiente de la Figura 1.11b, al comienzo de esta sección, para obtener el modelo de la izquierda de la Figura 1.15.

Ambos modelos de la Figura 1.15 describen comportamientos equivalentes, es decir, sus simulaciones van a ser indistinguibles. Sin embargo, el proceso de *limpieza* partiendo desde el modelo de la derecha para obtener el de la izquierda no es para nada trivial. Durante el proceso de traducción del MML, en algunas secciones de código no se justificó la inversión de tiempo para llevar a cabo dicha limpieza y decidimos usar directamente versiones similares a la de la derecha.

```
model CVAlgorithm
   Real a,b,c,d;
 algorithm
   a := 3;
   b := 4;
   c := 5;
   d := a*b -2*c -3;
end CVAlgorithm;
```

(a) Modelo Simple

```
model CVAlgorithm2
   Real a,b,c,d;
 algorithm
   a := 0;
   b := a+4;
   a := -5;
   c := a+10;
   a := 3;
   d := a*b -2*c -3;
end CVAlgorithm2;
```

(b) Modelo con muchas asignaciones

Figura 1.15: Ejemplos equivalentes del uso de `algorithm`

Un mismo modelo puede combinar secciones `equation` y `algorithm`, usando ecuaciones diferenciales ordinarias para algunas variables e instrucciones algorítmicas para otras.

1.2.2.6. Clases

Las clases en Modelica se asemejan a las usadas en lenguajes de programación, como Java o C++, porque describen una especificación general que luego puede ser reusada (instanciada) en distintos casos particulares. Por ejemplo, una clase puede llamarse `Persona`, que incluye un `DNI, Nombre, Ocupación y Domicilio`, y puede ser *instanciada* en casos particulares como una persona de nombre `Atahualpa`, con su respectiva información.

Dicha instanciación facilita mucho el reuso de código dado que instancias de una misma clase pueden ser usadas en contextos tan distintos como en modelado de ruedas de automóviles o en poblaciones de conejos. Tomemos de ejemplo un `Punto` de dos coordenadas, como el especificado en el lado izquierdo de la Figura 1.16 que declara las variables {`x,y`} y no contiene ecuaciones. Como vimos en las secciones anteriores, no es un modelo simulable por tener 3 variables y ninguna ecuación.

Las instancias de la clase `Punto` pueden ser usadas en nuevos modelos, como por ejemplo el presentado en el lado derecho de la Figura 1.16 que modela el movimiento de 3 puntos. El primero se mueve incrementando su coordenada del eje `x`, el segundo lo propio con la coordenada `y` mientras que el tercero lo hace incrementando ambas coordenadas en simultáneo. Este modelo tiene 6 variables, dos por cada punto, y por eso fue necesario especificar 6 ecuaciones. Además, notar que se especifican sus derivadas pero no sus valores iniciales y es porque en Modelica las variables de estado se asumen inicializadas en 0 por defecto.

```
class Punto
   Real x,y;
end Punto;
```

```
class PuntosEnMovimiento
   Punto p1, p2, p3;
equation
   // Punto 1
   der(p1.x) = 1;
   der(p1.y) = 0;
   // Punto 2
   der(p2.x) = 0;
   der(p2.y) = 1;
   // Punto 3
   der(p3.x) = 1;
   der(p3.y) = 1;
end PuntosEnMovimiento;
```

Figura 1.16: Definición de `Punto` y ejemplo de instanciación.

1.2.2.7. Parámetros

Para adaptar las instancias a casos particulares se pueden definir en las clases uno o más parámetros, llamados `parameter` en Modelica. Un modelo puede usar varias instancias de una misma clase pero configuradas con parámetros distintos, definidos al momento de la instanciación y no afectados por el tiempo durante el transcurso de la simulación. Estos parámetros pueden ser usados para inicializar variables, identificar modelos para facilitar la lectura de resultados, alterar el flujo de datos adentro de `if`, etc.

Si en el ejemplo anterior hubiéramos necesitado que cada punto tuviera un color, hubiéramos tenido que agregar un parámetro para representarlo. Podemos agregarle a la clase `Punto` un parámetro de tipo `String` que especifique el color en español[7]. En la Figura 1.17 mostramos un posible ejemplo de esto.

Notar que el único cambio con respecto al modelo anterior es en la instanciación de los puntos mientras que las ecuaciones son las mismas. Esto se debe a que en este caso el parámetro del color agrega información sobre las instancias que no afecta ni depende del comportamiento del modelo.

1.2.2.8. Tipos de clases

Existen distintos tipos de clases. En la sección anterior definimos modelos usando a `class` mientras que anteriormente usábamos a `model`. Son intercambiables, es decir si a los

[7] En un caso real, hubiera sido más favorable representar el color con una clase especializada que use RGB, HSV y/o HSL

```
model PuntoColorido
String color;
Real x,y;
end PuntoColorido;
```

```
model PuntosColoridosEnMovimiento
 PuntoColorido p1(color="carmesí"),
               p2(color="siena"),
               p3(color="añil");
equation
  // Punto 1
  der(p1.x) = 1;
  der(p1.y) = 0;
  // Punto 2
  der(p2.x) = 0;
  der(p2.y) = 1;
  // Punto 3
  der(p3.x) = 1;
  der(p3.y) = 1;
end PuntosColoridosEnMovimiento;
```

Figura 1.17: Puntos con colores como parámetros.

puntos los definíamos como `model Punto`, no hubiera habido diferencia alguna. Sin embargo, por convención se la toma a `class` como tipo base y a `model` como una especialización. Entre las otras especializaciones se tiene a `block`, que requiere que los conectores usados especifiquen si son de tipo `input` o `output`, a `record`, que no puede contener ecuaciones y es usada para almacenamiento de datos, y a `function` que es invocable y no es afectada por el tiempo. La Figura 1.18 resume los tipos de clases aquí mencionados.

Tipo	Descripción
`class`	Tipo base
`model`	Equivalente a `class`
`block`	Los conectores deben ser `input` o `output`
`record`	No puede haber `equation` o `algorithm`
`function`	No se usa `equation`, sólo `algorithm`, y no se tiene en cuenta el tiempo

Figura 1.18: Resumen de las especializaciones de `class`.

1.2.2.9. Partial Class

Lo que se llama *clase abstracta* en otros lenguajes de programación, como C++ o Java, se llama *partial class* en Modelica. Puede interpretarse como una *clase incompleta* y no es necesario que cumpla con algunas reglas sintácticas. Por ejemplo, puede tener *variables libres*, según la definición del área de teoría de lenguajes[8], que pueden ser instanciadas como parámetros o variables de Modelica por los modelos que extiendan (`extend`) la clase.

El uso de las *clases parciales* nos permite compartir el comportamiento de una misma clase entre distintos modelos cuando no es viable el uso de parámetros. Los modelos *heredan* el comportamiento de la *clase parcial* al extenderla pero pudiendo inicializar sus variables a gusto. Por ejemplo, en el lado izquierdo de la Figura 1.19 se define una clase parcial

[8] A fines prácticos de esta tesis, puede tomarse como un *símbolo* que en algún momento va a ser reemplazado por *algo*.

`CompCompartido` con una *variable libre* llamada `inits`, cuyo significado será dado por las clases que la extiendan.

Así es el caso de la clase `CasoUno` a su derecha, que inicializa a `inits` como un `InitsCasoUno`, que define a `p1` y `p2` como variables `Real` y les asigna 4 y -3 respectivamente. De esta manera, el modelo termina siendo completo y simulable. Si alguien quisiera definir un `CasoDos` con distintos valores de `p1` y `p2` entonces debería extender a `CompCompartido` de una manera similar.

```
partial class CompCompartido
  Real x, y;
equation
  x = inits.p1;
  y = inits.p2;
end CompCompartido;
```

(a) Clase parcial.

```
record InitsCasoUno
  Real p1 = 4;
  Real p2 = -3;
end InitsCasoUno;
```

```
class CasoUno
  extends CompCompartido;
  InitsCasoUno inits;
end CasoUno;
```

(b) Extensión.

Figura 1.19: Ejemplo de una clase parcial con una extensión posible.

En estos ejemplos, no es clara la ventaja de usar clases parciales en vez de parámetros, dando la sensación de que complejiza el código innecesariamente. Sin embargo, la situación es distinta cuando se tiene 40 o más parámetros, como es el caso del MML. Para inicializar varias instancias, cada una con sus datos de inicialización distintos, es más fácil separar su comportamiento, que es compartido y quedaría en la clase parcial, de su inicialización, que es individual y está definido en la clase que extiende. Para el MML usamos las clases parciales en modelado de los bloques[9], separando sus inicializaciones individuales del comportamiento compartido entre ellos.

1.2.2.10. Block

Una de las especializaciones de `class` mencionadas es `block`, usada en modelos que siguen el paradigma de entrada/salida de datos. Su interfaz está definida por conectores que pueden ser de entrada (`input`) o de salida (`output`), siendo estos el único medio de comunicación entre `blocks`. Estas características facilitan la modularización al dejar explícitas las dependencias de cada sub-modelo y las conexiones entre ellos. Además, su simplicidad hace que su representación gráfica sea más directa.

En la Figura 1.20 definimos el modelo `DosPorTres` que instancia objetos de las clases `Modelica.Blocks.Sources.Constant` y `Modelica.Blocks.Math.Product`, especializaciones de `block`. El modelo representa el cálculo del producto (`prod`) entre la constante 2 (`const_2`) y la constante 3 (`const_3`), indicando que las salidas de las constantes son entradas del producto mediante el uso de dos `connect`. Para los usos prácticos de esta tesis, la sentencia `connect(a.out,b.in)` es equivalente a la ecuación `a.out=b.in`.

Omitimos los contenidos de los `annotation` para facilitar la lectura y no confundir al lector, pero dentro de ellos está especificada la representación gráfica del objeto que

[9] Países Desarrollados, Latinoamérica, África y Asia, explicadas en la Sección 1.1.2.2

tienen asociado. En este ejemplo, las instancias, los conectores y el modelo en sí tienen un `annotation` asociado.

```
1 model DosPorTres
2   Modelica.Blocks.Sources.Constant const_2(k = 2) annotation(...);
3   Modelica.Blocks.Sources.Constant const_3(k = 3)  annotation(...);
4   Modelica.Blocks.Math.Product prod annotation(...);
5 equation
6   connect(const_2.y, prod.u1) annotation(...);
7   connect(const_3.y, prod.u2) annotation(...);
8 annotation(...);
9 end DosPorTres;
```

Figura 1.20: Modelo `DosPorTres` sin `annotation`.

1.2.2.11. Vista gráfica

Exhibimos los `annotation` faltantes del ejemplo anterior en la Figura 1.21, donde vemos que pueden llegar a ser bastante extensos. Aunque pueden ser modificadas a mano en caso de ser necesario, son manipuladas automáticamente por IDEs[10] ante cambios en la vista gráfica del modelo. Entre los `annotation` del ejemplo, podemos notar que las instancias usan del tipo `Placement` mientras que los `connect` usan del tipo `Line`. La `annotation` de la línea 16 corresponde al modelo `DosPorTres` y especifica la versión de Modelica con la que fue desarrollado.

En base a esas `annotation`, OMEdit produjo la vista gráfica presentada en la Figura 1.22. En ella están representadas las instancias `const_2`, `const_3` y `prod` y también las líneas correspondientes a los `connect`. Además, notar que para las constantes se indica su respectivo parámetro `k`, especificado en el código entre paréntesis en sus inicializaciones. Recurriremos al uso de vistas gráficas para proveer una representación visual compacta y no ambigua de la relación entre submodelos (sectores económicos) del MML. Estas relaciones expresan exactamente las equivalencias entre variables declaradas en el código del modelo.

1.2.2.12. Simulación

En las discusiones anteriores nos habíamos concentrado en los modelos y sus alternativas de diseño, pero también tuvimos que configurar algo independiente a ellos: el intervalo de tiempo para la simulación. Este aspecto no describe a los modelos sino a los experimentos que se llevan a cabo con ellos, llamados simulaciones, que en Modelica pueden ser configuradas de muy variadas maneras. En la Figura 1.23 presentamos una pequeña muestra de estas opciones, incluyendo los tiempos de comienzo ("Start Time" o `startTime`) y fin ("Stop Time" o `stopTime`). El resto de las opciones (como por ejemplo el método numérico usado para aproximar las ecuaciones diferenciales) a los fines prácticos de esta Tesis pueden quedar con su valor por defecto sin mayor análisis.

[10] Los Entornos de Desarrollo Integrado, IDE por sus siglas en inglés, ofrecen facilidades al programador como manejo de archivos, resaltamiento de sintaxis, vista gráfica de modelos, *debuggeo* integrado, etc. El usado por nosotros es OMEdit, ofrecido por el ambiente de desarrollo OpenModelica

```
model DosPorTres
  Modelica.Blocks.Sources.Constant const_2(k = 2) annotation( Placement(
    visible = true, transformation(origin = {-50, 50}, extent = {{-10,
    -10}, {10, 10}}, rotation = 0)));
  Modelica.Blocks.Sources.Constant const_3(k = 3)  annotation(
    Placement(visible = true, transformation(origin = {-50, 10},
    extent = {{-10, -10}, {10, 10}}, rotation = 0)));
  Modelica.Blocks.Math.Product prod annotation( Placement(visible
    = true, transformation(origin = {-10, 30}, extent = {{-10, -10},
    {10, 10}}, rotation = 0)));
equation
  connect(const_2.y, prod.u1) annotation( Line(points = {{-39,
    50}, {-30, 50}, {-30, 36}, {-22, 36}}, color = {0, 0, 127}));
  connect(const_3.y, prod.u2) annotation( Line(points = {{-39,
    10}, {-30, 10}, {-30, 24}, {-22, 24}}, color = {0, 0, 127}));
annotation(uses(Modelica(version = "3.2.2")));
end DosPorTres;
```

Figura 1.21: Modelo `DosPorTres` con `annotation`.

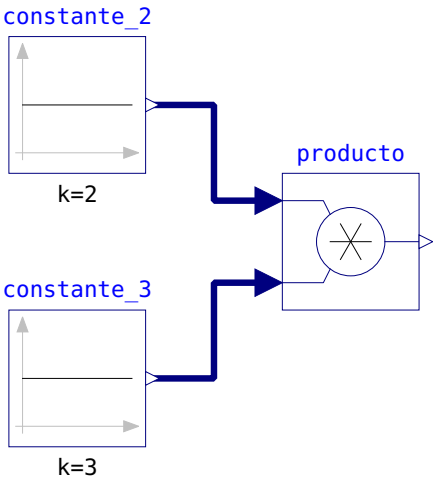


Figura 1.22: Vista gráfica del modelo `DosPorTres`.

1.2.3. Ejemplos

Como vimos en la Sección 1.2.1, Modelica ha sido la herramienta elegida para el modelado de objetos en entornos ingenieriles por parte de varias empresas multinacionales. Sin embargo, también es muy útil en el modelado de sistemas de poblaciones, económicos, naturales y hasta la combinación de ellos como es el caso del modelo estudiado en esta tesis.

En esta sección mostramos algunos ejemplos de aplicaciones no ingenieriles que consideramos sirven como introducción a la versión del MML por presentar en la Sección 5. Una exposición más formal de este tipo de aplicaciones puede encontrarse en el libro "Principles of object oriented modeling and simulation with Modelica" [Fri14], respectivamente en los capítulos "15.4: Biological and Ecological Systems" y "15.5: Economic Systems", y en el artículo "Human-Nature Interaction in World Modeling with Modelica" [CFC+14], que fue usado como base para el desarrollo de dichos capítulos.

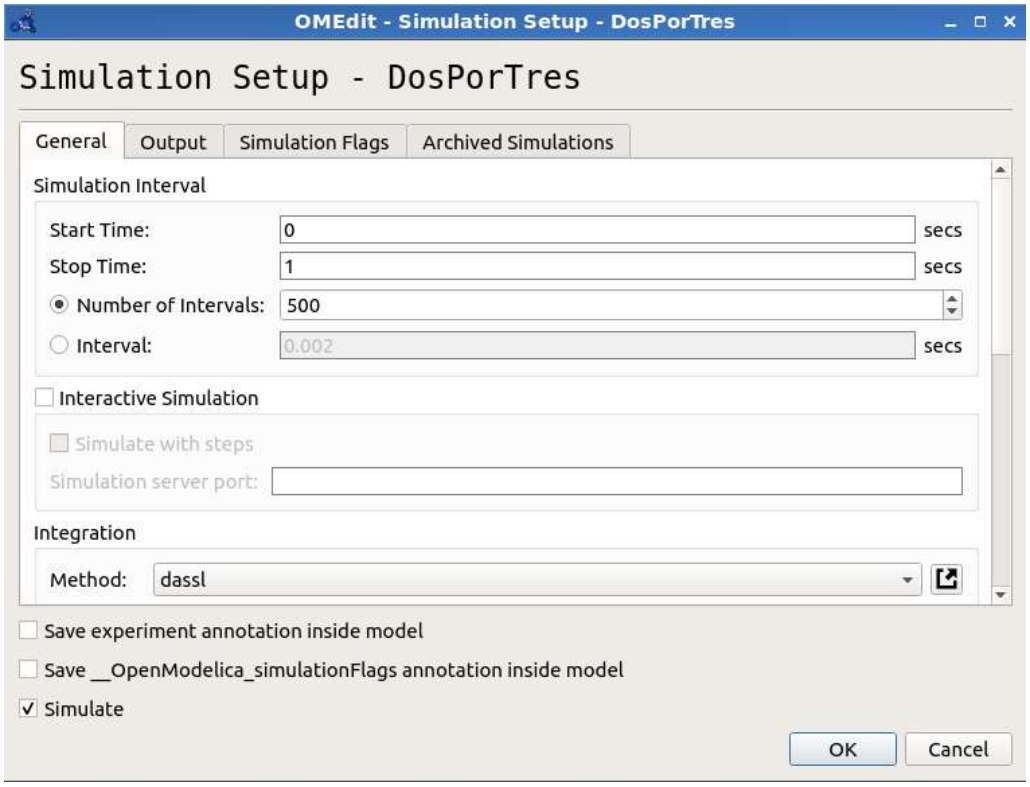


Figura 1.23: Configuración de la simulación en OMEdit.

1.2.3.1. Modelo presa-depredador (Lotka-Volterra)

Un ejemplo simple de ecuaciones diferenciales para implementar en Modelica puede ser Lotka-Volterra [Lei68], que simula las relaciones entre poblaciones de presas (por ejemplo, conejos) y depredadores (por ejemplo, lobos) en un ecosistema. Sus ecuaciones principales son las siguientes:

$$\text{Presas:} \quad \frac{dx}{dt} = \alpha x - \beta xy \tag{1.5}$$

$$\text{Depredadores:} \quad \frac{dy}{dt} = \delta xy - \gamma y \tag{1.6}$$

donde $x(t)$ es la población de presas, $y(t)$ la población de depredadores, $\frac{dy}{dt}$ y $\frac{dx}{dt}$ las respectivas tasas de cambio de las poblaciones, t el tiempo y $\{\alpha, \beta, \delta, \gamma\}$ los parámetros a definir por el modelista. En la Figura 1.24 mostramos una posible implementación en Modelica, definiendo valores ad-hoc por defecto de los parámetros α, β, δ y γ, además de los valores iniciales de `prey_pop` (población de presas) y `pred_pop` (población de depredadores).

```
class LotkaVolterra
  // Behavioural parameters
  parameter Real alpha = 0.1  "Prey: population growth rate";
  parameter Real beta  = 0.02 "Prey: death rate";
  parameter Real delta = 0.02 "Predator: population growth rate";
  parameter Real gamma = 0.09 "Predator: death rate";
  // Initialization parameters
  parameter Real init_prey_pop = 10 "Predator: initial population";
  parameter Real init_pred_pop = 10 "Predator: death rate";
  // Population variables
  Real prey_pop "Prey: population";
  Real pred_pop "Predator: population";
initial equation
  prey_pop = init_prey_pop;
  pred_pop = init_pred_pop;
equation
  der(prey_pop) = alpha*prey_pop          - beta*prey_pop*pred_pop;
  der(pred_pop) = delta*prey_pop*pred_pop - gamma*pred_pop;
end LotkaVolterra;
```

Figura 1.24: Implementación de Lotka-Volterra en Modelica.

En la Figura 1.25 presentamos tres simulaciones del modelo, graficando sólo las variables poblacionales. El primer gráfico corresponde a la corrida estándard para el intervalo $0 \leq t \leq 200$, donde ambas poblaciones comienzan con 10 y luego mientras una crece decrece la otra, y viceversa, exhibiendo un comportamiento oscilante recurrente (sistema marginalmente estable).

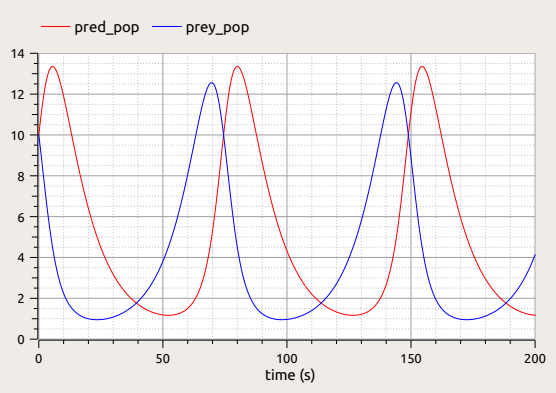
Los gráficos restantes corresponden a simulaciones en las cuales aplicamos modificaciones en los parámetros. En la Figura 1.25b, se simula para el mismo intervalo de tiempo e inicializando la población de presas `init_prey_pop` en 1 en vez de 10 lo cual afecta levemente el comportamiento del modelo pero luego entra en ciclos de duración y forma similar al de la corrida estándard.

En cambio, en la Figura 1.25c se simula para $0 \leq t \leq 80$ y se modifica uno de los factores de crecimiento poblacional de los depredadores (llamado `delta`). Este cambio tiene un efecto de sobrepoblación de depredadores, causando una extinción casi inmediata de las presas que luego causaría un efecto similar en los depredadores.

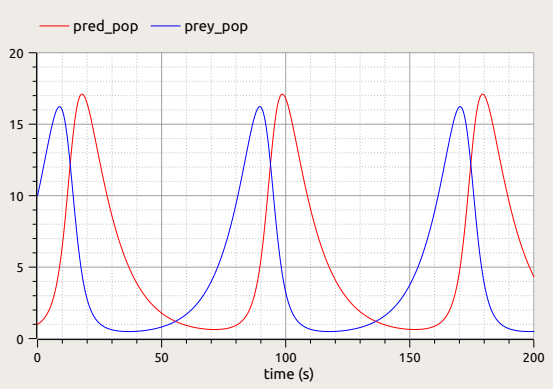
1.2.3.2. Versión Modelica del modelo global World3

En la Sección 1.1.1 mencionamos al modelo World3, modelo estudiado en el libro Limits To Growth [MMRBI72] y principal estimulante de la investigación que culminaría en el MML. Aunque la versión original fue implementada en Dynamo y luego traducida a Stella II, existe también una versión de terceros en Modelica que es parte de la biblioteca SystemDynamics [Cel08] y de acceso libre en GitHub [Cel].

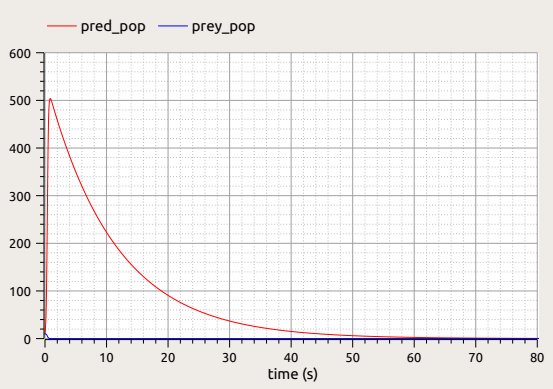
El libro del modelo enumera varios escenarios con pequeños cambios entre ellos y con aplicaciones de políticas distintas. El escenario usado para fundamentar *la catástrofe* corresponde al `Scenario 1`, cuya representación gráfica se muestra en la Figura 1.26. En ella podemos ver los varios submódulos del modelo, como los dedicados a la población, tierra cultivable, contaminación, etc, y como están conectados entre sí. Al ser parte de la biblioteca de SystemDynamics, el modelo tiene acceso a reimplementaciones de sus componentes originales como `tanks, levels, sinks o rates`. Un ejemplo de esto se



(a) Valores por defecto.



(b) `init_prey_pop = 1`.



(c) `delta = 1`.

Figura 1.25: Simulaciones del modelo Lotka-Volterra

puede ver en la Figura 1.27, correspondiente a la representación gráfica del submódulo `Population Dynamics`, donde se ve el uso de dichos componentes.

Los escenarios se diferencian en los valores por defecto de los 70 parámetros del modelo. Dichos parámetros pueden ser personalizados por el usuario en cualquier escenario, de forma tal de acomodar las simulaciones a sus necesidades. Por ejemplo, se pueden aumentar los recursos no renovables iniciales, los años de aplicaciones de políticas, el porcentaje de tierra usado en agricultura, etc.

La Figura 1.28 presenta los resultados de dos simulaciones del `Scenario 1`, ambas en el intervalo $1900 \leq t \leq 2100$ y para las variables de población y recursos no renovables disponibles, normalizada como `nr_resources_normalized = nr_resources/120` para poder graficar ambas variables juntas. Elegimos estas variables porque una de las conclusiones de los autores es que la catástrofe sería causada por una explosión en la población mundial, lo que pondría un estrés insostenible sobre los recursos no renovables del planeta.

En la primera simulación, Figura 1.28a, ningún parámetro fue modificado y vemos cómo los recursos no renovables se agotan al ritmo que aumenta la población, como remarcaban los autores del modelo. Para la segunda, Figura 1.28b, se disminuyó el factor de utilización de los recursos no renovables (`p_nr_use_fact_1`) desde 1 hasta 0.1 con el objetivo de reducir el estrés sobre ellos, resultando en una línea más suave en el gráfico. Sin embargo, esto no tuvo un efecto considerable sobre la población y su forma sigue describiendo una *campana*.

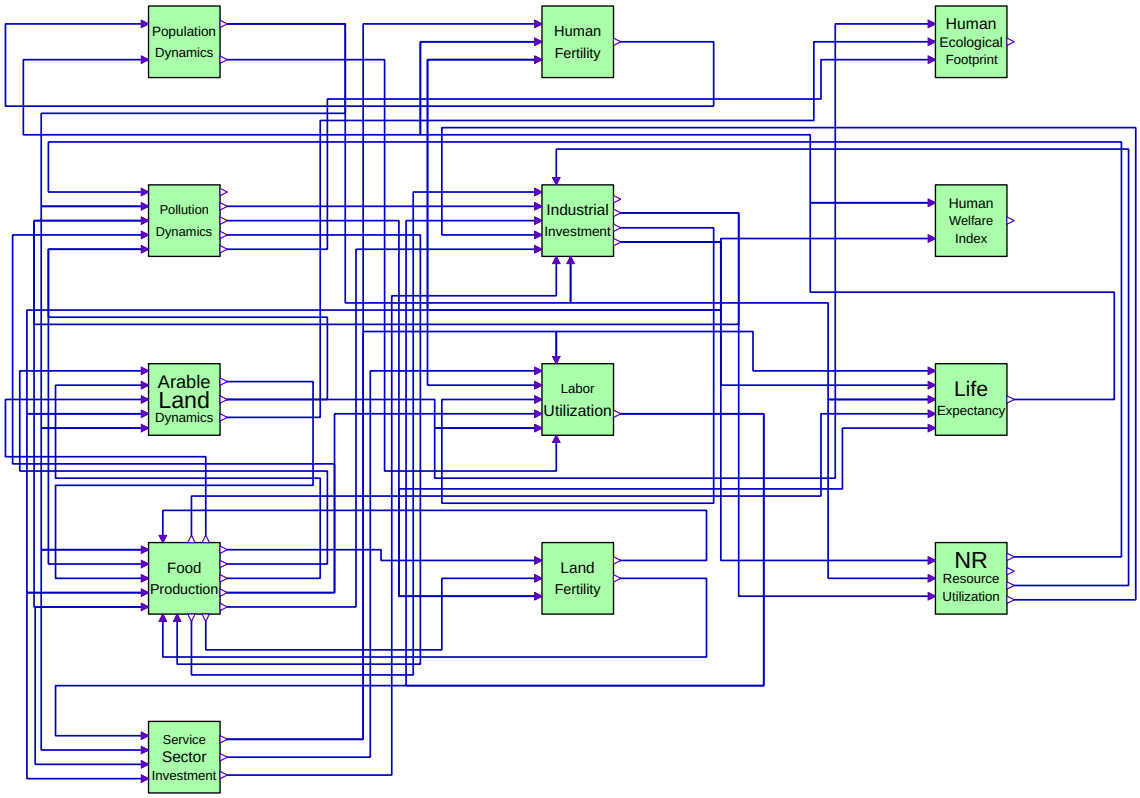


Figura 1.26: Vista gráfica del Scenario 1 de World3

Podemos concluir que la reducción en la dependencia en los recursos no renovables no es suficiente para prevenir la catástrofe según World3.

El objetivo de ambos ejemplos, los de Lotka-Volterra y World3, fue darle una muestra al lector de los experimentos posibles de hacer con Modelica y la facilidad con la que pueden llevarse a cabo. En las secciones siguientes nos enfocaremos únicamente en el MML, exponiendo los procesos de recuperación, traducción y desarrollo de interfaz web, haciendo experimentos de espíritu similar a los de esta sección en cada paso.

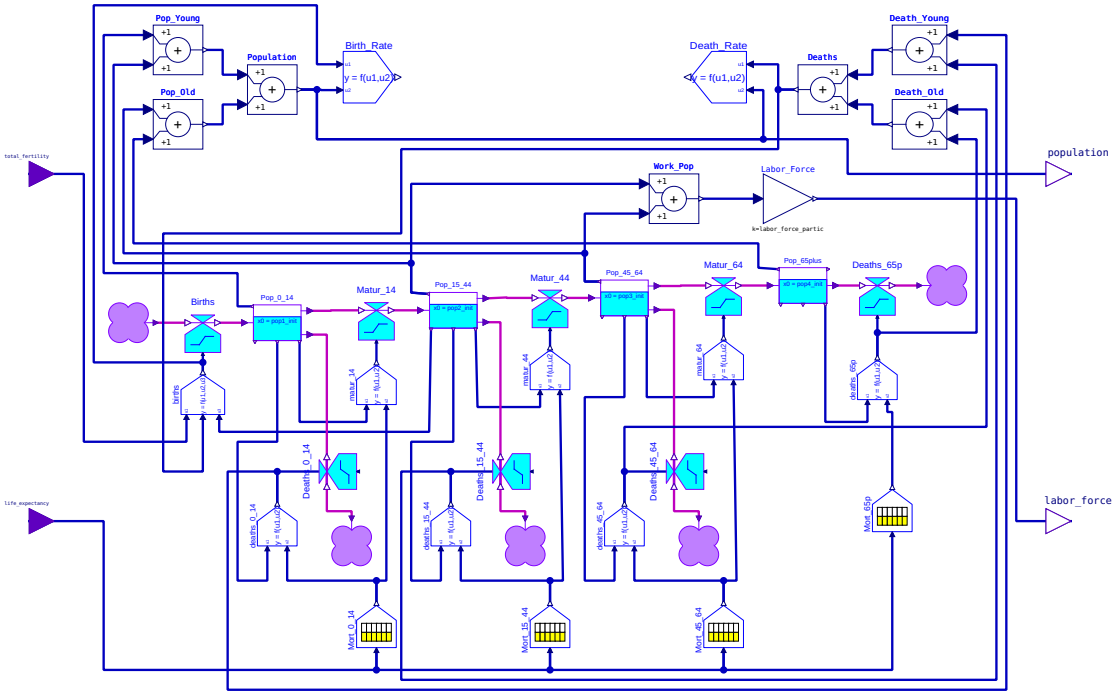
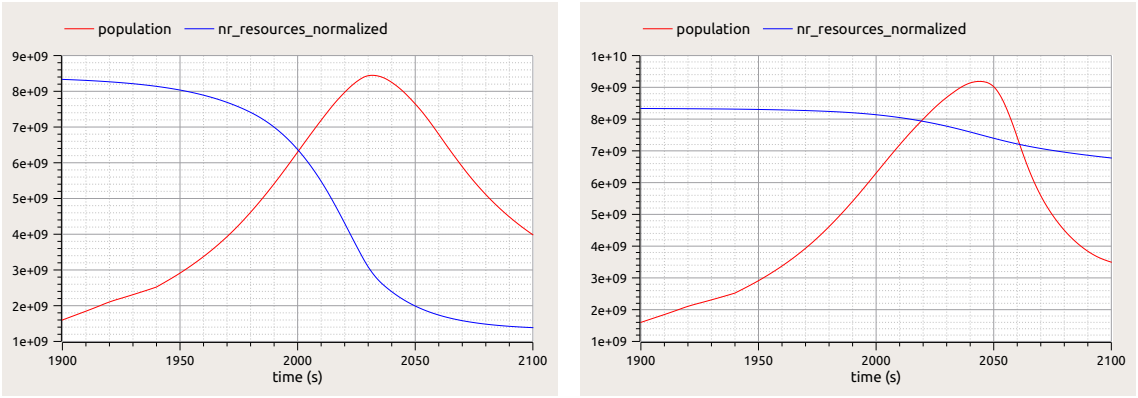


Figura 1.27: Vista gráfica del módulo `Population Dynamics` de World3



(a) Valores por defecto. *(b)* `p_nr_res_use_fact_1 = 0.1`.

Figura 1.28: Simulaciones del `Scenario 1` del modelo World3

2. RECUPERACIÓN

En esta sección describimos el proceso de recuperación del código FORTRAN77, comenzando con su digitalización, siguiendo por su puesta en funcionamiento, luego por la refactorización del código y culminando con la información necesaria para llevar a cabo corridas de la versión del MML en este lenguaje.

Para mejor ordenamiento de las diferentes versiones del MML a lo largo de este documento, proponemos el siguiente esquema de nombres:

- **MML86scan.** Escrito en FORTRAN77. Es la versión en papel de 1986 hallada, transcripta sin ninguna alteración. Incluye resultados de ejecuciones impresos en 1986.

- **MML86recup.** Escrito y compilado en FORTRAN77. Es la versión de MML86scan puesta en funcionamiento durante esta Tesis, luego del proceso de detección y corrección de errores.

- **MML20reing.** Escrito y compilado en FORTRAN90. Actualizado a 2020. Es el resultado de aplicar una reingeniería a la versión MML86recup.

- **MML20modelica.** Escrito en Modelica y diseñado en el entorno OpenModelica. Actualizado a 2020. Es el resultado de una traducción a Modelica partiendo desde MML20reing.

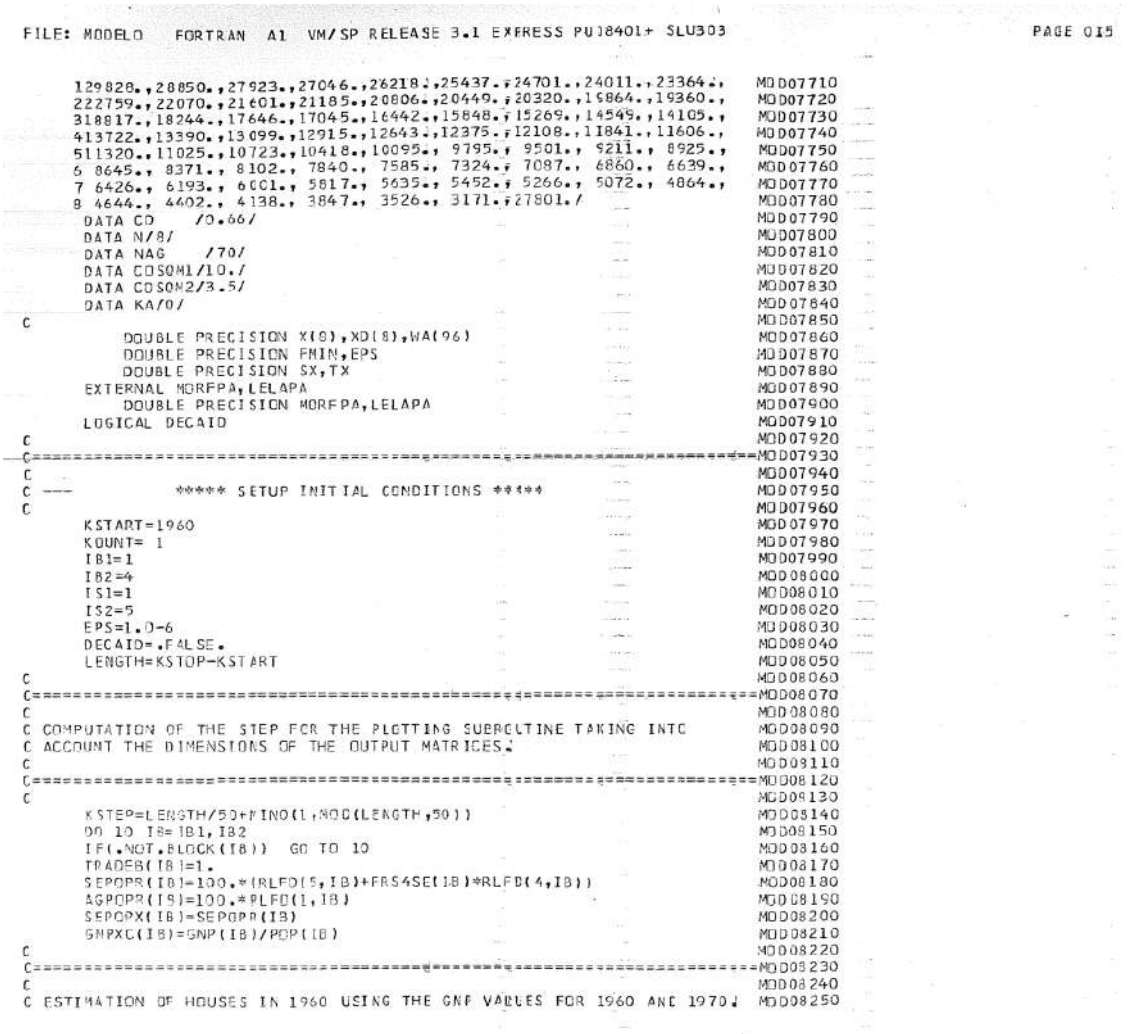
2.1. Digitalización

2.1.1. Fuentes

2.1.1.1. Código

Si bien no sobrevivieron las versiones en tarjeta perforada ni cinta magnética del código original en FORTRAN77, para el comienzo de esta Tesis se encontraron listados impresos correspondientes a 1986, de la cual mostramos un ejemplo en la Figura 2.1. Los números de las primeras líneas corresponden al final de la definición de un *DATA*, seguida de nuevo de varias definiciones de éstos y las declaraciones de algunas variables. Luego aparecen las instrucciones encargadas de inicializar las condiciones iniciales de la simulación y la página culmina con la configuración de algunos tecnicismos necesarios para la subrutina de graficación. Llamamos a esta versión en papel `MML86scan`.

Decidimos comenzar por llevar a cabo una recuperación con fines de documentación histórica, estrictamente fiel a la original, por lo que además del código y los datos incluimos en el proceso de digitalización los comentarios (líneas que comienzan con una `C`), los encabezados (`FILE MODELO ...PAGE 015`) y la numeración de líneas (`MOD07710`). Además de su relevancia documental histórica, esta *meta-data* nos fue útil en la identificación de problemas durante el proceso posterior de corrección de *bugs*, detallado en la siguiente sección.



```
FILE: MODELO   FORTRAN  A1  VM/SP RELEASE 3.1 EXPRESS PU18401+ SLU303                    PAGE 015

     129828.,28850.,27923.,27046.,26218.,25437.,24701.,24011.,23364.,   MOD07710
     222759.,22070.,21401.,21185.,20806.,20449.,20320.,19864.,19360.,   MOD07720
     318817.,18244.,17646.,17045.,16442.,15848.,15269.,14549.,14105.,   MOD07730
     413722.,13390.,13099.,12915.,12643.,12375.,12108.,11841.,11606.,   MOD07740
     511320.,11025.,10723.,10418.,10095., 9795., 9501., 9211., 8925.,   MOD07750
     6 8645., 8371., 8102., 7840., 7585., 7324., 7087., 6860., 6639.,   MOD07760
     7 6426., 6193., 6001., 5817., 5635., 5452., 5266., 5072., 4864.,   MOD07770
     8 4644., 4402., 4138., 3847., 3526., 3171.,27801./                 MOD07780
      DATA CO    /0.66/                                                 MOD07790
      DATA N/8/                                                         MOD07800
      DATA NAG    /70/                                                  MOD07810
      DATA COSOM1/10./                                                  MOD07820
      DATA COSOM2/3.5/                                                  MOD07830
      DATA KA/0/                                                        MOD07840
C                                                                       MOD07850
          DOUBLE PRECISION X(8),XD(8),WA(96)                            MOD07860
          DOUBLE PRECISION FMIN,EPS                                     MOD07870
          DOUBLE PRECISION SX,TX                                        MOD07880
      EXTERNAL MORFPA,LELAPA                                            MOD07890
          DOUBLE PRECISION MORFPA,LELAPA                                MOD07900
      LOGICAL DECAID                                                    MOD07910
C                                                                       MOD07920
C=======================================================================MOD07930
C                                                                       MOD07940
C ---          ***** SETUP INITIAL CONDITIONS *****                     MOD07950
C                                                                       MOD07960
      KSTART=1960                                                       MOD07970
      KOUNT= 1                                                          MOD07980
      IB1=1                                                             MOD07990
      IB2=4                                                             MOD08000
      IS1=1                                                             MOD08010
      IS2=5                                                             MOD08020
      EPS=1.D-6                                                         MOD08030
      DECAID=.FALSE.                                                    MOD08040
      LENGTH=KSTOP-KSTART                                               MOD08050
C                                                                       MOD08060
C=======================================================================MOD08070
C                                                                       MOD08080
C COMPUTATION OF THE STEP FOR THE PLOTTING SUBROUTINE TAKING INTO       MOD08090
C ACCOUNT THE DIMENSIONS OF THE OUTPUT MATRICES.                        MOD08100
C                                                                       MOD08110
C=======================================================================MOD08120
C                                                                       MOD08130
      KSTEP=LENGTH/50+MINO(1,MOD(LENGTH,50))                            MOD08140
      DO 10 IB=IB1,IB2                                                  MOD08150
      IF(.NOT.BLOCK(IB))  GO TO 10                                      MOD08160
      TRADEB(IB)=1.                                                     MOD08170
      SEPOPR(IB)=100.*(RLFD(5,IB)+FRS4SE(IB)*RLFD(4,IB))                MOD08180
      AGPOPR(IB)=100.*RLFD(1,IB)                                        MOD08190
      SEPOPX(IB)=SEPOPR(IB)                                             MOD08200
      GNPXC(IB)=GNP(IB)/POP(IB)                                         MOD08210
C                                                                       MOD08220
C=======================================================================MOD08230
C                                                                       MOD08240
C ESTIMATION OF HOUSES IN 1960 USING THE GNP VALUES FOR 1960 AND 1970.  MOD08250
```

Figura 2.1: Página 15 del código en versión papel.

2.1.1.2. Simulaciones

Acompañando al código, sobrevivieron resultados de simulaciones como la tabla de la Figura 2.2 y el gráfico de la Figura 2.3, ambos correspondientes a variables poblacionales de la región de Latinoamérica de una corrida estándar. Haber digitalizado las tablas como la del ejemplo permitió la sistematización del proceso de verificación y control de errores, explicado más en detalle en la Sección 2.4.

2.1.2. Procedimiento

Comenzamos la digitalización utilizando programas con funcionalidad OCR (Optical Character Recognition) para generar una versión inicial con algunas imperfecciones, como es normal con este tipo de herramientas. Un caso donde presentan dificultades es en discernir entre caracteres similares como por ejemplo `1, l, 7` e `I`. Sin embargo, el resultado de aplicar OCR tuvo un porcentaje de aciertos lo suficientemente alto como para ahorrar bastante trabajo en el comienzo.

```
***  LATINAMERICA AND THE CARIBBEAN  ***
CORRIDA STANDART

POPULATION INDICATORS: POP=P, POPR=A , EXLIFE=V, GRMOR=M, BIRTHR=N, CHMOR=B

          P            A            V            M            N            B
1960  0.2084E+09  0.2300E+01  0.5561E+02  0.1470E+02  0.3791E+02  0.1150E+03
1962  0.2197E+09  0.2681E+01  0.5695E+02  0.1191E+02  0.3731E+02  0.1203E+03
1964  0.2316E+09  0.2679E+01  0.5803E+02  0.1119E+02  0.3657E+02  0.1094E+03
1966  0.2442E+09  0.2676E+01  0.5914E+02  0.1045E+02  0.3579E+02  0.9831E+02
1968  0.2576E+09  0.2690E+01  0.6014E+02  0.9530E+01  0.3505E+02  0.8101E+02
1970  0.2715E+09  0.2660E+01  0.6105E+02  0.9049E+01  0.3431E+02  0.7724E+02
1972  0.2860E+09  0.2641E+01  0.6166E+02  0.8673E+01  0.3400E+02  0.7423E+02
1974  0.3013E+09  0.2632E+01  0.6227E+02  0.8404E+01  0.3364E+02  0.7204E+02
1976  0.3173E+09  0.2615E+01  0.6297E+02  0.8097E+01  0.3318E+02  0.6951E+02
1978  0.3340E+09  0.2595E+01  0.6370E+02  0.7771E+01  0.3267E+02  0.6669E+02
1980  0.3514E+09  0.2571E+01  0.6496E+02  0.7435E+01  0.3104E+02  0.6370E+02
1982  0.3688E+09  0.2393E+01  0.6687E+02  0.6652E+01  0.2856E+02  0.5669E+02
1984  0.3853E+09  0.2142E+01  0.6861E+02  0.5903E+01  0.2458E+02  0.4989E+02
1986  0.4002E+09  0.1851E+01  0.6954E+02  0.5309E+01  0.2180E+02  0.4426E+02
1988  0.4134E+09  0.1559E+01  0.6962E+02  0.5191E+01  0.1862E+02  0.4280E+02
1990  0.4249E+09  0.1371E+01  0.6969E+02  0.5251E+01  0.1853E+02  0.4251E+02
1992  0.4365E+09  0.1352E+01  0.6962E+02  0.5432E+01  0.1853E+02  0.4276E+02
1994  0.4482E+09  0.1333E+01  0.6967E+02  0.5605E+01  0.1853E+02  0.4276E+02
1996  0.4601E+09  0.1315E+01  0.6975E+02  0.5751E+01  0.1851E+02  0.4243E+02
1998  0.4722E+09  0.1294E+01  0.6983E+02  0.5911E+01  0.1849E+02  0.4212E+02
2000  0.4843E+09  0.1271E+01  0.6991E+02  0.6084E+01  0.1848E+02  0.4183E+02
2002  0.4965E+09  0.1241E+01  0.6998E+02  0.6272E+01  0.1847E+02  0.4155E+02
2004  0.5086E+09  0.1204E+01  0.7004E+02  0.6477E+01  0.1843E+02  0.4131E+02
2006  0.5206E+09  0.1164E+01  0.7009E+02  0.6695E+01  0.1839E+02  0.4110E+02
2008  0.5325E+09  0.1123E+01  0.7013E+02  0.6934E+01  0.1838E+02  0.4090E+02
2010  0.5442E+09  0.1087E+01  0.7015E+02  0.7208E+01  0.1837E+02  0.4080E+02
2012  0.5558E+09  0.1055E+01  0.7017E+02  0.7508E+01  0.1837E+02  0.4074E+02
2014  0.5674E+09  0.1026E+01  0.7017E+02  0.7827E+01  0.1837E+02  0.4069E+02
2016  0.5789E+09  0.9992E+00  0.7017E+02  0.8153E+01  0.1837E+02  0.4069E+02
2018  0.5903E+09  0.9745E+00  0.7020E+02  0.8467E+01  0.1836E+02  0.4061E+02
2020  0.6016E+09  0.9492E+00  0.7023E+02  0.8778E+01  0.1835E+02  0.4046E+02
2022  0.6128E+09  0.9229E+00  0.7023E+02  0.9097E+01  0.1835E+02  0.4045E+02
2024  0.6240E+09  0.8957E+00  0.7023E+02  0.9412E+01  0.1835E+02  0.4045E+02
2026  0.6349E+09  0.8681E+00  0.7024E+02  0.9722E+01  0.1835E+02  0.4042E+02
2028  0.6457E+09  0.8398E+00  0.7024E+02  0.1003E+02  0.1835E+02  0.4042E+02
2030  0.6563E+09  0.8128E+00  0.7024E+02  0.1032E+02  0.1835E+02  0.4042E+02
2032  0.6668E+09  0.7835E+00  0.7024E+02  0.1063E+02  0.1835E+02  0.4042E+02
2034  0.6770E+09  0.7549E+00  0.7024E+02  0.1093E+02  0.1835E+02  0.4042E+02
2036  0.6870E+09  0.7279E+00  0.7024E+02  0.1121E+02  0.1835E+02  0.4043E+02
2038  0.6967E+09  0.7024E+00  0.7024E+02  0.1147E+02  0.1835E+02  0.4042E+02
2040  0.7063E+09  0.6787E+00  0.7024E+02  0.1171E+02  0.1835E+02  0.4045E+02
2042  0.7157E+09  0.6601E+00  0.7024E+02  0.1191E+02  0.1835E+02  0.4045E+02
2044  0.7250E+09  0.6424E+00  0.7024E+02  0.1209E+02  0.1835E+02  0.4045E+02
2046  0.7342E+09  0.6278E+00  0.7023E+02  0.1223E+02  0.1835E+02  0.4046E+02
2048  0.7433E+09  0.6159E+00  0.7023E+02  0.1235E+02  0.1835E+02  0.4046E+02
2050  0.7524E+09  0.6073E+00  0.7023E+02  0.1243E+02  0.1836E+02  0.4047E+02
```

Figura 2.2: Tabla original de resultados de variables poblacionales.

El paso siguiente consistió en *limpiar* estas imperfecciones pero no fue una opción viable revisar hoja por hoja, caracter por caracter, las 74 hojas de la versión recuperada que incluían código, datos y comentarios. Fue por esto que decidimos revisar únicamente las páginas constituidas mayoritariamente por datos, delegando la identificación de los errores restantes primero al compilador de FORTRAN77 elegido y luego a los *scripts* de verificación de resultados.

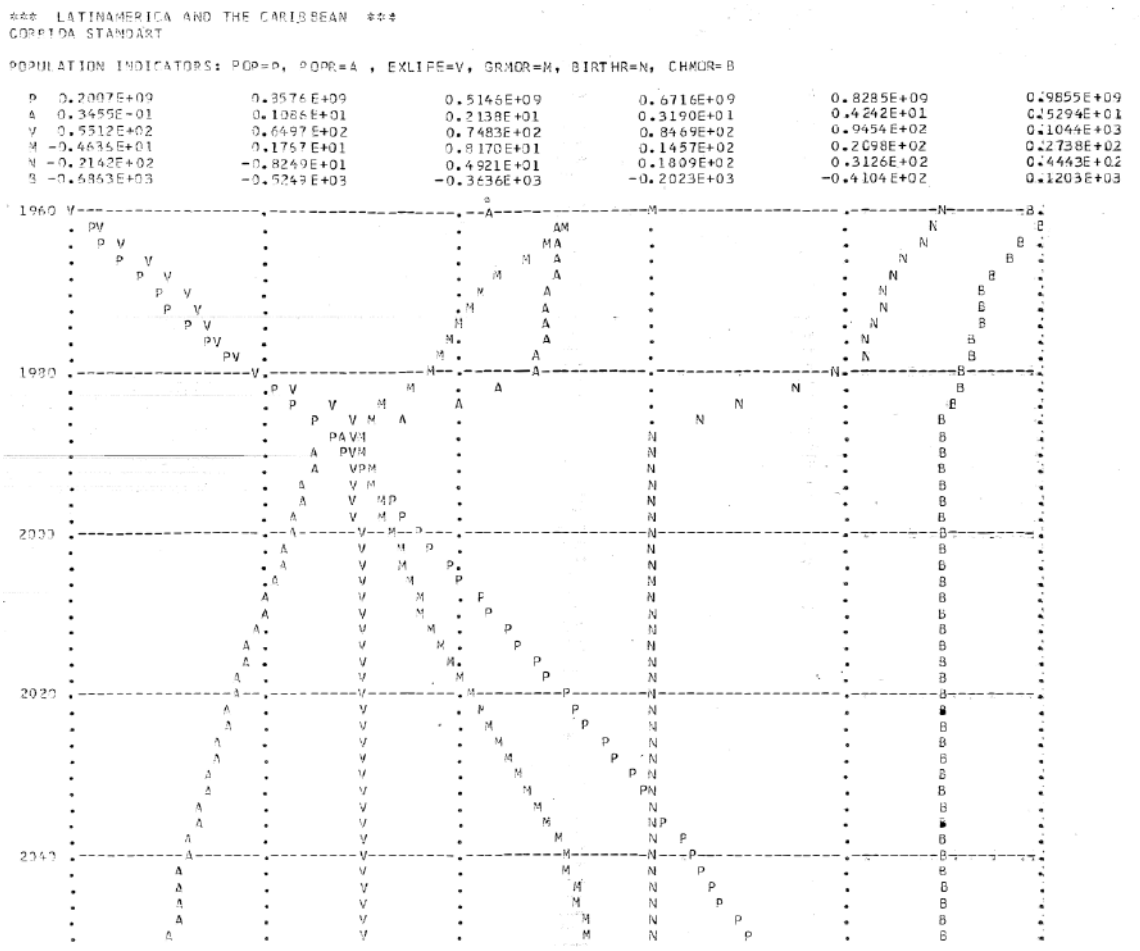


Figura 2.3: Gráfico original de resultados de variables poblacionales.

2.2. Puesta en funcionamiento

2.2.1. Compilación

Entre los compiladores de Fortran actuales nos decidimos por el compilador multiplataforma Intel Fortran Compiler [Int15] por su robustez y porque combinado con el IDE de Fortran Microsoft Visual Studio [Mic15] ofrecen un entorno de compilación, *debugging*, desarrollo y testeo muy poderosos.

Históricamente los programas en FORTRAN77 eran diseñados para un compilador en particular, que tomaba convenciones sobre el lenguaje no definidas en el estándar y podían variar considerablemente con respecto a otros compiladores. Por ejemplo, podían diferir en el tamaño de palabra[1], inicializar de manera distinta variables no declaradas, asumir equivalencias distintas entre tipos de datos, etc. El compilador elegido por nosotros es capaz de adaptarse a compiladores históricos utilizados en el pasado mediante el uso de *flags* que habilitan o deshabilitan las convenciones previamente mencionadas. Mediante procesos de prueba y error encontramos una configuración compatible con el compilador de

[1] El tamaño de palabra en ciencias de la computación se refiere a la unidad de representación de los datos usada por el procesador. Puede ser de 8 bits, 10 bits, 16bits, 32 bits, 50 bits, etc.

la época, que puede encontrarse en el Makefile del código que acompaña a este documento.

2.2.2. Detección y Corrección de Errores

Una vez obtenida una versión digital y un ambiente de compilación procedimos a solucionar *bugs* de distintos tipos, asociados tanto al error humano como al proceso de OCR durante la digitalización. Los categorizamos de la siguiente manera:

- **Errores de Compilación.** Identificados por el compilador de Fortran elegido [Int15].
- **Errores de Ejecución.** Pueden ser divisiones por 0, accesos indebidos de memoria, etc.
- **Errores de Resultados inválidos.** Identificados por la verificación, afectan los valores de los resultados de manera tal que los vuelven inválidos (por ejemplo, un valor de población negativa).

Están listados en el orden natural en el cual tienden a ser identificados. Es decir, para poder identificar errores en los resultados primero debemos poder completar las simulaciones, por lo que no debe haber errores de ejecución. A su vez, necesitamos haber purgado los errores de compilación para poder iniciar simulaciones.

Los errores de las dos primeras categorías fueron resueltos mediante el uso del compilador y funcionalidades del IDE, mientras que el arreglo de los errores de la restante requirió del proceso de verificación, explicado en la Sección 2.4. Todos fueron asistidos por el proceso de refactorización, explicado en la Sección 2.3, que llevó a cabo una limpieza y reestructuración del código que lo volvió mucho más intuitivo y organizado.

2.3. Refactorización

En ciencias de la computación, la refactorización de código [Fow99] consiste en la aplicación de cambios a su estructura con el objetivo de mejorar varios requerimientos no funcionales, como la legibilidad, extensibilidad, migrabilidad, mantenibilidad y adaptabilidad a cambios. En otras palabras, no busca cambiar *qué* es lo que hace el programa si no *cómo lo hace* y, en la mayoría de los casos, el usuario no debería poder notar la diferencia.

En nuestro caso en particular, requerimos reestructurar el código a unidades atómicas autocontenidas, llamadas módulos, que interactúen entre sí. Nuestro objetivo fue que el código sea cohesivo, siendo cada módulo responsable de sólo un aspecto de la funcionalidad deseada, y con bajo acoplamiento, siendo cada módulo lo más independiente posible del resto. De esta forma, la migración puede hacerse por partes que fuesen autocontenidas y verificables de manera incremental. Partiendo desde la versión `MML86recup`, al final de este proceso de refactorización obtuvimos la versión `MML20reing`.

Un proceso muy entrelazado con la refactorización es la ingeniería inversa [Eil05], que consiste en la deconstrucción del código para redescubrir su diseño, arquitectura y comportamiento en general. La refactorización puede asistir en el proceso de ingeniería inversa y viceversa. Por ejemplo, para saber cómo aplicar la reestructuración del código debíamos primero saber cómo estaba diseñado implícitamente el código (ingeniería inversa $\rightarrow$ refactorización) y una vez refactorizado, el código sería gradualmente más legible, lo que ayudaría al proceso de ingeniería inversa (refactorización $\rightarrow$ ingeniería inversa). En la sección 3 describiremos el proceso y presentaremos la nueva documentación obtenida.

2.4. Verificación

Otro proceso importante es el de verificación [Mar08], que consiste en comprobar que un componente es correcto respecto de alguna especificación no ambigua. En primer lugar debimos verificar que `MML86scan` fuese equivalente a `MML86recup` para luego tener asegurado que toda versión equivalente a esta segunda sería también equivalente por transitividad a la primera. Esta primera verificación consitió en comparar los resultados de las simulaciones de ambas, para `MML86scan` utilizamos los resultados digitalizados y para `MML86recup` los resultados generados por nuestras corridas. La comparación es entre los resultados de la región de Latinoamérica de la simulación estándar de ambas versiones[2].

Una vez asegurada esta primera equivalencia, las nuevas versiones (`MML20reing` y `MML20modelica`) pueden compararse directamente con `MML86recup`, de la cual se pueden producir resultados para todas las regiones y cambiando los parámetros, y, de pasar la verificación, ya tener asegurada la equivalencia con `MML86scan`. La comparación con más regiones y cambiando parámetros es más útil porque si las versiones no son compatibles entonces hay más información para *debuggear* el error.

2.4.1. Comparación entre MML86scan y MML86recup

Dependiendo del contexto y los objetivos de cada proyecto, se pueden aplicar distintas métricas de comparación para decidir el resultado de la verificación. En este trabajo elegimos comparar ambas versiones utilizando la siguiente fórmula calculando el error porcentual:

$$\text{error} = 100 * (\text{res(MML86recup)} - \text{res(MML86scan)}) \;/\; \text{res(MML86recup)} \qquad (2.1)$$

Para cada variable y cada año, si el error es negativo entonces el valor de `MML86recup` es menor que el valor de `MML86scan`, y viceversa si es positivo. Por otro lado, mientras más cercano a 0 esté el valor del error, mayor será la similitud entre ambas versiones para esa variable en ese año.

Para este trabajo decidimos concentrar nuestros esfuerzos en verificar las primeras fases: estimación 1960, regresión y proyección; dejando como trabajo a futuro el perfeccionamiento de la recuperación de la fase de optimización. Sin embargo, se pueden correr simulaciones de principio a fin incluyendo todas las fases. La precisión buscada fue que para el intervalo de años $1960 \sim 1980$ ambas versiones fuesen indistinguibles según la fórmula mencionada con $\epsilon \leq 10^{-6}$. Esta comparación fue sistematizada para que después de cada refactorización se calcule automáticamente si los cambios afectaron indebidamente el comportamiento del modelo, en el estilo de los tests de regresión [Mar08].

La verificación visual, no automática, fue llevada a cabo mediante el uso de un mapa de calor (heatmap), útil para analizar el error de cada variable en cada año, y mediante gráficos de líneas, cuyo enfoque es más general y es utilizado para analizar el comportamiento de las variables en ambas corridas. En esta sección analizamos el mapa de calor mientras que los gráficos de líneas pueden encontrarse en el Apéndice C.

En la Figura 2.4 mostramos el mapa de calor utilizando la Ecuación 2.1, coloreando en tonalidades de azul los valores negativos, en tonalidades de rojo los positivos y en blanco los ceros. La precisión elegida fue de cero puntos decimales por cuestiones de espacio de este documento. Se separan los intervalos de tiempo de $1960 \sim 1978$, previamente

[2] Recordar que sólo se pudieron recuperar resultados de estas características de la versión `MML86scan`

mencionado, 1980 $\sim$ 2000, años de optimización en la corrida estándar, y 2000 $\sim$ 2050, años extra corridos en los resultados de `MML86scan`. El índice de significados de las variables resultado puede encontrarse en el Apéndice A.

A simple vista, vemos que el color blanco domina la figura mientras que unas pocas variables presentan color. En el intervalo 1960 $\sim$ 1978 la simulación de `MML86recup` da perfecta para la precisión del gráfico, satisfaciendo los estándares buscados. Después de 1980, ésta se diferencia visiblemente de `MML86scan`, especialmente en los últimos años del intervalo 2000 $\sim$ 2050. Las variables más problemáticas son `EDUCR` (porcentaje de matriculación de la población), `GNPD(3)` (porporción de GNP asignado al sector Educación), y `RLFD(3)` (proporción de Mano de Obra asignada al sector Educación). Las tres corresponden al sector Educación, por lo que un análisis de este sector puede ser un buen punto de partida para trabajos futuros que refinan la recuperación focalizándose en la fase de optimización. En el Apéndice C graficamos estas variables y analizamos sus comportamientos.

