

动态随机一般均衡(DSGE)模型： 理论、方法和Dynare实践

李向阳 著

清华大学出版社
北京

内 容 简 介

本书为 DSGE 领域入门级专著, 讲述了 DSGE 模型的基本建模理论和求解逻辑, 并示例如何在 Dynare 中加以实现。DSGE 的基本建模理论涵盖了经典的 RBC 模型、新古典模型、RBC 模型的拓展 (MIU、CIA)、新凯恩斯模型和中等规模 DSGE 模型, 并着重介绍了税收设定、可变资本利用率、投资调整成本、投资边际效率冲击、金融加速器机制、开放经济建模、利率钉住、两种定义下的最优货币政策和 DSGE 中微观福利度量方法等问题。DSGE 的求解逻辑涵盖了一阶 (线性化、B&K 方法、Schur 方法和待定系数法) 和二阶的扰动算法、脉冲响应的计算、随机模拟和确定性模拟、最大似然和贝叶斯参数估计及其逻辑。此外, 对 Dynare 的安装、使用、编译逻辑、语法表示、使用技巧和错误排除等也进行了详细介绍并加以示例。

本书最大的特点就是理论密切联系实践。在讲述 DSGE 理论的同时, 辅以大量的模型示例来讲述求解过程和一阶条件, 并在 Matlab 和 Dynare 中加以编程实现, 并提供每个模型甚至每个图形对应的 mod 文件和 Matlab 源代码, 使得读者知其然, 更知其所以然。另外, 本书在讲述建模理论的同时, 注重分析其经济含义与背景, 帮助读者建立经济学直觉。如在讲解 MIU 建模理论时, 将其和著名的费雪货币数量理论相结合加以论述; 在讲解 CIA 建模理论时, 将其和弗里德曼规则相联系。本书中的专业术语注重英文再现, 可帮助初学者准确把握概念。虽然本书定位于 DSGE 领域入门级学习资料, 但目标读者群并不仅限于宏观经济专业的学生 (如高年级本科生、硕士和博士研究生)。对于那些从事宏观经济研究的学者, 特别是对 DSGE 模型不甚熟悉的学者, 以及银行、政府、企事业单位等相关研究机构的宏观经济研究人员, 本书同样适用。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

动态随机一般均衡 (DSGE) 模型: 理论、方法和 Dynare 实践 / 李向阳著. — 北京: 清华大学出版社, 2018
ISBN 978-7-302-49774-5

I. ①动… II. ①李… III. ①金融—经济模型 IV. ①F830.49

中国版本图书馆 CIP 数据核字 (2018) 第 037303 号

责任编辑: 刘 洋

封面设计: 李召霞

版式设计: 方加青

责任校对: 王凤芝

责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:

经 销: 全国新华书店

开 本: 187mm×235mm

印 张: 31.5

字 数: 586 千字

版 次: 2018 年 12 月第 1 版

印 次: 2018 年 12 月第 1 次印刷

定 价: 138.00 元

产品编号: 074344-01



前言

2017年11月，时值深秋，这是一个收获的季节，也是本书初稿完成的日子。这本书的问世，是对我近十年的求学和工作的一个小结。写作始于2016年炎热的夏季，得益于前期充足准备，已有部分手稿、源代码，这使得本书仅在16个多月的时间内便迅速成稿。

然而，写作是一个需要耐心和毅力的过程。针对某一个问题往往能花上好几天的时间来解决，通过反复调试源代码，找出问题所在。记得在写作“金融加速器机制及其Dynare实现”这一章节时，由于对原有模型进行了扩展，在银行均衡条件中加入成本溢价参数，结果发现原有的代码不能正常运行。为此对思路稍作调整，花了近一周的时间来解决这个问题。在碰到无法解决的问题时，那种无助感只有自己最清楚，有时真想放弃。在寒冷的冬夜和炎热的夏夜，长时间坐在计算机前不停地查阅资料、书写公式和编写代码，是考验耐心和毅力的过程，是一个艰辛的过程，更是一个蜕变的过程。

写作的过程是一个不断学习和自我提高的过程。只有当认真地书写每一个字符、每一个均衡条件，认真编写程序，才能发现真知，发现新知。在解决上述成本溢价参数问题的过程中，我彻底弄懂了Matlab调试模型的基本逻辑，并熟悉了大部分相关命令。这本书充满着各种数学公式、图形和源代码、复杂的结构和近500页的篇幅，使得Word 2003往往不堪重负，各种崩溃和排版问题蜂拥而至。因此，需要在解决DSGE模型相关专业问题的同时，还必须应对技术与排版问题。当然，解决这些问题之后获得的满足感和成就感也是巨大的。比如：对图形的标注和引用，明明已经定义并存在某个标签，但在插入题注时，标签却不显示已定义的标签。此时只有通过再定义同样的标签，才能“唤醒”Word 2003，找回已有的标签。不过遗憾的是至今仍有两个未解决的问题：问题一，“磁盘已满或打开文件过多”而无法保存的问题（明明不是磁盘已满的问题）；问题二，“Word正在分析文档……”无论如何按Esc键都不能取消而导致Word不能操作和假死问题（Word提示按Esc键取消）。有人告诉我这是Word本身的内在问题，最后只能作罢。

本书开始写作时就注重DSGE建模理论与Dynare实践相结合。因此，每一章都有相应的Matlab的源文件(*.m)和Dynare的模型文件(*.mod)，使得读者能够复制所有的结果，包括Matlab图形，所有代码均在Matlab 2012b和Dynare v4.4.1或v4.4.3版本内调试通过。

代码的下载可移步至清华大学出版社官网 (<http://www.tup.tsinghua.edu.cn/>)、经管之家论坛 (<http://bbs.pinggu.org/>) 或者直接发邮件 (dynare@foxmail.com) 索要。

本书的写作过程中，对专业术语注重英文再现，重要的专有术语都在中文后面标注了英文，以使得初学者准确把握专有术语。在参考文献排版上，本来打算采取脚注的形式，但同一篇参考文献在一章或一节中多次出现，多次注释造成版面浪费，因此决定采取章尾列示的方式。虽然牺牲了一点便利，但仍不失为一种好的解决办法。此外，对于参考文献，本书尽最大努力减少二手文献，以给读者最准确、最及时的引用源。另外，对于重要的、引用率高的期刊文献，在每章首次引用时，注明期刊名称或缩写，以帮助读者加深对文献的印象。如：Kydland & Prescott(1982, *Econometrica*)、Christiano, Motto & Rostagno(2014, *AER*)。

本书共 9 章内容，除第 0 章为总体介绍性内容外，其余 8 章为实质性内容。8 章中有 3 章内容介绍了 Dynare 软件及其应用；其余 5 章则详细介绍了 DSGE 基础的建模理论与方法，并穿插 Dynare 代码讲解。总体内容分为两篇。第 1 篇含前 5 章内容，定位于初级水平；第 2 篇为后 4 章内容，定位于进阶水平。

第 0 章为 DSGE 模型简介。首先第 1 节介绍了本书的写作初衷和相关背景，然后笔锋一转开始介绍 DSGE 模型 (分析框架) 的发展、作用及其面临的批评、问题；第 3 节则介绍了两种典型 DSGE 模型构建，给初学者一个轮廓性概览；最后一节对宏观经济模型数据库 MMB 做了简单的功能性介绍。

第 1 章介绍了 DSGE 模型的求解逻辑，分为 3 节内容。第 1 节和第 2 节分别介绍了 DSGE 模型的一阶和二阶求解逻辑，并结合 Dynare 的求解惯例加以说明；第 3 节则分别介绍 4 个方面相关的问题，包括稳态及其计算、AR(1) 过程的校准、随机差分方程的求解以及 HP 滤波分析的基本原理，并在 Matlab 中编程实现。

第 2 章为 Dynare 的安装、配置、运行和管理，并对如何获取使用帮助做了简单的说明，相信学完本章之后所有初学者都能够正确配置和运行 Dynare。

第 3 章和第 6 章分主题集中介绍 Dynare 的语法、运行原理及使用方法、技巧等。

第 3 章分为 10 节内容，为基础性应用。从最简单的 DSGE 模型的例子开始，到变量的分类、表达方式、Dynare 的运行原理、求解表示，再到确定性和随机性模拟，以及最后的参数估计，几乎覆盖了 Dynare 基础应用的大部分内容。

第 6 章介绍了 Dynare 的进阶应用，分为 5 节内容，分别从模型文件的循环调用、脉冲响应函数自定义编程、二阶模拟中的相关问题、常见错误示例到最后的宏命令编程，涵盖了高阶应用的相当部分内容。

第4章介绍了RBC模型和NK模型的基本理论。算是DSGE建模理论中较为重要的一块，也是初学者必须掌握的内容，因此不惜笔墨，整个内容占据了全书篇幅的1/4，分为3节内容。在RBC模型介绍时，着重从定性和定量两个方面力争呈现出RBC理论的真实面貌。在RBC模型拓展方面，则从MIU、CIA、投资、边际技术冲击等经典方面进行讲解，力争全面、易懂。对于每一个模型，其均衡条件的推导和经济含义都详细说明。在对新凯恩斯模型的介绍中，更详细介绍了黏性价格的来龙去脉，均衡条件的推导，并对3种不同的均衡做了详细界定，有助于初学者深入了解和全面把握建模和分析的基本逻辑。最后以一个含有价格黏性、工资黏性、消费习惯等要素的中等规模DSGE模型结束本章的分析。

接下来的4章内容为进阶篇。

第5章介绍了金融加速器机制的基本理论。本章内容分为8节，详细介绍了金融加速器模型中各行为主体在局部均衡和一般均衡下的决策。此处DSGE模型对文献中经典的模型做了简单的拓展，即在银行均衡条件中引入了成本溢价参数。当该成本溢价参数为零时，即为经典问题。

第7章是对部分经典文献和几个常见的建模问题做了解析，但很遗憾的是还不全面。本章分为3节内容。第1节首先对Galí(2008)介绍的小型开放模型进行了详细剖析和说明，并在此基础上加入了以福利损失函数为度量基准的最优货币政策，探析了相机抉择、规则承诺、含有成本推动型冲击等情况下最优货币政策问题。此部分内容算是对Galí(2008)中的大部分章节的内容做了概括性的解析。第2节介绍了文献中著名的零利率下限(ZLB)问题，以一个新凯恩斯模型为例，介绍了何为利率钉住。最后一节则着重分析了何为拉姆齐(Ramsey)最优货币政策。

第8章则聚焦分析了DSGE分析框架下微观福利的度量方法问题。首先介绍了条件福利水平和非条件福利水平的定义及在Dynare中的实现，然后介绍了消费补偿变化的方法作为政策福利度量和排序的依据，再进一步介绍了损失函数法在Dynare中的实现和数量刻画方法。

此外，各章节在写作过程中，尽量做到相对独立，也就是说读者可以从任一章开始阅读，涉及其他章节的知识点都做了详尽的标明，方便读者快速翻阅查找，获得帮助。然而，当本书完稿时，本人仍然在诚惶诚恐地阅读大量的中英文文献，特别是最新的工作论文和期刊论文，以进一步完善和修改。记忆最为深刻的当数第0章的写作，可谓数易其稿而不定。2016年以来，对DSGE模型批评和质疑的声音此起彼伏，以重量级的经济学家、现任世界银行行长Paul Romer在2016年年初发表的对DSGE模型的尖锐批评

为开端，2017 年以来，多位知名经济学家也陆续发声，包括 Jordi Galí、Oliver Blanchard 和著名经济学家、诺贝尔奖获得者 Joseph E. Stiglitz 等学者。欧洲经济政策研究中心 (CEPR) 还专门出版了一本电子书，刊发了 9 篇有关 DSGE 模型讨论的文章。因此，本书写作过程也紧跟了最新的文献。这里要特别感谢国内著名的微信群“货值 DSGE 研讨会”，从中我汲取了不少营养，得到了很多宝贵的资料。群里的很多大咖一直是我学习的榜样！

然而鉴于本人时间和精力所限，本书还存在诸多不足：第一，对于开放经济的建模理论涉及还较少；第二，对于异质性 (HANK) 模型的介绍没有涉及；第三，对 DSGE-VAR 相关的介绍仍然较为有限；第四，对于 MS-DSGE (Markov Switching) 的内容没有涉及；第五，缺乏相关的练习题；等等。在后续再版中，我将努力克服这些短板，力争为读者呈现出更多、更丰富的内容。

本书虽然定位为入门级学习资料，但读者群并不仅限于学生，如高年级本科生、研究生。对于那些从事宏观经济研究的学者，特别是对 DSGE 建模理论不甚熟悉的学者，以及政府、企业等相关研究机构的研究人员，本书同样适用。

“魔鬼总隐藏在细节里 (The Devil Is In Details)。”在初稿完成后，出版社和我一起先后对排版后的书稿进行 4 次校对，耗时之巨，力求失误最小。尽管我已尽最大努力减少本书每一页中的输入错误包括文字、公式和其他错误，但也难免会出现遗漏和差错。此外，由于作者水平有限，难免会出现技术性甚至系统性偏差和错误，因此我非常欢迎读者对本书提出建议和批评，以方便再版时及时更正。恳请发送邮件至：dynare@foxmail.com，我会力争在最短的时间内回复每一封来信，做到不遗漏。

本书的出版要感谢上海海关学院科研创新团队的支持 (No. 2313113)。没有团队成员和学校的支持，本书的出版估计还要一再被推迟。

在本书写作前的准备期，作者还受上海财经大学的资助赴美国圣母大学经济系访问一年，得到了 Nelson C. Mark、Eric Sims 和 Timothy S. Fuerst 教授启发和指导，并同时受益于宾夕法尼亚大学 Jesús Fernández-Villaverde 教授、西北大学 Lawrence J. Christiano 教授、芝加哥大学 John H. Cochrane 教授的讲座或讲课的启发。他们的指导和启发使我终身受益，在此一并表示衷心的感谢，当然文责自负！

本书的出版还得到了经管之家论坛的鼎力支持。感谢经管之家论坛给了我施展技艺的平台，同时也感谢给予 DSGE 视频录制帮助的几位工作人员，特别是小杨和小曾！经管之家论坛中活跃着上千万全国，甚至世界各地的经济学及其他相关学科的爱好者和工作者们，论坛中有非常丰富和宝贵的学习资料，为我的学习提供了不竭的源泉。

感谢清华大学出版社刘洋主任的鼎力支持，否则这本书也不会这么快面世。刘主任

热情的解答、专业的服务精神让人敬佩！

最后还要感谢我的家人。没有父亲、姐姐和妻子的理解和帮助，就不可能有本书的出版。每当夜深人静，伏案疾书之后，都意味着我第二天迟起，父亲、妻子总是能把家里照顾得很好，让我甚是感动！在我父亲有事不能照顾时，岳父母也会及时帮忙，解决我的后顾之忧！当然，我还要感谢小家伙 Cherry，她的降临成为我学习、工作与进步的不竭动力之源。

路漫漫其修远兮，吾将上下而求索！



2018年10月

@ 上海浦东花木





目 录

第 1 篇 初 级 篇

0 DSGE 模型简介	002
0.1 写作背景	002
0.2 DSGE 分析框架的发展	005
0.2.1 “三方方程”新凯恩斯模型	005
0.2.2 选择DSGE的原因	007
0.2.3 DSGE分析框架的表扬与批评	008
0.2.4 DSGE与VAR	018
0.2.5 DSGE与央行	019
0.3 DSGE 模型两种典型构建一瞥	022
0.3.1 CMR框架—金融加速器框架	023
0.3.2 小型开放经济模型的架构(产品市场)	024
0.4 宏观经济模型数据库 MMB	025
0.4.1 MMB源文件构成	026
0.4.2 MMB使用方法	026
参考文献	029
1 DSGE 模型求解逻辑	033
1.1 DSGE 一阶求解	033
1.1.1 一阶求解逻辑	033
1.1.2 线性化与对数线性化	036
1.1.3 B&K方法	041
1.1.4 Schur方法	050

1.1.5	待定系数法	057
1.1.6	线性模型的状态空间表示	060
1.1.7	脉冲响应和随机模拟	062
1.2	DSGE 高阶求解：Dynare 的求解逻辑	070
1.2.1	基于扰动项的泰勒近似方法	070
1.2.2	确定性等价和维数诅咒	078
1.3	DSGE 模型求解其他相关问题	086
1.3.1	确定性稳态值及其计算示例	086
1.3.2	外生冲击持续参数和标准差的校准	092
1.3.3	随机差分方程及其求解简析	095
1.3.4	HP滤波的基本逻辑	102
	参考文献	107
2	初识 Dynare	109
2.1	安装 Dynare	109
2.1.1	安装环境	110
2.1.2	下载安装	110
2.2	配置 Dynare	112
2.2.1	单次配置	112
2.2.2	永久配置	113
2.3	执行和编辑 Dynare 文件	114
2.3.1	执行Mod文件	114
2.3.2	编辑Mod文件	115
2.4	Dynare 多版本管理	116
2.5	获取 DYNARE 帮助	117
2.5.1	官方帮助文档	117
2.5.2	官方帮助论坛	118
2.5.3	其他在线方式	118
3	Dynare 基本应用	120
3.1	DSGE 模型：一个简单的例子	120

3.2	Dynare 内生变量的分类和书写规范	121
3.2.1	Dynare内生变量的分类	121
3.2.2	Dynare内生变量的书写规范	122
3.2.3	Dynare内生变量的排序	124
3.3	Dynare 文件基本结构	125
3.3.1	前导部分	125
3.3.2	模型部分	126
3.3.3	稳态或初值部分与外生冲击	127
3.3.4	计算部分	128
3.4	内生变量的表达形式: level or log-level	130
3.5	Dynare 文件的预编译和运行原理	133
3.5.1	Dynare文件的预编译与运行原理	134
3.5.2	表征模型的Matlab文件	136
3.6	Dynare 的解表示	137
3.6.1	一阶解表示	137
3.6.2	二阶解表示	138
3.7	求解结果分析和调用	140
3.7.1	屏幕输出结果	141
3.7.2	存储结果	142
3.8	确定性求解和模拟: simul	146
3.8.1	初始、终止条件	146
3.8.2	稳态求解命令: steady	148
3.8.3	确定性模拟	152
3.9	随机模拟分析: stoch_simul	163
3.9.1	随机模拟选项简介	163
3.9.2	随机模拟示例	164
3.10	参数估计简介	166
3.10.1	极大似然估计与贝叶斯估计的基本逻辑	167
3.10.2	马尔可夫链——蒙特卡洛(MCMC)方法	169
3.10.3	Dynare参数估计和一个例子	178
	参考文献	196

4	RBC 模型和 NK 模型	198
4.1	RBC 模型及其拓展	199
4.1.1	RBC模型与新古典增长模型	199
4.1.2	RBC模型的拓展	217
4.1.3	税收设定和Laffer曲线	248
4.2	新凯恩斯 (NK) 模型	255
4.2.1	家庭	255
4.2.2	黏性价格设定与价格离散核(Price Dispersion)	257
4.2.3	弹性价格均衡	267
4.2.4	有效均衡、弹性价格均衡和实际均衡	268
4.2.5	货币非中性分析	277
4.2.6	价格型和数量型规则的比较	285
4.2.7	“三方程”新凯恩斯模型	291
4.3	中等规模 DSGE 模型	303
4.3.1	模型	303
4.3.2	均衡	312
4.3.3	IRF分析	318
4.3.4	黏性设定对比分析	321
	参考文献	324

第 2 篇 进 阶 篇

5	金融加速器机制及其 Dynare 实现	328
5.1	金融加速器机制的背景	329
5.2	对数正态分布的基本概念	331
5.2.1	对数正态分布的定义	332
5.2.2	两个函数： Γ 和 G	333
5.2.3	简单的编程尝试	336
5.3	企业家的标准债务合约与杠杆	338

5.3.1	标准的债务合约	338
5.3.2	投资杠杆	339
5.3.3	异质性冲击的临界值	339
5.4	企业家期望净回报和银行均衡条件	340
5.4.1	企业家期望净回报	340
5.4.2	银行均衡条件	343
5.5	最优合约	345
5.6	模型均衡计算示例	348
5.6.1	模型均衡条件与基本参数	349
5.6.2	基于违约概率校准的局部均衡解	350
5.6.3	基于风险参数校准的局部均衡解	353
5.7	风险参数、风险溢价与清算成本变化的影响	353
5.7.1	风险参数	354
5.7.2	风险溢价	356
5.7.3	清算成本参数	357
5.8	金融加速器与随机波动模型示例	358
5.8.1	模型设定	359
5.8.2	Dynare实现及模型结果分析	363
	参考文献	373
6	Dynare 进阶应用	375
6.1	Dynare 模型文件的循环执行	375
6.1.1	循环执行的逻辑	375
6.1.2	一个简单示例	376
6.1.3	时间效率	379
6.2	脉冲响应函数和自定义编程	379
6.2.1	条件和无条件脉冲响应函数	380
6.2.2	自定义脉冲响应函数	383
6.3	二阶随机模拟中的一些问题	386
6.3.1	朴素模拟(Naïve Simulation)及其局限性	386
6.3.2	剪枝算法(Pruning)	387

6.3.3	二阶近似的伪稳态值	390
6.4	常见的 Dynare 运行错误	392
6.4.1	Dynare 错误分类	392
6.4.2	常见错误示例	393
6.4.3	使用调试 Debug 模式	395
6.5	Dynare 宏命令编程示例	396
6.5.1	模型	397
6.5.2	两国模型 Dynare 示例	398
6.5.3	多国模型 Dynare 宏语言示例	399
6.5.4	@#include 命令示例	402
	参考文献	403
7	部分经典文献解析和建模问题	404
7.1	货币政策和新开放宏观模型	404
7.1.1	货币政策和新开放宏观模型	405
7.1.2	基于福利损失的最优货币政策	415
7.1.3	技术附录	428
7.2	利率钉住与零利率下限模型	430
7.2.1	零利率下限和利率钉住的定义	430
7.2.2	一个新凯恩斯(NK)模型	432
7.2.3	Dynare 实现和 IRF 分析	434
7.3	拉姆齐 (Ramsey) 最优货币政策	437
7.3.1	拉姆齐最优货币政策的定义	437
7.3.2	黏性价格模型及其最优	438
7.3.3	Andy Levin's Code	465
	参考文献	467
8	DSGE 模型下微观福利度量方法	469
8.1	条件福利和非条件福利的定义	469
8.1.1	简单的 RBC 模型	469
8.1.2	条件福利水平	470

8.1.3 无条件福利水平	471
8.1.4 无条件消费补偿变化的其他度量	471
8.2 消费补偿变化示例	472
8.2.1 可加可分的效用函数	472
8.2.2 KPR 效用函数	474
8.2.3 Dynare代码实现	475
8.3 损失函数法	479
8.3.1 损失函数的推导	479
8.3.2 Dynare代码实现和结果解析	481
参考文献	483



第 1 篇

初 级 篇

0

DSGE 模型简介

本章对 DSGE 分析框架的发展做了简单的介绍，分为 4 节内容。第一节从本书的写作背景出发，简要地回顾了现有的图书文献，并引出本书写作的动机。第二节对 DSGE 分析框架的发展，从 5 个方面进行了阐述。第三节简要介绍了两种文献中常见的 DSGE 模型的构建，以给读者框架性概览。最后介绍了宏观经济模型数据库 MMB 的使用方法，以为读者提供导引，从而获取更多的学习素材和帮助。

0.1 写作背景

近年来，有关动态随机一般均衡 DSGE(Dynamic Stochastic General Equilibrium Framework/Model, DSGE Framework/Model) 分析框架(模型)的讨论越来越多。这种讨论不仅表现在日益增多的学术论文上，而且也表现在 DSGE 相关专著的层出不穷上。学术期刊上的宏观经济研究论文中采取 DSGE 分析框架的，可谓不胜枚举。对于图书和专著，不论是中文还是英文，都越来越丰富。

在国内，以刘斌老师为代表的研究者出版了多本 DSGE 专著，如由中国金融出版社出版的《动态随机一般均衡模型及其应用》一书到目前为止已经出版三版了。刘斌老师可谓笔耕不辍，紧跟文献研究最新发展动态，开国内 DSGE 理论研究之先河，令人钦佩。此书理论性研究较强，功底深厚，从参考文献的数量就能略知一二。但若通篇能与实践结合得更加紧密(如 DSGE 模型求解一章)，如能提供 Dynare 编程实践和数据、操作指引，则体验将会更佳。此外，刘斌老师还翻译了一系列佳作，如《动态随机一般均衡模型入门》^①《动态一般均衡建模——计算方法与应用》^②等都值得一读。

① Torres J L. *Introduction to Dynamic Macroeconomic General Equilibrium Models*[M]. Vernon Art and Science, 2015.

② Heer B, Maussner A. *Dynamic General Equilibrium Modeling-Computational Methods and Applications*(2nd edition)[M]. Springer, 2009.

国外入门级的 DSGE 专著要数 Galí(2008, 2015)、McCandless(2008) 和 Torres(2015)。Galí(2008, 2015) 和 Torres(2015) 都是非常经典的专著，值得精读。两者具有非常类似的写作风格——简洁而不简单，都从一个最简单的、经典的 DSGE 模型（新凯恩斯模型）开始，然后逐步融入各种建模要素，层层递进。Galí(2008,2015) 遵循的阐述路径是完全弹性、完全竞争的 DSGE 模型，到含有垄断竞争和价格黏性的 DSGE 模型，再到含有价格黏性与工资黏性的 DSGE 模型，最后拓展到小型开放经济模型；每一章节都有相应的文献综述、技术推导附录和习题练习，是入门级不可多得的宝贵资料；而且在每一章中都加入了对最优货币政策的讨论，可谓循循善诱。Torres(2015) 遵循的阐述路径是从最简单的 DSGE 模型开始，逐步加入消费习惯形成 (Consumption Habit Formation)、投资调整成本 (Investment Adjustment Cost)、投资专有技术 (Investment-Specific Technology)、技术改变 (Technological Change)、税收、公共资本 (Public Capital)、家庭生产 (Household Production)、非李嘉图特性的家庭 (Non-ricardian Agents)、垄断竞争等要素，可谓引人入胜，一气呵成。Galí(2008, 2015) 和 Torres(2015) 的区别在于：前者没有提供有关 DSGE 模型求解和估计的任何内容；而后者则在每一章后面都提供了相关的 Dynare 模型文件，以供读者参考。但很遗憾的是，Torres(2015) 并未对模型文件和 Dynare 进行深入了解。Galí(2008, 2015) 虽然没有提供 Dynare 模型文件，但这并不能成为拒绝阅读这本书的理由。相反，更应该通过学习 Dynare 软件，动手编程来实现每一章的模型求解和分析，从而能够获取更深层次的理解。

McCandless(2008) 这本书更专注于 RBC 理论及线性化方法的运用，但后半部分逐步深入凯恩斯模型的核心，引入黏性价格和黏性工资设定，最后介绍了小型开放模型。虽然通篇都采取线性化的求解方式，但其更加注重模型本身设定及经济学解释，而且提供了线性化求解的 Matlab 源代码帮助理解，是入门级专著中不可多得的学习资料。

除上述几本入门级专著外，还有一些其他的入门或中级专著，如 Walsh(2010)、Wickens(2008)、Heer & Maussner(2009)、DeJong & Dave(2007)。Walsh(2010) 这本《货币理论与政策》本身专注于货币在经济中的作用，将货币或货币政策融入每一章的分析中，内容翔实全面，并附有详细的技术推导细节和相应的习题。其中不少的章节涉及 DSGE 模型与货币，特别是新凯恩斯货币经济学、货币与开放经济部分都值得认真研读。如果该书能提供相关的 Dynare 模型文件或 Matlab 代码，帮助读者加深理解，则会锦上添花。Wickens(2008) 这本书也是一本优秀的教材，整本书都在 DSGE 分析框架下阐述宏观经济理论，覆盖范围较为广泛，包括对财政政策、货币政策、资产定价、金融市场、名义汇率、开放经济和货币经济等主题的讨论。值得一提的是，Wickens(2008) 提供了丰富的练

习题与参考答案，甚至给出了 Dynare 模型文件，值得仔细阅读。Heer & Maussner(2009) 更加关注于对求解技术 (如扰动算法、投影算法) 和算法本身的实现上，算得上是一本中级教材，但其缺乏对宏观经济模型、均衡条件与结果经济含义的解释与说明。Heer & Maussner(2009) 后半部分提供了有关异质性模型的介绍，比较难得。整本书虽然提供了相关算法的代码，但都以 Fortran 和 Gauss 为主。不可否认 Fortran 和 Gauss 仍然是高效率的结构化程序计算语言，在科学计算方面仍然占有重要的地位，而且不少初级 DSGE 模型能够通过 Gauss 程序求解，但在稍微复杂的 DSGE 求解、估计与后续分析方面，Gauss 程序，显得力不从心。此外，没有将 Fortran 和 Gauss 源程序转换为 Matlab 程序使此书的阅读与受众范围受到了一定的限制。DeJong & Dave(2007) 这本书虽然取名《结构宏观计量经济学》，但其大部分内容是关于 DSGE 模型及其求解，因此是名副其实的 DSGE 专著。从行文风格来看，和 Heer & Maussner(2009) 较为类似，偏重于技术本身的介绍，缺乏相关编程实践。因此，对初学者来说有些枯燥，很难吸收其艰深的理论，当然如果能够坚持啃下来，肯定会有不少收获。

在开放宏观经济学领域，Uribe & Schmitt-Grohé (2017) 的《开放经济宏观经济学》巨作可谓非常权威，两位作者也在宏观经济领域非常活跃，贡献不小。该书不仅覆盖了开放经济领域大部分主题，而且易读性也非常强，提供了数据、PPT 和相关的 Matlab 代码文件，其问题分析的框架也大都是一般均衡分析框架，因此本书是开放宏观领域不可多得的专著。但其目标范围并非入门人群，而是具有一定开放宏观经济学基础的研究人员。

越来越多有关 DSGE 模型的探讨还表现在如何求解和估计 DSGE 模型上。以 Dynare 为代表的软件包的开发与成熟，标志着 DSGE 模型的求解、估计不再是少数人的游戏，而成为所有宏观经济学习者 (包括初学者) 和专业研究者的日常工具。也就是说，Dynare 问世使得 DSGE 模型的求解和估计不再具有明显的技术门槛，只要通过简单的编程和指令就能完成以前复杂的求解与估计工作，使得遥不可及的任务变得唾手可得。

然而，目前尚没有一本将 DSGE 理论建模和 DSGE 编程实践 (求解、估计与结果分析) 有机融合的入门级专著，从而为刚刚入门学习宏观经济学的学生或者从事宏观经济研究但尚不熟悉 DSGE 分析框架的学者提供一个快速的导引。即，目前仍缺乏这样一本 DSGE 专著，它既能够使初学者对 DSGE 建模及其背后的经济学理论、数学推导等有一个快速的了解和掌握，又能够对构建的 DSGE 模型进行编程求解和估计提供帮助。随着时间的推移和个人经验的逐步积累，越来越觉得弥补这一缺陷变得相当紧迫和必要。

本书尽最大努力来尝试弥补这一缺陷。在行文中除介绍 DSGE 建模本身的理论外，还穿插介绍诸多 Dynare 软件及其功能，并对关键实现代码进行详细解读，不惜花费 1/3

的篇幅力求全面，尽量做到建模理论与编程实践的有机融合，使得读者知其然，更知其所以然。

0.2 DSGE 分析框架的发展

DSGE 代表动态随机一般均衡，可认其为一种宏观经济学的分析框架。基于此分析框架所建立的模型（即具备动态和随机的一般均衡模型）都可称之为 DSGE 模型^①。DSGE 模型最早可追溯到由 Kydland & Prescott(1982, *Econometrica*) 提出的真实经济周期模型 (Real Business Cycle, RBC)。Gali(2017) 将 RBC 模型称为 DSGE 模型的乌托邦 (Utopian)，而 Blanchard(2017)^② 则将其称为没有包含任何扭曲和摩擦设定的最早版本 DSGE 模型 (The Earliest DSGE Model)，即完全弹性价格和完全竞争设定，并考虑外生技术冲击对经济增长的重要作用。因此 RBC 模型作为“前辈”，对 DSGE 模型的产生、发展起到奠基性作用。而后，在 RBC 模型的基础之上加入了各种经济行为主体（厂商、金融中介、政府部门等）、各种市场（产品市场、劳动力市场和金融市场等）以及各种摩擦（如黏性价格、黏性工资和黏性信息等）、扭曲（如垄断竞争、信息不对称等）和各种外生冲击^③，形成了新凯恩斯模型 (New Keynesian) 模型以及 DSGE 模型。新凯恩斯模型可看作初级版的 DSGE 模型，是雏形；而一般情况下，更复杂的 DSGE 模型可看作是中、大规模 (Medium and Large-Scale) 的新凯恩斯模型，引入各种扭曲、摩擦设定以及外生冲击，以提升数据拟合能力和预测能力。

0.2.1 “三方程”新凯恩斯模型

很长一段时间内，新凯恩斯模型占据宏观经济模型和宏观经济分析的主流地位。至今，DSGE 模型仍然没有脱离新凯恩斯模型的建模思路。带有扭曲和摩擦（如黏性价格、

① 在以下的叙述中，并未严格区分 DSGE 分析框架和 DSGE 模型，有时将其混用。

② 法国著名经济学家 Olivier Blanchard 在 2016 年撰文 *Do DSGE models have a future?* 首先发表在皮尔森经济研究所 (Peterson Institute for International Economics) 网站上 (<https://piie.com>)，后被收录至电子书 *DSGE Models in the Conduct of Policy: Use as intended*, edited by Refet S. Gürkaynak and Cédric Tille, CEPR Press, London, April 2017. 此外，牛津大学出版社在 2018 年第 34 卷第 1 期和第 2 期的《牛津经济政策评论》(*Oxford Review of Economics Policy*) 中专门刊登了宏观经济模型相关的论文，特别是 DSGE 模型，值得一读。

③ 文献中，扭曲一词对应的英文为 *distortion*。摩擦一词对应的英文为 *friction*。其模型设定对应着各种黏性 (*stickiness*) 或者称为名义刚性 (*nominal rigidity*) 或名义摩擦 (*nominal friction*)。本书未严格区分这两种设定在行文中的表达。无论是扭曲还是摩擦设定，在模型中都被称为无效率设定 (*source of inefficiency*) 或不完美设定 (*nominal or real imperfection*)。

垄断竞争) 设定的新凯恩斯模型, 可最终用 3 个最关键的方程来表示, 这恰是新凯恩斯模型的核心所在^①, 其实也是 DSGE 模型的核心所在。这 3 个 (对数线性化) 方程分别如下所示。

第一个是总需求方程 (也称之为 New Keynesian IS Curve)^②, 即 Euler 方程:

$$X_t = E_t X_{t+1} - \frac{1}{\sigma} (i_t - E_t \pi_{t+1} - r_t^f) \quad (0.2.1)$$

其中, X_t 为产出缺口; i_t 为名义利率; π_t 为通胀; r_t^f 为自然利率, 假设其服从 AR(1) 过程^③, $\sigma > 0$ 为参数。总需求方程描述了产出缺口的决定关系。

第二个是价格调整方程, 即新凯恩斯菲利普斯曲线 (New Keynesian Philips Curve, NKPC) 方程:

$$\pi_t = \kappa X_t + \beta E_t \pi_{t+1} \quad (0.2.2)$$

其中, π_t 为通胀率; κ 为模型结构参数的函数; $\beta > 0$ 为贴现因子。NKPC 曲线描述了通胀的决定关系。

第三个是货币政策规则, 比如 Taylor 规则:

$$i_t = \rho_i i_{t-1} + (1 - \rho_i) (\phi_\pi \pi_t + \phi_x X_t) + \epsilon_{it} \quad (0.2.3)$$

其中, ρ_i 、 ϕ_π 、 $\phi_x > 0$ 为参数。

Blanchard(2017) 在评价 DSGE 模型时, 针对 Euler 方程和 NKPC 方程提出了严厉的批评: 对于 Euler 方程, 其最原始的含义是刻画消费和实际利率之间的关系。即消费与实际利率呈反向变动关系, 但这与许多经验研究的结论相左, 也就是说消费和实际利率并不负相关, 不过其并未给出具体的例子。对于 NKPC 方程, 其虽然加入了通胀预期, 但式 (0.2.2) 完全没有考虑通胀的惯性 (Inertia), 即缺少通胀的滞后项^④。此外, Blanchard(2017) 还指出对于简单的 Taylor 规则, 已经不能满足研究和决策的需要, 这是因为在当下非常规货币政策工具的使用和非常规宽松环境下 (如量化宽松、前瞻指引^⑤ 与零利率下限等), 某一单个的政策规则已经不能满足研究和决策的需要。

① 具体推导过程可参考本书的“4.2.7 ‘三方方程’新凯恩斯模型”。

② 在 IS-LM 框架下, IS 代表投资 (Investment) 和储蓄 (Savings), IS 曲线即投资等于储蓄。但是在简单的新凯恩斯模型中, 并没有投资变量, 之所以将其称为 IS 曲线, 其基本的考量在于当期的消费支出需求和实际利率的反向关系。之所以称其为新凯恩斯 IS 曲线, 是因为当期的消费支出需求 (产出) 不仅依赖于传统的变量, 即实际利率, 也依赖于消费支出 (产出) 的期望值。

③ 关于自然利率, 本节稍后会有介绍, 也可参考本书“4.2.7 ‘三方方程’新凯恩斯模型”一节中相关的推导过程。

④ 此问题可以通过简单的建模设定加以消除, 比如 indexation 技术。

⑤ 本节稍后会进一步解释。

从本质上讲, DSGE 模型引入的大部分模型设定, 归根结底是对上述 3 个方程从不同的角度的拓展和修改 (Blanchard, 2017), 以提升其现实解释和经济数据预测能力。

0.2.2 选择DSGE的原因

不可否认, DSGE 已经成为现代宏观经济学的主流研究框架, 不仅被学术界 (包括顶级学术期刊在内) 竞相推崇, 也被广泛应用于政府经济与金融决策机构的研究中, 如 OECD 国家央行和新兴经济体央行 (如中国人民银行) 等。那为什么 DSGE 能够成为宏观经济学主流的分析框架呢? Galí(2017) 认为这是由宏观经济现象本身的属性和 DSGE 分析框架的定义决定的。

首先, 宏观经济现象是动态的 (Dynamic)。多种行为主体相互联系, 在各种市场 (产品市场、劳动力市场和金融市场) 中做出跨期决策行为, 因此静态模型 (无论是确定性亦或局部均衡模型) 都无法很好地描述这些经济现象, 比如跨期消费决策、投资、利率、汇率和财政赤字等都呈现出复杂的动态性。其次, 宏观经济充满了各种不确定性 (Uncertainty)。经济本身或行为主体本身都面临着各种内部或外部不确定性冲击, 只有随机模型 (Stochastic) 才能很好地对这些不确定性进行建模。最后, 宏观经济现象是所有经济行为主体相互联系、共同作用的结果。局部均衡无法描述所有主体动态作用的结果, 因此需要在一般均衡框架下进行分析。

DSGE 分析框架在研究与实施政策组合方面具有天然的优势 (这些政策组合包括货币、财政政策以及结构性改革政策, 如 TPP 协定等)。Ghironi(2017) 认为当下实施全面的政策组合, 已成为决策层的最新呼吁, 而研究不同政策的传导渠道及其之间的相互作用与影响非常重要, 同时也成为宏观经济研究的一个挑战。在分析框架的选择上, Ghironi(2017) 认为 DSGE 分析框架能够很好应对这一挑战, 成为不可多得的研究框架。DSGE 分析框架的动态属性能够帮助决策者理解不同政策在短期和长期之间的效果差异, 以及不同政策之间的互补或替代关系。DSGE 分析框架的随机属性使其具备对不确定性的现象进行微观建模的基本能力, 即各个行为主体的决策都基于不完全信息。最后, DSGE 分析框架的一般均衡要求经济中的“价”(如利率、通胀等)与“量”(如产出、消费、投资等)由模型自身的约束与一阶条件共同确定, 而无须施加影响“价”与“量”的先验假设。

此外, 宏观经济学的大部分问题都可以纳入 DSGE 分析框架中。2017 年 5 月, 政治经济学杂志 (*JPE*) 主编 Harald Uhlig 在上海财经大学举办的第三届思想中国论坛上, 提出了当前五大宏观经济学前沿问题: 科技进步与经济增长、人口老龄化与储蓄过剩、房价

与金融市场、宏观经济学下的医疗保健与环境问题，并且对每一个问题都给出若干子问题。Harald Uhlig 认为这些问题的研究均可纳入 DSGE 分析与研究框架^①，可见 DSGE 作为一个分析框架，其应用范围之广。

DSGE 模型是结构性模型 (Structural)，这也意味着数据的拟合要依赖于经济背后的基本驱动力，即外生冲击。然而这些外生冲击往往是不可观测的 (Unobservable)，需要使用诸如贝叶斯估计的方法将其从数据中“提取”出来，并同时估计出某些结构参数的值。在有了“估计”的参数值和外生冲击，从而可以解释过去，并预测未来^②。除了外生冲击外，模型中还有很多其他不可观测的变量，而这些变量恰恰是决策者们非常感兴趣的：比如产出缺口、自然利率水平等，而这也恰恰是经典的结构模型所做不到的。

0.2.3 DSGE 分析框架的表扬与批评

英国著名统计学家 George Box 有一个知名的论断：所有的模型都是错的，但有些是有用的^③。虽然 Box 身为统计学家，所提的论断针对统计学模型，但该论断在经济学中同样适用。没有哪一个简单的经济模型能够完全精确地拟合复杂的经济现象。但经过精心设计的某些模型却能提供对现实经济现象的高度近似，因而值得关注和研究。Prescott(2016) 也提出类似的观点^④。DSGE 模型也不例外，其对高度复杂的经济现象做出简化 (Simplification)，并最终使用一系列均衡条件 (包括最优化一阶条件、市场出清条件等) 和少数变量 (内生和外生变量，相对于宏观统计变量而言) 来表征经济整体。客观地说，DSGE 模型并非完美无瑕，也不是一无是处。下面从 DSGE 分析框架受到的表扬与批评两个方面来简要阐述。

① 此外 Harald Uhlig 还力荐其作为主编之一的《宏观经济学手册》(*Handbook of Macroeconomics*, 2016) 作为这些问题研究的起点和参考。这本《宏观经济学手册》多达 2 693 页，涵盖了现代宏观经济学的主要研究前沿，各章内容均为该领域的国际知名学者执笔，值得一读。其中相当一部分的研究与 DSGE 模型相关，包括了第 9 章 DSGE 模型的求解与估计方法、第 22 章的 RBC 理论与方法和第 26 章新古典经济模型，等等。

② 此为文献中所谓的讲故事 (story-telling)。

③ Box, G. E. P. (1979), *Robustness in the strategy of scientific model building*, in Launer, R. L.; Wilkinson, G. N., *Robustness in Statistics*, Academic Press, pp. 201–236. 在该篇文章中，有一节以 “All models are wrong but some are useful” 为标题展开：... Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations... For such a model there is no need to ask the question “Is the model true?” If “truth” is to be the “whole truth” the answer must be “No”. The only question of interest is “Is the model illuminating and useful?”

④ (In Abstract, 摘要中)... Reality is complex, and any model economy used is necessarily an abstraction and therefore false. This does not mean, however, that model economies are not useful in drawing scientific inference.

1. 表扬

从学术界和金融机构决策研究中的广泛采用中，在一定程度上已经说明了 DSGE 模型有其合理性和有用性。但相比 DSGE 模型受到的批评和质疑，其受到的表扬似乎相当有限。

Blanchard(2017) 认为 DSGE 模型的 3 个基本特征无疑是非常必要的，也是相当成功的：第一，微观基础 (Micro-founded)；第二，引入了各种摩擦和扭曲 (Distortions)；第三，多方程联立估计，而非逐个估计 (如贝叶斯估计)；Del Negro & Giannoni(2017) 指出相比简化式模型 (Reduced Form)^①，DSGE 分析框架的一个重要优点在于其不受卢卡斯批判 (Lucas Critique) 的影响。这是因为模型的参数是结构性 (Structural)，其不受政策变化的影响^②。基于此，Del Negro & Giannoni(2017) 认为，DSGE 分析框架为央行估计不同的政策反应函数提供了天然的分析框架，这也是简化式模型所不能做到的。Justiniano, Primiceri & Tambalotti(2017) 指出，DSGE 分析框架的一个重要贡献在于其帮助识别经济周期背后的驱动力 (Sources of Business Cycles)。从 Kydland & Prescott(1982) 提出的 RBC 模型开始，技术进步作为经济增长和波动的主要驱动力，到 Justiniano, Primiceri & Tambalotti(2010, *JME*; 2011, *RED*) 提出投资边际效率冲击能够解释大部分经济波动，再到 Christiano, Motto & Rostagno(2014, *AER*) 指出风险冲击超过投资边际效率冲击能够解释大部分经济波动^③[当引入金融摩擦并包含金融变量 (股票市值、贷款等) 作为参数估计的输入序列时，方差分解的结果证实了该结论]，见表 0.1。因此，随着研究的不断深入，DSGE 分析框架会带来更多更深入的认识。

表 0.1 经济周期背后的主要驱动力

冲击类型	解 释	文 献
技术冲击	全要素生产率冲击 (TFP)	Kydland & Prescott(1982, <i>Econometrica</i>)
投资边际效率冲击	影响投资向资本存量转换的技术冲击	Greenwood, Hercowitz & Huffman (1988, <i>AER</i>)、Justiniano, Primiceri & Tambalotti(2010, <i>JME</i> ; 2011, <i>RED</i>)
风险冲击	异质不确定性风险冲击的标准差	Christiano, Motto & Rostagno(2014, <i>AER</i>)

数据来源：作者自行总结。

① 在本书的“4.1.1 RBC 模型与新古典增长模型”一节中会对简化式模型的含义做进一步介绍。

② 从理论角度说，Del Negro & Giannoni(2017) 的分析具有一定的道理。既然是结构性参数，就意味着其不会轻易发生变化，或者说不受政策影响，这也是其取名的本意。但在实际研究中，结构性参数的估计值会随着所用的宏观序列本身的不同以及序列长度的不同而发生变化，而宏观时间序列本身受到政策变化的影响。因此作者认为应对 Del Negro & Giannoni(2017) 的观点持有保留意见。

③ 关于风险冲击的更多讨论，可参考本书“5.8 金融加速器与随机波动模型示例”一节。

2. 批评

从 2016 年开始，对 DSGE 分析框架的质疑与批评可谓此起彼伏。归纳起来，批评和质疑可大致分为两类：一类是带有否定倾向的尖锐批评；以经济学巨擘 Paul Romer^①、Joseph Stiglitz 和经济学后起之秀 Anton Korinek 为代表；另一类是包容性批评，在指出缺点的同时给出方向性或策略性改进建议，以 Galí(2017)、Blanchard(2017) 等为代表。

DSGE 模型存在缺陷已经成为文献中不争的事实。对 DSGE 模型的批评最著名的要数 Romer(2016)，其对 DSGE 分析框架从两个方面进行了尖刻的批评。

第一，作为 DSGE 模型的“前辈”，RBC 模型将经济波动归因于虚拟 (Imaginary) 的冲击 (经典的 RBC 模型中仅为技术冲击)，而且这些冲击不受内生变量的影响，即外生 (Exogenous，不受经济行为人活动的影响，不受模型内生变量的影响，变化独立于模型本身)。Romer(2016) 进一步指出 DSGE 模型只不过是在此基础上添加更多的虚拟外生冲击而已。除个别例外，模型均假定大多数外生冲击无法直接观测和度量。Romer(2016) 认为这是极其荒谬的，并给出了一个具体的例子：在 Paul Volcker 执政美联储期间^②，实际联邦基金利率从 0 上升至 5% 附近并持续数年，这是货币政策有效调控实体经济最好的例证，而非 RBC 或 DSGE 模型所描述的——货币政策的作用相当有限，或多由虚拟的外生冲击所致。此外，Romer(2016) 指出 RBC 模型引以为豪的微观基础假设，却无法从微观基础上对负向 TFP 冲击 (Negative Technology or Phlogiston Shock) 给出合理解释，也无法提供任何相关的理论依据。而宏观经济学家对此的辩解无外乎引经据典，糊弄过关：一是 Friedman(1953) 的论断，即理论越卓越和显著，假设越脱离现实；二是 George Box 的名言，即所有的模型都是错的。

第二，识别 (Identification) 问题始终是 DSGE 模型无法绕过的“坎”。识别问题的本质是参数的真实值 (True Value) 无从知晓。从参数估计的技术上讲，引入多少虚拟外生冲击，就需要引入多少观测变量，而这使得识别问题更加恶化。Romer(2016) 以一个简单

① Paul M. Romer 因其在内生经济增长研究方面的卓越贡献获得 2018 年诺贝尔经济学奖。瑞典皇家科学院对他的评价是“for integrating technological innovations into long-run macroeconomic analysis”。

② 1979 年 8 月—1987 年 8 月，任美联储主席，服务两任美国总统：卡特总统和里根总统，其强有力的执政能力使得美国 20 世纪 70 年代的高通胀得以遏制，被称为 Volcker Deflation。

的线性联立方程模型为例子，生动地说明了识别问题^①：

$$x = Sx + c + \epsilon \quad (0.2.4)$$

其中， x 为 m 个可观测的内生变量组成的向量； S 为待估系数矩阵，对角线元素为零（即内生变量是其他变量的线性组合，而非自身）； c 为常数项（即截距项）； ϵ 为独立同分布的误差项。

可以发现 (0.2.4) 中有 m^2 个未知参数需要估计：系数矩阵 S 中有 $m \times (m-1)$ 个再加上截距项 c 中有 m 个。但系统只有 m 个方程，因此模型存在识别不足问题，需要添加额外约束才能识别。在 Smets & Wouters(2007, *AER*) 的线性化的 DSGE 模型中，模型共有 7 个可观测变量，因此有 49 个待估参数，7 个约束方程，理论上只能识别 7 个参数，因此 42 个参数需要额外增加约束条件才能识别。进一步，Romer(2016) 认为理性预期假设 (Lucas & Sargent, 1979) 使得识别问题更加严重，而非减轻。以劳动供给和工资水平之间的关系为例，Romer(2016) 指出劳动供给不仅受到当期工资的影响，而且也受期望工资水平的影响，因此模型中引入期望值有其合理性。具体说来，如果在模型中引入理性预期变量，如

$$x = Sx + B\mu(x) + c + \epsilon \quad (0.2.5)$$

其中， $\mu(x)$ 为 x 的期望值； B 为对应的待估系数矩阵。

事实上，引入系数矩阵 B ，额外增加了 m^2 个未知待估参数，此时所有待估参数实际上为 $2m^2$ 个，因此 Romer(2016) 认为理性预期的假设使得识别问题更加恶化。更为尴尬的是，即使能够识别，其识别结果无法区分截距项 c 和期望值 $B\mu(x)$ 。Romer(2016) 还认

① Romer 还在文章中指出了宏观经济学的真正麻烦所在：宏观经济学的真正麻烦并不在于宏观经济学家们坚持的理论和学说与事实相左，而是许多经济学家对已经存在的理论错误和荒谬视而不见，不去公开提出批判，挑战学术权威，而是选择接受和尊崇 (Deference)。Romer 同时以理论物理学中著名的弦理论 (String Theory) 为例，来类比宏观经济学发展的现状。弦理论作为粒子理论中的重要学说，认为自然界的基本构成单元不是电子、原子和夸克之类的点状粒子，而是很小很小的一维线状的“弦”组成。弦的不同振动和运动就产生出各种不同的基本粒子，能量与物质是可以转化的。然而很遗憾的是弦理论目前还停留于纯数学模型阶段 (Smolin, 2007, Chapter13: Smolin, L. (2007). *The Trouble With Physics: The Rise of String Theory, The Fall of a Science, and What Comes Next*, Houghton Mifflin Harcourt)。而 Romer 认为这恰是宏观经济学中的现状，仅仅关注于纯数学的理论模型，盲从学术权威的意志和设定路线，从而偏离追求事实的基本原则。Romer 特别点名批评了 3 位经济学权威：Robert Lucas, Ed. Prescott 和 Thomas Sargent。这 3 位学术权威都是经济学诺奖获得者，都为宏观经济学做出了巨大理论贡献。Romer 认为之所以选择接受和尊崇，不是因为信息不对称，也不是因为偏好问题，而是因为公开批评和挑战会带来代价，特别是那些仍以学术为其主要生涯的学者来说尤其如此。Romer 认为自己是一个理论践行者 (Practitioner, 现任世界银行行长)，而不再是一个学者 (Academic)，不再需要以发表学术论文来实现自己的目标，因此对他来说公开批评和挑战带来的代价较小。这也是为什么 Romer 发表这篇尖刻批评论文的原因之一。

为贝叶斯估计并没有解决识别问题，而是将识别问题隐藏得更深：通过选择适当的先验分布来得到想要的后验分布，也就是说可以根据想要的任何结果来倒推预先设定的先验分布。

很显然，Romer(2016) 的批评有其一定的合理性，但整个宏观经济学，包括 DSGE 分析框架并没有像 Romer(2016) 所说的那样遭受空前的麻烦。陈彦斌和陈惟 (2017) 对 Romer(2016) 的质疑与批评从 3 个方面进行了强有力的回应。第一，Romer(2016) 的批评仅仅是宏观经济学的的一个分支而已，而非全部，因此不能说明整个宏观经济学遭遇大麻烦。第二，Romer(2016) 对 DSGE 模型的现实解释力不足，DSGE 模型无法解决参数识别问题的批判标准过于严苛。陈彦斌和陈惟 (2017) 则认为理论滞后于现实是常态，不能奢望刚诞生不久的 DSGE 模型立马就有十足的现实解释能力，而且 DSGE 模型在不断从宏观经济事件中“总结经验教训”，提升模型的现实解释能力。

Stiglitz(2018) 的批判更是入木三分，耐人寻味。他特别强调 DSGE 模型对于政策设计的无用性及其可能带来的后果 (Consequences) 的来源：DSGE 模型关键的失败之处在于其微观基础。这种微观基础来自于简单的竞争性均衡模型，而这种模型早已被行为经济学 (Behavioural Economics)、博弈论 (Game Theory) 和信息经济学 (Economics of Information) 抛弃。这种建模方式的一个通病就是假设名义刚性 (如价格和工资刚性) 来解释失业和其他经济失衡现象。然后 Stiglitz(2018) 从金融危机的发生和应对的角度来尖锐地批判了 DSGE 模型的无能为力：其内生建模机制既无法预测到金融危机的发生，也无法提供任何建议。首先，DSGE 模型代表性家庭 (Representative Household) 的假设使得模型不具有任何分配效应 (Distribution Effect)^①，因此过于绝对。以边际消费倾向为例，收入位于前 1% 和后 80% 的人在边际消费倾向上，肯定存在巨大差异。更一般地，任何影响分配的因素一定影响总需求，因而具有重要的宏观经济影响。但代表性家庭的假设抹杀了这一切。此外，代表性家庭的假设使得 DSGE 模型无法描述宏观经济中一个重要现象：外部性 (单个个体行为的加总会有较强的宏观作用和影响，但代表性家庭从根本上排除了这一点)。其次，从冲击的角度讲，DSGE 模型从根本上排除了不稳定性冲击 (如系统性风险，Systemic Risk)，使得模型更关注于诸如通胀等因素是否影响经济安全，而忽略金融稳定这一重大问题，因此不可能预测金融危机的发生，更无法解释小冲击具有大作用现象及其机理。然后，DSGE 对金融部门建模不足。比如 DSGE 对于债务问题几乎没有建模，在标准的模型中均衡时总假设其为零 (Netted Out)。再比如对流动性约束 (信

^① 代表性家庭假设的新凯恩斯模型被简称为 RANK 模型 (Representative Agent New Keynesian)，因此与之对应的异质性家庭假设的新凯恩斯模型被简称为 HANK 模型 (Heterogeneous Agent New Keynesian)。

贷配给、股权配给现象)几乎视而不见。最后,DSGE模型总使得我们倾向于认为经济总在动态均衡路径上运行^①,并专注于分析跨期替代效应。Stiglitz(2018)认为这两者都不是短期或中期分析的核心所在,而且进一步的DSGE模型对于经济长期增长的分析几乎空白。

不可否认,Stiglitz(2018)的批评有其合理性,虽然带有否定性的倾向,但其仍给出了建设性的改进方向。Stiglitz(2018)最后给出了一个耳目一新的宏观经济分析模型——三期简单模型(Three-period Models),将“分配效应”“银行机构(Bank)”“信贷约束(Credit Constraint)”3个未在DSGE分析框架中出现的元素纳入该模型,用于分析各种政策对产出和就业的影响,值得关注。

Korinek(2017)对DSGE分析框架的批判同样尖刻。他指出,DSGE分析框架概念相关的限制性约束(Conceptual Restriction)、复杂的求解、模拟算法以及由此导致各种偏差(Bias)可能会阻碍宏观经济学的发展^②。Korinek(2017)总体认为DSGE分析框架在应该花费力气却没有花费足够的力气,在不应该花费力气的地方浪费了太多。比如DSGE分析框架应该在概念相关的限制性约束上下功夫,尽力减少不现实、甚至错误的约束和假设。再比如应减少复杂的数学逻辑,转而注重模型的经济内涵,这会更加有助于解决实际经济问题,从而更具有现实意义。

相比带有否定性倾向的“尖刻”批评,文献中更多的是包容性批评。Galí(2017)指出DSGE分析框架有两个令人不安的“紧箍咒”:第一,无限期生存的代表性家庭的基本假设(Infinitely-Lived Representative Household);第二,由外生冲击驱动的平稳性、线性均衡。从目前来看,假设家庭具有无限生存期限几乎是文献中的“标配”,不仅从数理推导上方便,而且并没有太多的妨碍。Galí(2017)指出代表性家庭无限期生存期限的假设使得模型无法研究如下3类问题。

第一类,收入分配或收入不平等相关问题。这是因为代表性家庭的根本假设是家庭是“相同”的,并没有不同,因而无法研究收入分配或不平等问题。为此必须引入家庭

① Hendry and Muellbauer(2018)(*Oxford Review of Economic Policy*,34(1-2), pp: 287-328.): The notion that the economy follows a stable long-run trend is highly questionable...The world is usually in disequilibrium: economies are wide-sense non-stationary from evolution and sudden, often unanticipated, shifts both affecting key variables directly and many more indirectly. Technology, globalization, both in trade and in finance, trade union power, credit conditions, monetary and fiscal policy rules and other legislation, social mores, skills, wars, resource and financial crises, climate, demography, health and longevity, and income and asset distributions all change over time. These, and other innovations keep perturbing the economic system in ways that not even rational individuals can foresee...

② ... biases that risk holding back progress in macroeconomics.

的异质性 (Heterogeneity) 设定，才能研究此类问题。如果再结合其他模型设定，比如不完全金融市场假设，Galí(2017) 认为至少可以研究如货币政策的收入分配效应、异质性在货币政策传导过程的作用、前瞻指引之谜 (Forward Guidance Puzzle)^①3 方面问题。

第二类，无法对持续性的负自然利率水平进行建模^②。在标准的假设下，代表性家庭的 Euler 方程决定了实际利率与主观贴现因子之间的密切联系（比如在对数消费效用下）：

$$\frac{1}{C_t} = \beta E_t \frac{1}{C_{t+1}} \frac{1+R_t}{1+\pi_{t+1}} \quad (0.2.6)$$

其中， R_t 为名义利率； π_t 为通货膨胀率； $\beta < 1$ 为贴现因子。

如果定义实际利率 r_t 为

$$1+r_t \equiv E_t \frac{1+R_t}{1+\pi_{t+1}} \quad (0.2.7)$$

那么在稳态时，实际利率将是贴现因子的倒数与 1 之差：

$$r = \frac{1}{\beta} - 1 \quad (0.2.8)$$

① 所谓前瞻指引是指央行对短期利率的未来走势进行公开承诺。特别是最近几年，和量化宽松 (Quantitative Easing) 一样，是发达经济体的央行常常使用的工具之一。而标准的 DSGE 模型在估计前瞻指引对宏观经济的影响时，总是高估该影响 (GDP 和通胀等关键的宏观经济变量往往对将来期望的微小变动做出过分的反应，即过于灵敏，而不合实际，Del Negro & Giannoni, 2017)，此被称为前瞻指引之谜。相关文献可参考：Negro M. D., Giannoni M., Patterson C. *The Forward Guidance Puzzle*[J]. Federal Reserve Bank of New York Staff Reports, No.574, 2015. McKay, Alisdair, Emi Nakamura and Jon Steinsson (2016), *The Power of Forward Guidance Revisited*, *American Economic Review*, 106(10), 3133-3158.