	0-5	11-70	6-17	AL	BIRTHR	CALOR	CAPD(1)	CAPD(2)	CAPD(3)	CAPD(4)	CAPD(5)	CAPT	CHMOR	EAPOPR	EDUCR	ENROL	EXCAL	EXLIFE	FALU	FERT	GNP	GNPD(1)	GNPD(2)	GNPD(3)	GNPD(4)	GNPD(5)	GNPXC	GRMOR	HOUSER	HSEXFL	PERXFL	POP	POPR	PROT	REND	RLFD(1)	RLFD(2)	RLFD(3)	RLFD(4)	RLFD(5)	SEPOPR	TLF	TURBH	URBANR
1960	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1962	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1964	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1966	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1968	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1970	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1972	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1974	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1976	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1978	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1980	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1982	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1984	0	0	0	0	2	0	1	-2	-9	1	2	0	2	0	-3	-3	0	0	0	1	1	0	-2	-4	0	1	1	2	-3	-3	0	0	1	0	0	0	3	0	0	1	0	0	1	-2
1986	1	0	0	0	2	0	1	-1	-8	3	-3	0	2	0	-5	-4	3	0	0	1	1	-1	-2	-5	1	-1	1	2	-2	-4	-2	0	2	0	0	0	0	-2	-1	2	1	0	2	-4
1988	1	0	0	-1	2	0	-11	-3	-5	6	5	-3	0	0	-1	-1	-1	0	0	-1	2	-7	-5	-3	2	2	2	0	-5	-4	1	0	3	0	0	0	1	1	-1	3	1	0	3	-4
1990	1	0	0	-1	0	0	-11	1	10	6	-6	1	0	0	3	0	-4	0	0	-1	2	-7	0	0	4	-4	2	0	3	4	1	0	0	0	0	-1	1	-3	1	-1	1	0	0	0
1992	1	0	0	-1	0	0	-8	0	10	5	-3	0	0	0	1	0	-4	0	0	-1	1	-5	-1	0	3	-3	1	0	1	1	0	0	0	0	0	-1	1	-3	1	0	0	0	0	0
1994	0	0	0	-1	0	0	-8	0	10	4	-2	0	0	0	2	0	-5	0	0	-1	1	-5	-1	1	2	-2	1	0	0	0	0	0	0	0	0	-1	1	-2	1	-1	0	0	0	0
1996	0	0	1	-1	0	0	-8	0	10	4	-5	0	0	0	2	0	-5	0	0	-1	1	-5	-1	1	3	-4	1	0	0	0	0	0	0	0	0	-1	0	-2	1	-1	0	0	0	0
1998	0	0	1	-1	0	0	-8	0	10	3	-7	-1	0	0	-1	0	-5	0	0	-1	1	-5	-1	-1	3	-5	0	0	0	0	0	0	0	-3	0	-1	1	-6	1	-1	0	0	0	0
2000	0	0	1	-1	0	0	-8	1	10	2	-6	-1	0	0	-4	0	-5	0	0	-1	0	-4	0	-4	2	-4	0	0	0	0	0	0	0	0	0	-1	0	-11	1	0	0	0	0	0
2002	0	0	1	-1	0	0	-8	1	10	2	-9	-1	0	0	-6	0	-5	0	0	-1	0	-4	0	-6	2	-6	0	0	0	0	0	0	0	0	0	0	0	-14	1	-1	0	0	0	0
2004	0	0	0	-1	0	0	-7	2	10	2	-17	-2	0	0	-8	0	-6	0	0	-1	-1	-4	1	-7	3	-9	-1	0	0	0	0	0	0	0	0	-1	0	-17	1	-1	0	0	0	0
2006	0	0	0	-1	0	0	-7	3	10	1	-16	-3	0	0	-9	0	-6	0	0	-1	-1	-2	1	-7	2	-9	-2	0	0	0	0	0	0	0	0	1	0	-17	0	-3	-1	0	1	0
2008	0	0	0	-1	0	0	-7	4	9	2	-26	-4	0	0	-12	0	-6	0	0	-1	-2	-2	2	-9	3	-14	-3	0	0	0	0	0	0	0	0	2	2	-22	0	-3	-1	0	1	0
2010	0	0	0	-1	0	0	-7	5	-4	1	-14	-5	0	0	-21	0	-6	0	0	-1	-3	-1	3	-17	2	-8	-3	0	0	0	0	0	0	0	0	3	1	-26	0	-2	-1	0	1	0
2012	0	0	0	-1	0	0	-7	6	-5	0	-4	-6	1	0	-25	0	-7	0	0	-1	-3	-2	3	-21	1	-2	-3	0	0	0	0	0	0	0	0	2	-2	-32	0	-1	0	0	1	0
2014	0	0	0	-1	0	0	-7	6	-5	1	-10	-6	0	0	-25	0	-7	0	0	-1	-3	-2	3	-21	1	-5	-3	0	0	0	0	0	0	0	0	1	-1	-32	0	0	0	0	1	0
2016	0	0	0	-1	0	0	-8	6	-5	-1	2	-6	0	0	-26	0	-7	0	0	-1	-3	-1	3	-22	0	1	-3	0	0	0	0	0	0	0	0	3	-1	-33	0	-1	-1	0	1	0
2018	0	0	0	-1	0	0	-11	5	-3	1	-9	-5	1	0	-30	0	-7	0	0	-1	-3	-2	3	-26	1	-4	-3	0	0	0	0	0	0	0	0	4	-1	-41	-1	0	-1	0	1	0
2020	0	0	0	-1	0	0	-11	5	-3	0	-4	-5	1	0	-30	0	-8	0	0	-1	-3	-2	3	-26	1	-3	-3	0	0	0	0	0	0	0	0	4	-1	-42	-1	-1	-1	0	0	0
2022	0	0	0	-1	0	0	-11	5	-3	-1	4	-5	1	0	-32	0	-8	0	0	-1	-3	-2	3	-28	1	1	-3	0	0	0	0	0	0	0	0	4	-1	-45	-1	-2	-1	0	1	0
2024	0	0	0	-1	0	0	-11	5	2	-1	1	-4	1	0	-31	0	-9	0	0	-1	-3	-2	3	-27	1	1	-3	0	0	0	0	0	0	0	0	4	-1	-47	-1	0	-1	0	1	0
2026	0	0	0	-1	0	0	-11	4	3	-1	3	-4	1	0	-30	0	-9	0	0	-1	-2	-2	3	-27	0	2	-3	0	0	0	0	0	0	0	0	5	-1	-47	-1	0	-1	0	1	0
2028	0	0	0	-1	0	0	-12	4	5	-2	4	-4	1	0	-31	0	-10	0	0	-1	-2	-2	2	-28	0	2	-3	0	0	0	0	0	0	0	0	5	-2	-50	-1	-1	-1	0	1	0
2030	0	0	0	-1	0	0	-12	4	6	-2	5	-3	1	0	-32	0	-11	0	0	-1	-2	-2	2	-29	0	3	-2	0	0	0	0	0	0	0	0	5	-2	-52	-1	-1	-1	0	1	0
2032	0	0	0	0	0	0	-1	3	8	-2	2	-3	1	0	-31	0	-11	0	0	0	-2	2	2	-28	0	3	-2	0	0	0	0	0	0	0	0	4	-5	-52	-1	2	0	0	1	0
2034	0	0	0	0	0	0	-1	3	12	-1	-13	-2	1	0	-31	0	-11	0	0	0	-2	2	2	-28	1	-5	-2	0	0	0	0	0	1	0	0	4	-5	-57	-1	1	0	0	1	0
2036	0	0	0	0	0	0	-1	3	8	-2	0	1	1	0	-33	0	-12	0	0	0	-2	2	2	-30	0	0	-2	0	0	0	0	0	1	0	0	4	-5	-57	-1	-1	-1	0	1	0
2038	0	0	0	0	0	0	-1	3	8	-2	-2	-2	1	0	-34	0	-12	0	0	0	-1	1	1	-32	1	0	-1	0	0	0	0	0	1	0	0	2	-4	-57	0	1	0	0	1	0
2040	0	0	0	0	0	0	-1	3	11	-1	-19	-2	0	0	-34	0	-13	0	0	0	-2	1	2	-31	2	-9	-2	0	0	0	0	0	1	0	0	2	-4	-60	0	-1	0	0	1	0
2042	0	0	0	0	0	0	-1	4	7	-1	-10	-3	0	0	-37	0	-14	0	0	0	-2	1	2	-34	1	-5	-2	0	0	0	0	0	1	0	0	2	-4	-62	0	0	0	0	1	0
2044	0	0	0	0	0	0	-1	4	4	-1	-1	-4	0	0	-37	0	-15	0	0	0	-2	1	2	-34	1	0	-2	0	0	0	0	0	1	0	0	2	-4	-59	0	1	0	0	1	0
2046	0	0	0	0	0	0	-1	4	8	-1	-11	-3	1	0	-35	0	-16	0	0	0	-2	1	2	-32	1	-6	-2	0	0	0	0	0	1	0	0	2	-4	-59	0	-1	0	0	1	0
2048	0	0	0	0	0	0	-1	4	3	-2	7	-3	1	0	-38	0	-18	0	0	0	-1	1	2	-35	0	3	-2	0	0	0	0	0	1	0	0	2	-4	-59	0	0	0	0	1	0
2050	0	0	0	0	0	0	1	4	10	-1	-15	-3	1	0	-36	0	-20	0	0	0	-2	0	2	-33	2	-8	-2	0	0	0	0	0	1	0	0	-1	-5	-62	2	-2	1	0	1	0

Figura 2.4: Mapa de calor del error porcentual entre MML86scan y MML86recup, ambos para corrida estándar en Latinoamérica.

2.4.2. Comparación entre MML86recup y MML20reing

En la sección anterior comparamos la versión `MML86scan` con `MML86recup`, primera versión ejecutable que tuvimos en nuestro poder. Luego, a ésta le aplicamos iterativamente operaciones de refactorización de código para obtener una versión equivalente llamada `MML20reing`. Los resultados de estas dos versiones son indistinguibles, por lo que es `MML20reing` la que utilizamos siempre que querramos correr el modelo en Fortran y `MML86recup` es útil hoy sólo como documento histórico.

2.5. Simulación

2.5.1. Como ejecutar el modelo

Las versión `MML20reing` puede simularse ejecutando el binario o bien compilando el código fuente, ambos incluidos acompañando este documento o en un repositorio git en BitBucket.

Una tercera opción es simular de manera remota utilizando una nueva interfaz web desarrollada para esta Tesis, explicada en la Sección 4 la cual invoca a esta versión del modelo.

En la Figura 2.5 mostramos cómo compilar el código y luego correr simulaciones, que podrían ejecutarse directamente usando el ejecutable provisto. En el comando de la línea 4, `mml` es el nombre del ejecutable, `standardConfigValues.cfg` es el archivo de configuración con valores estándar y `res.txt` es el nombre del archivo donde queremos escribir el resultado. De querer correr simulaciones ad-hoc, recomendamos hacer una copia del archivo de configuración estándar y aplicarle cambios, como el archivo llamado `ejemplo.cfg` usado en el comando de la línea 6.

```
1
2 # Compilar
3 make
4 # Corrida estándar
5 ./mml standardConfigValues.cfg res.txt
6 # Corrida ejemplo
7 ./mml ejemplo.cfg res_ejemplo.tx
```

Figura 2.5: Comandos Bash para compilar y correr simulaciones del modelo de Fortran.

La capacidad de leer la configuración desde archivos estructurados fue agregada en este trabajo para posibilitar la automatización de ejecuciones, y es equivalente de manera exacta a la configuración interactiva utilizando el *Monitor* de la versión original. Para más información sobre los parámetros, incluyendo su descripción, valor mínimo, valor máximo, valor por defecto, etc. puede consultarse el Apéndice A.

2.5.2. Resultados

En el "Capítulo 9: Factibilidad material de la sociedad propuesta" del libro del modelo [HSC$^+$76, HSC$^+$04] se analizan los resultados de la simulación estándar para cada región en base a una lista de variables. Como ejercicio ilustrativo, veremos si las observaciones del libro para Latinoamérica valen para `MML86recup` (y por transitividad para `MML20reing`).

Las variables mencionadas en el libro, directa o indirectamente, son PBN por habitante (GNPXC), porcentaje de matriculación (ENROL), calorías por día por persona (CALOR), promedio de viviendas por familia (HSEXFL) y tasa de crecimiento de la población (POPR). En el modelo, se asume que se cumplieron las necesidades básicas si se cumplen los 3 requisitos siguientes:(1) el porcentaje de matriculación llegó al máximo de 98 % ,(2) las calorías por día por persona llegaron al máximo de 3000 y (3) hay por lo menos una vivienda por familia (en promedio).

En la Figura 2.6 proveemos un gráfico con los resultados de dichas variables en la nueva versión de `MML86recup` corriendo entre los años 1960 y 2050. Una de las observaciones del libro es que las necesidades básicas son satisfechas en 1990, lo que también se cumple para la nueva versión dado que en dicho año vemos como ENROL y CALOR llegan a sus respectivos topes mientras que HSEXFL es mayor a 1. Además se observó que el PBN por persona arrancaba valiendo menos que US$500 y que llegaba a valer alrededor de US$800 al momento de cumplirse las necesidades básicas mencionadas, y nuestro gráfico corrobora esta información. Finalmente, se menciona a la baja del crecimiento poblacional en relación a la subida del nivel de bienestar general y que para el final de la corrida está en camino de valer 0 %, indicando un equilibrio poblacional, siendo consistente con lo que vemos en el gráfico.

Corroboramos así que la versión utilizada en el libro es equivalente a `MML86recup` para la corrida estándary la región de Latinoamérica.

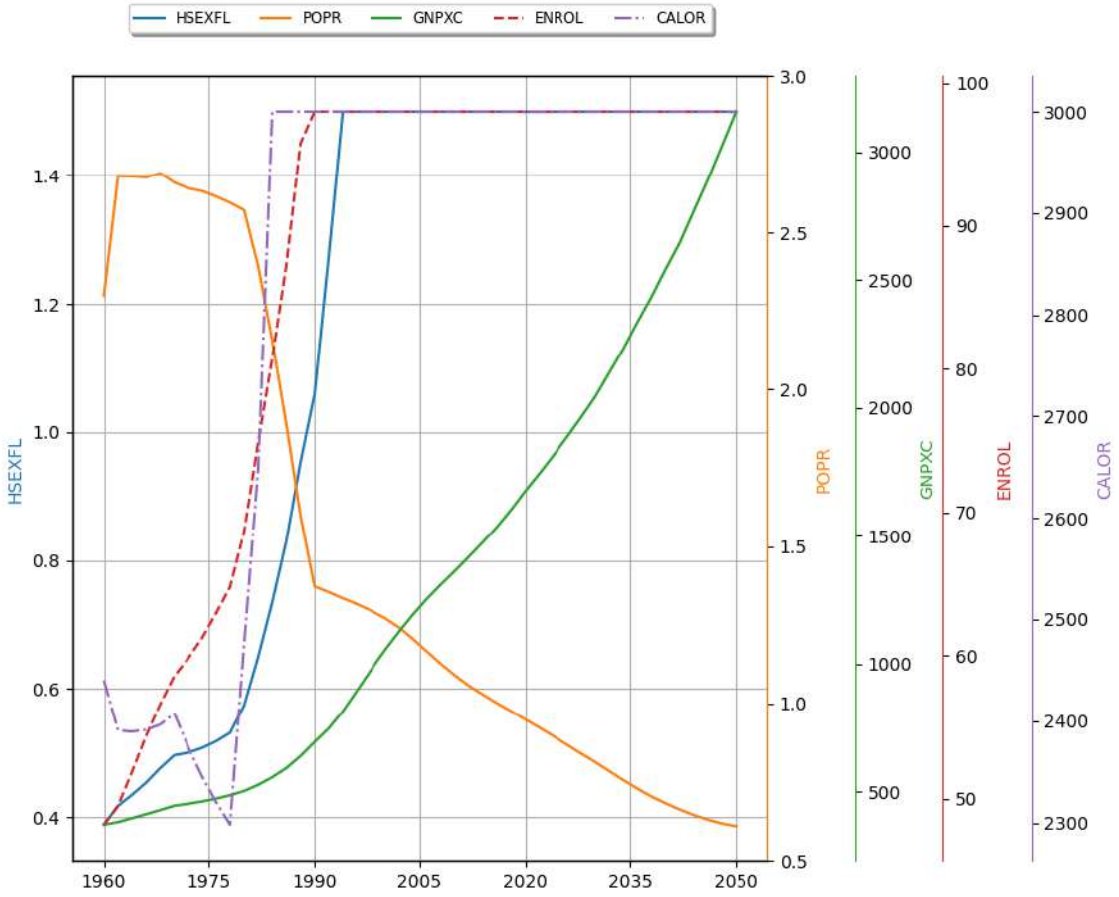


Figura 2.6: Resultados para Latinoamérica de la simulación estándar usando la versión MML86recup.

3. INGENIERÍA INVERSA

En este Capítulo presentamos al proceso de ingeniería inversa que aplicamos como parte de la recuperación, tanto en sus aspectos generales como en relación al MML, y también los resultados que produjo.

Es un proceso que consideramos necesario ya que requerimos obtener información sobre la implementación que no estaba presente en las publicaciones disponibles, con el objetivo de obtener una versión del MML que satisfaga ciertos **atributos de calidad** de software que nos proponemos considerar.

Los **atributos de calidad** [BCK12] son características deseables de un sistema que no están relacionadas con su funcionalidad. Es decir, no están enfocados en *qué* se hace si no en *qué tan bien* se hace.

En [LY13] se propone como resolver problemas de calidad caracterizados en términos de atributos de calidad y aplicando procesos de re-ingeniería.

Inspirados en este enfoque de la ingeniería de software, definimos como objetivo mejorar los siguientes atributos de calidad de la versión MML86recup del MML:

- **Legibilidad**: Pocas veces mencionada por sí sola, la legibilidad del código tiende a ser incluida implícitamente cuando se habla de otros atributos como mantenibilidad, adaptabilidad, facilidad de testeo (*testability*) y facilidad de ser corregido (*debuggability*) porque todos estos son afectados negativamente si el código presenta dificultades de lectura. Por ejemplo, un aspecto que dificulta la legibilidad es el largo forzado en 6 caracteres de los nombres de variables en FORTRAN77, que limita las opciones de nombres descriptivos al programador y puede resultar en que el estudio del código sea más laborioso que lo deseado.

- **Mantenibilidad**: Las técnicas de diseño de código que hoy son estándar, como la refactorización de código duplicado, límite en el largo de los métodos o los principios de diseño S.O.L.I.D[1]; recién estaban comenzando a ser estudiadas en la época y no fueron consideradas entre los requerimientos del MML. Esto influye en la mantenibilidad del código porque tiende a haber acoplamiento, código duplicado, difícil legibilidad, entre otros, lo que dificulta la aplicación de cambios de mantenimiento. Otro aspecto que afecta la mantenibilidad es el hecho de que *el modelo matemático está acoplado al flujo de control*, exigiendo al programador a definir el orden de ejecución de los módulos (al llamarse a un módulo sus variables de entrada ya deben haber sido inicializadas). Esto afecta la mantenibilidad porque la aplicación de cambios en los módulos podría volver incompatible al orden de ejecución elegido. Uno de nuestros objetivos es que el programador sea agnóstico de estos aspectos implementativos y que el propio lenguaje se encargue de proveer un orden de ejecución válido para los módulos matemáticos implementados (puede haber más de un orden), como ocurre con el lenguaje Modelica.

- **Portabilidad:** Como mencionamos en la Sección 2.2.1, los compiladores de la época tomaban sus sus propias convenciones para el lenguaje. Esto afecta la portabilidad

[1] *Single Responsibility Principle, Open-Closed Principle, Liskov principle, Interface Segregation Principle, Dependency Inversion Principle.* Atribuidos a Robert C. Martin [Mar08]

porque estamos acoplados al compilador utilizado, dificultando la distribución del código y obligando a quienes deseen estudiar el modelo a utilizar el mismo compilador utilizado al momento del desarrollo. Buscamos que el modelo obtenido luego de la reingeniería esté lo menos acoplado posible a una versión de compilador. Al optar por un lenguaje *estándar* de modelado matemático orientado a objetos como Modelica, cualquier compilador o entorno de modelado que adhiera al estándar podrá simular la nueva versión **MML20modelica**.

Algunos atributos de calidad que no recibieron atención fueron la *performance* (el modelo se simula bastante rápido, en el orden de segundos, y por el momento no necesitamos mejor rendimiento), la seguridad (el modelo no trata con *datos sensibles* y además planeamos distribuirlo de manera abierta) o escalabilidad (similar a performance, por ahora no es prioridad escalar en cantidad de corridas). No descartamos la posibilidad de que sean objeto de mejoras en trabajos futuros.

Enfatizamos nuevamente que esta Tesis incluye una intención documental y de preservación histórica, por lo que orientamos los esfuerzos particularmente a explicar exactamente el código recuperado, sin intención de juzgar su diseño. Más adelante aplicaremos mejoras, rediseños y traducción.

3.1. Proceso

La ingeniería inversa es el proceso en el cual un objeto hecho por humanos es desconstruido para estudiar su diseño y arquitectura. En el ámbito de las ciencias de la computación puede tener distintas aplicaciones, como para producir documentación de sistemas heredados (*legacy systems*), estudio de programas de la competencia, piratería, detección de código malicioso escondido, migración de una implementación a otra y búsqueda de vulnerabilidades (por mencionar algunas) [Eil05]. El producto de la ingeniería inversa suele ser una nueva documentación que complemente a la preexistente, además de permitir una mejor comprensión del funcionamiento del sistema bajo estudio.

En el contexto de esta Tesis, el objetivo fue la modernización del modelo, que según *Modernizing Legacy Systems* [sPL03] consiste en tres procesos:

1. La reconstrucción de una o dos descripciones lógicas de alto nivel del sistema en base a artefactos existentes
2. La transformación de las descripciones mencionadas en nuevas y mejoradas descripciones lógicas
3. El refinamiento de estas nuevas y mejoradas descripciones lógicas en el código fuente

El mismo libro refiere, además, al "modelo de herradura" (1999, Software Engineering Institute,Carnegie Mellon [BSWW99]) que reproducimos en la Figura 3.1. En este modelo se representan los tres pasos mencionados: la reconstrucción (*reconstruction*) subiendo en la parte izquierda de la herradura, la transformación (*transformation*) cruzando la parte superior y el refinamiento (*refinement*) descendiendo en la parte derecha. Se parte de un estado no deseable en la punta inferior izquierda y se arriba a un estado mejorado en la parte inferior derecha.

En nuestro caso la recuperación y posterior traducción del MML siguió en forma parcial este modelo, que fue tomado a modo de inspiración. Por ejemplo, el proceso de *reconstrucción* podría pensarse como la producción de la nueva documentación (presentada en este

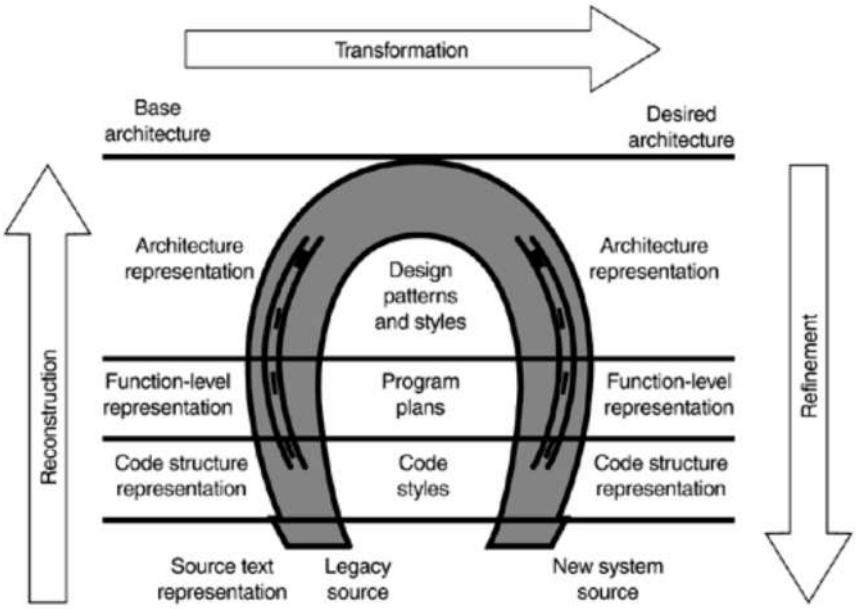


Figura 3.1: El modelo "herradura" que describe el proceso de reingeniería para *legacy systems*. (Reproducido de [BSWW99])

capítulo), la *transformación* como el cambio de paradigma de un lenguaje de propósito general como Fortran a uno de modelado como Modelica (no presentado explícitamente en la Tesis) y el *refinamiento* como la implementación final en Modelica (Sección 5).

3.2. Entendiendo el modelo

Como primer paso estudiamos el funcionamiento del modelo a *alto nivel* para luego relacionar los módulos mencionados en la documentación publicada, como la demografía o los sectores económicos, con las secciones del código encargadas de modelar sus respectivos comportamientos. En la Sección 1.1.2 recopilamos la información indispensable que, a nuestro criterio, es necesaria para entender dicho funcionamiento a alto nivel y que resulta esencial para el proceso de ingeniería inversa.

El primer método que aplicamos fue el análisis estático de código, que consiste en analizar la implementación algorítmica del MML por sí misma, es decir sin llevar a cabo ninguna simulación. Para ello, aplicamos la herramienta Understand [ST] y produjimos diagramas de flujo de control auto-generados y de bajo nivel como el de la Figura 3.2 que describe el cálculo de Capital total llevado a cabo anualmente durante las simulaciones. En la figura están identificadas las asignaciones, los ciclos y las decisiones para facilitar el seguimiento del flujo. En este caso de ejemplo, el nuevo Capital base se calcula en primer paso en base al Capital total anterior más el GNP del sector 5 (Bienes de Capital), luego se le restan iterativamente los deterioros de los capitales de todos los sectores y finalmente se le aplica un cálculo en base a la balanza comercial. En el caso de que la simulación incluya la ayuda económica, se encuentre en fase de optimización y se esté calculando para las regiones de África o Asia entonces a dicho Capital base se le suma la ayuda económica proveniente de los países desarrollados para obtener el Capital final del año.

Como complemento al análisis estático, también llevamos a cabo un análisis dinámico

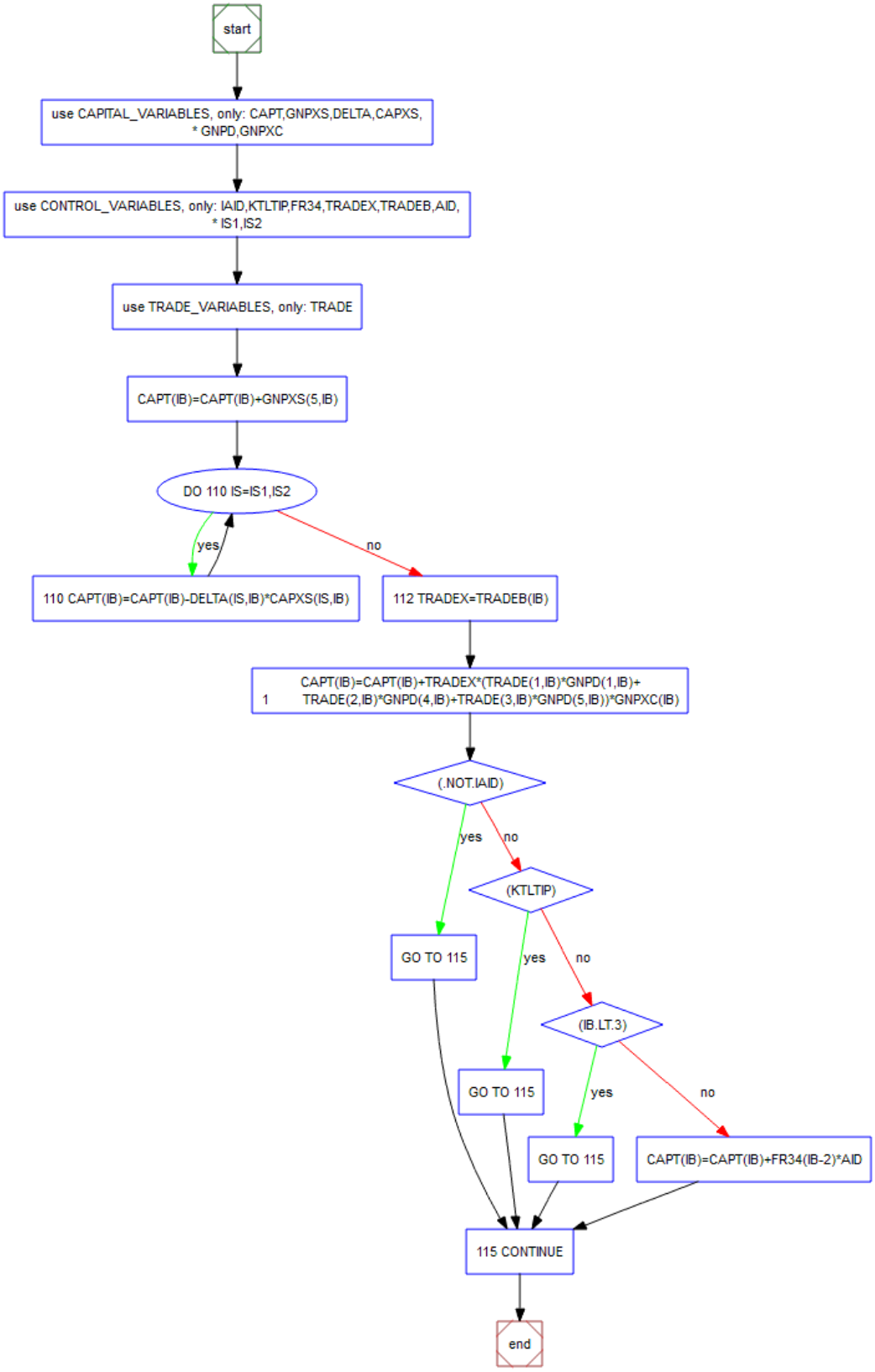


Figura 3.2: Diagrama de Flujo autogenerado para el caso del cálculo de Capital Total.

del código haciendo uso de las facilidades de *debugging* para FORTRAN77 provista por el IDE Visual Studio mencionado en la Sección 2.2.1. Este análisis consistió en el seguimiento intrucción a instrucción de simulaciones para comprender de manera interactiva, por ejemplo, qué secciones de código se ejecutan primero, cuáles les siguen, qué valores tienen asignadas las variables en un instante dado de la ejecución, etc.

Finalmente, complementamos los análisis con el apoyo del Prof. Hugo Scolnik, Director Alterno del Proyecto MML en la Fundación Bariloche en los años 70 y programador Fortran experimentado. Su memoria sobre los objetivos que se perseguían lograr con cada parte del código resultó un insumo esencial para clarificar dudas sobre conceptos de bajo nivel que no resultaron deducibles de forma evidente de la documentación publicada o de los análisis estático y dinámico de código.

3.3. Resultados

En esta sección presentamos los diagramas obtenidos en el proceso de ingeniería inversa. En la Sección 3.3.1 presentamos una línea de tiempo con las variables temporales del modelo, mientras que en las Secciones 3.3.2 y 3.3.3 presentamos los diagramas de flujo de datos y control, respectivamente.

Consideramos que los diagramas que presentamos aquí son del mismo nivel de importancia que la posterior traducción del modelo a Modelica, por su potencial utilidad en el futuro. Son lo suficientemente generales como para obviar detalles implementativos de bajo nivel (inevitables en cualquier implementación) y a la vez son lo suficientemente detallados como para servir de guía para posibles futuras reimplementaciones del modelo en nuevos lenguajes. Además, sirven para fundamentar la teoría subyacente a los distintos módulos del modelo, para que puedan ser analizados independientemente de cualquier implementación.

3.3.1. La Línea de tiempo

En la Figura 3.3 se muestran una línea de tiempo, con información visual sobre la organización temporal de las variables, y una tabla con información detallada de cada una de ellas. Pueden ser utilizadas para responder dudas sobre los valores por defecto de las variables, su mutabilidad y si son relativas o absolutas con respecto a los años. Se diferencian en que el gráfico da información colectiva y general sobre ellas mientras que la tabla da información individual y específica.

En la línea de tiempo se identifican las fases de regresión, proyección y optimización, explicadas en la Sección 1.1.2.3 representando el transcurso de simulación desde su valor por defecto en 1960, indicado por `KSTART`, hasta su valor por defecto en 2000, indicado por `KSTOP`. Los años desde 2000 en adelante son incluidos en la simulación sólo si el usuario le asigna un año posterior a `KSTOP`, y están identificados en la figura con fondo blanco.

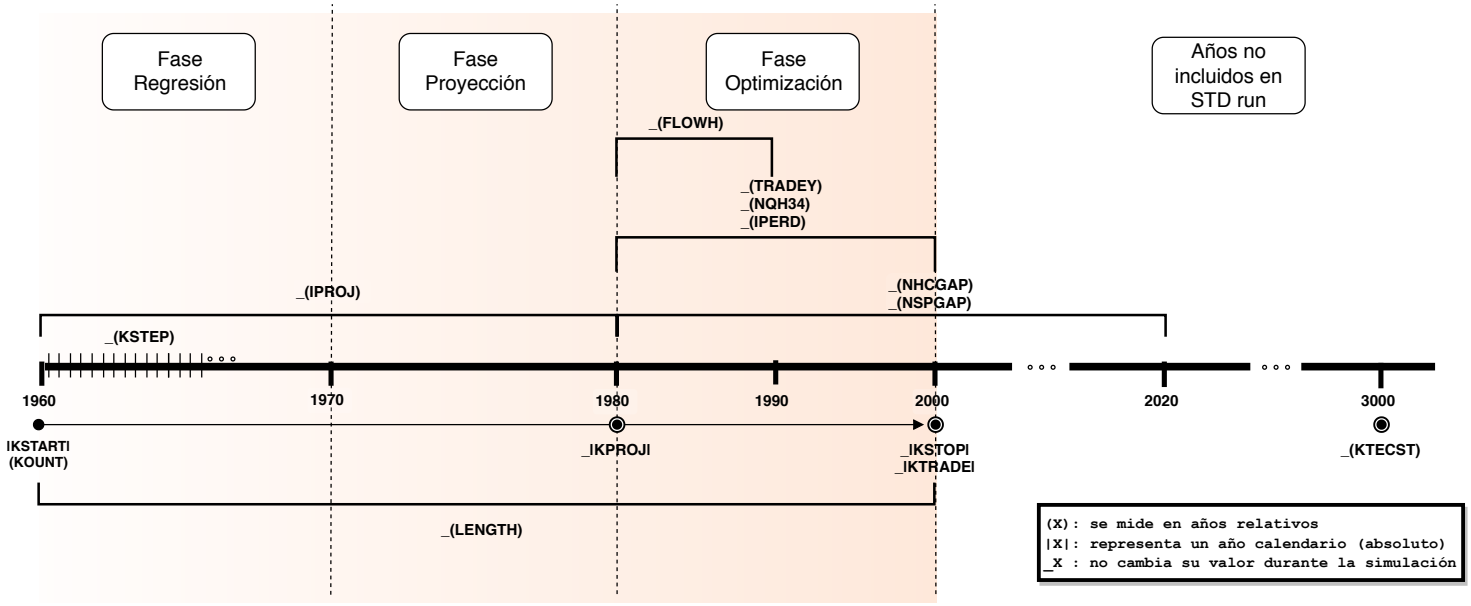
En la figura, las variables pueden estar:

- Encerradas por **paréntesis**, si refiere a años relativos.
- Encerradas por **barras verticales**, si refiere a años absolutos.
- Precedidas por un **guión bajo**, si su valor no muta durante la simulación.

Por su parte, las columnas de la tabla representan la siguiente información:

- **Variable:** nombre de la variable.
- **Valor Inicial:** valor asignado inicialmente a la variable.
  Si se especifica más de uno, puede deberse a un cambio de "significado". Por ejemplo, en su inicialización la variable. `FLOWH` representa los años hasta maximizar la ayuda económica pero durante la corrida representa su inversa multiplicativa.
- **Tipo:** puede ser relativo ("20 años después de iniciada la corrida"), absoluto (1980) o *booleano* ("Verdadero" o "Falso").
- **Mutación:** puede ser constante o mutar una o más veces durante la simulación.
- **Descripción:** descripción del significado de la variable.

Esta síntesis es resultado del estudio del código y permite comprender mejor el comportamiento temporal del sistema y el rol de cada variable. Por ejemplo, fue muy útil para interpretar el correcto funcionamiento de `KSTART`, cuyo nombre remite intuitivamente al año inicial de la simulación, que es correcto al principio, pero luego su valor muta año a año y es usada en lugares del código que a primera vista podrían parecer anti-intuitivos. Recurriendo a la línea de tiempo vemos que la variable comienza valiendo KSTART=1960, como es de esperar, pero luego va cubriendo todos los años hasta `KSTOP`, que representa el último año a simular. Luego en la tabla vemos que se incrementa en 1 por año en concordancia con el pasaje del tiempo de la simulación, representando el año actual durante la simulación y no sólo el valor inicial de ésta, como sugiere su nombre.



Variable	Valor inicial	Tipo	Mutación	Descripción
FLOWH	10 (DATA) 1/FLOWH	Rel.	Única usando DATA	1/Años hasta maximizar ayuda económica
IFPROT	20	Rel.	Const.	Años hasta que CALPRT de los no-desarrollados alcanza a Desarrollados
IPERD	20	Rel.	Const.	Años hasta que pérdidas post-cosecha de no-desarrollados alcanzan a desarrollados
IPROJ	KPROJ-KSTART+1	Rel.	Const.	Años hasta comienzo de fase optimización
KOUNT	1	Rel.	+1 por año. Afectada para graficar	Contador de años
KPROJ	1980	Abs.	Const.	Año del comienzo de fase optimización
KSET	Varía	Bool	V: llamada a PRLAWM F: otro caso	Indica si se está en fase de optimización
KSTART	1960	Abs.	+1 por año	Año actual
KSTEP	LENGTH/50 + MIN(1,MOD(LENGTH,50))	Rel.	Const.	Cada cuantos años escribir el resultado
KSTOP	2000	Abs.	Const.	Último año de la corrida
KT	KOUNT-IPROJ	Rel.	Al comienzo del año	Diferencia entre ``años corridos'' y ``años hasta optim.''
KTECST(IB)	3000 (DATA) KTECST(IB)=KTECST(IB)-KSTART+1	Rel.	Única usando DATA	Año del fin del progreso tecnológico
KTLTIP	KT > 0	Bool	Al comienzo del año	Indica si se está en fase de proyección
KTRADE	2000	Abs.	Const.	Año del equilibrio de balanza comercial
LENGTH	KSTOP-KSTART+1	Rel.	Const.	Duración de la corrida
NHCGAP	40	Rel.	Const.	Parámetro usado para igualar costos viviendas de África y Asia con Desarrollados*[1]
NQH34(IB)	0 (Desarrollados y L.A.) 20 (África y Asia)	Rel.	Const.	Años hasta que África y Asia alcanzan estándares de viviendas de L.A.
NSPGAP	40	Rel.	Const.	Similar NHCGAP pero espacio p. persona en vez de costos de viviendas.
TRADEY	KTRADE - KPROJ	Rel.	Const.	Años hasta el equilibrio de la balanza comercial

*[1] Durante optimización, años desde que las necesidades básicas de África y Asia fueron satisfechas hasta que los costos de sus viviendas alcanzan los valores de los países desarrollados.

Figura 3.3: Línea de tiempo de variables relacionadas con años.

3.3.2. Flujo de Datos

Los diagramas de flujo de datos son utilizados para representar el flujo de información entre distintos componentes de un sistema indicando sus entradas (*inputs*), salidas (*outputs*) y las operaciones llevadas a cabo entre ambas.

Categorizamos a los diagramas entre los de **alto nivel**, donde la notación es simple y sólo se indica el flujo de datos entre módulos, y los de **bajo nivel**, en los cuales la notación es más compleja y se indican comportamientos internos de cada módulo.

3.3.2.1. Vista de Alto Nivel

El diagrama de la Figura 3.4 presenta el flujo de datos entre los distintos sub-modelos de manera similar al incluido en la Sección 1.1.2.7 pero ahora detallando qué variables componen dicho flujo. Puede ser utilizado para el estudio de dependencias entre módulos analizando el origen y destino del flujo por cada variable. Su sintaxis es simple y no requiere de una especificación detallada, a diferencia del diagrama de bajo nivel presentado en la sección siguiente.

Por cada módulo se indican las variables de entrada y de salida. Tomemos por ejemplo el módulo `Costos` del grupo `Cálculo previo a sectores`. Tiene sólo una variable de entrada, llamada `Viviendas por Familia`, cuya flecha indica que proviene del módulo `Viviendas`. Sus variables de salida son el costo del sector Alimentación, el costo del sector Viviendas, el espacio por persona y la colección de los costos de los sectores de alimentación, viviendas y educación.

Como vimos en el ejemplo anterior, las salidas y entradas pueden corresponder a colecciones de variables de varios sectores. Si se trata de 4 sectores o menos, se los enumera como fue el caso en dicho ejemplo. Si la colección incluye a los 5 sectores, se indica que es `por sector` como en las entradas `Subtotales Capital por sector` y `Subtotales Mano de Obra por sector` del módulo `GNP`.

Analizando el diagrama, podemos notar que sólo dos variables son entradas de varios módulos. La variable `Viviendas por familia` del módulo `Viviendas` se usa en los módulos de `Costos` y `Demografía` mientras que `Población` del módulo `Demografía` se usa en `Alimentación`, `Viviendas`, `Educación`, `GNP` y `Mano de Obra`. Para reducir la cantidad de flechas y facilitar la lectura del gráfico, utilizamos unas barras negras que deben interpretarse como equivalentes a que estén todas las flechas uniendo el origen y el destino.

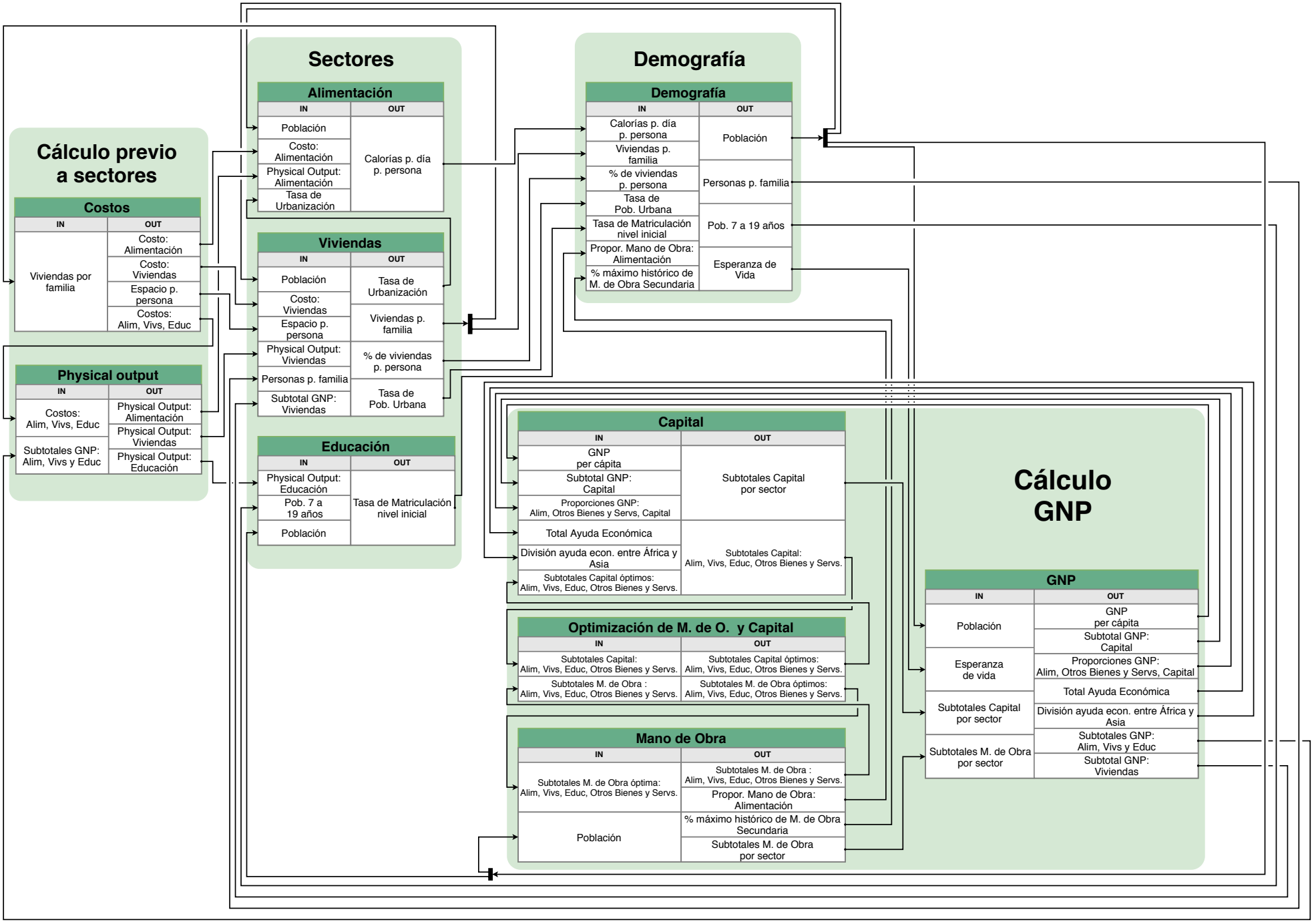


Figura 3.4: Flujo de datos entre los módulos del MML.

Reglas

1. Nos enfocamos en el paso general sin explicitar el paso inicial. Es decir, no anotamos inicializaciones en las estructuras DATA o estimaciones de 1960. Es por esto que hay variables que en el diagrama aparentan ser usadas sin antes ser definidas; por ejemplo, en el módulo `Capital` las variables de la balanza comercial de los sectores Alimentación, Otros Bienes y Servicios y Capital no son inicializadas explícitamente.
2. Omitimos detalles que dependen mucho de la implementación. Por ejemplo, no incluimos las transformaciones trigonométricas que se llevan a cabo antes y después de la optimización de la función LELAPA.
3. Omitimos detalles de las funciones que se usan sólo en optimización (por ejemplo FOODKP, que se usa en LELAPA y en MORFPA pero no en proyección). Esto obedece a que el alcance de esta Tesis se enfoca en la migración de la fase de proyección, dejando la fase de optimización como próximo paso.
4. Colores: Utilizamos color **AZUL** en regresión, color **VERDE** (principalmente) en proyección y color **ROJO** (principalmente) en optimización. El resto se ejecuta en todos los años.

Figura 3.5: Reglas del diagrama de bajo nivel.

3.3.2.2. Vista de Bajo nivel

A diferencia de la vista de alto nivel, los diagramas presentados en esta sección son más específicos y necesitan de una notación más poderosa. Para facilitar su diseño e interpretación desarrollamos un sistema propio de reglas y elementos visuales.

En este tipo de diagramas no existe la noción de *momento de ejecución* sino que todos los flujos están activos en simultáneo. Es por esto que en la notación estricta no existen operaciones de control, tal como decisiones del estilo *if*.

Sin embargo, decidimos dotar a nuestra notación ad-hoc con la capacidad de expresar también algunos conceptos mínimos de control, para facilitar la lectura e interpretación de los diagramas. La alternativa de ser fieles a una notación estrictamente de flujo de datos resultaría en una explosión de sub-diagramas todos muy similares entre sí, diferenciándose sólo en unas pocas operaciones. Por ejemplo, debería haber un diagrama por cada fase del modelo, cuando éstas se diferencian poco entre sí.

Por cada módulo del diagrama presentado en la sección anterior aquí se expone su flujo interno de datos. Tener en cuenta que el módulo de Demografía fue dividido en dos partes por limitaciones de espacio de este documento. Decidimos hacer una parte enfocada en el cálculo poblacional y otra enfocada en el cálculo de la Esperanza de Vida al Nacer y la tasa de nacimientos.

En la Figura 3.5 presentamos las reglas y convenciones que definimos para el desarrollo de los diagramas.

Las leyendas, consistentes en todos los diagramas, se presentan en las Figuras 3.6 y 3.7. En ellas se describen las representaciones de los componentes mediante ejemplos simples y autocontenidos.

Los diagramas de los módulos tienen anotaciones para aclarar cuestiones implementativas o para resaltar casos en los que hicimos abuso de notación. La referencia de dichas anotaciones puede encontrarse en la Figura 3.8.

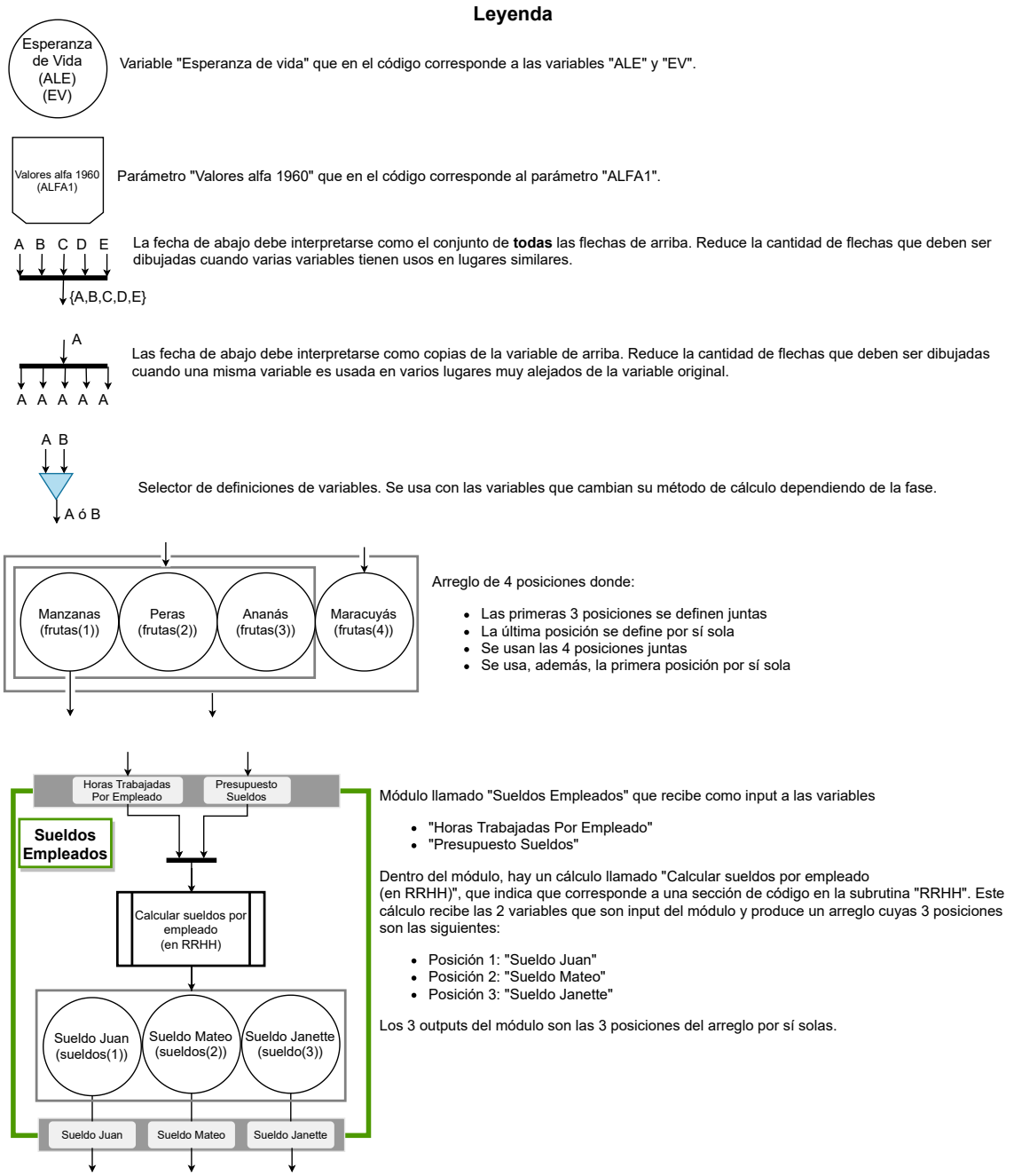
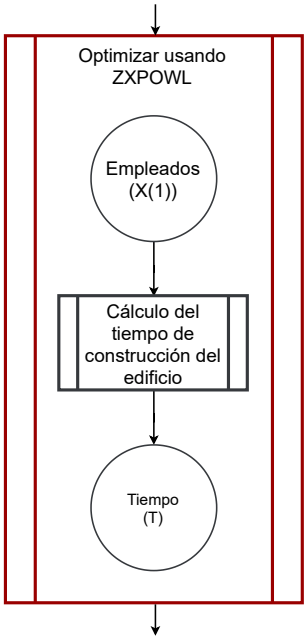


Figura 3.6: Leyenda del diagrama: 1/2.



Optimización de una función **f** modificando la entrada **x** para minimizar el resultado **f(x)** utilizando el optimizador **ZXPOWL**.

- **x:** variable de "Empleados" que en código se llama "X(1)"
- **f:** función de "Cálculo del tiempo de construcción del edificio"
- **f(x):** variable Tiempo que en el código se llama "T"
- **ZXPOWL:** nombre del optimizador que implementa el algoritmo de optimización de Michael J. D. Powell.

En este caso, se busca la la mano de obra que minimice el tiempo de construcción de un edificio.

En el MML, la optimización se usa sólo con las funciones de MORFPA en Alimentación y LELAPA para calcular los valores óptimos de capital y mano de obra. En ambos casos, f(x) es ignorado una vez terminada la optimización, dándole importancia sólo a x y a los cálculos colaterales usados por las funciones.

Figura 3.7: Leyenda del diagrama: 2/2

Notas

1. La función de E. de V. recibe a AGX como entrada, que corresponde a Mano de Obra del sector de alimentación. Sin embargo, en la fase de optimización la Mano de Obra de dicho sector podría cambiar sin actualizar a la variable AGX. Esto puede generar inconsistencias.
2. Las proporciones de Mano de Obra son proyectadas durante la fase de proyección. En el primer año de la fase de optimización se proyectan pero sólo como valor inicial temporal. El valor final del cada año de esta fase, incluyendo el primero, es siempre obtenido calculando las proporciones en base a los subtotales de los sectores.
3. Se tiene en cuenta la Esperanza de Vida al Nacer de las regiones de África y Asia.
4. MORFPA también llama a FOODKP. No fue desglosado para reducir la complejidad del diagrama.
5. Se definen los subtotales de M. de O. en las últimas 4 posiciones del vector X antes de llamar a LELAPA/PRLAWM, sin actualizar explícitamente los valores de RLF hasta después de volver de dicha función. Para ese cálculo se usa POP(IB) (población año anterior) y no POPX (población año corriente).
6. El primer año de optimización se fijan además costos iniciales de viviendas sólo para África y Asia.
7. Aunque las partes internas de LELAPA ya aparecen individualmente en el diagrama, fue necesario incluir también a la función por su cuenta, dado que sus llamadas durante la fase de optimización afectan directamente a la Mano de Obra y el Capital de los primeros 4 sectores.
8. Segundo año de optimización en adelante.
9. Proyección y primer año de la optimización.
10. La ayuda económica se tiene en cuenta sólo durante la fase de optimización y sólo en corridas con el flag IAID habilitado. Se le resta GNP a los Países Desarrollados que se les suma en forma de Capital a África y Asia. Se fracciona la ayuda en base a una proporción calculada usando las expectativas de vida y las poblaciones totales de ambas regiones.
11. Entre 1961 y 1969 se interpola en base a los valores de 1960 (ALFA1) y 1970 (ALFA2). Desde 1970 en adelante se hace una extrapolación constante con los valores de 1970.
12. Los costos de Educación y Viviendas son más *permisivos* inicialmente en América Latina, África y Asia y se los incrementa gradualmente hasta alcanzar los niveles de los países desarrollados.
13. Incrementar los costos de viviendas en África y Asia pone mucha presión sobre sus respectivas economías. Se comienza con costos menos *dignos* pero más *realistas*.
14. Todos los años de optimización.
15. Primer año de optimización para obtener valores base de la fase.
16. Cuando se corre con ayuda económica, primero se calculan el GNP y Capital normalmente y luego se les extrae y adiciona, respectivamente, la cantidad correspondiente a la ayuda dependiendo de la región.

Figura 3.8: Referencias de las notas presentes en el diagrama.

El primer diagrama corresponde al módulo del sector de Alimentación, presentado en la Figura 3.9. Está dividido entre los sub-módulos "Sub-sectores", donde se hacen los cálculos de las variables de alto nivel como la ganadería, pesca y agricultura; y "Producción", enfocado en las de bajo nivel como calorías y proteínas producidas. El submódulo de "Sub-sectores" no ahonda en los cálculos internos de las funciones FOODKP y de MORFPA, que son invocadas únicamente en la fase de optimización, pero sí podemos ver que en el caso de MORFPA ésta se optimiza mediante ZXPOWL. Finalmente, vemos como la entrada de "physical output" de este sector se utiliza para calcular la producción de calorías en la fase de proyección mientras que en optimización se utilizan en cambio los resultados obtenidos por el submódulo de "Sub-sectores".

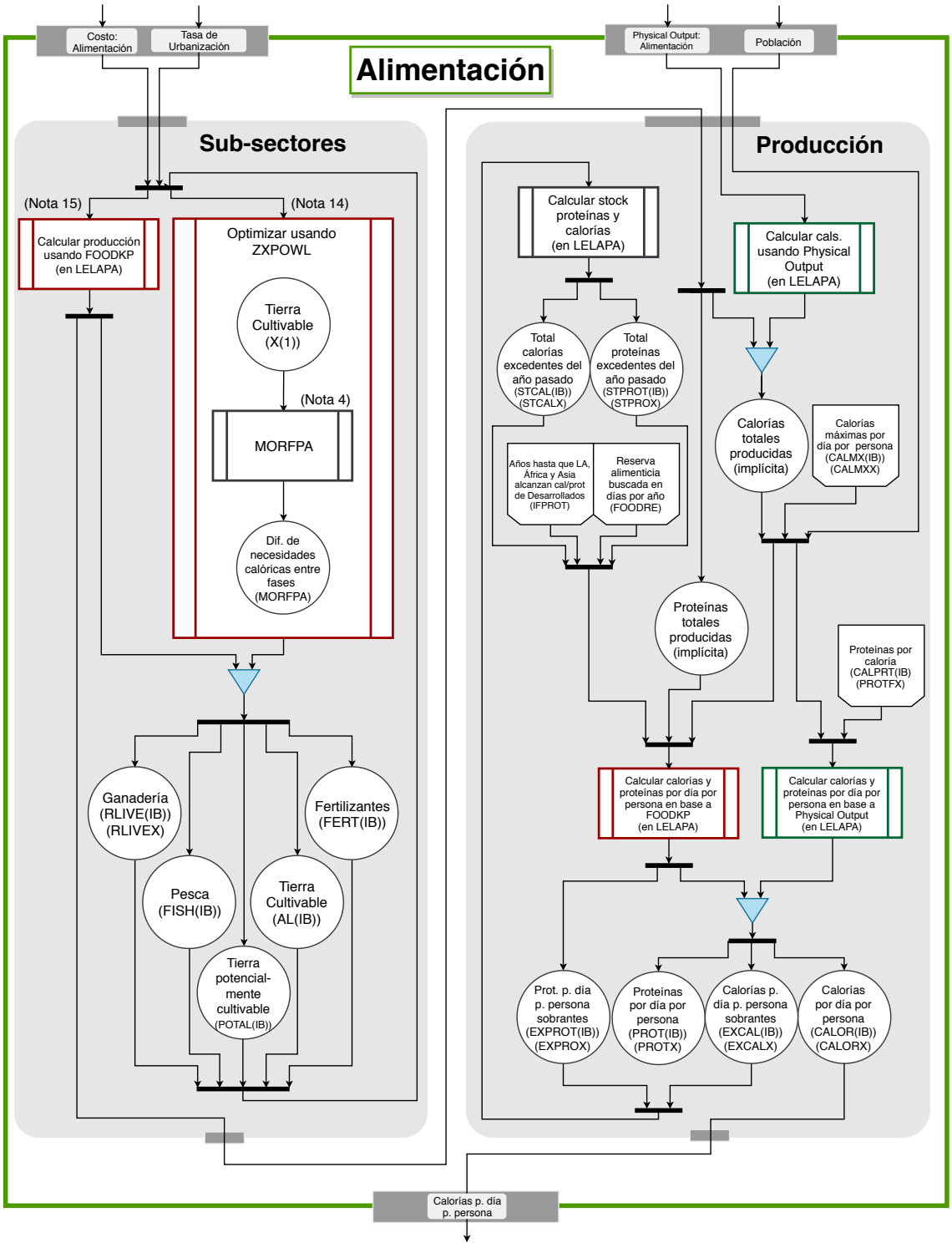


Figura 3.9: Módulo Sector Alimentación.

Continuamos con el sector Viviendas, presentado en la Figura 3.10, que está dividido en "Construcción" y "Urbanización". Para disminuir la cantidad de flechas, los flujos de las variables "Personas por familia" y "Población" se dividen en varias oportunidades. El

uso del "physical output" es similar al del sector de Alimentación, siendo utilizado sólo en la fase de proyección para ser reemplazado por un cálculo en base al GNP y costos del sector durante la fase de optimización.

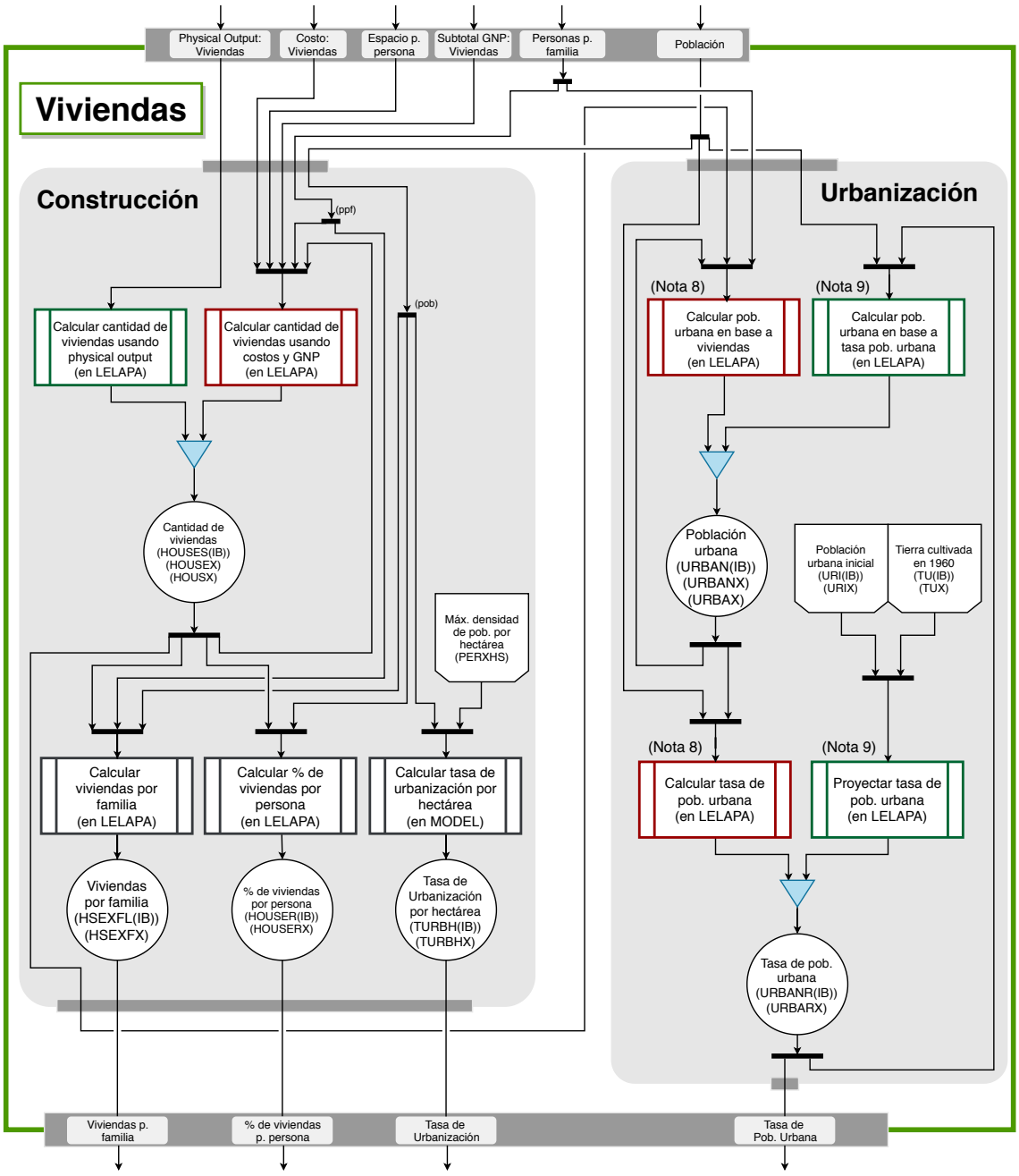


Figura 3.10: Módulo Viviendas.

En la Figura 3.11 se encuentra el módulo de Educación, el más simple de los primeros 3 sectores. Está compuesto sólo por los cálculos de matriculación de nivel inicial y del porcentaje de matriculación de la población. Esta última variable no es utilizada en otros módulos, por lo que no es incluida como "output" de este sector, pero sí es incluida

en los resultados de las simulaciones originales. El sector de Educación es el único que calcula su producción en base al "physical output" durante las fases de Proyección y Optimización, mientras que Alimentación y Viviendas (como vimos anteriormente) sólo lo utilizan durante la Proyección mientras que hacen uso de otras variables internas durante la Optimización.

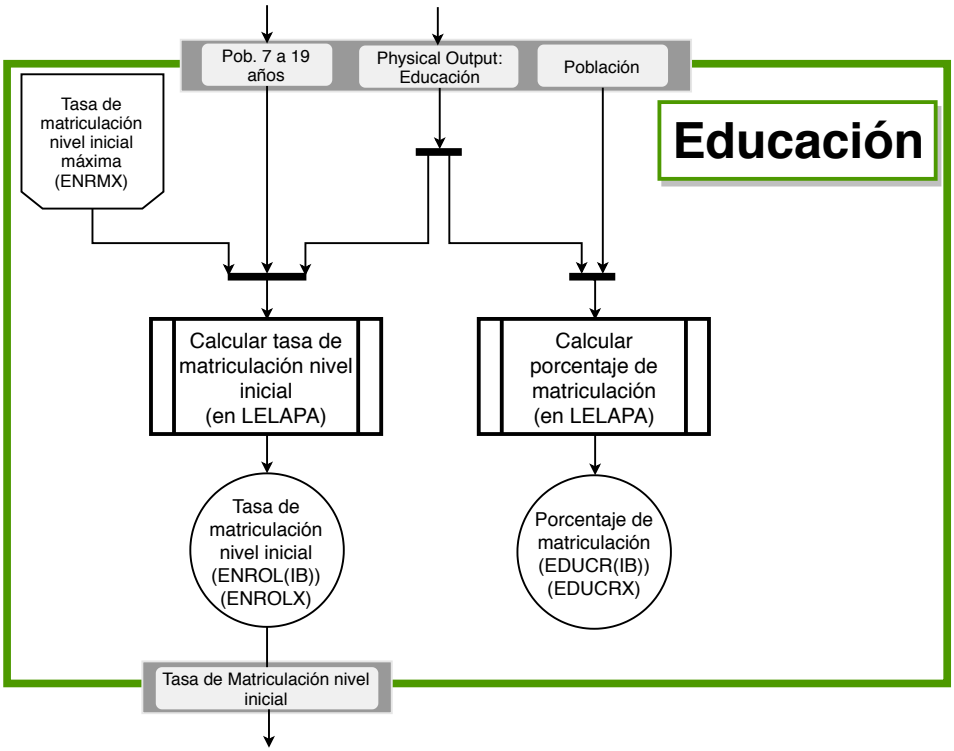


Figura 3.11: Módulo Educación.

Por limitaciones de espacio de este documento, el módulo de Demografía fue dividido en dos partes. En la Figura 3.12 se incluyen los cálculos poblacionales, divididos entre "Tasas de supervivencia", "Tasas de Fertilidad", "Población" y "Personas por familia". Estos cálculos son independientes de las fases y juntos calculan los únicos 3 *outputs* de este submódulo. El otro submódulo de Demografía se presenta en la Figura 3.13, encargado de los cálculos de la Esperanza de Vida al Nacer y la tasa de nacimientos. Es interesante ver como ambos cálculos se influyen mutuamente, lo que indica que dichas variables están fuertemente relacionadas.

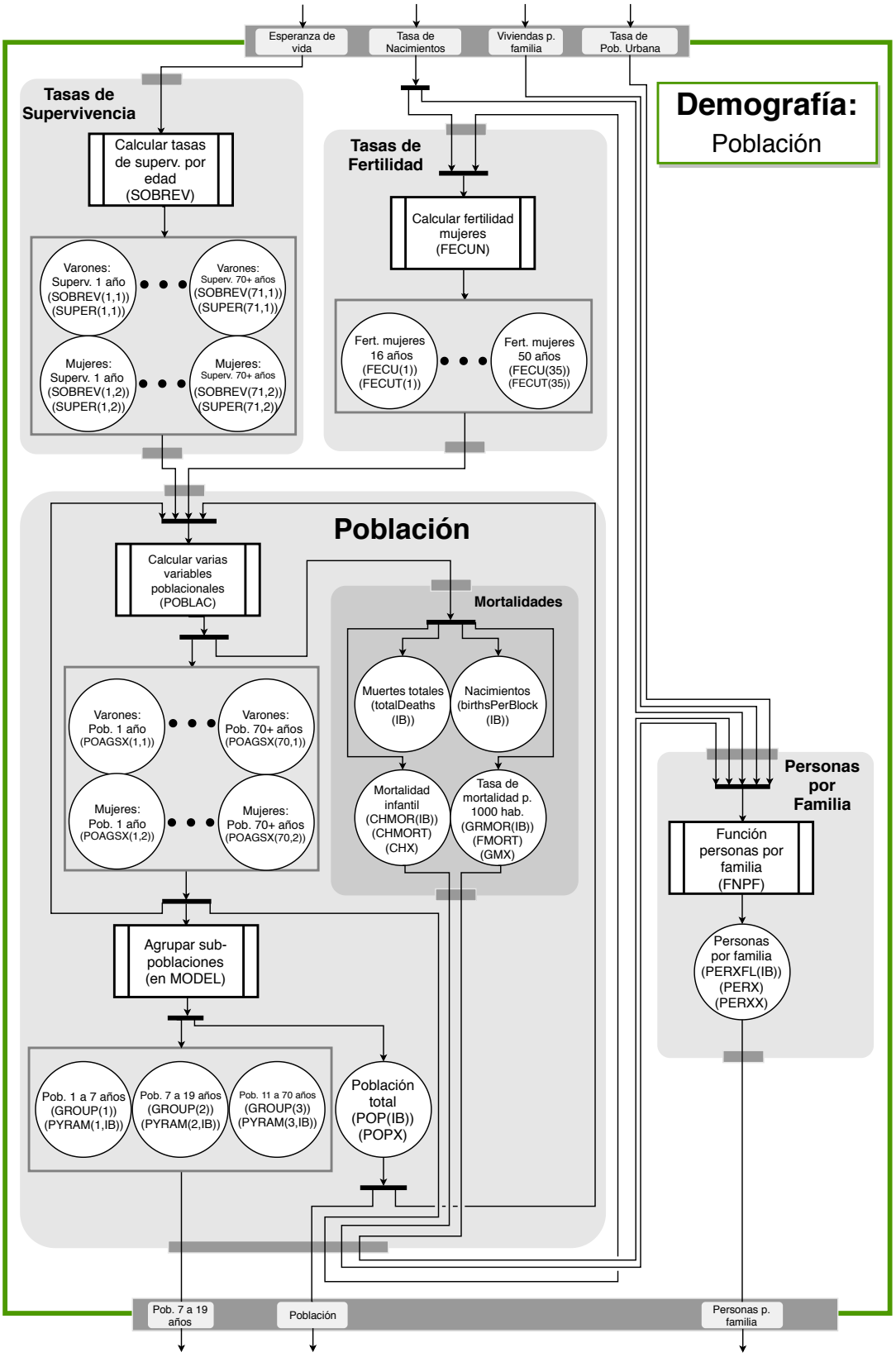


Figura 3.12: Sub-módulo de Demografía: Población.

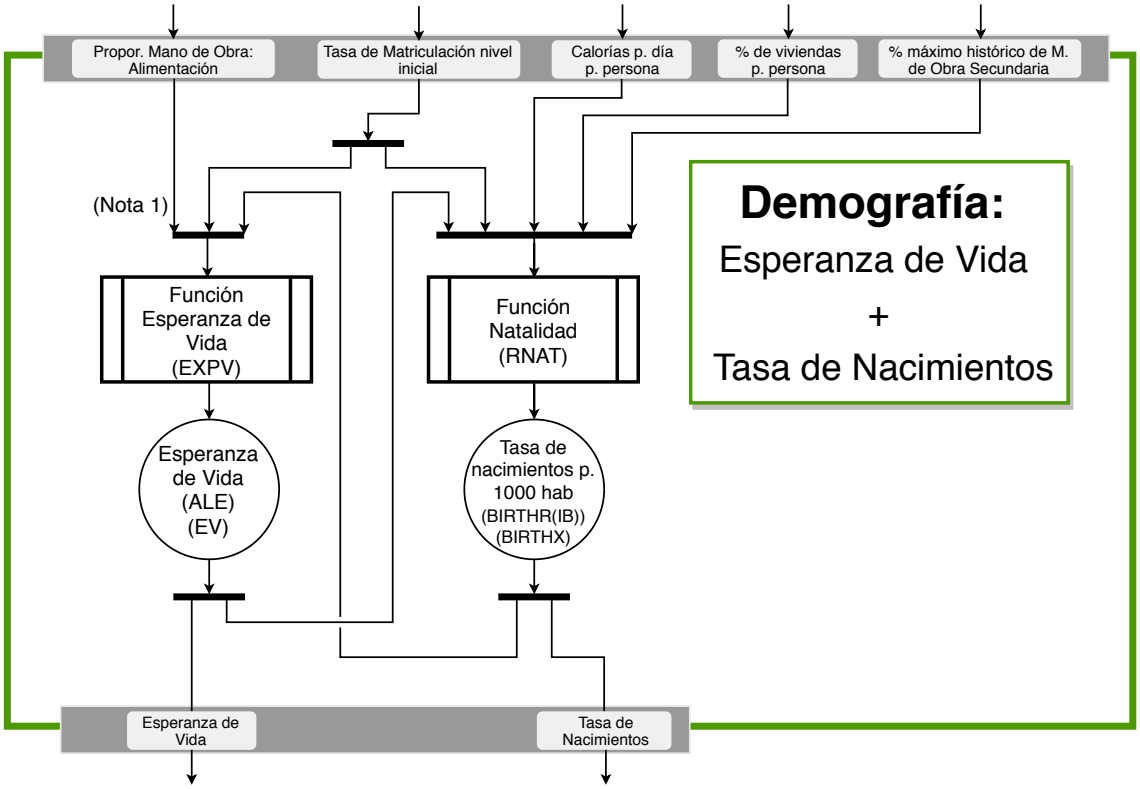


Figura 3.13: Sub-módulo de Demografía: Esperanza de Vida al Nacer y Tasa de Natalidad.

El "physical output", cuyo flujo de datos internos es presentado en la Figura 3.14, se calcula para los sectores de Alimentación, Viviendas y Educación combinando los subtotales de GNP y costos respectivos de cada sector. Como vimos anteriormente, es utilizado en los módulos de dichos sectores para el cálculo de la producción anual, siendo el de Educación el único que lo utiliza en todas las fases.

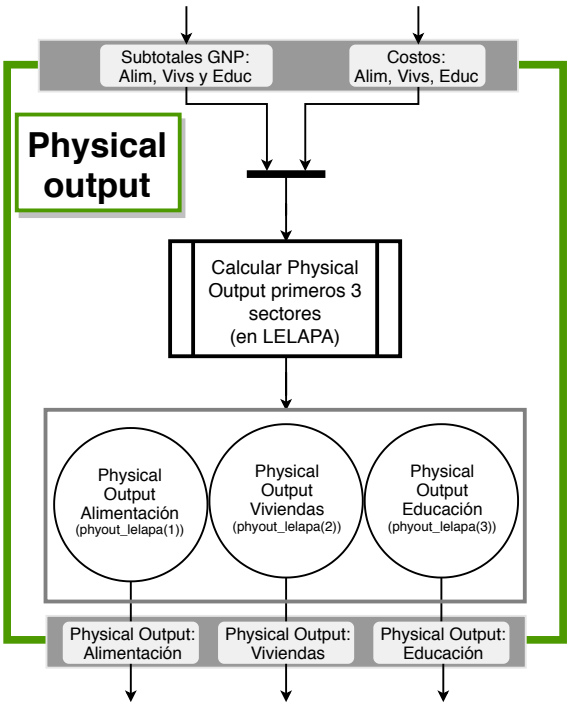


Figura 3.14: Módulo Physical Output.

El módulo encargado del cálculo de los costos sectoriales, presentado en la Figura 3.15, basa la mayoría de sus operaciones en parámetros definidos antes de la corrida, precisando de sólo el *input* de "Viviendas por familia". Esto da la pauta que sus *outputs*, los costos de los primeros tres sectores más el espacio por persona, son bastante independientes del resto de las variables del modelo. Además, vemos que todos los cálculos ocurren sólo durante la fase de optimización[2] por lo que las variables mantienen un valor constante en las fases anteriores.

[2] Todos los cálculos, denotados como cajas con barras verticales a los costados, tienen los bordes rojos, indicando que se ejecutan sólo durante la fase de optimización.

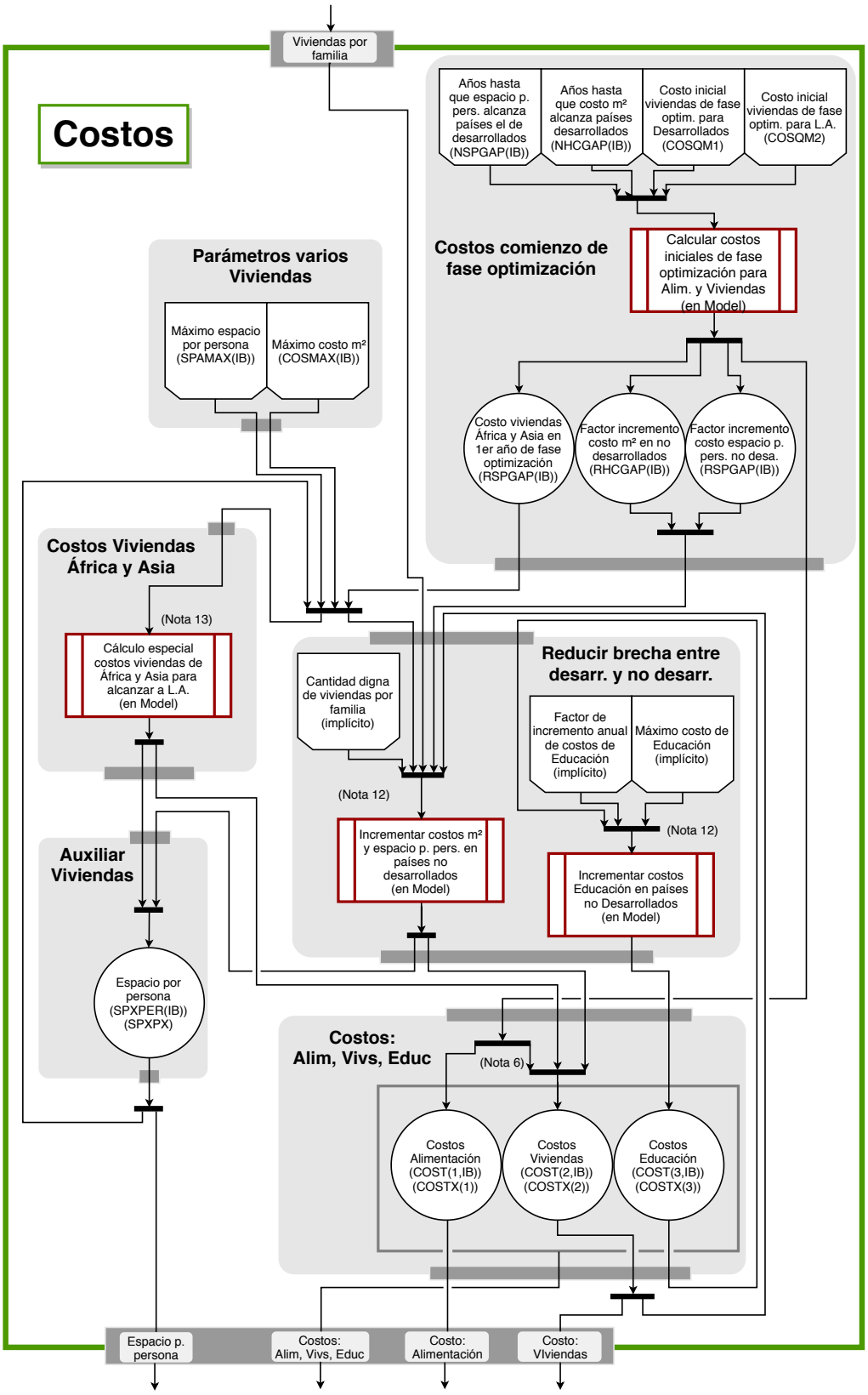


Figura 3.15: Módulo Costos.

En la Figura 3.16 se presenta el módulo de la Mano de Obra, en el cual se calculan sus subtotales y proporciones por sector, además del porcentaje máximo histórico de Mano de Obra secundaria. Durante la fase de proyección, los subtotales de Mano de Obra de los primeros 4 sectores se calculan en base a la población y sus proporciones respectivas, mientras que las éstas últimas son proyectadas en base a los valores del año anterior. En cambio, durante la fase de optimización los subtotales se reciben como entrada, provenientes de la optimización de la función LELAPA, y las proporciones se calculan en base a ellos. Podemos notar que durante ambas fases la Mano de Obra del sector "Bienes de Capital" es producto de los otros cuatro subtotales. Además, el cálculo de la proporción de la población económicamente activa ocurre sólo durante la fase de proyección, por lo que la variable se mantendrá en un valor constante durante toda la fase de optimización.

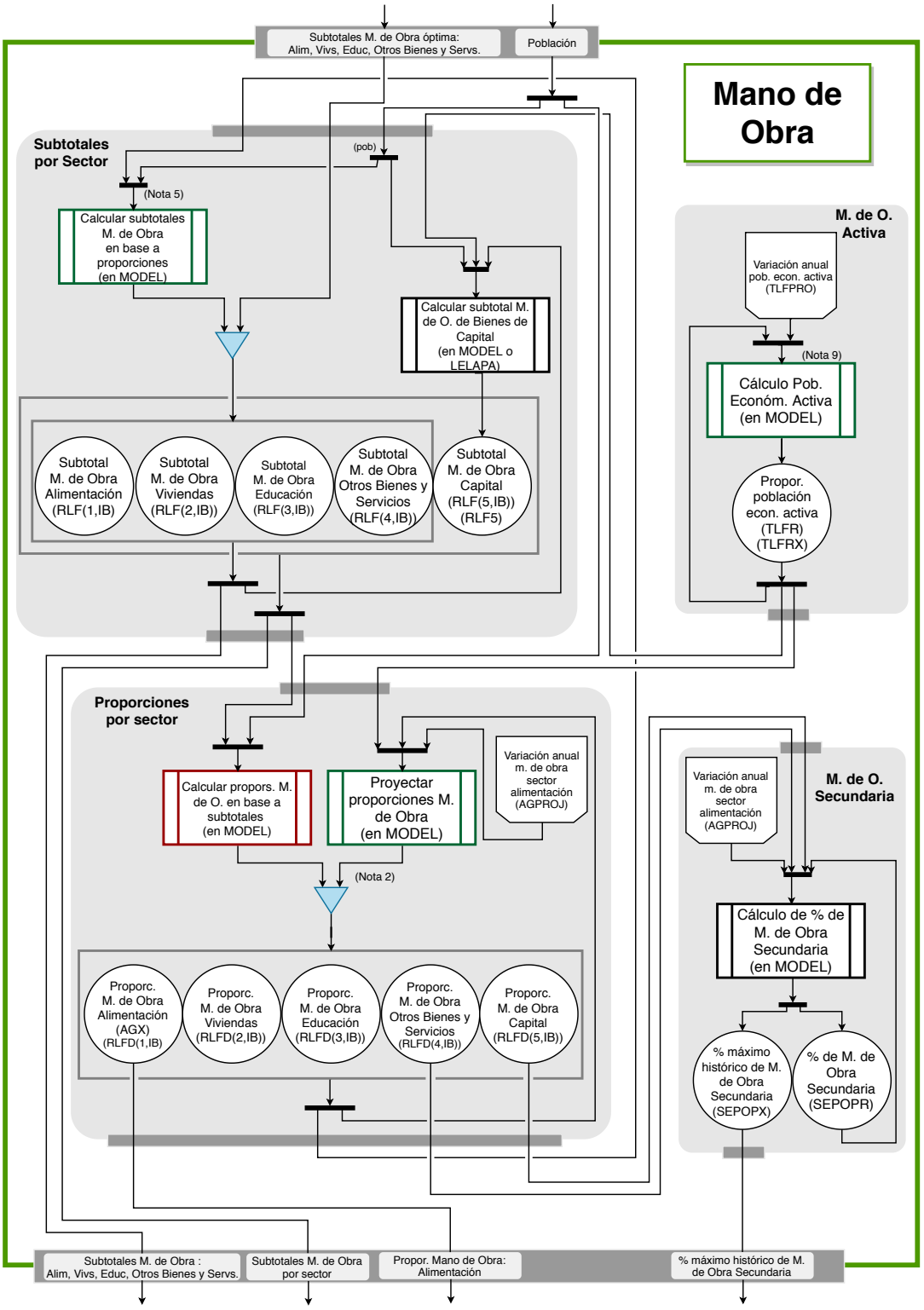


Figura 3.16: Módulo Mano de Obra.

Analizando el módulo de Capital, presentado en la Figura 3.17, vemos cómo el cálculo de los subtotales se asemeja bastante al del módulo de Mano de Obra, calculándose los

primeros 4 sectores en base a proporciones durante la fase de proyección mientras que durante la fase de optimización se les asignan los valores óptimos de Capital provenientes de la optimización de LELAPA. Con respecto al subtotal del sector 5 vemos que también se calcula en base a los de los otros cuatro. Para el cálculo de las variables sectoriales mencionadas, se precisa calcular la balanza comercial y el deterioro de Capital, presentados con sus respectivos cálculos, variables y parámetros. Se incluye como entrada la ayuda económica, que será sumada durante la fase de optimización al Capital de las regiones de África y Asia en las corridas que la lleven a cabo. Notar que las proporciones están agrupadas en un único componente para ahorrar espacio abusándonos del hecho de que sólo son utilizadas colectivamente y no individualmente.

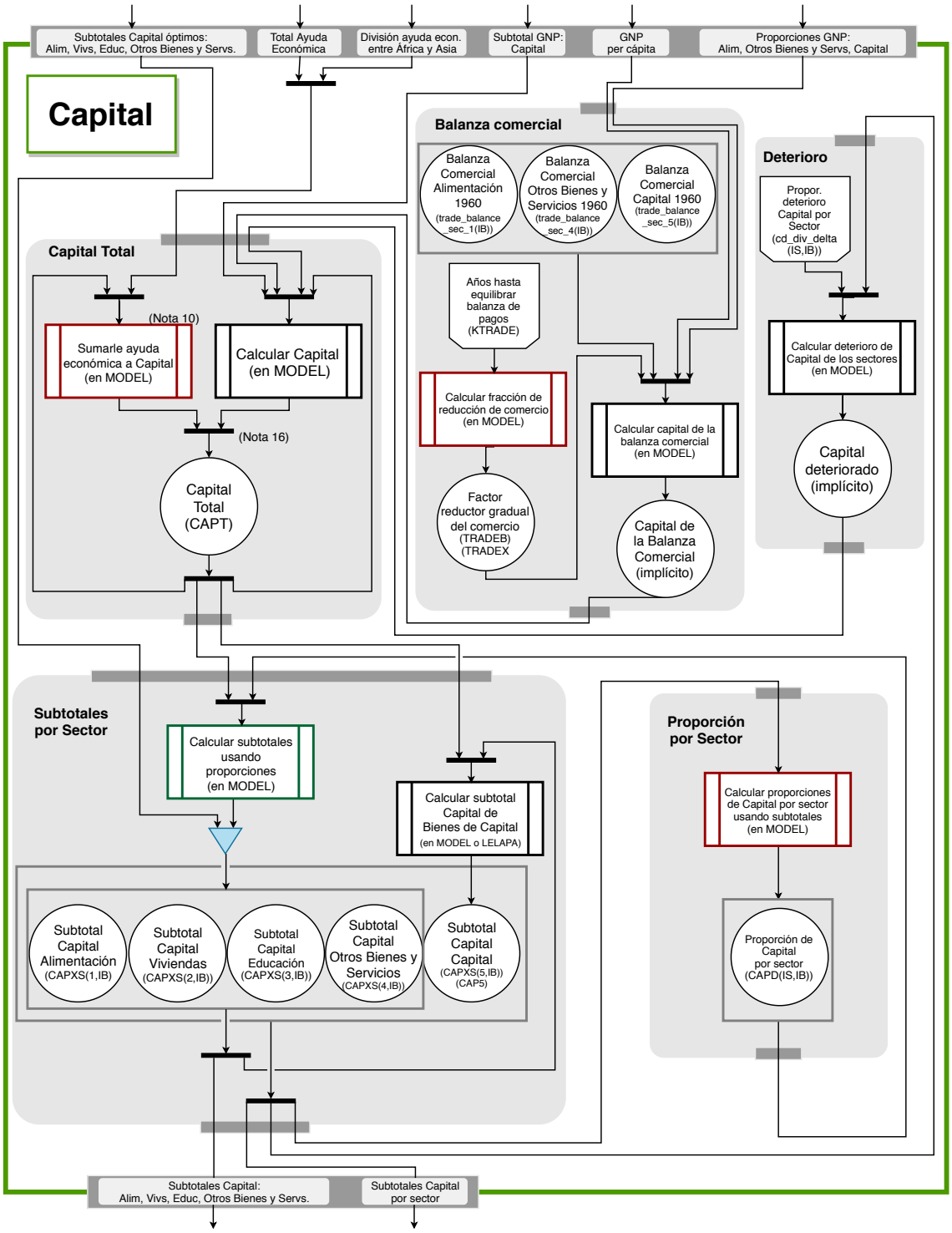


Figura 3.17: Módulo Capital.

El flujo de datos interno del módulo de GNP está presentado en la Figura 3.18, el cual está compuesto por varios submódulos. De manera similar a la Mano de Obra y Capital, se tienen submódulos encargados del cálculo de subtotales y proporciones por sector, diferenciándose de ellos en que los subtotales dependen de un cálculo llevado a cabo

sólo en la fase de regresión. Dicho cálculo corresponde a la proyección de los *alfas* usados en la función de producción de Cobb-Douglas [CD28] y es el único cálculo de todo el modelo llevado a cabo en dicha fase y en ninguna otra. En el lado derecho del módulo incluimos al cálculo de la ayuda económica, perteneciente a la fase de optimización sólo para la región Países Desarrollados y sólo en las corridas que la hayan activado. Corresponde a una extracción del GNP de dicha región que será sumada al Capital de África y Asia, como vimos en el diagrama del flujo de datos de Capital.

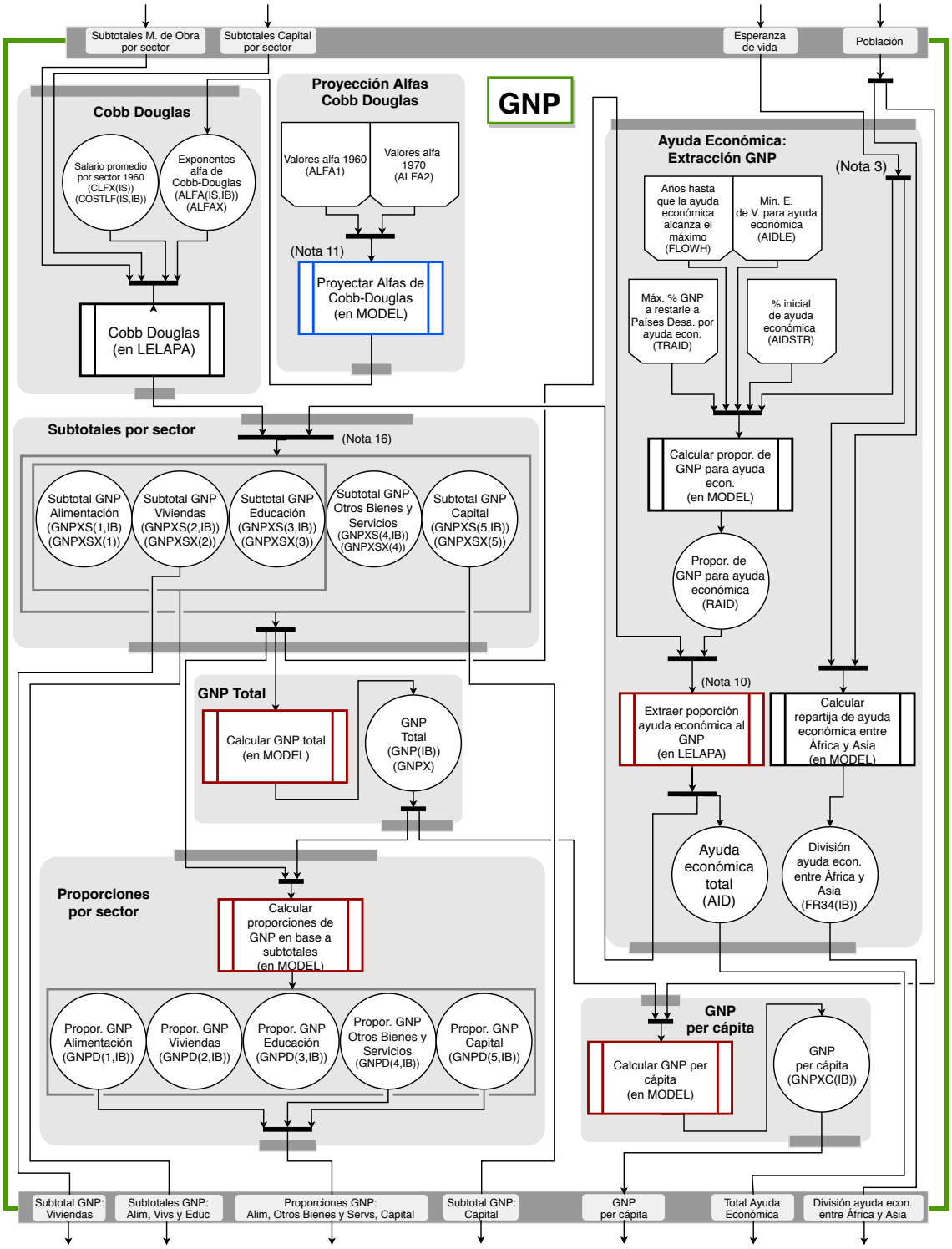


Figura 3.18: Módulo producto nacional bruto (GNP).

El último flujo de datos corresponde a la optimización de los subtotales de Mano de Obra y Capital por sector, que puede verse en la Figura 3.19. Como lo explica la nota referenciada en el diagrama, su presencia es un abuso de notación, dado que dicha

optimización se lleva a cabo sobre la función LELAPA, cuyos cálculos internos ya fueron incluidos en varios de los diagramas anteriores. Con esto queremos decir que todos los cálculos que incluyen un "(LELAPA)" en su nombre son tenidos en cuenta en este módulo de manera "repetida". Sin embargo, a fines prácticos esto no influye en la interpretación del flujo de datos colectivo, dado que podemos tomarlo como un cálculo aislado, pudiendo así representar los cálculos de los subtotales *óptimos* por sector del Capital y Mano de Obra. Esta optimización es la columna vertebral de la fase que lleva su nombre y de ella dependen, directa o indirectamente, una gran cantidad de las variables del modelo.

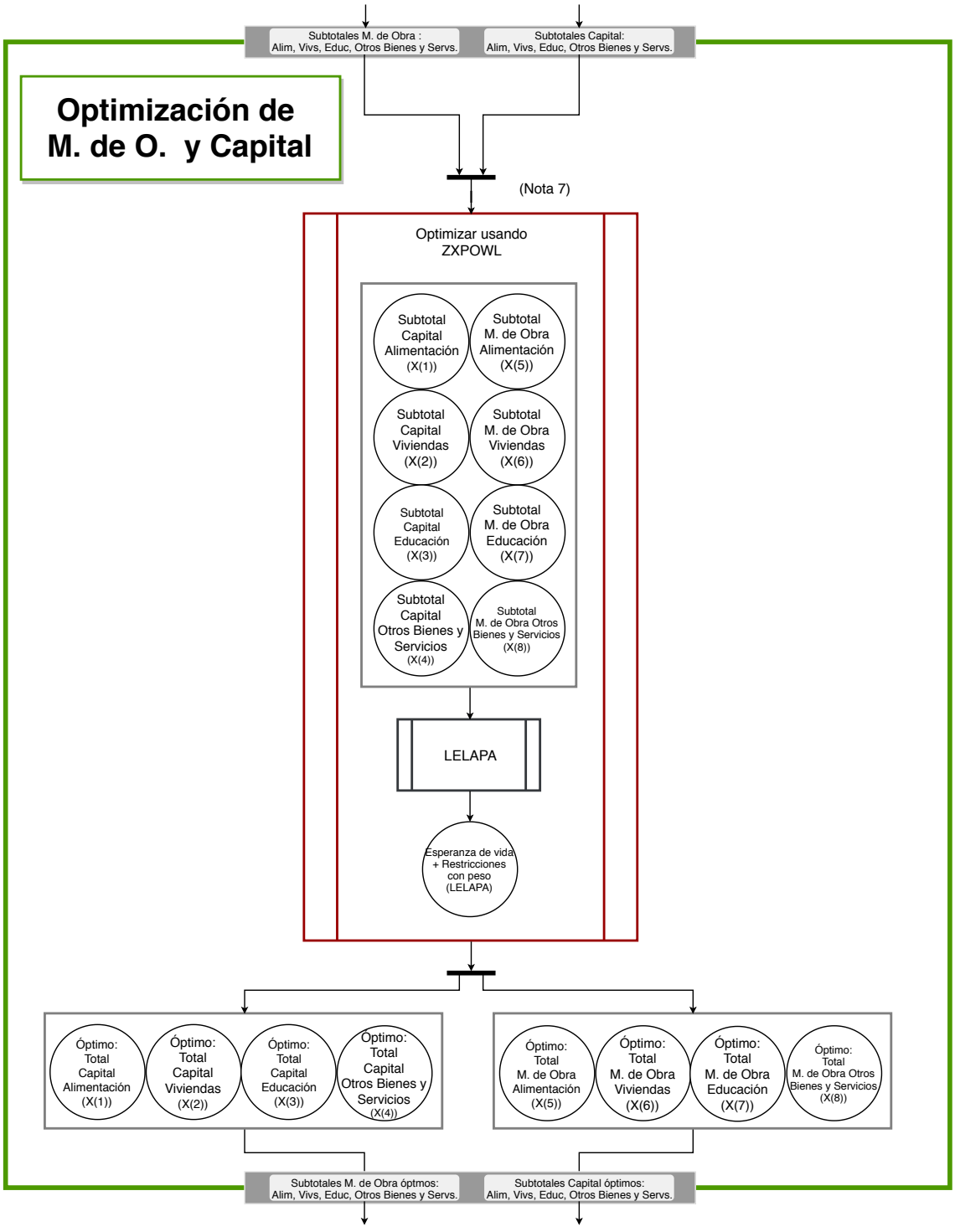


Figura 3.19: Módulo Optimización Mano de Obra y Capital.

Algunos de los flujos de datos acá presentados son similares a los flujos de datos del libro original [HSC+76, HSC+04], por lo que incluimos los diagramas del libro en el Apéndice B.1 e indicamos como se relacionan con los de esta sección.

3.3.3. Flujo de Control

Los diagramas de flujo de control son utilizados para modelar algoritmos o procesos, poniendo énfasis en el orden cronológico de las operaciones. Se diferencian de los diagramas de datos en que tienen un principio, un fin, aceptan operaciones de control y que los datos tienden a no ser explicitados, pero de serlo toman un rol secundario.

El Handbook [SFL+77] incluye un diagrama del flujo principal del modelo, que facilitamos en el Apéndice B.2 y sirvió de inspiración para la versión detallada que presentamos aquí. A diferencia del Handbook, aquí se provee también un diagrama de la función LELAPA.

3.3.3.1. Flujo Principal

El primer diagrama de flujo de control que presentaremos corresponde al del Flujo Principal, que describe el proceso de simulación de principio a fin, incluyendo la `Inicialización`, `Simulación` y `Escritura de resultados`.

En la versión original del modelo, un *Monitor* era el encargado de interactuar con el usuario para inicializar los parámetros de simulación a su gusto. Como vemos en el módulo de `Inicialización`, nuestra versión también ofrece la posibilidad de leer la configuración desde un archivo, lo que facilita el *testing* y la experimentación al poder automatizar la corrida de simulaciones. En base a la configuración y los datos del modelo, se inicializan algunas variables que serán usadas para estimar las restantes, teniendo como objetivo que todas las variables hayan sido inicializadas con sus valores respectivos para el año 1960.

Una vez concluida la inicialización, el control pasa al módulo de `Simulación` que itera las regiones dos veces por año, la primera para hacer unos cálculos iniciales del año y la segunda para llevar a cabo el cálculo principal de la fase de Optimización o de Proyección, según corresponda acorde al año. Una vez que se hayan iterado todas las regiones dos veces para todos los años, concluye el módulo y se procede a la escritura a disco de los resultados.

El primer ciclo de regiones mencionado en el párrafo anterior, llamado `Ciclo Regiones 1` en el diagrama, prepara los coeficientes alfa de las funciones de producción Cobb Douglas (de encontrarse en la sub-fase de regresión), algunas variables de alimentación (si es el primer año de optimización) o algunas variables de la balanza comercial y viviendas (si corresponde a los años siguientes de optimización). En este último caso también se calcularía la fracción de ayuda comercial de ser una corrida con esta opción activada. Por cada iteración del ciclo sólo se ejecutaría uno de los casos mencionados o, de encontrarse en la fase de proyección y ya terminada la fase de regresión, ninguno.

El segundo ciclo de las regiones, llamado `Ciclo Regiones 2` en el diagrama, es bastante más extenso que el anterior y podría pensarse como el cálculo principal de las fases de proyección y optimización. Está dividido en los siguientes sub-módulos:

1. `Cálculo previo a fases`: cálculo del Capital total, la demografía, algunas variables de viviendas y, en el caso de fase de proyección o primer año de fase de optimización, la Mano de Obra sectorial.

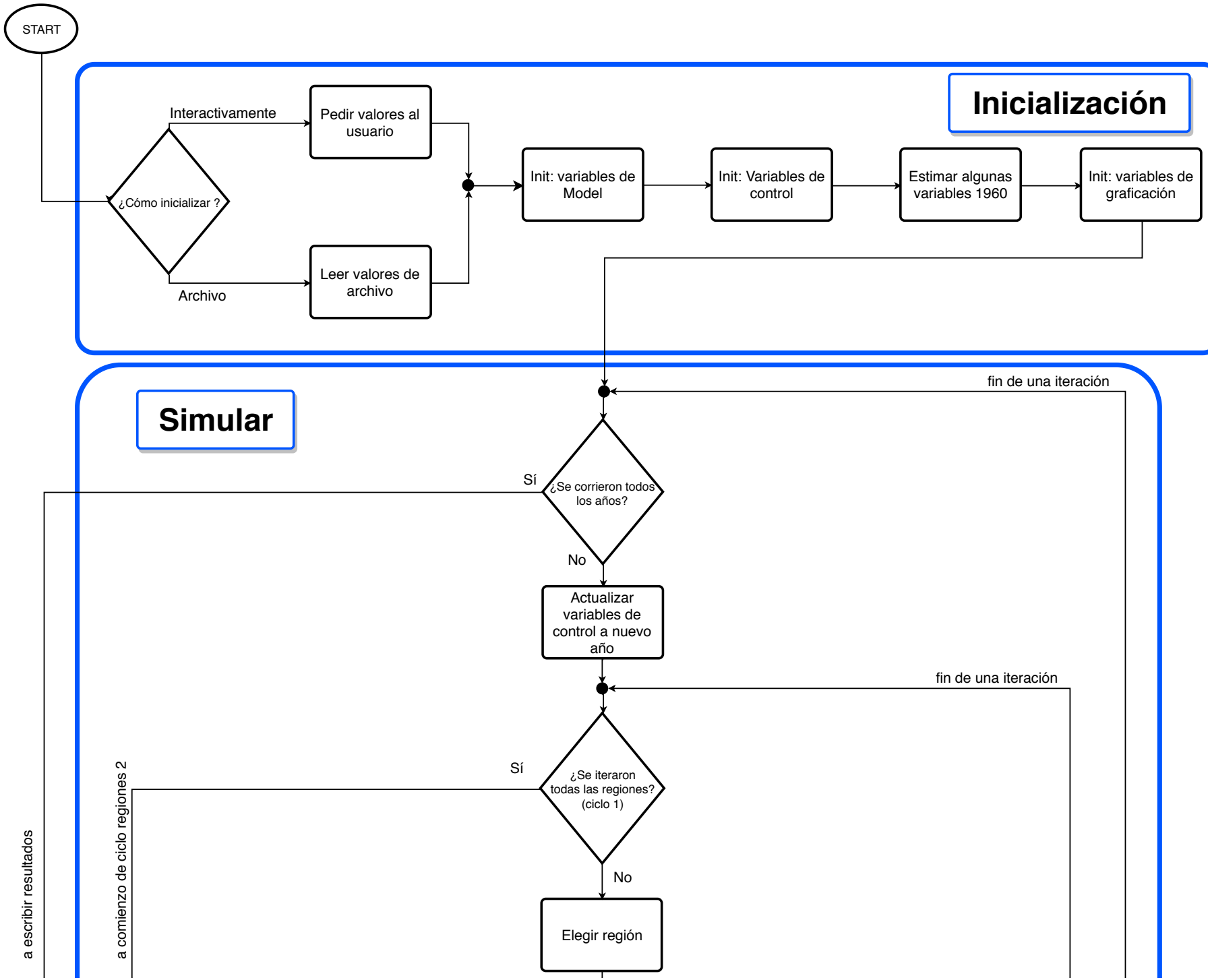
2. `Fases`: donde se calculan los subtotales por sector de Capital y Mano de Obra y las variables dependientes de ambos. La forma de calcularlos varía dependiendo de la fase.

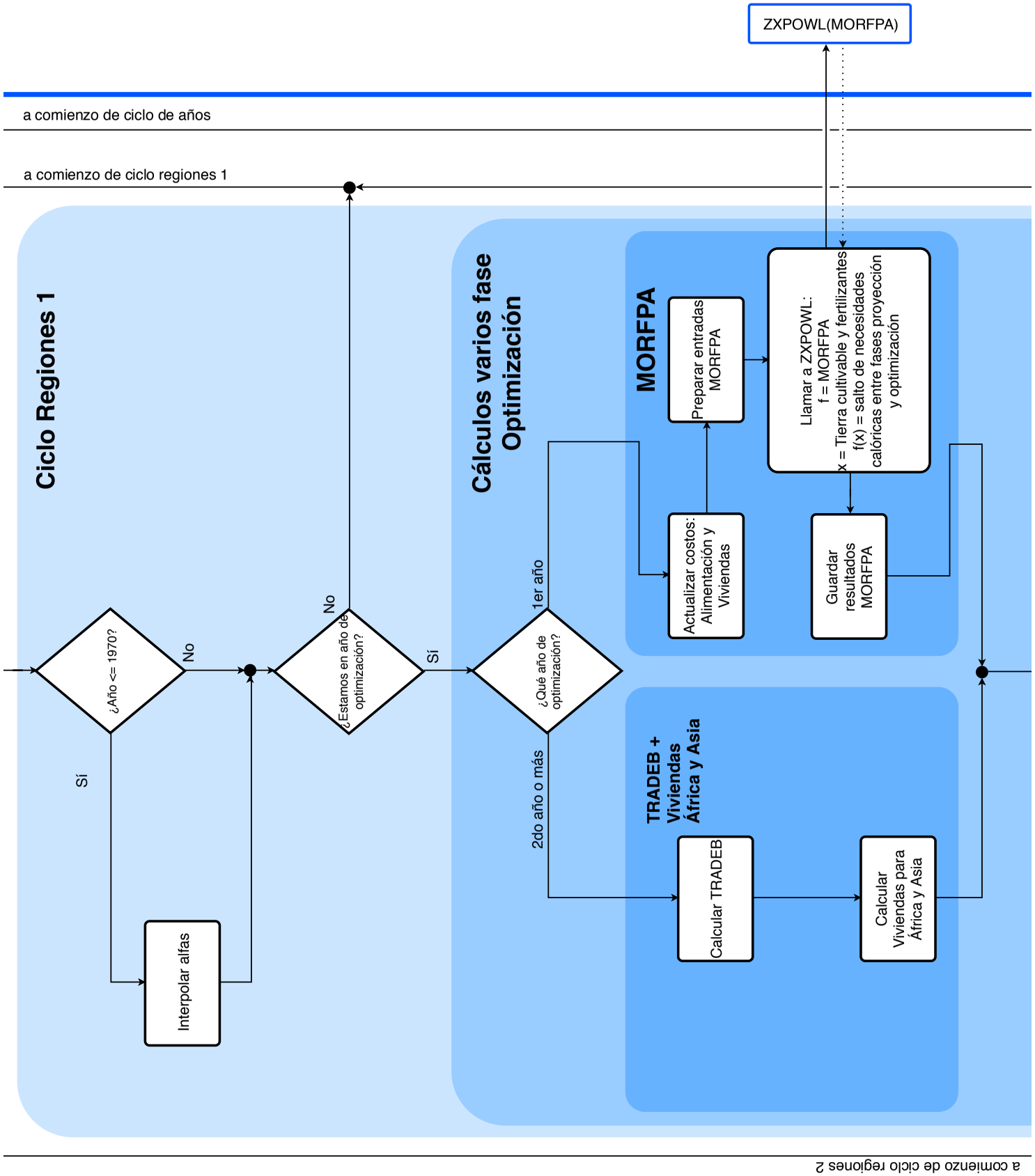
- Fase de Proyección (`llamada a PRLAWM`): se hace una llamada a un punto de entrada especial de la función LELAPA, para calcular las variables. Aunque corresponde a la fase de proyección, esta llamada se hace todos los años durante todas las fases.
- Fase de Optimización (`Optimización de LELAPA`): a grandes rasgos, lleva a cabo una optimización de la función `LELAPA` haciendo uso de la rutina de optimización `ZXPOWL`. La primera llamada a `LELAPA` sin optimizar se lleva a cabo para tener un resultado base por si la optimización no funciona correctamente, y las operaciones trigonométricas son necesarias por cuestiones matemáticas de reescalado de valores para la rutina de optimización.

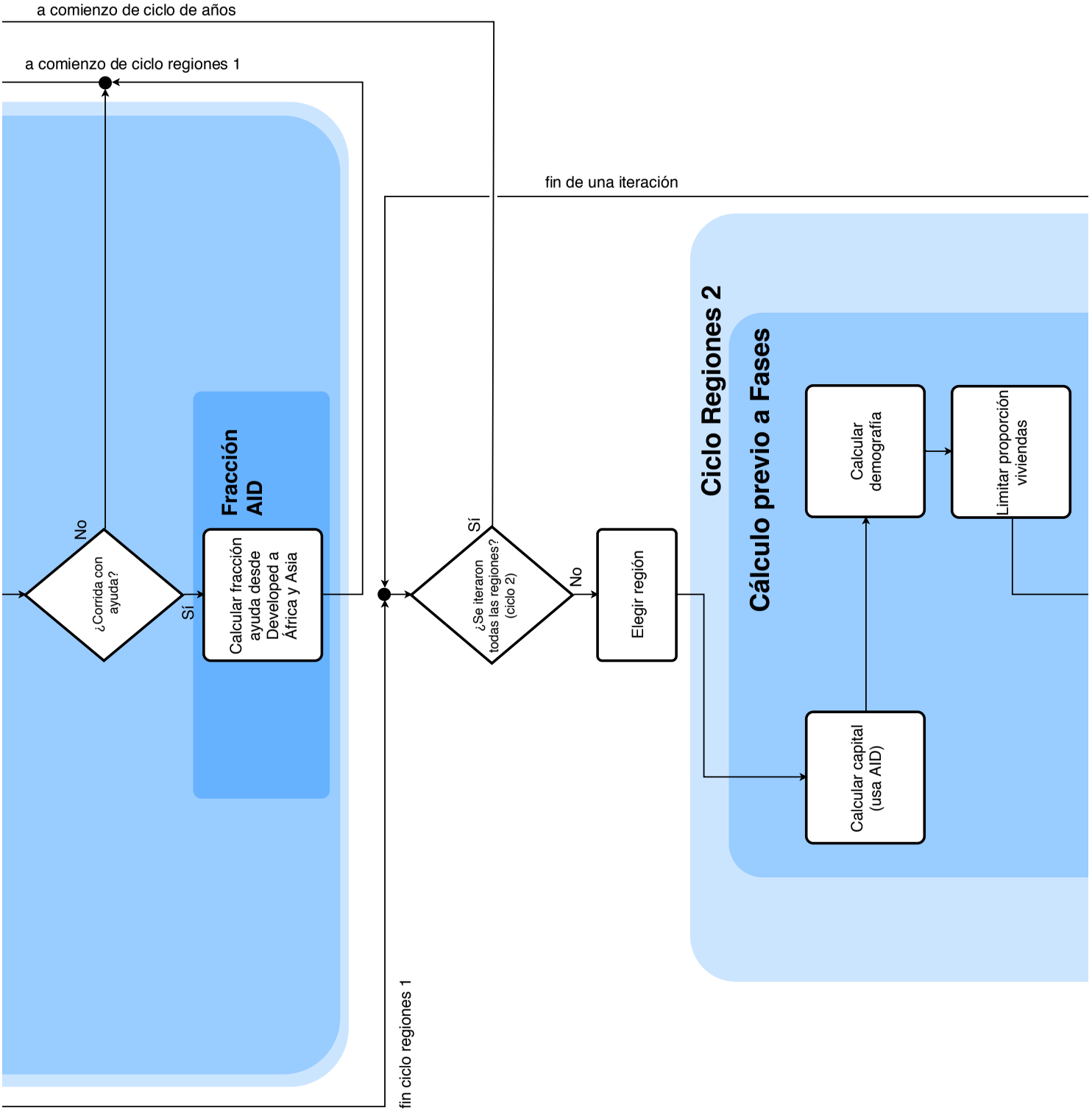
3. `Ayuda desde países desarrollados hacia África y Asia`: de ser una corrida con ayuda económica activada, aquí se calcula cómo distribuir la ayuda entre África y Asia.

4. `Reducción de brecha entre países desarrollados vs. Latinoamérica, África y Asia`: dado que la brecha entre los países desarrollados y el resto del mundo es considerable, aquí se busca tender a reducirla incrementando los costos de educación y viviendas en las zonas menos privilegiadas.

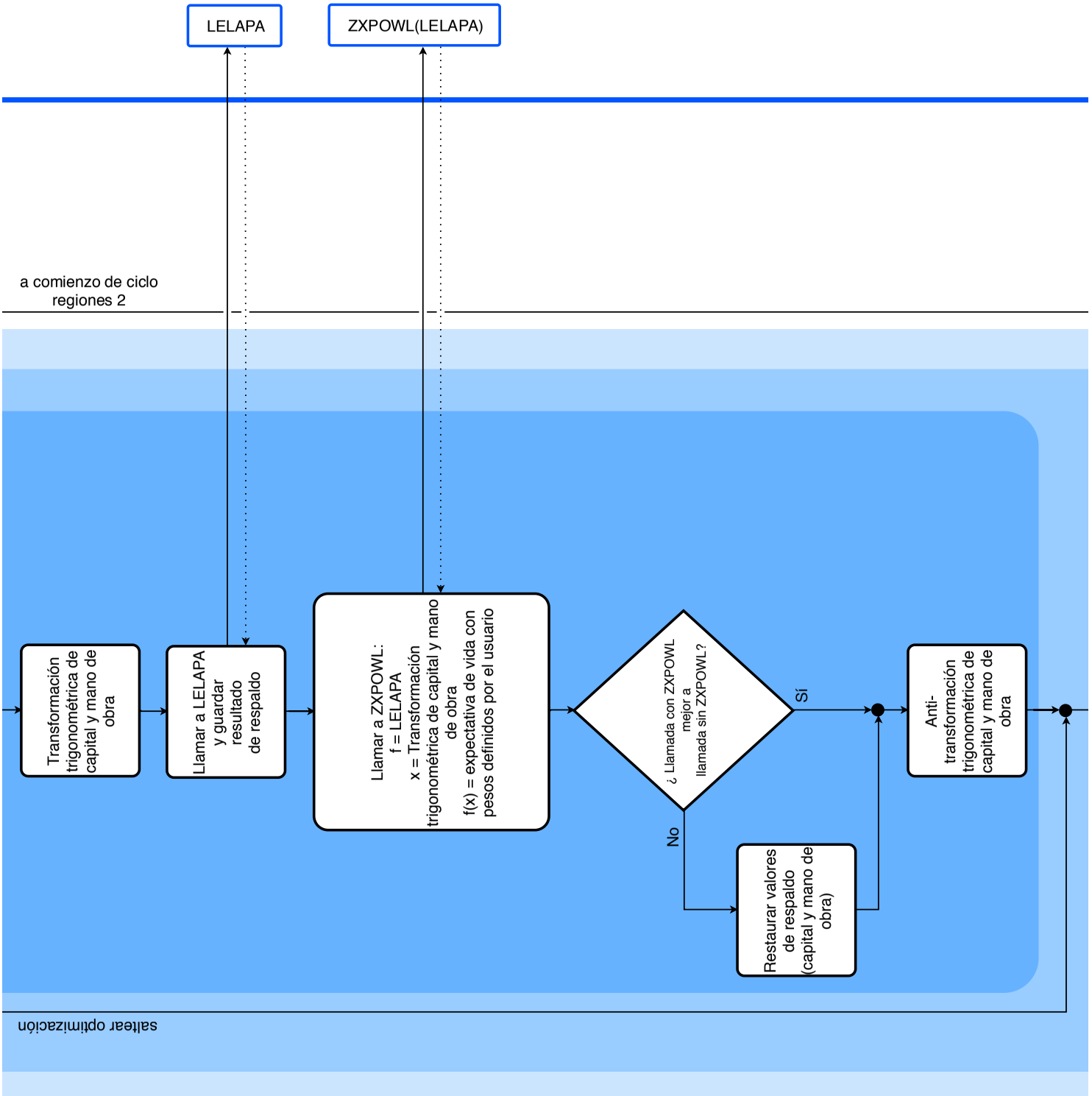
Finalmente, una vez que la simulación ha finalizado se escriben a disco los resultados. Por compatibilidad hacia atrás la nueva versión mantiene la escritura en el formato original pero además agregamos una escritura de archivos CSV (*Comma Separated Values*), uno por cada región, para facilitar el estudio de los resultados.


START
Inicialización
Interactivamente
Pedir valores al usuario
¿Cómo inicializar ?
Archivo
Leer valores de archivo
Init: variables de Model
Init: Variables de control
Estimar algunas variables 1960
Init: variables de graficación
Simular
fin de una iteración
Sí
¿Se corrieron todos los años?
No
Actualizar variables de control a nuevo año
fin de una iteración
Sí
¿Se iteraron todas las regiones? (ciclo 1)
No
Elegir región
a escribir resultados
a comienzo de ciclo regiones 2

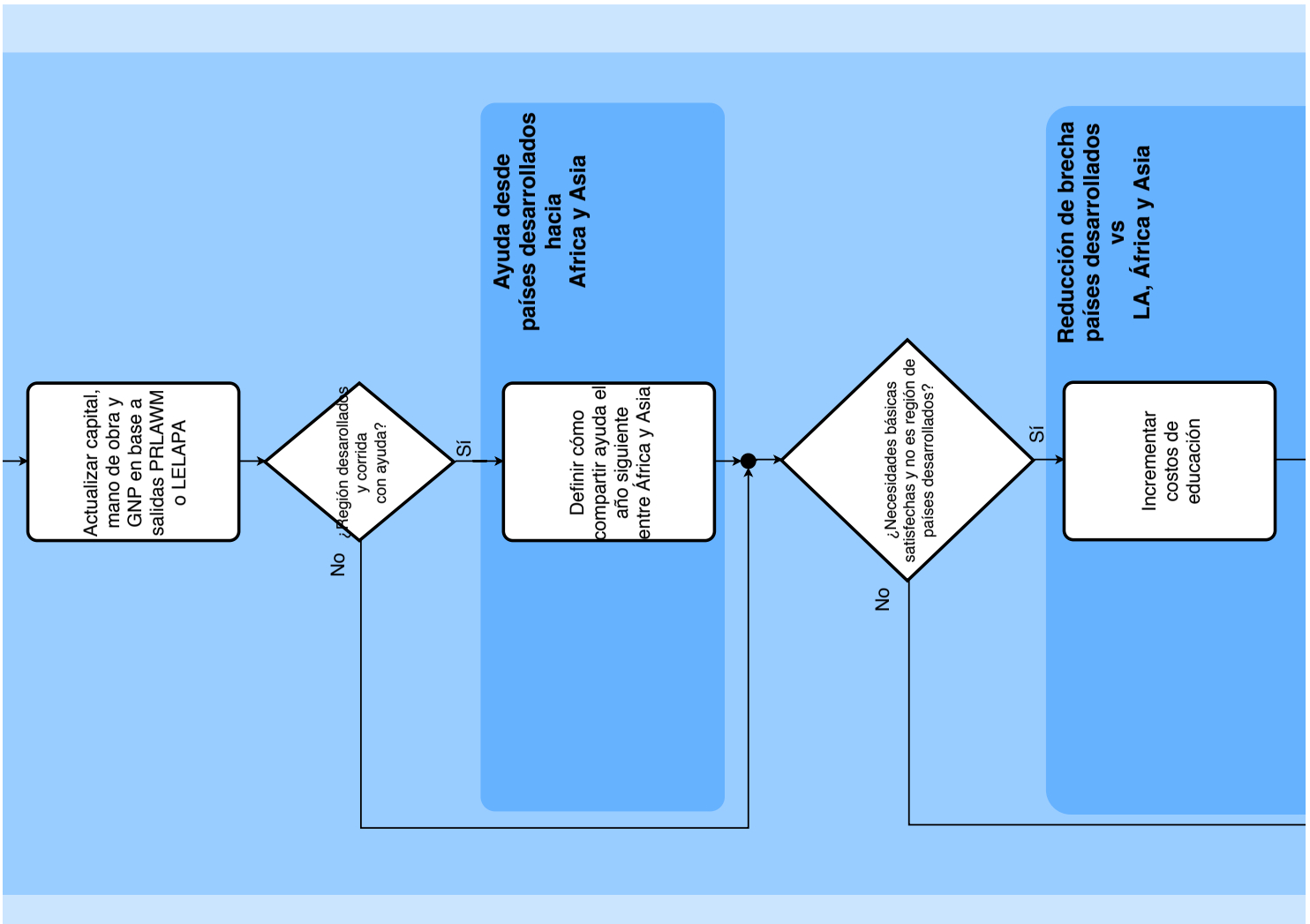

ZXPOWL(MORFPA)
a comienzo de ciclo de años
a comienzo de ciclo regiones 1
Ciclo Regiones 1
¿Año <= 1970?
No
Sí
Interpolar alfas
¿Estamos en año de optimización?
No
Sí
Cálculos varios fase Optimización
¿Qué año de optimización?
1er año
2do año o más
MORFPA
Actualizar costos: Alimentación y Viviendas
Preparar entradas MORFPA
Llamar a ZXPOWL:
f = MORFPA
x = Tierra cultivable y fertilizantes
f(x) = salto de necesidades calóricas entre fases proyección y optimización
Guardar resultados MORFPA
TRADEB + Viviendas África y Asia
Calcular TRADEB
Calcular Viviendas para África y Asia
a comienzo de ciclo regiones 2
a escribir resultados

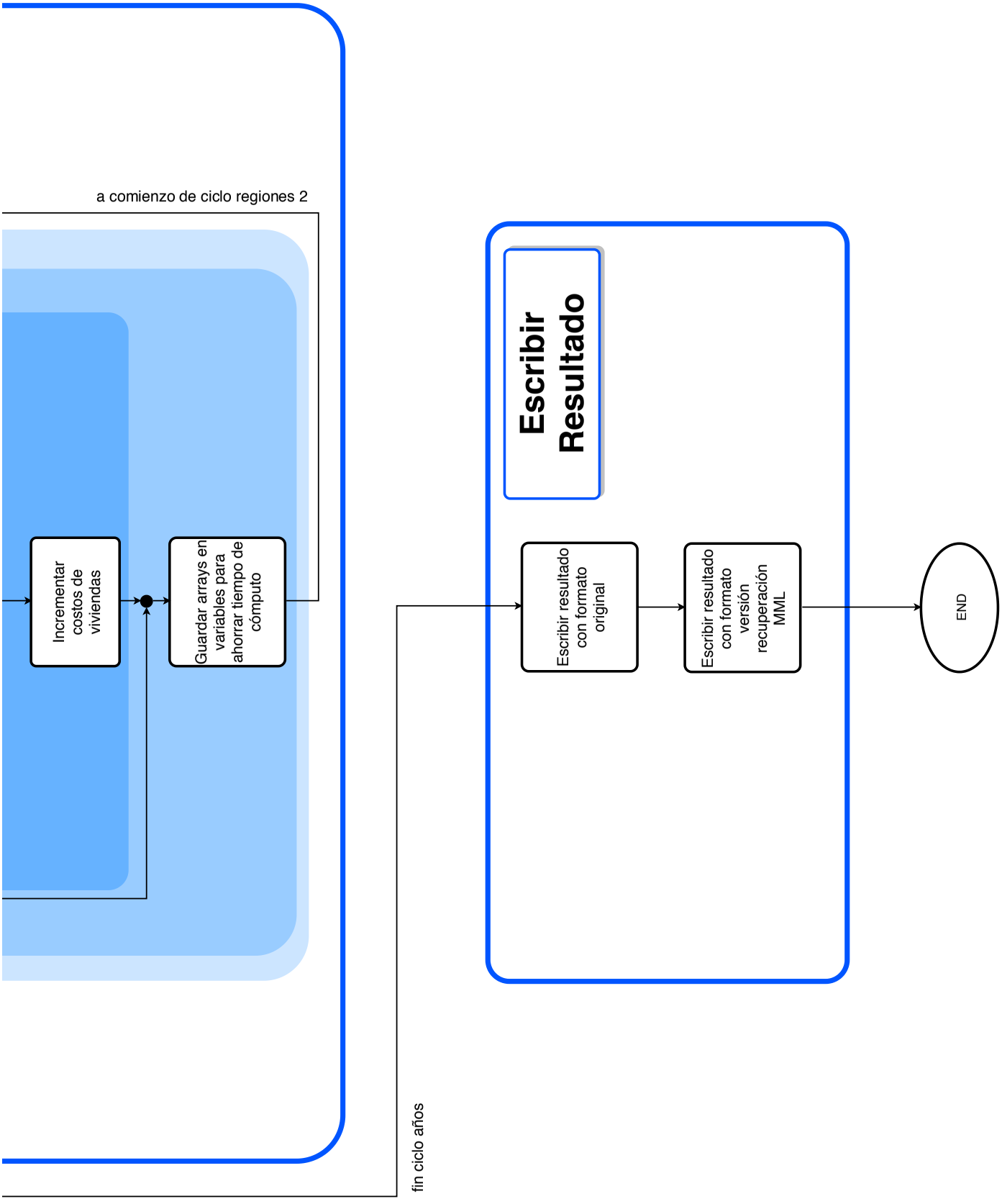

a comienzo de ciclo de años
a comienzo de ciclo regiones 1
fin de una iteración
Fracción AID
¿Corrida con ayuda?
No
Sí
Calcular fracción ayuda desde Developed a África y Asia
¿Se iteraron todas las regiones? (ciclo 2)
Sí
No
Elegir región
Ciclo Regiones 2
Cálculo previo a Fases
Calcular demografía
Limitar proporción viviendas
Calcular capital (usa AID)
fin ciclo regiones 1
a escribir resultados

a comienzo de ciclo regiones 2
* correspondería a la fase de Proyección pero se llama todos los años
LELAPA: PRLAWM
Sí
¿Estamos en 2do año o más de optimización?
No
Calcular mano de obra
Fases
Guardar arrays en variables para ahorrar tiempo de cómputo
Llamada a PRLAWM *
Llamar a PRLAWM con capital y mano de obra
No
¿Estamos en fase de optimización?
saltear optimización
Sí
Optimización de LELAPA
Inicializar pesos para la optimización ponderada
a escribir resultados


LELAPA
ZXPOWL(LELAPA)
a comienzo de ciclo
regiones 2
Transformación trigonométrica de capital y mano de obra
Llamar a LELAPA y guardar resultado de respaldo
Llamar a ZXPOWL:
f = LELAPA
x = Transformación trigonométrica de capital y mano de obra
f(x) = expectativa de vida con pesos definidos por el usuario
¿ Llamada con ZXPOWL mejor a llamada sin ZXPOWL?
Sí
No
Restaurar valores de respaldo (capital y mano de obra)
Anti-transformación trigonométrica de capital y mano de obra
saltear optimización
a escribir resultados

a comienzo de ciclo regiones 2




a comienzo de ciclo regiones 2
Incrementar costos de viviendas
Guardar arrays en variables para ahorrar tiempo de cómputo
fin ciclo años
Escribir Resultado
Escribir resultado con formato original
Escribir resultado con formato versión recuperación MML
END

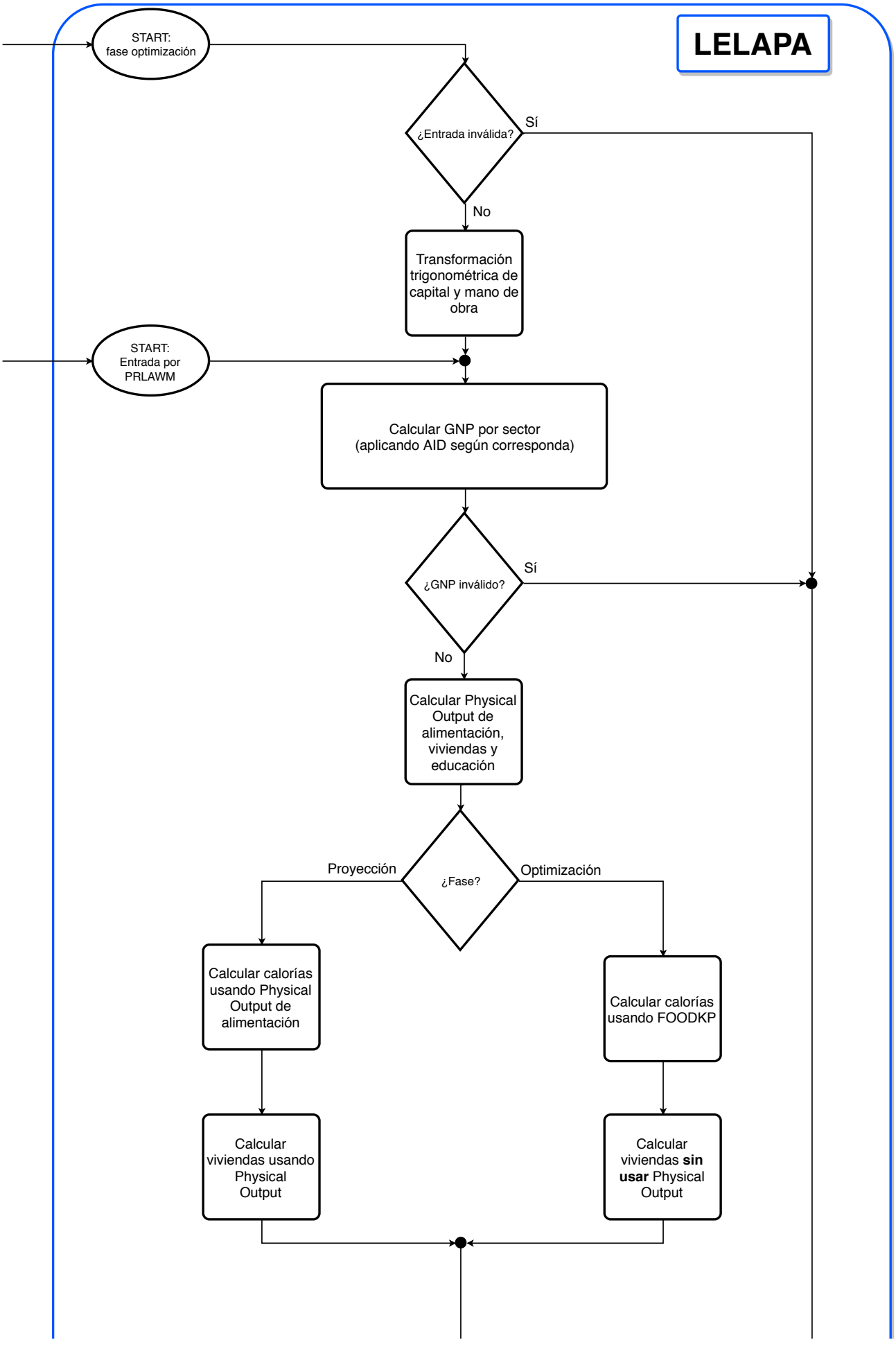
3.3.3.2. Función LELAPA

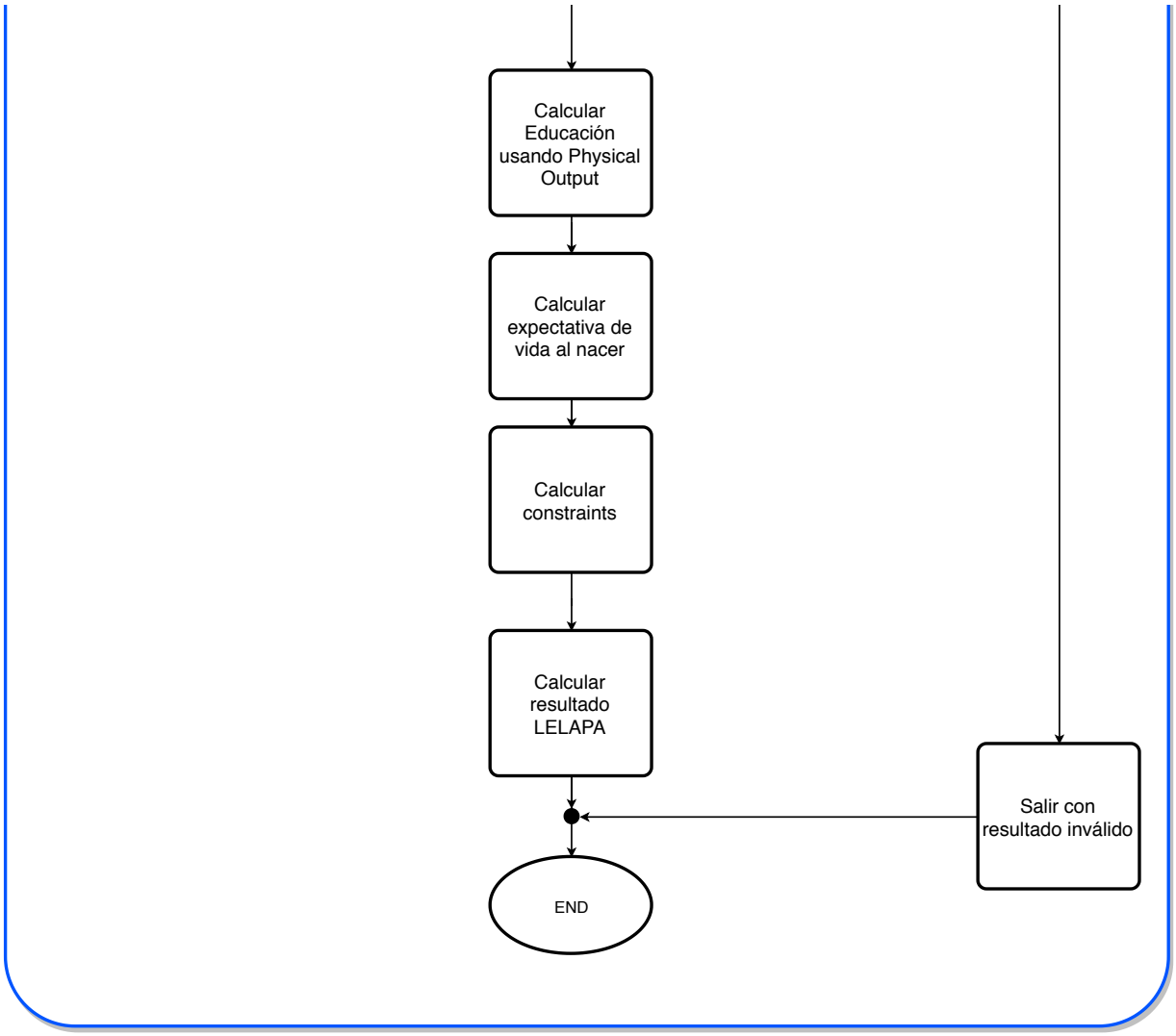
En la Sección 1.1.3 mencionamos la función LELAPA encargada de llevar a cabo el cálculo de la Esperanza de Vida al Nacer en base a los subtotales de Capital y Mano de Obra, encapsulando todos los cálculos intermedios. Fue desarrollada de esta forma para poder ser optimizada utilizando la rutina ZXPOWL. El diagrama aquí presentado está basado en el código original, pero no existían diagramas similares de la función por lo que decidimos incluirlo en el proyecto.