② 自然利率水平一般是指模型中不存在各种扭曲或摩擦设定下的实际利率水平（即完全弹性价格和完全竞争下）。也有文献将自然利率水平解释为没有任何货币政策影响下的实际利率水平。自然利率水平也称之为中性利率 (Neutral Interest Rate)，美联储也将其称之为均衡实际联邦基金利率 (Equilibrium Real Fed Interest Rate)，很显然前任美联储主席耶伦女士接受了“自然利率”这一概念，并运用到实际决策中。2016 年 6 月，耶伦在费城的演讲中给出了自己的看法：“... One useful measure of the stance of policy is the deviation of the federal funds rate from a ‘neutral’ value, defined as the level of the federal funds rate that would be neither expansionary nor contractionary if the economy was operating near potential. This neutral rate changes over time, and, at any given date, it depends on a constellation of underlying forces affecting the economy. At present, many estimates show the neutral rate to be quite low by historical standards—indeed, close to zero when measured in real, or inflation-adjusted, terms...”(https://www.federalreserve.gov/newsevents/speech/yellen20160606a.htm)”这也可以从美联储系统研究人员的相关报告和论文中找到蛛丝马迹，如 Del Negro & Giannoni(2017)；Laubach, Thomas, and John C. Williams (2003), “*Measuring the Natural Rate of Interest*”, *Review of Economics and Statistics*, 85(4), November, 1063–1070. 关于自然利率的另外介绍，可参考本书“4.2.7 ‘三方程’新凯恩斯模型”一节中关于自然利率水平的推导。

因此只要 $\beta < 1$, (自然) 实际利率不可能长期为负。然而, 在最近几年里, 特别是发达经济体的短期名义利率接近于零 (Zero Low Bound, ZLB), 甚至在某些极端情况下为负。在温和的通胀下, 这意味着长期以来实际利率为负。因此代表性家庭的无限生存期限的假设很难对此类问题建模。

第三类, 理性泡沫 (Rational Bubble) 问题。Galí(2017) 指出理性泡沫的出现意味着家庭的遍历性约束条件不再成立^①。标准的 DSGE 模型最终的均衡都是平稳的。特别是有关“金融危机”的 DSGE 模型, 其危机发生机制是由于较大的外生冲击一次性突然发生从而导致危机发生。从本质上讲, 这种发生机制仍然是平稳均衡下的“正常”现象, 和实际危机发生的机制并不相符。实际危机的发生首先是危机前的经济或金融过热, 是风险集聚的过程, 是逐步积累的失衡现象, 具有内在非平稳或非线性性质, 并最终由违约、信贷骤停、资产价格骤降引发危机, 造成总需求、就业的严重下降, 从而导致经济萧条^②。Del Negro & Giannoni(2017) 在阐述现行的 DSGE 模型处理前瞻指引问题时, 发现关键变量出现了不可思议的过分灵敏, 从而造成脱离实际的结果。因此, Del Negro & Giannoni(2017) 认为 DSGE 模型在设定上可能存在一定的缺陷。正如 Galí(2017) 指出的那样, 引入家庭异质性约束和贷款与流动性约束等设定有助于解决该问题。但是异质性设定不是没有代价: 模型求解与估计异常困难, 甚至无法求解。因此需要在异质性设定与求解估计的难易度之间寻找一个平衡点或其他的解决办法。纽约联邦储备银行已经着手使用更加合适的编程语言来处理这类问题, 如 Julia^③。

虽然对 DSGE 模型批评与质疑不绝于耳, 但仍有不少学者对 DSGE 模型持有乐观态度, 认为其仍然是未来宏观经济学的核心分析工具。Blanchard(2017) 指出虽然 DSGE 模型具有严重缺陷, 但仍有巨大改善和发展的空间, 仍将是未来宏观经济学的核心, 这一观点也得到了 Galí(2017) 强有力的支持^④。相比 Romer(2016) 的尖锐批评, Blanchard(2017) 的批评更加温和, 更具有包容性和建设性。Blanchard(2017) 列举了 DSGE 模型的 4 个缺陷。

第一, DSGE 模型的前提假设不能令人信服, 甚至和已有关于消费者和厂商的微观研究结论相反。

① 在理性泡沫以名义利率的速度下增长并且由代表性家庭持有的条件下, 遍历性条件将不再被满足。

② Boissay et al. (2016) 则尝试突破这一平稳性限制: Boissay, Frédéric, Fabrice Collard, and Frank Smets (2016), *Booms and Banking Crises*, *Journal of Political Economy*, 124(2), 489 – 538.

③ 此外由 Thomas J. Sargent 等力荐的定量经济学 (Quantitative Economic) 也开始使用 Julia 和 Python 这两个较新的编程语言处理宏观经济问题: <https://lectures.quantecon.org/>.

④ “... I wouldn't know how to disagree.”

第二，DSGE 模型的标准估计方法并不令人信服，即校准加贝叶斯估计的混合方法。也就是说一部分参数先校准，然后在此基础上其他参数使用贝叶斯估计。无论是校准还是贝叶斯估计，在实际研究中都需要满足一定的前提条件或合理假设，然而这些条件或假设往往并不存在或不合理^①。

第三，DSGE 虽然可用于规范经济学的基本意图^②，即依据某些判断标准（比如福利）提出政策建议，但这些判断标准本身的准确性并不令人信服^③。

第四，DSGE 模型是糟糕的交流工具^④，即不能清晰地传递建模思路和逻辑。虽然 DSGE 模型存在上述 4 个缺陷，但 Blanchard(2017) 仍然认为这 4 个缺陷加起来并不足以构成丢弃 DSGE 模型而另辟蹊径的理由。相反，这些缺陷能够被逐步解决，最终 DSGE 模型仍然是宏观经济学基本的工具选择。在 DSGE 模型未来发展的方向上，Blanchard(2017) 给出了两个建设性意见：第一个建议是 DSGE 模型应更具包容性。简言之，经济学各领域的相关发现（比如行为经济学中消费者行为理论、大数据经验研究成果等）都可以而且应该最终纳入 DSGE 分析框架下进行讨论，只有这样才能有更加可行的、合理的模型设定与先验分布的假设，而这恰恰是 DSGE 框架下所欠缺的。第二个建议是不同的研究目的需要有不同的模型，DSGE 模型并非万能。Blanchard(2017) 认为此处不同的 DSGE 模型是指其在理论纯度 (Theoretical Purity) 上的差异。追求最高理论纯度的 DSGE 模型应当更加注重其模型结构的本身合理或清晰与否，而非数据拟合程度。而对于政策研究而言，特别央行或国际金融机构在使用模型，更加注重的是其数据拟合，甚至允许数据来决定各种关系的动态结构，也就是说此时的理论纯度没有那么高。最后对用于预测的模型（如 VAR 模型，BVAR 模型，SVAR 模型等）仍然在一段时间内优于结构模型（比如 DSGE 模型），因此追求理论纯度的 DSGE 模型此时并不占优。

Gali(2017) 同样对 DSGE 分析框架面临的批评进行了强有力的回应：任何反对使用 DSGE 模型人都会发现，很难再找到另外一种能替代 DSGE 模型从而成为宏观经济学分

① Blanchard(2017) 认为校准的参数需要有公认的理论或经验研究支撑，而大多数情况下并不是这样；而贝叶斯估计时参数的先验分布必须是紧致先验分布 (tight prior)，可以将其理解为 informative prior。然而大多数情况下的先验分布往往反映的是研究者自己主观的先验分布，而非参数本身。

② 即经济政策或行为决策应该是什么，带有主观判断，和 positive 相对应，是一对经济学范畴。

③ 以福利为例，Blanchard(2017) 认为福利变化依赖于模型中扭曲设定的方法或引入方式，而研究者往往以求解便利的方式来引入扭曲设定，因而不具有令人信服的福利影响。在经典的福利损失函数与福利水平函数（本书“8.1 条件福利和非条件福利的定义”和“8.3 损失函数法”中会进行简单介绍）中进行选择的话，Blanchard(2017) 认为其会选择前者：经典的福利损失函数法。

④ Blanchard(2017) 认为对于 DSGE 中引入的各种拓展或扭曲设定，往往不能够自圆其说，读者很难理解这种扭曲设定到底有何作用，并且和其他扭曲设定之间有何相互作用关系。

析的核心工具。Gali(2017)认为任何模型或建模技术选择都会受到质疑。而受到质疑并非坏事,反而会成为该模型或技术选择向前发展的推动力,特别是那些有建设性的批评应该受到欢迎。

Ghironi(2017)也对DSGE分析框架面临的批评进行了回应。Ghironi(2017)认为DSGE分析框架之所以受到批评和质疑,其主要原因在于其建模要素甚至求解方法本身。进一步,Ghironi(2017)认为这些批评和质疑完全不是DSGE分析框架的问题所在,而是具体建模和求解选择问题,言下之意是这些批评和质疑并不能成立。Ghironi(2017)指出:第一,DSGE分析框架并不一定要使用理性预期或依赖于外生冲击来驱动经济波动,也不需要贝叶斯估计要求的那样,必须构建一定数量的外生冲击来拟合宏观序列数据;第二,DSGE分析框架并不一定要求有标准的Euler方程(消费的边际效用增长与实际利率之间的关系);第三,DSGE分析框架并不一定要求均衡唯一、完全市场假设、无任何实质作用的金融中介部门(言下之意指完全可以赋予金融中介部门更为复杂的模型角色)、Calvo价格黏性;第四,DSGE分析框架并不一定要求货币政策使用Taylor规则(也可使用数量规则);第五,DSGE分析框架的求解并不一定要求使用一阶、二阶或三阶近似算法(扰动算法);第六,就连DSGE分析框架的基石即微观基础,在具体建模时都可以根据需要选择不同广度和深度进行建模。也就是说,Ghironi(2017)认为上述建模要素或求解只是DSGE模型的一种选择而已,而非全部,也非必需。之所以选择这些建模要素或求解技术,是因为这些要素和技术提供了非常简单的分析基础和框架,帮助理解更为复杂的模型设定的含义和结果。

Williams(2017)认为DSGE分析框架中缺乏对长期经济问题分析的基础。在现行的分析框架下,DSGE模型大都假设外生冲击为暂时性冲击,即外生冲击下系统从最初的稳态回到稳态,聚焦于短期分析^①;从而很难解决央行所关心的中长期现实问题如人口变迁、消费偏好改变和结构变化。这些问题恰恰是长期一系列相关外生冲击作用的累积结果(相当于永久性冲击),而非某一暂时性冲击的结果。因此忽略永久性分析和长期内的结构变化会带来结论偏误。Edge, Laubach & Williams(2007, *JME*)指出暂时性和永久性生产冲击会导致完全不同的财富效应。此外,Williams(2017)还指出假设所有外生冲击为暂时性冲击对预测自然利率水平有严重影响。从长期看,DSGE模型认为自然利率水平为常数,短期受暂时性冲击围绕该常数上下波动。但是经济理论显示自然利率水平随着产出的趋势增长率的变化(Trend Growth Rate of Output)而发生变化^②。Laubach & Williams(2003,

① 因此从本质上讲,Williams(2017)认为DSGE模型不允许存在结构变化。

② 请参考本书“4.2.7 ‘三方程’新凯恩斯模型”一节中的式(4.2.105)。

RES) 通过构建一个允许自然利率水平具有低频时变特征的模型, 发现 20 世纪 90 年代以来自然利率水平呈显著下降趋势, 并非 DSGE 模型假设的常数值。因此, 忽略永久性冲击或长期内外生冲击的累积结果会造成较大的结果偏误。

0.2.4 DSGE与VAR

从方法论上讲, DSGE 和 VAR 两者完全不同。从前述分析可知, DSGE 模型具有微观基础, 基于经济主体的优化并引入各种名义刚性设定, 然后估计并求解, 因此 DSGE 模型更倾向于理论建模, 尽力做到“以理服人”。而 VAR 模型则从统计学和经验分析的角度出发, 将所有变量看作内生变量, 以其各阶滞后变量为解释变量, 建立联立方程, 从而寻找内生变量的 DGP 过程。此外, VAR 模型中一般要求外生冲击(误差项)的个数与内生变量的个数相等, 而 DSGE 模型则不同, 往往外生冲击的个数小于内生变量的个数。为了贝叶斯估计的需要, DSGE 模型在构建时会加入较多的外生冲击, 以引入更多的宏观统计数据序列, 进行参数估计。DSGE 模型中的冲击有两种解释: 第一, 结构性冲击, 比如消费偏好冲击等 (Smets & Wouters, 2003); 第二, 测量误差 (Measurement Error, Ireland, 2004, JEDC)。正如前文指出的那样, 虽然大量外生冲击的加入有助于提高模型数据的拟合能力, 但其牺牲了理论严谨性, 因而可能导致结果偏误。

从上述分析过程可知, DSGE 模型更倾向于理论建模, 而 VAR 模型则更倾向于经验建模。在文献中, DSGE 模型的数据拟合能力常常和 VAR 模型进行对比, 比如 Galí(1999, AER), Christiano, Eichenbaum & Evans (2005), Smets & Wouters(2007) 等。虽然两者在方法论上差异巨大, 但这并不意味着两者之间并非没有关系, 相反, 两者关系密切。在本书“1.1.6 线性模型的状态空间表示”一节可知, 当 DSGE 模型使用一阶求解算法进行求解时(比如对数线性化), 那么在满足一定的条件下, 该状态空间表示可以表示成 VAR 形式^①。

在文献中 DSGE 与 VAR 模型之间的关系被认为是互补关系, 而非竞争关系。SVAR 模型的研究结果和经验证据常常被用来指导 DSGE 分析框架下的建模 (Christiano, Eichenbaum & Vigfusson, 2006)。一个经典的例子是货币政策冲击的脉冲响应。在理性预期假设下, 没有扭曲和摩擦设定的 DSGE 模型对货币政策冲击的脉冲响应很弱, 持续性不强。而 SVAR 模型下, 对于货币政策的冲击、产出和通胀都呈现出明显的“驼峰状” (Hump-shaped)。这种经验研究的发现为后来 DSGE 分析框架下的建模提供了重要

① 矩阵 D 可逆且矩阵 N 的特征值都小于 1。

启示，即通过加入消费习惯、价格和工资黏性设定、投资调整成本等扭曲和摩擦设定，使得 DSGE 模型的结论和 SVAR 模型趋同。反过来，DSGE 模型的研究结果可以缓解 SVAR 模型面临的识别问题 (Identification)。SVAR 模型的识别约束通常是基于方差和协方差矩阵的等号约束，即零约束。然而零约束作为一种较强的识别约束，受到了广泛质疑，因此 Uhlig(2005) 提出了符号约束。而 DSGE 模型则可以为 SVAR 模型提供符号约束指引，即方向性指引。比如 DSGE 模型的研究结果表明，一般情况下，扩张性的货币政策会使得产出增加、价格上升、利率下降。因此，在 SVAR 模型识别时，处理此类问题时，至少不能出现相反的结论。

不可否认，从预测效果看目前 VAR 模型仍然较好，一般领先于 DSGE 模型。不过两者的差距在缩小。在 VAR 模型中，特别是贝叶斯 VAR(Bayesian VAR)，结合先验约束，通常情况下有更好的预测效果。早期的 DSGE 模型主要用于政策分析，而预测则不是其主要目标。然而其不断提高的数据拟合能力，也使得 DSGE 模型具有一定的预测能力。Smets & Wouters(2007) 发现他们构建的 DSGE 模型的预测能力能够媲美贝叶斯 VAR 模型。Del Negro & Schorfheide (2004, *IER*) 则构建了一个既能用于政策分析，又有较好预测能力的 DSGE 模型。

综上所述，DSGE 与 VAR 模型之间的关系也决定了它们各自的用处，因此在相当长的一段时间内，两者将在央行的政策研究工具箱中共存^①。

0.2.5 DSGE与央行

现阶段许多决策机构，特别是中央银行仍在继续使用大型宏观计量模型 (Macroeconometric Models)，并辅以其他的结构模型，如结构 VAR 模型进行辅助决策。虽然中央银行仍然怀疑 DSGE 模型在政策决策，特别是货币政策决策过程中的价值与地位，但不可否认 DSGE 模型已经被广泛纳入世界多数央行的政策研究工具箱中 (见表 0.2)。

^① Olivier Blanchard 在皮尔森经济研究所网站 (PIIE) 发表文章 (*On the Need for (At Least) Five Classes of Macro Models*, <https://piie.com/blogs/realtime-economic-issues-watch/need-least-five-classes-macro-models>, 2017 年 4 月)，指出宏观经济模型至少需要 5 类：第一，基础性模型 (Foundational Models) 比如 Paul Samuelson 的消费—信贷模型，Peter Diamond 的世代交叠 OLG 模型，Ed Prescott 的证券溢价 (Equity Premium) 模型，Diamond, Mortensen & Pissarides 的搜寻 (Search Models) 模型，Neil Wallace 或 Randy Wright 的货币模型；第二，DSGE 模型；第三，政策模型 (Policy Model)，也称之为结构计量模型；第四，玩具模型 (Toy Models)，包括 IS-LM 模型，Mundell-Fleming 模型，RBC 模型和新凯恩斯模型 (New Keynesian)；第五，预测模型 (Forecasting Model)。Olivier Blanchard 强调任何模型都必须建立在局部均衡 (Partial Equilibrium Foundations) 和经验证据 (Empirical Evidence) 之上，而且不同的任务需要不同的宏观模型，而且不同宏观模型之间需要相互学习，取长补短。

表 0.2 世界主要中央银行及其 DSGE 模型

央 行	DSGE 模型
美联储系统	EDO(美国经济模型)、SIGMA(多国模型)
欧洲央行	New Area Wide Model (NAWN); Smets & Wouter(2003)
国际货币基金组织	GEM,GFM,GIMF
英国央行	Medium-Term Macro Models(MTMM,1999), BEQM(2004)/COMPASS(2011)
智利央行	MAS
瑞典央行	RAMSES
秘鲁央行	MEGA-D
挪威央行	NEMO

数据来源：Tovar(2008); Williams(2017)。

表 0.2 中列示了 7 个中央银行和 1 个国际组织所构建的部分 DSGE 模型的名称。众所周知，美国联邦储备系统由位于华盛顿特区的联邦储备委员会和全美 12 家地区性的联邦储备分支机构组成（如纽约、达拉斯、旧金山等）。就目前而言，美联储所开发和使用的 DSGE 模型相比其他央行和国际机构而言较为完备，而在全系统内共享使用，而新的编程语言已经开始被用于解决现行工具很难或无法解决的复杂问题，如家庭异质性问题等。

DSGE 模型之所以能够进入诸多央行的政策工具箱，是因为其有诸多央行认可的优点。

Justiniano, Primiceri&Tambalotti(2017) 认为 DSGE 模型对于中央银行的主要吸引力在于其较好的数据拟合能力，而且在估计后能够“讲好故事”，即可解释过去、预测未来，也能进行随机模拟与仿真试验。而 Del Negro & Giannoni(2017) 以纽约联邦储备银行为例，分析了 DSGE 模型在预测、仿真试验方面的作用。他们认为 DSGE 模型除了能够估计不可观测的外生冲击外，还可以估计（过去）和预测（未来）其他不可观测的重要经济变量如产出缺口和自然利率水平，而这恰恰是纽约联邦储备银行 DSGE 模型一个不可或缺的任务。Del Negro & Giannoni(2017) 还举了一个生动的例子，即通过计算自然利率水平，并与实际利率水平相对比，即可为判断货币政策是否合意提供非常有用的参考^①。

但 DSGE 模型也有其缺陷。很显然，正如本节前述中提到的 George Box 的名言：所有的模型都是错的，只有一些是有用的。因此不论如何努力，都不可能得到完全准确的“真实”模型，因而模型设定偏误 (Misspecification) 会一直存在。Korinek(2017) 指出成为好的研究者其精妙之处在于能够（根据研究的问题来）区分哪些设定可以简化，哪些不可以简化。因此为了尽可能地减小模型设定偏误造成的影响，央行的政策研究工具箱中

① Del Negro & Giannoni(2017) 还指出通过估计自然利率水平，他们发现自金融危机以来，金融风险和其他因素致使家庭储蓄增加，企业削减投资，从而成为推动实际利率下降到较低水平的关键因素所在。

不仅仅包括 DSGE 模型一种工具，很多传统或自建的结构模型仍然在相当长的时间内存在。如美联储系统的政策研究工具箱中，还有许多传统的结构模型，如 FRB/US, FRB/Global(Brayton, Levin & William, 1997)。从稳健性的角度说，工具箱中的工具越多越好，可谓多多益善。依赖任何单一模型、方法或分析框架进行政策分析与决策都具有极大的风险。这是金融危机以来，欧美央行决策者们得到的关键教训之一^①。而且在 DSGE 模型中，也因目的不同会构建不同的模型，比如专门用于预测目的的 DSGE 模型。

Gerlach(2017) 指出虽然 DSGE 模型在货币政策委员会内部讨论时较为有用，但在实际决策时起到的作用非常有限。其主要原因如下：第一，在面对全新的外部冲击时，DSGE 模型的重构耗时耗力，无法满足决策者快速应对的需求。比如最近出现的英国脱欧(Brexit)和大规模难民迁徙问题。此时传统的模型仍然具有优势，比如传统的 VAR 模型，完全可以在其中加入汇率变量来快速考察英国脱欧事件对金融市场的影响。第二，假设 DSGE 不受卢卡斯批判是没有道理的。虽然 DSGE 模型具有微观基础和“深度结构参数”，但其稳定性仍然不足，即结构参数仍然对外部冲击时发生变化，而不是不变。第三，在给定的一个 DSGE 模型下，其覆盖的宏观变量太少，不足以满足决策者的全方位需求。特别是对于金融市场的诸多关键变量，尤其如此。事实上，正如 Buiter(2009) 指出的那样，央行实际决策时，更多依赖于经验法则(Empirical Regularities)、预感(Hunches)、直觉(Intuition)等。

Williams(2017) 指出在美联储实际决策中遇到 3 个非常关键的问题，而美联储的 DSGE 模型在解决这 3 个问题上几乎无能为力：劳动力市场的萧条、生产力的增速放缓以及自然(均衡)利率水平^②。充分就业作为美联储政策目标的重要组成部分，是非常现实的问题。美联储的 DSGE 模型在劳动力市场的建模上几乎是空白的。因此在研究劳动力市场的结构性和周期性波动时，特别当研究失业率、劳动参与、职位空缺和兼职等问题时，DSGE 模型几乎苍白无力。所以，Williams(2017) 认为这在一定程度上限制了

① ... “the key lesson I would draw from our experience is the danger of relying on a single tool, methodology or paradigm. Policy-makers need to have input from various theoretical perspectives and from a range of empirical approaches. Open debate and a diversity of views must be cultivated – admittedly not always an easy task in an institution such as a central bank. We do not need to throw out our DSGE and asset-pricing models: rather we need to develop complementary tools to improve the robustness of our overall framework.” 摘自: Trichet, Jean-Claude (2010), “Reflections on the nature of monetary policy nonstandard measures and finance theory”, Opening address at the ECB Central Banking Conference, Frankfurt, 18 November.

② Williams(2017) 认为自然利率水平为无扭曲设定下的短期实际利率。

DSGE 模型的实际应用价值^①，因此也成为下一步改进的方向。

央行在 DSGE 模型的开发和研究上，从来都不是和纯学术研究相割裂的，相反两者往往都是有机融合在一起的。Justiniano, Primiceri & Tambalotti(2017) 指出了学术圈和央行决策研究圈之间的关系以及未来的发展方向。首先，学术研究的某些结论可称为央行决策的关键依据。比如 Eggertsson & Woodford (2003) 研究了零利率下限约束下的最优货币政策问题，其主要的政策含义在于经济开始复苏之后（金融危机后）仍然需要采用较低的政策利率。而这成为不少央行实施名义利率前瞻指引的重要依据。其次，央行决策研究部署的 DSGE 模型滞后于学术研究：在当前非常规货币政策环境下（比如量化宽松、ZLB），纯学术研究已经着手研究金融中介与实体经济之间的相互作用，以及央行资产负债表效应（大小、组成、资产购买等）对上述相互作用的影响（Gertler & Karadi, 2011, *JME*; Gertler & Kiyotaki, 2011; Del Negro, Eggertsson, Ferrero & Kiyotaki, 2017, *AER*）。但是央行部署的 DSGE 模型目前仍然没有针对这些问题进行建模，因而无法研究相关问题。最后，央行决策研究追求“大而全”的 DSGE 模型，追求模型的功能和覆盖面，以全面辅助决策；而学术研究则更倾向于“小而精”的 DSGE 模型，追求模型的简洁和清晰度，以研究某一问题。而 Justiniano, Primiceri & Tambalotti(2017) 则认为，“小而精”的模型是学术圈和央行决策研究圈进行有效交流和沟通的桥梁与纽带。

0.3 DSGE 模型两种典型构建一瞥

DSGE 模型的构建千差万别，依据问题不同而不同，但本质是对新凯恩斯模型核心三方程在不同程度、不同方面的拓展和修改。本节就几种典型的 DSGE 模型构建做简单的介绍，以给读者一个轮廓性概览，但并不包罗万象。

经常有人问：标准的 DSGE 模型存在吗？答案是否定的。但文献中有几个公认的、经典的 DSGE 模型，如，封闭模型的典型代表作 Christiano, Eichenbaum & Evans (2005, *CEE*, *JPE*)，Smets & Wouters(2007, *AER*)；开放经济的代表作如 Obstfeld & Rogoff (1995, *JPE*)，Galí & Monacelli (2005, *RES*) 等。这些模型都值得精读。CEE(2005) 这篇文献在 DSGE 模型中引入了很多建模元素（如名义刚性），多数已经成为标准的 DSGE 建模要素和出发点，因此 CEE(2005) 成为高被引文献之一。CEE(2005) 引入的主要建模要素包括：

^① 在劳动力市场的建模上，近年来虽然取得了较大的进步，但仍然不足以胜任劳动力市场的深度分析和决策的需求。美联储所构建的 DSGE 模型尤其如此：“... the models in use at the Fed typically treat it (labor market) in a cursory manner or abstract from it entirely.”