Como vemos en el diagrama, la función tiene dos entradas posibles: una correspondiente a la fase de optimización y otra a la de proyección por PRLAWM. Esta distinción fue necesaria porque, por cuestiones implementativas, antes de optimizar esta subrutina con ZXPOWL, se le aplican algunas operaciones trigonométricas a las variables de entrada.

Cualquiera sea la entrada, la función calcula inicialmente los subtotales por sector del GNP en base a los subtotales de Capital y Mano de Obra. Este resultado luego es usado para calcular el *physical output* que finalmente es usado para obtener, sólo durante la fase de proyección, la producción de los sectores de Alimentación y Viviendas. Durante la fase de optimización, la producción alimenticia se calcula utilizando la función FOODKP y la producción de viviendas usando el GNP y los costos del sector.

En lo que resta de la función las fases no llevan a cabo cálculos distintos, como en el cálculo del sector de educación, que durante ambas se calcula en base al *physical output*, obteniendo el último dato necesario para poder calcular la Esperanza de Vida al Nacer que es luego calculada. En el final de la función se calculan las restricciones y el resultado de la función LELAPA que son utilizados sólo durante la fase de optimización y son ignorados durante la proyección.


LELAPA
START:
fase optimización
¿Entrada inválida?
Sí
No
Transformación trigonométrica de capital y mano de obra
START:
Entrada por PRLAWM
Calcular GNP por sector
(aplicando AID según corresponda)
¿GNP inválido?
Sí
No
Calcular Physical Output de alimentación, viviendas y educación
¿Fase?
Proyección
Optimización
Calcular calorías usando Physical Output de alimentación
Calcular calorías usando FOODKP
Calcular viviendas usando Physical Output
Calcular viviendas **sin usar** Physical Output


Calcular Educación usando Physical Output
Calcular expectativa de vida al nacer
Calcular constraints
Calcular resultado LELAPA
Salir con resultado inválido
END

4. INTERFAZ WEB DE SIMULACIÓN

Para facilitar la experimentación con el modelo desarrollamos un sitio web [DCS17] para correr, analizar y guardar simulaciones del Modelo Mundial Latinoamericano recuperado. Cualquier usuario puede explorar los resultados de una corrida estándar de ejemplo y acceder a documentación sobre las variables y parámetros. Usuarios que soliciten acceso pueden diseñar nuevos experimentos y mantener un registro de las simulaciones y sus resultados.

4.1. Guía de Uso

En esta Sección hacemos una introducción a las distintas funcionalidades provistas por la página web y algunas muestras ilustrativas se sus pantallas principales.

El primer paso consiste en autenticarse mediante un par usuario y contraseña (se solicita por mail) como se ve en la Figura 4.1. Cualquier usuario sin clave puede acceder como Invitado limitado a operaciones de solo lectura (no puede ejecutar nuevas simulaciones, solo explorar resultados de corridas de ejemplo).

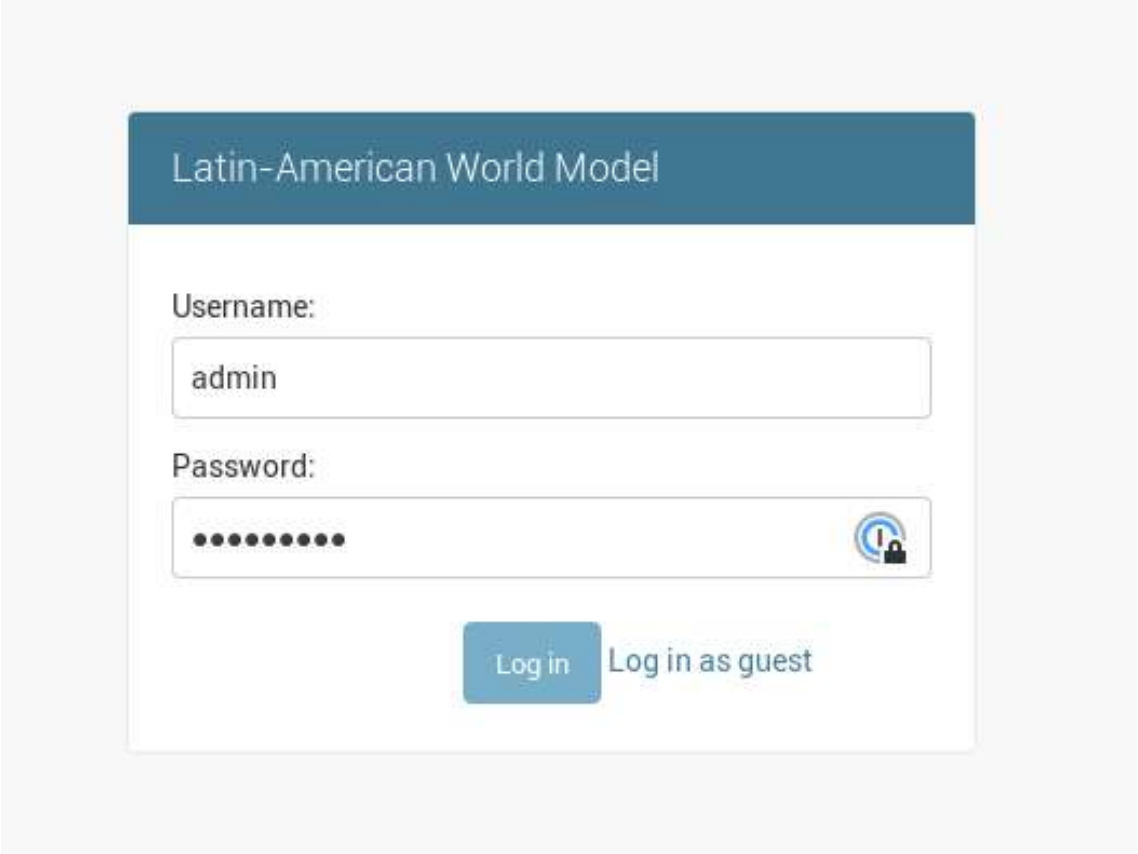
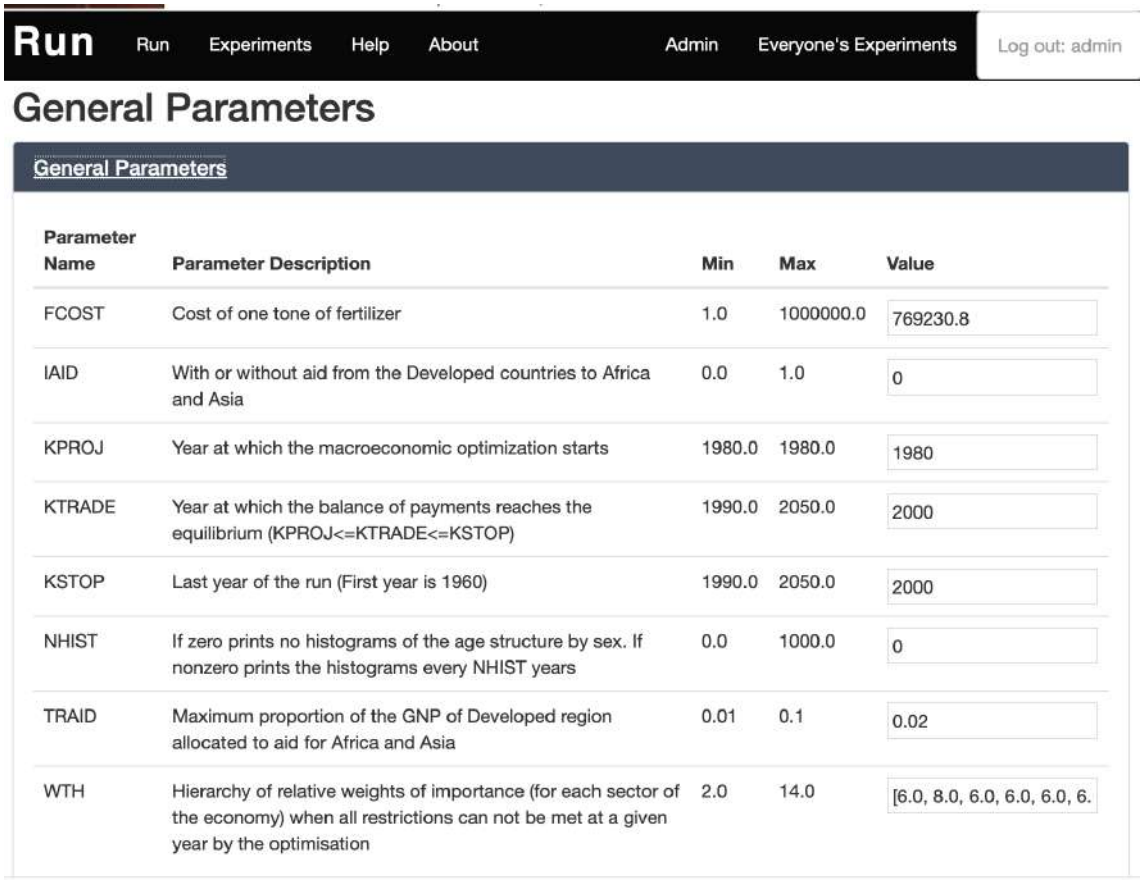


Figura 4.1: Autenticación de la página.

Una vez autenticado el usuario es dirigido a la sección de la página encargada de correr

simulaciones, como vemos en la Figura A.1. Se pueden configurar los parámetros o utilizar sus valores por defecto, teniendo a disposición los valor mínimos y máximos posibles para varios de ellos.



Parameter Name	Parameter Description	Min	Max	Value
FCOST	Cost of one tone of fertilizer	1.0	1000000.0	769230.8
IAID	With or without aid from the Developed countries to Africa and Asia	0.0	1.0	0
KPROJ	Year at which the macroeconomic optimization starts	1980.0	1980.0	1980
KTRADE	Year at which the balance of payments reaches the equilibrium (KPROJ<=KTRADE<=KSTOP)	1990.0	2050.0	2000
KSTOP	Last year of the run (First year is 1960)	1990.0	2050.0	2000
NHIST	If zero prints no histograms of the age structure by sex. If nonzero prints the histograms every NHIST years	0.0	1000.0	0
TRAID	Maximum proportion of the GNP of Developed region allocated to aid for Africa and Asia	0.01	0.1	0.02
WTH	Hierarchy of relative weights of importance (for each sector of the economy) when all restrictions can not be met at a given year by the optimisation	2.0	14.0	[6.0, 8.0, 6.0, 6.0, 6.0, 6.

Figura 4.2: Configuración de los parámetros de una nueva simulación.

Cuando finaliza la corrida se presenta una página con sus resultados, como la mostrada en la Figura 4.3. En la parte de arriba se muestra la *metadata* como el número de identificación de la corrida, su fecha de inicio, el usuario y un comentario (de existir).

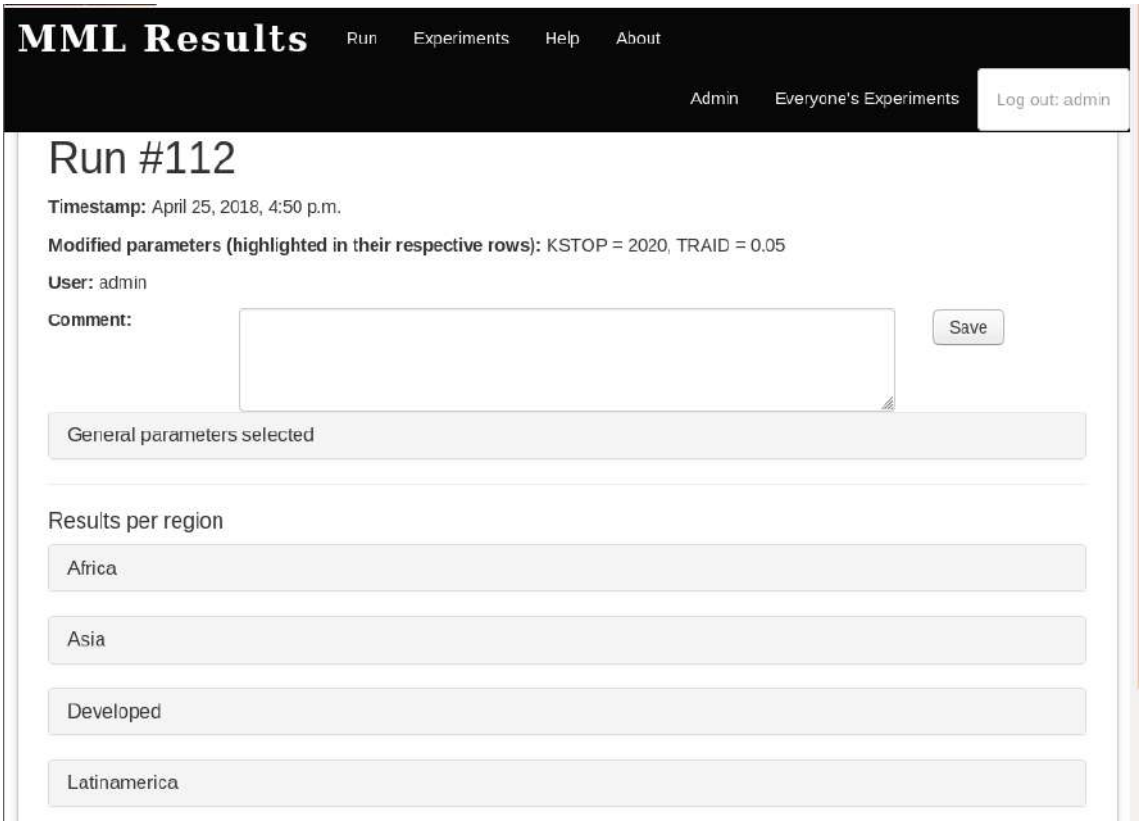
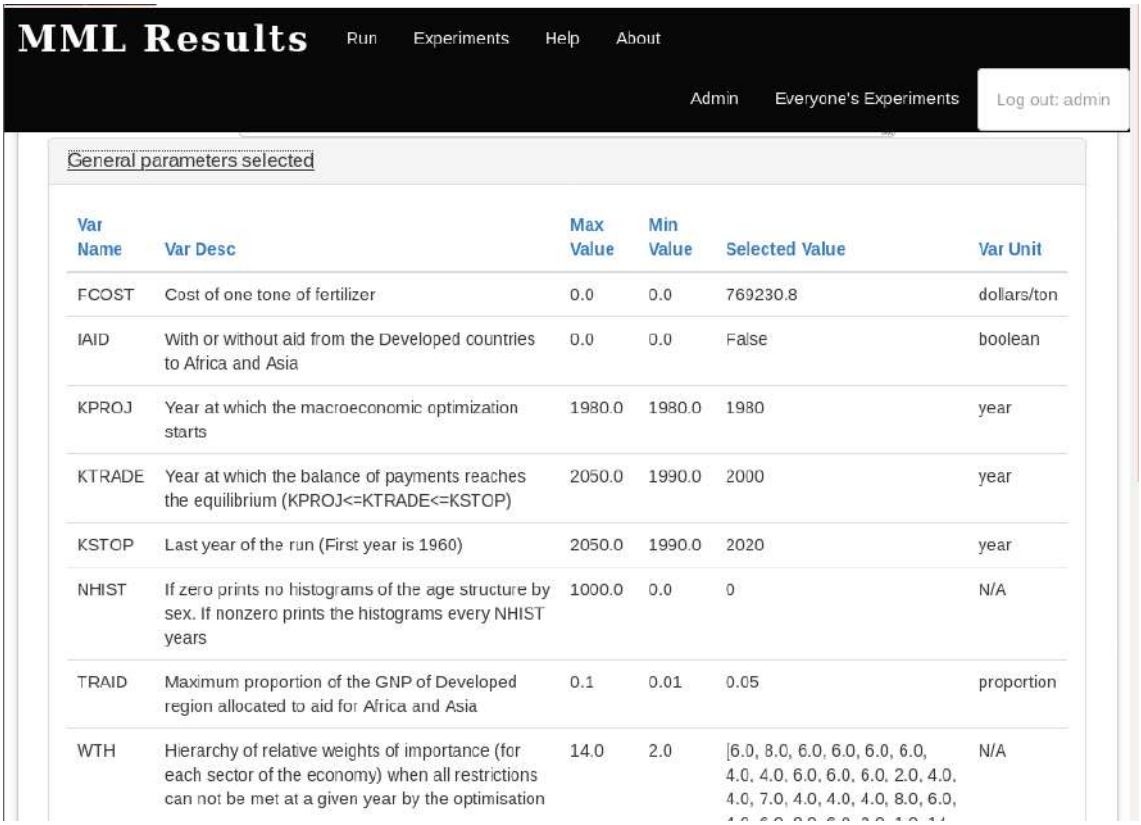


Figura 4.3: Página base de los resultados de una simulación.

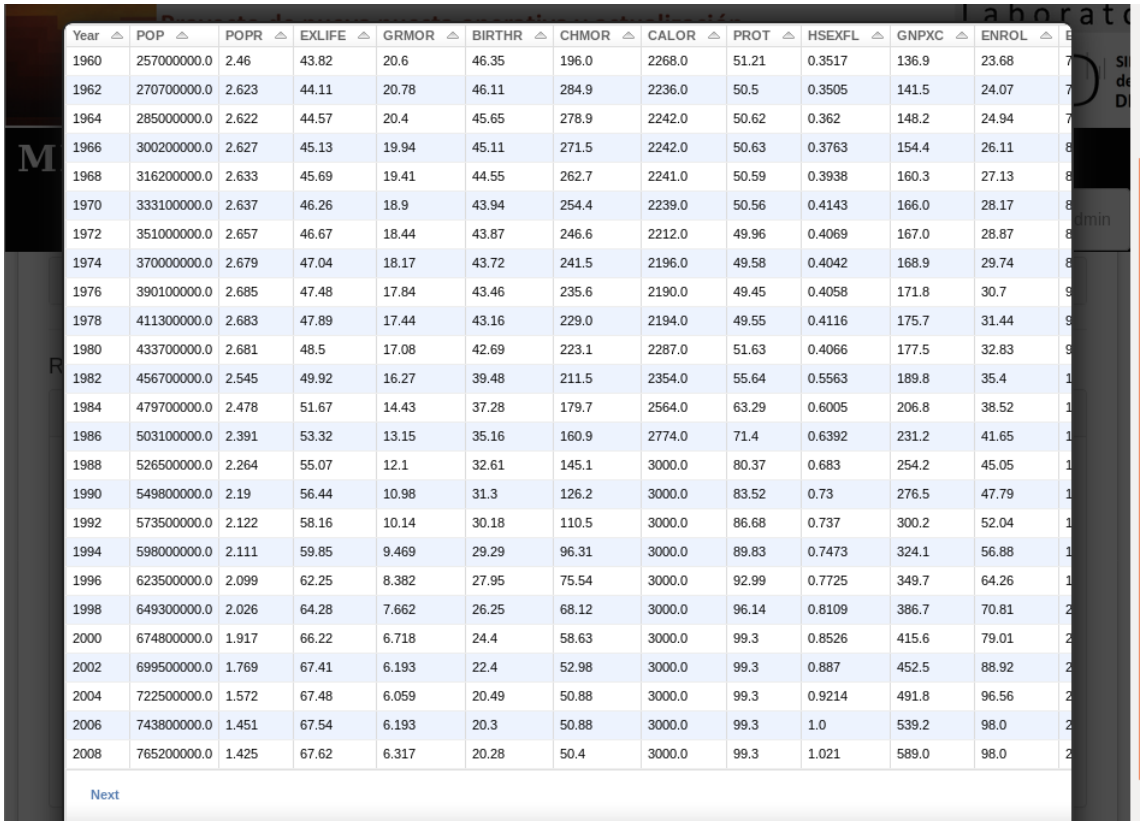
Para facilitar al usuario el análisis de los resultados, el sistema recuerda los valores de los parámetros elegidos, como puede verse en la Figura 4.4, que muestra la información contenida en el *acordeón* "General Parameters Selected". Aquí se vuelve a mostrar la información de los parámetros como su descripción, valor máximo y valor mínimo pero ahora se especifica el valor asignado por el usuario, que puede ser o no el valor por defecto.



Var Name	Var Desc	Max Value	Min Value	Selected Value	Var Unit
FCOST	Cost of one tone of fertilizer	0.0	0.0	769230.8	dollars/ton
IAID	With or without aid from the Developed countries to Africa and Asia	0.0	0.0	False	boolean
KPROJ	Year at which the macroeconomic optimization starts	1980.0	1980.0	1980	year
KTRADE	Year at which the balance of payments reaches the equilibrium (KPROJ<=KTRADE<=KSTOP)	2050.0	1990.0	2000	year
KSTOP	Last year of the run (First year is 1960)	2050.0	1990.0	2020	year
NHIST	If zero prints no histograms of the age structure by sex. If nonzero prints the histograms every NHIST years	1000.0	0.0	0	N/A
TRAID	Maximum proportion of the GNP of Developed region allocated to aid for Africa and Asia	0.1	0.01	0.05	proportion
WTH	Hierarchy of relative weights of importance (for each sector of the economy) when all restrictions can not be met at a given year by the optimisation	14.0	2.0	[6.0, 8.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0, 6.0, 6.0, 6.0, 2.0, 4.0, 4.0, 7.0, 4.0, 4.0, 4.0, 8.0, 6.0,	N/A

Figura 4.4: Los parámetros elegidos asociados a los resultados de una simulación.

Con respecto a los resultados de la simulación, una de las forma de presentarlos es en forma de tabla, como se ve en la Figura 4.5. En ella se incluyen los valores anuales de las variables de salida de manera equivalente a los resultados escritos por la versión original del modelo, sólo que en formato de tabla.



Year	POP	POPR	EXLIFE	GRMOR	BIRTHR	CHMOR	CALOR	PROT	HSEXFL	GNPXC	ENROL
1960	257000000.0	2.46	43.82	20.6	46.35	196.0	2268.0	51.21	0.3517	136.9	23.68
1962	270700000.0	2.623	44.11	20.78	46.11	284.9	2236.0	50.5	0.3505	141.5	24.07
1964	285000000.0	2.622	44.57	20.4	45.65	278.9	2242.0	50.62	0.362	148.2	24.94
1966	300200000.0	2.627	45.13	19.94	45.11	271.5	2242.0	50.63	0.3763	154.4	26.11
1968	316200000.0	2.633	45.69	19.41	44.55	262.7	2241.0	50.59	0.3938	160.3	27.13
1970	333100000.0	2.637	46.26	18.9	43.94	254.4	2239.0	50.56	0.4143	166.0	28.17
1972	351000000.0	2.657	46.67	18.44	43.87	246.6	2212.0	49.96	0.4069	167.0	28.87
1974	370000000.0	2.679	47.04	18.17	43.72	241.5	2196.0	49.58	0.4042	168.9	29.74
1976	390100000.0	2.685	47.48	17.84	43.46	235.6	2190.0	49.45	0.4058	171.8	30.7
1978	411300000.0	2.683	47.89	17.44	43.16	229.0	2194.0	49.55	0.4116	175.7	31.44
1980	433700000.0	2.681	48.5	17.08	42.69	223.1	2287.0	51.63	0.4066	177.5	32.83
1982	456700000.0	2.545	49.92	16.27	39.48	211.5	2354.0	55.64	0.5563	189.8	35.4
1984	479700000.0	2.478	51.67	14.43	37.28	179.7	2564.0	63.29	0.6005	206.8	38.52
1986	503100000.0	2.391	53.32	13.15	35.16	160.9	2774.0	71.4	0.6392	231.2	41.65
1988	526500000.0	2.264	55.07	12.1	32.61	145.1	3000.0	80.37	0.683	254.2	45.05
1990	549800000.0	2.19	56.44	10.98	31.3	126.2	3000.0	83.52	0.73	276.5	47.79
1992	573500000.0	2.122	58.16	10.14	30.18	110.5	3000.0	86.68	0.737	300.2	52.04
1994	598000000.0	2.111	59.85	9.469	29.29	96.31	3000.0	89.83	0.7473	324.1	56.88
1996	623500000.0	2.099	62.25	8.382	27.95	75.54	3000.0	92.99	0.7725	349.7	64.26
1998	649300000.0	2.026	64.28	7.662	26.25	68.12	3000.0	96.14	0.8109	386.7	70.81
2000	674800000.0	1.917	66.22	6.718	24.4	58.63	3000.0	99.3	0.8526	415.6	79.01
2002	699500000.0	1.769	67.41	6.193	22.4	52.98	3000.0	99.3	0.887	452.5	88.92
2004	722500000.0	1.572	67.48	6.059	20.49	50.88	3000.0	99.3	0.9214	491.8	96.56
2006	743800000.0	1.451	67.54	6.193	20.3	50.88	3000.0	99.3	1.0	539.2	98.0
2008	765200000.0	1.425	67.62	6.317	20.28	50.4	3000.0	99.3	1.021	589.0	98.0

Next

Figura 4.5: Tabla con los valores de las variables por cada año de la simulación.

Otra manera de obtener los resultados es visualizándolos en un gráfico interactivo, como vemos en la Figura 4.6. En ese ejemplo se muestran los valores porcentuales anuales de la Mano de Obra sectorial y mediante el uso del puntero se pueden mostrar valores específicos de un año y una variable.

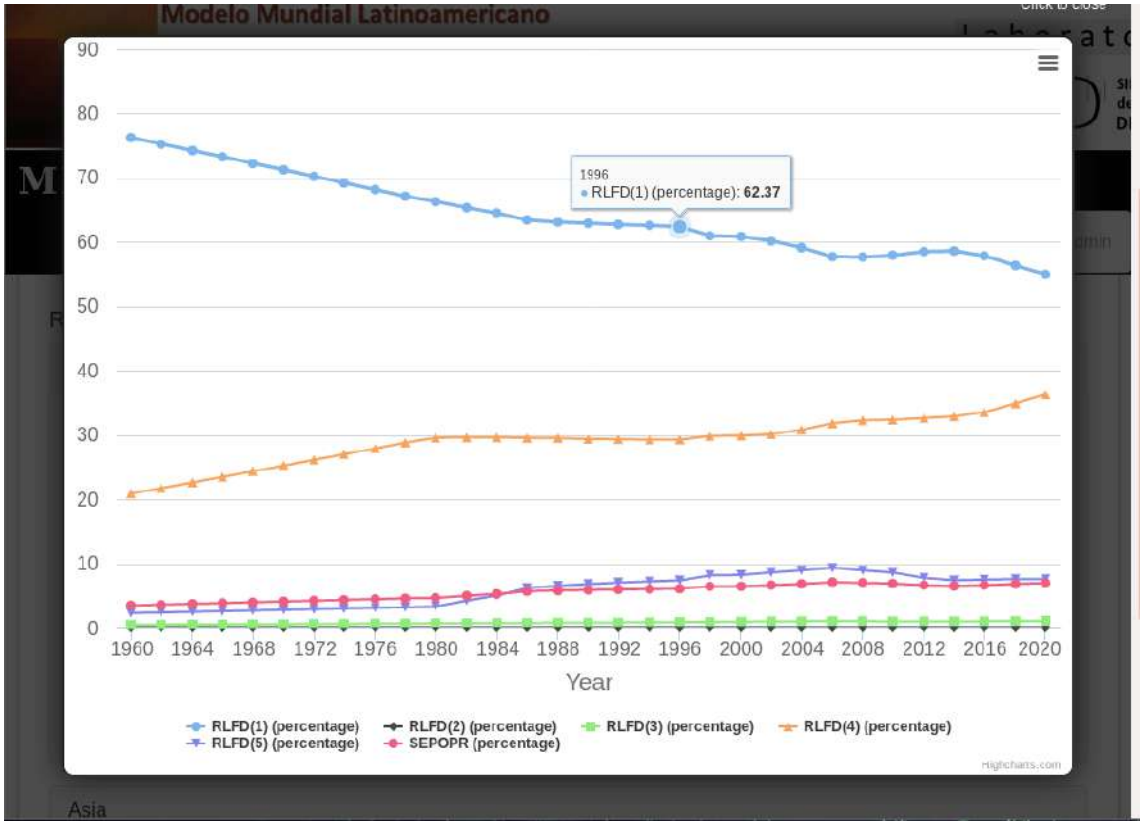


Figura 4.6: Gráfico con los porcentajes anuales de Mano de Obra sectorial de una simulación.

Otra funcionalidad muy útil es el historial de experimentos, como el mostrado en la Figura 4.7, accesible mediante la pestaña “Experiments”. En él se guardan todas las corridas llevadas a cabo por el usuario para análisis futuros.

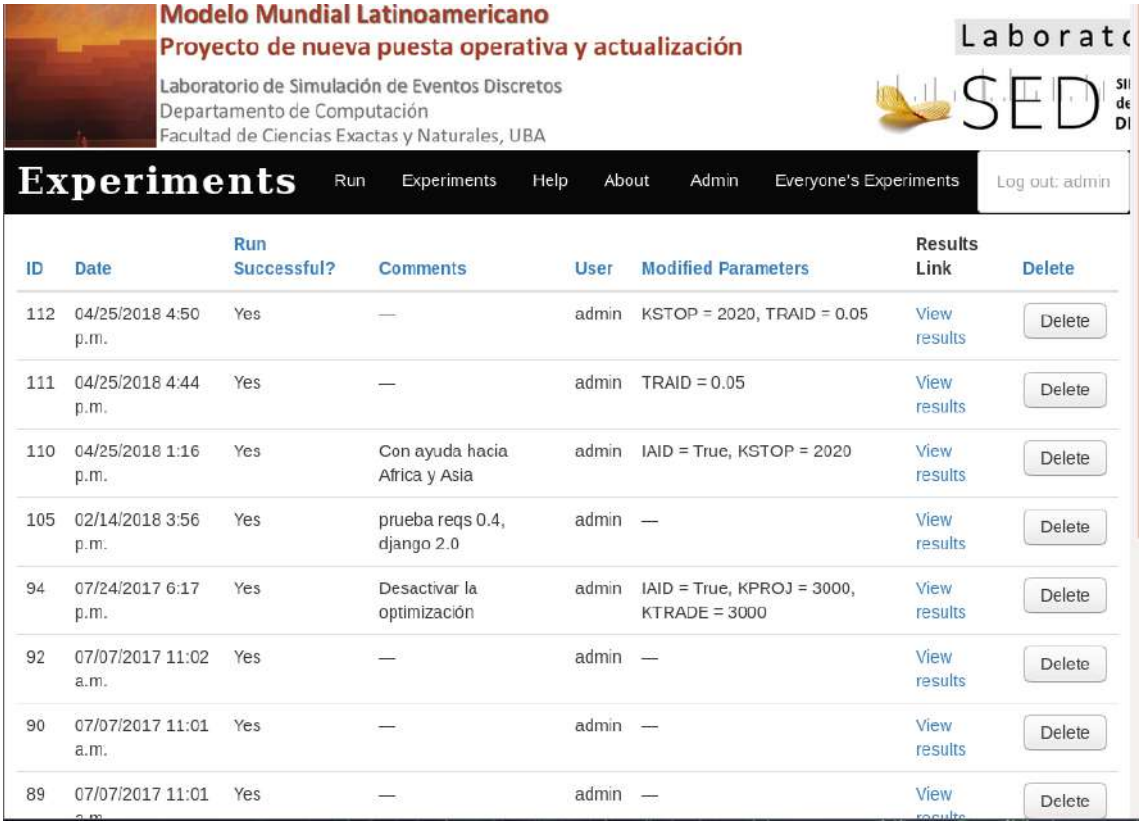


Figura 4.7: Historial de corridas del usuario.

Finalmente, la página web provee al usuario la pestaña "Help" con información útil para guiar el uso de las distintas funcionalidades. Por ejemplo, en la Figura 4.8 se muestra la guía donde se enumeran los pasos de un caso de uso común de experimentación con el MML, partiendo de la especificación de la simulación, siguiendo con la corrida y análisis de los resultados, y culminando con el historial de los experimentos pasados.

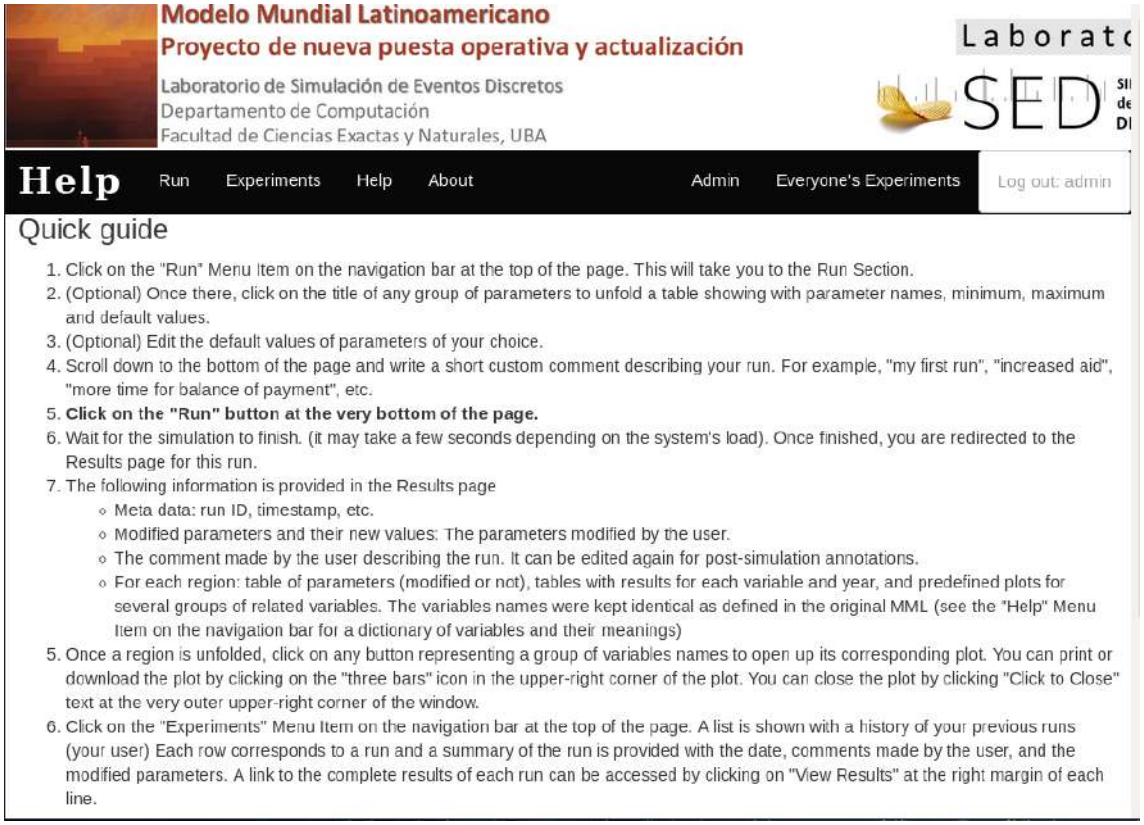


Figura 4.8: Guía de pasos a seguir para usar la página web.

En la misma pestaña “Help” se incluye la descripción de los parámetros, como la de la pestaña “Run” para nuevas corridas, además de la descripción de las variables de salida que extrajimos de la documentación provista por el código original. En la Figura 4.9 se muestran algunas variables del grupo de “Population Indicators” que también incluimos en este documento para comodidad del lector en la tabla del Apéndice A.

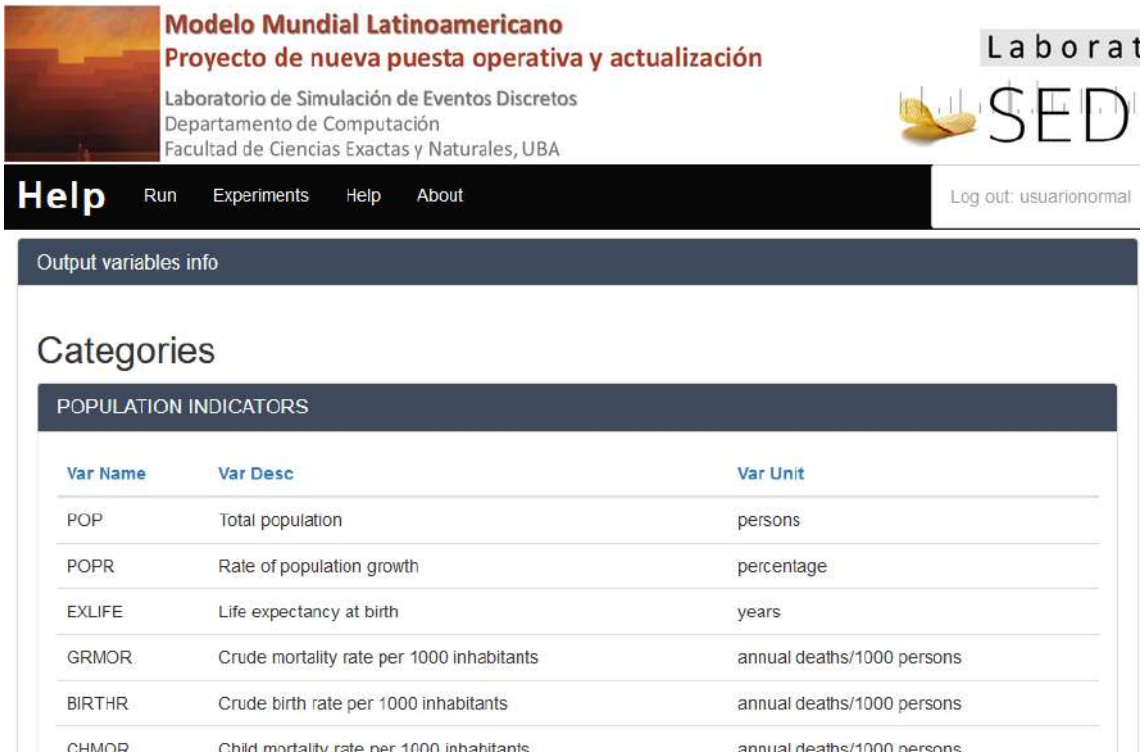


Figura 4.9: Descripción de las variables de salida mostradas en los resultados.

4.2. Arquitectura

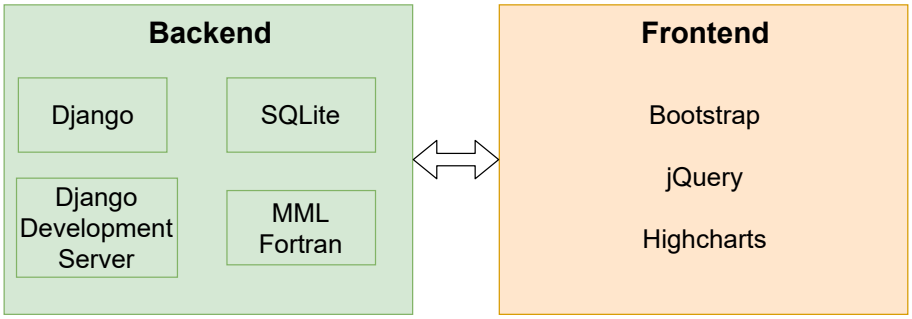


Figura 4.10: Herramientas utilizadas en el *Backend* y en el *Frontend*

La Figura 4.10 agrupa las herramientas utilizadas para el desarrollo del *backend* y el *frontend*, que fueron desarrollados independientemente. Una de las herramientas listadas en el *backend* es el *web framework* Django [Fou] que ofrece un ambiente pre-armado para los requerimientos más comunes de páginas web dinámicas como la autenticación, autorización, manejo de *endpoints*, mapeo objeto-relacional (ORM, por sus siglas en inglés), etc.

El módulo de ORM ofrece compatibilidad con varias bases de datos relacionales, como PostgreSQL, MySQL, Oracle y SQLite y mediante el uso de *plugins* de terceros esta lista se extiende considerablemente. Esto brinda una gran flexibilidad para escalar o extender el sitio en el futuro. Optamos por adoptar la base de datos SQLite [Conb] por su sencillez

de configuración y porque cumple todos los requerimientos de manejo de datos para este proyecto.

El último componente del *backend* es el simulador de MML en sí mismo, que corresponde a la versión `MML20reing`. El código Django tiene acceso a un ejecutable del modelo que compilamos a partir del código recuperado, para correr simulaciones en base a las especificaciones del usuario indicadas mediante la interfaz web y los resultados son cargados en la base de datos SQLite ya mencionada.

Por el lado del *frontend*, utilizamos Bootstrap [Tea], un *framework* para facilitar el manejo de estilos visuales (o CSS, Cascade Style Sheets); también utilizamos jQuery [jT], una biblioteca de javascript con funcionalidades para consultas sobre bases de datos, y Highcharts [Hig], una biblioteca también de javascript para crear los gráficos interactivos de los resultados de las simulaciones.

Con esta arquitectura, el sistema queda modularizado de manera elegante permitiendo avanzar en mejoras sin afectar el resto del sitio. Por ejemplo, una nueva versión mejorada del MML puede reemplazar al ejecutable actual, manteniendo el resto del sistema intacto. Por su parte, si se requiriese, una mejora de navegabilidad y exploración de resultados (o incluso cambiar totalmente de tecnología de sitio web) no debería impactar en el código del modelo.

5. TRADUCCIÓN A MODELICA

El proceso de traducción consistió en el desarrollo de una versión del MML en el lenguaje Modelica (presentado en la Sección 1.2) utilizando como punto de partida la versión `MML86recup`. En la Sección 5.1 exponemos la estrategia utilizada para traducir las estructuras de regiones, fases y módulos, y luego en la Sección 5.2 describimos las metodologías utilizadas para la verificación de la versión Modelica con respecto a la versión Fortran.

5.1. Estrategia

5.1.1. Fases

Dada la magnitud del MML y la necesidad de acotar el proyecto, definimos el alcance de la traducción eligiendo cuáles facetas del modelo incluir, cuáles recortar y cuáles dejar como tareas para un trabajo futuro.

Decidimos incluir todas las facetas que fuesen parte de la *esencia* del modelo en el espíritu de la teoría del libro [HSC+76], recortar las facetas puramente implementativas de la época (como el *Monitor* o el graficador) y dejar para un paso futuro aspectos complejos de la Fase de Optimización. Esto obedece a que muy probablemente sea razonable reconsiderar la elección del algoritmo de optimización a la luz de la evolución de esta disciplina en los últimos 50 años. Sin embargo, destacamos que el método de Powell sin cálculo de derivadas implementado para la versión original es robusto y eficiente en el tratamiento de discontinuidades. En líneas generales no ha habido hasta la fecha avances notables en relación a estas características.

Entre los aspectos recortados se encuentra la Fase de Estimación para 1960[1], cuya inclusión en el modelo original estuvo fundamentada en la necesidad de estimar variables no disponibles en la época. Algunas variables son inicializadas en base a datos de 1960 y luego son utilizadas para estimar el resto de las variables para el mismo año. Decidimos evitar esta estimación y en cambio inicializar a todas las variables del modelo con sus valores de 1960 de la corrida estándar.

Decidimos incluir la Fase de Regresión por su sencillez de implementación al diferenciarse de la Fase de Proyección por sólo unas pocas ecuaciones: las encargadas de los cálculos de los exponentes *alfa* en las funciones Cobb-Douglas [CD28]. De manera análoga a las versiones de Fortran, en el MML Modelica se utilizan las ecuaciones de esta fase anualmente entre los años 1960 y 1970 y para los años siguientes simplemente se utiliza el valor de 1970. De manera similar, no es distinguida en el código y se lleva a cabo implícitamente en la Fase de Proyección.

El grueso del esfuerzo de traducción debió concentrase en la Fase de Proyección. Sus ecuaciones no son triviales de convertir desde una especificación en un lenguaje imperativo de programación a un lenguaje declarativo de modelado matemático, presentando numerosos obstáculos. Consideramos a este paso crucial para llevar el modelo un paso más cerca de una especificación puramente matemática. Esta tarea hubiese sido imposible sin el proceso de ingeniería inversa y documentación sistemática presentada la Sección 3.

[1] En la Sección 1.1.2.3 fueron introducidas las fases del modelo original.

El solapamiento de ecuaciones entre la Fase de Optimización y la Fase de Proyección es muy grande, por lo que ya están presentes muchas de sus ecuaciones, pero también presenta un comportamiento excepcional como la optimización ponderada con objetivos cuya implementación en Modelica no es trivial, y lo consideramos un problema importante en sí mismo. Es por esto que delegamos esta tarea para un trabajo futuro. En su ausencia, la simulación nunca transiciona desde la Fase de Proyección hacia a la Fase de Optimización.

Cada una de estas fases está compuesta por su propio conjunto de ecuaciones que describen su funcionamiento. No importa si dichos conjuntos comparten muchas ecuaciones, es suficiente con que haya una distinta para que haya una contradicción entre los conjuntos. Por ejemplo, si en una fase una variable x se define en base a la ecuación $x = y^2$ pero en otra se utiliza la ecuación $x = 2 * y$ entonces el compilador de ecuaciones de Modelica detecta una inconsistencia [2], ya que un sistema de ecuaciones no puede estar definido de manera ambigua y en este caso no sabría cuál ecuación utilizar para asignarle un nuevo valor a x. La forma correcta de modelar esta situación en Modelica es definir un *multiplexor* de fases encargado de activar y desactivar las fases en las transiciones. Su implementación debería ser bastante simple dado que se puede utilizar un condicional que elija qué ecuaciones utilizar dependiendo del año, pero dado que por ahora sólo tenemos las ecuaciones de la Fase de Proyección desde 1961 en adelante, decidimos no incluirlo para simplificar el modelo. La idea de dicho multiplexor queda sugerida en la Figura 5.1.

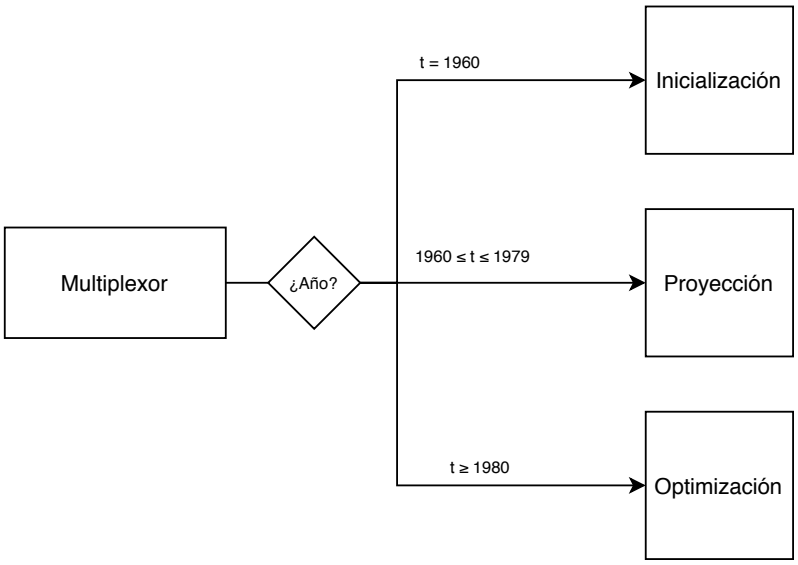


Figura 5.1: Idea de un futuro multiplexor de fases.

5.1.2. Regiones

Como vimos en la Sección 3, durante la Fase de Proyección todas las regiones comparten el mismo comportamiento, mientras que en Optimización algunos cálculos dependen de la región (como los de los costos de viviendas para Latinoamérica, África y Asia). Como decidimos no incluir la Fase de Optimización, la implementación de las regiones es simple al no haber comportamiento específico en ninguna de ellas, diferenciándose sólo en los valores de inicialización.

[2] Recordar el sistema de ecuaciones de Modelica explicado en la Sección 1.2

5.2. Testing

5.2.1. Tests de Unidad

En ciencias de la computación, el *testing* de unidad (*unit testing* en inglés) es un método de *testeo* por el cual se verifica el comportamiento individual y aislado de unidades de código, como pueden ser módulos, funciones, métodos, clases, etc. El objetivo es eliminar factores externos, como los otros módulos del sistema, para comprobar que las unidades de código atómicas cumplen con el *contrato* cuando se les provee como *inputs* valores que podrían recibir cuando se ejecuten en situaciones *reales*.

Para el *testing* de unidad de `MML20modelica`, decidimos comprobar que los módulos se comportasen de manera similar a los módulos de `MML86reing` no sólo para un año sino para todos los de la corrida estándar. Para ello, creamos un "escenario de testing" con una "fuente de datos" que alimenta a todos los módulos para que en cada año de la simulación sus variables de entrada estén definidas con los valores respectivos del mismo año de la corrida estándar de la versión en Fortran. Por ejemplo, si la variable "personas por familia" vale 3.5 en 1970 en `MML86reing`, esta fuente de datos alimentará con este valor al módulo Viviendas para 1970. Luego, el test verificaría que en base a esa entrada el módulo en `MML20modelica` dé los mismos resultados, bajo cierto grato de libertad, que el módulo original en `MML86reing`. La Figura 5.2 pretende dar una imagen ilustrativa de esto.

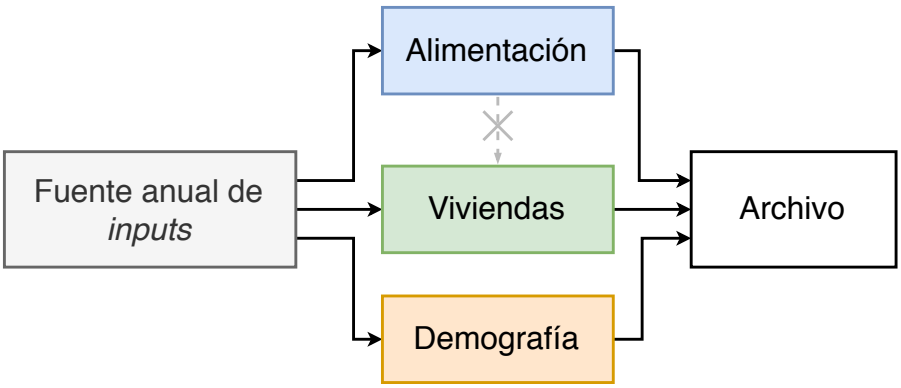


Figura 5.2: Imagen ilustrativa del *testing* de unidad

Como se puede ver en la imagen y con lo explicado en el párrafo anterior, el *testing* elegido consiste en simular un "escenario de prueba" en el cual las variables de entrada de los módulos están fijas y el único fin de sus variables de salidas es ser escritas entre los resultados, por lo que los módulos son independientes entre sí. Después de cada simulación comparamos los resultados de los módulos con los originales para ver si cada uno está funcionando de manera adecuada.

Una ventaja importante de este método es que al estar aislados, los módulos son indiferentes a adiciones o remociones de otros módulos. Esto facilita el "paso a paso" del pasaje de `MML86reing` a `MML20modelica` porque cada nuevo módulo podría ser agregado a este "escenario de testing" sin afectar al resto y de esta manera confirmar inmediatamente si su pasaje fue llevado a cabo correctamente.

Lamentablemente, por distintas refactorizaciones de código para mejorar la reusabilidad y legibilidad de las regiones, el "escenario de testing" se volvió inválido, requiriendo

esfuerzos extra para ponerlo operativo. Sin embargo, con los *tests* de integración explicados en la próxima sección, es suficiente para verificar el funcionamiento del sistema en su uso práctico.

5.2.2. Tests de Integración

El *testing* de integración (*integration testing* en inglés) corrobora que el comportamiento colectivo de los módulos es el esperado. En nuestro caso, lo utilizamos para *testear* que las corridas `MML20modelica` fuesen equivalentes a las corridas en `MML86reing` utilizando el escenario de la corrida estándar, sin tener que crear un modelo ad-hoc como tuvimos que hacer en el *testing* de unidad. La Figura 5.3 da una idea de cómo implementamos este tipo de *testing* mostrando como ahora los módulos están interconectados entre sí y sin incluir la "fuente de datos".

La manera de llevar a cabo el *testing* de integración fue muy similar al proceso de verificación presentado en la Sección 2.4, en el cual se buscaba verificar que `MML86recup` producía resultados equivalentes a `MML86scan`. En ambos casos utilizamos un mapa de calor como herramienta de comparación analizando visualmente las diferencias de las variables en cada año de la simulación y se utilizó una fórmula similar:

$$\text{error} = 100 * (\text{res(MML20modelica)} - \text{res(MML86reing)}) \ / \ \text{res(MML20modelica)} \tag{5.1}$$

A continuación se presentan los mapas de calor de todas las regiones, no sólo de Latinoamérica como en la Sección 2.4, que corresponden a corridas iniciando en 1960 y terminando en 2000. Como `MML20modelica` no incluye la Fase de Optimización, configuramos las corridas de `MML20reing` para nunca entrar en dicha fase y así hacer más justa la comparación. Excluimos la comparación de la fila de condiciones iniciales en el año 1960 al ser simplemente valores de inicialización, como mencionamos en la Sección 5.1, y lo denotamos con un color gris. Del mismo color se presentan algunas variables que no fueron incluidas en `MML20modelica` al ser utilizadas sólo durante la Fase de Optimización en el modelo original. Estas variables son `AL` (tierra cultivada), `FALU` (fracción de tierra potencialmente cultivable), `FERT` (producción de fertilizantes medida en 1000 toneladas), `REND` (rendimiento de la agricultura medida en toneladas por hectárea) `EXCAL` (exceso de calorías por día por persona), cuyas descripciones pueden encontrarse en el Apéndice A.

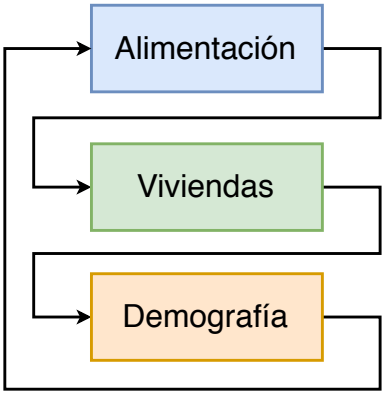


Figura 5.3: Imagen ilustrativa del *testing* de integración

	0-5	11-70	6-17	AL	BIRTHR	CALOR	CAPD(1)	CAPD(2)	CAPD(3)	CAPD(4)	CAPD(5)	CAPT	CHMOR	EAPOPR	EDUCR	ENROL	EXCAL	EXLIFE	FALU	FERT	GNP	GNPD(1)	GNPD(2)	GNPD(3)	GNPD(4)	GNPD(5)	GNPXC	GRMOR	HOUSER	HSEXFL	PERXFL	POP	POPR	PROT	REND	RLFD(1)	RLFD(2)	RLFD(3)	RLFD(4)	RLFD(5)	SEPOPR	TLF	TURBH	URBANR
1960																																											
1961	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1962	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1963	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1964	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1965	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1966	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1967	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1968	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1969	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1970	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1971	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1972	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1973	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1974	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1975	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1976	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1977	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1978	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1979	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1980	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1981	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1982	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1983	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1984	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1985	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
1986	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1987	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
1988	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1989	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1990	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1991	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0
1992	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1993	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1994	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1995	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1996	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1997	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1998	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
1999	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0
2000	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	1	0	0

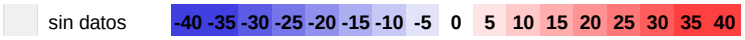


Figura 5.4: Mapa de calor del error porcentual entre MML20reing y MML20modelica, ambos para Fase Proyección en Países Desarrollados.

	0-5	11-70	6-17	AL	BIRTHR	CALOR	CAPD(1)	CAPD(2)	CAPD(3)	CAPD(4)	CAPD(5)	CAPT	CHMOR	EAPOPR	EDUCR	ENROL	EXCAL	EXLIFE	FALU	FERT	GNP	GNPD(1)	GNPD(2)	GNPD(3)	GNPD(4)	GNPD(5)	GNPXC	GRMOR	HOUSER	HSEXFL	PERXFL	POP	POPR	PROT	REND	RLFD(1)	RLFD(2)	RLFD(3)	RLFD(4)	RLFD(5)	SEPOPR	TLF	TURBH	URBANR
1960																																											
1961	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1962	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1963	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1964	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1965	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1966	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1967	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1968	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1969	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1970	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1971	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1972	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1973	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1974	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1975	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1976	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1977	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1978	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1979	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1980	0	0	0		0	0	0	0	0	0	0	0	0	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1981	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1982	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1983	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1984	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1985	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1986	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1987	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1988	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1989	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1990	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1991	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1992	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1993	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1994	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1995	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1996	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1997	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1998	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	-1	2	0	0
1999	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	0	0	0	0	0	1	0	0	-2	0		0	0	0	0	0	-1	2	0	0
2000	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	0	0	0	0	0	1	0	0	-2	0		0	0	0	0	0	-1	2	0	0

sin datos | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40

Figura 5.5: Mapa de calor del error porcentual entre MML20reing y MML20modelica, ambos para Fase Proyección en Latinoamérica.

	0-5	11-70	6-17	AL	BIRTHR	CALOR	CAPD(1)	CAPD(2)	CAPD(3)	CAPD(4)	CAPD(5)	CAPT	CHMOR	EAPOPR	EDUCR	ENROL	EXCAL	EXLIFE	FALU	FERT	GNP	GNPD(1)	GNPD(2)	GNPD(3)	GNPD(4)	GNPD(5)	GNPXC	GRMOR	HOUSER	HSEXFL	PERXFL	POP	POPR	PROT	REND	RLFD(1)	RLFD(2)	RLFD(3)	RLFD(4)	RLFD(5)	SEPOPR	TLF	TURBH	URBANR
1960																																											
1961	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1962	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1963	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1964	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1965	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1966	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1967	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1968	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1969	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1970	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1971	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	2	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1972	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	-1	0	2	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1973	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			1	0	0	-1	0	2	0	0	0	0	0	-3	0		0	0	0	0	0	-1	3	0	0
1974	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1975	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1976	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1977	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1978	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1979	0	0	0		0	0	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	0		0	0	0	0	0	-1	3	0	0
1980	0	0	0		0	1	0	0	0	0	0	1	-1	3	0	0		0			1	0	0	-1	0	2	1	0	1	1	0	-3	1		0	0	0	0	0	-1	3	0	0
1981	0	0	0		0	1	0	0	0	0	0	1	-1	3	0	0		0			1	-1	0	-1	0	2	1	0	1	1	0	-3	1		0	0	0	0	0	-1	3	0	0
1982	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			1	-1	0	-1	0	2	1	0	1	1	0	-3	1		0	0	0	0	0	-2	3	0	0
1983	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			1	-1	0	-1	0	2	1	0	2	1	0	-3	1		0	0	0	0	0	-2	3	0	0
1984	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			1	-1	0	-1	0	2	1	0	2	2	0	-3	1		0	0	0	0	0	-2	3	0	0
1985	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			1	-1	0	-1	0	2	1	0	2	2	0	-3	1		0	0	0	0	0	-2	3	0	0
1986	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			2	-1	0	-1	0	2	2	0	2	2	0	-3	1		0	0	0	0	0	-2	3	0	0
1987	0	0	0		0	1	0	0	0	0	0	2	-1	3	0	0		0			2	-1	0	-1	0	2	2	0	2	2	0	-3	1		0	0	0	0	0	-2	3	0	0
1988	0	0	0		-1	1	0	0	0	0	0	2	-1	3	0	0		0			2	-1	1	-1	-1	2	2	0	2	2	0	-3	1		0	0	0	0	0	-2	3	-1	0
1989	0	0	0		-1	1	0	0	0	0	0	2	-1	2	0	0		0			2	-1	1	-2	-1	2	2	-1	2	2	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1990	0	0	0		-1	1	0	0	0	0	0	3	-1	2	0	0		0			2	-1	1	-2	-1	2	2	-1	3	2	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1991	0	0	0		-1	1	0	0	0	0	0	3	-1	2	0	0		0			2	-1	1	-2	-1	2	2	-1	3	2	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1992	0	0	0		-1	1	0	0	0	0	0	3	-1	2	0	0		0			2	-1	1	-2	-1	2	2	-1	3	2	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1993	0	0	0		-1	1	0	0	0	0	0	3	-2	2	0	0		0			2	-1	1	-2	-1	2	2	-1	3	3	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1994	-1	0	0		-1	1	0	0	0	0	0	3	-2	2	0	1		0			2	-1	1	-2	-1	2	2	-1	3	3	0	-3	1		0	-1	-1	-1	-1	-2	2	-1	0
1995	-1	0	0		-1	1	0	0	0	0	0	3	-2	2	1	1		0			2	-1	1	-2	-1	2	2	-1	3	3	0	-4	1		0	-1	-1	-1	-1	-2	2	-1	0
1996	-1	0	0		-1	2	0	0	0	0	0	3	-2	2	1	1		0			2	-1	1	-2	-1	2	2	-1	3	3	-1	0	-4	2		0	-1	-1	-1	-1	-2	2	-1	0
1997	-1	0	0		-1	2	0	0	0	0	0	3	-2	2	1	1		0			2	-1	1	-2	-1	2	3	-1	3	3	-1	0	-4	2		0	-1	-1	-1	-1	-2	2	-2	0
1998	-1	0	0		-2	2	0	0	0	0	0	4	-2	2	1	1		0			2	-1	1	-2	-1	2	3	-1	4	3	-1	0	-4	2		0	-1	-1	-1	-1	-2	2	-2	0
1999	-1	0	0		-2	2	0	0	0	0	0	4	-2	2	1	1		0			2	-1	1	-2	-1	2	3	-1	4	3	-1	0	-4	2		0	-1	-1	-1	-1	-2	2	-2	0
2000	-1	0	0		-2	2	0	0	0	0	0	4	-3	2	1	1		0			2	-1	1	-2	-1	2	3	-1	4	3	-1	0	-4	2		0	-1	-1	-1	-1	-2	2	-2	0

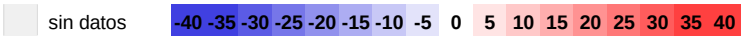


Figura 5.6: Mapa de calor del error porcentual entre MML20reing y MML20modelica, ambos para Fase Proyección en África.

	0-5	11-70	6-17	AL	BIRTHR	CALOR	CAPD(1)	CAPD(2)	CAPD(3)	CAPD(4)	CAPD(5)	CAPT	CHMOR	EAPOPR	EDUCR	ENROL	EXCAL	EXLIFE	FALU	FERT	GNP	GNPD(1)	GNPD(2)	GNPD(3)	GNPD(4)	GNPD(5)	GNPXC	GRMOR	HOUSER	HSEXFL	PERXFL	POP	POPR	PROT	REND	RLFD(1)	RLFD(2)	RLFD(3)	RLFD(4)	RLFD(5)	SEPOPR	TLF	TURBH	URBANR
1960																																											
1961	0	0	0		0	0	0	0	0	0	0	0	-1	3	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	3	0	0
1962	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1963	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1964	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1965	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1966	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-2	0		0	0	0	0	0	0	2	0	0
1967	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	0	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1968	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1969	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1970	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1971	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1972	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	0	2	0	0
1973	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1974	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1975	0	0	0		0	0	0	0	0	0	0	0	-1	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1976	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1977	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1978	0	0	0		0	0	0	0	0	0	0	0	0	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1979	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	0	0	1	0	0	0	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1980	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1981	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	0	0	-3	0		0	0	0	0	0	-1	2	0	0
1982	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1983	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1984	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1985	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	0	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1986	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			0	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1987	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1988	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1989	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1990	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1991	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1992	0	0	0		0	0	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	0		0	0	0	0	0	-1	2	0	0
1993	0	0	0		0	1	0	0	0	0	0	1	0	2	0	0		0			1	0	0	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	0	0
1994	0	0	0		0	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
1995	0	0	0		0	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
1996	0	0	0		0	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
1997	0	0	0		-1	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
1998	0	0	0		-1	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
1999	0	0	0		-1	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	0	-3	1		0	0	0	0	0	-1	2	-1	0
2000	0	0	0		-1	1	0	0	0	0	0	1	0	2	0	0		0			1	0	1	-1	0	1	1	0	1	1	-1	0	-3	1		0	0	0	0	0	-1	2	-1	0

sin datos | -40 | -35 | -30 | -25 | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40

Figura 5.7: Mapa de calor del error porcentual entre MML20reing y MML20modelica, ambos para Fase Proyección en Asia.

En los cuatro mapas de calor presentados la gran mayoría de las celdas resultaron error nulo. La región con menor incidencia de errores es la de Países Desarrollados con sólo algunos valores de `EAPOPR`(crecimiento de la Mano de Obra), `POPR` (crecimiento poblacional) y `TLF` (Mano de Obra total) afectados por discrepancias.

En el caso de Latinoamérica estas variables también presentaron diferencias pero en mayor magnitud, llegando algunas hasta $\pm 3\,\%$ de diferencia. También notamos que en los últimos años presentaron diferencias pequeñas las variables `CAPT` (Capital total), `HOUSER` (porcentaje de viviendas por persona) y `SEPOPR` (porcentaje de Mano de Obra secundaria).

La región de África fue la que más diferencias presentó y además de las variables previamente mencionadas, que en esta región tuvieron una discrepancia aún mayor, se suman varias más, lo que nos dejó como posible tarea de un futuro proyecto analizar la razón de estas anomalías.

Finalmente, el mapa de calor de Asia podría pensarse como un punto medio entre Latinoamérica y África, presentando algunas pocas diferencias más que el primero pero no al nivel elevado del segundo.

Los valores casi despreciables de estas diferencias y los resultados exactos para variables importantes como la población y Esperanza de Vida al Nacer, entre otras, nos dan la tranquilidad de que la traducción fue llevada a cabo satisfactoriamente y que la razón de las discrepancias pueden deberse incluso a diferencias de representación numérica entre lenguajes. Sin embargo, es de interés analizar a fondo la razón subyacente de estas diferencias entre `MML20reing` y `MML20modelica`, especialmente para África.

En la Sección C graficamos cada variable de cada región y analizamos el comportamiento de cada una.

6. MML MODELICA

En esta sección haremos foco en la implementación de `MML20modelica` dando una introducción a sus componentes en la Sección 6.1 y a la forma de simularlo en la Sección 6.2.

6.1. Paquetes (Packages)

Implementamos al modelo en Modelica dividiéndolo jerárquicamente en lo que el lenguaje llama como *packages* (paquetes) con el objetivo de independizar el código de los datos y que cada módulo sea lo más conciso posible. Cada paquete puede además contener paquetes internos sub-dividiendo sus responsabilidades. En OMEdit, estos paquetes son listados en el *libraries browser* como se puede ver a la izquierda en la Figura 6.1 y al abrirlos se los muestra en la sub-ventana principal, que en el caso de dicha figura se muestra el *block* Projection.

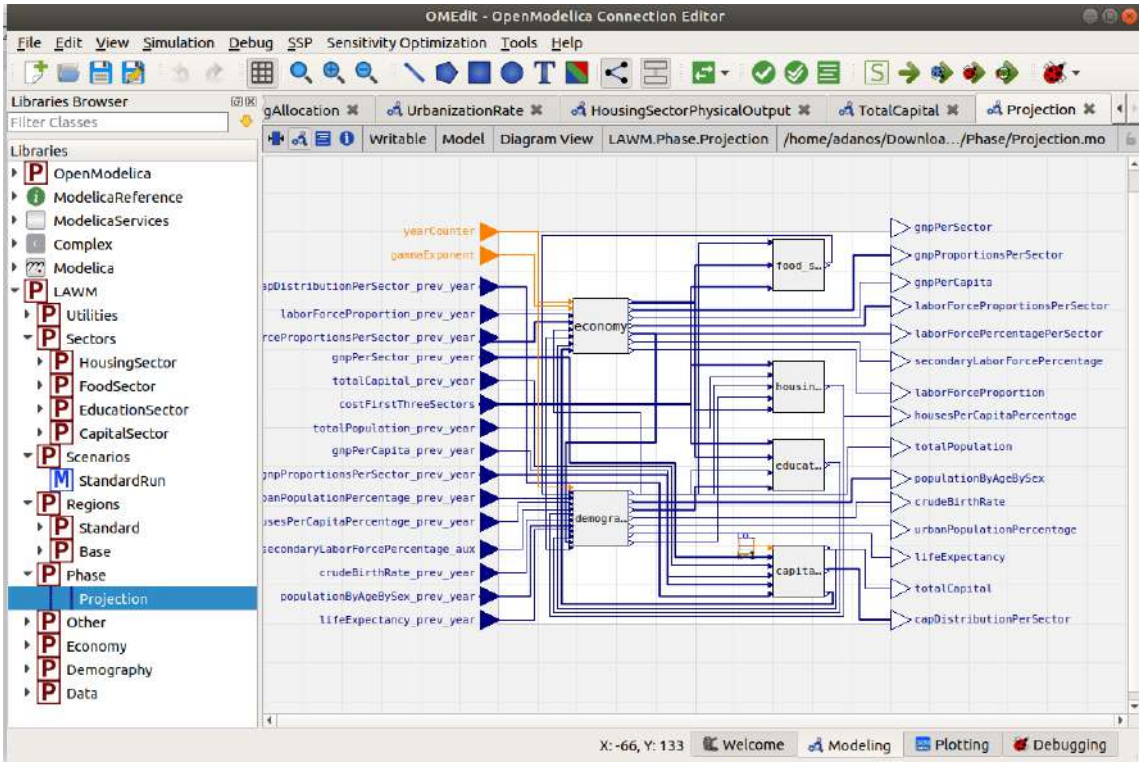


Figura 6.1: Captura de pantalla de OMEdit mostrando los paquetes en la sub-ventana de la izquierda y la vista gráfica del *block* Projection en la sub-ventana principal.

Los paquetes de más alto nivel son los siguientes:

- **Data:** Paquete utilizado en el *testing* de unidad, explicado en la Sección 5.2, para proveer a los distintos módulos los valores de las variables para cada año de la corrida estándar de Fortran. Hoy en día no es utilizado por incompatibilidades del

unit testing con cambios técnicos relacionados con las regiones. Sin embargo, decidimos dejarlo porque puede ser de utilidad en el futuro si es que se decide volver a implementar los *tests* de unidad.

- **Demography (Demografía):** Incluye el bloque principal de la demografía que utiliza internamente los bloques necesarios para calcular la población por rango etario, la fertilidad, mortalidad infantil, etc; de manera muy similar al modelo original. Es uno de los paquetes que hacen uso de los *algorithm*, explicados en la Sección 1.2.1, porque algunas secciones de su código original son bastante complejas y no son triviales sus pasajes desde un lenguaje imperativo hacia el paradigma de ecuaciones diferenciales.
- **Economy (Economía):** Contiene los cálculos de GNP (*Gross National Product*), en base a la función de *Cobb-Douglas*, y los cálculos de Mano de Obra. Incluye sólo las ecuaciones de la Fase de Proyección como vimos en las secciones anteriores.
- **Other (Otros):** Contiene operaciones auxiliares relacionadas con la aritmética o la interpolación de números.
- **Phase (Fase):** Contiene el *block* que representa a la Fase de Proyección donde se definen las distintas interacciones entre los distintos bloques (como la demografía, la economía, los sectores, etc) correspondientes a esta fase. Se explicará más en detalle este *block* en la Sección 6.1.1. Con respecto a la Fase de Inicialización mencionada en secciones anteriores decidimos no hacerla explícita como en el modelo original y se lleva a cabo en las regiones en sus inicializaciones de variables.
- **Regions (Regiones):** módulos nuevos representando cada uno de los bloques regionales con sus respectivas ecuaciones, parámetros y valores de inicialización. Los explicaremos en detalle en la Sección 6.1.2.
- **Scenarios (Escenarios):** Contiene a la *StandardRun* (corrida estándar) que instancia a las 4 regiones. Por cómo está implementado, se pueden crear nuevos escenarios en el futuro que incluyan más o menos regiones a definir por los usuarios.
- **Sectors (Sectores):** Los 5 sectores del modelo original ahora son representados en sus propios módulos inspirados en los diagramas de la Sección 3.
- **Utilities (Utilidades):** Contiene las interfaces de los distintos *blocks* para poder representar más fielmente los tipos de sus entradas y salidas. Entre ellos se encuentran los *inputs* y *outputs* de tipo *integer* (entero) y *real* (real).

6.1.1. Projection (Fase de Proyección)

Representamos a la Fase de Proyección con el *block* `LAWM/Phase/Projection.mo` y en ella se incluye la interacción entre los módulos correspondientes a 4 de los 5 sectores económicos (alimentación, viviendas, educación y capital) además de los de economía y demografía. Esta interacción es similar a la presentada en los diagramas de flujo de datos de la Sección 3.3

En la Figura 6.2 se puede ver la vista gráfica del *block* correspondiente a la Fase de Proyección. En ella, los *inputs* están agrupados a la izquierda y sus *outputs* a la derecha mientras que los submódulos mencionados en el párrafo anterior son representados como

rectángulos en el centro. También podemos notar que uno de los *inputs* del módulo de capital es la "constante 1" correspondiente al factor de disminución del comercio, que en esta fase nunca disminuye, por lo que es siempre 1 pero en una futura implementación de la Fase de Optimización esta entrada variará con el paso de los años.

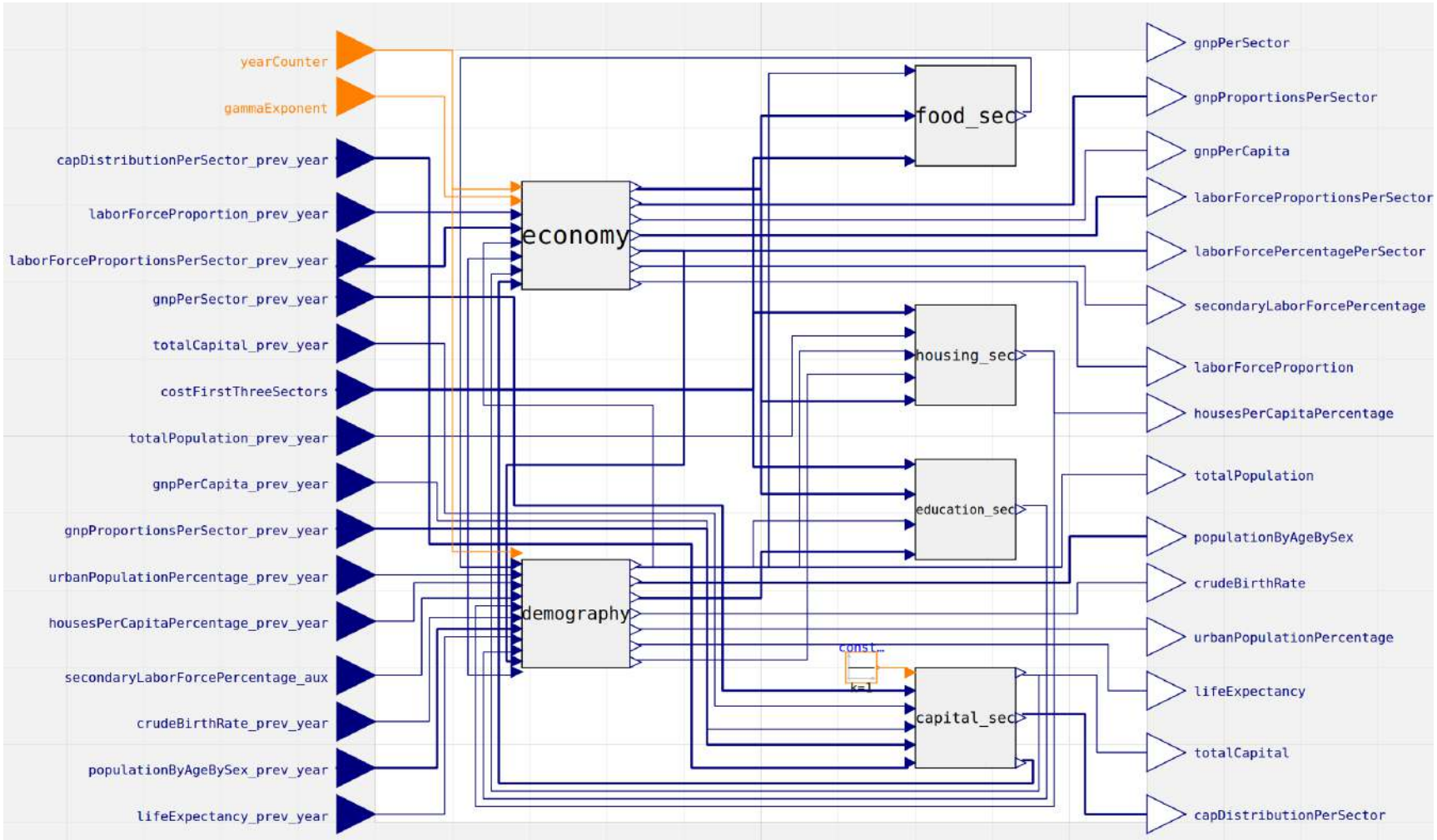


Figura 6.2: Vista gráfica del *block* `LAWM.Phase.Projection`

6.1.2. Regions (Bloques regionales)

Las regiones en esta versión del MML son representadas por *blocks* que definen sus comportamientos de principio a fin de la simulación. En OMEdit pueden ser accedidas desde el *libraries browser*, normalmente posicionado a la izquierda, y se presentan como en la Figura 6.3.

Figura 6.3: Las regiones como son mostradas en el *libraries browser* de OMEdit

Como mencionamos anteriormente, durante la Fase de Proyección las 4 regiones presentan el mismo comportamiento, por lo que todas extienden del *partial block* `Region`, del cual presentamos una simplificación en la Figura 6.4. En ella vemos que se instancian los objetos representando las variables de este año, las del anterior, y al *block* que representa a la Fase de Proyección, que es inicializado utilizando los datos de cada respectiva región.

Luego, en la sección de `initial equation` se inicializan sus variables regionales con los valores de 1960, llevando a cabo implícitamente la Fase de Inicialización. A continuación vemos la sección `equation`, que es donde se define el comportamiento general de la región y que está compuesta por un ciclo de años desde 1961 hasta el fin de la simulación. Dentro del ciclo, primero se le proveen los *inputs* al *block* de la Fase de Proyección, luego se guardan los outputs de dicho *block* en las variables regionales y por último se actualizan las variables del año anterior con sus respectivos valores.

Aunque en esta descripción le damos un orden cronológico, es puramente ilustrativo y debemos recordar que las ecuaciones en Modelica no se ven afectadas por el orden de

su declaración, ya que describen un sistema de ecuaciones matemáticas, no un algoritmo secuencial.

```
partial block Region
  RegionVariables             vars(n_sectors=n_sectors);
  RegionPreviousYearVariables prev_yr_vars(n_sectors=n_sectors);
  LAWM.Phase.Projection proj( /* inits and params from region */)
initial equation
  // Implicit "initialization phase": values for 1960:
  vars.totalCapital = inits.totalCapital_init;
  // (... other 1960 variables ...)
equation
    // Years loop
    when sample(1961, 1) then
      // Assign values to projection phase inputs
      proj.totalCapital_prev_year = prev_yr_vars.totalCapital_prev_year;
      // (... other projection phase inputs ...)

      // Get projection phase outputs
      vars.totalCapital = proj.totalCapital;
      // (... other projection phase outputs ...)

      // Assign previous years values
      prev_yr_vars.totalCapital_prev_year = pre(vars.totalCapital);
      // (... other "previous years" variables ...)
    end when;
end Region;

```

Figura 6.4: Simplificación de `Region`.

Cada región inicializa sus variables de 1960 con sus propios datos y a sus fases con sus propios parámetros, definidos en *records* que extienden del *partial record* `RegionInits` y *partial record* `RegionParameters` respectivamente. En la Figura 6.5 presentamos una versión simplificada de `LAWM.Regions.Standard.RegionInitsLatinAmerica`, que contiene los datos de inicialización de Latinoamérica, mientras que en la Figura 6.6 presentamos lo análogo para los valores de sus parámetros. En ambos casos, los valores pueden ser fácilmente modificados en sus respectivos *records* para pruebas simples o reciclados para la creación de nuevas regiones si se necesitan llevar a cabo pruebas más complejas.

```
record RegionInitsLatinAmerica
 extends LAWM.Regions.Base.RegionInits(
  totalCapital_init = 0.22590028E+12,
  gnpPerCapita_init = 0.37183728E+03,
  totalGNP_init     = 0.77509108E+11,
  agePyramid_init   = {0.40046604E+08, 0.62536276E+08, 0.14524165E+09},
  // other inits ...
 );
end RegionInitsLatinAmerica;

```

Figura 6.5: Simplificación de `RegionInitsLatinAmerica`.

```
record RegionParametersLatinAmerica
 extends LAWM.Regions.Base.RegionParameters(
  cultivatedArableLand1960             = 0.8163,
  gammaPerSector                       = {1.01, 1.01, 1.005, 1.01, 1.015},
  regressionPhaseYears                 = 11,
  FRS4SE                               = 0.42391,
  laborForceVariationInProjectivePhase = -0.00077,
  // other parameters and their values  ...
 );
end RegionParametersLatinAmerica;

```

Figura 6.6: Simplificación de `RegionParametersLatinAmerica`.

6.1.3. Ejemplo de ecuaciones para el Paquete de Mano de Obra

En la Figura 6.7 mostramos las ecuaciones incluidas en el paquete de Mano de Obra cuyo *path* en Modelica es `LAWM.Economy.LaborForce`, del cual se puede deducir que su ruta en el sistema de archivos es `LAWM/Economy/LaborForce.mo`.

Vemos que el paquete incluye dos secciones `block` con dinámica diferenciada, el primero es el bloque `LaborForceProjection` que comienza en la línea 5 y termina en la línea 36 y es el encargado de calcular las variables de Mano de Obra sectoriales durante la fase de proyección. El segundo es el bloque `SecondaryLaborForcePercentage`, que comienza en la línea 38 y termina en la línea 48 y es el encargado de calcular el porcentaje de Mano de Obra secundaria.

Cada uno tiene su zona de definición de parámetros, entradas y salidas y su zona de definición de ecuaciones en la sección `equation`. Por ejemplo, en el caso de `LaborForceProjection` vemos cómo laborForceProportionsPerSector[] se calcula de manera diferenciada para el *Food Sector* que para los 4 restantes (filas 26 a 33). En el bloque de `SecondaryLaborForcePercentage` vemos cómo para calcular el porcentaje de Mano de Obra secundaria se hace uso de una sola ecuación utilizando los valores del parámetro FRS4SE[1] y las Mano de Obra sectoriales correspondientes a Otros Bienes y Bienes de Capital.

Para cambiar el procedimiento de, por ejemplo, el cálculo de Mano de Obra secundaria simplemente es necesario modificar o agregar ecuaciones en el bloque mencionado.

Esta es una muestra simplificada de como es posible evolucionar de manera modular y sencilla diferentes versiones de las ecuaciones que definen el comportamiento de diferentes módulos del modelo.

[1] Parámetro utilizado como estimación de la proporción de Mano de Obra secundaria del sector Otros Bienes

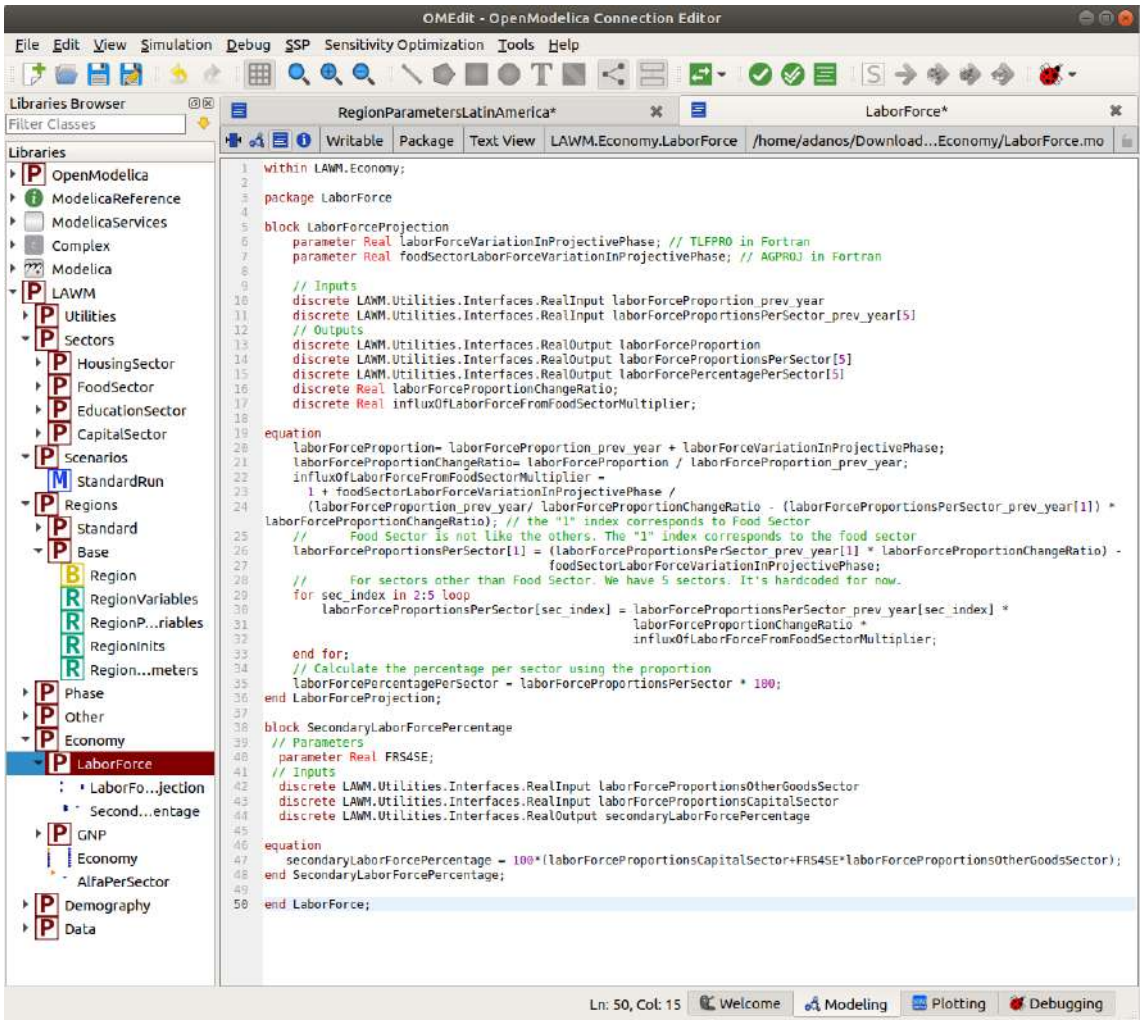


Figura 6.7: Ecuaciones del paquete de Mano de Obra

6.2. Simulación

La versión `MML2Omodelica` puede simularse abriendo el archivo `LAWM/package.mo` en OMEdit, incluido acompañando este documento o en un repositorio git en BitBucket. Luego se abre el modelo `LAWM.Scenarios.StandardRun` que corresponde a la corrida estándar del MML, y se simula de manera similar a la utilizada en la Sección 1.2 definiendo a 1960 y 2000 como *startTime* y *stopTime* respectivamente[2].

El programa OMEdit es muy útil tanto para desarrollar modelos como para graficar rápida e intuitivamente los resultados de las simulaciones del MML. En la Figura 6.8 vemos el graficador interactivo que permite elegir qué variables mostrar utilizando el *variables browser* de la derecha (incluido un buscador en la parte superior de éste) y el gráfico propiamente dicho en la sub-ventana principal, en el cual podemos ver detalles de valores individuales de las curvas graficadas. Analizando la información presentada, vemos cómo hay una explosión de la población en Asia, logrando para 2000 un valor cuatro veces mayor que el de Países Desarrollados, la segunda región con más población para ese año. Esto es compatible con lo que ocurrió en el mundo en los años simulados, incrementándose más aún la diferencia en los 21 años siguientes hasta hoy.

En la Figura 6.9, graficamos el capital total para las mismas regiones, seleccionando de nuevo en el *variables browser* las variables deseadas. En este caso, también hay una región que se aleja del resto, pero ahora es Países Desarrollados y su crecimiento es aún mayor y para el año 2000 alcanza un capital total varias veces superior al de Asia. Este resultado fue una de las bases para las políticas propuestas por los investigadores del MML, que fundamentaron que este crecimiento acelerado del capital en los Países Desarrollados podría ser utilizado para ayudar económicamente a las regiones de África y Asia para intentar reducir la brecha mostrada en el gráfico.

En conclusión, el programa OMEdit sintetiza en un mismo entorno facilidades poderosas para realizar ciclos iterativos de modelado, simulación y análisis gráfico de resultados.

[2] La actualización de OMEdit a fines de 2020 generó una incompatibilidad parcial con el MML. Para simular el MML sin errores en esta nueva versión es necesario elegir la opción de *old frontend* que nos da a elegir el programa cuando simulamos un modelo por primera vez

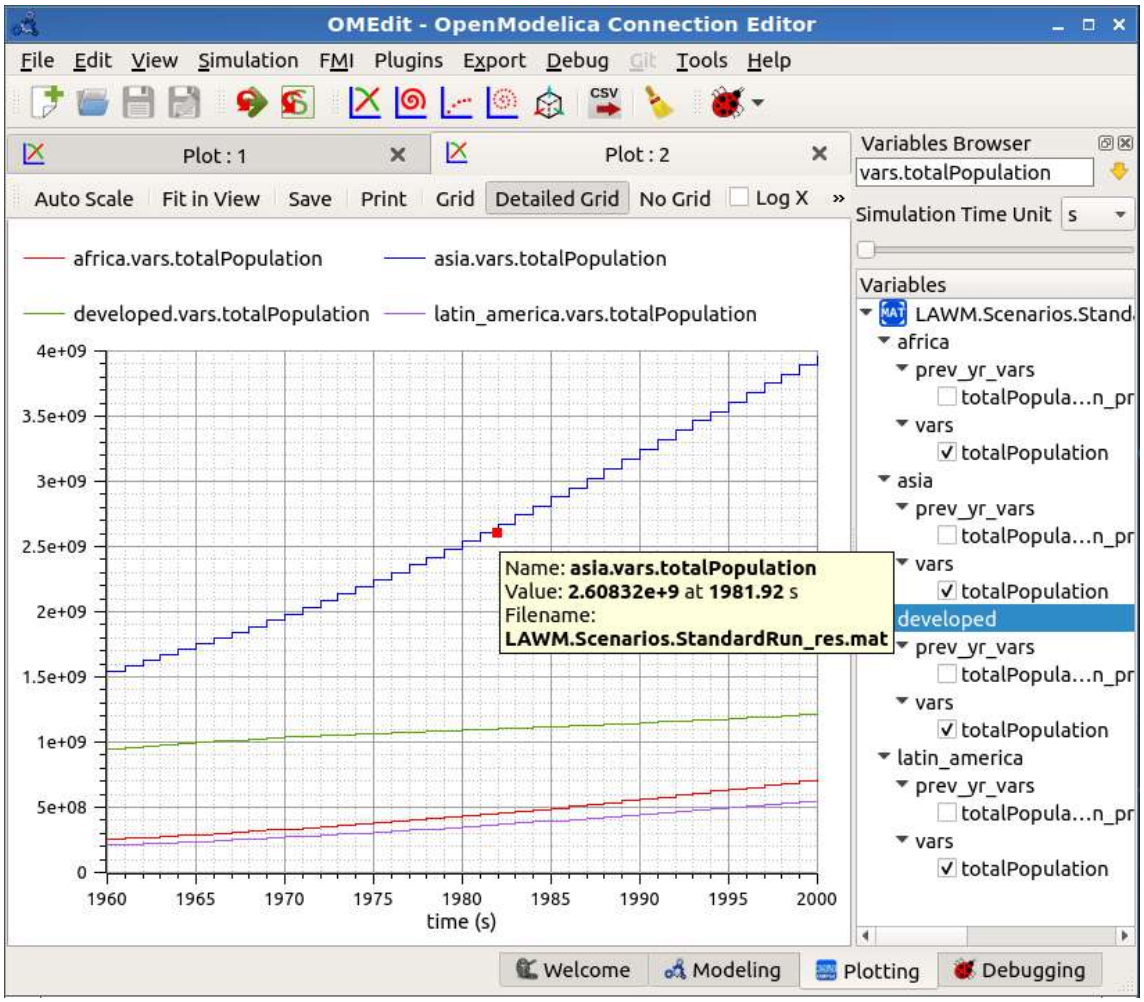


Figura 6.8: Población de las 4 regiones para la simulación estándar de `MML20modelica`

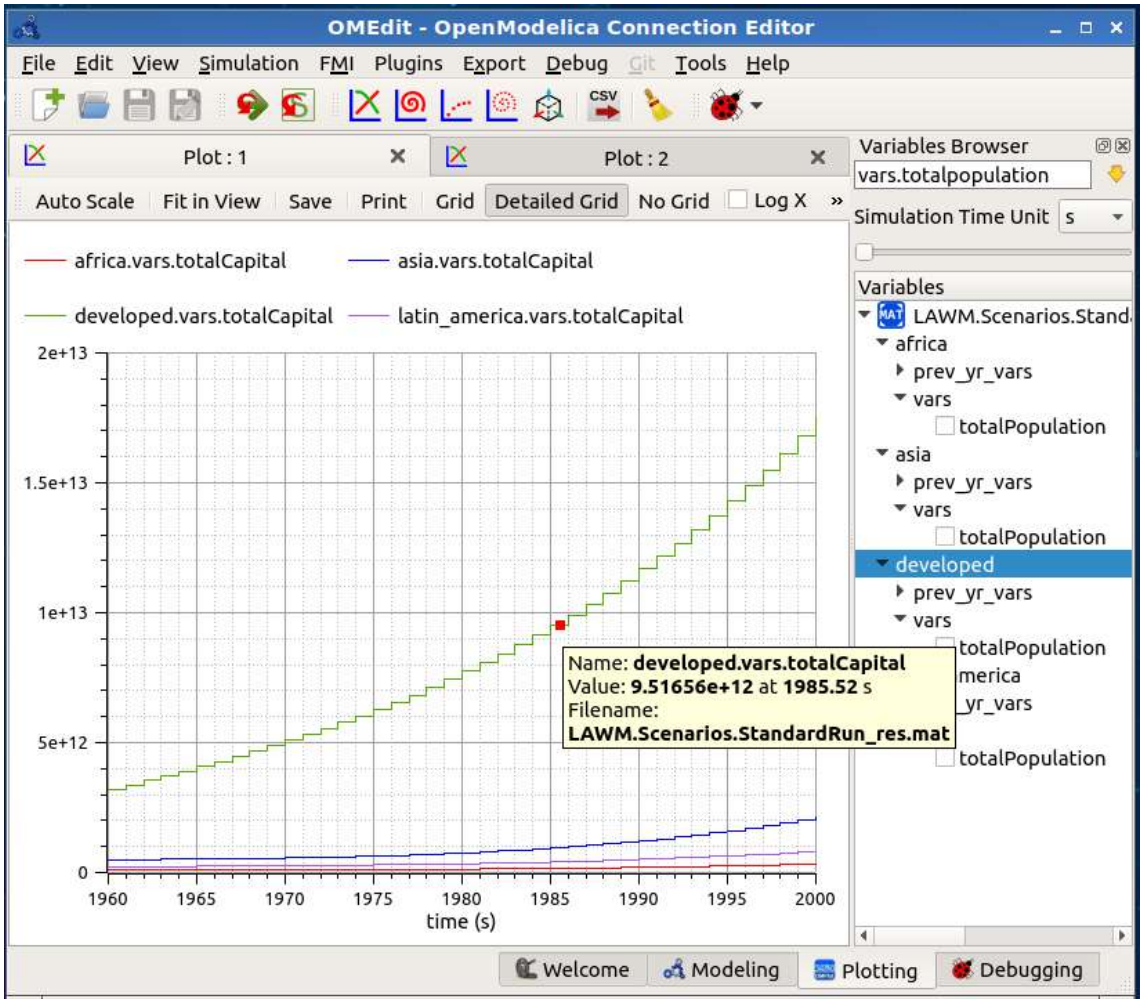


Figura 6.9: Capital total de las 4 regiones para la simulación estándar de `MML20modelica`

7. CONCLUSIÓN

7.1. Discusión del proceso de traducción

La traducción del modelo desde Fortran hacia Modelica fue uno de los objetivos más importantes de este proyecto, por ser este último un lenguaje más *amigable* con no-programadores al ser un lenguaje de modelado de alto nivel, a diferencia del primero que es un lenguaje imperativo de propósito general. Sin embargo, sentimos que, a diferencia de otros lenguajes modernos como Python, Go o Ruby, no ofrece algunas herramientas que facilitan el diseño de código, como manejo y manipulación de archivos CSVs, operaciones de programación funcional, polimorfismo entre objetos de distintas jerarquías de clases, representación intuitiva de datos, etc. Éste fue uno de los obstáculos que sentimos que necesitó más inversión de tiempo que lo planeado.

Más allá de la falta de *features* extra, en el lenguaje no es trivial el cambio de paradigma desde imperativo a ecuaciones y nos vimos obligados a omitirlo en algunas secciones, como en algunos *blocks* del módulo de demografía, teniendo que recurrir a los *algorithm* que ofrece Modelica. Esto no fue ideal pero tampoco demasiado grave, dado que la gran mayoría del modelo pudo ser traducida sin problemas y las secciones que no lo fueron igual fueron refactorizadas para dejarlas lo más limpias posible.

7.2. Objetivos alcanzados

Aunque no hayamos cumplido con todos los objetivos propuestos al principio de este trabajo, los logros alcanzados no fueron menores y construyeron los cimientos para todo futuro trabajo sobre el Modelo Mundial Latinoamericano, ayudando a que pueda salir de la oscuridad y que sea objeto de estudio por quién lo desee. El modelo recuperado en Fortran es un tesoro histórico que además puede ser utilizado para correr simulaciones fieles a las originales de hace 40 años. Por otro lado, el modelo traducido a Modelica es indispensable para investigadores sin antecedentes en programación y, junto a la nueva documentación producto del proceso de ingeniería inversa, son fundamentales para un futuro proyecto que tenga la ambición de completar el pasaje a este lenguaje.

Finalmente, aunque no fue un objetivo original al planear el proyecto, la página web construida es una herramienta que ahorra a investigadores la compilación y ejecución de simulaciones, proveyendo la posibilidad de correr simulaciones y guardar resultados desde cualquier parte del mundo de una manera intuitiva y elegante.

7.3. Trabajo a futuro

Entre los obstáculos de esta investigación que no pudieron ser resueltos y que nos parecen importantes atacar en un futuro proyecto se encuentran la traducción de la Fase de Optimización, pospuesta por su complejidad, y la adaptación de la corrida del testing de unidad, que quedó incompatible con la última implementación de las regiones. El primero es un aspecto fundamental del proyecto, dado que dicha fase es donde se lleva a cabo la búsqueda de la asignación óptima del GNP que cumpla con los requerimientos impuestos

por el usuario, y la segunda porque al tener tests de unidad los errores son mucho más fáciles de encontrar que sólo con tests de integración.

También nos parece conveniente llevar a cabo más análisis enfocados en las diferencias que restan ser explicadas, tanto entre `MML86scan` y `MML86recup` como entre `MML86reing` y `MML20modelica`. El primero tendría como objetivo mejorar la adhesión del modelo recuperado al original y el segundo buscaría confirmar que las diferencias son causadas por detalles técnicos entre un lenguaje y otro y no por por errores involuntarios durante la traducción.

Más allá de mejoras al modelo en sí, se pueden utilizar los resultados de este proyecto como base de estudio en base a distintos ejes, como por ejemplo los siguientes

- **Teorías subyacentes.** Cada submódulo, sea de alimentación, vivienda, educación, economía, etc., basa su comportamiento en alguna teoría elegida por el especialista respectivo de cada disciplina del proyecto original. Podría experimentarse con teorías alternativas y el efecto de éstas sobre el comportamiento del modelo.

- **Rutina de optimización.** El modelo tiene incorporada una rutina de optimización con objetivos ponderados [Pow64] para buscar año a año la asignación de Producto Nacional Bruto más cercana a los intereses del usuario, como, por ejemplo, darle mayor prioridad a la educación que a la construcción de viviendas, o viceversa. Se podría experimentar con el uso de rutinas de optimización alternativas en `MML20reing` y, en el caso de haber completado la traducción del modelo, también en `MML20modelica`.

- **Datos.** La disponibilidad de datos es distinta hoy que hace 40 años. Por ejemplo, algunos datos con los que se inicializa el modelo en 1960 son aproximaciones dado que no estaban disponibles sus mediciones durante el desarrollo del MML. Hoy se puede acceder con relativa mayor facilidad a datos de ese entonces y también más recientes, con lo que se podría experimentar inicializando el modelo con datos alternativos de 1960, o hasta adaptar el modelo para arrancar en años más cercanos a la fecha actual.

En definitiva, estamos muy conformes con los avances logrados en esta tesis y pensamos que este es un primer paso necesario para volver a colocar al Modelo Mundial Latinoamericano en la mesa de los modelos globales relevantes para apoyar la planificación y la toma de decisiones en política pública.

Finalmente, este ejercicio de recuperación es un homenaje a aquel sueño que comenzó con un grupo de investigadores en el verano de 1972 en Bariloche y fue abruptamente interrumpido por un proceso de violencia institucional representante de valores exactamente opuestos a los que se pusieron en juego al concebir el tipo de sociedad igualitaria al que aspiró, de manera utópica, el Modelo Mundial Latinoamericano.

Apéndice

A. ÍNDICE DE VARIABLES Y PARÁMETROS

En esta sección presentamos información sobre las variables y parámetros del modelo. Esta información fue extraída del módulo `DICTIO` del código original, razón por la cual está en inglés, con cambios mínimos nuestros como arreglos de *typos*.

En la Figura A.1 se describen los parámetros proveyendo su descripción, valores mínimos y máximos (donde corresponda), valor por defecto y unidad. Las variables descritas en la Figura A.2 son las incluidas en los resultados originales y las agrupamos según su categoría en dichos resultados. De ellas se provee su descripción y unidad.

	Parameter	Description	Min	Max	Default	Unit
GENERAL	FCOST	Cost of one tonne of fertilizer	0	0	769230.8	dollars/ton
	IAID	Enable economic aid			False	boolean
	KPROJ	Year at which the macroeconomic optimization starts	1980	1980	1980	year
	KTRADE	Year at which the balance of payments reaches the equilibrium (KPROJ<=KTRADE<=KSTOP)	1990	2050	2000	year
	KSTOP	Last year of the run (First year is 1960)	1990	2050	2000	year
	NHIST	If zero prints no histograms of the age structure by sex. If nonzero prints the histograms every NHIST years	0	1000	0	N/A
	TRAID	Maximum proportion of the GNP of Developed region allocated to aid for Africa and Asia	0.01	0.1	0.02	proportion
	WTH	Hierarchy of relative weights of importance (for each sector of the economy) when all restrictions can not be met at a given year by the optimization	2	14	[6.0, 8.0, 6.0, 6.0, 6.0, 6.0, 4.0, 4.0, 6.0, 6.0, 6.0, 2.0, 4.0, 4.0, 7.0, 4.0, 4.0, 4.0, 8.0, 6.0, 4.0, 6.0, 8.0, 6.0, 2.0, 1.0, 14, 14, 14, 14]	N/A
REGIONAL	CALMX	Maximum consumption of calories per day per person	2600	3200	3200 (developed) 3000 (other)	calories
	COSMAX	Maximum cost of the built square meter	5	20	10 (developed) 7 (other)	dollars
	GAMMA	Coefficients of technological progress by sector	1	1.1	[1.01, 1.01, 1.005, 1.01, 1.015]	coefficient
	KTECST	Year at which technological progress stops	1990	3000	3000	year
	NHCGAP	Number of years (after basic needs are satisfied) at which the cost of building in developing countries is the same as that of the developed ones	20	100	40	years
	NQH34	Number or years in which Africa and Asia are expected to reach the desired level of housing of Latinamerica (only Asia and Africa have effect)	10	50	0 (Developed & L.A.) 20 (Africa & Asia)	years
	NRESER	Desired food stock	100	730	365	-
	NSPGAP	Same as NHCGAP but for the level of land space per person	10	100	40	years
	SPAMAX	Maximum space per person	10	100	30	m2
	SPXPER	Desired space per person	5	50	30 (developed) 10 (L.A.) 7 (Africa & Asia)	m2
	UPTOIN	Maximum fraction of the GNP allocated to Sector 5 (Capital goods)	0.1	0.35	0.25	proportion

Figura A.1: Parámetros.

	Variable	Description	Unit
POPULATION INDICATORS	POP	Total population	persons
	POPR	Rate of population growth	percentage
	EXLIFE	Life expectancy at birth	years
	GRMOR	Crude mortality rate per 1000 inhabitants	annual deaths/1000 pers.
	BIRTHR	Crude birth rate per 1000 inhabitants	annual births/1000 pers.
	CHMOR	Child mortality rate per 1000 inhabitants	annual child deaths/1000 pers.
BASIC NEEDS INDICATORS	CALOR	Calories consumption per day per capita	calories/(day*person)
	PROT	Proteins consumed per day per person (in grams)	proteins/(day*person)
	HSEXFL	Fraction of families having a suitable house	fraction of families
	GNPXC	GNP per capita	dollars/person
	ENROL	Perc. of pop. between 7 and 18 which is matriculated	percentage
	EDUCR	Percentage of total population which is matriculated	percentage
LABOR FORCE	EAPOPR	Rate of change of economically active population	percentage
	TLF	Total labor force	persons
	RLFD(1)	Fraction of population in sector 1	percentage
	RLFD(2)	Fraction of population in sector 2	percentage
	RLFD(3)	Fraction of population in sector 3	percentage
	RLFD(4)	Fraction of population in sector 4	percentage
	RLFD(5)	Fraction of population in sector 5	percentage
CAPITAL SECTOR AND SOME AGE STRUCTURES	CAPT	Total capital	dollars
	CAPD(1)	Capital distribution for sector 1	proportion
	CAPD(2)	Capital distribution for sector 2	proportion
	CAPD(3)	Capital distribution for sector 3	proportion
	CAPD(4)	Capital distribution for sector 4	proportion
	CAPD(5)	Capital distribution for sector 5	proportion
	0-5	Population for ages 0-5 (from variable PYRAM)	persons
	6-17	Population for ages 6-17 (from variable PYRAM)	persons
	11-70	Population for ages 11-70 (from variable PYRAM)	persons
FOOD SECTOR	AL	Cultivated arable land	1000 ha
	EXCAL	Excedent of calories	calories
	FERT	Production of fertilizers (1000 tons)	1000 tons
	REND	Yield in agriculture (TONS/HA)	tons/ha
	FALU	Fraction of potentially arable land	proportion
OTHER INDICATORS	URBANR	Urban population	percentage
	TURBH	Total urbanization rate (HA/YEAR)	ha/year
	SEPOPR	Percentage of secondary labor force	percentage
	HOUSER	100 * HOUSES / POPULATION	100*houses/person
	PERXFL	People per family (average family size)	persons/family
GNPD DISTRIBUTION	GNP	Total GNP	dollars
	GNPD(1)	GNP distribution for sector 1	proportion
	GNPD(2)	GNP distribution for sector 2	proportion
	GNPD(3)	GNP distribution for sector 3	proportion
	GNPD(4)	GNP distribution for sector 4	proportion
	GNPD(5)	GNP distribution for sector 5	proportion

Figura A.2: Descripción de las variables incluidas en los resultados

B. DIAGRAMAS DEL LIBRO "CATÁSTROFE O NUEVA SOCIEDAD"

B.1. Flujo de datos

En la Sección 3.3.2.2 se presentaron los diagramas de flujo de datos resultantes de la ingeniería inversa del código recuperado. Algunos de ellos exhibían información similar a la presentada en el libro original [HSC$^+$76, HSC$^+$04], diferenciándose en que los de esta tesis llevan un enfoque más implementativo mientras que los del libro uno más teórico.

La sintaxis es distinta entre ambos, siendo la del libro más simple y la de la tesis más detallista, y además puede que las variables se llamen de manera distinta. Por ejemplo la variable "Población económicamente activa en la agricultura" de la Figura B.3, extraída del libro, corresponde a la variable "Propor. Mano de Obra: Alimentación" de la Figura 3.13 de esta tesis. Más allá de estas diferencias, no hay información contradictoria entre ambos y se complementan entre sí.

En la Figura B.1 podemos ver las relaciones entre los diagramas. Los casos particulares son los de Flujo del Modelo Matemático, en el cual el diagrama del libro muestra información contenida en varios diagramas de esta tesis, Demografía, donde en la tesis hay dos figuras por limitaciones de espacio contra una del libro, y los de Alimentación, donde en la tesis hay un diagrama y en el libro hay cinco. La justificación de este último es que el libro hace mucho énfasis en dicho sector comparado a los otros, como Viviendas y Educación, y por eso también presentan diagramas de sus subsectores de Agricultura, Ganadería y Pesquería.

Módulo	**Tesis**	**Libro**	
	Figura	**Figura**	**Capítulo**
Flujo del Modelo Matemático	1.6	8	Capítulo 4: El Modelo Matemático
Demografía	3.12 y 3.13	9	Capítulo 5: Demografía y Salud
Alimentación	3.9	10, 13, 14, 15 y 16	Capítulo 6: Alimentación
Viviendas	3.10	17	Capítulo 7: Viviendas
Educación	3.11	18	Capítulo 8: Educación

Figura B.1: Relación entre las figuras de flujo de datos de esta Tesis y las del libro original.

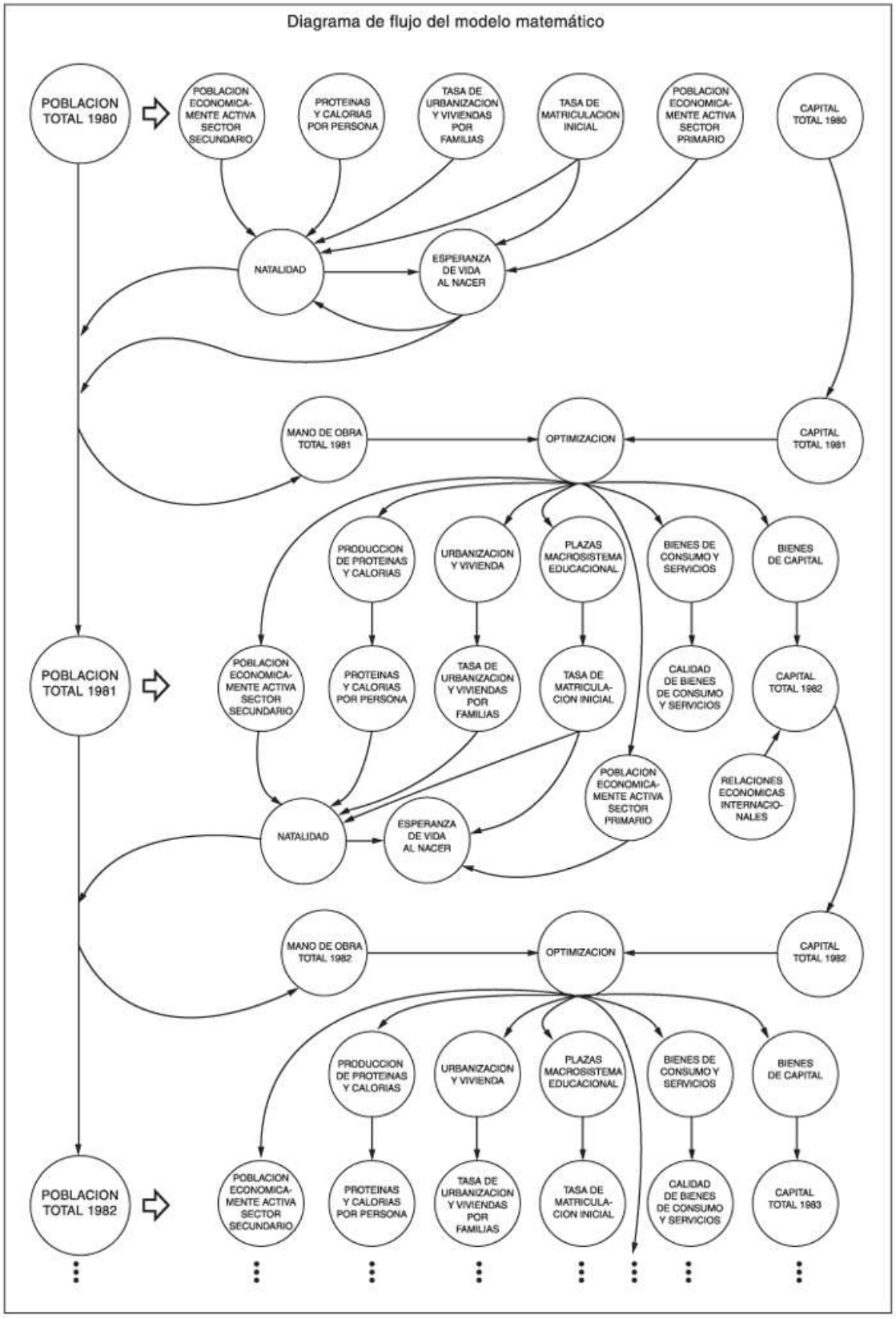
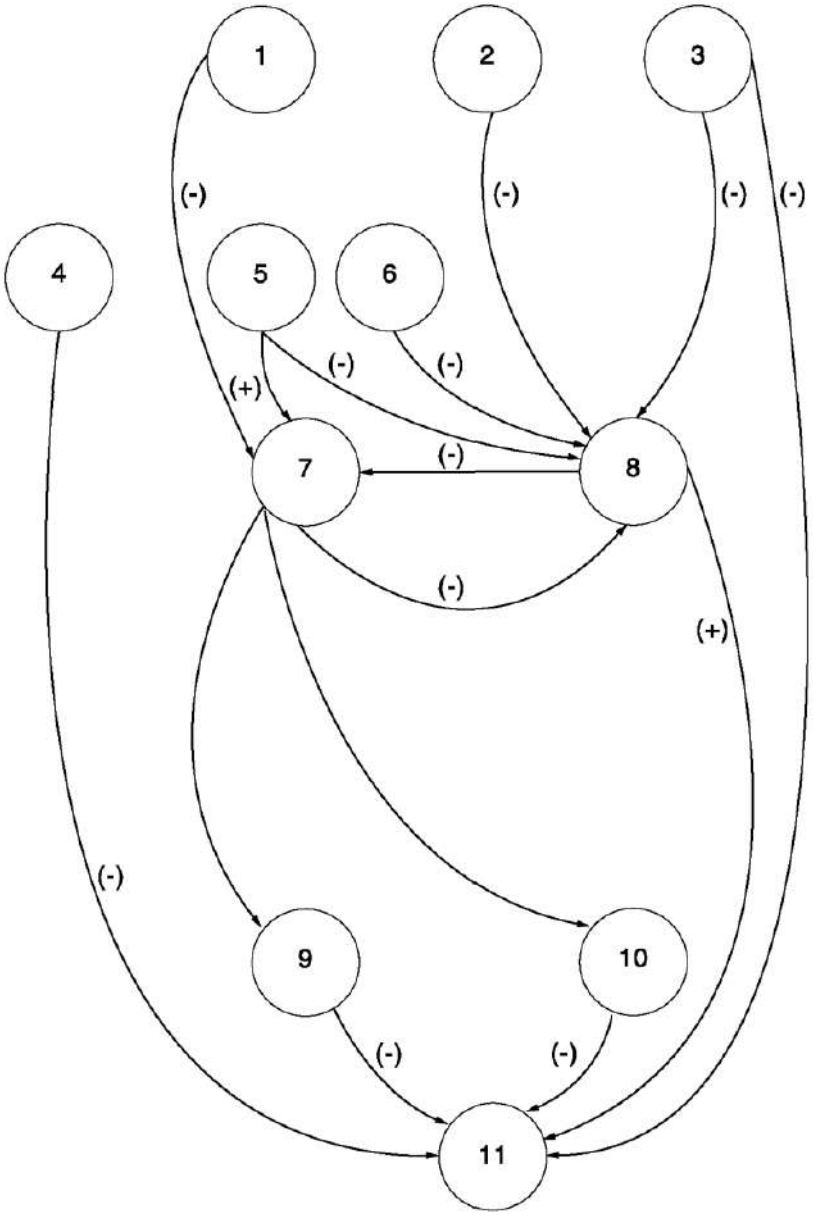


Figura B.2: Flujo del Modelo matemático. Figura 8 en el libro original.



Variables usadas para construir el submodelo de población, y sus relaciones.

1. Población económicamente activa en la agricultura
2. Población económicamente activa sector secundario
3. Viviendas por familias
4. Urbanización
5. Matriculación
6. Calorías
7. Esperanza de vida al nacer
8. Natalidad
9. Mortalidad infantil
10. Mortalidad bruta
11. Personas por familia

Figura B.3: Flujo de datos: Demografía. Figura 9 en el libro original.

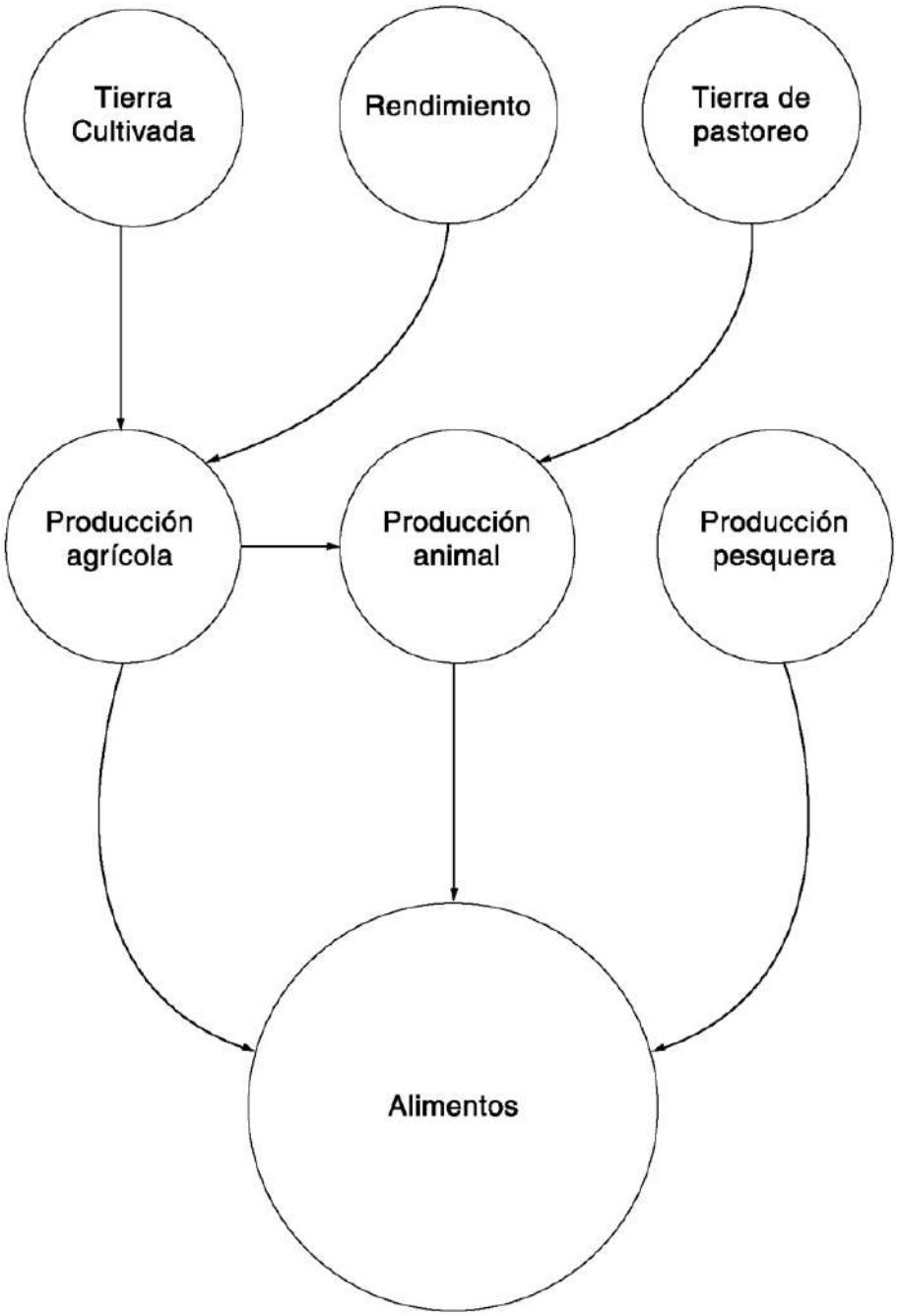


Figura B.4: Principales divisiones del sector alimentación. Figura 10 en el libro original.

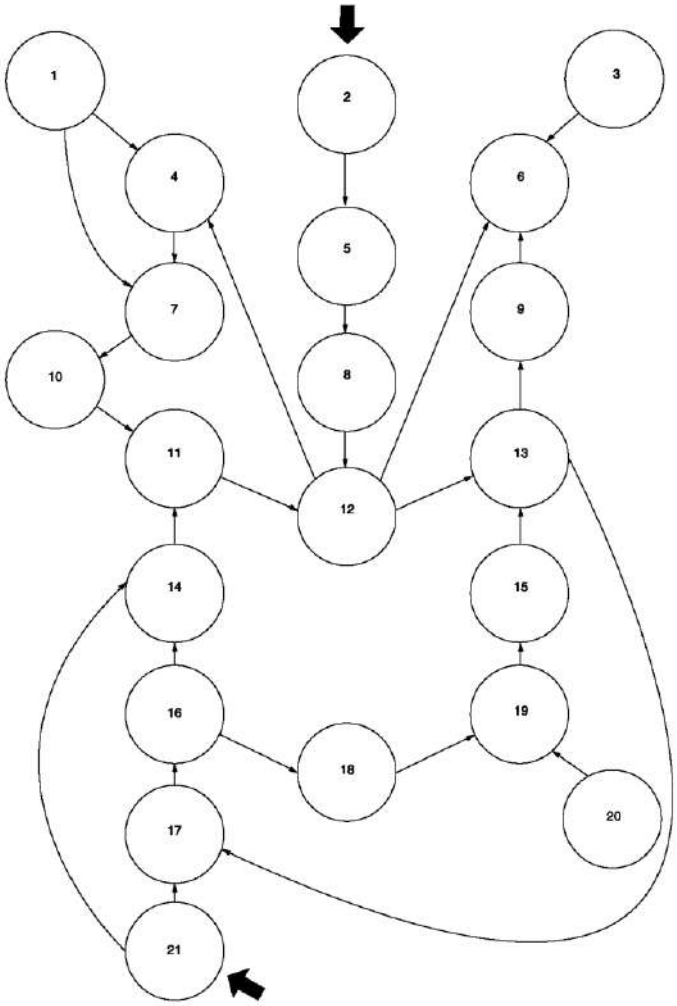


Diagrama de flujo del subsector agricultura

1. Tierra potencialmente cultivable
2. Tasa de urbanización total
3. Pérdidas por procesamiento
4. Tierras potencialmente arables todavía no cultivadas
5. Tasa de urbanización de la tierra arable
6. Producción de alimentos del sector agrícola
7. Fracción remanente de tierras potencialmente cultivables
8. Tasa de degradación de tierras cultivables
9. Rendimiento
10. Costo unitario de colonización de tierras
11. Tasa de colonización de tierras
12. Tierra arable cultivada
13. Fertilizantes por Ha.
14. Capital disponible para colonización de tierras
15. Fertilizantes disponibles
16. Capital para insumos agrícolas
17. Fertilizantes
18. Capital para inversión en producción de nuevos fertilizantes
19. Fertilizantes adicionales producidos cada año
20. Costo unitario de nuevos fertilizantes
21. Capital para agricultura

Figura B.5: Flujo de datos: Subsector Agricultura. Figura 13 en el libro original.