可变资本利用率 (Variable Utilization Rate of Capital), 以提升模型的传播机制、消费习惯 (Habit Formation)、工作资本贷款机制 (Working Capital Loan Mechanism) 和投资调整成本 (Adjustment Cost of Investment) 等。另外一篇高被引文献是 Smets & Wouters (2007)。这篇文献构建了中等规模的 DSGE 模型, 并使用贝叶斯估计, 以美国宏观经济数据 (1966 Q1—2004Q4) 估计了 (线性化) DSGE 模型。该模型引入了黏性价格和黏性工资设定, 中间品和劳动使用了 Kimball 加总技术 (Kimball Aggregator, Kimball, 1995, *JMCB*), 而非 Dixit-Stiglitz 加总, 从而允许估计时变需求弹性。模型引入了 7 种冲击: 技术 (TFP) 冲击、风险溢价 (Risk Premium) 冲击、投资边际效率冲击、工资和价格 (Markup) 加成冲击 (也被称之为成本推动型冲击, Cost-Push Shocks) 和两个政策冲击即货币政策 (利率冲击) 与财政政策冲击 (政府支出冲击)。

CEE(2005) 和 Smets & Wouters (2007) 这两篇文献构建的 DSGE 也成为美联储和欧洲央行构建自己的 DSGE 模型的重要参考和起点。

下面对两种主要的建模框架, 给予简单的解释和说明。

0.3.1 CMR框架—金融加速器框架

CMR 框架来源于 Christiano, Motto & Rostagno(2010), 用于构建含有金融加速器机制的 DSGE 模型。金融加速器机制及其建模将在后续章节中具体介绍。此处仅对模型的逻辑架构进行简要阐述, 如图 0.1 所示。

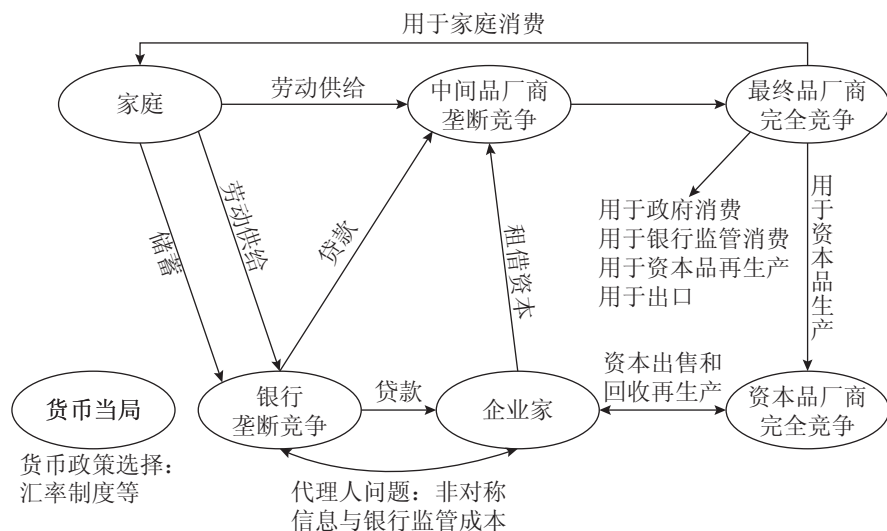


图 0.1 含有金融加速器机制的模型

和含有黏性价格的 DSGE 模型一致，模型包含家庭、中间品厂商和最终品厂商等 3 个部门，其中中间品和最终品厂商是黏性价格设定所必须引入的部门，其中中间品厂商的垄断定价能力恰是黏性价格设定的关键所在。

为了引入金融加速器机制，模型引入了银行、企业家和资本品生产者等行为主体或部门。家庭在银行储蓄，并从银行获取无风险收益。家庭提供劳动给中间品厂商，获取工资收入，最终品生产商生产的产品用于家庭最终消费、资本品再生产等。企业家从银行获取贷款，从资本品生产者那里购买资本品，并出租给中间品厂商。最后货币当局选择货币政策。当然，这个模型框架可以拓展到开放经济中。最终品厂商的产品可用于出口，货币当局可以选择汇率制度等。

0.3.2 小型开放经济模型的架构(产品市场)

此处的小型开放经济模型(产品市场)的架构来源于 Christiano, Trabandt & Walentin(2011, *JEDC*)，如图 0.2 所示。

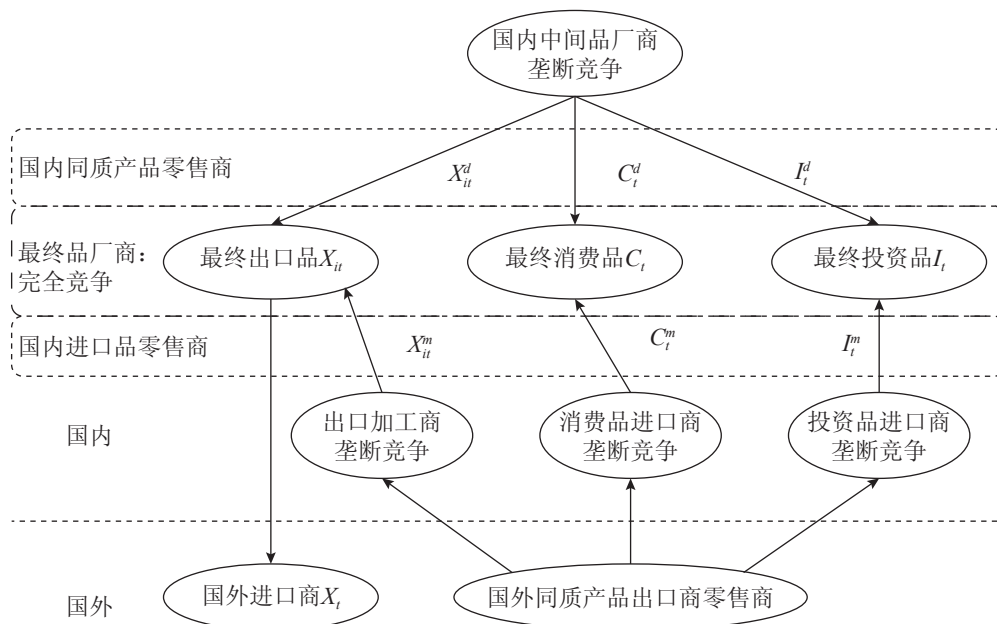


图 0.2 含有进出口部门的开放经济模型

产品市场的架构稍微复杂。同样地，模型含有黏性价格设定，模型存在中间品厂商和最终品厂商。但此处存在 3 类最终品厂商，一类是用于消费的最终品；一类是用于投

资的最终品；一类是用于出口的最终品。为了对开放经济进行建模，3类最终品的生产都引入了国外产品，也就是说最终品是由国内中间品和国外产品的复合。最终出口品的生产包含了国内产品和劳动的部分，也包括了由进口而复加工成分，因此在一定程度上可将最终出口品部门解释为加工贸易部门。

此外，关于开放经济的建模文献还很多。比如 Obstfeld & Rogoff (1995, *JPE*) 被认为是新开放宏观经济模型的开山之作。已经有不少专著对此文献做了专门解读，如 Obstfeld & Rogoff (1996)、Wickens (2008) 和 Walsh (2010) 等，此处不再涉及。Galí & Monacelli (2005) 经典的小型开放经济模型 (Small Open) 已进入教科书，本书将在“7.1.1 货币政策和新开放宏观模型”一节中再做详细介绍。

0.4 宏观经济模型数据库 MMB

宏观经济模型数据库 (Macroeconomics Model Database, MMB) 提供了文献中经典的、著名的宏观经济模型 (特别是 DSGE 模型) 之间相互比较的平台，对每一个 DSGE 模型都提供了对应的模型文件。MMB 软件包运行于 Matlab 平台，是一个不可多得的学习工具和宝贵资料^①。

2017年10月，MMB 的最新版本为 2.2，提供了总共 82 个模型^②，分为 6 类，如表 0.3 所示。随着版本的不断更新，相信会有更多的经典模型被加入 MMB 中，从而使得 MMB 内容更加丰富并更具吸引力。

表 0.3 MMB 2.2 模型分类

类 别	模型个数
小型校准模型	21
估计的美国模型	26
估计的欧元区模型	10
估计或校准的多国模型	8
估计的其他国家模型	6
适用性学习模型	11

数据来源：<http://www.macromodelbase.com/>。

① 更多关于项目的起源、背景的具体介绍，请参考 MMB 官方网站：<http://www.macromodelbase.com/>。

② 共 82 个不同的模型，部分模型被复用，因此总共有 93 个模型。关于所用的模型 MMB 提供了简单的说明，请参考 \MMB_2_2\MMB_OPTIONS\MMB_model_description.pdf。

0.4.1 MMB源文件构成

MMB 官方网站提供了两个软件包。以 MMB 2.2 版本为例，第一个是用于模型间比较的软件包即 MMB_2_2.zip。该软件包内包括了所有用于模型比较的 m 文件和 Dynare 模型文件。但此处的 Dynare 模型文件不仅包括了模型本身的代码，还加入了 MMB 相关的代码，以便于 MMB 用于模型之间的比较，此种类型的模型文件包括了所有的模型^①。MMB_2_2.zip 的目录结构如图 0.3 所示。

名称	修改日期	类型
ALTOOL	2017/6/17 0:52	文件夹
MMB_OPTIONS	2017/9/7 6:32	文件夹
MODELS	2017/6/27 17:47	文件夹
OUTPUT	2017/6/28 0:50	文件夹
MMB.m	2017/9/14 2:04	MATLAB Code
Notes on MMB_AL implementation.txt	2017/2/17 21:12	文本文档
User guide_MMB2_0.pdf	2015/5/7 15:33	Adobe Acrobat

图 0.3 MMB_2_2.zip 的目录结构

第二个是不用于模型比较而只用于学习和复制的软件包，MMB_replications_2_2.zip。这个软件包中，包括了 58 个模型的模型文件，即并未包括全部已覆盖模型。虽然未能全部涵盖，但这些模型文件大都是 MMB 团队收集并经过测试的可运行模型文件，是学习经典文献不可多得的宝贵资料。因此推荐读者阅读相关的模型文件代码，更加深入地理解文献。

0.4.2 MMB使用方法

MMB 提供了非常详细的使用方法说明^②，此处不再详细介绍，而是简单地加以说明。由于不同的模型之间千差万别，模型之间的比较也变得比较困难。因此 MMB 所能提供的比较，只能是特定的货币政策规则下，共同变量的（称为 MMB 变量，通常为利率、通胀、产出和产出缺口等）脉冲响应函数 (IRF) 和自相关函数 (Autocorrelation) 的比较，所以比较的内容也较为有限。

① 请移步至官网下载最新版本的软件包。关于此类模型文件的具体结构和功能，请参考使用说明；MMB_2_2.zip 软件包中包括了 User guide_MMB2_0.pdf，此 PDF 文件为详细使用说明。

② 请参考使用说明。

MMB 使用较为简单,请在 Matlab 中直接运行 MMB_2_2.zip 软件包的 MMB.m 文件,此为主文件和主入口,执行成功后 MMB 将弹出第一个主对话框,如图 0.4 所示。

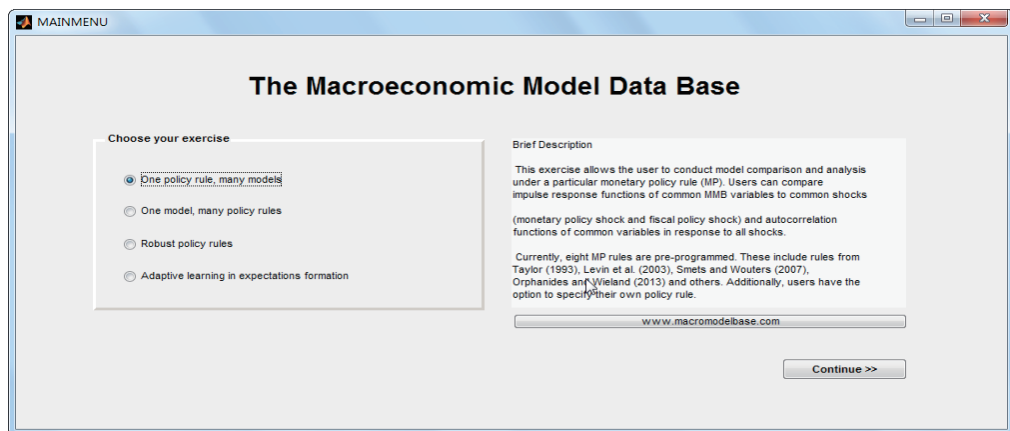


图 0.4 MMB 主运行窗口

MMB 首先要求选择适当的比较方式。目前 MMB 实现了两种模型比较的方式:第一种为同一个货币政策规则不同模型比较;第二种为同一个模型不同货币政策规则的比较^①。目前只支持两种冲击:货币政策冲击(Monetary Policy Shock)和财政政策冲击(Fiscal Policy Shock)。但货币政策规则本身却提供了不同的形式,不仅包括了经典的 Taylor 规则 [Taylor (1993)],而且还包括了文献中其他的不同规则,如 Smets & Wouters (2007)、Christiano, Eichenbaum & Evans (2005) 等,多达 9 种规则^②,并且可以自行指定货币政策规则,灵活性较高。

在选择好模型比较方式后,单击继续(Continue)按钮,MMB 会弹出第二个对话框,如图 0.5 所示。以第一种比较方式为例,即选择一个货币政策规则不同模型之间的比较。在此处,首先选择需要比较的模型,比如 US_SW07 和 US_CMR10 两个模型,然后选择货币政策如 Smets & Wouters (2007),并指定画出 20 期脉冲响应函数,使用货币政策冲击(即一单位正向利率冲击),然后单击继续(Continue)按钮,稍作等待,即可得到结果,脉冲响应图形如图 0.6 所示。如果还选择了“显示无条件方差”(Show Unconditional Variances),那么命令行窗口中会分别显示两个模型共同变量的无条件方差。

① 另外两种比较方式尚未实现: Robust Policy rules 和 Adaptive Learning in expectations formation. 但适应性模型在两种已实现的比较方式中可使用。

② 关于货币政策规则的描述,请参考 \MMB_2_2\MMB_OPTIONS\MMB_MPrule_description.pdf。

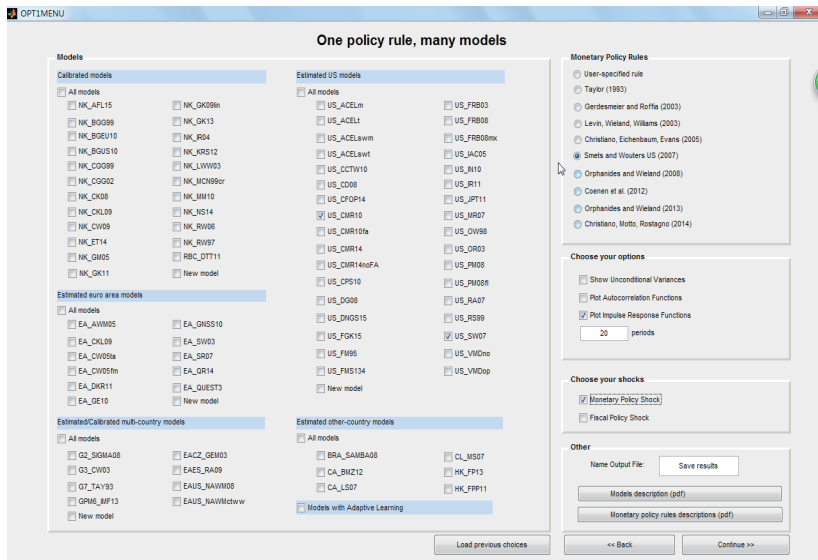


图 0.5 同一货币政策不同模型比较

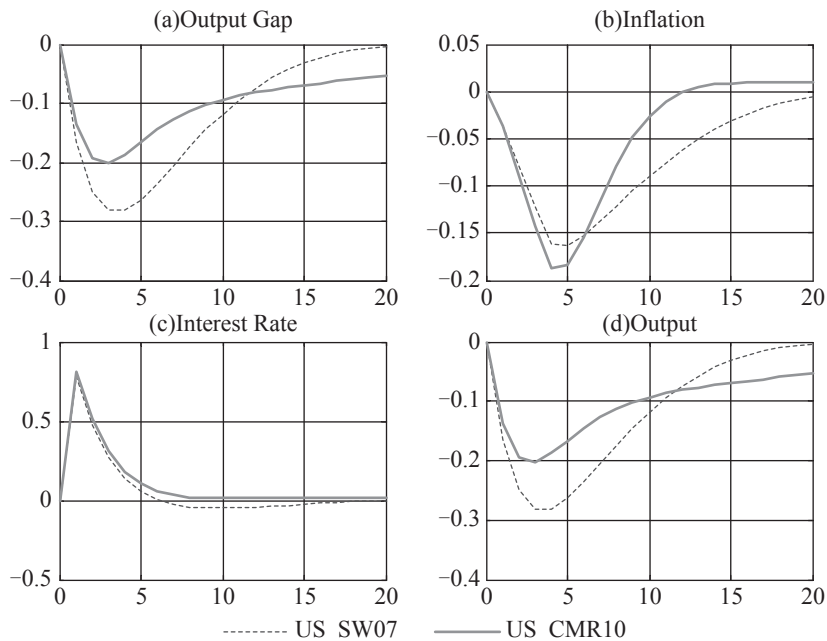


图 0.6 一单位正向货币政策冲击下两个模型的脉冲响应函数

注：Output Gap: 产出缺口；Inflation: 通货膨胀；Interest Rate: 利率；Output: 产出。

参考文献

- [1] Blanchard O. *Do DSGE Models Have a Future?*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:93-100.
- [2] Brayton F., Levin A., Tryon R., Williams J. C.(1997). *The Evolution of Macro Models at the Federal Reserve Board*[J]. Carnegie-Rochester Conference Series On Public Policy,47,December):43-81.
- [3] Buiter W. *The Unfortunate Uselessness of Most “State of the Art” Academic Monetary Economics*[N]. Financial Times, 2009-03-30.
- [4] Christiano L. J., Eichenbaum M., Evans C. L.(2005). *Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy*[J]. Journal of Political Economy,113(1):1-45.
- [5] Christiano L. J., Eichenbaum M., Vigfusson R.(2006). *Assessing Structural VARs*[J]. *NBER Macroeconomics Annual*,21:1-105.
- [6] Christiano L. J., Motto R., Rostagno M.(2014). *Risk Shocks*[J]. American Economic Review,104(1):27-65.
- [7] Christiano L. J., Trabandt M., Walentin K.(2011). *Introducing Financial Frictions and Unemployment Into a Small Open Economy Model*[J]. Journal of Economic Dynamics and Control,35(12):1999-2041.
- [8] Del Negro M., Eggertsson G. B., Ferrero A., Kiyotaki N.(2017). *The Great Escape? A Quantitative Evaluation of the Fed’s Liquidity Facilities*[J]. American Economic Review,107(3):824-857.
- [9] Del Negro M., Giannoni M. *Using Dynamic Stochastic General Equilibrium Models at the New York Fed*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:40-47.
- [10] Del Negro M., Schorfheide F.(2004). *Priors From General Equilibrium Models for Vars*[J]. International Economic Review,45(2):643-673.
- [11] Edge R. M., Laubach T., Williams J. C.(2007). *Learning and Shifts in Long-Run Productivity Growth*[J]. Journal of Monetary Economics,54(8):2421-2438.
- [12] Eggertsson G. B., Woodford M.(2003). *The Zero Bound on Interest Rates and Optimal Monetary Policy*[J]. *Brookings Papers On Economic Activity*,34(1):139-235.

- [13] Friedmann M.(1953). *The Case for Flexible Exchange Rates*[J]. Essays in Positive Economics, Chicago, University of Chicago Press.
- [14] Galí J.(2008). *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*[M]. Princeton University Press.
- [15] Galí J.(2015). *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework and its Applications (2nd Edition)*[M]. Princeton University Press.
- [16] Galí J. *Some Scattered Thoughts On DSGE Models*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:86-92.
- [17] Galí J.(1999). *Technology, Employment, and the Business Cycle: Do Technology Shocks Explain Aggregate Fluctuations?*[J]. American Economic Review, 89(1):249-271.
- [18] Galí J., Monacelli T.(2005). *Monetary Policy and Exchange Rate Volatility in a Small Open Economy*[J]. The Review of Economic Studies,72(3):707-734.
- [19] Gerlach S. *DSGE Models in Monetary Policy Committees*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:31-37.
- [20] Gertler M., Karadi P.(2011). *A Model of Unconventional Monetary Policy*[J]. Journal of Monetary Economics,58(1):17-34.
- [21] Gertler M., Kiyotaki N. *Financial Intermediation and Credit Policy in Business Cycle Analysis*[M].Benjamin M. F. A. M. Handbook of Monetary Economics. Elsevier, 2010:547-599.
- [22] Gihoni F. *Policy Packages: Challenge and Opportunity for DSGE Research*[M]. Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:61-69.
- [23] Greenwood J., Hercowitz Z., Huffman G. W.(1988). *Investment, Capacity Utilization, and the Real Business Cycle*[J]. American Economic Review,78(3):402-417.
- [24] Heer B., Maussner A.(2009). *Dynamic General Equilibrium Modeling-Computational Methods and Applications*[M]. Springer.
- [25] Ireland P. N.(2004). *A Method for Taking Models to the Data*[J]. Journal of Economic Dynamics and Control,28(6):1205-1226.

- [26] Justiniano A., Primiceri G., Tambalotti A. *Empirical DSGE Models: From the Great Moderation, to the Great Recession and Beyond*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:49-59.
- [27] Justiniano A., Primiceri G., Tambalotti A.(2010). *Investment Shocks and Business Cycles*[J]. Journal of Monetary Economics,57(2):132-145.
- [28] Justiniano A., Primiceri G., Tambalotti A.(2011). *Investment Shocks and the Relative Price of Investment*[J]. Review of Economic Dynamics,14(1):102-121.
- [29] Kimball M. S.(1995). *The Quantitative Analytics of the Basic Neomonetarist Model*[J]. Journal of Money, Credit and Banking,27(4):1241-1277.
- [30] Korinek A. *Thoughts on DSGE Macroeconomics: Matching the Moment, but Missing the Point?*[M].Guzman M. Economic Theory and Public Policies: Joseph Stiglitz and the Teaching of Economics. New York: Columbia University Press,forthcoming, 2017.
- [31] Kydland F. E., Prescott E. C.(1982). *Time to Build and Aggregate Fluctuations*[J]. Econometrica,50(6):1345-1370.
- [32] Laubach T., Williams J. C.(2003). *Measuring the Natural Rate of Interest*[J]. Review of Economics and Statistics,85(4):1063-1070.
- [33] McCandless G.(2008). *The ABCs of RBCs: An Introduction to Dynamic Macroeconomic Models*[M]. Harvard University Press, 448.
- [34] Obstfeld M., Rogoff K.(1995). *Exchange Rate Dynamics Redux*[J]. Journal of Political Economy,103(3):624-660.
- [35] Obstfeld M., Rogoff K.(1996). *Foundations of International Macroeconomics*[M]. MIT Press.
- [36] Prescott E. C. *Chapter 22 - RBC Methodology and the Development of Aggregate Economic Theory*[M].Taylor J. B., Uhlig H. Handbook of Macroeconomics. Elsevier, 2016:1759-1787.
- [37] Romer P.(2016). *The Trouble with Macroeconomics*[J]. The American Economist(forthcoming). Delivered Jan.,5,2016 at the Commons Memorial Lecture of the Omicron Delta Epsilon Society. <https://paulromer.net>.
- [38] Smets F., Wouters R.(2003). *An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area*[J]. Journal of the European Economic Association,1(5):1123-1175.

- [39] Smets F., Wouters R.(2007). *Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach*[J]. The American Economic Review,97(3):586-606.
- [40] Stiglitz J. E.(2018). *Where Modern Macroeconomics Went Wrong*[J]. Oxford Review of Economic Policy,34(1-2):70-106.
- [41] Torres J. L.(2015). *Introduction to Dynamic Macroeconomic General Equilibrium Models*[M]. Vernon Art and Science.
- [42] Tovar C. E.(2008). *DSGE Models and Central Banks*[J]. BIS Working Papers,No.258.
- [43] Uhlig H.(2005). *What are the Effects of Monetary Policy On Output? Results From an Agnostic Identification Procedure*[J]. Journal of Monetary Economics,52(2):381-419.
- [44] Uribe M., Schmitt-Grohé S.(2017). *Open Economy Macroeconomics* [M]. Princeton University Press.
- [45] Walsh C. E.(2010). *Monetary Theory and Policy*[M]. MIT Press.
- [46] Wickens M.(2008). *Macroeconomic Theory-A Dynamic Gereral Equilibrium Approach*[M]. Princeton University Press.
- [47] Williams J. C. *DSGE Models: A Cup Half Full*[M].Gürkaynak R. S., Tille C. DSGE Models in the Conduct of Policy: Use as intended. London: Centre for Economic Policy Research (CEPR), 2017:16-21.
- [48] 陈彦斌, 陈惟. 从宏观经济学百年简史看“宏观经济学的麻烦”[J]. 经济学动态, (2017) (1):4-13.



1

DSGE 模型求解逻辑

自从 Kydland & Prescott (1982, *Econometrica*)、King, Plosser & Rebelo (1988, *JME*) 发表以来, 对非线性 DSGE 模型使用线性近似算法求解已经越来越普遍。线性近似求解算法虽然不能解决所有问题, 但在某些情况下仍然不失为一种简单、有效的方法。当外生冲击的波动较小且模型的内点稳定解存在时, 一阶近似算法能够准确地回答诸如均衡局部存在性和确定性、内生变量二阶矩 (Second Moments) 的大小等问题 (Schmitt-Grohé & Uribe, 2004, *JEDC*)。

1.1 DSGE 一阶求解

本节首先介绍一阶求解的基本逻辑, 然后介绍对数线性化方法, 最后对 3 种常见的、简单的一阶求解方法如 B&K 方法、Schur 方法和待定系数法进行介绍, 并分别给出示例, 帮助理解。

1.1.1 一阶求解逻辑

DSGE 模型求解目标是找到模型的政策函数 (Policy Function^①), 即将选择变量 (Control Variables or Choice Variables) 表示成外生变量 (Exogenous Variables or State Variables) 和预先决定内生变量 (Predetermined Endogenous Variables) 的函数。因此, 求解的基本逻辑围绕这一基本目标展开。

1. DSGE 求解逻辑

在求解模型之前, 首先要定义模型均衡。这包括界定模型内生变量和均衡条件 (Equilibrium Conditions)。模型的内生变量一般是选择变量。除预先决定的内生变量外,

① 也称为决策函数, decision function; 决策规则, decision rule。

内生变量一般需要依据模型的外生变量或状态变量来决定。模型均衡条件一般由最优化问题的一阶条件 (FOCs) 和非一阶条件 (如预算约束、资源约束等条件) 组成。值得注意的是, 模型均衡要求内生变量的个数和均衡条件的个数必须相等。否则无法继续求解, 这是求解的必要条件。

其次, 确定模型结构参数 (Structural Parameters)。一般说来, 模型结构参数在求解之前必须已知^①。结构参数可以通过校准 (Calibration) 或估计 (Estimation) 的方法获取其值。参数校准通常从数据中依据某种算法计算数据的某一统计值 (Long Run Moments) 作为参数的值。比如消费占产出的比值, 可使用宏观统计序列, 计算每个时期消费占产出的比值, 然后使用该比值的均值作为参数值。当然, 也可以从经典的文献中直接借鉴。

参数估计是使用某种统计方法和统计数据, 对一个或多个参数进行估计。参数估计的方法多种多样, 常用的有贝叶斯估计 (Bayesian)、极大似然估计 (MLE)、广义矩估计 (GMM)、模拟矩估计 (SMM) 等。本书“3.10 参数估计简介”一节会对贝叶斯估计和极大似然估计做进一步介绍。

很明显, 无论参数的校准还是估计, 都会因方法和数据的不同而存在差异, 从而影响模型求解和后续分析。一般情况下, 如果参数的赋值差异不大, 不会出现截然相反的分析结论, 也就是说不会对定性分析的结果有实质性影响。

最后, 根据需要选择合适的求解算法, 寻找政策函数, 并进行后续分析。目前主流的求解算法多为数值求解。Aruoba, Fernández-Villaverde & Rubio-Ramírez(2006, *JEDC*) 从计算耗时、可用性和复杂度、精确度等方面总结分析了几种常见的 DSGE 模型求解算法如基于扰动项的泰勒近似 (一阶、二阶和三阶, Perturbation Method)、有限元方法 (Finite Element Method)、契比雪夫多项式 (Chebyshev Polynomials) 法、值函数递归 (Value Function Iteration, VFI) 法等, 并附有相关源程序和数据, 有兴趣的读者可以深入探讨学习。

在 Dynare 的求解逻辑中, 使用了基于扰动项的泰勒近似算法 (Taylor Approximation)。从数学分析或高等数学中可知, 泰勒近似算法因近似阶数不同, 会出现不同的近似结果 (就近似精确度而言)。因此在 Dynare 的求解中, 同样存在不同近似阶数的选择问题。泰勒近似会随着阶数的增加, 其复杂度呈几何级数增长, 考虑到 DSGE 模型中存在大量的状态变量和内生变量, 这种复杂程度可想而知。截至目前^②, Dynare

① DSGE 模型很少有解析, 一般都需要数值求解算法来求数值解。因而需要参数值已知, 才能进行数值求解。

② 本书正式写作始于 2016 年年中, 并持续到 2017 年年底。截至 2018 年 10 月, Dynare 的最新版本为 v4.5.6, 但 Dynare++ 中提供四阶、五阶甚至更高阶的求解算法。

最大可求解阶数只为3阶，可见高阶求解具体实现异常复杂。

此外，需要指出的是，泰勒近似求解算法给出的解是局部解，而非全局解。因为泰勒展开定理本身要求基于某个给定的点来近似逼近。在DSGE模型求解中，这个点往往是系统所谓的“稳态”(Steady State)。当然这并不排除基于其他点的近似展开。

2. 一阶求解逻辑

一般来说，对于如下形式DSGE模型的均衡条件：

$$E_t \{ f(y_{t+1}, y_t, y_{t-1}, u_t) \} = 0 \quad (1.1.1)$$

其中， y_t 为内生变量， u_t 为外生冲击 (Exogenous Shocks)，在期初被观测到，满足：

$$E(u_t) = 0, \quad E(u_t u_t') = \Sigma_u, \quad E(u_t u_s') = 0, \quad t \neq s \quad (1.1.2)$$

在 t 期，未知的随机变量仅为 y_{t+1} 和 u_{t+1} ，其余变量均被观测到。

一般情况下，外生冲击只出现在当期，即 t 期变量^①；内生变量 y_t 可能有超过一阶的先导 (lead) 或滞后 (lag) 变量，这时需要引入辅助变量来处理，从而仍然能将均衡写成式 (1.1.1) 的形式。此外，并不是所有的内生变量都有先导或滞后变量出现。

DSGE 模型求解的最终目标是寻找如下的政策函数：

$$y_t = g(y_{t-1}, u_t) \quad (1.1.3)$$

一阶求解首先将均衡条件线性化 (可通过一阶泰勒近似或对数线性化)，然后求解政策函数的一阶近似，即寻找如下形式的一阶近似解：

$$y_t = g(y_{t-1}, u_t) \approx \bar{y} + g_y \hat{y}_{t-1} + g_u u_t \quad (1.1.4)$$

即寻找系数 g_y 和 g_u ，其中 $\hat{y}_t = y_t - \bar{y}$ 表示 y_t 的线性化变量，下一小节有介绍； \bar{y} 为稳态，外生冲击的线性化为其本身。

目前，文献中关于一阶求解的方法有许多种，其中 Blanchard & Kahn(1980, *Econometrica*) (以下简称 B&K 方法)、Klein(2000, *JEDC*)、Sims(2002)、Uhlig(1999) 较为常见。Dynare 的一阶求解使用 Klein(2000) 和 Sims(2002) 提出的 Schur 方法。其中前 3 篇文献探讨的方法较为类似，都通过矩阵分块的思想，先求解特征值进而求解，而 Uhlig(1999) 提出的待定系数法 (Undetermined Coefficient Method) 则使用矩阵的克氏乘积 (Kronecker Product) 来求解。由于求解逻辑的差异，使得 B&K 方法、Schur 方法与待定系数法对变量的划分逻辑也不尽相同。Dynare 的一阶求解逻辑，将在下一节中结合高阶求解逻辑一并介绍。

① 这也是外生变量的特征。

McCandless(2008) 对一阶求解方法有非常详尽的介绍，而且通篇都通过一阶求解的方法进行分析，可当作入门级教程，是非常不错的 DSGE 模型启蒙书籍，值得阅读。

1.1.2 线性化与对数线性化

相比非线性系统，线性化系统的求解最为简单，求解技术最为完备。此外，由于非线性系统可以使用线性系统进行逼近^①，因此求解复杂的非线性系统，研究其一阶近似线性系统的性质，是最自然的选择，也是良好的开端。

对于复杂的非线性 DSGE 模型来说，直接求解几乎是不可能的。而且对于大多数模型，研究其一阶近似系统，已经能够满足解决问题的需要。一阶线性系统最为简单、直观，能够帮助研究者建立变量之间的“直觉”关系，从而有助于理解和解决问题。