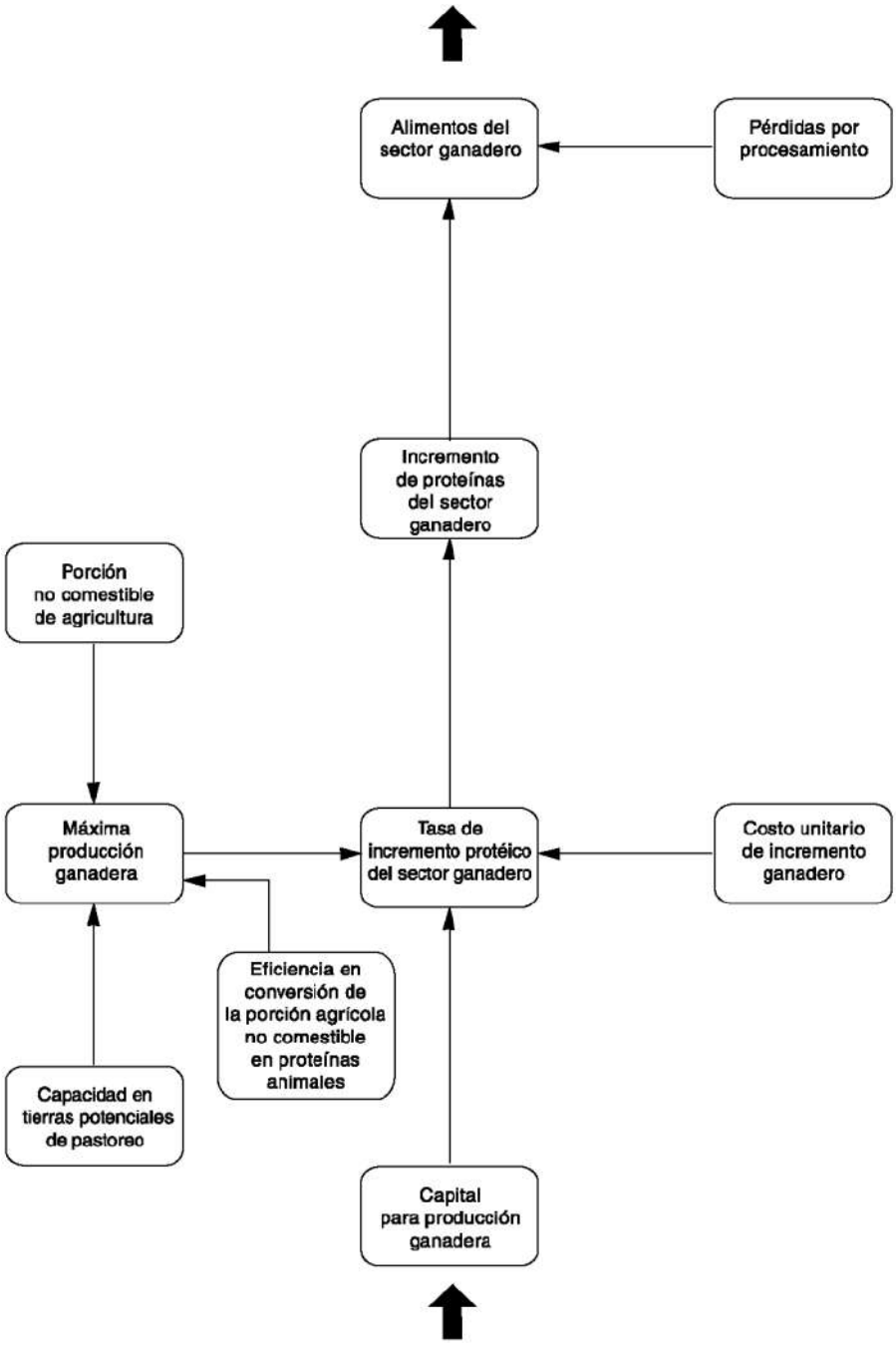


Figura B.6: Flujo de datos: Subsector Ganadería. Figura 14 en el libro original.

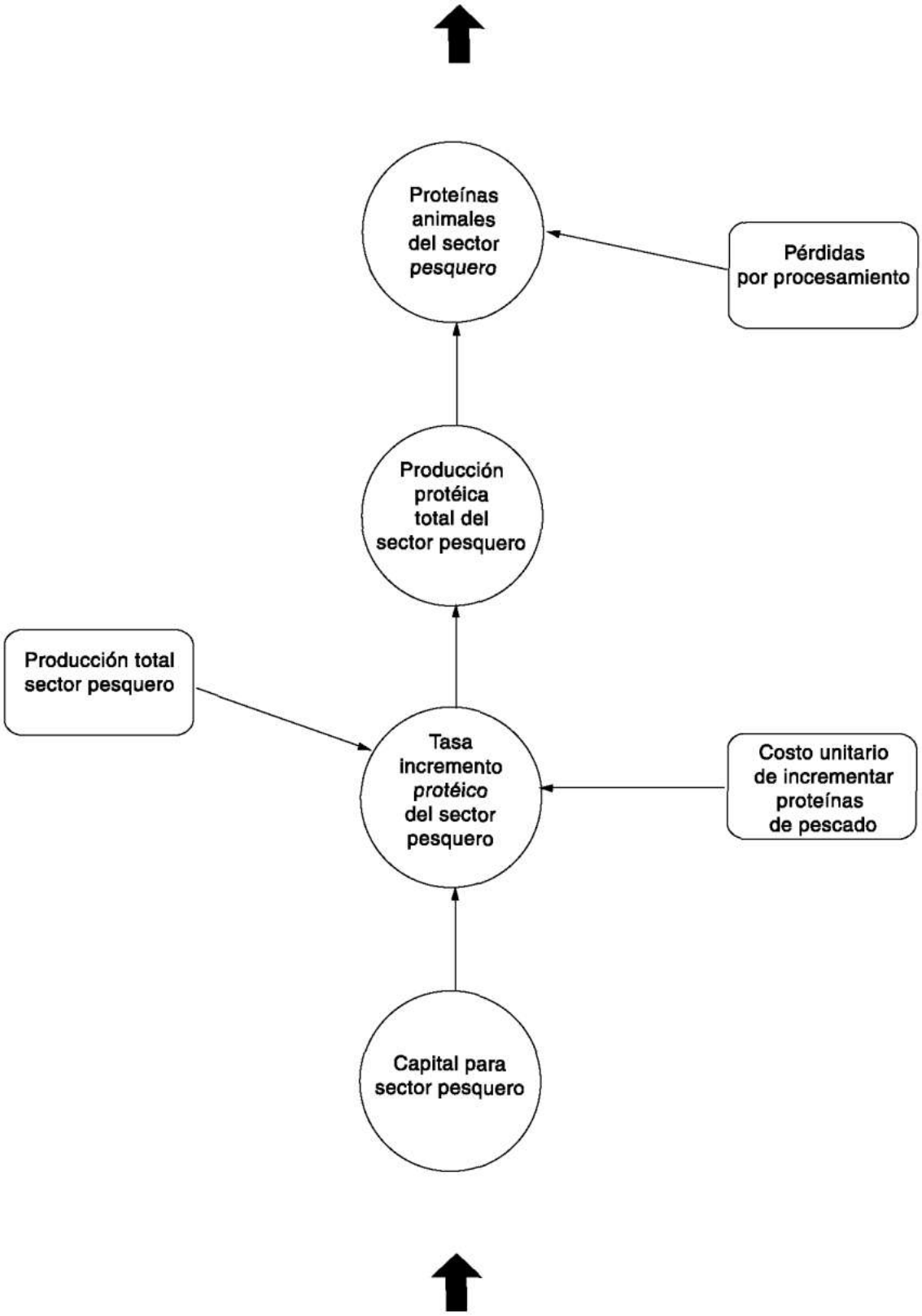


Figura B.7: Flujo de datos: Subsector Pesquería. Figura 15 en el libro original.

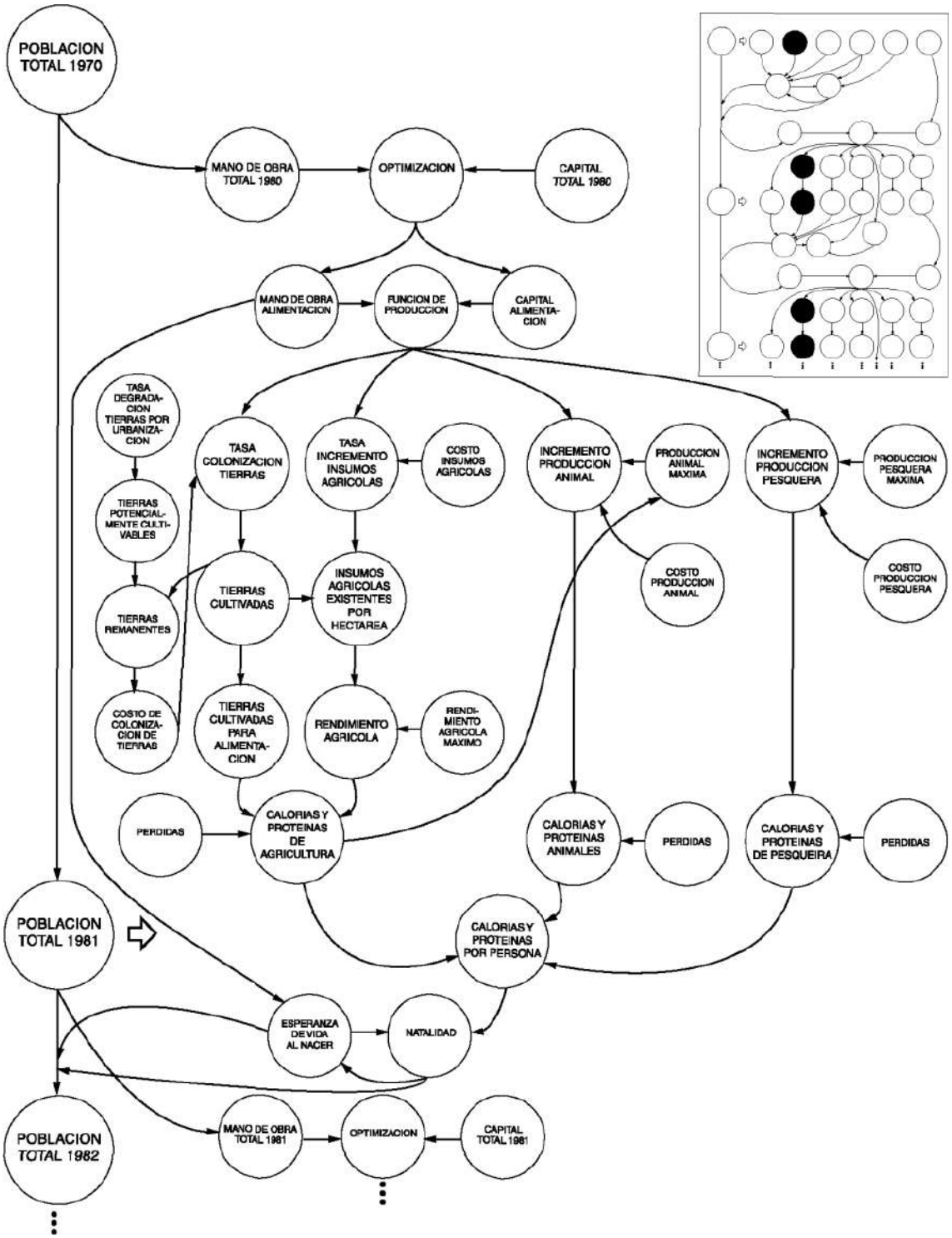


Figura B.8: Estructura y operación del sector alimentación durante la optimización. Figura 16 en el libro original.

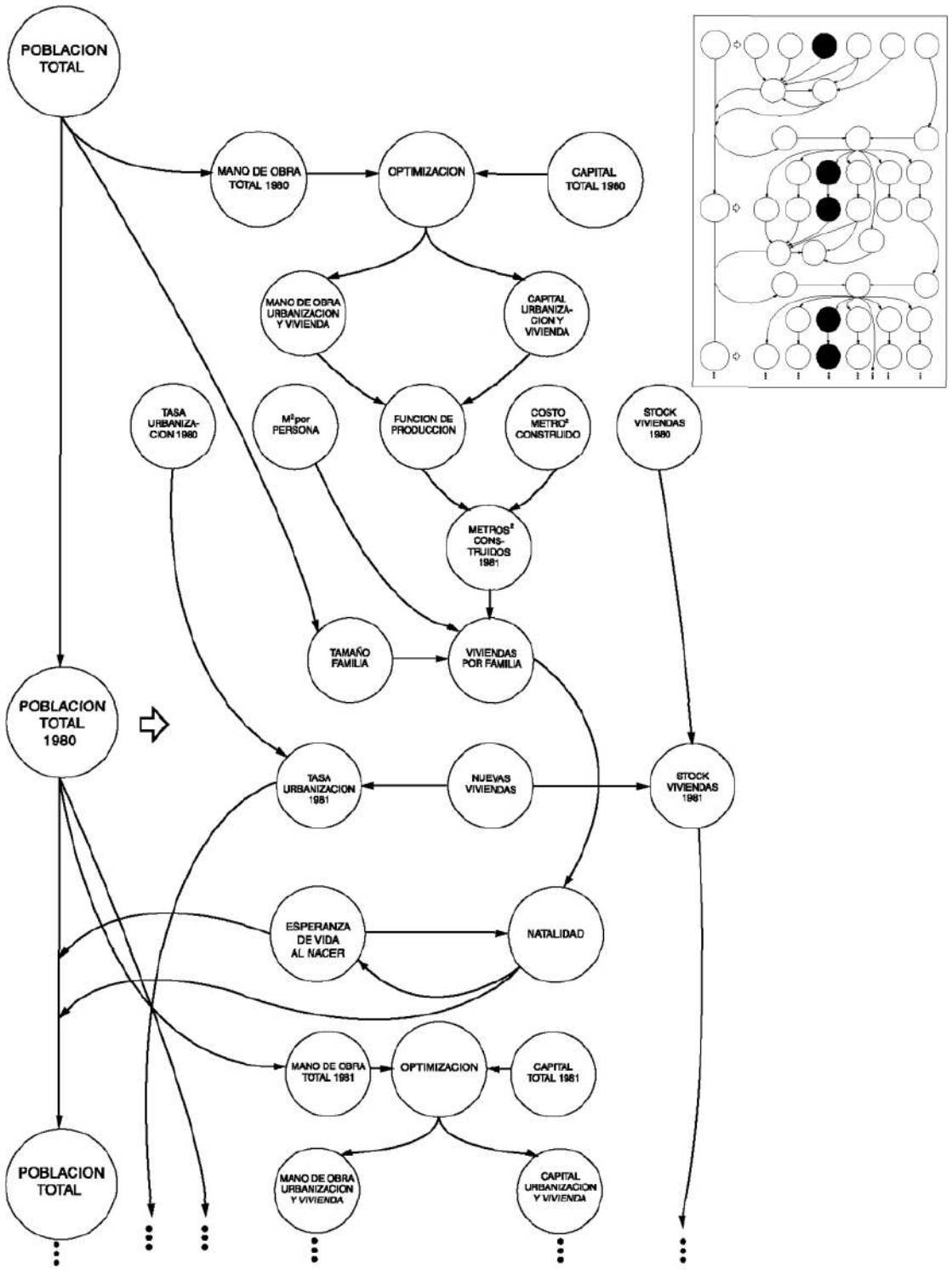


Figura B.9: Operación del sector vivienda y urbanización durante la optimización. Figura 17 en el libro original.

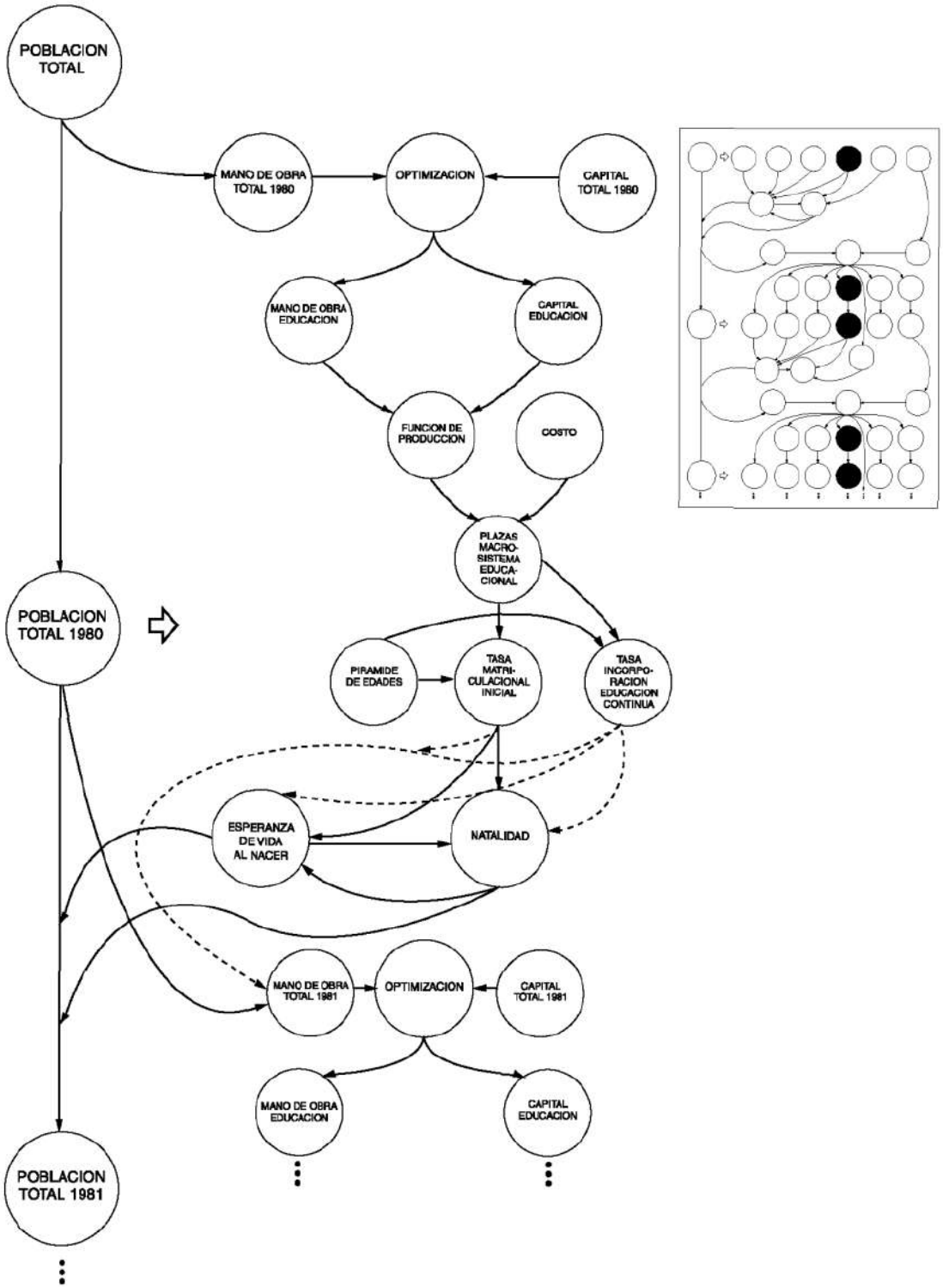


Figura B.10: Operación del sector educación durante la optimización. Figura 17 en el libro original.

B.2. Flujo de control

En la Sección 3.3.3 se presentaron dos diagramas de control obtenidos en parte por la ingeniería inversa del código recuperado y también en parte basado en el diagrama presentado en el Handbook [SFL+77], que facilitamos para el usuario en esta sección.

En esencia, los diagramas de este documento y los del Handbook son similares y no presentan información contradictoria. Se diferencian en el nivel detalle, siendo el diagrama del Handbook de más alto nivel, mientras que el nuevo es más detallista; en pequeñas diferencias en la inicialización, la nueva versión ahora puede leer la configuración de archivo; y en que el del Handbook sólo describe el flujo principal, mientras el nuevo también el de la función LELAPA.

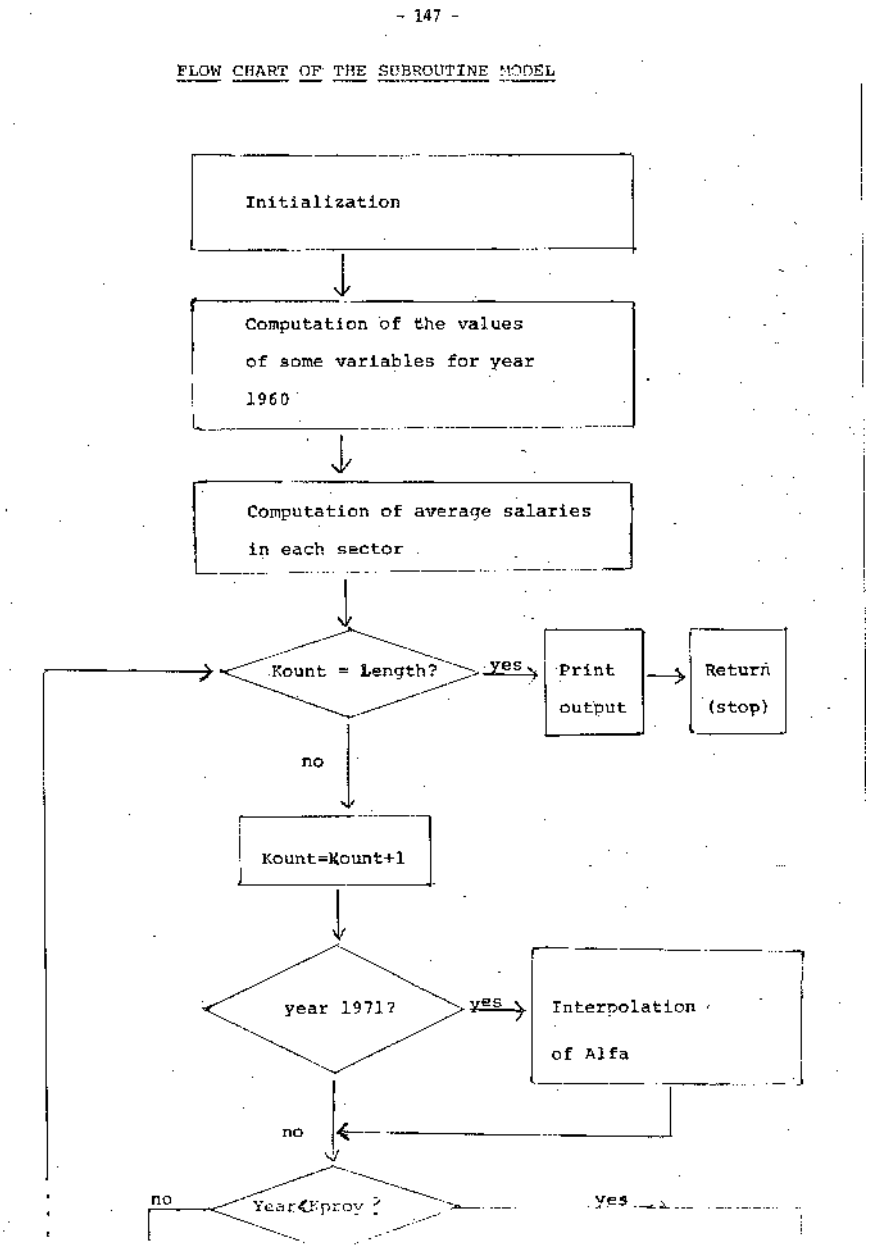


Figura B.11: Diagrama de Control en el Handbook: Parte 1

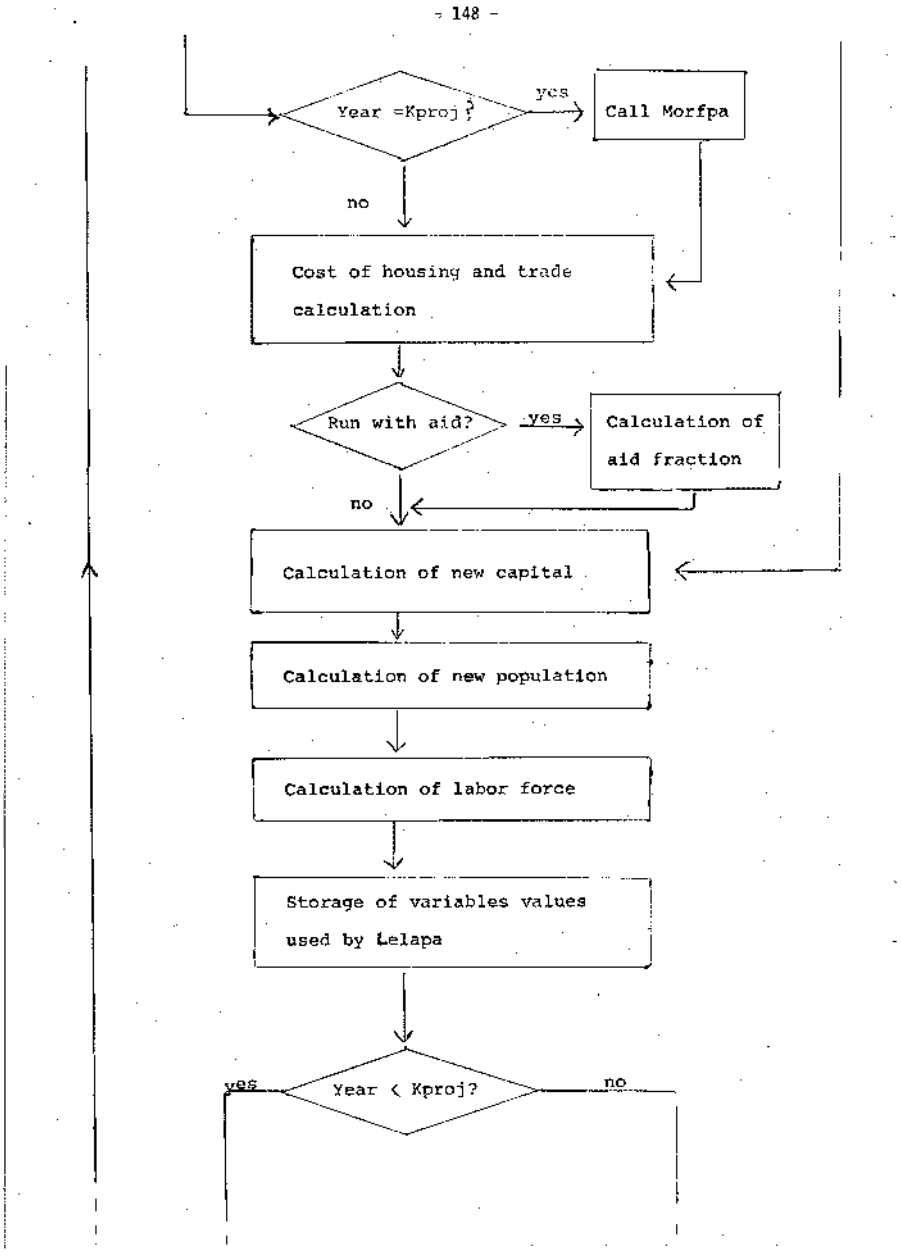


Figura B.12: Diagrama de Control en el Handbook: Parte 2

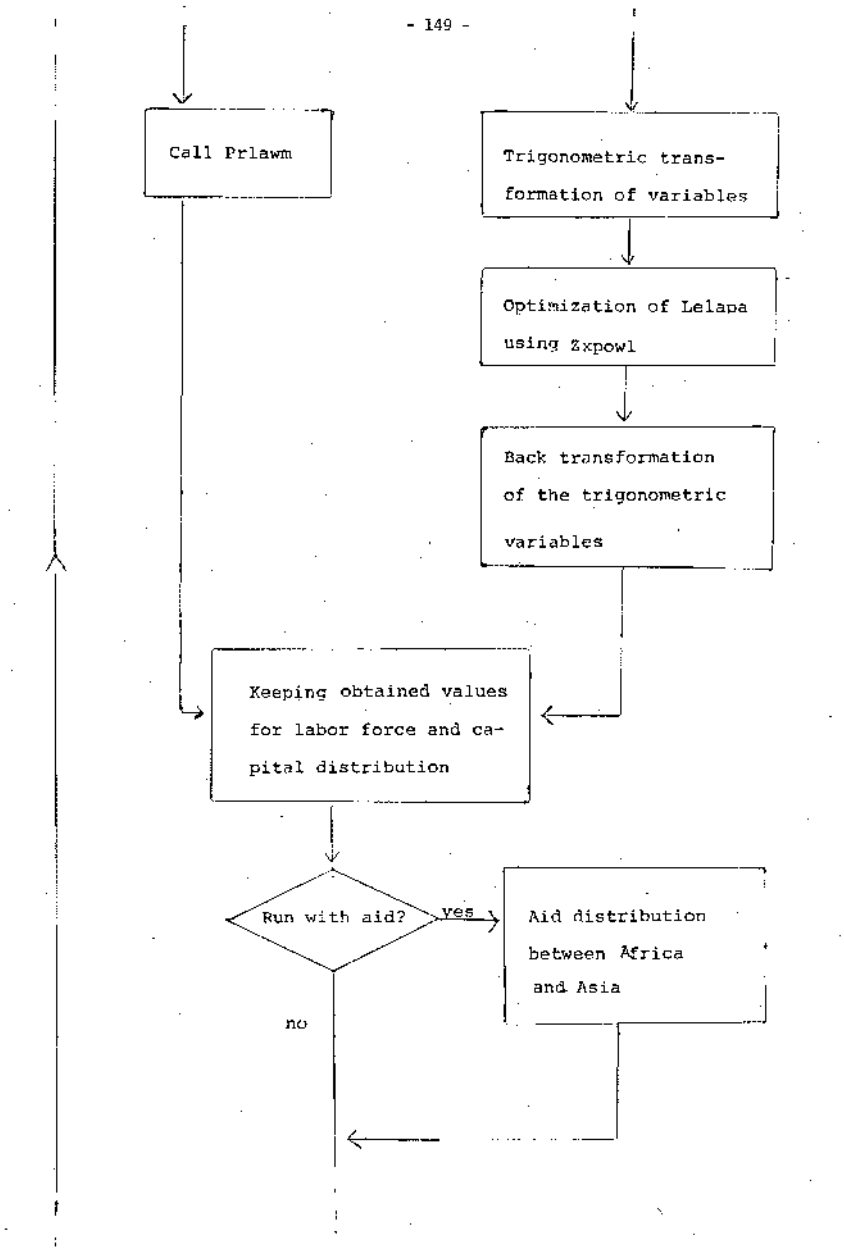


Figura B.13: Diagrama de Control en el Handbook: Parte 3

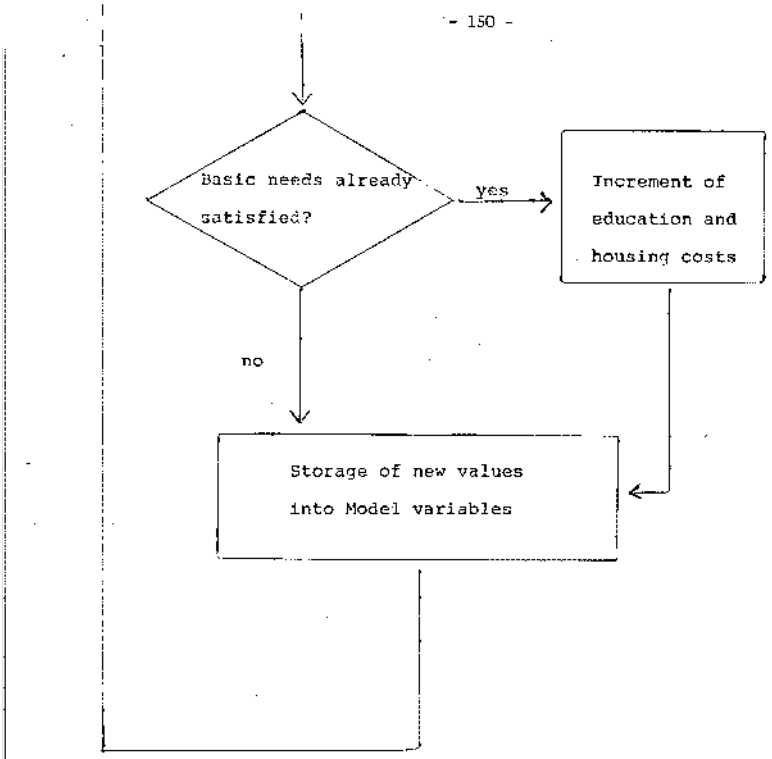


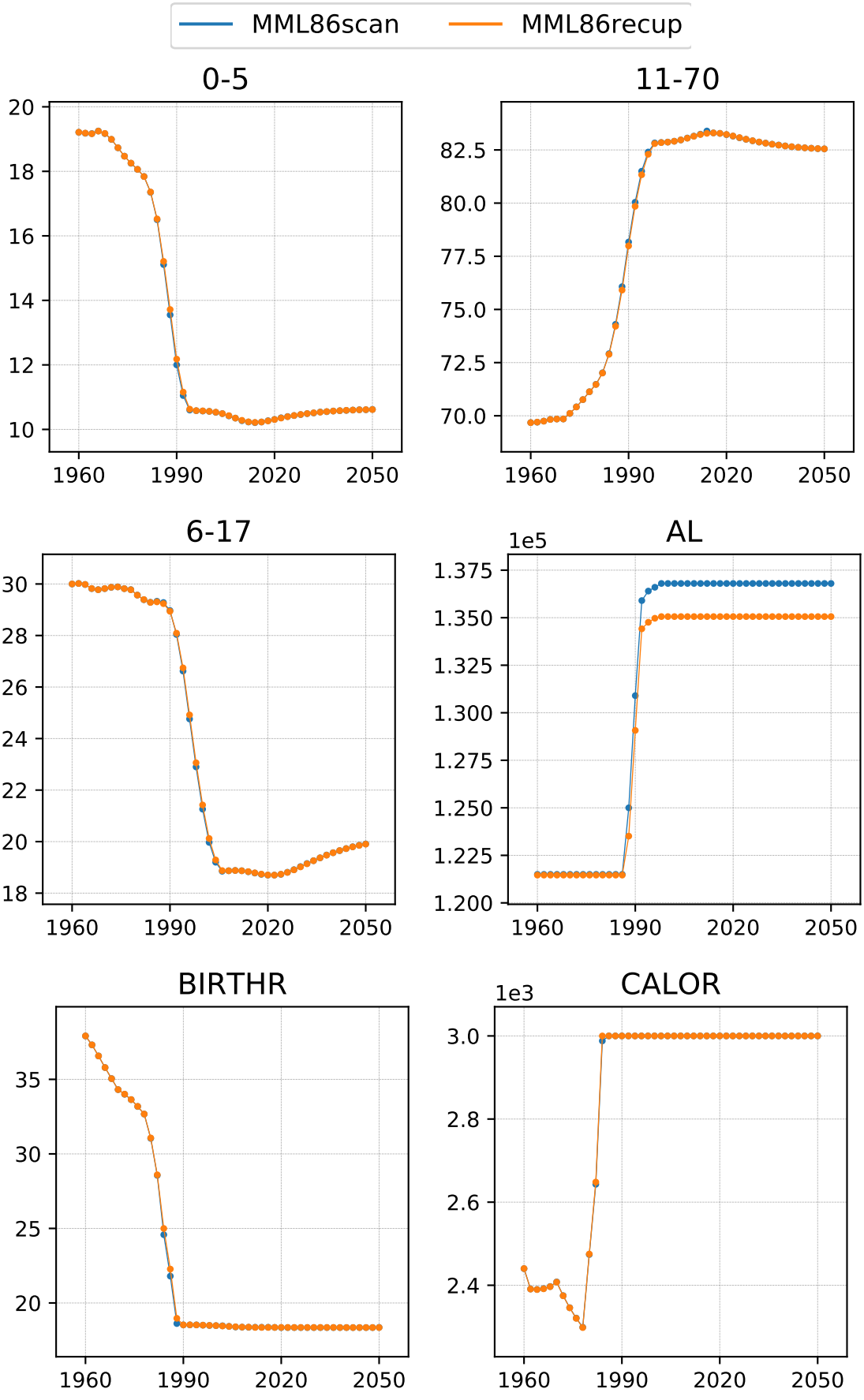
Figura B.14: Diagrama de Control en el Handbook: Parte 4

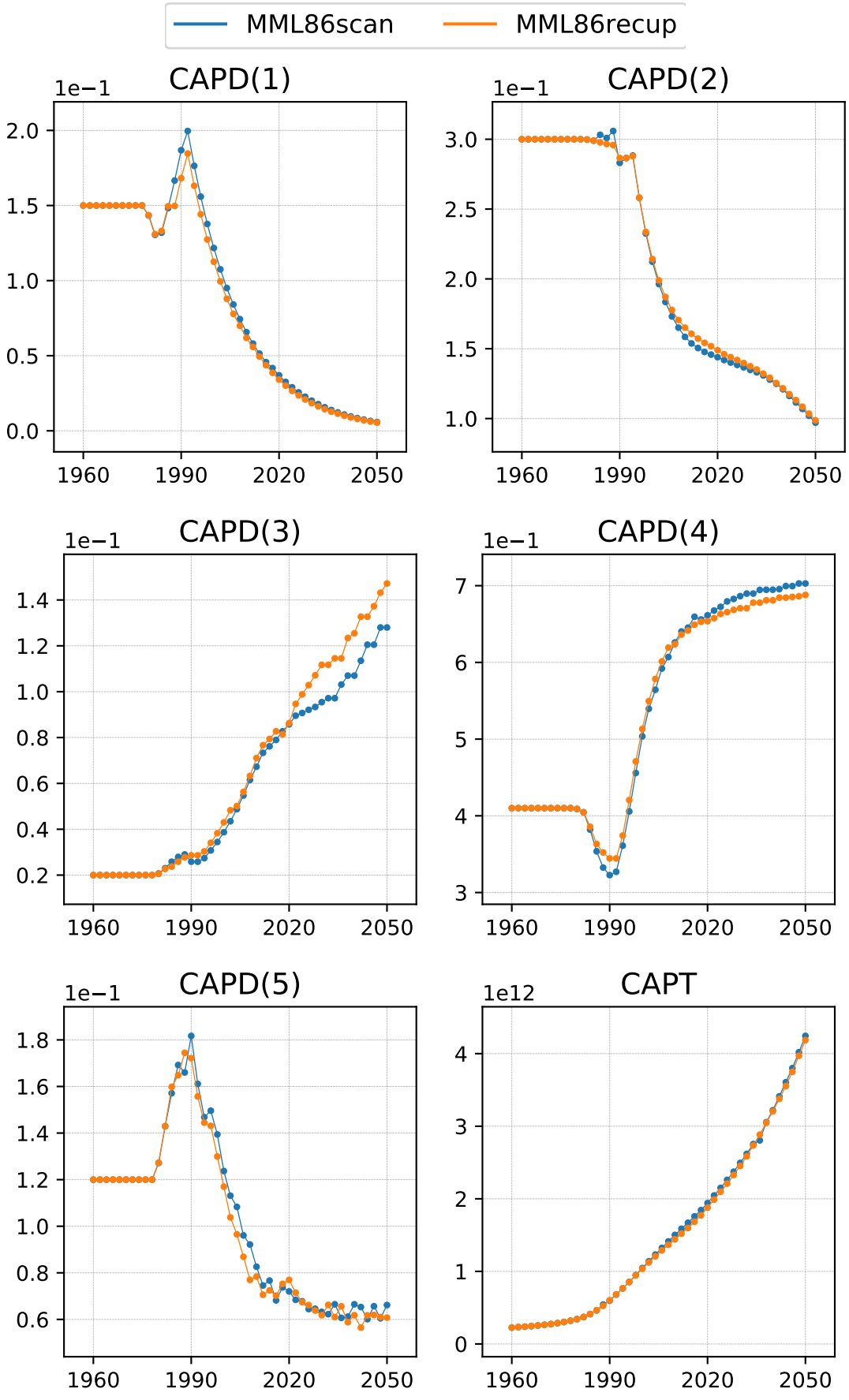
C. COMPARACIÓN GRÁFICA ENTRE VERSIONES

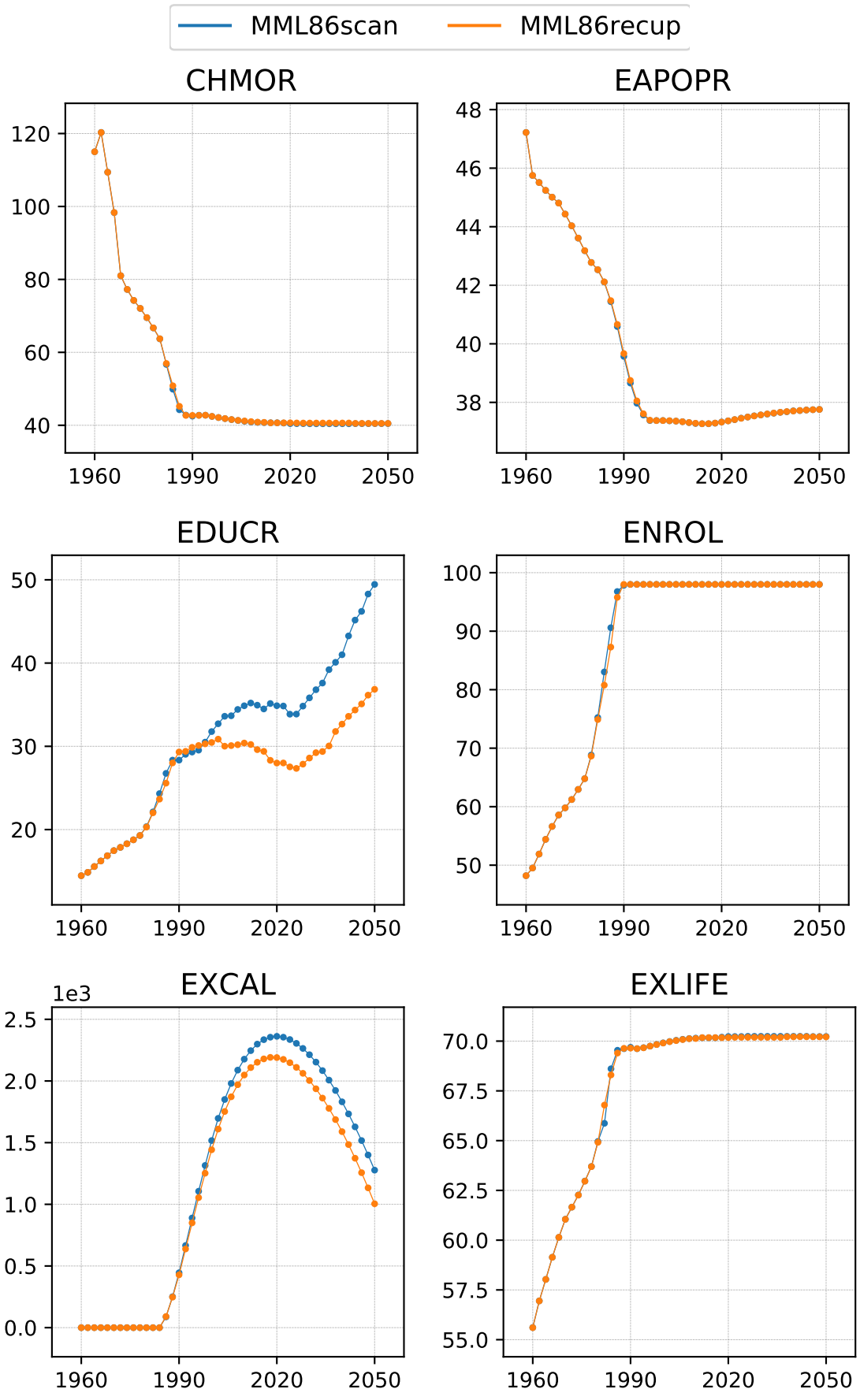
C.1. Comparación entre MML86scan y MML86recup

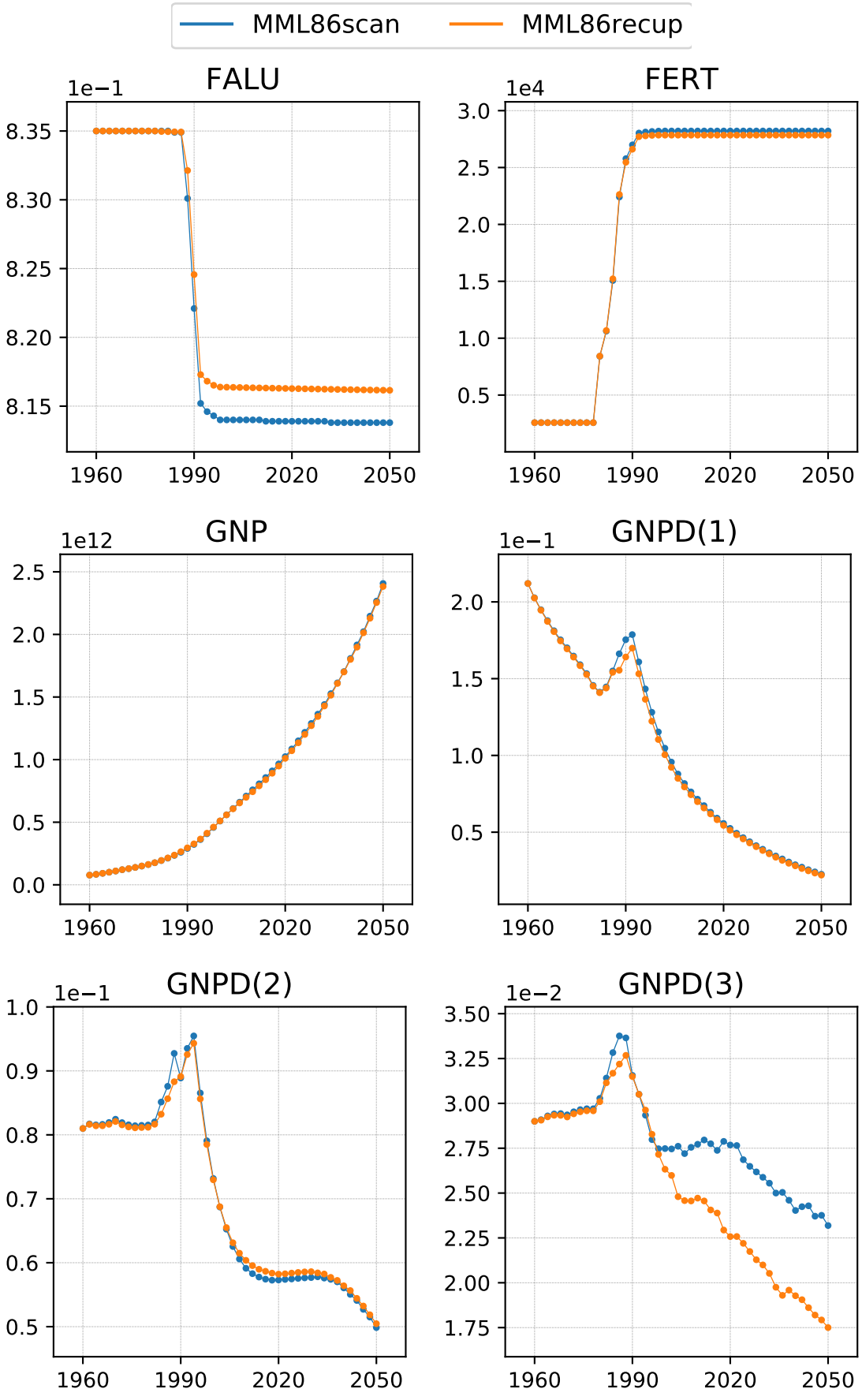
En esta Sección incluimos un gráfico comparando los resultados de la simulación del modelo corrido por última vez hace 35 años (`MML86scan`) y los resultados de la versión contemporánea en Fortran `MML86recup`. Ambas simulaciones utilizan los valores estándar de los parámetros, excepto por `KSTOP` que fue cambiado para extender la corrida hasta 2050. Se incluyen gráficos para cada variable de salida original pero sólo para la región de Latinoamérica, dado que fue la única reportada en `MML86scan`.

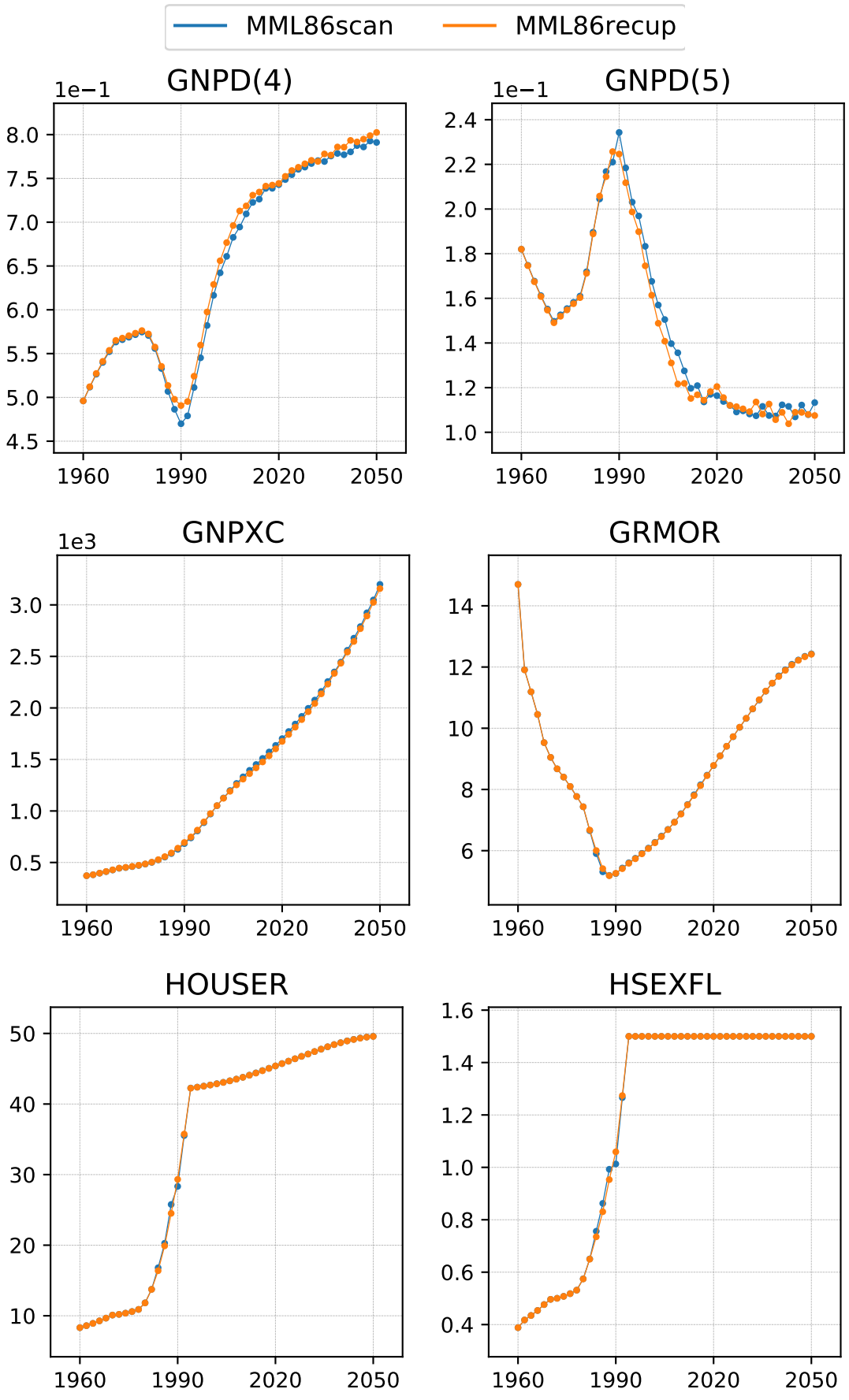
En estos gráficos podemos ver en detalle lo dicho en la Sección 2.4, donde se analizó el mapa de calor de estas mismas simulaciones, ratificando que los resultados de ambas simulaciones son prácticamente indistinguibles en todas las variables para el intervalo de tiempo $1960 \sim 1978$ y que para los años siguientes la diferencia crece constantemente. En la segunda mitad de la simulación, el caso más serio es el de `RLFD(3)` que es la única variable que no sólo revela diferencias en cada valor, si no que también cambia completamente el comportamiento entre una simulación y otra, oscilando alrededor del valor 1,85 en la Simulación de 1986 pero cayendo constantemente en la Simulación de 2020. Esto nos da la pauta de que la diferencia en las otras variables puede tener sus raíces en algún error en el cálculo de `RLFD(3)` y es un buen punto de partida para un análisis futuro.

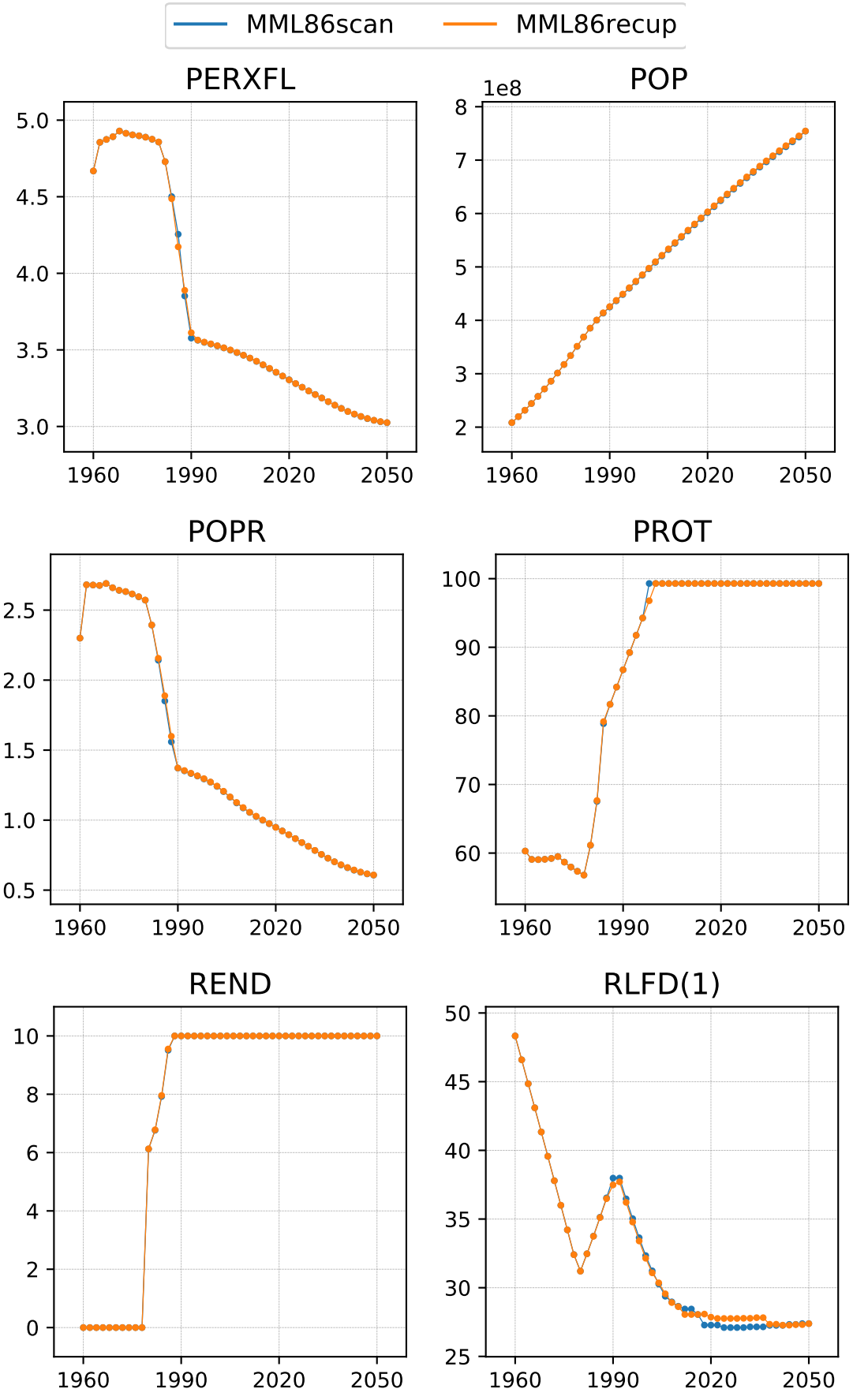

MML86scan
MML86recup
0-5
11-70
6-17
AL
1e5
BIRTHR
CALOR
1e3
1960
1990
2020
2050

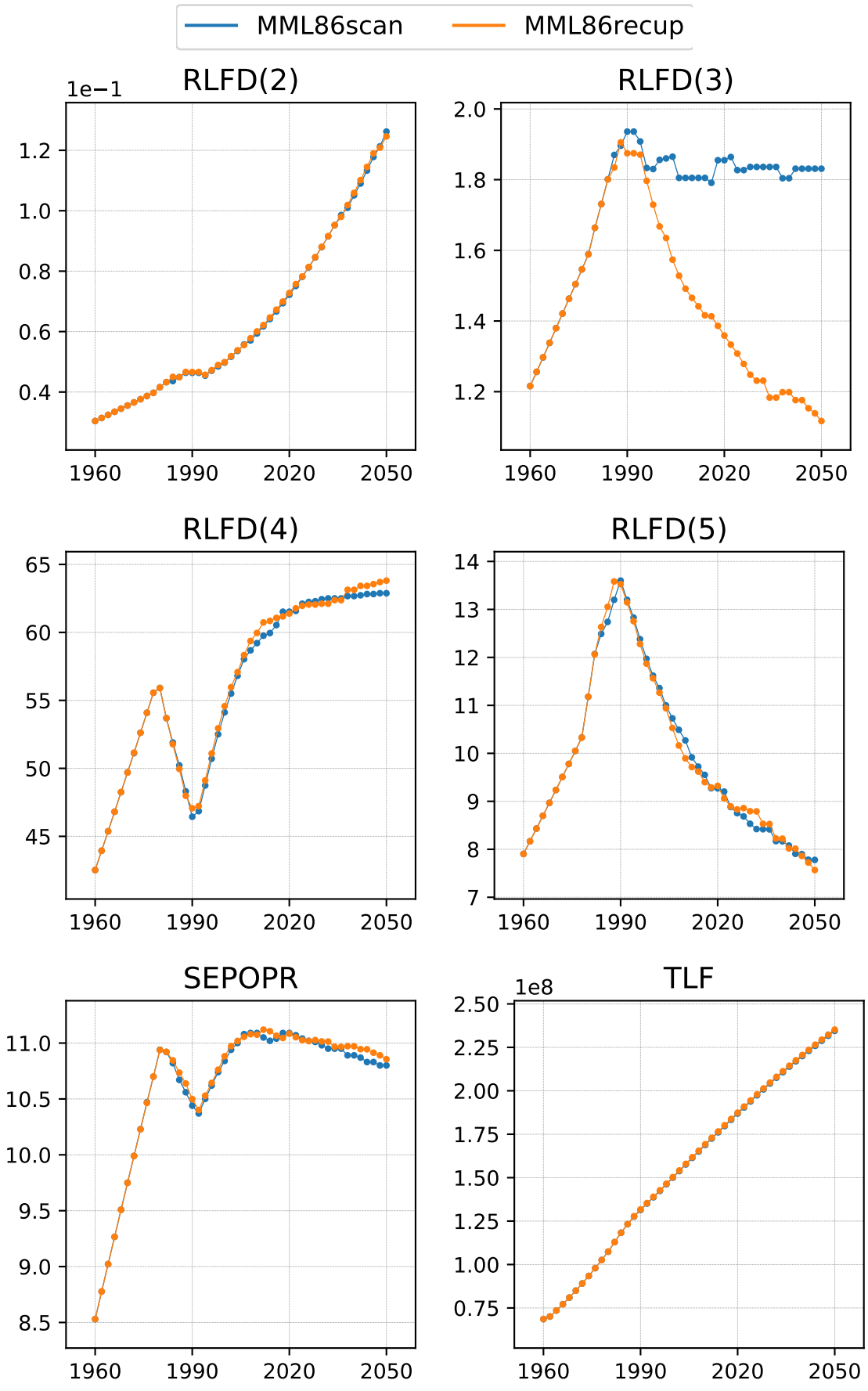

MML86scan
MML86recup
CAPD(1)
1e−1
2.0
1.5
1.0
0.5
0.0
1960
1990
2020
2050
CAPD(2)
1e−1
3.0
2.5
2.0
1.5
1.0
CAPD(3)
1e−1
1.4
1.2
1.0
0.8
0.6
0.4
0.2
CAPD(4)
1e−1
7
6
5
4
3
CAPD(5)
1e−1
1.8
1.6
1.4
1.2
1.0
0.8
0.6
CAPT
1e12
4
3
2
1
0

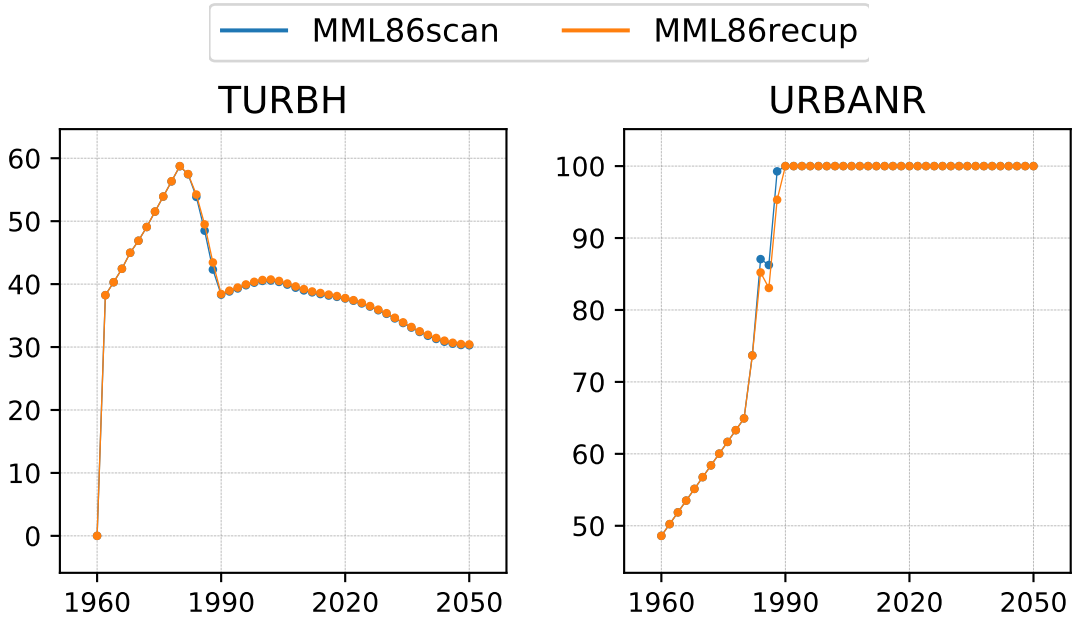

MML86scan
MML86recup
CHMOR
EAPOPR
EDUCR
ENROL
EXCAL
EXLIFE
1e3
1960
1990
2020
2050


MML86scan
MML86recup
FALU
1e−1
8.35
8.30
8.25
8.20
8.15
1960
1990
2020
2050
FERT
1e4
3.0
2.5
2.0
1.5
1.0
0.5
GNP
1e12
2.5
2.0
1.5
1.0
0.5
0.0
GNPD(1)
1e−1
2.0
1.5
1.0
0.5
GNPD(2)
1e−1
1.0
0.9
0.8
0.7
0.6
0.5
GNPD(3)
1e−2
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75


MML86scan
MML86recup
GNPD(4)
1e−1
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
1960
1990
2020
2050
GNPD(5)
1e−1
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
GNPXC
1e3
3.0
2.5
2.0
1.5
1.0
0.5
GRMOR
14
12
10
8
6
HOUSER
50
40
30
20
10
HSEXFL
1.6
1.4
1.2
1.0
0.8
0.6
0.4


MML86scan
MML86recup
PERXFL
5.0
4.5
4.0
3.5
3.0
1960
1990
2020
2050
POP
1e8
8
7
6
5
4
3
2
1960
1990
2020
2050
POPR
2.5
2.0
1.5
1.0
0.5
1960
1990
2020
2050
PROT
100
90
80
70
60
1960
1990
2020
2050
REND
10
8
6
4
2
0
1960
1990
2020
2050
RLFD(1)
50
45
40
35
30
25
1960
1990
2020
2050


MML86scan
MML86recup
RLFD(2)
1e−1
1.2
1.0
0.8
0.6
0.4
1960
1990
2020
2050
RLFD(3)
2.0
1.8
1.6
1.4
1.2
RLFD(4)
65
60
55
50
45
RLFD(5)
14
13
12
11
10
9
8
7
SEPOPR
11.0
10.5
10.0
9.5
9.0
8.5
TLF
1e8
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75

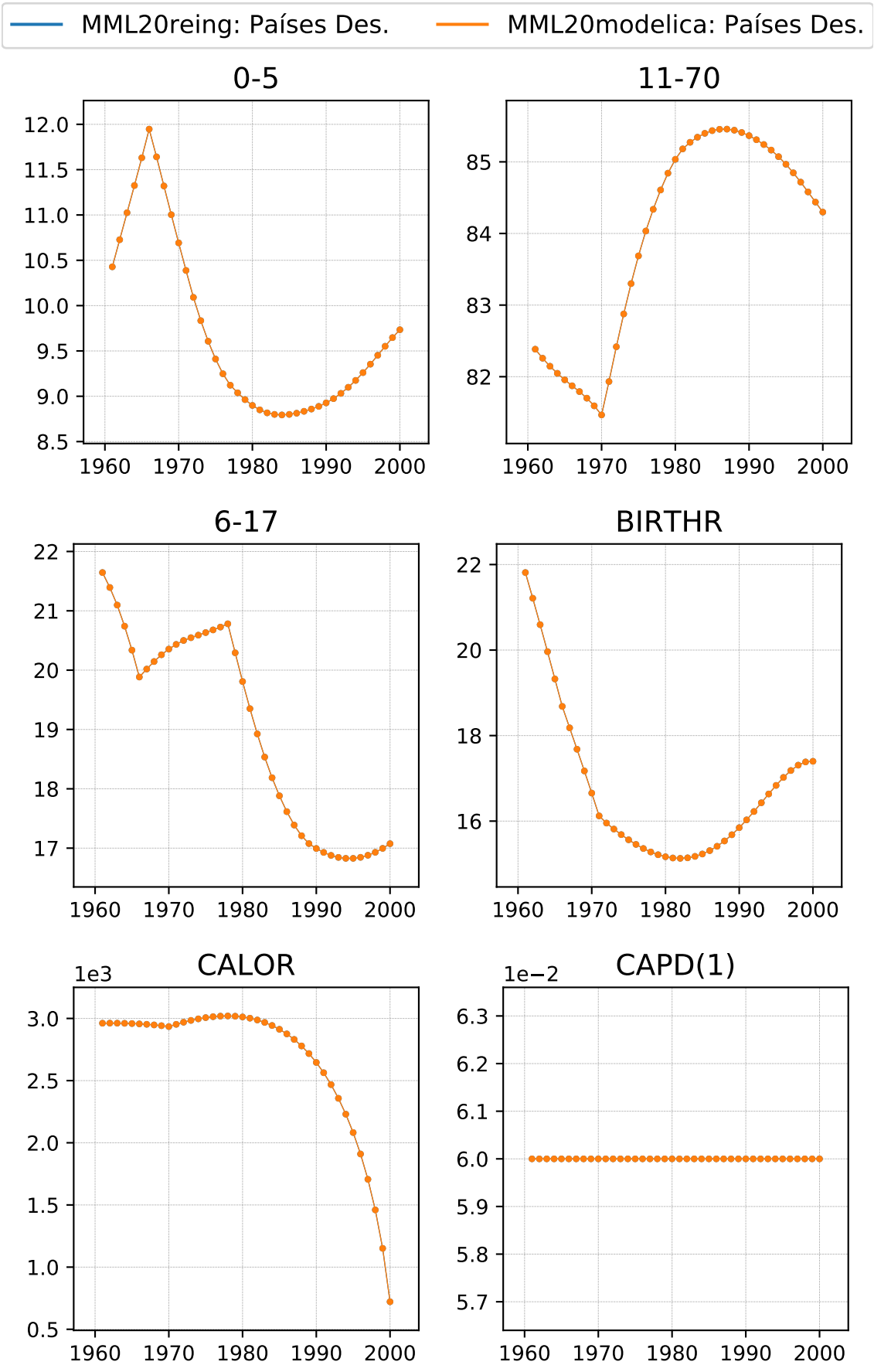


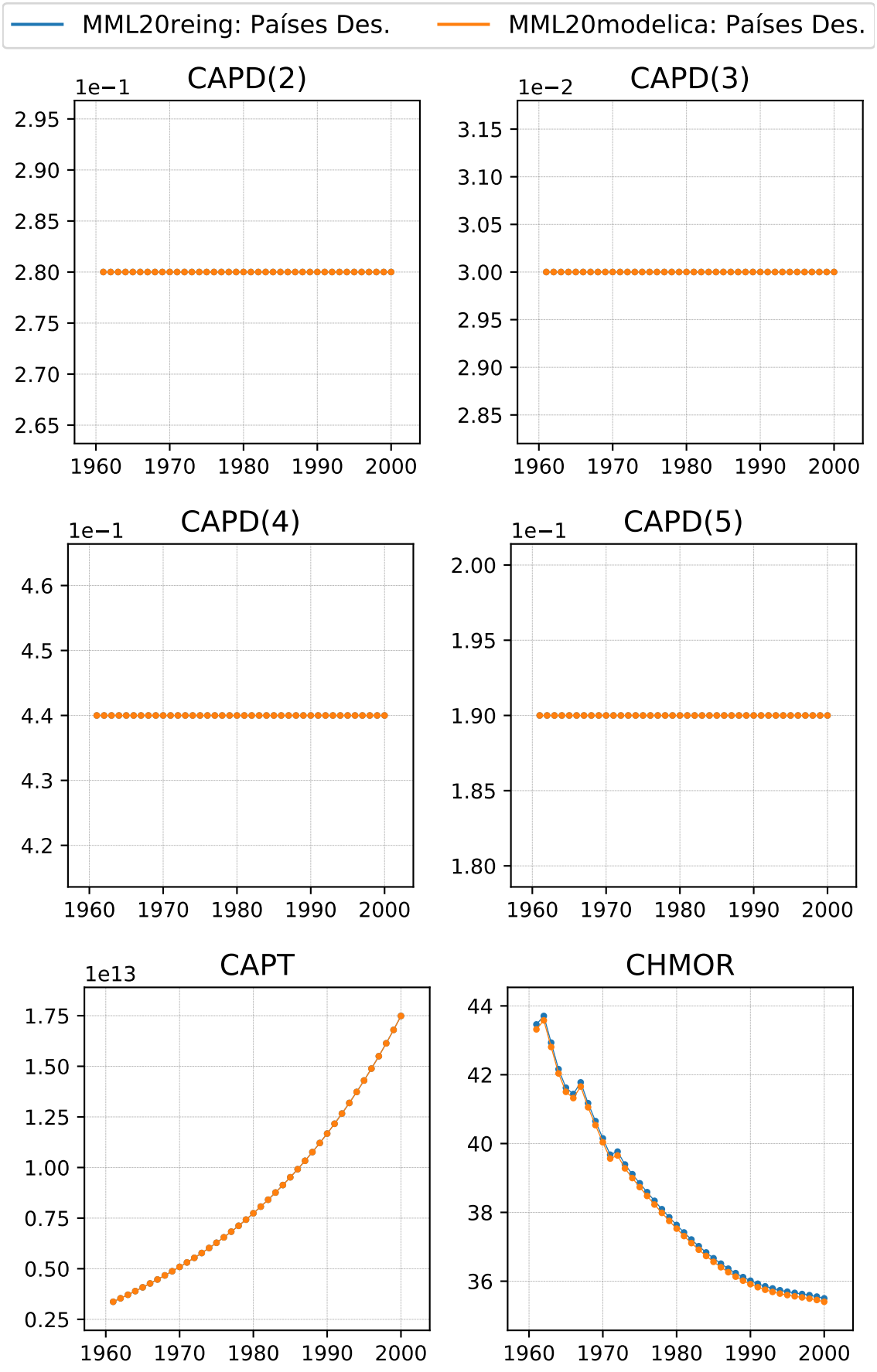
C.2. Comparación entre MML20reing y MML20modelica

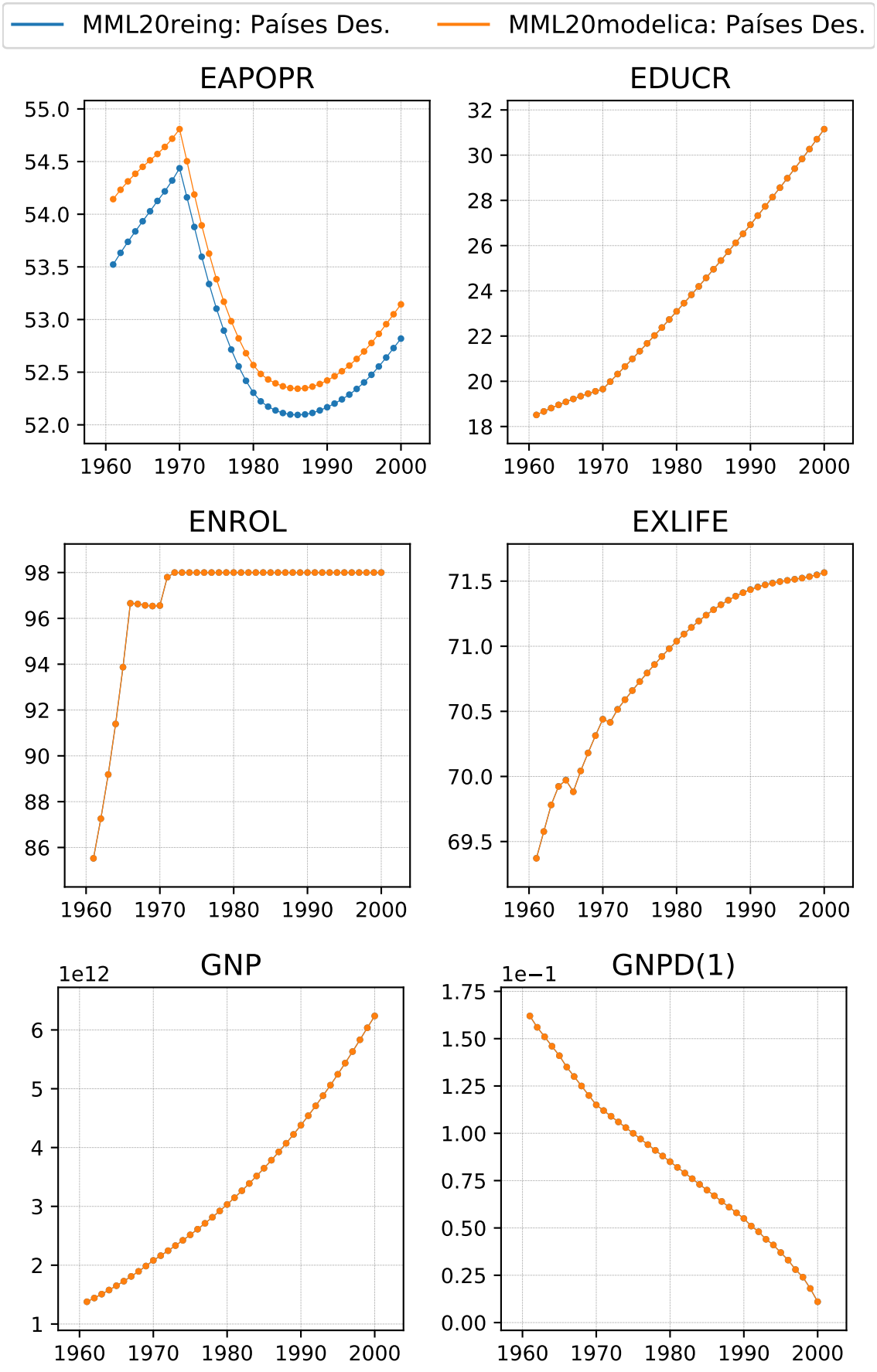
En esta Sección presentamos los gráficos comparando el modelo en Fortran recuperado (Fortran 2020) y la nueva versión del modelo en Modelica, que no incluye la Fase de Optimización. Por esto la simulación del modelo de Fortran fue corrida con `KPROJ` en 2000 (para que nunca transicione de la Fase de Proyección) y además no se grafican las variables `AL, FALU, FERT, REND` y `EXCAL` (que no son utilizadas en Proyección). El resto de los parámetros mantienen su valor estándar, por lo que se simulan las cuatro regiones originales y la simulación abarca el intervalo de años 1960 $\sim$ 2000. Sin embargo, por limitaciones técnicas de Modelica al utilizar saltos discretos, los valores de 1960 no son precisos, por lo que decidimos no incluirlos en los gráficos para disminuir el ruido visual.

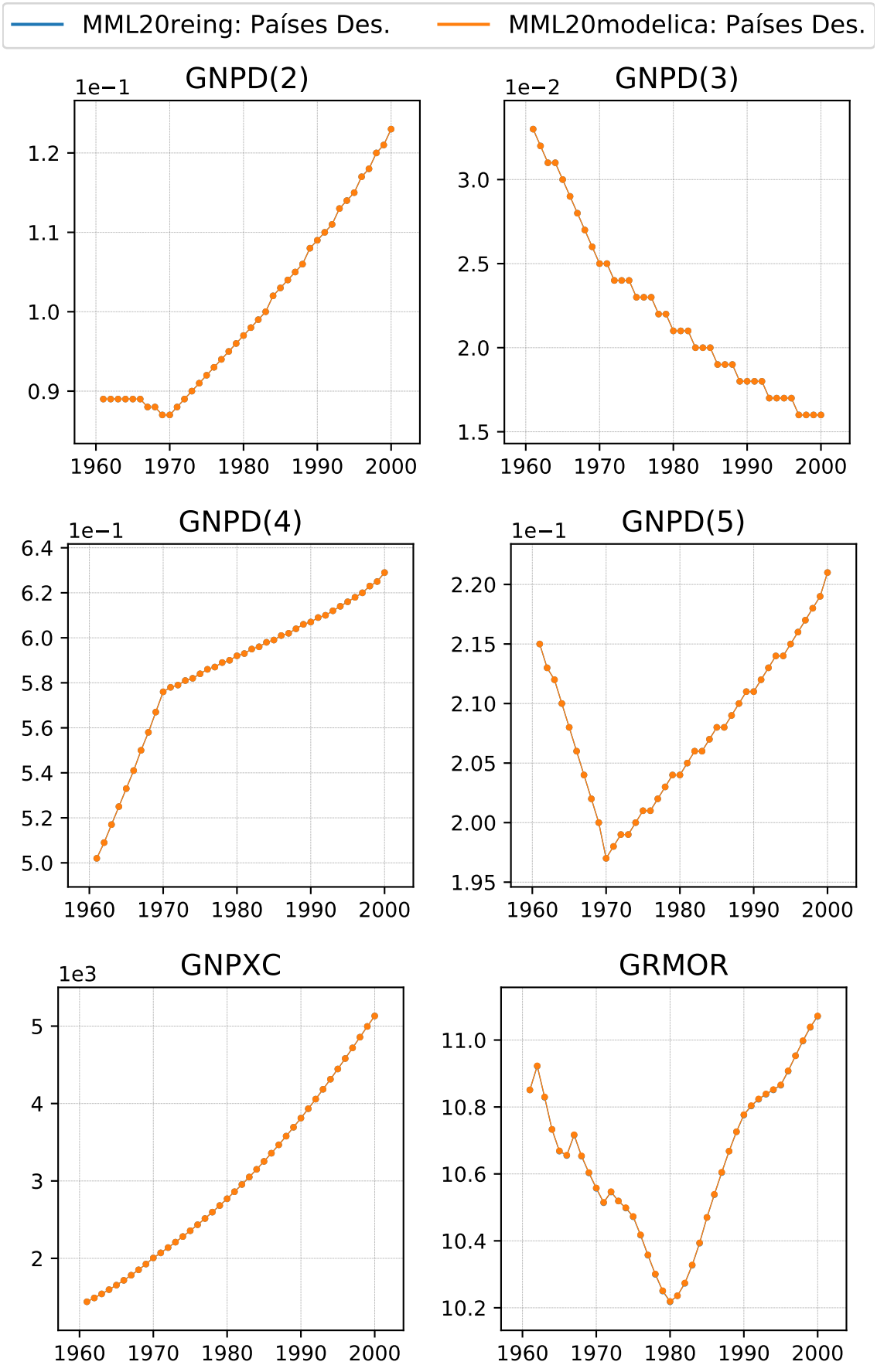
Al analizar los gráficos podemos confirmar lo mencionado en la Sección 5.3, en donde se mencionaba a las variables `EAPOPR, POPR` y `TLF` como problemáticas en todas las regiones, siendo África la región más afectada por estas y otras variables como `GNPD(1), GNPD(1),GNPD(1),GNPD(1)` y `GNPD(5)`, las cuales presentan diferencias distinguibles. Sin embargo, en todas las variables de todas las regiones podemos ver cómo la *forma* que describen los puntos es la misma entre ambas versiones y las diferencias pueden ser resultado de diferencias numéricas entre lenguajes o por algún error de transcripción de parámetros o datos.

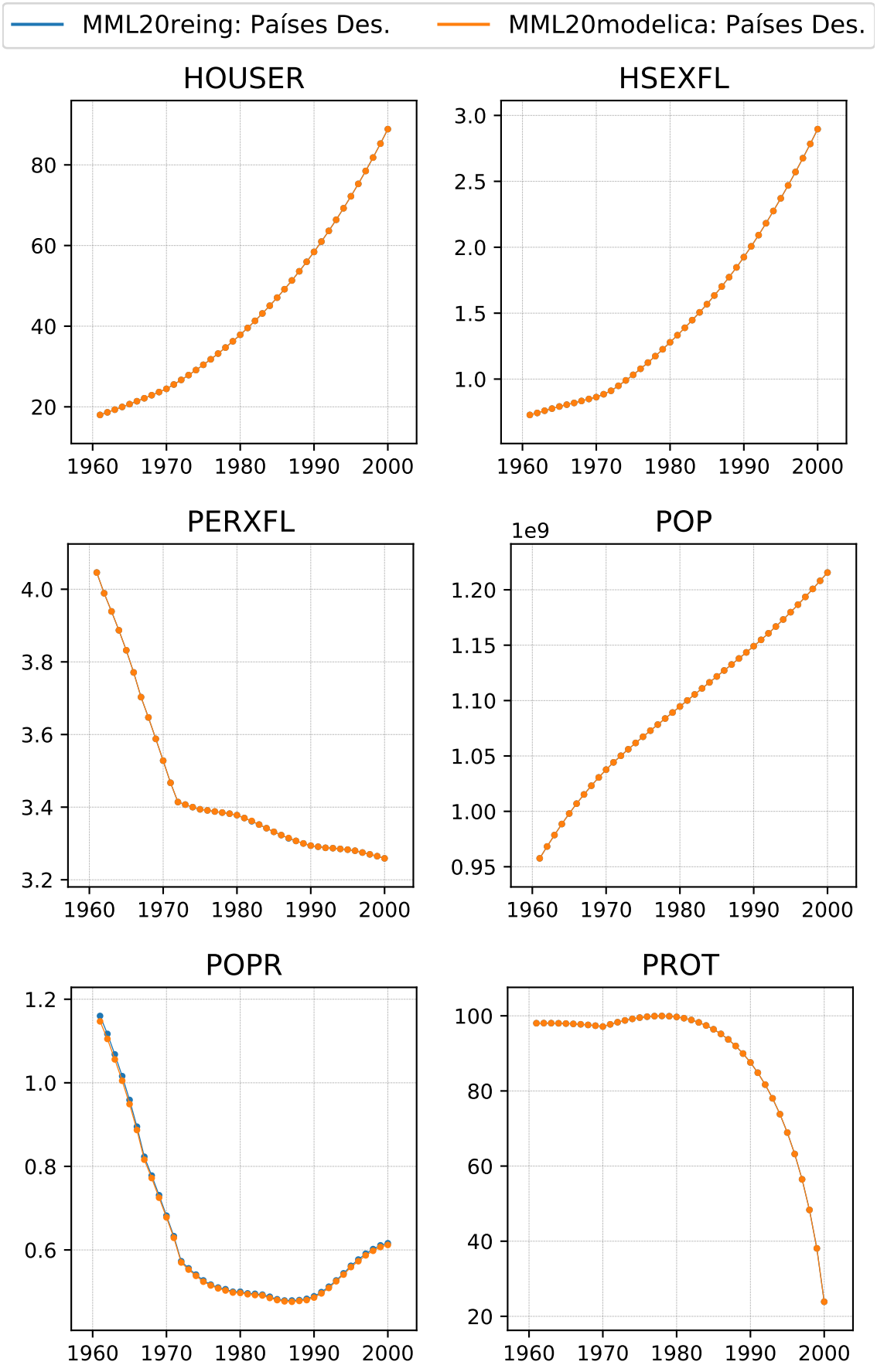
C.2.1. Países Desarrollados

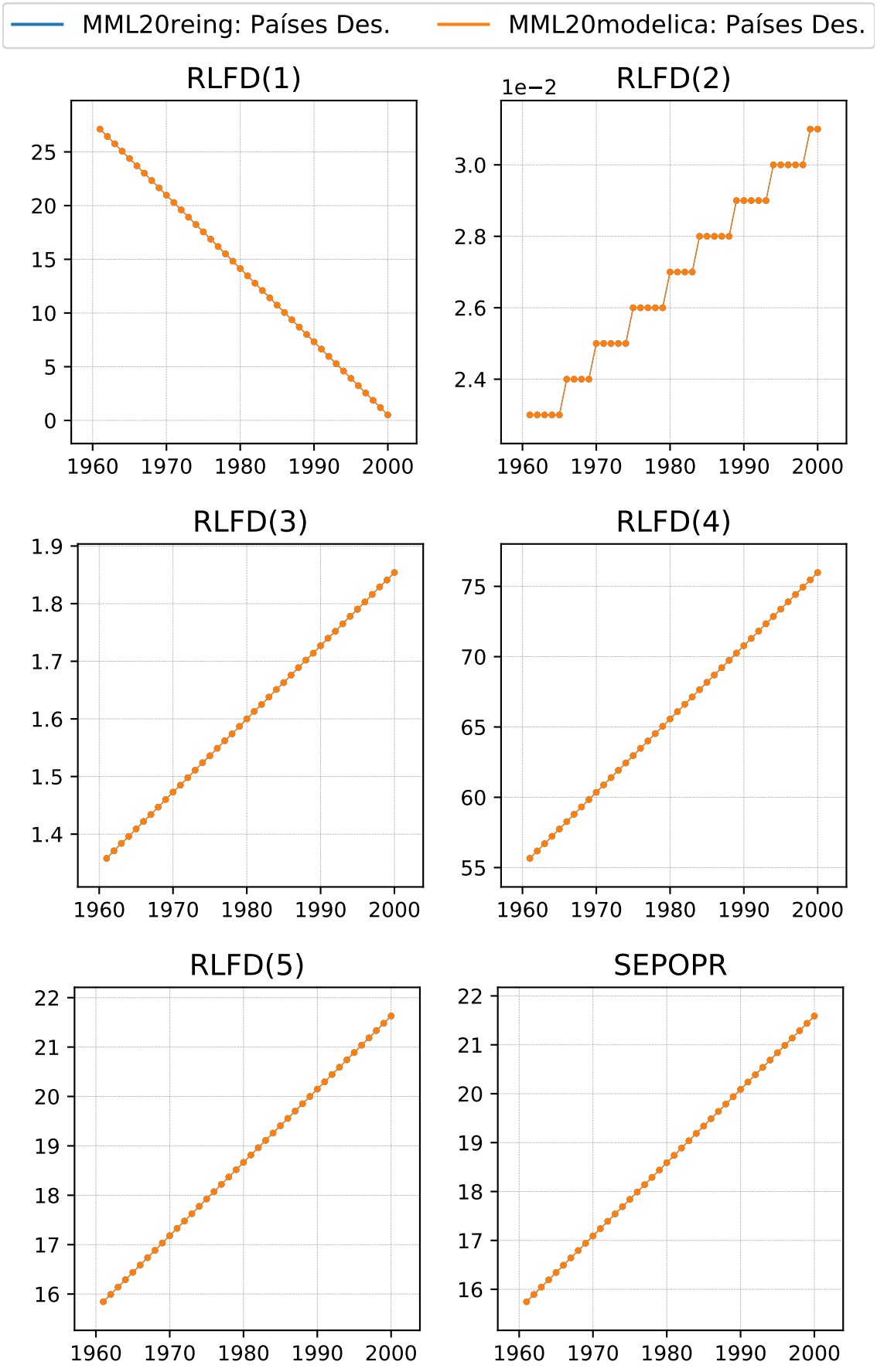

MML20reing: Países Des.
MML20modelica: Países Des.
0-5
11-70
6-17
BIRTHR
CALOR
1e3
CAPD(1)
1e−2
1960
1970
1980
1990
2000

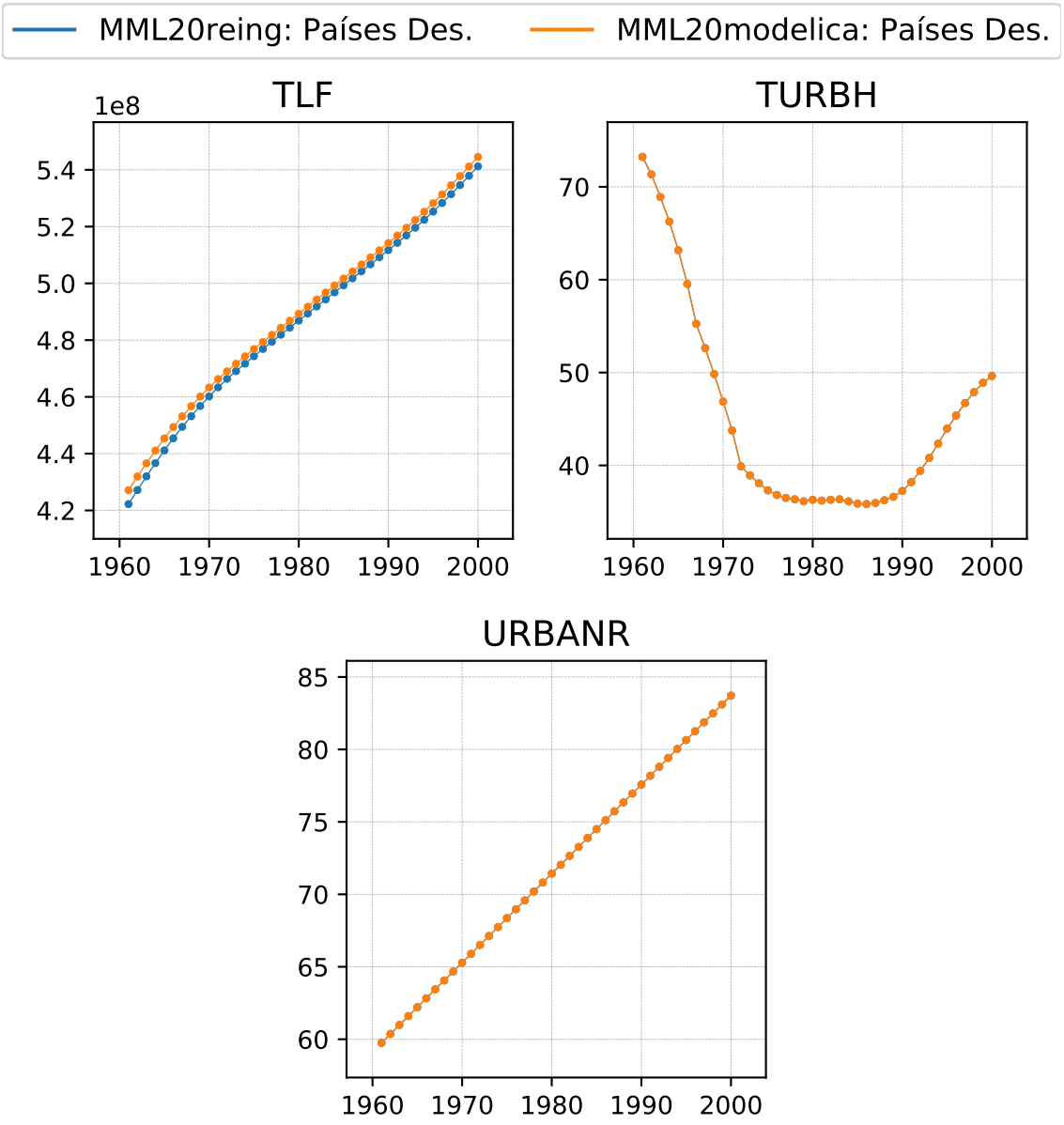

MML20reing: Países Des.
MML20modelica: Países Des.
CAPD(2)
1e−1
2.95
2.90
2.85
2.80
2.75
2.70
2.65
1960 1970 1980 1990 2000
CAPD(3)
1e−2
3.15
3.10
3.05
3.00
2.95
2.90
2.85
1960 1970 1980 1990 2000
CAPD(4)
1e−1
4.6
4.5
4.4
4.3
4.2
1960 1970 1980 1990 2000
CAPD(5)
1e−1
2.00
1.95
1.90
1.85
1.80
1960 1970 1980 1990 2000
CAPT
1e13
1.75
1.50
1.25
1.00
0.75
0.50
0.25
1960 1970 1980 1990 2000
CHMOR
44
42
40
38
36
1960 1970 1980 1990 2000


MML20reing: Países Des.
MML20modelica: Países Des.
EAPOPR
55.0
54.5
54.0
53.5
53.0
52.5
52.0
1960 1970 1980 1990 2000
EDUCR
32
30
28
26
24
22
20
18
1960 1970 1980 1990 2000
ENROL
98
96
94
92
90
88
86
1960 1970 1980 1990 2000
EXLIFE
71.5
71.0
70.5
70.0
69.5
1960 1970 1980 1990 2000
GNP
1e12
6
5
4
3
2
1
1960 1970 1980 1990 2000
GNPD(1)
1e−1
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1960 1970 1980 1990 2000

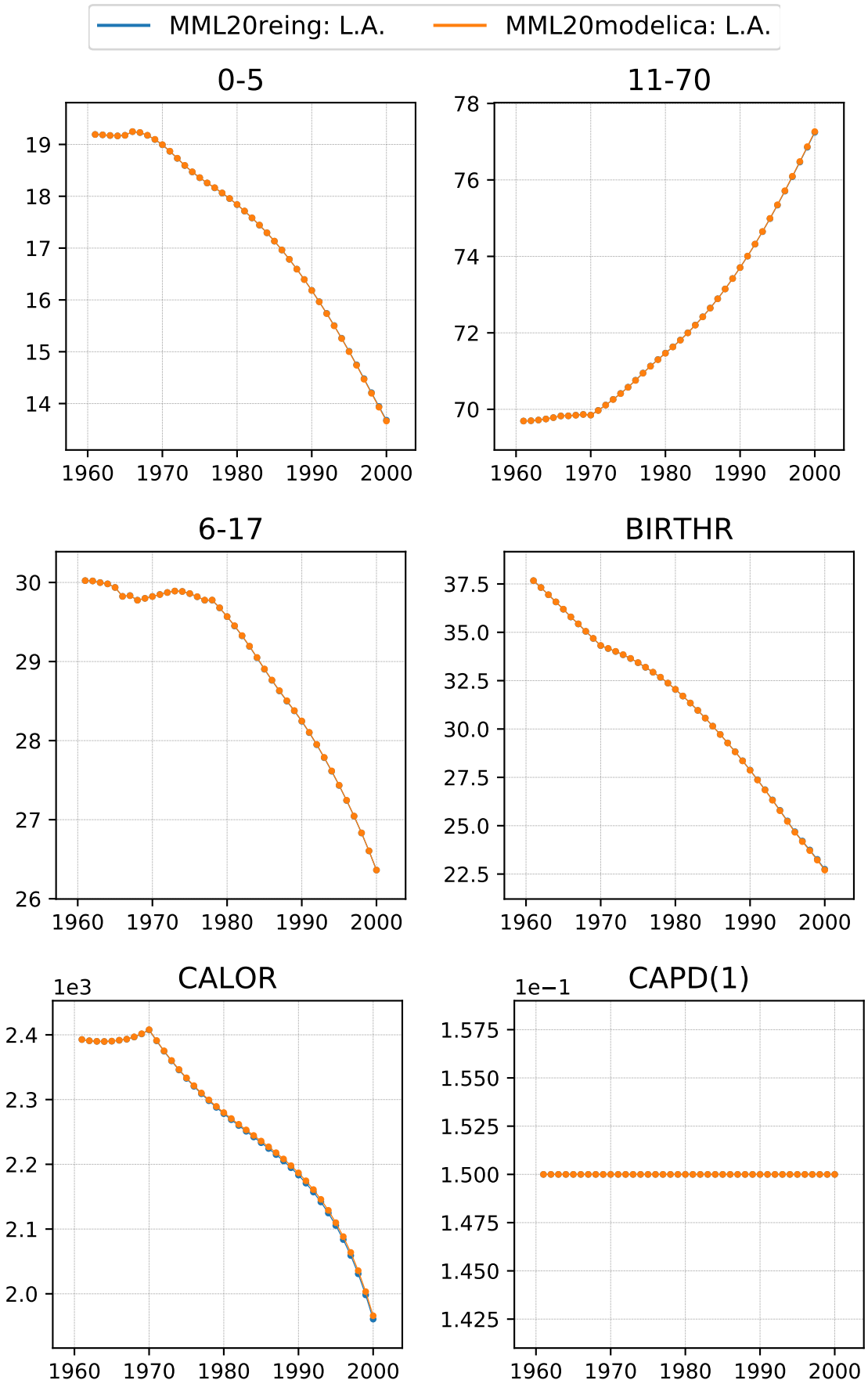

MML20reing: Países Des.
MML20modelica: Países Des.
GNPD(2)
1e−1
1.2
1.1
1.0
0.9
1960 1970 1980 1990 2000
GNPD(3)
1e−2
3.0
2.5
2.0
1.5
1960 1970 1980 1990 2000
GNPD(4)
1e−1
6.4
6.2
6.0
5.8
5.6
5.4
5.2
5.0
1960 1970 1980 1990 2000
GNPD(5)
1e−1
2.20
2.15
2.10
2.05
2.00
1.95
1960 1970 1980 1990 2000
GNPXC
1e3
5
4
3
2
1960 1970 1980 1990 2000
GRMOR
11.0
10.8
10.6
10.4
10.2
1960 1970 1980 1990 2000

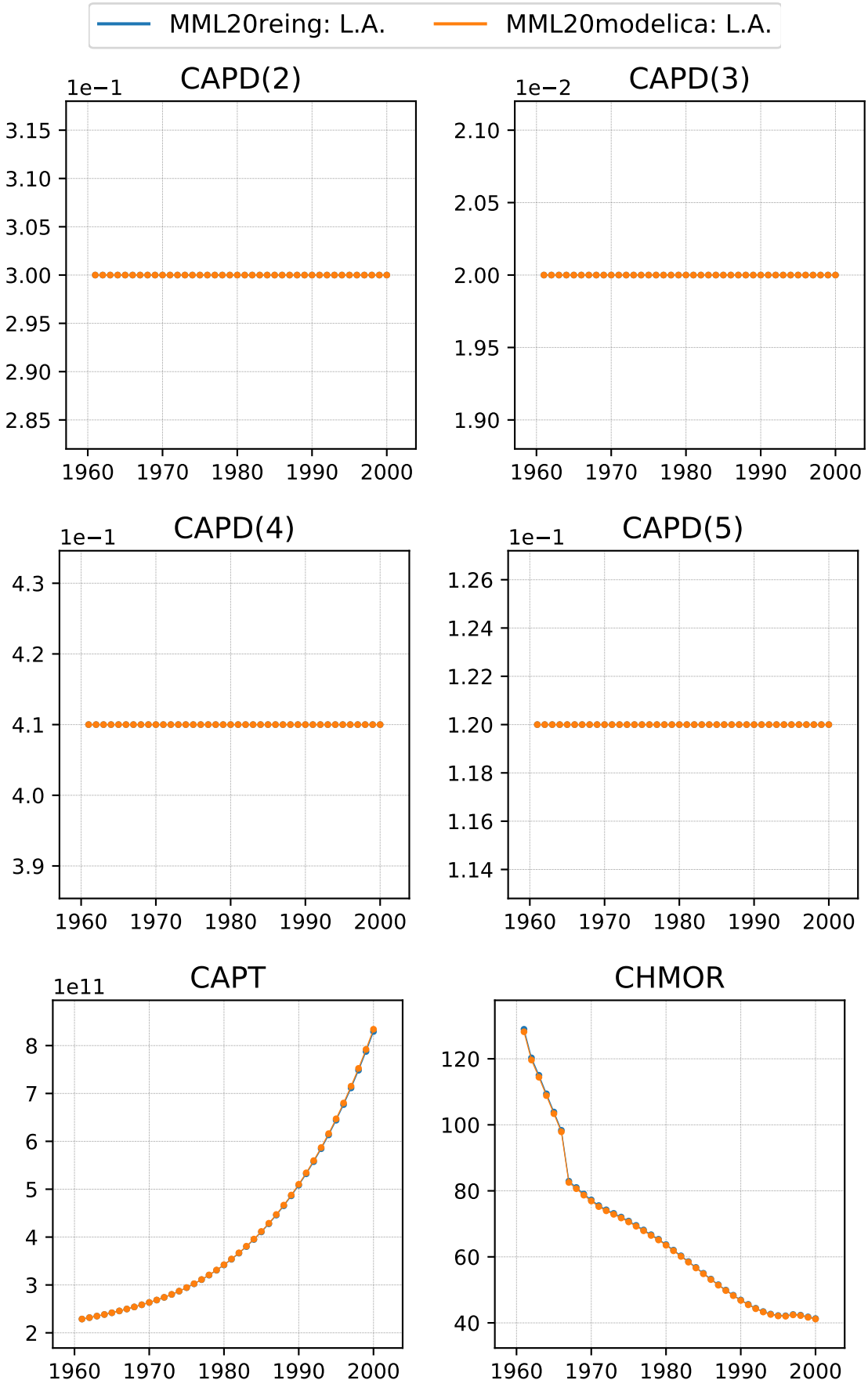

MML20reing: Países Des.
MML20modelica: Países Des.
HOUSER
HSEXFL
PERXFL
POP
1e9
POPR
PROT
1960
1970
1980
1990
2000

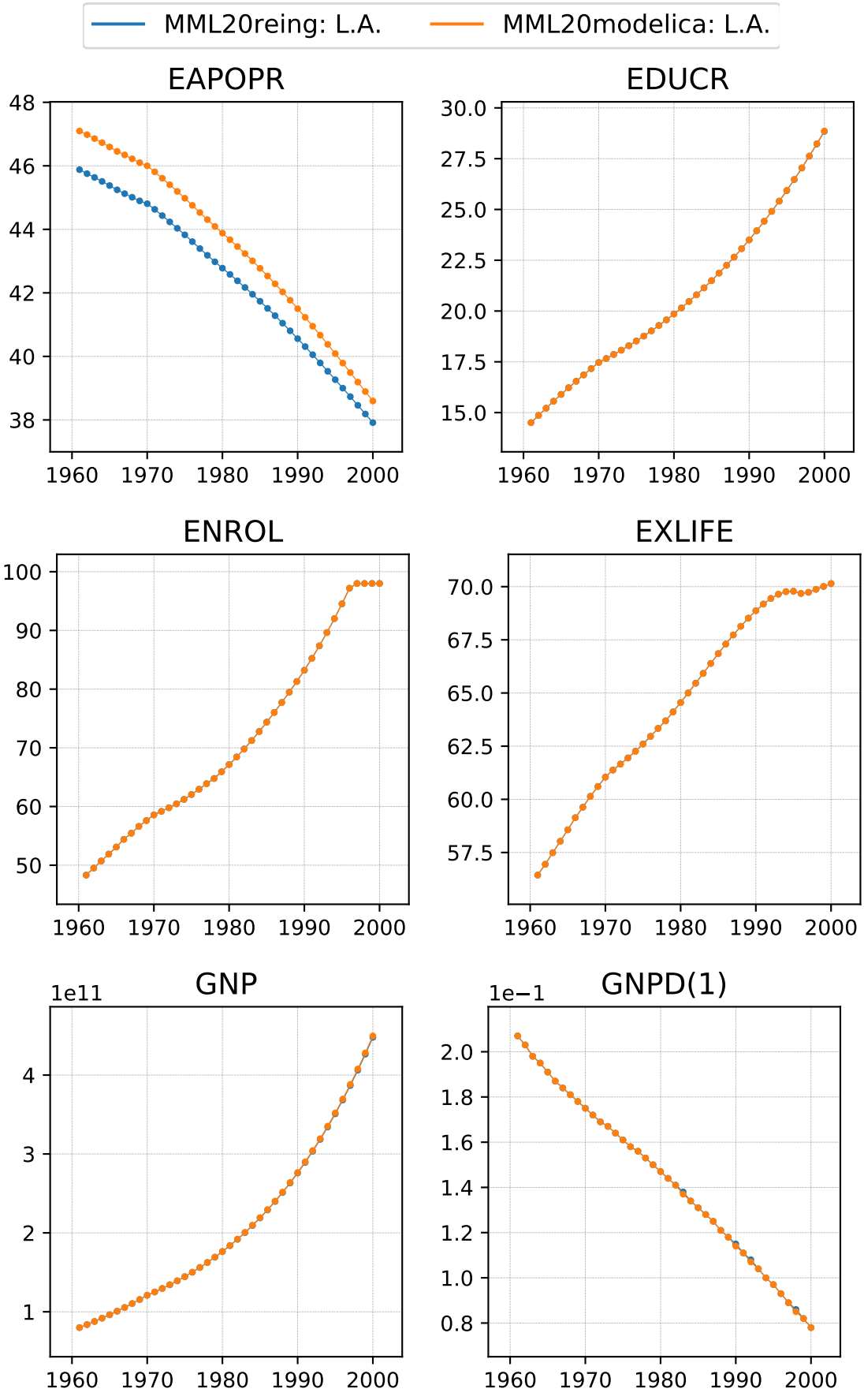

MML20reing: Países Des.
MML20modelica: Países Des.
RLFD(1)
RLFD(2)
1e−2
RLFD(3)
RLFD(4)
RLFD(5)
SEPOPR
1960
1970
1980
1990
2000

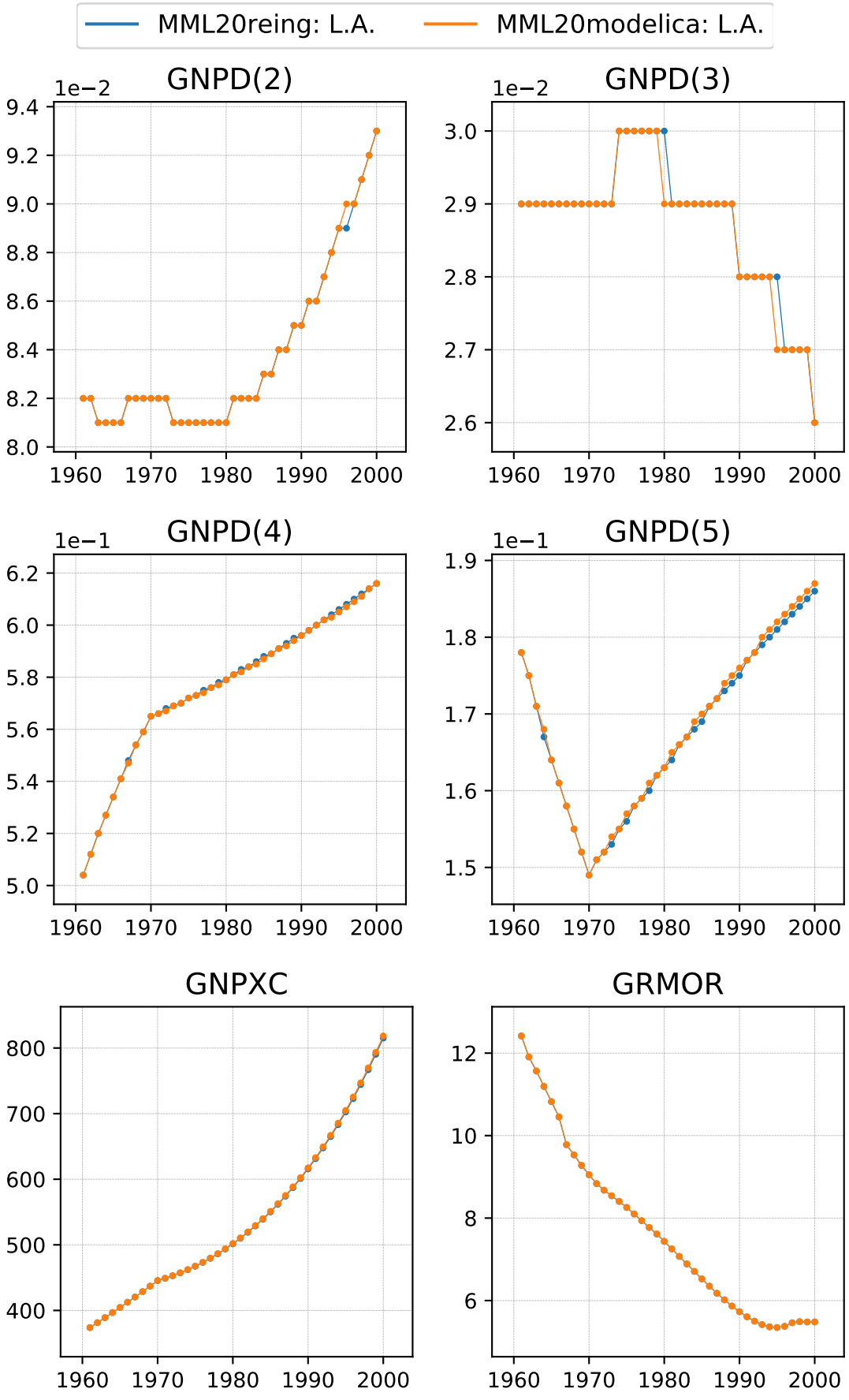


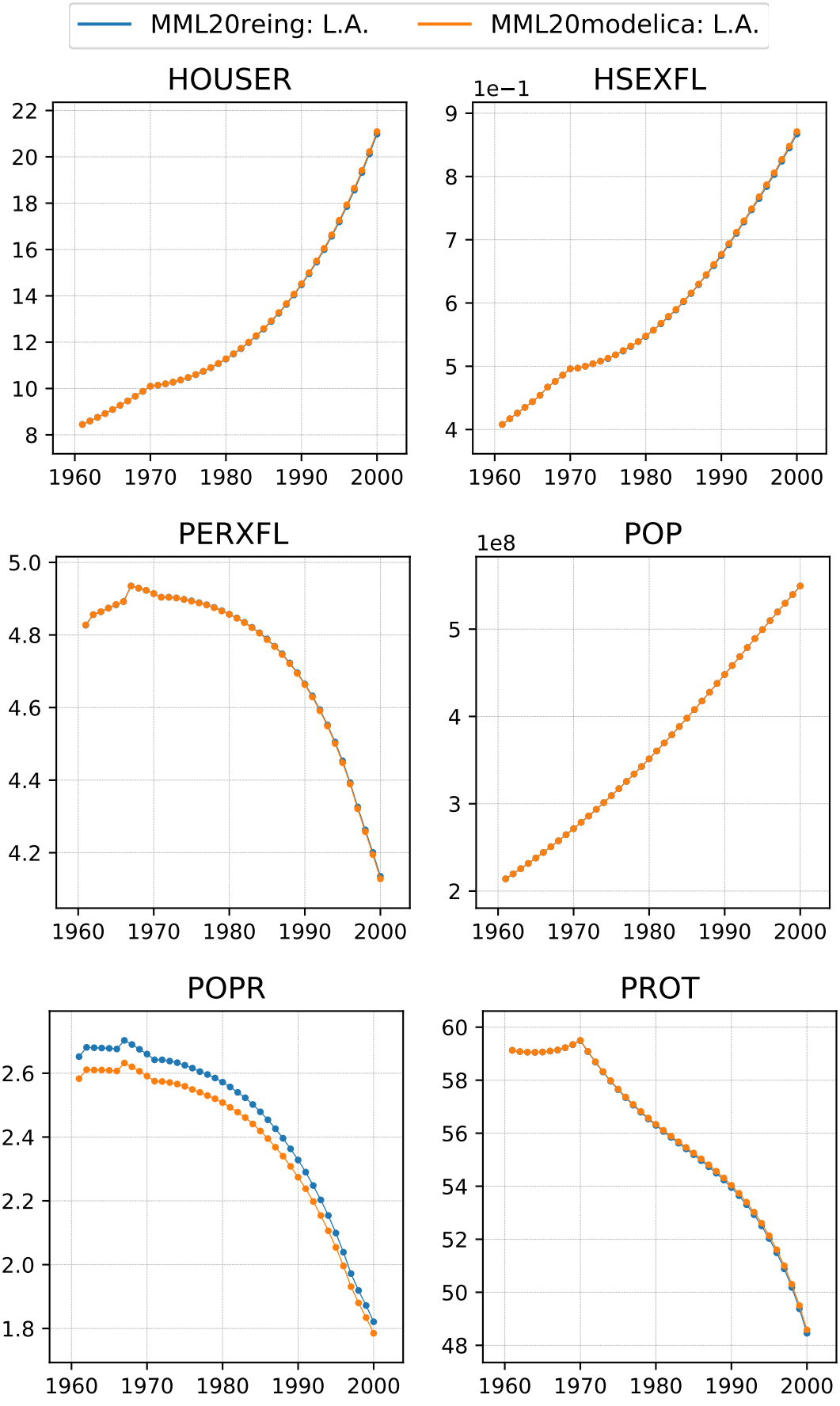
C.2.2. Latinoamérica

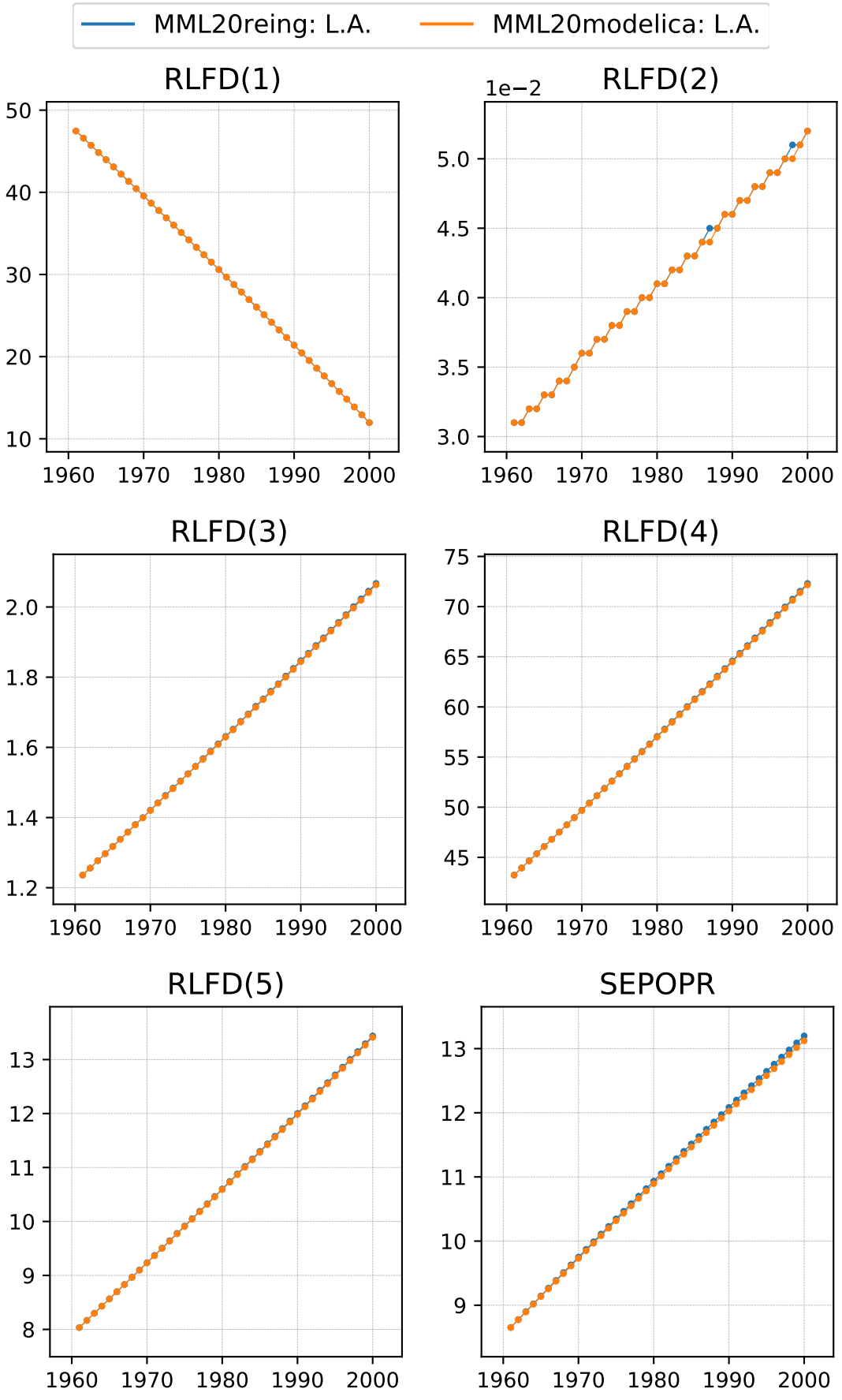

MML20reing: L.A.
MML20modelica: L.A.
0-5
11-70
6-17
BIRTHR
CALOR
1e3
CAPD(1)
1e−1
1960
1970
1980
1990
2000

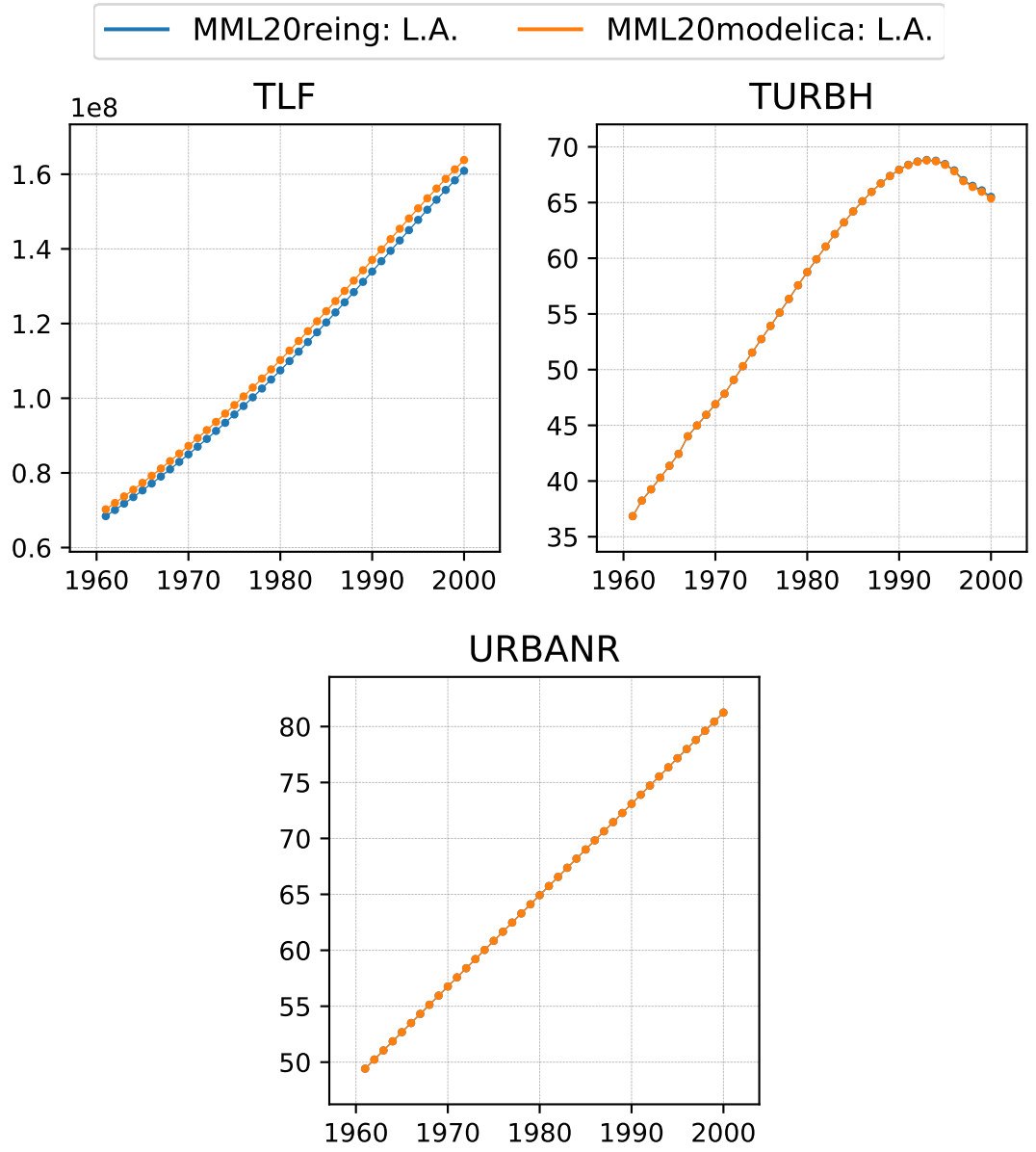

MML20reing: L.A.
MML20modelica: L.A.
CAPD(2)
1e−1
3.15
3.10
3.05
3.00
2.95
2.90
2.85
1960 1970 1980 1990 2000
CAPD(3)
1e−2
2.10
2.05
2.00
1.95
1.90
1960 1970 1980 1990 2000
CAPD(4)
1e−1
4.3
4.2
4.1
4.0
3.9
1960 1970 1980 1990 2000
CAPD(5)
1e−1
1.26
1.24
1.22
1.20
1.18
1.16
1.14
1960 1970 1980 1990 2000
CAPT
1e11
8
7
6
5
4
3
2
1960 1970 1980 1990 2000
CHMOR
120
100
80
60
40
1960 1970 1980 1990 2000


MML20reing: L.A.
MML20modelica: L.A.
EAPOPR
EDUCR
ENROL
EXLIFE
GNP
1e11
GNPD(1)
1e−1
1960
1970
1980
1990
2000

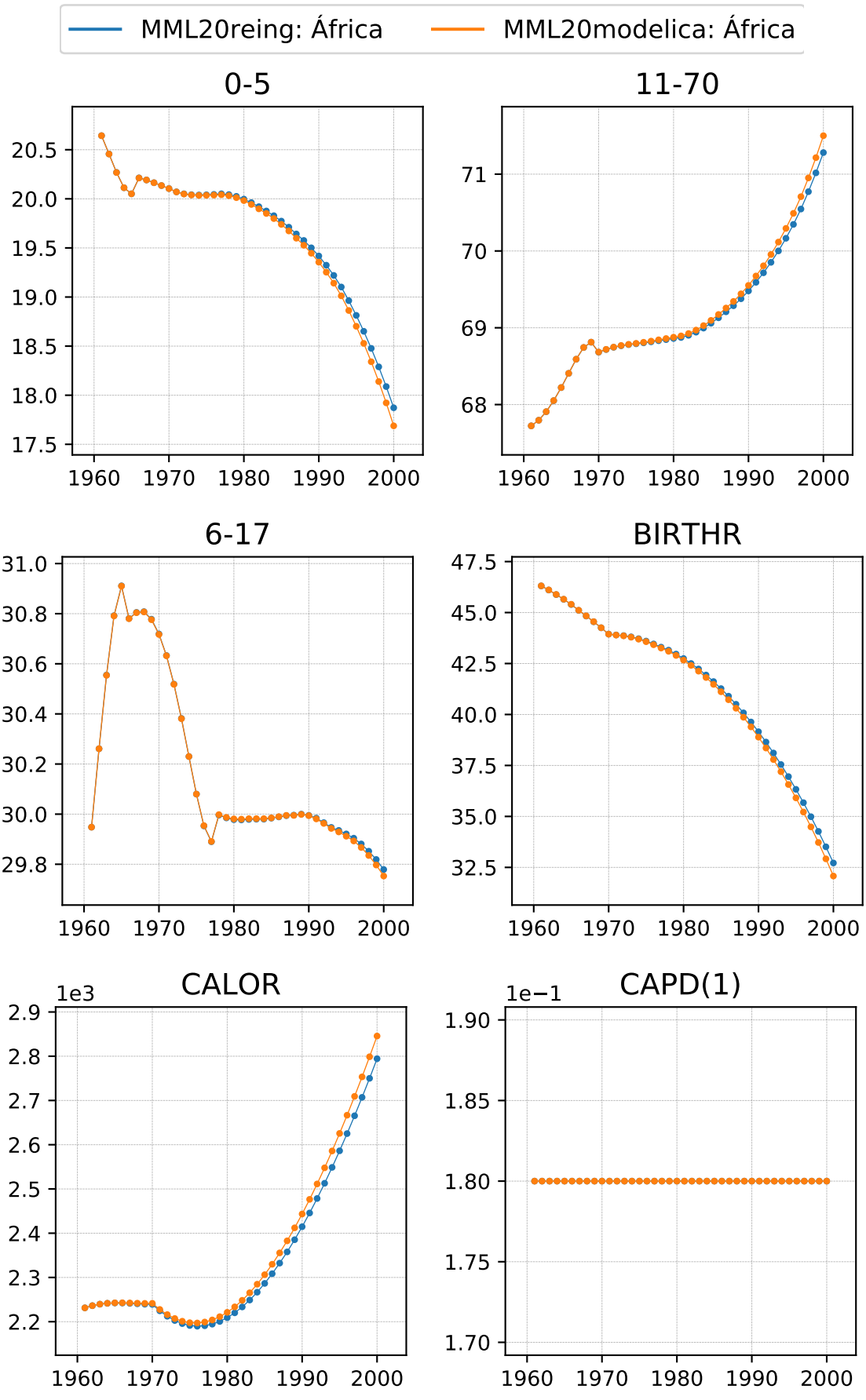

MML20reing: L.A.
MML20modelica: L.A.
GNPD(2)
1e−2
9.4
9.2
9.0
8.8
8.6
8.4
8.2
8.0
1960
1970
1980
1990
2000
GNPD(3)
1e−2
3.0
2.9
2.8
2.7
2.6
GNPD(4)
1e−1
6.2
6.0
5.8
5.6
5.4
5.2
5.0
GNPD(5)
1e−1
1.9
1.8
1.7
1.6
1.5
GNPXC
800
700
600
500
400
GRMOR
12
10
8
6

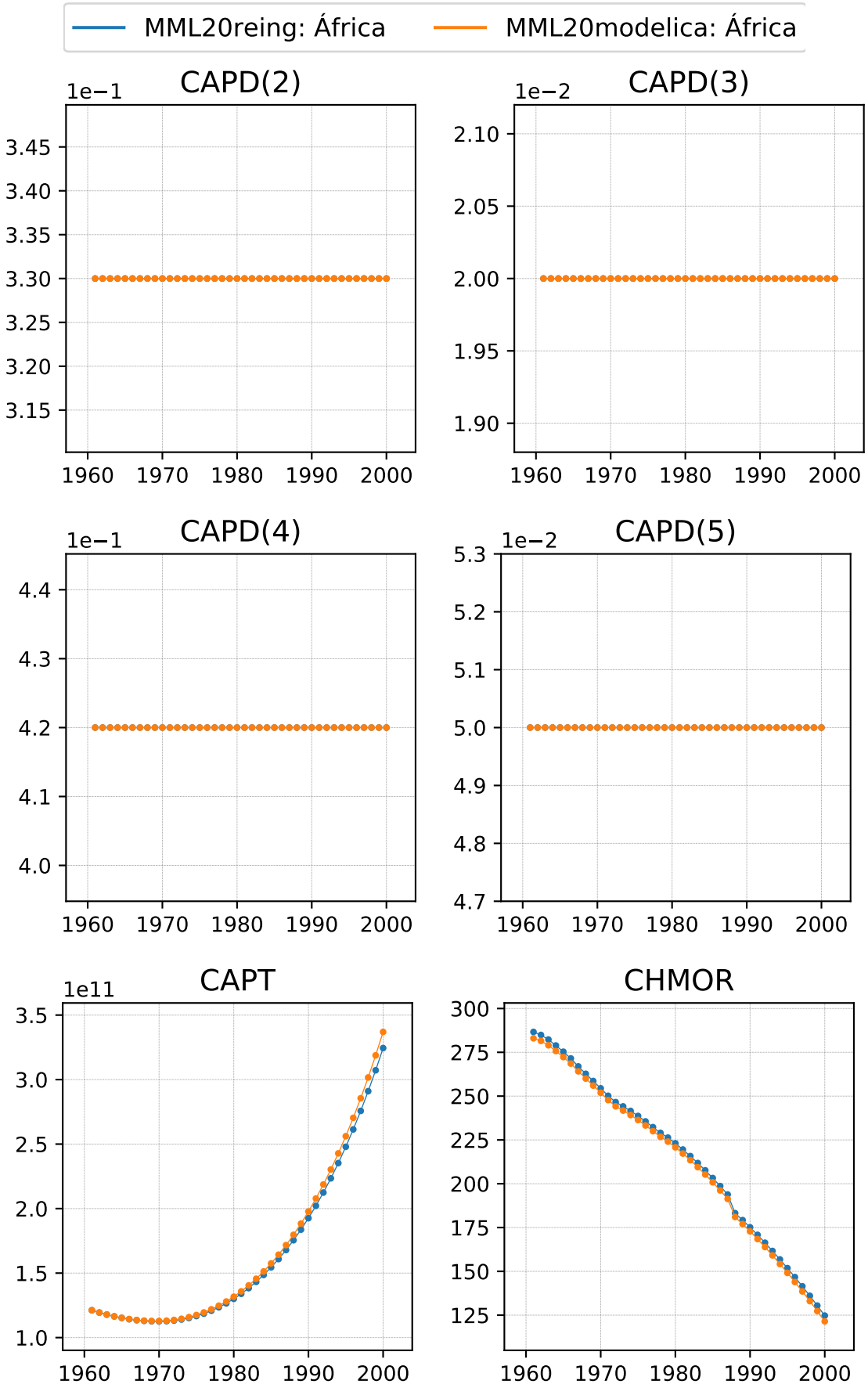

MML20reing: L.A.
MML20modelica: L.A.
HOUSER
HSEXFL
1e−1
PERXFL
POP
1e8
POPR
PROT
1960
1970
1980
1990
2000