本小节将着重介绍如何对 DSGE 模型非线性均衡条件进行线性化处理，并将所有非线性均衡条件转换为线性条件，从而获取原始非线性均衡的一阶近似均衡，然后通过某种求解方法来求解该一阶近似均衡系统。

1. 线性化

线性化不仅针对变量本身，而且更重要的是由变量组成的均衡条件。首先，针对变量 x_t ，定义其线性化变量为 \hat{x}_t ：

$$\hat{x}_t \equiv \frac{x_t - x}{x} \quad (1.1.5)$$

其中， $x \neq 0$ 表示变量 x_t 的稳态值； \hat{x}_t 读作 x_t -hat 即帽子的意思。线性化变量 \hat{x}_t 有时也被称为变量 x_t 的离差形式 (Deviation Form)。

然而针对比率变量，即百分比变量而言，根据百分比变量本身的定义，其线性化变量的定义也稍微不同。下面通过一个简单的例子加以说明。假设通货膨胀率用 π_t 表示，其为比率变量，无量纲，如果其定义为

$$\pi_t \equiv \frac{P_t}{P_{t-1}} \quad (1.1.6)$$

即此时 π_t 为总通货膨胀率，称之为 Gross 变量，此时其线性化变量的定义仍然遵循式 (1.1.5) 的定义。但当 π_t 被定义成如下的形式：

^① 根据魏尔斯特拉斯逼近定理 (Weierstrass Theorem)，闭区间上的连续函数都可以由多项式一致逼近。具体说来，6 大类基本初等函数 (指数函数、对数函数、幂函数、三角函数、反三角函数和常数函数) 和初等函数 (基本初等函数经过有限次四则运算和复合运算得到的函数)，都可以由多项式函数任意逼近。多项式是线性时 (即一次多项式) 为一阶逼近；二次及以上多项式为高阶逼近。

$$\pi_t \equiv \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (1.1.7)$$

即此时 π_t 被定义成净通货膨胀率，称为 Net 变量，此时其线性化变量的定义不同于式 (1.1.5)：

$$\hat{\pi}_t \equiv \pi_t - \pi \quad (1.1.8)$$

此时 π 为净通货膨胀率的目标值或稳态值，比如 2%，而非总通货膨胀率的目标值或稳态值即 102%。如果对于净通货膨胀率定义其线性化变量仍然采用式 (1.1.5) 的形式，会出现无意义的结果，因为对于净值比率，其稳态值或目标值一般比较小，比如 2%，如果此稳态值出现在分母上，此时即使净通货膨胀率仅对其稳态正向偏离 2%，那么其线性化变量会出现 100% 的异常值，此时并没有经济含义，也就是说此时会出现比率的比率，因而没有经济含义：

$$\hat{\pi}_t \equiv \frac{\pi_t - \pi}{\pi} = \frac{4\% - 2\%}{2\%} = \frac{2\%}{2\%} = 1 \quad (1.1.9)$$

对于可导函数 $f(x_t)$ 基于 x_t 稳态值 x 的泰勒近似为：

$$f(x_t) = f(x) + f'(x)(x_t - x) + \frac{f''(x)}{2!}(x_t - x)^2 + \dots \quad (1.1.10)$$

一阶近似只需保留一阶项即线性项，丢弃高阶项即可：

$$f(x_t) \simeq f(x) + f'(x)(x_t - x) \quad (1.1.11)$$

对于二元函数 $f(x_t, y_t)$ 而言，其一阶近似为

$$f(x_t, y_t) \simeq f(x, y) + f_x(x, y)(x_t - x) + f_y(x, y)(y_t - y) \quad (1.1.12)$$

上述一阶泰勒近似所得到的结果完全和一阶线性化 (Linearization) 的结果相同。

2. 对数线性化

对数线性化，顾名思义是使用对数函数的基本性质（本质是对数函数 $\log x$ 或指数函数 e^x 的一阶泰勒展开式）来实现线性化，和上述一般线性化的定义 (1.1.5) 殊途同归。对于均衡条件的对数线性化，特别是均衡条件中含有期望算子 (expectation operator) 时，使用对数线性化比较方便，也比较容易理解。对于给定的变量 x_t ，定义其对数线性化变量 \tilde{x}_t 为

$$\tilde{x}_t \equiv \log x_t - \log x \quad (1.1.13)$$

对数线性化变量 \tilde{x}_t 被定义成变量 x_t 的对数对其对数稳态值的偏离，读作 x_t -tilde^①。

① 由于对数线性化和线性化在本质上没有区别，两者近似相同，因此在本书的后续分析中并未严格区分。

这种方法由 Uhlig(1999) 最早使用。当变量 x_t 在稳态附近时，上述对数线性化式 (1.1.13) 和一般线性化式 (1.1.5) 的定义几乎等价，即 $\tilde{x}_t \simeq \hat{x}_t$ 。这是由于

$$\tilde{x}_t \equiv \log x_t - \log x = \log \frac{x_t}{x} = \log \left(1 + \frac{x_t - x}{x} \right) \simeq \frac{x_t - x}{x} \equiv \hat{x}_t \quad (1.1.14)$$

上式使用了指数函数 e^x 的泰勒展开式：

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad (1.1.15)$$

其中的仅一阶项： $e^x \simeq 1 + x$ ，即 $\log(1+x) \simeq x$ (当 x 较小时)。因而，在无特殊说明的情况下认为一般线性化和对数线性化是等价的，因此可不做任何区别。一般线性化相当于一阶泰勒近似，因而对数线性化的本质是关于对数形式变量的一阶泰勒近似。此外，由式 (1.1.13)，可使用线性化变量表示原变量

$$x_t = x \exp(\tilde{x}_t) \simeq x(1 + \tilde{x}_t) \quad (1.1.16)$$

即原变量大约等于该变量的稳态值乘以 1 加对数线性化变量的和。

由于对数运算具有很多优良的性质，比如将乘积转换为加和形式，将方幂转换为乘法等，因此对数线性化在某些情况下 (如仅含有乘积形式的均衡条件)，已成为线性化的首选方法。

以上主要是针对变量的线性化定义。接下来通过几个简单的例子，介绍如何线性化均衡条件。一般说来对不含有期望算子的均衡条件，其线性化遵循如下的步骤：先取对数，然后泰勒展开：

第一，均衡条件两边取自然对数；

第二，确定各变量的稳态值，然后在等式两边分别进行一阶泰勒展开；

第三，简化、合并同类项、整理表达式。

值得注意的是，上述步骤中的第二步，也可以直接在等式两边取全微分 (更易理解和操作)，而无须一阶泰勒展开，但两者实质上是一致的，具体请参考本节最后一个例子。

对含有期望算子的表达式，建议直接使用式 (1.1.16) 进行替换，并使用幂级数展开公式 (1.1.17) 中的一阶项进行近似计算，即：

$$(1+x)^\alpha \simeq 1 + \alpha x + \frac{\alpha(1-\alpha)}{2} x^2 + \dots, \quad \alpha \neq 0, \quad |x| < 1 \quad (1.1.17)$$

注意，在替换和近似计算的过程中，会出现 \tilde{x}_t 的高阶项和与其他变量的交乘项，这时直接舍弃即可，因为此时只关注线性项。

3. 几个简单的例子

(1) 乘积示例：Cobb-Douglass 生产函数

$$Y_t = A_t K_t^\alpha N_t^{1-\alpha} \quad (1.1.18)$$

两边取自然对数，即

$$\log Y_t = \log A_t + \alpha \log K_t + (1-\alpha) \log N_t \quad (1.1.19)$$

在稳态时成立，即

$$\log Y = \log A + \alpha \log K + (1-\alpha) \log N \quad (1.1.20)$$

在式 (1.1.19) 两边，对每一项在稳态处都进行一阶泰勒展开，可得

$$\log Y + \frac{Y_t - Y}{Y} = \log A + \frac{A_t - A}{A} + \alpha \left(\log K + \frac{K_t - K}{K} \right) + (1-\alpha) \left(\log N + \frac{N_t - N}{N} \right) \quad (1.1.21)$$

因此可得生产函数的线性化表示：

$$\hat{Y}_t = \hat{A}_t + \alpha \hat{K}_t + (1-\alpha) \hat{N}_t \quad (1.1.22)$$

此外，也可以使用对数线性化的定义，将式 (1.1.19) 和式 (1.1.20) 相减，可得式 (1.1.22)。可见，对于只含有乘积的表达式，使用对数线性化将非常简单、方便。

(2) 加和示例：资源约束条件

$$Y_t = C_t + I_t \quad (1.1.23)$$

稳态时，各变量满足

$$Y = C + I \quad (1.1.24)$$

将式 (1.1.23) 减去式 (1.1.24)，两边除以产出的稳态值 Y ，可得

$$\frac{Y_t - Y}{Y} = \frac{C}{Y} \frac{C_t - C}{C} + \frac{I}{Y} \frac{I_t - I}{I} \quad (1.1.25)$$

即

$$\hat{Y}_t = \frac{C}{Y} \hat{C}_t + \frac{I}{Y} \hat{I}_t \quad (1.1.26)$$

此外也可以先取对数，然后使用泰勒近似展开的方法。首先，两边取自然对数可得

$$\log Y_t = \log(C_t + I_t) \quad (1.1.27)$$

然后两边分别一阶泰勒近似可得

$$\log Y + \frac{Y_t - Y}{Y} = \log(C + I) + \frac{C}{C + I} \frac{C_t - C}{C} + \frac{I}{C + I} \frac{I_t - I}{I} \quad (1.1.28)$$

因而同样可得到式 (1.1.26)。

(3) 加和示例：资本积累方程

$$K_{t+1} = I_t + (1-\delta)K_t \quad (1.1.29)$$

稳态时，有 $I = \delta K$ 。上式两边同时减去资本存量的稳态值 K ，然后同时除以 K ，可得

$$\frac{K_{t+1} - K}{K} = \delta \frac{I_t - \delta K}{\delta K} + (1-\delta) \frac{(K_t - K)}{K} \quad (1.1.30)$$

可得线性化方程

$$\hat{K}_{t+1} = \delta \hat{I}_t + (1-\delta) \hat{K}_t \quad (1.1.31)$$

同样，也可以先取对数，然后使用泰勒近似的方法，此处省略。

(4) 带有期望算子的均衡条件：Euler 方程

$$C_t^{-\sigma} = \beta E_t C_{t+1}^{-\sigma} (1+r_t) \quad (1.1.32)$$

首先，实际利率 r_t 的稳态为

$$r = \frac{1}{\beta} - 1 > 0$$

且 r_t 为 Net 比率变量，其线性化定义为 $\hat{r}_t \equiv r_t - r$ 。其次，由于均衡条件中含有期望算子，而且期望算子只具有线性运算的性质，因此暂时不能在两边取对数，此时可考虑使用，即 $x_t = x \exp(\tilde{x}_t) \simeq x(1 + \tilde{x}_t)$ 。将当期消费 C_t 和下一期消费 C_{t+1} 使用该公式进行替换：

$$C_t \simeq C(1 + \hat{C}_t), \quad C_{t+1} \simeq C(1 + \hat{C}_{t+1})$$

代入 Euler 方程

$$\left(C(1 + \hat{C}_t) \right)^{-\sigma} = \beta E_t \left(C(1 + \hat{C}_{t+1}) \right)^{-\sigma} (1 + \hat{r}_t + r) \quad (1.1.33)$$

再由幂级数展开式 (1.1.17)，并简单整理可得 (舍弃交叉项)：

$$1 - \sigma \hat{C}_t = \beta E_t \left(1 - \sigma \hat{C}_{t+1} \right) \left(\hat{r}_t + \frac{1}{\beta} \right) \simeq \beta E_t \left(\hat{r}_t - \frac{1}{\beta} \sigma \hat{C}_{t+1} + \frac{1}{\beta} \right) \quad (1.1.34)$$

因而可得 Euler 方程一阶线性化的方程

$$\hat{C}_t = E_t \left(\hat{C}_{t+1} - \frac{\beta}{\sigma} \hat{r}_t \right) \simeq E_t \left(\hat{C}_{t+1} - \frac{1}{\sigma} \hat{r}_t \right) \quad (1.1.35)$$

(5) 加和示例——全微分

$$Y_t = C_t + I_t \quad (1.1.36)$$

首先两边取自然对数，并在稳态处取全微分：

$$d \log Y_t = d \log (C_t + I_t) \quad (1.1.37)$$

即

$$dY_t \frac{1}{Y_t} \Big|_{Y_t=Y} = \frac{dC_t + dI_t}{(C_t + I_t)} \Big|_{C_t=C, I_t=I} \quad (1.1.38)$$

若定义 $dY_t \equiv Y_t - Y$ ，那么式 (1.1.38) 为式 (1.1.25)。或者在式 (1.1.36) 两边直接取全微分，然后两边同时除以产出的稳态 Y ，同样可得相同的结果。

1.1.3 B&K方法

B&K 方法来源于经典的文献 Blanchard & Kahn(1980)。该方法假设线性化系统具有如下的一般形式^①：

$$\underset{(m+n) \times (m+n)}{\mathbf{B}} \times E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \underset{(m+n) \times (m+n)}{\mathbf{A}} \times \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (1.1.39)$$

其中， \mathbf{B} 、 \mathbf{A} 均为系数矩阵（行对应均衡条件，列对应内生变量）； m 为状态变量或预先决定变量的个数； n 为前向变量的个数。需要指出的是式 (1.1.39) 虽然从形式上预先决定变量 x_{t+1} 具有时间下标 $t+1$ ，但并不表示其为前向变量。

当矩阵 \mathbf{B} 可逆时，问题变得比较简单，即两边同时乘以 \mathbf{B}^{-1} 可得：

$$E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \mathbf{B}^{-1} \mathbf{A} \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (1.1.40)$$

此时系统可使用 B&K 方法进行求解，经过简单的 Matlab 编程即可实现。当矩阵 \mathbf{B} 不可逆时，问题变得稍微复杂，需求使用 QZ 分解来求解（如图 1.1 所示，图中省略了期望算子），具体内容会在本书的“1.1.4 Schur 方法”中做详细介绍。

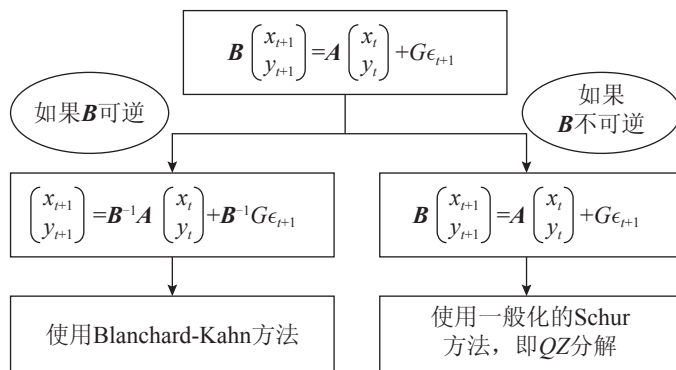


图 1.1 一阶求解方法：B&K 和 Schur 方法

① 此处将状态变量放在前面，前向变量放在后面，这并不影响模型的求解结果，只影响矩阵的定义和模型推导。放置顺序完全可以相反，即前向变量在前，状态变量在后。

1. B&K 方法

本小节首先介绍如何使用 B&K 方法求解系统，然后举例说明。B&K 方法首先要求对矩阵 $B^{-1}A$ 进行相似分解，即寻找矩阵 Γ (Gamma) 和 Λ (Lambda) 满足如下的关系式：

$$B^{-1}A = \Gamma \Lambda \Gamma^{-1} \quad (1.1.41)$$

其中， Γ 为特征向量 (Eigenvector) 组成的矩阵， Λ 为特征值 (Eigenvalue) 组成的矩阵。一般说来， Λ 为对角矩阵。然后对特征值矩阵 Λ 中的特征值依据模值 (Modulus)，从小到大依次排列，形成如下的分块矩阵：

$$\Lambda = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \quad (1.1.42)$$

其中， A_1 中的特征值均小于 1，称 A_1 为稳定的 (Stable)，记为 $|A_1| < 1$ ，假设其维度为 $S \times S$ ； A_2 中的特征值均不小于 1，称 A_2 为非稳定的 (Unstable)，记为 $|A_2| \geq 1$ ，假设其维度为 $T \times T$ 。很显然 $m+n=S+T$ 。

将 $B^{-1}A$ 矩阵分解的结果式 (1.1.41) 和式 (1.1.42) 代入原系统式 (1.1.40)，可得

$$E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \Gamma \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \Gamma^{-1} \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (1.1.43)$$

两边同时左乘 Γ^{-1} ，可得

$$E_t \begin{pmatrix} Z_{1,t+1} \\ Z_{2,t+1} \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} Z_{1t} \\ Z_{2t} \end{pmatrix}, \quad \Gamma^{-1} \begin{pmatrix} x_t \\ y_t \end{pmatrix} \equiv Z_t \equiv \begin{pmatrix} Z_{1t} \\ Z_{2t} \end{pmatrix} \quad (1.1.44)$$

因此可得

$$E_t Z_{1,t+1} = A_1 Z_{1t} \quad (1.1.45)$$

$$E_t Z_{2,t+1} = A_2 Z_{2t} \quad (1.1.46)$$

对于任意给定的初始值 $Z_{2t} = Z_{21}$ ，式 (1.1.46) 意味着：

$$E_t Z_{2,t+1} = A_2 Z_{2t} = A_2^2 Z_{2,t-1} = \dots = A_2^t Z_{21} \quad (1.1.47)$$

注意到 $|A_2| \geq 1$ ，若系统均衡是存在的或者说系统是稳定的，那么必然有：

$$Z_{2t} \equiv 0, \quad \forall t \quad (1.1.48)$$

此为系统稳定或均衡存在的充分必要条件。依此条件为出发点来推导系统的政策函数： $y_t = \phi x_t$ ， ϕ 为常数矩阵。首先对矩阵 Γ^{-1} 进行分块划分如下：

$$\begin{pmatrix} Z_{1t} \\ S \times 1 \\ Z_{2t} \\ T \times 1 \end{pmatrix} \equiv Z_t \equiv \Gamma^{-1} \begin{pmatrix} x_t \\ m \times 1 \\ y_t \\ n \times 1 \end{pmatrix} = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ S \times m & S \times n \\ \Gamma_{21} & \Gamma_{22} \\ T \times m & T \times n \end{pmatrix} \begin{pmatrix} x_t \\ m \times 1 \\ y_t \\ n \times 1 \end{pmatrix}, \quad \Gamma^{-1} \equiv \begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ S \times m & S \times n \\ \Gamma_{21} & \Gamma_{22} \\ T \times m & T \times n \end{pmatrix} \quad (1.1.49)$$

其次, 根据均衡存在的必要条件式 (1.1.48) 和式 (1.1.49) 可得

$$0 \equiv Z_{2t} = \Gamma_{21} x_t + \Gamma_{22} y_t \quad (1.1.50)$$

这是一个具有 T 个方程, n 个未知量 (y_t 的维度为 n) 的齐次线性方程组。线性代数的基本知识表明, 政策函数 $y_t = \phi x_t$ 存在的充分必要条件是 $T = n$, 并且 Γ_{22}^{-1} 存在。该充分必要条件即著名的 B&K 条件^①: 前向变量的个数 (n) 要和不少于 1 的特征值 (Unstable) 的个数 (T) 要相等。当 $T > n$ 时, 线性齐次方程组方程个数大于未知量的个数, 此时无解; 当 $T < n$ 时, 线性齐次方程组方程个数小于未知量的个数, 此时有无穷多组解, 系统未决 (Indeterminacy)。当 $T = n$ 时, 政策函数为

$$y_t = -\Gamma_{22}^{-1} \Gamma_{21} x_t \equiv \phi x_t, \quad \phi \equiv -\Gamma_{22}^{-1} \Gamma_{21} \quad (1.1.51)$$

至此, 完成了政策函数的求解。

将 B&K 方法求解政策函数的基本步骤总结如下:

第一, 对系数矩阵 $B^{-1}A$ 进行相似分解;

第二, 对特征值矩阵 Γ , 按特征值模值从小到大进行排序, 并分块;

第三, 施加系统均衡约束, 即 B&K 条件, 求解政策函数。

在获取政策函数式 (1.1.51) 后, 即可进一步来确定系统状态转换方程。将政策函数式 (1.1.51) 代入原系统式 (1.1.40), 可得:

$$E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = B^{-1}A \begin{pmatrix} x_t \\ \phi x_t \end{pmatrix} \quad (1.1.52)$$

将矩阵 $B^{-1}A$ 写成分块矩阵的形式:

$$B^{-1}A \equiv \begin{pmatrix} BA_{11} & BA_{12} \\ m \times m & m \times n \\ BA_{21} & BA_{22} \\ n \times m & n \times n \end{pmatrix} \quad (1.1.53)$$

因此可将式 (1.1.52) 第一行元素表达为

$$E_t x_{t+1} = (BA_{11} + BA_{12}\phi)x_t \quad (1.1.54)$$

因此式 (1.1.51)、式 (1.1.54) 共同构成了线性化系统的解。接下来, 通过一个简单的

① 相信不少读者都看到过 Dynare 中的一个错误提示: Blanchard & Kahn 条件不满足: Blanchard Kahn conditions are not satisfied: no stable equilibrium. 此即前向变量的个数 (n) 大于或小于特征值大于 1 的个数 (T)。

RBC 模型来讲解，如何通过 Matlab 编程实现使用 B&K 方法求解线性系统。

2. 一个例子：经典 KBC 模型

本小节来考虑一个仅含有投资但不含劳动的简单随机增长模型^①：中央计划者选择消费和资本存量最大化无穷期贴现效用：

$$\max_{\{C_t, K_{t+1}\}} E_0 \sum_{t=0}^{\infty} \beta^t \frac{C_t^{1-\gamma} - 1}{1-\gamma}$$

其中， $\gamma > 0$ 为参数，预算约束为

$$C_t + I_t \leq Y_t \quad (1.1.55)$$

生产函数为经典的柯布道格拉斯形式：

$$Y_t = A_t K_t^\alpha$$

其中， $0 < \alpha < 1$ 为参数，技术生产率 A_t 的对数满足经典的 AR(1) 过程：

$$\log A_t = \rho \log A_{t-1} + \epsilon_t, \quad 0 < \rho < 1 \quad (1.1.56)$$

资本存量满足经典的积累方程：

$$K_{t+1} = I_t + (1-\delta)K_t \quad (1.1.57)$$

效用最大化问题的拉格朗日函数 (Lagrangian) 为^②

$$\mathcal{L} \equiv E_0 \sum_{t=0}^{\infty} \beta^t \left\{ \frac{C_t^{1-\gamma} - 1}{1-\gamma} + \lambda_t (A_t K_t^\alpha - C_t + (1-\delta)K_t - K_{t+1}) \right\}$$

关于消费 C_t 、资本存量 K_{t+1} 和拉格朗日乘子 λ_t 的一阶条件分别为

$$\partial C_t : C_t^{-\gamma} = \lambda_t$$

$$\partial \lambda_t : K_{t+1} = A_t K_t^\alpha - C_t + (1-\delta)K_t$$

$$\partial K_{t+1} : \lambda_t = \beta E_t \lambda_{t+1} (A_{t+1} \alpha K_{t+1}^{\alpha-1} + 1 - \delta)$$

其中，拉格朗日乘子 λ_t 表示消费的边际效用。有时 λ_t 也被称为影子价格 (Shadow Price)，这是指当放松预算约束一个单位时，所能带来的效用恰好为 λ_t 。其中关于资本存量 K_{t+1} 的一阶条件，其具体推导如下 (暂时省略期望算子)：

① 该模型虽然不显示含有劳动，但其实是包括劳动在内，只不过其隐含假设认为劳动不变 (Fixed Labor)，并标准化为单位 1。

② 效用最大化问题的拉格朗日函数也称为即期拉格朗日函数 (current value)，指贴现因子乘以预算约束，这是最常用的方法。之所以被称为即期，是因为拉格朗日乘子代表了当期消费的边际效用。另外一种方法称为现值拉格朗日函数 (present value)，其乘子等于消费的边际效用贴现到第 0 期。

$$\begin{aligned} \mathcal{L} = & \beta^0 \left\{ \frac{C_0^{1-\gamma} - 1}{1-\gamma} + \lambda_0 (A_0 K_0^\alpha - C_0 + (1-\delta)K_0 - K_1) \right\} + \dots + \\ & \beta^t \left\{ \frac{C_t^{1-\gamma} - 1}{1-\gamma} + \lambda_t (A_t K_t^\alpha - C_t + (1-\delta)K_t - K_{t+1}) \right\} + \\ & \beta^{t+1} \left\{ \frac{C_{t+1}^{1-\gamma} - 1}{1-\gamma} + \lambda_{t+1} (A_{t+1} K_{t+1}^\alpha - C_{t+1} + (1-\delta)K_{t+1} - K_{t+2}) \right\} + \dots \end{aligned}$$

关于资本存量 K_{t+1} 求一阶偏导并令其为零 (其他变量视为给定) 可得

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial K_{t+1}} = & \beta^t \frac{\partial \left\{ \frac{C_t^{1-\gamma} - 1}{1-\gamma} + \lambda_t (A_t K_t^\alpha - C_t + (1-\delta)K_t - K_{t+1}) \right\}}{\partial K_{t+1}} \\ & + \beta^{t+1} \frac{\partial \left\{ \frac{C_{t+1}^{1-\gamma} - 1}{1-\gamma} + \lambda_{t+1} (A_{t+1} K_{t+1}^\alpha - C_{t+1} + (1-\delta)K_{t+1} - K_{t+2}) \right\}}{\partial K_{t+1}} \end{aligned}$$

整理可得关于资本存量的一阶条件^①。将关于消费、资本存量和乘子的一阶条件加以整理, 可得 Euler 方程和资源约束方程分别为

$$C_t^{-\gamma} = \beta E_t C_{t+1}^{-\gamma} (A_{t+1} \alpha K_{t+1}^{\alpha-1} + 1 - \delta) \quad (1.1.58)$$

$$K_{t+1} = A_t K_t^\alpha - C_t + (1-\delta)K_t \quad (1.1.59)$$

模型的均衡条件由 Euler 方程 (1.1.58)、资源约束方程 (1.1.59) 和技术生产率的 AR(1) 过程 (1.1.56) 共 3 个方程组成。

模型均衡还有一个重要的必要条件, 即遍历性条件 (Transversality Condition):

$$\lim_{T \rightarrow +\infty} \beta^T \lambda_T K_T = 0 \quad (1.1.60)$$

遍历性条件从有穷期的角度来理解, 会更加容易。在时间终点时, 遍历性条件要求不能留下任何效用, 否则无法最大化效用。这是因为不会再有第二天用来获取剩下的效用。 K_T 代表时间在终点时的资本存量, $\lambda_T K_T$ 表示时间终点时还有未获取的效用, 如果贴现值 $\beta^T \lambda_T K_T$ 不为零, 那么这部分效用将被浪费, 存在“过度储蓄”, 因此无法最大化效用。遍历性条件为必要条件, 无须进入系统均衡条件。

在对均衡条件线性化处理之前, 首先求出系统的稳态值。对于技术生产率变量 A_t 的稳态值, 一般假定为单位 1。一般说来, 服从 AR(1) 过程的变量, 其稳态值都假定为 1, 即稳态时这些变量不对系统有任何贡献。对技术生产率变量 A_t 来说, 它是生产函数的一

① 注意, 最大化问题虽然是基于初始期的期望值, 但这并不妨碍最大化问题关于 t 期变量求导, 并基于 t 期期望得到相应的一阶条件。

个系数，对产出有扩张 ($A_t > 1$) 或紧缩效应 ($A_t < 1$)。此外容易从 Euler 方程 (1.1.58)、资源约束方程 (1.1.59) 求出资本存量和消费的稳态值：

$$K = \left(\frac{\alpha}{\frac{1}{\beta} - (1-\delta)} \right)^{\frac{1}{1-\alpha}}, C = K^\alpha - \delta K$$

其次，对系统进行线性化。根据上一小节介绍的线性化方法，可得如下的线性化均衡系统：

$$\hat{K}_{t+1} = -\frac{C}{K} \hat{C}_t + \frac{1}{\beta} \hat{K}_t + K^{\alpha-1} \hat{A}_t \quad (1.1.61)$$

$$\gamma \hat{C}_{t+1} = \gamma \hat{C}_t + \beta(\alpha-1)\alpha K^{\alpha-1} \hat{K}_{t+1} + \beta\alpha K^{\alpha-1} \hat{A}_{t+1} \quad (1.1.62)$$

$$\hat{A}_t = \rho \hat{A}_{t-1} + \epsilon_t \quad (1.1.63)$$

然后，将上述 3 个线性方程组成的 3 方程、3 变量均衡系统写成 (1.1.39) 形式：

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \gamma \end{pmatrix} E_t \begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \\ \hat{C}_{t+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\beta} & \frac{r}{\alpha} & -\frac{C}{K} \\ 0 & \rho & 0 \\ r(\alpha-1) & \beta r(\rho+r\frac{\alpha-1}{\alpha}) & \gamma - \frac{C}{K}\beta r(\alpha-1) \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \\ \hat{C}_t \end{pmatrix} \quad (1.1.64)$$

其中， $r \equiv \frac{1}{\beta} - (1-\delta) = \alpha K^{\alpha-1}$

最后，校准参数 $\alpha=0.36$, $\beta=0.99$, $\delta=0.025$, $\gamma=2$, $\rho=0.9$ ，根据稳态计算公式，可得资本存量和消费的稳态值分别为 $K=37.9839$, $C=2.7543$ ，总结于如表 1.1 所示。

表 1.1 模型稳态值——BK 方法

变 量	K	C	Y	I
稳 态 值	37.9893	2.7543	3.7041	0.9497

数据来源：依据模型稳态计算逻辑，作者自行计算。

根据式 (1.1.39) 和式 (1.1.64)，可知矩阵 B 可逆，并且容易求出：

$$B^{-1}A = \begin{pmatrix} \frac{1}{\beta} & \frac{r}{\alpha} & -\frac{C}{K} \\ 0 & \rho & 0 \\ \frac{r(\alpha-1)}{\gamma} & \frac{\beta r}{\gamma} \left(\rho + r\frac{\alpha-1}{\alpha} \right) & 1 - \frac{C}{K} \frac{\beta r(\alpha-1)}{\gamma} \end{pmatrix} = \begin{pmatrix} 1.0101 & 0.0975 & -0.0725 \\ 0 & 0.9000 & 0 \\ -0.0112 & 0.0146 & 1.0008 \end{pmatrix} \quad (1.1.65)$$