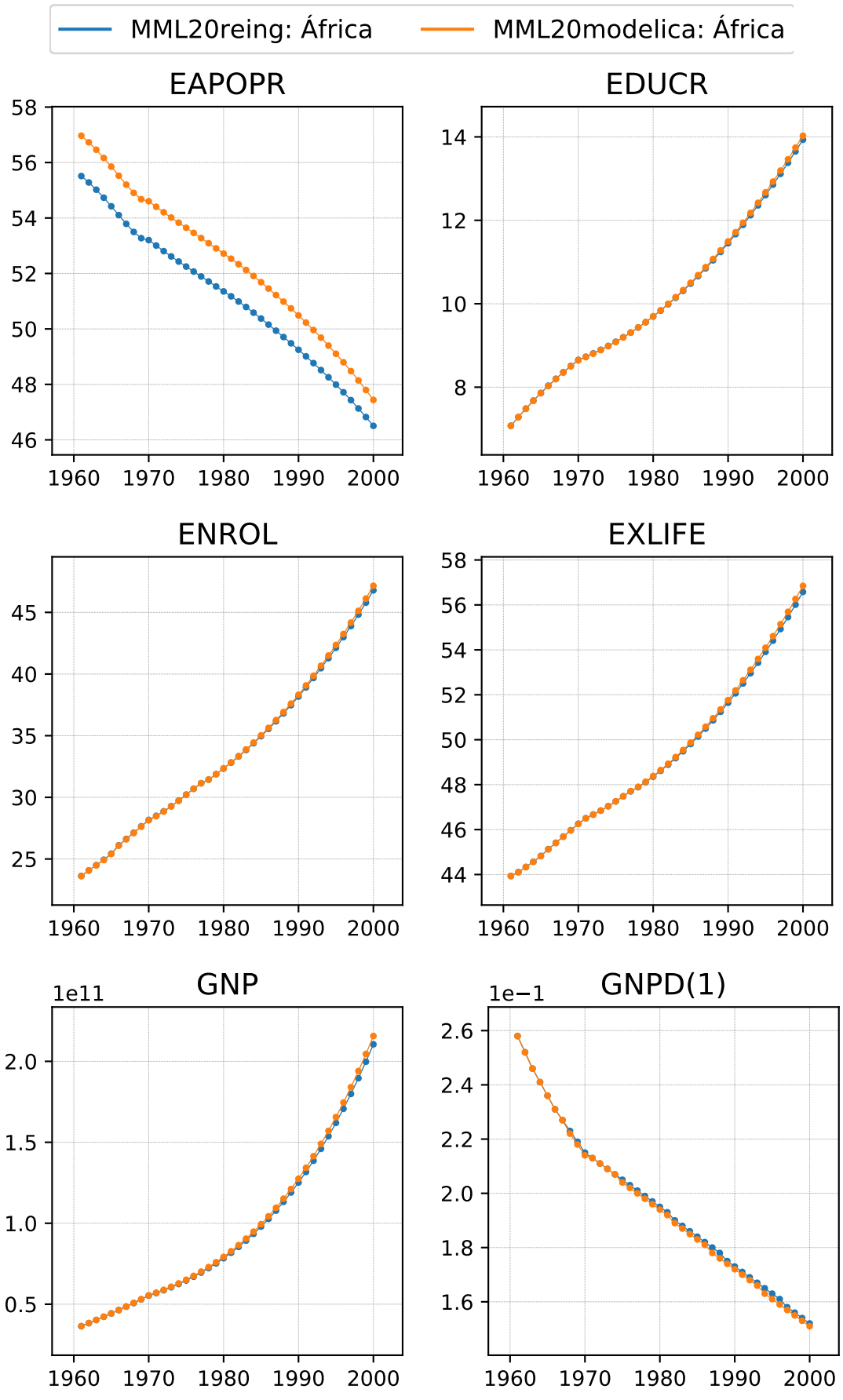

MML20reing: L.A.
MML20modelica: L.A.
RLFD(1)
RLFD(2)
1e−2
RLFD(3)
RLFD(4)
RLFD(5)
SEPOPR
1960
1970
1980
1990
2000

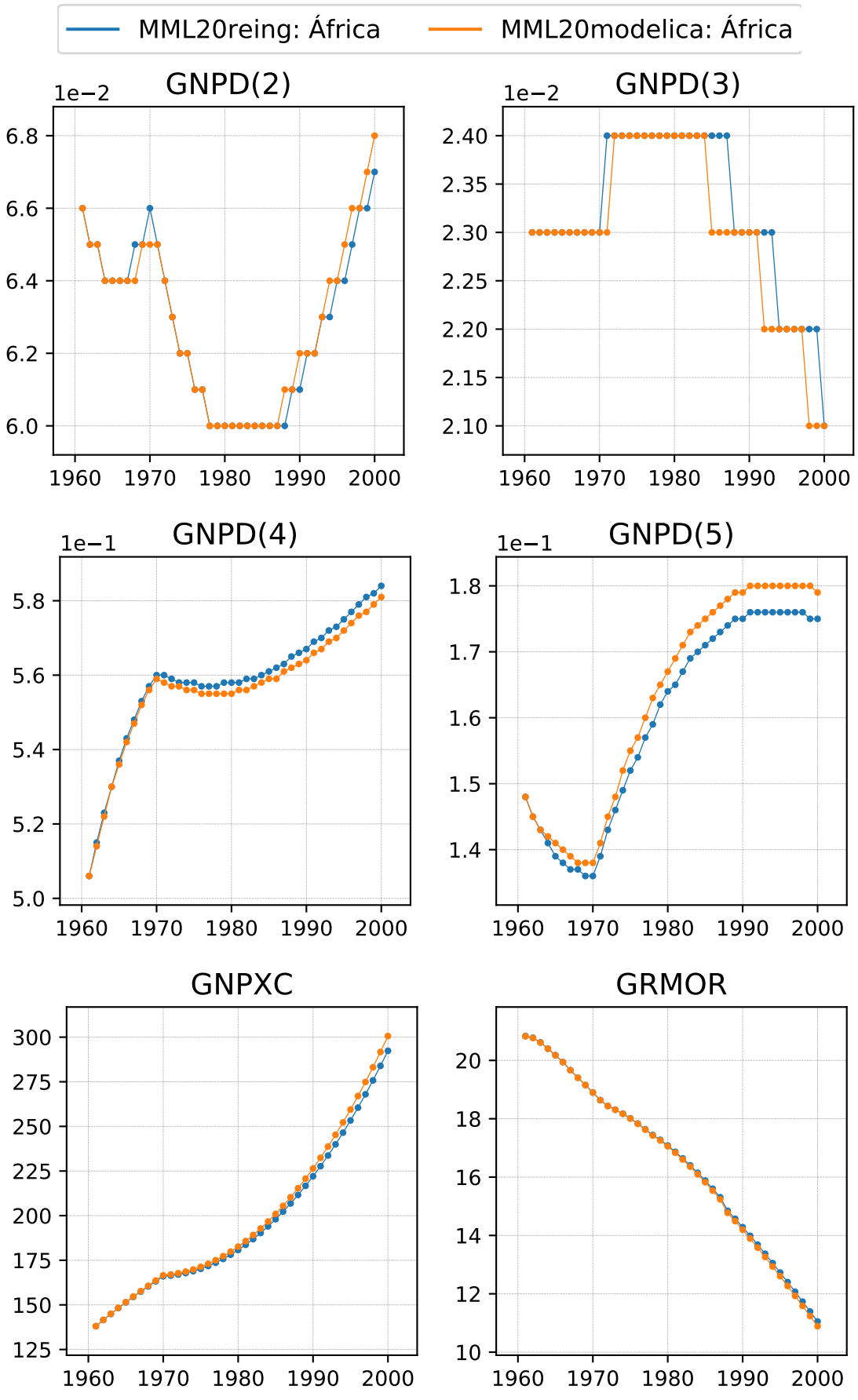

MML20reing: L.A.
MML20modelica: L.A.
TLF
1e8
1.6
1.4
1.2
1.0
0.8
0.6
1960
1970
1980
1990
2000
TURBH
70
65
60
55
50
45
40
35
1960
1970
1980
1990
2000
URBANR
80
75
70
65
60
55
50
1960
1970
1980
1990
2000

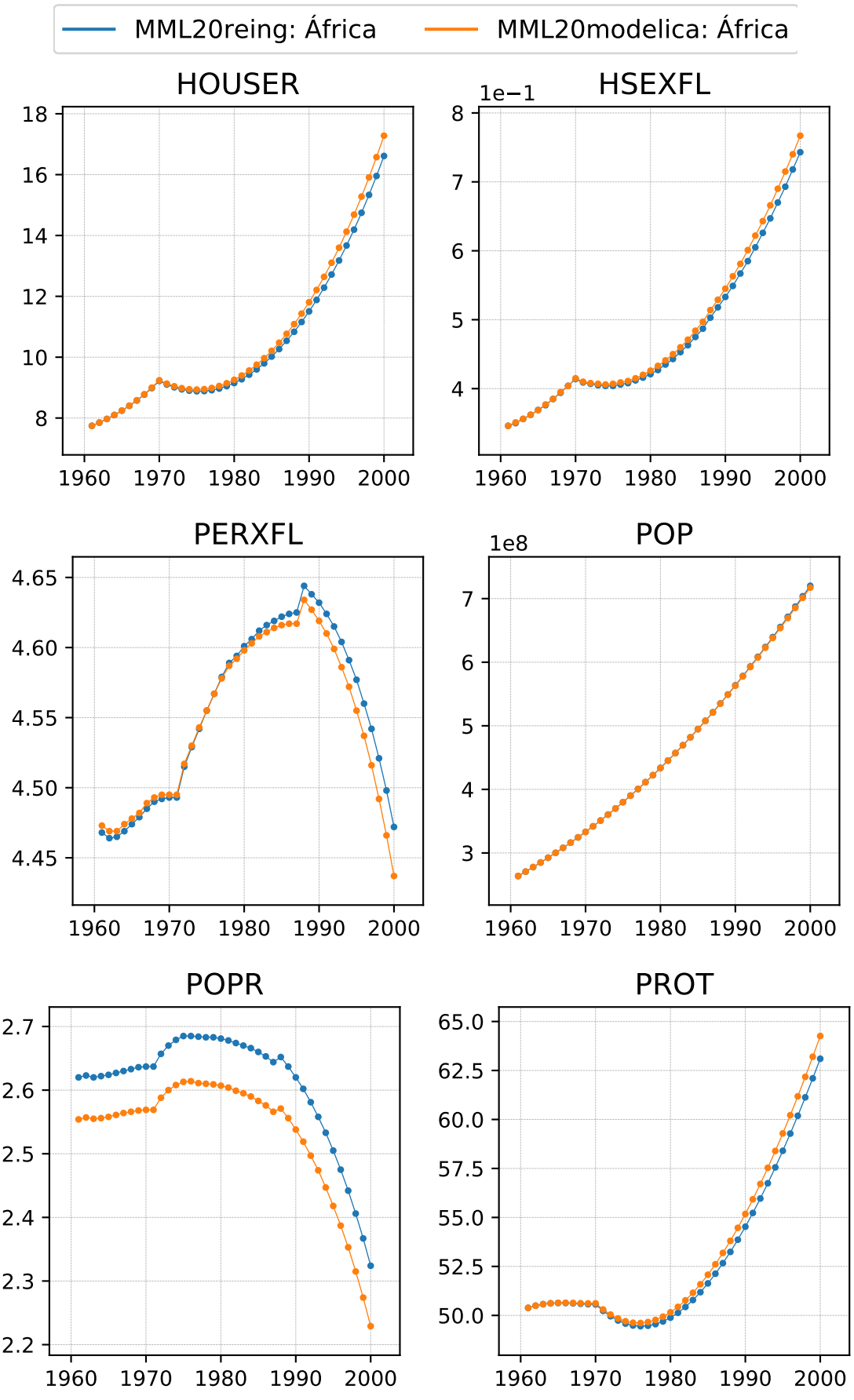
C.2.3. África

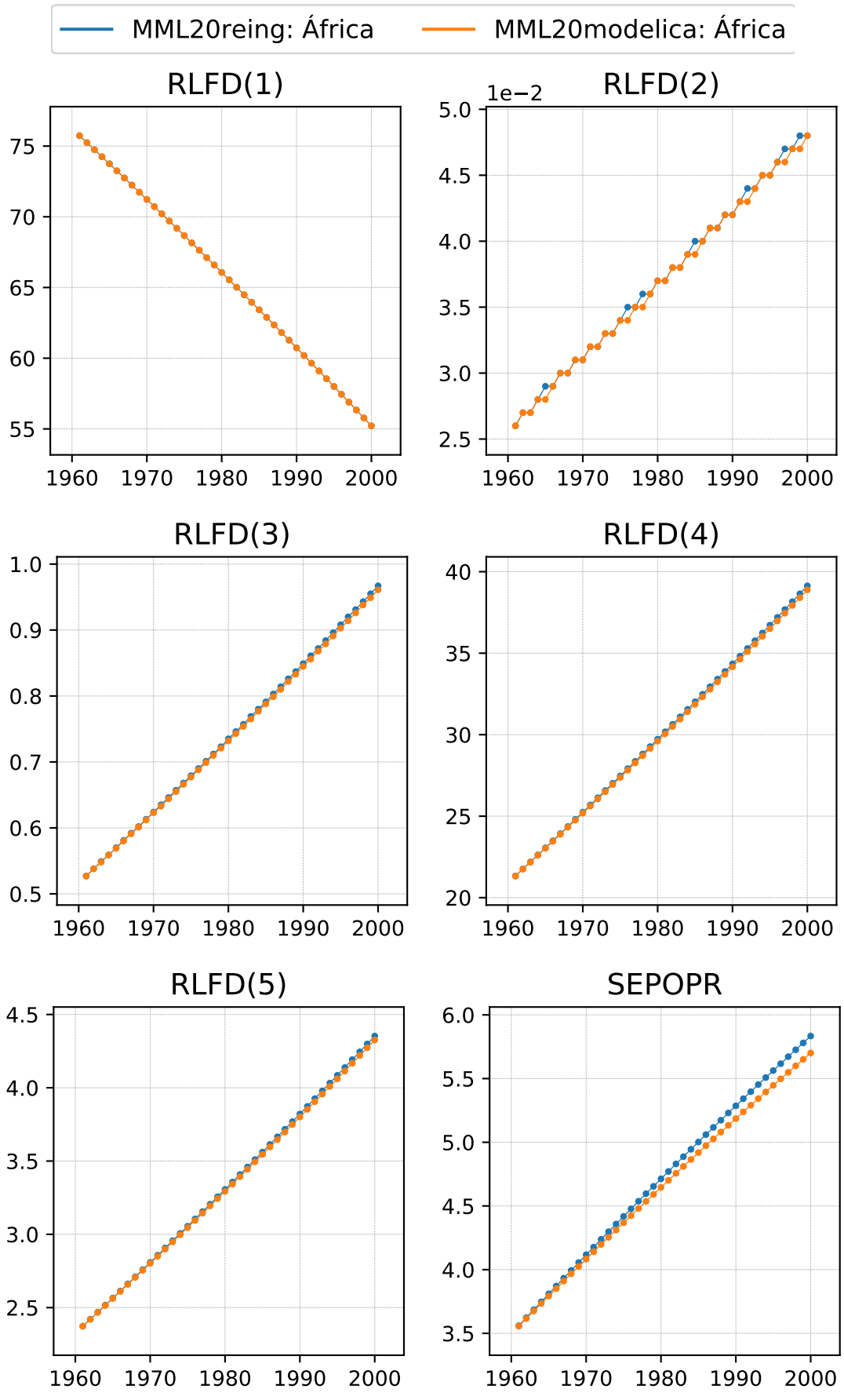

MML20reing: África
MML20modelica: África
0-5
11-70
6-17
BIRTHR
CALOR
1e3
CAPD(1)
1e−1
1960
1970
1980
1990
2000

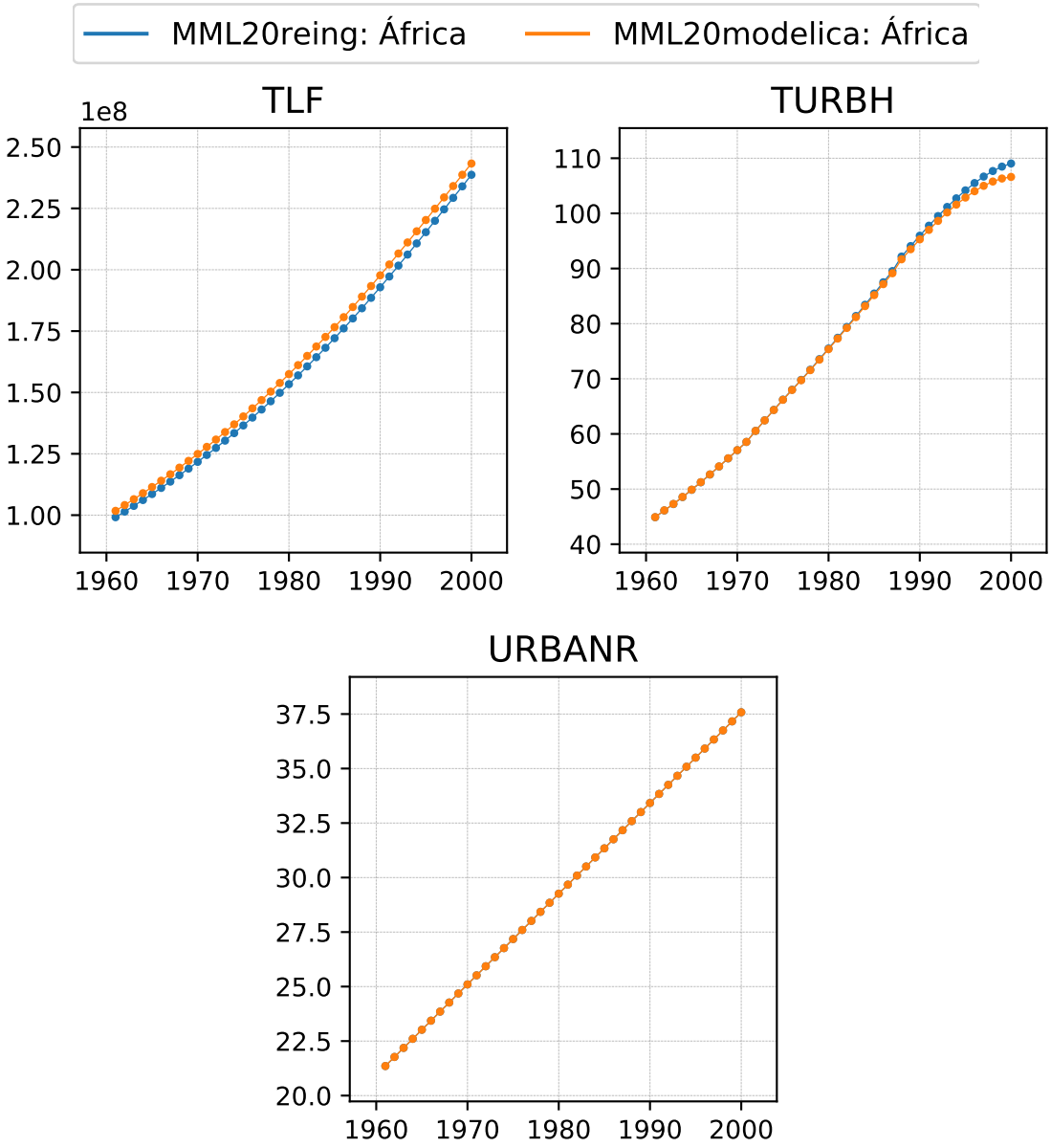

MML20reing: África
MML20modelica: África
CAPD(2)
1e−1
3.45
3.40
3.35
3.30
3.25
3.20
3.15
1960
1970
1980
1990
2000
CAPD(3)
1e−2
2.10
2.05
2.00
1.95
1.90
CAPD(4)
1e−1
4.4
4.3
4.2
4.1
4.0
CAPD(5)
1e−2
5.3
5.2
5.1
5.0
4.9
4.8
4.7
CAPT
1e11
3.5
3.0
2.5
2.0
1.5
1.0
CHMOR
300
275
250
225
200
175
150
125


MML20reing: África
MML20modelica: África
EAPOPR
EDUCR
ENROL
EXLIFE
GNP
1e11
GNPD(1)
1e−1
1960
1970
1980
1990
2000

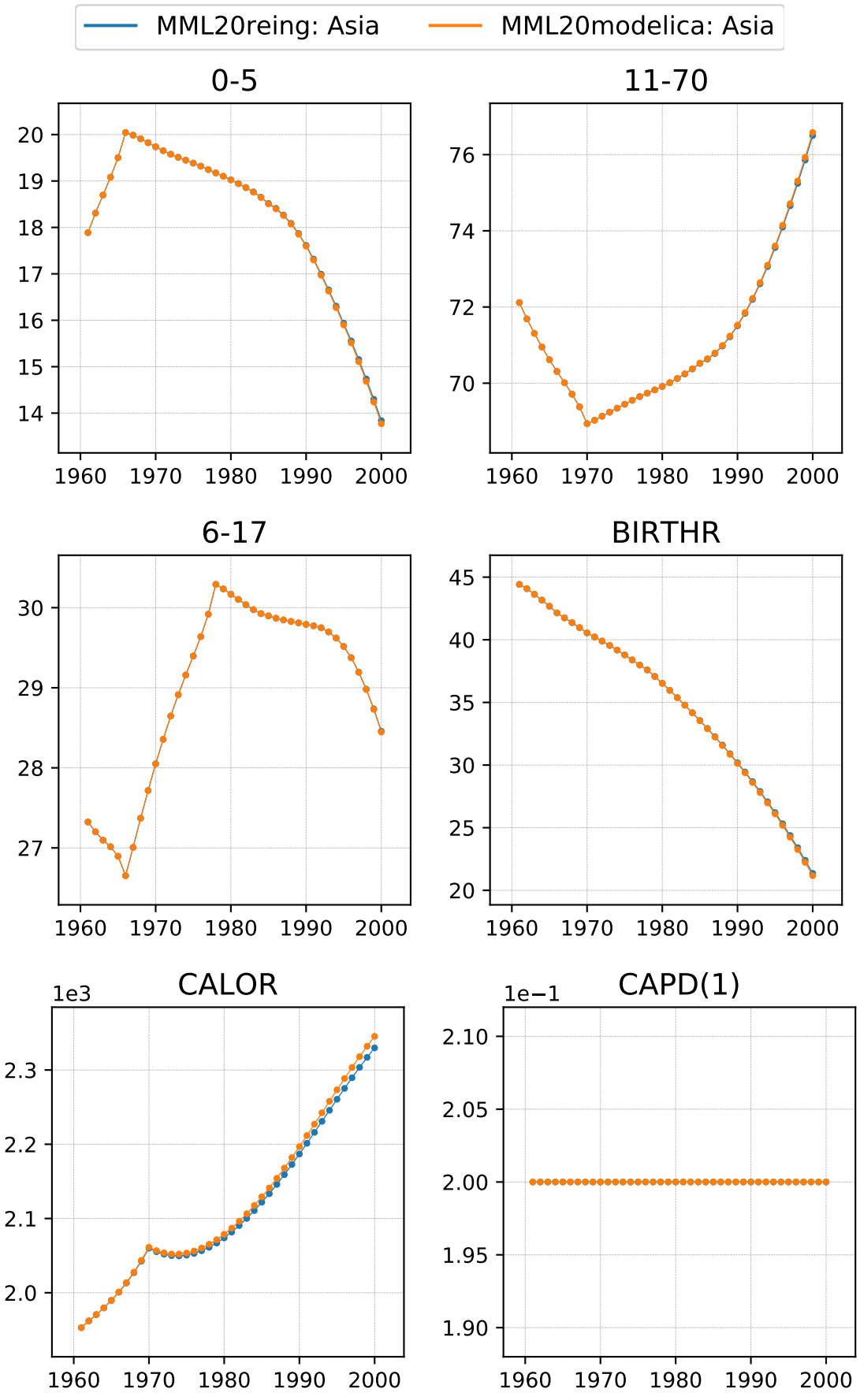

MML20reing: África
MML20modelica: África
GNPD(2)
1e−2
6.8
6.6
6.4
6.2
6.0
1960
1970
1980
1990
2000
GNPD(3)
1e−2
2.40
2.35
2.30
2.25
2.20
2.15
2.10
GNPD(4)
1e−1
5.8
5.6
5.4
5.2
5.0
GNPD(5)
1e−1
1.8
1.7
1.6
1.5
1.4
GNPXC
300
275
250
225
200
175
150
125
GRMOR
20
18
16
14
12
10

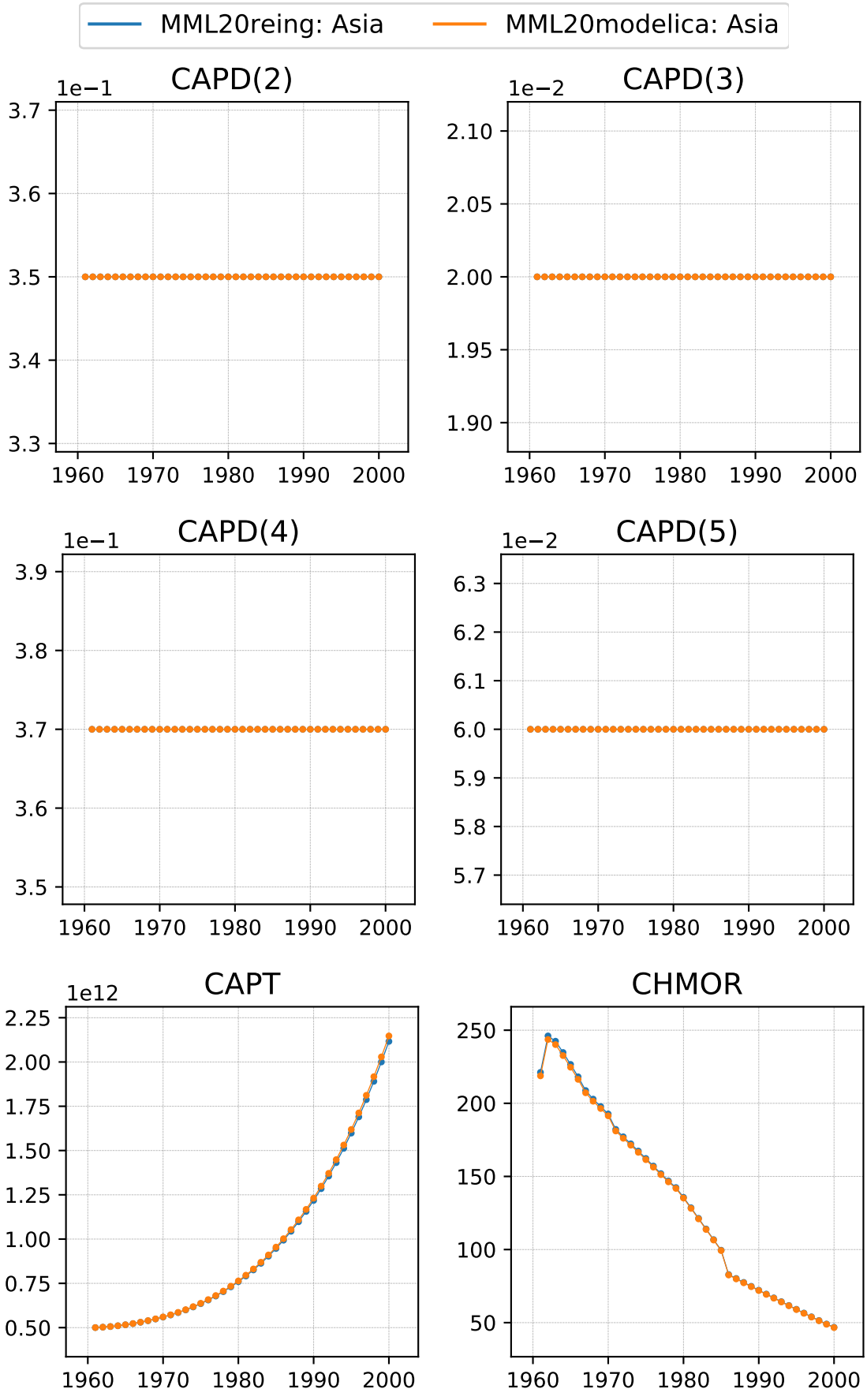

MML20reing: África
MML20modelica: África
HOUSER
HSEXFL
1e−1
PERXFL
POP
1e8
POPR
PROT
1960
1970
1980
1990
2000

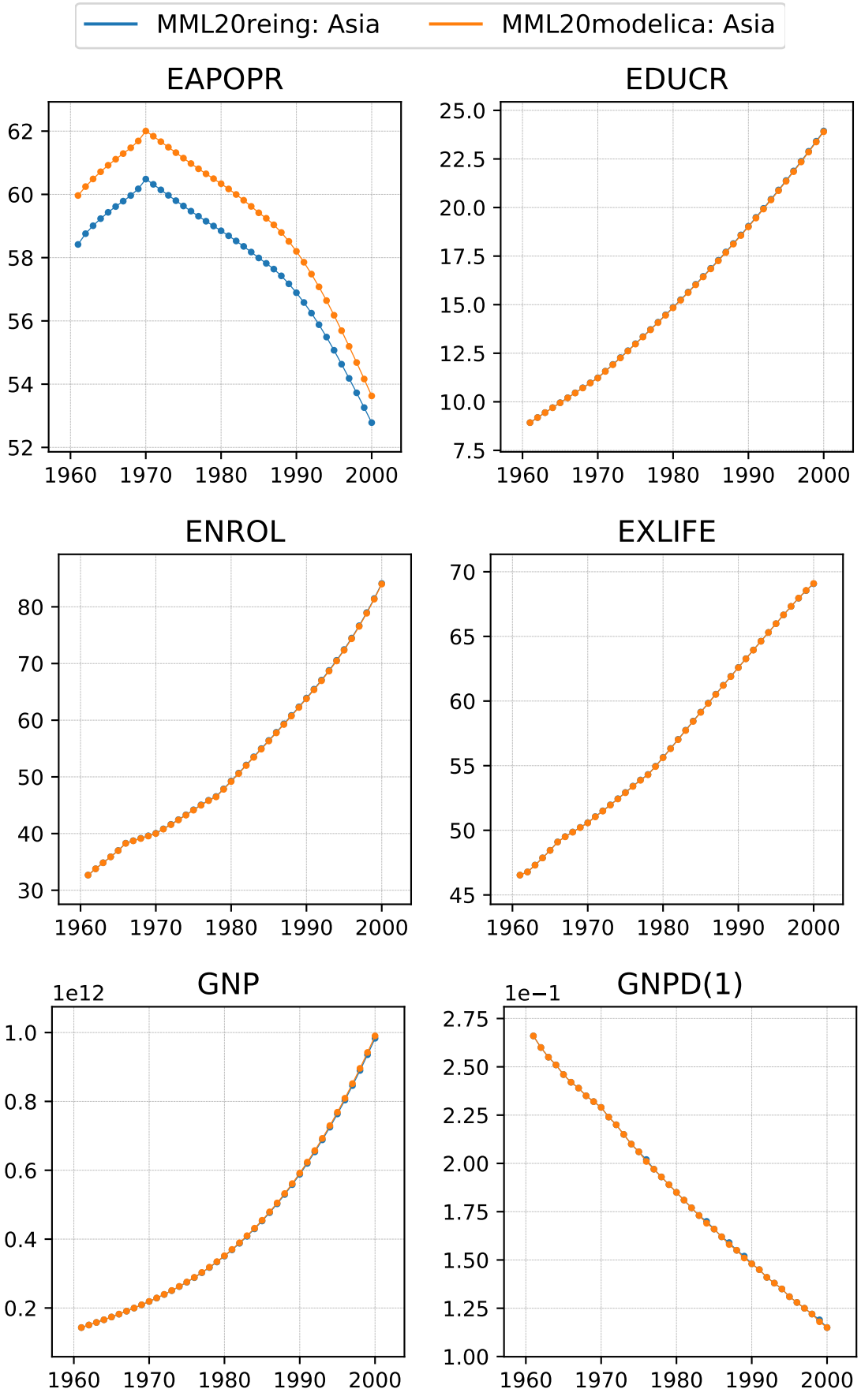

MML20reing: África
MML20modelica: África
RLFD(1)
RLFD(2)
1e−2
RLFD(3)
RLFD(4)
RLFD(5)
SEPOPR
1960
1970
1980
1990
2000

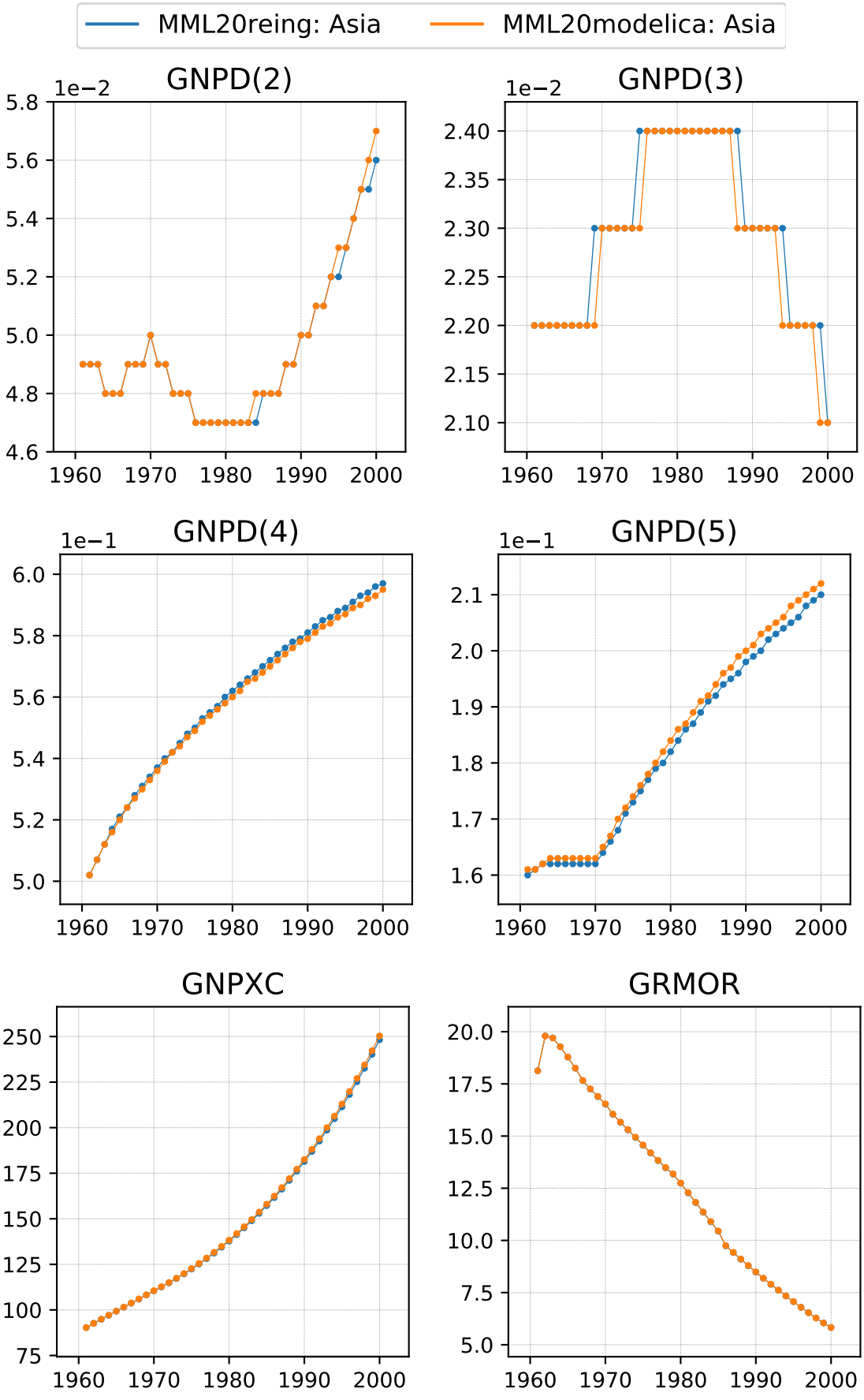

MML20reing: África
MML20modelica: África
TLF
1e8
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1960
1970
1980
1990
2000
TURBH
110
100
90
80
70
60
50
40
URBANR
37.5
35.0
32.5
30.0
27.5
25.0
22.5
20.0

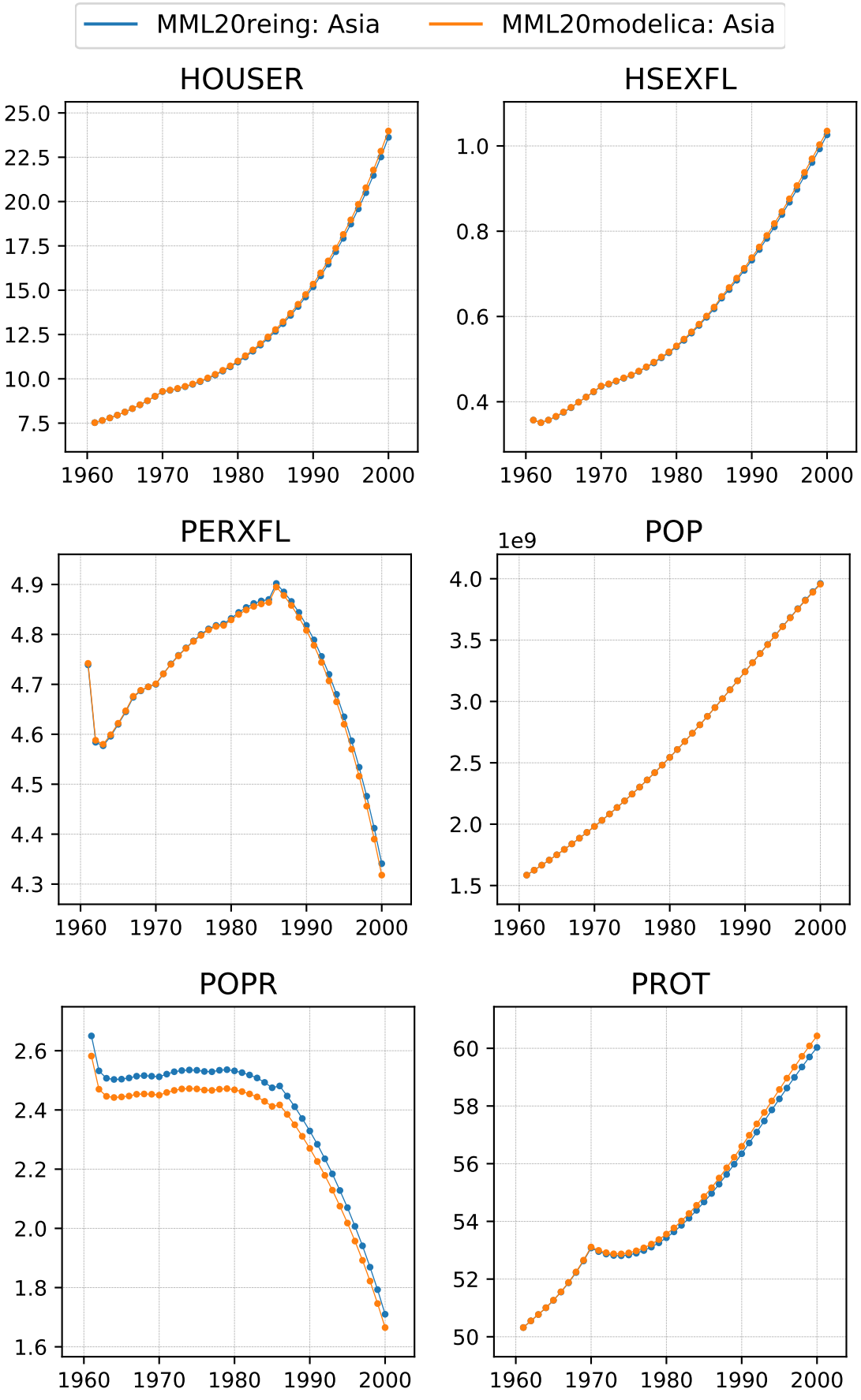
C.2.4. Asia

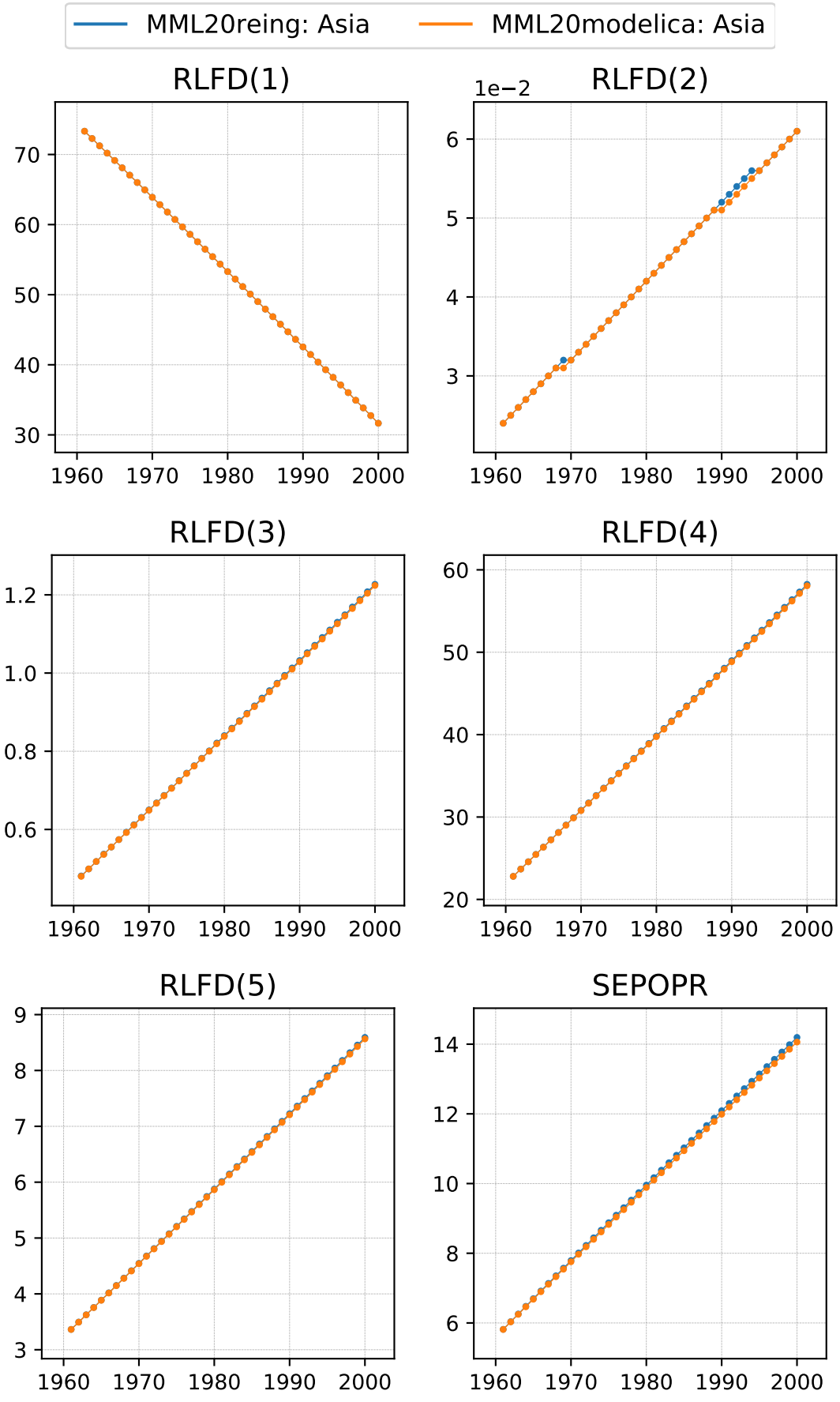

MML20reing: Asia
MML20modelica: Asia
0-5
11-70
6-17
BIRTHR
CALOR
1e3
CAPD(1)
1e−1
1960
1970
1980
1990
2000

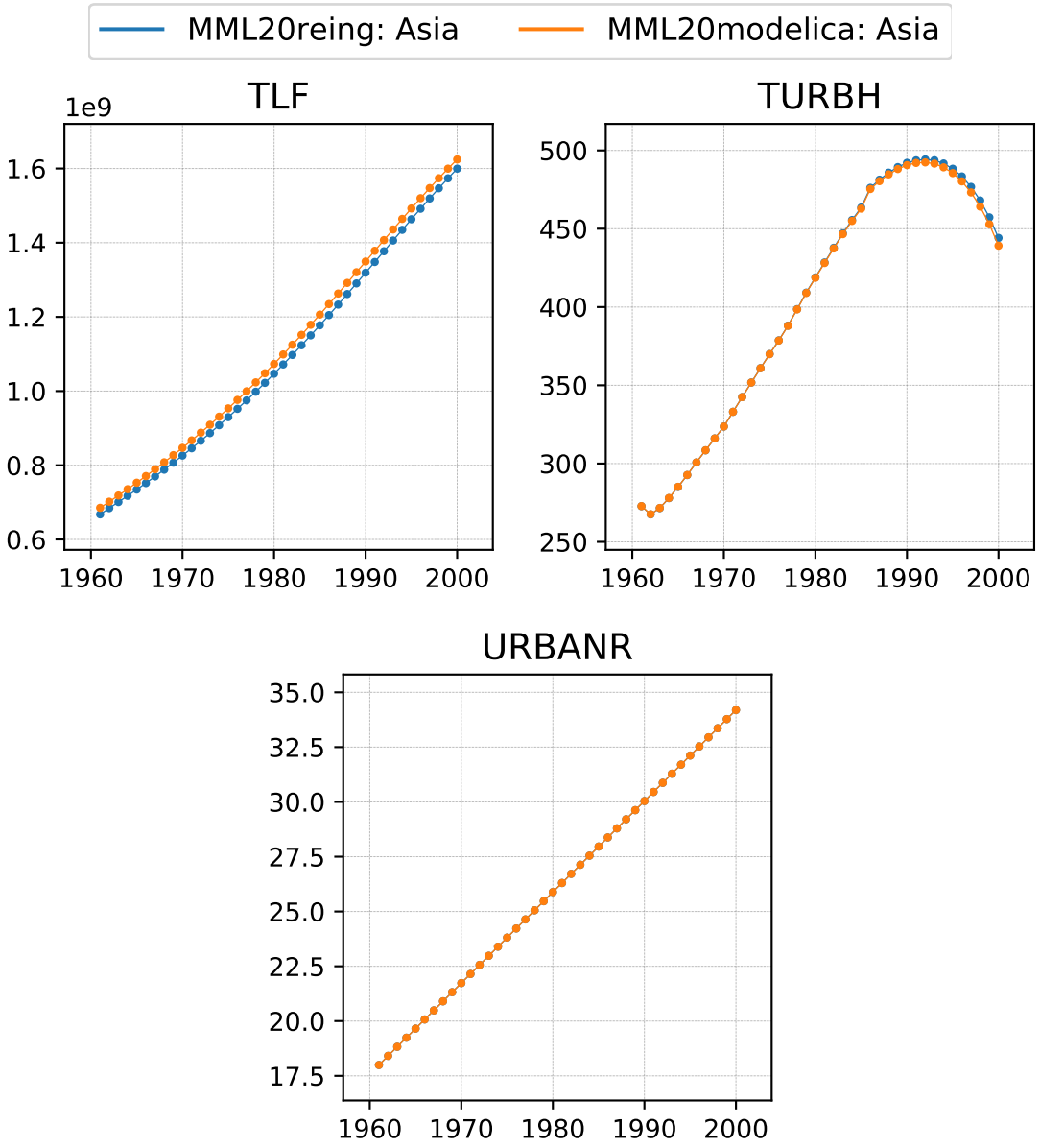

MML20reing: Asia
MML20modelica: Asia
CAPD(2)
1e−1
3.7
3.6
3.5
3.4
3.3
CAPD(3)
1e−2
2.10
2.05
2.00
1.95
1.90
CAPD(4)
1e−1
3.9
3.8
3.7
3.6
3.5
CAPD(5)
1e−2
6.3
6.2
6.1
6.0
5.9
5.8
5.7
CAPT
1e12
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
CHMOR
250
200
150
100
50
1960
1970
1980
1990
2000


MML20reing: Asia
MML20modelica: Asia
EAPOPR
62
60
58
56
54
52
1960
1970
1980
1990
2000
EDUCR
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
1960
1970
1980
1990
2000
ENROL
80
70
60
50
40
30
1960
1970
1980
1990
2000
EXLIFE
70
65
60
55
50
45
1960
1970
1980
1990
2000
GNP
1e12
1.0
0.8
0.6
0.4
0.2
1960
1970
1980
1990
2000
GNPD(1)
1e−1
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1960
1970
1980
1990
2000


MML20reing: Asia
MML20modelica: Asia
GNPD(2)
1e−2
5.8
5.6
5.4
5.2
5.0
4.8
4.6
1960
1970
1980
1990
2000
GNPD(3)
1e−2
2.40
2.35
2.30
2.25
2.20
2.15
2.10
GNPD(4)
1e−1
6.0
5.8
5.6
5.4
5.2
5.0
GNPD(5)
1e−1
2.1
2.0
1.9
1.8
1.7
1.6
GNPXC
250
225
200
175
150
125
100
75
GRMOR
20.0
17.5
15.0
12.5
10.0
7.5
5.0


MML20reing: Asia
MML20modelica: Asia
HOUSER
HSEXFL
PERXFL
POP
1e9
POPR
PROT
1960
1970
1980
1990
2000


MML20reing: Asia
MML20modelica: Asia
RLFD(1)
RLFD(2)
1e−2
RLFD(3)
RLFD(4)
RLFD(5)
SEPOPR
1960
1970
1980
1990
2000


MML20reing: Asia
MML20modelica: Asia
TLF
1e9
1.6
1.4
1.2
1.0
0.8
0.6
1960
1970
1980
1990
2000
TURBH
500
450
400
350
300
250
1960
1970
1980
1990
2000
URBANR
35.0
32.5
30.0
27.5
25.0
22.5
20.0
17.5
1960
1970
1980
1990
2000

Bibliografía

[AB] Modelon AB. jmodelica.org. URL: https://jmodelica.org/.

[Assa] Modelica Association. Modelica standard library. URL: http://fundacionbariloche.org.ar.

[Assb] Modelica Association. Modelica standard library. URL: https://www.modelica.org/libraries.

[Bar76] Fundación Bariloche. Modelo mundial latinoamericano. *Fundación Bariloche*, 1976. URL: http://nuso.org/articulo/modelo-mundial-latinoamericano/.

[BCK12] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 3rd edition, 2012.

[Bru74] Gerhart Bruckmann. Latin american world model proceedings of the second iiasa symposium on global modeling. *International Institute for Applied Systems Analysis, Lavenburg, Austria*, 1974. URL: http://pure.iiasa.ac.at/585/1/CP-76-008.pdf.

[BSK+09] Brückmann, Strenkert, Keller, Wiesner, and Junghanns. Model-based development of a dual-clutch transmission using rapid prototyping and sil. In *International VDI Congress Transmissions in Vehicles*, 2009.

[BSWW99] John Bergey, Dennis Smith, Nelson Weiderman, and Steve Woods. Options analysis for reengineering (oar): Issues and conceptual approach. Technical Report CMU/SEI-99-TN-014, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1999. URL: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=13319.

[CD28] Charles W Cobb and Paul H Douglas. A theory of production. In *Proceedings of the Fortieth Annual Meeting of the American Economic Association*, volume 139, page 165, 1928.

[Cel] François E Cellier. System dymanics - modelica. URL: https://github.com/modelica-3rdparty/SystemDynamics.

[Cel08] François E Cellier. World3 in modelica: Creating system dynamics models in the modelica framework. In *Proc. 6th International Modelica Conference*, 2008.

[CFC+14] Rodrigo Castro, Peter Fritzson, François Cellier, Safa Motesharrei, and Jorge Rivas. Human-nature interaction in world modeling with modelica. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, pages 477–488. Linköping University Electronic Press, 2014.

[Cha00] Richard W. Chadwick. Global modeling: Origins, assessment, and alternative futures. *Simulation & Gaming*, 31(1):50–73, 2000. URL: https://doi.org/10.1177/104687810003100105, arXiv:https://doi.org/10.1177/104687810003100105, doi:10.1177/104687810003100105.

[CJ15] Rodrigo Castro and Pablo Jacovkis. Computer-based global models: From early experiences to complex systems. *Journal of Artificial Societies and Social Simulation*, 18(1):13, 2015. URL: http://jasss.soc.surrey.ac.uk/18/1/13.html, doi:10.18564/jasss.2651.

[CL79] Sam Cole and Henry Lucas. *Models, Planning and Basic Needs*. Elsevier, 1979. URL: https://doi.org/10.1016/c2013-0-03117-9, doi:10.1016/c2013-0-03117-9.

[Com15] Fortran Company. Fortran programming language, 2015. URL: https://www.fortran.com/.

[Cona] Open Source Modelica Consortium. Openmodelica. URL: https://openmodelica.org/.

[Conb] SQLite Consortium. Sqlite. URL: https://sqlite.org.

[CP77] Andrew R. Conn and Tomasz Pietrzykowski. A penalty function method converging directly to a constrained optimum. *SIAM Journal on Numerical Analysis*, 14(2):348–375, April 1977. URL: https://doi.org/10.1137/0714022, doi:10.1137/0714022.

[DCS17] Alejandro Danós, Rodrigo Castro, and Hugo Scolnik. Latin american world model 2.0 web interface, 2017. URL: https://lawm.exp.dc.uba.ar/.

[Eil05] Eldad Eilam. *Reversing: Secrets of Reverse Engineering*. John Wiley & Sons, Inc., USA, 2005.

[Fou] Django Software Foundation. Django. URL: https://djangoproject.com.

[Fow99] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Boston, MA, USA, 1999.

[Fri14] Peter Fritzson. *Principles of object-oriented modeling and simulation with Modelica 3.3: A Cyber-Physical Approach*. John Wiley & Sons, 2014.

[Gal01] Gilberto C Gallopin. The latin american world model (a.k.a. the bariloche model): three decades ago. *Futures*, 33(1):77 – 89, 2001. URL: http://www.sciencedirect.com/science/article/pii/S0016328700000550, doi:https://doi.org/10.1016/S0016-3287(00)00055-0.

[Gir18] Leandro Ariel Giri. *Modelos para un mundo mejor: un análisis histórico y epistemológico de los modelos globales*. PhD thesis, Universidad Nacional de Tres de Febrero, 2018. URL: https://ri.conicet.gov.ar/handle/11336/93958.

[Gmb] ESI ITI GmbH. Simulation x. URL: https://www.simulationx.com/.

[Gro] ESI Group. Cymodelica. URL: https://www.esi-group.com/.

[Gro20] Ana Grondona. Los límites del desarrollo rebatidos desde el sur. circulación, representaciones y olvidos alrededor del modelo mundial latinoamericano. *Pasado Abierto*, 6(11), 2020. URL: https://fh.mdp.edu.ar/revistas/index.php/pasadoabierto/article/view/4071.

[Hig] Highsoft. Highcharts. URL: https://higcharts.com/.

[HSC+76] Amílcar Herrera, Hugo Scolnik, Graciela Chichilnisky, Gilberto Gallopín, Jorge E. Hardoy, International Development Research Centre, Ottawa , Ontario Canada, and Buenos Aires (Argentina) Fundación Bariloche. *Catastrophe or new society? A Latin American world model.* International Development Research Centre, 01 1976.

[HSC+04] Amílcar O. Herrera, Hugo D. Scolnick, Gabriela Chichilinsky, Gilberto C. Gallopin, Jorge E. Hardoy, Diana Mosovich, Enrique Oteiza, Gilda L. de Romero, Carlos E. Suárez, and Luis Talavera. *¿Catástrofe o nueva sociedad?: modelo mundial latinoamericano 30 años después.* IIED, 2004. URL: http://hdl.handle.net/10625/31088.

[HSM75] Michael Hopkins, Hugo Scolnik, and Mick Mclean. Basic needs, growth and redistribution; a quantitative approach. *International Labour Organization*, 1975. URL: https://ideas.repec.org/p/ilo/ilowps/991635323402676.html.

[Int15] Intel. Intel fortran compiler, 2015. URL: https://software.intel.com/en-us/fortran-compilers.

[Joh81] W.A. Johr. The bariloche model: a latin american world model. *Schweizerische Zeitschrift fur Volkswirtschaft und Statistik*, 1981. URL: https://www.ncbi.nlm.nih.gov/pubmed/12157694.

[jT] The jQuery Team. jquery. URL: https://jquery.com/.

[Lei68] E.G. Leigh. *Ecological Role of Volterra's Equations.* Lectures on mathematics in the life sciences. Princeton University, 1968. URL: https://books.google.com.ar/books?id=ILcKNQAACAAJ.

[LY13] Ge Liang and Liang Yu. *Qualiy Driven Re-engineering Framework.* PhD thesis, School of Computing, Blekinge Institute of Technology, Sweden, 2013. URL: http://www.diva-portal.org/smash/get/diva2:829429/FULLTEXT01.

[MA] Modelica association. URL: https://modelica.org.

[Map] MapleSoft. Maplesim. URL: https://www.maplesoft.com/.

[Mar08] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship.* Prentice Hall PTR, USA, 1 edition, 2008.

[Mic15] Microsoft. Microsoft visual studio, 2015. URL: https://visualstudio.microsoft.com/.

[MLS17] Modelica language specification, version 3.4, 2017. URL: https://modelica.org/documents/ModelicaSpec34.pdf.

[MMRBI72] Dennis Meadows, Donella Meadows, Jørgen Randers, and William W Behrens III. *The Limits to Growth: A Report for the Club of Rome's Project on the Predicament of Mankind.* Universe Books, New York, 1972.

[MRB82] Donella Meadows, John Richardson, and Gerhart Bruckmann. *Groping in the dark : the first decade of global modelling.* Wiley, Chichester West Sussex New York, 1982.

[Nat76] United Nations. Modelo mundial latinoamericano - síntesis informativa. *Fundación Bariloche*, 1976.

[NBT03] Charles Newman, John Batteh, and Michael Tiller. Spark-ignited engine cycle simulation in modelica. In *Proc. 3rd International Modelica Conference*, 2003.

[Nor75] W.D. Nordhaus. World modelling from the bottom up. Technical report, IIASA, Laxenburg, Austria, March 1975. URL: http://pure.iiasa.ac.at/id/eprint/506/.

[Ott] Martin Otter. Modelica overview. URL: https://www.modelica.org/education/educational-material/lecture-material/english/ModelicaOverview.pdf.

[Pow64] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964. URL: http://dx.doi.org/10.1093/comjnl/7.2.155, arXiv:/oup/backfile/content_public/journal/comjnl/7/2/10.1093/comjnl/7.2.155/2/070155.pdf, doi:10.1093/comjnl/7.2.155.

[QST15] Juan Quiroga, Angeles Smart, and Laura Totonelli. Fundación bariloche: Vigencia del modelo mundial latinoamericano (1970), cuatro décadas después. *IX Jornadas Latinoamericanas de Estudios Sociales de la Ciencia y la Tecnología*, 06 2015. URL: https://www.researchgate.net/publication/278963279_Fundacion_Bariloche_Vigencia_del_Modelo_Mundial_Latinoamericano_1970_cuatro_decadas_despues.

[Res] Wolfram Research. Wolfram systemmodeler. URL: https://www.wolfram.com/system-modeler/.

[Ric82] John M. Richardson. A decade of global modelling. *Futures*, 14(2):136 – 145, 1982. URL: http://www.sciencedirect.com/science/article/pii/0016328782900878, doi:https://doi.org/10.1016/0016-3287(82)90087-8.

[RLS77] Carlos A. Ruiz, Irene Loiseau, and Hugo Scolnik. Adaptation of the bariloche model to brazil. In *Proceedings of the Meeting of Experts on the Applicability of GLobal Modelling Techniques to Integrated Planning in Developing Countries*, 1977.

[RLS79] Carlos A. Ruiz, Irene Loiseau, and Hugo D. Scolnik. Adaptation of the bariloche model to a national scenario. In Sam Cole and Henry Lucas, editors, *Models, Planning and Basic Needs*, pages 63 – 73. Pergamon, 1979. URL: https://www.sciencedirect.com/science/article/pii/B9780080237329500164, doi:https://doi.org/10.1016/B978-0-08-023732-9.50016-4.

[Sau15] Sandra Sauro. Cosmovisiones, utopías y polémicas a propósito del club de roma y del modelo mundial latinoamericano. *Revista de la Red de Intercátedras de Historia de América Latina Contemporánea 28 (Segunda época)*, pages 28–45, 2015. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=5769499.

[SFL+77] Hugo Scolnik, Ana Friedlander, Irene Loiseau, M Martinez, and Carlos A. Ruiz. Handbook of the latin american world model, ss.77/ws/11. *UNESCO, Paris*, July 1977.

[SIE] SIEMENS. Amesim. URL: https://www.plm.automation.siemens.com/global/en/products/simcenter/simcenter-amesim.html.

[SKT03] Erik Surewaard, Eckhard Karden, and Michael Tiller. Advanced electric storage system modeling in modelica. In *Proc. 3rd International Modelica Conference*, 2003.

[SM02] S. Soejima and T. Matsuba. Application of mixed mode integration and implicit inline integration at toyota. In *Proc. 2nd International Modelica Conference*, 2002.

[sPL03] Robert C. seacord, Daniel Plakosh, and Grace A. Lewis. *Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices*. Addison-Wesley Longman Publishing Co., Inc., USA, 2003.

[ST] Inc. Scientific Toolworks. Understand - scitools. URL: https://scitools.com/features/.

[Sysa] Dassault Systèmes. Catia systems. URL: https://www.3ds.com/products-services/catia/disciplines/systems-engineering/.

[Sysb] Dassault Systèmes. Dymola. URL: https://www.3ds.com/products-services/catia/products/dymola/.

[TBD03] Michael Tiller, Paul Bowles, and Mike Dempsey. Development of a vehicle modeling architecture in modelica. In *Proc. 3rd International Modelica Conference*, 2003.

[Tea] Bootstrap Team. Bootstrap. URL: https://getbootstrap.com/.

[TSGT08] E. D. Tate, Michael Sasena, Jesse Gohl, and Michael Tiller. Model embedded control: A method to rapidly synthesize controllers in a modeling environment. In *Proc. 6th International Modelica Conference*, 2008.

[WvGC08] Henrik Wigermo, Johannes von Grundherr, and Thomas Christ. Implementation of a modelica online optimization for an operating strategy of a hybrid powertrain. In *Proc. 6th International Modelica Conference*, 2008.

Los documentos que integran la Biblioteca PLACTED fueron reunidos por la Cátedra Libre Ciencia, Política y Sociedad (CPS). Contribuciones a un Pensamiento Latinoamericano, que depende de la Universidad Nacional de La Plata. Algunos ya se encontraban disponibles en la web y otros fueron adquiridos y digitalizados especialmente para ser incluidos aquí.

Mediante esta iniciativa ofrecemos al público de forma abierta y gratuita obras representativas de autores/as del **Pensamiento Latinoamericano en Ciencia, Tecnología, Desarrollo y Dependencia (PLACTED)** con la intención de que sean utilizadas tanto en la investigación histórica, como en el análisis teórico-metodológico y en los debates sobre políticas científicas y tecnológicas. Creemos fundamental la recuperación no solo de la dimensión conceptual de estos/as autores/as, sino también su posicionamiento ético-político y su compromiso con proyectos que hicieran posible utilizar las capacidades CyT en la resolución de las necesidades y problemas de nuestros países.

PLACTED abarca la obra de autores/as que abordaron las relaciones entre ciencia, tecnología, desarrollo y dependencia en América Latina entre las décadas de 1960 y 1980. La Biblioteca PLACTED por lo tanto busca particularmente poner a disposición la bibliografía de este período fundacional para los estudios sobre CyT en nuestra región, y también recoge la obra posterior de algunos de los exponentes más destacados del PLACTED, así como investigaciones contemporáneas sobre esta corriente de ideas, sobre alguno/a de sus integrantes o que utilizan explícitamente instrumentos analíticos elaborados por estos.

Derechos y permisos

En la Cátedra CPS creemos fervientemente en la necesidad de liberar la comunicación científica de las barreras que se le han impuesto en las últimas décadas producto del avance de diferentes formas de privatización del conocimiento.

Frente a la imposibilidad de consultar personalmente a cada uno/a de los/as autores/as, sus herederos/as o los/as editores/as de las obras aquí compartidas, pero con el convencimiento de que esta iniciativa abierta y sin fines de lucro sería del agrado de los/as pensadores/as del PLACTED, ***requerimos hacer un uso justo y respetuoso de las obras, reconociendo y citando adecuadamente los textos cada vez que se utilicen, así como no realizar obras derivadas a partir de ellos y evitar su comercialización.***

A fin de ampliar su alcance y difusión, la Biblioteca PLACTED se suma en 2021 al repositorio ESOCITE, con quien compartimos el objetivo de "recopilar y garantizar el acceso abierto a la producción académica iberoamericana en el campo de los estudios sociales de la ciencia y la tecnología".

Ante cualquier consulta en relación con los textos aportados, por favor contactar a la cátedra CPS por mail: catedra.cienciaypolitica@presi.unlp.edu.ar