值得指出的是，此处用于求解的均衡系统中并未包括产出和投资这两个变量。由于这两个变量都可以表示成均衡系统中的 3 个状态变量（资本存量、消费和技术变量）的

线性组合，因此一旦政策函数求出，可最终将产出和投资表示成这3个变量，即状态变量和预先决定变量的函数，因而无须将其包括在均衡系统中。因此，这两个变量也被称为冗余变量 (Redundant Variable) 或静态变量 (Static Variable)。静态变量从时间下标来看，在均衡条件中只出现时刻 t ，而没有出现时刻 $t-1$ 或 $t+1$ ，因此无法将静态变量归入 x_t 和 y_t 中，否则会造成无法求解政策函数，即 Γ_{22}^{-1} 可能不存在的情况。

接下来，将在 Matlab 中进行编程。首先，根据模型定义状态变量和前向变量：

$$x_t \equiv \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix}, y_t \equiv (\hat{C}_t), m=2, n=1$$

其次，求解政策函数：

$$\hat{C}_t = \phi_1 \hat{K}_t + \phi_2 \hat{A}_t \quad (1.1.66)$$

即确定系数 ϕ_1, ϕ_2 ，也即为系数矩阵 ϕ 的元素。如果政策函数 (1.1.66) 一旦确定，可以通过如下的线性化方程，将产出和投资表示为状态变量和预先决定变量的函数：

$$\hat{Y}_t = \hat{A}_t + \alpha \hat{K}_t, \hat{I}_t = \frac{Y}{I} \hat{Y}_t - \frac{C}{I} \hat{C}_t \quad (1.1.67)$$

进一步可表示为

$$\hat{Y}_t = (\alpha, 1) \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix}, \hat{I}_t = \left(\alpha \frac{Y}{I} - \phi_1 \frac{C}{I}, \frac{Y}{I} - \phi_2 \frac{C}{I} \right) \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} \quad (1.1.68)$$

B&K 方法的 Matlab 源代码如下^①：

源代码1 B&K方法的Matlab实现

```
clear;
%the parameters and steady states
beta = .99;
alpha = .36;
gamma = 2;
delta = .025;
rho = .9 ;
ks = (alpha/(1/beta - 1 + delta))^(1/(1-alpha)); %steady state of
capital
cs = ks^alpha - delta*ks; %steady state of consumption;
ck= cs/ks; %consumption capital ratio
R = 1/beta - 1 + delta; %simplifying parameter
ys = ks^alpha; %steady state of prouduction
is = ys - cs; %steady state of investment

%preparation for solution
n = 1; %number of jumper
m =2; %number of states
```

① 具体代码请参考 \Sources\Chap1_DSGE_basics\1.1_1st_logic\BK\BK.m。

```

BA=zeros(3,3); %the coefficient matrix
BA(1,1)= 1/beta;
BA(1,2)= R/alpha;
BA(1,3)= -ck;
BA(2,2) = rho;
BA(3,1)=(alpha-1)*R/ gamma;
BA(3,2) =beta*R*(rho + (alpha-1)*R/alpha)/ gamma;
BA(3,3)=1- ck*(alpha-1)*beta*R/ gamma;
[vv,lamb] = eig(BA); %find the eigenvalues of BA;
[lamb_sorted, index] = sort(abs(diag(lamb)));

%sort the eigenvector matrix
for ii = 1:m + n
    vv_sorted(:,ii) = vv(:,index(ii));
end

%find the index of the first eigenvalue whose value >=1
first_unstable_index = find(abs(lamb_sorted)>=1, 1 );

% num of stable eigenvalue
S = first_unstable_index - 1;
% num of unstable eigenvalue
T = m+n - S;

Gamma = inv(vv_sorted);
Gamma21 = Gamma(S+1:S+T,1:m); % low left, size = T*m;
Gamma22 = Gamma(S+1:S+T,m+1:n+m);% lower right, size = T*n;
%the policy function coefficients
pol= -inv(Gamma22)*Gamma21;
    
```

上述 Matlab 源代码中，仅用于求解本例。如果用于求解其他例子，有几个地方需要注意：

第一， $B^{-1}A$ 矩阵的元素赋值是根据模型不同而不同，应依情况赋值。

第二，获取特征值使用了 Matlab 的内置函数 `eig`；排序使用了内置函数 `sort`；寻找第一个大于 1 的特征值的位置，使用了内置函数 `find`^①。

第三，如果线性均衡系统 (1.1.40) 中变量 x_t 和 y_t 的位置发生互换，此时推导部分和相应的 Matlab 代码也应做相应的变化。感兴趣的读者可使用本例自行推导，结果应该是一样的。

直接运行上述源代码，可得矩阵 ϕ ，即政策函数的系数矩阵：

$$\phi = (0.4629, 0.2276) \quad (1.1.69)$$

① 关于内置函数的具体使用方法，请在 Matlab 命令行中使用 `help` 命令如 `help sort`，即可找到关于 `sort` 函数的基本语法信息。

因而政策函数和静态变量的表达式如下^①：

$$\hat{C}_t = 0.4629\hat{K}_t + 0.2276\hat{A}_t \quad (1.1.70)$$

$$\hat{Y}_t = \hat{A}_t + 0.36\hat{K}_t \quad (1.1.71)$$

$$\hat{I}_t = 3.2401\hat{A}_t + 0.0616\hat{K}_t \quad (1.1.72)$$

用矩阵形式可表达为

$$\begin{pmatrix} \hat{C}_t \\ \hat{Y}_t \\ \hat{I}_t \end{pmatrix} = \begin{pmatrix} 0.4629 & 0.2276 \\ 0.3600 & 1 \\ 0.0616 & 3.2401 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} \quad (1.1.73)$$

然后根据技术变量 \hat{A}_t 的AR(1)过程，把资本存量和技术变量组成的二维列向量表示成：

$$\begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0.9000 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \epsilon_t \quad (1.1.74)$$

上式代入即可得到最终的政策函数和静态变量的表达式：

$$\begin{pmatrix} \hat{C}_t \\ \hat{Y}_t \\ \hat{I}_t \end{pmatrix} = \begin{pmatrix} 0.4629 & 0.2048 \\ 0.3600 & 0.9000 \\ 0.0616 & 2.9161 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} 0.2276 \\ 1.0000 \\ 3.2401 \end{pmatrix} \epsilon_t \quad (1.1.75)$$

在获取政策函数和静态变量表达式后，接下来可求出状态转换方程。根据式(1.1.54)和式(1.1.65)，可得：

$$\begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \end{pmatrix} = \begin{pmatrix} 0.9765 & 0.0810 \\ 0 & 0.9000 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} \quad (1.1.76)$$

结合式(1.1.74)，不难求得系统的状态转换方程为

$$\begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \end{pmatrix} = \begin{pmatrix} 0.9765 & 0.0729 \\ 0 & 0.9000 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} 0.0810 \\ 1.0000 \end{pmatrix} \epsilon_t \quad (1.1.77)$$

即可进行后续的分析，如脉冲响应分析(IRF, Impulse Response Function)和随机模拟(Simulation)等。本书“1.1.7 脉冲响应和随机模拟”会提及脉冲响应和随机模拟分析，更

^① 感兴趣的读者，可以验证此结论。使用Dynare编写该模型的mod文件，然后运行即可验证。注意，此处B&K方法是使用线性化变量进行求解，因此mod文件必须使用对数水平log-level的形式，关于对数水平书写方法详见本书“3.4 内生变量的表达形式：level or log-level”一节中的介绍。此处mod文件具体源代码请参考：\Sources\Chap1_DSGE_basics\1.1_1st_logic\BK\BK_check.mod。

多详细地分析参见本书的后续章节，如“3.9 随机模拟分析：stoch_simul”、“6.2 脉冲响应函数和自定义编程”。

1.1.4 Schur方法

本小节介绍线性模型求解的另外一个方法：Schur 方法。Schur 方法比 B&K 方法更为一般，因此适用求解范围更大一些。Schur 方法的本质是 QZ 矩阵分解。在矩阵分解后的推导几乎和 B&K 方法一致。

首先给出理论推导，然后举例说明如何使用 Matlab 实现使用 Schur 方法求解简单的随机增长模型。

1. Schur 方法

在上一小节中，已经说明了何时使用 Schur 方法。即当线性化系统式 (1.1.39)

$$\underset{(m+n) \times (m+n)}{\mathbf{B}} \times E_t \begin{pmatrix} x_{t+1} \\ \vdots \\ y_{t+1} \\ \vdots \\ x_t \\ \vdots \\ y_t \end{pmatrix} = \underset{(m+n) \times (m+n)}{\mathbf{A}} \times \begin{pmatrix} x_t \\ \vdots \\ y_t \end{pmatrix}$$

中的系数矩阵 \mathbf{B} 不可逆时，需要对系数矩阵 \mathbf{A} 、 \mathbf{B} 使用 QZ 分解。所谓 QZ 分解是指对于两个矩阵 \mathbf{A} 、 \mathbf{B} ，存在正交矩阵 \mathbf{Q} 、 \mathbf{Z} (Orthogonal Matrix)^①，使得

$$\mathbf{A} = \mathbf{Q}\mathbf{U}\mathbf{Z}^T, \quad \mathbf{B} = \mathbf{Q}\mathbf{V}\mathbf{Z}^T \quad (1.1.78)$$

其中， \mathbf{U} 、 \mathbf{V} 均为上三角矩阵 (Upper Triangular)^②， \mathbf{Z}^T 表示 \mathbf{Z} 的转置。 \mathbf{U} 、 \mathbf{V} 对角线上的元素为一般化的特征值 (Generalized Eigenvalues)^③。系统的特征值由下式定义：

$$\lambda_{ii} \equiv \begin{cases} \frac{u_{ii}}{v_{ii}} & v_{ii} \neq 0 \\ +\infty & u_{ii} > 0, v_{ii} = 0 \\ -\infty & u_{ii} < 0, v_{ii} = 0 \\ \in \mathbf{C} & u_{ii} = 0, v_{ii} = 0 \end{cases} \quad (1.1.79)$$

其中， u_{ii} 、 v_{ii} 是矩阵 \mathbf{U} 、 \mathbf{V} 对角线上第 i 个元素。当 u_{ii} 、 v_{ii} 都为零时 (\mathbf{C} 表示复数集)，此时特征值为复值 (Complex Eigenvalues，成对出现)。将其代入原系统 (1.1.39)，可得

① 所谓正交矩阵是指该矩阵左乘、右乘以其转置都等于单位矩阵。如果一个矩阵是正交矩阵，那么它的转置也是正交矩阵。

② 更准确地说，矩阵 \mathbf{U} 可为准上三角矩阵 (Quasi-upper Triangular)。

③ 可使用 Matlab 函数 `diag` 获取 \mathbf{U} 、 \mathbf{V} 对角线上的元素：`alpha=diag(U)`；`beta=diag(V)`。而系统的特征值有 `alpha` 向量点除 `beta` 向量获得：`lambda= alpha./beta`。或者使用 `eig(A,B)` 命令获取 `lambda`。特征值的排序是指对 `lambda` 中元素的排序。

$$QVZ^T E_t \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = QUZ^T \begin{bmatrix} x_t \\ y_t \end{bmatrix} \quad (1.1.80)$$

消去矩阵 Q ，并将 U 、 V 、 Z^T 写成分块矩阵的形式（假设分块矩阵 $|V_{22}^{-1}U_{22}| \geq 1$ ，即系统特征值已经按照从小到大的顺序排列）^①，可得

$$\begin{pmatrix} V_{11} & V_{12} \\ 0 & V_{22} \end{pmatrix} \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix} \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (1.1.81)$$

先考察第二行的元素：

$$V_{22} E_t (Z_{21} x_{t+1} + Z_{22} y_{t+1}) = U_{22} (Z_{21} x_t + Z_{22} y_t) \quad (1.1.82)$$

假设 V_{22} 可逆，注意到 $|V_{22}^{-1}U_{22}| \geq 1$ ，如果系统均衡存在，则要求：

$$Z_{21} x_t + Z_{22} y_t = 0, \quad \forall t \quad (1.1.83)$$

和 B&K 方法一样，系统均衡存在需要满足 B&K 条件： $T = n$ ，并且 Z_{22}^{-1} 存在。如果均衡存在，由式 (1.1.83) 可知政策函数为

$$y_t = -(Z_{22}^{-1}) Z_{21} x_t \equiv -N x_t \quad (1.1.84)$$

其中， $N \equiv (Z_{22}^{-1}) Z_{21}$ 。将政策函数 (1.1.84) 代入原系统 (1.1.39)，并将矩阵 A 、 B 写成分块形式：

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} E_t \begin{pmatrix} x_{t+1} \\ -N x_{t+1} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_t \\ -N x_t \end{pmatrix} \quad (1.1.85)$$

第一行的元素可写为

$$(B_{11} - B_{12} N) E_t x_{t+1} = (A_{11} - A_{12} N) x_t \quad (1.1.86)$$

如果 $B_{11} - B_{12} N$ 可逆，则有

$$E_t x_{t+1} = (B_{11} - B_{12} N)^{-1} (A_{11} - A_{12} N) x_t \equiv M x_t \quad (1.1.87)$$

其中， $M \equiv (B_{11} - B_{12} N)^{-1} (A_{11} - A_{12} N)$ 。因而线性系统的解为

$$\begin{aligned} y_t &= -N x_t \\ E_t x_{t+1} &= M x_t \end{aligned} \quad (1.1.88)$$

2. 一个例子：不可分劳动模型

经典的 RBC 模型，总是假设劳动的波动来自于平均工作时长 (Intensive Margin，集

^① 即系统的特征值 λ_{ii} 已经按照从小到大的顺序排列。Christopher A. Sims 曾专门写过一个 Matlab 程序 qzdiv.m，来专门处理这个问题。本节的模型中同样使用这个 Matlab 文件。该文件位于 \Sources\Chap1_DSGE_basics\1.1_1st_logic\schur_QZ 目录下。Matlab 也有内置函数 ordqz，能达到同样的目的。

约边际)，而不是来源于工作和失业这两种状态来回转换造成的波动 (Extensive Margin, 广延边际)。一般说来，根据劳动合同，劳动者的劳动时间是相对固定的，比如说每天 8 小时工作制。而劳动时长的波动更多来自于工作和失业这两种状态的转换。因此，RBC 模型的假设和实际情况相差较远。

现实生活中，很少有工人能够自行调节工作时间长度。一般说来，要么劳动合同大都要求每周或每月固定工作时长，要么就没有劳动合同，即失业状态，工作时间为零。这种要么工作、要么失业的劳动时间假定，实际上暗含着劳动时间在微观上不可分的含义或者说不能自由分配，这也是不可分劳动的由来。Hansen(1985, *JME*) 将不可分劳动 (Indivisible Labor) 引入经典的 RBC 模型，成功解决了 RBC 模型中劳动的波动率较小的问题。

不可分劳动模型，假设家庭和厂商签订劳动合同，每期以概率 α_t 提供固定时间的劳动 h_0 ，以概率 $1 - \alpha_t$ 处于失业状态，劳动时间为零。因此，平均来说，家庭每期提供的期望劳动时间为 $H_t \equiv \alpha_t h_0$ 。无论工作与否，家庭都会得到与期望劳动时间相匹配的工资 w_t 。也就是说，不可分劳动实际上引入了一种失业保险，从而平滑消费和工作时间。假设效用函数是消费和工作时间的二元函数，因此使得效用最大化问题的定义域是凸集合，即消费和工作时间组成的二维平面 (McCandless, 2008)。

Hansen(1985) 的不可分劳动模型，是一个含有资本存量和劳动的简单随机增长模型^①。假设家庭的可用时间被标准化为 1，效用函数是消费、期望劳动时间的函数：

$$U(C_t, \alpha_t) = \log C_t + \alpha_t \log(1 - h_0) + (1 - \alpha_t) \log(1) \quad (1.1.89)$$

其中， h_0 为固定工作时间。式 (1.1.89) 中后两项表示家庭从提供的期望劳动时间 $H_t \equiv \alpha_t h_0$ 中获取的期望效用。简单整理后，可得

$$U(C_t, H_t) = \log C_t + H_c H_t, \quad H_c \equiv \frac{\log(1 - h_0)}{h_0} < 0 \quad (1.1.90)$$

其中， $H_c < 0$ 为待决系数。不可分劳动的引入，使得劳动以一次项的形式出现在效用函数中。这区别于普通的效用函数形式，如 $U(C_t, H_t) \equiv \log C_t + \chi \log(1 - H_t)$ ， χ 为参数。

家庭选择消费、资本存量和劳动来最大化如下的终身贴现效用：

$$\max_{C_t, K_{t+1}, H_t} E_0 \sum_{t=0}^{\infty} \beta^t (\log C_t + H_c H_t) \quad (1.1.91)$$

^① 此例子来源于 McCandless(2008)，P112。本书此处关于变量的处理方式和状态空间的表达有别于 McCandless(2008)。McCandless(2008) 将产出当作状态变量或预先决定变量的做法，有待商榷。

其预算约束为

$$\begin{aligned} C_t + K_{t+1} - (1-\delta)K_t &= Y_t \\ Y_t &= A_t K_t^\theta H_t^{1-\theta} \end{aligned} \quad (1.1.92)$$

$$\log A_t = \gamma \log A_{t-1} + \epsilon_t$$

其中, $\theta, \gamma, \delta, \beta > 0$ 为待决参数。关于消费、资本存量和劳动的一阶条件为

$$1 = \beta E_t \left(\frac{C_t}{C_{t+1}} (r_{t+1} + (1-\delta)) \right) \quad (1.1.93)$$

$$C_t = -\frac{(1-\theta)Y_t}{H_c H_t} \quad (1.1.94)$$

$$r_t \equiv \theta \frac{Y_t}{K_t} \quad (1.1.95)$$

其中, 资本存量的收益率 r_t 被定义成资本的边际产出。

接下来, 计算模型的稳态值。假设不带时间下标的变量表示对应的稳态值。由 Euler 方程 (1.1.93), 可得资本存量收益率的稳态为

$$r = \frac{1}{\beta} - (1-\delta) \quad (1.1.96)$$

由资本收益率的定义 (1.1.95) 可得, 产出资本比为

$$\frac{Y}{K} = \frac{r}{\theta} \quad (1.1.97)$$

对劳动的稳态值, 采取和 McCandless(2008) 相同的校准值: $H = 0.3335$, 因此由生产函数的定义 (1.1.92), 可得资本存量的稳态值为

$$K = \left(\frac{r}{\theta} \right)^{\frac{1}{1-\theta}} H \quad (1.1.98)$$

从而依据产出资本比 (1.1.97) 来求出产出的稳态值 Y , 依据资源约束方程 (1.1.92) 求出消费的稳态值 C , 依据劳动一阶条件 (1.1.94) 求出待决参数 H_c 。

首先, 对结构参数进行校准。假定 $\beta = 0.99$, $\delta = 0.025$, $\theta = 0.36$, $\gamma = 0.95$ 。以此计算各变量的稳态如表 1.2 所示。

表 1.2 不可分劳动模型各变量的稳态值

变 量	Y	C	K	r	H	H_c
稳 态 值	1.235 3	0.918 6	12.669 4	0.035 1	0.333 5	-2.580 8

数据来源: 依据模型稳态计算逻辑, 作者自行计算。

然后，将均衡线性化，线性化后的一阶条件为

$$K\hat{K}_{t+1} = Y\hat{Y}_t - C\hat{C}_t + (1-\delta)K\hat{K}_t \quad (1.1.99)$$

$$\hat{A}_t = \gamma\hat{A}_{t-1} + \epsilon_t \quad (1.1.100)$$

$$0 = \hat{A}_t - \theta\hat{Y}_t + \theta\hat{K}_t - (1-\theta)\hat{C}_t \quad (1.1.101)$$

$$0 = \hat{K}_t + \hat{r}_t - \hat{Y}_t \quad (1.1.102)$$

$$E_t\hat{C}_{t+1} - \beta E_t\hat{r}_{t+1} = \hat{C}_t \quad (1.1.103)$$

和 B&K 方法处理的方式一致，求解模型时，需要暂时消除静态变量：产出。如果包括产出变量，此时 Z_{22}^{-1} 不存在，从而无法求出政策函数。由资本收益率方程 (1.1.102) 将产出替换为资本存量和收益率的和，然后代入资源约束方程 (1.1.99) 和生产函数方程 (1.1.101)。

再次，定义模型均衡。此时模型均衡由 4 个变量即消费 \hat{C}_t 、资本收益率 \hat{r}_t 、资本存量 \hat{K}_t 、技术生产率变量 \hat{A}_t 和 4 个均衡条件即资源约束方程 (1.1.99)、技术生产率方程 (1.1.100)、生产函数方程 (1.1.101) 和 Euler 方程 (1.1.103) 组成。定义状态变量和前向变量如下：

$$x_{t+1} \equiv \begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \end{pmatrix}, y_{t+1} \equiv \begin{pmatrix} \hat{C}_{t+1} \\ \hat{r}_{t+1} \end{pmatrix}, E_t \begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} \equiv E_t \begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \\ \hat{C}_{t+1} \\ \hat{r}_{t+1} \end{pmatrix} \quad (1.1.104)$$

因此，状态变量的个数 $m=2$ ，前向变量的个数 $n=2$ 。因此目标是求解

$$\begin{pmatrix} \hat{C}_t \\ \hat{r}_t \end{pmatrix} = -N \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix}, E_t \begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \end{pmatrix} = M \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} \quad (1.1.105)$$

中的系数矩阵 N 、 M 。将线性化的系统写成 (1.1.39) 的形式，则系数矩阵 A 、 B 分别为

$$A \equiv \begin{pmatrix} Y+(1-\delta)K & 0 & -C & Y \\ 0 & \gamma & 0 & 0 \\ 0 & 1 & -(1-\theta) & -\theta \\ 0 & 0 & 1 & 0 \end{pmatrix}, B \equiv \begin{pmatrix} K & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\beta \end{pmatrix} \quad (1.1.106)$$

很显然，矩阵 B 不可逆，无法使用 B&K 方法。

最后，编程实现。源代码 2 给出了求解不可分劳动模型 Schur 方法的 Matlab 实现^①。代码分为 4 个部分：第一部分为参数校准和内生变量稳态计算；第二部分为矩阵 A 、 B 的定义，即依据 (1.1.106) 进行元素逐一赋值；第三部分即 QZ 分解；第四部分首先依

① 此 Matlab 代码位于 \Sources\Chap1_DSGE_basics\1.1_1st_logic\schur_QZ\schur_stochastic.m。

据式 (1.1.84) 求解政策函数的系数矩阵 N ，然后根据式 (1.1.87) 求解系数矩阵 M 。具体源代码如下：

源代码2 Schur方法示例

```
%Schur Decompostion Example
%Hansen(1985, JME): Indivisible labor model
clear all;
beta =0.99;
delta =0.025;
theta = 0.36;
gamma= 0.95;

Hbar = 1/3;
rs = 1/beta -(1-delta);
Kbar = (theta/rs)^(1/(1-theta))*Hbar;
Ybar = Kbar^theta*Hbar^(1-theta);
Cbar = Ybar - delta*Kbar;
Hc=-(1-theta)*Ybar/Cbar/Hbar;

%define the coefficient martix
A= zeros(4,4);    B=zeros(4,4);
A(1,1)=(1-delta)*Kbar + Ybar;    A(1,3)=-Cbar;    A(1,4) = Ybar;
A(2,2)=gamma;                A(3,2)=1;    A(3,3)=- (1-theta);
A(3,4)=-theta;                A(4,3)=1;

B(1,1)=Kbar;    B(2,2)=1;    B(4,3)=1;    B(4,4)=-beta;

%the number forward-looking variables
%one is consumption and the other is capital rental rate.
n=2;

% QZ decomposition: A,B is defined above
[U1,V1,Q1,Z1,Vec1,Vec2]=qz(A,B);

%order the general eigenvalues in ascending manner
[V,U,Q,Z,Vec1]=qzdiv(1,V1,U1,Q1,Z1,Vec1);
%[V,U,Q,Z,Vec1]=ordqz(1,V1,U1,Q1,Z1,Vec1);

alpha=diag(U);
beta =diag(V);
%you can verify the lower right conner to make sure the eigenvalues are
%probably ordered.
alphabeta=[alpha beta];
%recover the transpose of Z
Zp=Z';
[a,b]=size(Z1);
```

```

%recover the policy function coefficients matrix
%where N and C corresponds to the ones defined in the text.
%inv 为Matlab内置函数，表示求逆矩阵
N=inv (Zp(a-n+1:a,a-n+1:a))*Zp(a-n+1:a,1:a-n);
invBBN=inv(B(1:a-n,1:a-n)-B(1:a-n,a-n+1:a)*N);
M=invBBN*(A(1:a-n,1:a-n)-A(1:a-n,a-n+1:a)*N);
    
```

值得强调的是，源代码 2 中的第三部分 QZ 分解，首先调用了 Matlab 内置函数 `qz`： $[U_1, V_1, Q_1, Z_1, Vec_1, Vec_2] = qz(A, B)$ ，该函数接受两个参数即矩阵 A 、 B ，然后返回 6 个参数并满足： $Q_1 * A * Z_1 = U_1$ ， $Q_1 * B * Z_1 = V_1$ 。其中 Q_1 、 Z_1 为正交矩阵。 U_1 、 V_1 为上三角矩阵，其对应的特征值 $\lambda = \text{diag}(U_1) ./ \text{diag}(V_1)$ 没有按照需要的从小到大的顺序排序。因而需要对 U_1 、 V_1 做进一步处理。 Vec_1 、 Vec_2 为对应的一般化的特征向量。

此时有两个选择：使用 Matlab 外部函数 `qzdiv` 来处理，另外一个 Matlab 内置函数 `ordqz`，两者殊途同归。这两个函数都是 Matlab 编写的，都使用 `qz` 分解输出的上三角矩阵做进一步的处理。`qzdiv` 函数接受 `qz` 分解得到的两个上三角矩阵 U_1 、 V_1 和对应的正交矩阵 Q_1 、 Z_1 ，对特征值进行排序，并把对应的正交矩阵及特征向量进行排序，得到最终的排序后的上三角矩阵 U, V 以及正交矩阵 Q, Z ，`qzdiv` 接受参数 `stake`，表示放置于右下角的特征值应该不小于 `stake`。`ordqz` 函数几乎和 `qzdiv` 一致，只不过两者接受参数的稍微有差异。注意代码中的正交矩阵 Z 即理论推导部分的 Z ，而代码中的正交矩阵 Q 的转置，相当于前述理论推导部分的 Q 。

运行该 Matlab 程序，即可得到系数矩阵 N 和 M ^①：

$$N = \begin{pmatrix} -2.8888 & 1.3400 \\ 5.1356 & -5.1600 \end{pmatrix}, M = \begin{pmatrix} 0.3623 & 0.6003 \\ 0 & 0.9500 \end{pmatrix} \quad (1.1.107)$$

因而：

$$\begin{pmatrix} \hat{C}_t \\ \hat{r}_t \end{pmatrix} = \begin{pmatrix} 2.8888 & -1.3400 \\ -5.1356 & 5.1600 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix}, E_t \begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_{t+1} \end{pmatrix} = \begin{pmatrix} 0.3623 & 0.6003 \\ 0 & 0.9500 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} \quad (1.1.108)$$

注意到

$$\begin{pmatrix} \hat{K}_t \\ \hat{A}_t \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0.9500 \end{pmatrix} \begin{pmatrix} \hat{K}_{t-1} \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \epsilon_t \quad (1.1.109)$$

① 虽然使用了相同的参数校准值，但是由于对资本收益率变量的线性化方法的差异（资本收益率变量属于比率变量，不应该使用对其稳态偏离的百分比表示，而是直接使用对其稳态的偏离作为线性化结果），使得此处得到的结果和 McCallum(2008)(Section 6.8.4, The Generalized Schur Method, P134-139) 并不一致。

因而政策函数为

$$\begin{pmatrix} \hat{C}_t \\ \hat{r}_t \end{pmatrix} = \begin{pmatrix} 2.8888 & -1.2730 \\ -5.1356 & 4.9020 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} -1.3400 \\ 5.1600 \end{pmatrix} \epsilon_t \quad (1.1.110)$$

状态转换方程为

$$\begin{pmatrix} \hat{K}_{t+1} \\ \hat{A}_t \end{pmatrix} = \begin{pmatrix} 0.3623 & 0.5703 \\ 0 & 0.9500 \end{pmatrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_{t-1} \end{pmatrix} + \begin{pmatrix} 0.6003 \\ 1.0000 \end{pmatrix} \epsilon_t \quad (1.1.111)$$

此外，可以验证排序后的结果：系统的特征值 $\lambda = \text{diag}(U) / \text{diag}(V)$ ，确以从小到大的顺序排列：

$$\text{diag}(U) = \begin{pmatrix} -0.9075 \\ 2.7787 \\ 1.4854 \\ 1.2406 \end{pmatrix}, \text{diag}(V) = \begin{pmatrix} -2.5047 \\ 2.9250 \\ 1.3850 \\ 0 \end{pmatrix}, \lambda = \begin{pmatrix} 0.3623 \\ 0.9500 \\ 1.0725 \\ \text{Inf} \end{pmatrix} \quad (1.1.112)$$

其中，Inf 代表无穷大。

1.1.5 待定系数法

本小节简单介绍非线性模型一阶近似线性系统的另外一种解法：待定系数法。该方法由 Uhlig(1999) 提出，并给出了非常详细的介绍^①。

Uhlig(1999) 提出两种方法可供选择：第一种方法未区分内生变量的类型，即未区分内生状态变量和内生非状态变量^②；第二种方法则做了区分。首先来看第一种方法。

1. 内生与外生变量法

假设线性化后的均衡系统可写为

$$0 = E_t(\mathbf{F}x_{t+1} + \mathbf{G}x_t + \mathbf{H}x_{t-1} + \mathbf{L}z_{t+1} + \mathbf{M}z_t) \quad (1.1.113)$$

$$z_{t+1} = \mathbf{N}z_t + \epsilon_{t+1}, E_t \epsilon_{t+1} = 0 \quad (1.1.114)$$

其中， x_t 为内生变量； z_t 为外生变量； \mathbf{F} 、 \mathbf{G} 、 \mathbf{H} 、 \mathbf{L} 、 \mathbf{M} 、 \mathbf{N} 为已知常系数矩阵。此种方法的目标在于寻找具有如下形式的解：

① Uhlig(1999) 不仅给出了具体的例子，还给出了非常详细的 Matlab 代码，以及线性化的具体方法，本书的源代码中包括了 this toolkit，供有兴趣的读者阅读。

② Uhlig(1999) 认为这种方法简单粗暴 (brute force)，但仍基本可行，不过实践中多使用第二种方法，Uhlig(1999) 也认为第二种方法更具灵活性 (with sensitivity)，因为第二种方法对内生变量做了区分，分为内生状态变量和内生前向变量。

$$x_t = Px_{t-1} + Qz_t \quad (1.1.115)$$

其中, P 、 Q 为待定系数矩阵。Uhlig(1999) 将该近似的一阶线性系统的解 (1.1.15), 称为递归均衡运动律 (Recursive Equilibrium Law of Motion)。Uhlig(1999) 指出, 上述线性系统均衡存在的充分必要条件是系数矩阵 P 的所有特征值的绝对值小于 1, 并且如果均衡存在, 须满足^①

$$0 = FP^2 + GP + H \quad (1.1.116)$$

$$V \times \text{vec}(Q) = -\text{vec}(LN + M) \quad (1.1.117)$$

$$V \equiv N^T \otimes F + I_k \otimes (FP + G) \quad (1.1.118)$$

其中, $\text{vec}(\cdot)$ 表示列向量化函数^②, \otimes 为矩阵克氏乘积 (Kronecker Tensor Product)^③。

2. 内生状态变量与前向变量法

假设线性化后的均衡系统可写为

$$0 = Ax_t + Bx_{t-1} + Cy_t + Dz_t \quad (1.1.119)$$

$$0 = E_t (Fx_{t+1} + Gx_t + Hx_{t-1} + Jx_{t+1} + Ky_t + Lz_{t+1} + Mz_t) \quad (1.1.120)$$

$$z_{t+1} = Nz_t + \epsilon_{t+1}, \quad E_t \epsilon_{t+1} = 0 \quad (1.1.121)$$

其中, x_t 为内生的状态变量 (States); y_t 为其他内生变量 (也称为前向变量或 Jumper Variables, 此时可包括静态变量^④); z_t 为外生变量。大写字母均表示系数矩阵, 且已知。目标是寻找如下形式的解:

$$x_t = Px_{t-1} + Qz_t \quad (1.1.122)$$

$$y_t = Rx_{t-1} + Sz_t \quad (1.1.123)$$

其中, P 、 Q 、 R 、 S 均为待决系数矩阵。和第一种方法类似, Uhlig(1999) 指出, 上述线性系统均衡存在的充分必要条件是系数矩阵 P 的所有特征值的绝对值均小于 1。Uhlig(1999) 定理 2 中给出了更为一般的结论, 即当内生状态变量的个数和前向变量的个数 (出现在含有期望算子的方程中, 一般说来, 有多少个前向变量就有多少个期望方程) 不相同系统如何求解, 由于过程较为复杂, 不再详细探讨。此处仅考虑当内生状态变量的个数和前向变量的个数相同的情况^⑤, 此时待决系数矩阵 P 、 Q 、 R 、 S 满

① Uhlig(1999), 定理 1。

② 列向量操作符, 将矩阵的各列按顺序依次叠放在第一列下方, 形成一个列向量。在 Matlab 中如 2×3 矩阵 $X=[1,2;3,4;5,6]$, $\text{vec}(X)=[1 \ 3 \ 5 \ 2 \ 4 \ 6]'$; 或者 $\text{vec}(X)=X(:)$ 。

③ 请查阅 Matlab 对应的函数 `kron`, 在命令行窗口处输入 `help kron` 即可得到相应的帮助。

④ 因此从这个意义上讲, 待定系数法要比 B&K 方法、Schur 方法更为方便, 使用范围更广。

⑤ ...that there are as many expectational equations as there are endogenous state variables.

足如下的方程^①：

$$0 = (F - JC^{-1}A)P^2 - (JC^{-1}B - G + KC^{-1}A)P - KC^{-1}B + H \quad (1.1.124)$$

$$R = -C^{-1}(AP + B) \quad (1.1.125)$$

$$V \times \text{vec}(Q) = \text{vec} \left[(JC^{-1}D - L)N + KC^{-1}D - M \right] \quad (1.1.126)$$

$$S = -C^{-1}(AQ + D) \quad (1.1.127)$$

$$V \equiv N^T \otimes (F - JC^{-1}A) + I_k \otimes (JR + EP + G - KC^{-1}A) \quad (1.1.128)$$

3. 一个例子：不可分劳动模型

此处使用和 Schur 方法中相同的模型即不可分劳动模型，来对比求解结果。使用方法二来求解，由于待定系数法不要求在求解时去除静态的内生变量，因而使用范围更广。

首先，直接借用上一小节线性化后的模型均衡条件：

$$K\hat{K}_{t+1} = Y\hat{Y}_t - C\hat{C}_t + (1 - \delta)K\hat{K}_t \quad (1.1.129)$$

$$\hat{A}_t = \gamma\hat{A}_{t-1} + \epsilon_t \quad (1.1.130)$$

$$0 = \hat{A}_t - \theta\hat{Y}_t + \theta\hat{K}_t - (1 - \theta)\hat{C}_t \quad (1.1.131)$$

$$0 = \hat{K}_t + \hat{r}_t - \hat{Y}_t \quad (1.1.132)$$

$$E_t\hat{C}_{t+1} - \beta E_t\hat{r}_{t+1} = \hat{C}_t \quad (1.1.133)$$

上述均衡条件中不带时间下标的变量表示对应变量的稳态值。此外，把生产函数的线性化方程也添加进均衡条件：

$$\hat{Y}_t = \hat{A}_t + \theta\hat{K}_t + (1 - \theta)\hat{H}_t \quad (1.1.134)$$

其次，定义变量。定义内生状态变量 x_t ，内生的其他变量 y_t 和外生变量 z_t 如下：

$$x_t \equiv \hat{K}_{t+1}, \quad y_t \equiv \begin{pmatrix} \hat{Y}_t \\ \hat{C}_t \\ \hat{H}_t \\ \hat{r}_t \end{pmatrix}, \quad z_t \equiv \hat{A}_t \quad (1.1.135)$$

由于期望方程的个数 (Expectational Equations) 和内生的状态变量 (即资本存量 \hat{K}_{t+1}) 的个数相同，都为 1，因而可以使用方法二求解。于是根据方法二和线性化后的共 6 个均衡条件 (1.1.129)~(1.1.134)，可得系数矩阵如下：

① Uhlig(1999)，定理 2，推论 1。

$$A \equiv \begin{pmatrix} -K \\ 0 \\ 0 \\ 0 \end{pmatrix}, B \equiv \begin{pmatrix} (1-\delta)K \\ \theta \\ 1 \\ \theta \end{pmatrix}, C \equiv \begin{pmatrix} Y & -C & 0 & 0 \\ -\theta & \theta-1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 1-\theta & 0 \end{pmatrix}, D \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \quad (1.1.136)$$

$$F \equiv (0), G \equiv (0), H \equiv (0), J \equiv \begin{pmatrix} 0 \\ 1 \\ 0 \\ -\beta \end{pmatrix}, N \equiv (\gamma), L \equiv (0), M \equiv (0) \quad (1.1.137)$$

最后，编程求解。为了节省篇幅，此处不再列示源代码^①，直接运行该代码，即可得到系数矩阵 P 、 Q 、 R 、 S 如下：

$$P=0.3623, Q=0.6003, R=\begin{pmatrix} -4.1356 \\ 2.8888 \\ -7.0244 \\ -5.1356 \end{pmatrix}, S=\begin{pmatrix} 5.1600 \\ -1.3400 \\ 6.5000 \\ 5.1600 \end{pmatrix} \quad (1.1.138)$$

此结果和 Schur 方法的结果 (1.1.110)、(1.1.111) 完全一致。

1.1.6 线性模型的状态空间表示

当模型中所有均衡条件被线性化之后，并且已经求出政策函数，此时可以根据变量的类型，将模型写成紧致形式即状态空间表示 (State Space Representation)。将模型写成状态空间形式，有助于求解后的各项分析，如模拟、脉冲响应等分析。

假设线性化后的模型可写成：

$$E_t \mathbf{X}_{t+1} = \mathbf{M} \times \mathbf{X}_t \quad (1.1.139)$$

$\begin{matrix} (n+m) \times 1 & & \begin{matrix} (n+m) \times (n+m) \\ (n+m) \times 1 \end{matrix} & & \begin{matrix} (n+m) \times 1 \end{matrix} \end{matrix}$

其中， n 为前向变量 (Forward-looking) 的个数， m 为状态变量 (含预先决定内生变量) 的个数。若定义：

$$\mathbf{X}_t \equiv \begin{pmatrix} X_{1t} \\ \vdots \\ X_{2t} \\ \vdots \\ X_{nt} \end{pmatrix}, \mathbf{M} \equiv \begin{pmatrix} M_{11} & M_{12} \\ \vdots & \vdots \\ M_{21} & M_{22} \\ \vdots & \vdots \\ M_{n1} & M_{n2} \end{pmatrix} \quad (1.1.140)$$

$\begin{matrix} \begin{matrix} m \times 1 \\ n \times 1 \end{matrix} & & \begin{matrix} M_{11} & M_{12} \\ \vdots & \vdots \\ M_{21} & M_{22} \\ \vdots & \vdots \\ M_{n1} & M_{n2} \end{matrix} \\ \begin{matrix} m \times m & m \times n \\ n \times m & n \times n \end{matrix} \end{matrix}$

① Matlab 源代码文件：\Sources\Chap1_DSGE_basics\1.1_1st_logic\Uhlig\Uhlig1999.m。

则 (1.1.139) 可写为

$$E_t \begin{pmatrix} X_{1,t+1} \\ X_{2,t+1} \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \times \begin{pmatrix} X_{1t} \\ X_{2t} \end{pmatrix} \quad (1.1.141)$$

假设在已求出的政策函数

$$X_{2t} = \phi X_{1t} \quad (1.1.142)$$

的情况下, 可将 (1.1.141) 写为

$$E_t X_{1,t+1} = M_{11} X_{1t} + M_{12} X_{2t} = (M_{11} + M_{12} \phi) X_{1t} \equiv \mathbf{A} X_{1t} \quad (1.1.143)$$

$$E_t X_{2,t+1} = M_{21} X_{1t} + M_{22} X_{2t} = (M_{21} + M_{22} \phi) X_{1t} \equiv \mathbf{C} X_{1t} \quad (1.1.144)$$

其中:

$$\mathbf{A} \equiv M_{11} + M_{12} \phi, \mathbf{C} \equiv M_{21} + M_{22} \phi \quad (1.1.145)$$

于是式 (1.1.141) 可进一步表示为

$$E_t \begin{pmatrix} X_{1,t+1} \\ X_{2,t+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \times X_{1t} \quad (1.1.146)$$

由理性预期 (Rational Expectation) 的定义, 加入误差项 ϵ_t , 状态方程 (1.1.143) 可写成

$$X_{1,t+1} = \mathbf{A} X_{1t} + \mathbf{B} \epsilon_{t+1} \quad (1.1.147)$$

或

$$X_{1t} = \mathbf{A} X_{1,t-1} + \mathbf{B} \epsilon_t \quad (1.1.148)$$

其中, 矩阵 \mathbf{B} 依据模型均衡条件确定。由状态方程 (1.1.148), 可将政策函数 (1.1.142) 写成:

$$X_{2t} = \phi X_{1t} = \phi \mathbf{A} X_{1,t-1} + \phi \mathbf{B} \epsilon_t \equiv \mathbf{C} X_{1,t-1} + \mathbf{D} \epsilon_t \quad (1.1.149)$$

其中, $\mathbf{D} = \phi \mathbf{B}$ 。式 (1.1.149) 的最后一个等号, 使用了等价代换: $\mathbf{C} = \phi \mathbf{A}$ 。此等价代换可由式 (1.1.144) 和式 (1.1.149) 得到, 即在式 (1.1.149) 两边取期望, 并和式 (1.1.144) 对比即可。

综上所述, 可得求解后, DSGE 模型的一阶近似线性系统的状态空间表示如下:

$$X_{2t} = \mathbf{C} X_{1,t-1} + \mathbf{D} \epsilon_t \quad (1.1.150)$$

$$X_{1t} = \mathbf{A} X_{1,t-1} + \mathbf{B} \epsilon_t \quad (1.1.151)$$

方程 (1.1.150) 可视为观测方程 (Observation Equation 或 Measurement Equation), 状态变量的 AR(1) 过程 (1.1.151) 可视为状态方程 (State Equation 或 Transition Equation)。

当 DSGE 模型一阶近似解表示成式 (1.1.150) 和式 (1.1.151) 的状态空间形式时, 前向变量 X_{2t} 可以表示 VAR(∞) 的形式。假设矩阵 D 可逆, 首先从观测方程 (1.1.150) 可知:

$$\epsilon_t = D^{-1}X_{2t} - D^{-1}CX_{1,t-1} \quad (1.1.152)$$

将变形后的观测方程 (1.1.152) 代入状态转换方程 (1.1.151) 中, 可知:

$$\begin{aligned} X_{1t} &= (A - BD^{-1}C)X_{1,t-1} + BD^{-1}X_{2t} \\ &\equiv NX_{1,t-1} + BD^{-1}X_{2t}, \quad N \equiv A - BD^{-1}C \end{aligned} \quad (1.1.153)$$

假设矩阵 N 的特征值的模值均小于 1, 那么式 (1.1.153) 前向迭代可得:

$$X_{1t} \equiv NX_{1,t-1} + BD^{-1}X_{2t} = BD^{-1}X_{2t} + NBD^{-1}X_{2,t-1} + N^2BD^{-1}X_{2,t-2} + \dots \quad (1.1.154)$$

将 (1.1.154) 代入观测方程 (1.1.150), 从而可得 X_{2t} 的 VAR(∞) 表达形式:

$$X_{2t} = G_1X_{2,t-1} + G_2X_{2,t-2} + \dots + u_t, \quad u_t \equiv D\epsilon_t, \quad G_i \equiv CN^{i-1}BD^{-1}, \quad i=1, 2, \dots \quad (1.1.155)$$

综合可知, 只有当矩阵 D 可逆且矩阵 N 的特征值的模值均小于 1 的情况下, DSGE 模型一阶近似解才能表示 VAR 的形式, 此时称该 DSGE 模型为可逆的 (Invertible); 否则称之为不可逆 (Non-invertible, Sims, 2012)。实际上该 VAR 模型为结构 VAR, 即 SVAR。

1.1.7 脉冲响应和随机模拟

本小节将通过手动编程, 利用一阶近似系统的状态空间表示来计算各变量的脉冲响应, 并进行随机模拟。通过本小节的学习, 特别是通过 Matlab 编程, 将会使得读者对脉冲响应计算和随机模拟分析有一个初步的认识, 也为后续深入学习 Dynare 中的脉冲响应和随机模拟分析奠定基础。此处将使用本书“1.1.3 B&K 方法”一节中的例子来进行编程。

首先, 手动编程必须要清楚各变量的排序。在上述几小节的分析中, 采取了先状态变量、后前向变量的形式:

$$\text{位置顺序} \equiv \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{pmatrix} \hat{K}_t \\ \hat{A}_t \\ \hat{C}_t \end{pmatrix} \quad (1.1.156)$$

1. 脉冲响应分析

所谓脉冲响应 (Impulse Response), 直观上说是指当某一个外生冲击在某一期发生, 而后所有期该冲击都消失时各变量的变化路径 (暗含其他外生冲击都为零, 不仅在该冲击发生的当期, 而且以后所有期都如此)。在“6.2 脉冲响应函数和自定义编程”一节中将给出脉冲响应的数学定义, 并做进一步探讨。根据“1.1.3 B&K 方法”一节中的例子,

此处考察在初期技术冲击发生一单位正向变化，此后各期为零时各变量的脉冲响应。

计算脉冲响应有如下两种方法。

第一种方法是使用上述脉冲响应的直观定义和政策函数来实现(源代码3)。在代码中首先声明了用于存储资本、技术和消费的脉冲响应数组(注意各变量均为线性化变量)，并定义了技术冲击的脉冲响应在第一期为单位1^①，资本存量为0，即意味着资本存量位于稳态；然后对消费脉冲响应的第一期值使用政策函数赋值；接着使用式(1.1.40)进行循环计算；最后利用式(1.1.67)计算产出、投资和资本收益率的脉冲响应，并画出图形。

源代码3 手动编程实现脉冲响应分析(基于IRF的定义)^②

```
%% do the IRF by Definition ,H=30
%number of simulation; if you use great number,like H=1500, you will get
strange results, but for small number it works
H = 30;
%n+m rows correspondent to the number of variables
%in B&K example, m=2, n=1;
%position index: (capital, technology,consumption)=(1,2,3)
IRF = zeros(n+m,H);
%setting up the first period
IRF(2,1) = 1; %one unit shock to technology shock
IRF(1,1) = 0; % for capital, starts from steady state
% for the controls, actually only for consumption, n=1, the policy
function
IRF(m+1:m+n,1) = pol*IRF(1:m,1);
% by definition
for ii=2:H
    IRF(:,ii) = BA*IRF(:,ii-1);
end
%including another two variables: output and investment
IRF2 = zeros(3,H);
for jj=1:H
    IRF2(1,jj) = IRF(2,jj) + alpha*IRF(1,jj);%for output
    IRF2(2,jj) = ys/is*IRF2(1,jj) - cs/is*IRF(3,jj);%for investment
    IRF2(3,jj) = IRF2(1,jj)- IRF(1,jj);%for capital rental rate
end
%plot the IRF
figure(1)
title('IRF of Technology Shock')
subplot(3,2,1)
plot(IRF(1,:), '-bo', 'Linewidth', 1, 'MarkerSize', 4)
title('Capital')
……(此处省略, 参看源代码)
```

① 注意在实际计算时，应将其设定为技术冲击的标准差，此处仅为示例，将标准差设定为1。

② Matlab 源代码：\Sources\Chap1_DSGE_basics\1.1_1st_logic\1.1.7_IRF_Simulation\myIRF.m, 模块 (cell): do the IRF by Definition。

基于 IRF 的定义，分别计算 30 期和 1 500 期脉冲响应，如图 1.2 和图 1.3 所示。图 1.2 中的脉冲响应图的形状是正确的。

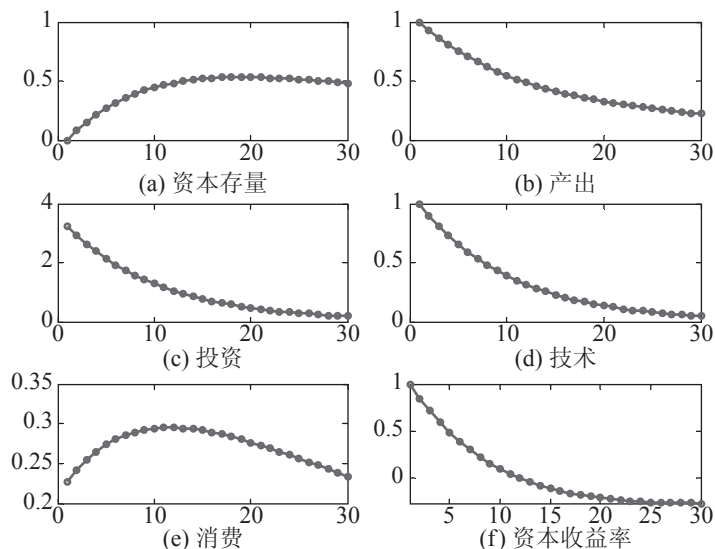


图 1.2 一单位正向技术冲击下各变量的脉冲响应图 (模拟 30 期)

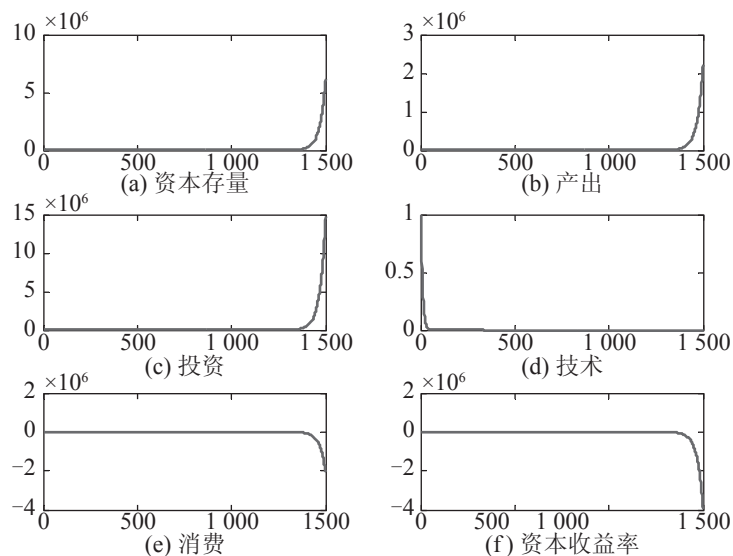


图 1.3 一单位正向技术冲击下各变量的脉冲响应图 (模拟 1 500 期)

然而出乎意料的是，图 1.3 中主要变量的脉冲响应呈现出发散状态 (即趋于无穷大)。

这到底是为什么呢？假设从某一状态 $(\hat{K}_0, \hat{A}_0)^\top$ 出发，根据政策函数：

$$\hat{C}_0 = \phi \begin{pmatrix} \hat{K}_0 \\ \hat{A}_0 \end{pmatrix} \quad (1.1.157)$$

再根据原系统的定义 (1.1.40)：

$$\begin{pmatrix} \hat{K}_t \\ \hat{A}_t \\ \hat{C}_t \end{pmatrix} = (B^{-1}A)^t \begin{pmatrix} \hat{K}_0 \\ \hat{A}_0 \\ \hat{C}_0 \end{pmatrix} = (B^{-1}A)^t \begin{pmatrix} \hat{K}_0 \\ \hat{A}_0 \\ \phi(\hat{K}_0, \hat{A}_0)^\top \end{pmatrix} = (B^{-1}A)^t \begin{pmatrix} I_2 \\ \phi \end{pmatrix} \begin{pmatrix} \hat{K}_0 \\ \hat{A}_0 \end{pmatrix} \quad (1.1.158)$$

其中， I_2 表示二阶单位矩阵。由于矩阵 $B^{-1}A$ 中含有大于 1 的特征值，并且政策函数系数矩阵存在数值计算误差，当模拟期数足够大，大于 1 的特征值会使得计算误差无限放大，从而出现趋于无穷大的情况。虽然上述计算方法在理论上对的，但在实际计算时却很难行得通，因此出现如图 1.3 所示的情况，即尾部趋于无穷大。

为了避免出现这种情况，需要对此非稳定特征值进行处理，此时需要用到脉冲响应计算的第二种方法，即使用状态空间表示的均衡解进行脉冲响应计算，以稳妥处理非稳定特征值问题。

如果使用状态空间解表示形式即式 (1.1.146)，此时 AR(1) 系数矩阵仅仅依赖于状态变量 x_t ，因为状态变量的 AR(1) 系数是“稳定”的 (By Construction)，因而能够处理非稳定特征值问题。具体实现方式，参考源代码 4。

源代码4 手动编程实现脉冲响应分析(基于状态空间表示)^①

```
%no matter how large of H, the IRFs are correct, by construction is
stable
H = 100;

%n+m rows correspondent to the number of variables
IRF = zeros(n+m,H);

%position index: (capital, technology, consumption)=(1,2,3)
%setting up the first period, the initial states
IRF(1,1) = 0; % for capital, starts from steady state
IRF(2,1) = 1; %one unit shock to technology shock

%the initial controls
% for the controls, actually only for consumption, n=1, the policy
function
IRF(m+1:m+n,1) = pol*IRF(1:m,1);
```

① Matlab 源代码：\Sources\Chap1_DSGE_basics\1.1_1st_logic\1.1.7_IRF_Simulation\myIRF.m, cell: do the IRF by state space representation。感兴趣的读者可自行验证，无论模拟多少期，都不会出现发散的情况。

```

%state sapce, by construction is stable
M1 = BA(1:m,:); %the first m rows
M2 = BA(m+1:n+m,:); % the last n rows;
BA11=M1(:,1:m);
BA12=M1(:,m+1:n+m);
A = BA11 + BA12*pol;

for ii=2:H
    IRF(1:m,ii)=A*IRF(1:m,ii-1);%transition equation
    IRF(m+1:n+m,ii)=pol*A*IRF(1:m,ii-1); %measurement equation
end

%including another two variables
……(此处省略, 同方法一)

%plot the IRFs
figure(1)
title('IRF of Technology Shock')
subplot(3,2,1)
plot(IRF(1,:), ' -b' , ' Linewidth' ,2)
title('Capital')
……(此处省略, 参看源代码)
    
```

可以发现，使用状态空间解表示形式，无论模拟多少期，都不会出现发散状态。

2. 随机模拟分析

所谓随机模拟，顾名思义就是基于随机外生冲击的样本对内生变量进行数值模拟。在模拟周期内，每期都有外生冲击发生。从实现技术上讲，每期都会随机抽样^①，实现随机外生冲击。从 Matlab 编程实践来看，这种外生冲击往往就是从白噪声 (White Noise) 过程中抽取的样本，一般使用内置 `randn` 函数，即抽取方差为 1，均值为 0 的正态随机变量的样本。

本部分首先列示出随机模拟的 Matlab 实现，然后基于随机模拟考察模拟的 (HP 滤波后) 变量的波动特征。需要注意的是，本部分的模拟都是针对线性化后的变量，而非水平变量的模拟，即离差形式 (Deviation Form) 变量的模拟，后续的分析也是基于 HP 滤波后离差形式变量进行的。当然可以从离差形式的变量中提取水平变量进行相关分析。本书“3.9 随机模拟分析：stoch_simul”一节中会进一步介绍关于随机模拟的内容。

源代码 5 给出了基于状态空间表示的均衡解的随机模拟实现 (B&K 方法中的例子)。模拟首先抽取了标准差为 0.1，均值为 0 的 200 个白噪声样本。然后通过构建状态空间表示状态转换方程中的 AR(1) 系数矩阵 A 。注意此时状态变量有两个，第一个是资本存量，第二个是技术变量，两者具有位置顺序，在编写程序时不能混淆。接着构建了扩展的矩

① 在编程时，为了方便，往往一次抽取所有模拟期内的外生冲击的样本。

阵 C ，通过观测方程对消费、产出和投资进行模拟。最后使用了 HP 滤波进行统计分析。HP 滤波的介绍会在本书“1.3.4 HP 滤波的基本逻辑”一节中做详细分析。

源代码5 手动编程实现随机模拟(基于状态空间表示的模拟)^①

```

%% do simulation of the model by state space representation
randn('state',1234567); %ensure the replication of simulation results
T = 200; %number of data to be simulated

%sampling from standard normal distribution, i.e. the technology shock;
e = 0.1*randn(1,T);
%state sapce, by construction is stable
M1 = BA(1:m,:); %the first m rows, in this example, m=2
M2 = BA(m+1:n+m,:); % the last n rows, in this example, n=1
BA11=M1(:,1:m);
BA12=M1(:,m+1:n+m);
A = BA11 + BA12*pol; %pol, coefficient of the policy function
C = pol;
Tech=[0;1];
% [consumption, output ,investment]
% this matrix corresponds to eq., coefficient matrix in text
C = [C;alpha 1;(ys/is)*alpha - (cs/is)*pol(1,1) (ys/is)*1 - (cs/
is)*pol(1,2)];
s = zeros(2,T); % states, [capital , technology]
c = zeros(3,T); % controls, [consumption, output ,investment]

%s(:,1) = 0; means that we starts from steady states
for j = 2:T
    s(:,j) = A*s(:,j-1) + Tech*e(1,j); %the state equations
end
for j = 1:T
    c(:,j) = C*s(:,j); %the extended policy function, states map to
controls
end
lambda = 1600; % do hp filtering
%*t, trend part; *c, cycle part;
% hp_filter函数为自定义函数, 请参考1.3.4 HP滤波的基本逻辑
[yt,yc] = hp_filter(c(2,:)',lambda);
[ct,cc] = hp_filter(c(1,:)',lambda);
[it,ic] = hp_filter(c(3,:)',lambda);

% find out the simulated moments:standard deviations
% from cyclic part of the data
stdy = std(yc); stdc = std(cc); stdi = std(ic);

% relative volatility
Rc = stdc/stdy; Ri = stdi/stdy;

```

① Matlab 源代码: \Sources\Chap1_DSGE_basics\1.1_1st_logic\1.1.7_IRF_Simulation\mySimulation.m.

图 1.4 给出了消费、产出、投资和技术冲击的随机模拟路径图。很明显，技术变量是平稳的 AR(1) 过程，产出和消费的波动低于投资冲击，而消费的波动低于产出。

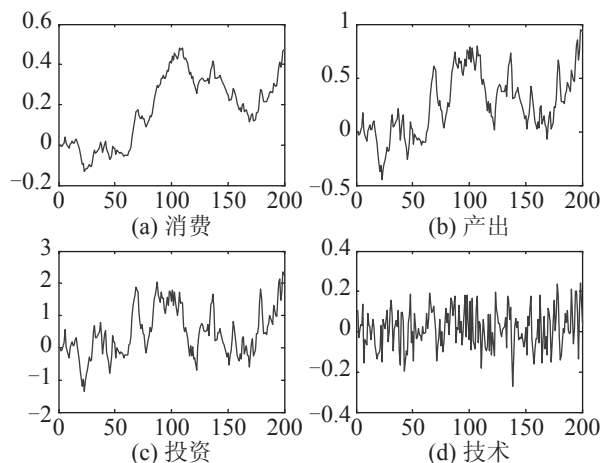


图 1.4 随机模拟 (离差形式, Deviation Form)

图 1.5 给出了当技术变量 AR(1) 过程的持续性参数 (ρ , 也可称为平滑参数) 变化时, 产出、消费和投资的波动变化 (即标准差)。持续性参数 ρ 的变化范围是 0.1~0.96。

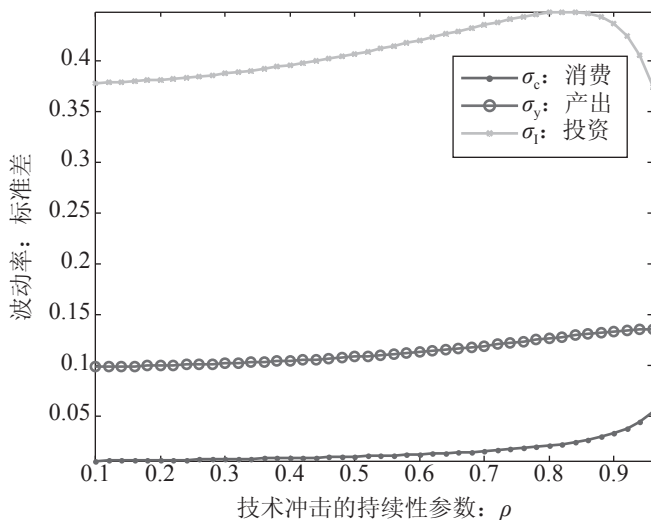


图 1.5 技术变量的持续参数变化对 HP 滤波后的产出、消费和投资波动的影响^①

① Matlab 源代码: \Sources\Chap1_DSGE_basics\1.1_1st_logic\1.1.7_IRF_Simulation\persistence_volatility_main.m。直接运行即可得到该图, 该函数调用 find_Persistence_Volatility.m。

一个非常有趣的现象是消费的波动随持续性参数的增长而增加，投资的波动则先增后减。更进一步，图 1.6 表明消费的相对波动（相对于产出）是持续性参数的增函数，投资相对波动（相对于产出）是持续性参数的减函数。这是为什么呢？其实这可由著名的永久收入假说 (Permanent Income Hypothesis^①) 来解释。

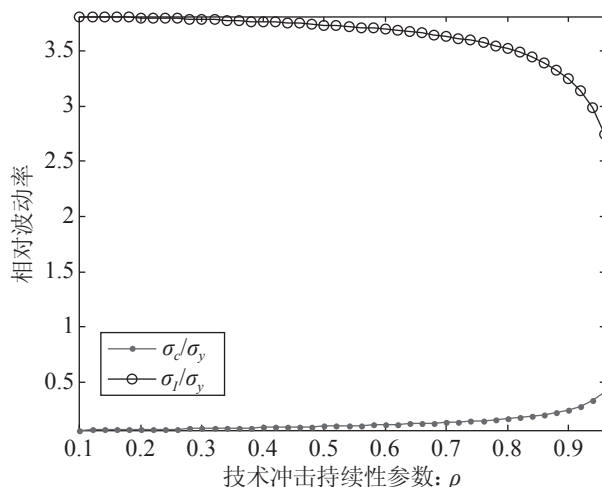


图 1.6 技术变量 AR(1) 过程的持续性参数变化对消费和投资相对波动的影响^②

永久收入假说认为，消费者的当期消费行为，不仅受到当期收入的影响，更加受到其期望的、长期收入的影响，即永久性收入的影响。换句话说，永久性收入的改变能显著改变消费者当期消费行为。

持续性参数衡量了技术冲击持续的程度。持续性参数越大，外生技术冲击影响的程度越大，时间越长，因而产出持续增加时间越长，进而预期收入增长的时间变长。当持续性参数较大时，面对一个外生技术冲击，消费者会预期其未来收入将保持增长，因而其消费行为将发生变化，进而产生较大的消费波动。相反，当持续性参数较小时，技术冲击的影响很快消失，因而对预期收入影响较小，不能显著影响消费行为，消费波动较小。这就解释了为什么消费（相对）的波动是持续性参数的增函数。

① 最早由 Friedman 提出。Friedman, Milton (1957). *The Permanent Income Hypothesis. A Theory of the Consumption Function*. Princeton University Press. ISBN 0-691-04182-2.

Hall(1978) 则较早地对此理论做了验证。Hall, Robert E. (1978). *Stochastic Implications of the Life Cycle-Permanent Income Hypothesis: Theory and Evidence*. Journal of Political Economy. 86 (6): 971-87. doi:10.1086/260724.

② Matlab 源代码：\Sources\Chap1_DSGE_basics\1.1_1st_logic\1.1.7_IRF_Simulation\persistence_volatility_main.m。直接运行即可得到该图，该函数调用 find_Persistence_Volatility.m。

那为什么投资的相对波动是持续性参数的减函数呢？事实上，根据线性化后的资源约束方程

$$\hat{Y}_t = \frac{C}{Y} \hat{C}_t + \frac{I}{Y} \hat{I}_t \quad (1.1.159)$$

可知，产出对稳态的偏离是消费和投资对其稳态偏离的加权平均。从图 1.5 可见，产出的波动受持续性参数变化的影响不大，其波动率均值为 0.113 4，最小值为 0.099 1，最大值为 0.136 2，可见产出的波动受持续性参数变化不敏感。其实这并不意外，因为持续性参数的水平效应较强，而波动效应较弱，因而可基本认为产出的波动为常数。从这个意义上说，根据式 (1.1.159)，消费对稳态偏离的增加，往往伴随着投资对稳态偏离的减少，因而投资波动会呈现下降趋势。

图 1.6 中，消费的相对波动的均值为 0.130 7，投资的相对波动的均值为 3.635 6。这基本都和数据吻合。即消费的波动一般较小，小于产出，而投资的波动很大，往往是产出的数倍。

1.2 DSGE 高阶求解：Dynare 的求解逻辑

本节将围绕 Schmitt-Grohé & Uribe(2004) 展开论述。Schmitt-Grohé & Uribe(2004)^① 提出了如何对离散时间 DSGE 模型进行一阶和二阶基于扰动项的泰勒近似方法，给出了 Matlab 实现，有重要的理论贡献，并最终被 Dynare 采用作为一阶和二阶求解的标准算法。

由于三阶和三阶以上近似的扰动算法过于复杂，因此不在本书的讨论范围内。目前而言，Dynare 仅仅实现了三阶近似。这一工作基于 Andreasen, Fernández-Villaverde & Rubio-Ramírez (2013)。该论文的技术附录多达 277 页，可见技术处理的复杂度之高，有兴趣的读者可详细阅读^②。此外，作为 Dynare 平台的一个副产品，Dynare++ 可以实现更高阶的求解，比如四阶、五阶甚至更高阶，有兴趣的读者可查看 Dynare 安装目录下的帮助文档。

1.2.1 基于扰动项的泰勒近似方法

假设模型的均衡可由 (1.1.1) 定义：

$$E_t f(y_{t+1}, y_t, y_{t-1}, u_t) = 0$$

① 这是一篇非常重要的论文，推荐大家仔细阅读。

② 可去 Fernández-Villaverde 的个人网页下载最新版：<http://economics.sas.upenn.edu/~jesusfv/research.html>。

其中, y_t 为内生变量, u_t 为外生冲击。此处论述为了和前节保持一致, 未采用 Schmitt-Grohé & Uribe(2004) 引入的符号, 即上式中未对内生变量做区分, 但基本求解算法是一致的。

Schmitt-Grohé & Uribe(2004) 使用了扰动算法^①, 而非值函数递归方法 (VFI, Value Function Iteration)。扰动算法的基本想法是引入一个随机调节系数 σ (Stochastic Scale Variable), 用于调节外生冲击标准差的大小, 并作为政策函数的一个参变量 (Argument)。在进行泰勒近似时, 将 σ 看作变量进行一阶和二阶求导。该方法由 Fleming(1971) 首次引入, 后来被广泛应用于经济模型求解中^②。在求解时同时引入一个辅助随机变量 ϵ_t (Auxiliary Random Variable), 使得

$$u_t = \sigma \epsilon_t \quad (1.2.1)$$

并且辅助随机变量满足

$$E(\epsilon_t) = 0, \quad E(\epsilon_t \epsilon_t') = \Sigma_\epsilon, \quad E(\epsilon_t \epsilon_s') = 0, t \neq s, \quad \Sigma_u = \sigma^2 \Sigma_\epsilon \quad (1.2.2)$$

此时, 模型的解 (1.1.3), 即政策函数可写为

$$y_t = g(y_{t-1}, u_t, \sigma) \quad (1.2.3)$$

这样做的好处在于, 来能够通过调节系数 σ 的变化来改变模型的类型: 当 $\sigma = 0$ 时, 模型不存在不确定性, 从而为确定性模型 (Deterministic); 当 $\sigma > 0$ 时, 模型为随机模型 (Stochastic), 当 $\sigma \rightarrow 0^+$ 时, 随机模型趋于确定性模拟, 也就是说确定性模型是随机模型的极限情况。当 $\sigma = 1$ 时, 模型不受调节系数的影响, 为原来的模型。因此, 来调节系数 σ 也被称为扰动项或扰动变量, 这也是扰动算法名称的由来。

如果 (1.2.3) 为已知, 前向迭代一期, 即

$$y_{t+1} = g(y_t, u_{t+1}, \sigma) = g[g(y_{t-1}, u_t, \sigma), u_{t+1}, \sigma]$$

从而模型均衡 (1.1.1) 可进一步写成

$$F(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \equiv f \left(\underbrace{g(g(y_{t-1}, u_t, \sigma), \sigma \epsilon_{t+1}, \sigma)}_{y_{t+1}}, \underbrace{g(y_{t-1}, u_t, \sigma)}_{y_t}, y_{t-1}, \sigma \right) \quad (1.2.4)$$

$$E_t \{ F(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \} = 0 \quad (1.2.5)$$

接下来, 简略介绍 Dynare 如何实现一阶和二阶求解^③。Dynare 的一阶、二阶和

① 该文还给出了两个不错的 notes, 感兴趣的读者可自行阅读, 见参考文献目录: \References\chap1\perturbationmethods_I.pdf 和 \perturbationmethods_II.pdf。

② 如 Judd, K. L. *Numerical methods in economics* [M]. Cambridge, MA, MIT Press, 1998.

③ 在 Dynare 的实现上, 由位于安装目录 matlab 文件夹下的 m 文件: stochastic_solvers.m 实现一阶和二阶求解。

三阶近似算法都是基于扰动项的泰勒近似法。此处着重介绍一阶和二阶泰勒近似实现逻辑。

1. Dynare 一阶求解

前文已经提及，Dynare 一阶求解使用的是一阶泰勒近似法，其本质上和上一节介绍的 B&K 方法、Schur 方法是一致的。因而其本质任务同样是根据解的一阶线性表达式 (1.1.4) 来寻求系数 g_y, g_u 。本节在更一般的形式下，介绍 Dynare 如何进行一阶近似求解。具体说来，近似展开是对非线性系统式 (1.2.4) 或式 (1.2.5) 基于确定性稳态值 $(\bar{y}, 0, 0)$ ^① 的一阶泰勒展开 (1st Order)。关于 y_{t-1}, u_t, σ 这三个变量的一阶求导，可得

$$\begin{aligned}
 0 &= E_t \left\{ F^{(1)}(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \right\} \\
 &= E_t \left\{ \begin{aligned} &f(\bar{y}, \bar{y}, \bar{y}, 0) \\ &+ \frac{\partial f}{\partial y_+} \left(\frac{\partial g}{\partial y} \left(\frac{\partial g}{\partial y_-} \hat{y} + \frac{\partial g}{\partial u} u + \frac{\partial g}{\partial \sigma} \sigma \right) + \frac{\partial g}{\partial u_+} \sigma \epsilon_+ + \frac{\partial g}{\partial \sigma} \sigma \right) \\ &+ \frac{\partial f}{\partial y} \left(\frac{\partial g}{\partial y_-} \hat{y} + \frac{\partial g}{\partial u} u + \frac{\partial g}{\partial \sigma} \sigma \right) + \frac{\partial f}{\partial y_-} \hat{y} + \frac{\partial f}{\partial u} u \end{aligned} \right\} \quad (1.2.6)
 \end{aligned}$$

其中，符号中的加号“+”代表 $t+1$ 期变量，“-”代表 $t-1$ 期变量，未标号变量代表当期； $\hat{y} \equiv y_{t-1} - \bar{y}$ ，外生变量和随机调节系数的线性化为其本身；各偏导数均在稳态 $(\bar{y}, 0, 0)$ 处取值，也就是说在确定性稳态处，各偏导也是确定的，即常数。注意到 $E_t \epsilon_t = 0$ ，合并同类项，去除期望算子，并简化符号，则式 (1.2.6) 可写为

$$(f_{y+} g_y g_y + f_y g_y + f_{y-}) \hat{y} + (f_{y+} g_y g_u + f_y g_u + f_u) u + (f_{y+} g_y + f_y + f_{y+}) g_\sigma \sigma = 0 \quad (1.2.7)$$

其中

$$\begin{aligned}
 f_{y+} &\equiv \frac{\partial f}{\partial y_+} \Big|_{(\bar{y}, \bar{y}, \bar{y}, 0)}, & f_y &\equiv \frac{\partial f}{\partial y} \Big|_{(\bar{y}, \bar{y}, \bar{y}, 0)}, & f_{y-} &\equiv \frac{\partial f}{\partial y_-} \Big|_{(\bar{y}, \bar{y}, \bar{y}, 0)}, \\
 g_y &\equiv \frac{\partial g}{\partial y} \Big|_{(\bar{y}, 0, 0)} = \frac{\partial g}{\partial y_-} \Big|_{(\bar{y}, 0, 0)}, & g_u &\equiv \frac{\partial g}{\partial u} \Big|_{(\bar{y}, 0, 0)}, & g_\sigma &\equiv \frac{\partial g}{\partial \sigma} \Big|_{(\bar{y}, 0, 0)}
 \end{aligned}$$

由于政策函数解析式未知，因而 g_y, g_u 以及 g_σ 均为未知量。根据式 (1.1.4)，接下来的任务是寻求系数 g_y, g_u 以及 g_σ 。式 (1.2.7) 为 0，意味着每项系数为 0：

① 内生变量稳态值的定义请参考第 3 节。外生冲击稳态时为 0，随机调节系数 σ 稳态时也为 0。

$$f_{y+}g_y g_y + f_y g_y + f_{y-} = 0 \quad (1.2.8)$$

$$f_{y+}g_y g_u + f_y g_u + f_u = 0 \quad (1.2.9)$$

$$(f_{y+}g_y + f_y + f_{y+})g_\sigma = 0 \quad (1.2.10)$$

从形式上看,三方程,三未知量,因而上述系统可解。首先来看式(1.2.9)和式(1.2.10)。从形式上看,这两个方程是关于 g_u 和 g_σ 的线性齐次方程,即二元一次线性方程,因而求解简单。从式(1.2.9)中,可得^①:

$$g_u = -(f_{y+}g_y + f_y)^{-1} f_u \quad (1.2.11)$$

从式(1.2.10)中可得(一般情况下括号内的项不为0):

$$g_\sigma = 0 \quad (1.2.12)$$

这就是一阶近似的确定性等价(Certainty Equivalence)性质。也就是说,一阶近似中随机调节变量的系数为0,即不依赖于随机调节变量。

其次,从式(1.2.8)中求解 g_y 。这是一个关于 g_y 的二次矩阵方程。从理论上可以求解,特别是当 g_y 为一维时,直接使用一元二次方程的求根公式即可。Dynare 使用了状态空间表示的形式,即式(1.1.39)来求解。令式(1.2.7)中第一项为0,经过简单的变形可得

$$\begin{pmatrix} 0 & f_{y+} \\ I & 0 \end{pmatrix} \begin{pmatrix} I \\ g_y \end{pmatrix} g_y \hat{y} = \begin{pmatrix} -f_{y-} & -f_y \\ 0 & I \end{pmatrix} \begin{pmatrix} I \\ g_y \end{pmatrix} \hat{y} \quad (1.2.13)$$

其中, I 表示单位矩阵(Identity Matrix)。若记:

$$A \equiv \begin{pmatrix} -f_{y-} & -f_y \\ 0 & I \end{pmatrix}, B \equiv \begin{pmatrix} 0 & f_{y+} \\ I & 0 \end{pmatrix} \quad (1.2.14)$$

则(1.2.13)可写为

$$B \begin{pmatrix} I \\ g_y \end{pmatrix} g_y \hat{y} = A \begin{pmatrix} I \\ g_y \end{pmatrix} y \quad (1.2.15)$$

或(省略期望算子)

$$Bx_{t+1} = Ax_t, \quad x_t \equiv \begin{pmatrix} y_{t-1} - \bar{y} \\ y_t - \bar{y} \end{pmatrix} \quad (1.2.16)$$

接下来对矩阵 A, B 进行 QZ 分解,逻辑和第一节中介绍的 B&K 方法、Schur 方法是一致的,从而可求出 g_y 。

① 可证明在稳定均衡存在的条件下,矩阵 $f_y + g_y + f_y$ 可逆。可参考 Hong Lan and Alexander Meyer-Gohde, 2012, *Existence and Uniqueness of Perturbation Solutions to DSGE models*, SFB649 Discussion Papers, Humboldt University.

在完成 g_y, g_u 的求解后，即完成了政策函数的求解，一阶线性近似解 (1.1.4) 此时已知。内生变量的一阶矩和二阶矩分别为

$$E(y_t) = \bar{y} \quad (1.2.17)$$

$$\Sigma_y \equiv \text{var}(y_t), \Sigma_y = g_y \Sigma_y g_y' + \sigma^2 g_u \Sigma_u g_u' \quad (1.2.18)$$

(1.2.17) 揭示了一阶近似的一个重要特征：无条件均值等于稳态。更进一步说，期望值和稳态相同，平均而言外生冲击不起作用，即确定性等价的另外一种表现形式，后续会详细说明。关于二阶矩即方差，式 (1.2.18) 是关于方差矩阵 Σ_y 的李雅普诺夫 (Lyapunov) 方程，需要使用特定的算法才能求解出来，此处不再详细说明。

至此，完成了 Dynare 一阶求解实现的介绍。

2. Dynare 二阶求解

Dynare 的二阶求解逻辑和一阶求解一致，所不同的是对非线性系统式 (1.2.4) 或式 (1.2.5) 基于确定性稳态值 $(\bar{y}, 0, 0)$ 的二阶泰勒展开 (2nd Order)。关于 y_{t-1}, u_t, σ 这三个变量二阶求导可得：

$$\begin{aligned} 0 &= E_t \left\{ F^{(2)}(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \right\} \\ &= E_t \left\{ \begin{array}{l} F^{(1)}(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \\ + \frac{1}{2} \left(F_{y-y-} (\hat{y} \otimes \hat{y}) + F_{uu} (u \otimes u) + F_{u+u+} \sigma^2 (\epsilon_+ \otimes \epsilon_+) + F_{\sigma\sigma} \sigma^2 \right) \\ + F_{y-u} (\hat{y} \otimes u) + F_{y-u+} (\hat{y} \otimes \sigma \epsilon_+) + F_{y-\sigma} \hat{y} \sigma + F_{u+} (u \otimes \sigma \epsilon_+) \\ + F_{u\sigma} u \sigma + F_{u+\sigma} \sigma^2 \epsilon_+ \end{array} \right\} \quad (1.2.19) \\ &= E_t \left\{ F^{(1)}(y_{t-1}, u_t, \epsilon_{t+1}, \sigma) \right\} \\ &\quad + \frac{1}{2} \left(F_{y-y-} (\hat{y} \otimes \hat{y}) + F_{uu} (u \otimes u) + F_{u+u+} \sigma^2 \Sigma_{\otimes \epsilon} + F_{\sigma\sigma} \sigma^2 \right) \\ &\quad + F_{y-u} (\hat{y} \otimes u) + F_{y-\sigma} \hat{y} \sigma + F_{u\sigma} u \sigma \end{aligned}$$

其中， $\Sigma_{\otimes \epsilon} \equiv E(\epsilon_+ \otimes \epsilon_+)$ ，第三个等号使用了 $E_t(\epsilon_+) = 0$ ，各项系数具体如下：

$$F_{y-y-} \equiv f_{y+} \left(g_{yy} (g_y \otimes g_y) + g_y g_{yy} \right) + f_y g_{yy} + T.I.P. \quad (1.2.20)$$

$$F_{y-u} \equiv f_{y+} \left(g_{yy} (g_y \otimes g_u) + g_y g_{yu} \right) + f_y g_{yu} + T.I.P. \quad (1.2.21)$$

$$F_{uu} \equiv f_{y+} \left(g_{yy} (g_u \otimes g_u) + g_y g_{uu} \right) + f_y g_{uu} + T.I.P. \quad (1.2.22)$$

$$F_{u+u+} \equiv f_{y+y+} (g_u \otimes g_u) + f_{y+} g_{uu} \quad (1.2.23)$$

$$F_{\sigma\sigma} \equiv f_{y+} (g_{\sigma\sigma} + g_y g_{\sigma\sigma}) + f_y g_{\sigma\sigma} \quad (1.2.24)$$

$$F_{y\sigma} \equiv f_{y+} g_y g_{y\sigma} + f_y g_{y\sigma} \quad (1.2.25)$$

$$F_{u\sigma} \equiv f_{y+} g_y g_{u\sigma} + f_y g_{u\sigma} \quad (1.2.26)$$

其中, $T.I.P.$ 表示未包括政策函数 g 的二阶导数的项, 即只包含一阶导数的项。

接下来, 求解由政策函数 g 的二阶项的 6 个系数矩阵 g_{yy} 、 g_{yu} 、 $g_{y\sigma}$ 、 g_{uu} 、 $g_{u\sigma}$ 、 $g_{\sigma\sigma}$ 以获取二阶近似解, 注意此时假设一次项各系数矩阵均已知。首先, 令式 (1.2.20) 为 0, 并整理可得:

$$(f_{y+} g_y + f_y) g_{yy} + f_{y+} g_{yy} (g_y \otimes g_y) = -T.I.P. \quad (1.2.27)$$

上式是关于 g_{yy} 的线性方程, 很遗憾不能直接简单求出, 这是著名的 Sylvester 方程, 需要一定的算法才能求出, 此处不再详细介绍, 感兴趣的读者可查阅相关资料或阅读 Dynare 相关源代码。

其次, 令式 (1.2.21) 和式 (1.2.22) 分别为 0, 可很容易求出 g_{yu} 和 g_{uu} 。

再次, 令式 (1.2.23) 和式 (1.2.24) 之和为 0, 从而容易求解 $g_{\sigma\sigma}$ 。

最后, 从式 (1.2.25) 和式 (1.2.26) 中可知 $g_{y\sigma} = 0$, $g_{u\sigma} = 0$, 也就是说, 二阶近似解中不存在随机调节变量的一次项。此时二阶近似解可写为 (令 $\sigma=1$)

$$y_t = \bar{y} + \frac{1}{2} g_{\sigma\sigma} \sigma^2 + g_y \hat{y}_{t-1} + g_u u_t + \frac{1}{2} (g_{yy} (\hat{y}_{t-1} \otimes \hat{y}_{t-1}) + g_{uu} (u_t \otimes u_t) + 2g_{yu} (\hat{y}_{t-1} \otimes u_t)) \quad (1.2.28)$$

其中, $\hat{y}_{t-1} \equiv y_{t-1} - \bar{y}$; 一阶导数 g_y , g_u 构成的系数矩阵已经在一阶求解逻辑中求出。二阶近似解无条件期望 (均值, Unconditional Mean) 为

$$E(y_t) = \bar{y} + \frac{1}{2} (\mathbf{I} - g_y)^{-1} (g_{\sigma\sigma} + g_{yy} \Sigma_{\otimes y} + g_{uu} \Sigma_{\otimes \epsilon}) \quad (1.2.29)$$

其中, \mathbf{I} 表示单位矩阵 (Identity Matrix), $\Sigma_{\otimes y} \equiv E(\hat{y}_{t-1} \otimes \hat{y}_{t-1})$ 。此时可知, 二阶近似时无条件期望不再和稳态值相同。

3. 一个例子: 经典 RBC 模型

此处使用本书“1.1.3 B&K 方法”一节中简单 RBC 模型的例子来阐述一阶求解逻辑。由于二阶手动求解较为繁杂, 此处不再详细探讨。为方便起见, 此处仅针对资本存量 K_t 的水平变量进行求解, 而非对数水平变量。然后使用资本积累方程消去投资变量, 并假定:

$$f(K_t, A_t) \equiv A_t K_t^\alpha + (1 - \delta) K_t \quad (1.2.30)$$

此时资源约束为

$$C_t + K_{t+1} = f(K_t, A_t) \quad (1.2.31)$$

其次，引入随机调节系数 σ ：

$$\log A_t = \rho \log A_{t-1} + \sigma \epsilon_t \quad (1.2.32)$$

即

$$A_t = \exp(\sigma \epsilon_t) A_{t-1}^\rho \quad (1.2.33)$$

最后，引入资本存量的政策函数：

$$K_{t+1} = g(K_t, A_{t-1}, \sigma) \quad (1.2.34)$$

接下来的任务是寻找上述政策函数的一阶近似解

$$K_{t+1} \approx K + g_k (K_t - K) + g_a (A_{t-1} - 1) \quad (1.2.35)$$

中的一次项系数 g_k 和 g_a 。为了解 g_k 和 g_a ，需要对一阶条件进行泰勒展开。

首先，由效用最大化问题，可得一阶条件 (Euler 方程)：

$$E_t [u'(C_t) - \beta u'(C_{t+1}) \cdot f_K(K_{t+1}, A_{t+1})] = 0 \quad (1.2.36)$$

将资源约束方程 (1.2.31) 代入式 (1.2.36)，可得

$$E_t \left[u' \left(\overbrace{f(K_t, A_t) - K_{t+1}}^{C_t} \right) - \beta u' \left(\overbrace{f(K_{t+1}, A_{t+1}) - K_{t+2}}^{C_{t+1}} \right) f_K(K_{t+1}, A_{t+1}) \right] = 0 \quad (1.2.37)$$

进一步将政策函数 (1.2.34) 代入 (1.2.37) 可得

$$E_t(K_t, A_{t-1}, \sigma; g) \equiv E_t \left[\begin{array}{l} u' \cdot (f(K_t, A_t) - g(K_t, A_{t-1}, \sigma)) \\ -\beta u' \cdot (f(g(K_t, A_{t-1}, \sigma), A_{t+1}) - g(g(K_t, A_{t-1}, \sigma), A_t, \sigma)) \\ \cdot f_K(g(K_t, A_{t-1}, \sigma), A_{t+1}) \end{array} \right] = 0 \quad (1.2.38)$$

其次，对式 (1.2.38) 左边关于 K_t 、 A_{t-1} 、 σ 这 3 个变量在其稳态 $(K, 1, 0)$ 处分别求偏导，并令其为 0：

$$E_K(K_t, A_{t-1}, \sigma; g) = E_A(K_t, A_{t-1}, \sigma; g) = E_\sigma(K_t, A_{t-1}, \sigma; g) = 0 \quad (1.2.39)$$

具体而言，

$$E_K = u'' \cdot (f_K - g_k) - \beta u'' \cdot (f_{KK} g_K - g_k^2) f_K - \beta u' \cdot f_{KK} g_K = 0 \quad (1.2.40)$$

$$\begin{aligned} E_A &= u'' \cdot (f_A \rho - g_a) - \beta u' \cdot (f_{KA} \rho^2 + f_{KK} g_A) - \beta u'' \cdot f_K f_A \rho^2 \\ &\quad - \beta u'' \cdot f_K (f_K - g_k - \rho) g_A = 0 \end{aligned} \quad (1.2.41)$$

$$\begin{aligned}
 E_\sigma &= -u''g_\sigma - \beta u'' \cdot f_K (f_K g_\sigma - g_K g_\sigma - g_\sigma) - \beta u' \cdot f_{KK} g_\sigma \\
 &\quad \text{确定性等价的来源} \\
 &= -\left(u'' + \beta u'' \cdot f_K (f_K - g_K - 1) + \beta u' \cdot f_{KK} \right) g_\sigma = 0
 \end{aligned} \tag{1.2.42}$$

其中, u' 、 u'' 、 g_K 、 g_A 、 g_σ 、 f_K 、 f_A 、 f_{KK} 、 f_{KA} 均表示相应下标变量的一阶、二阶或交叉偏导数在稳态处的取值。在计算过程中使用了如下的等式, 由技术变量的定义(1.2.32)可知:

$$\left. \frac{\partial A_{t+1}}{\partial A_t} \right|_{A_t=1, \sigma=0, \epsilon_t=0} = \rho, \quad \left. \frac{\partial A_{t+1}}{\partial \sigma} \right|_{A_t=1, \sigma=0, \epsilon_t=0} = 0 \tag{1.2.43}$$

式(1.2.40)是关于 g_K 的一元二次方程; 在 g_K 求出后, 由于式(1.2.41)是关于 g_A 的线性方程, 因此可立即求解。在求解之前, 需要先求出 u' 、 u'' 、 f_K 、 f_A 、 f_{KK} 、 f_{KA} 的值。

首先, 稳态的资本存量由一阶条件求出

$$K = \left(\frac{\alpha\beta}{1-(1-\delta)\beta} \right)^{\frac{1}{1-\alpha}} \tag{1.2.44}$$

而由 $f(K, A)$ 的定义, 可知

$$\begin{aligned}
 f(K, A) &= K^\alpha + (1-\delta)K, \quad f_K = \alpha K^{\alpha-1} + 1 - \delta = \frac{1}{\beta} \\
 f_A &= K^\alpha, \quad f_{KK} = \alpha(\alpha-1)K^{\alpha-2}, \quad f_{KA} = \alpha K^{\alpha-1}, \quad A=1
 \end{aligned} \tag{1.2.45}$$

再由效用函数的定义, 可知

$$u' = C^{-\gamma}, \quad u'' = -\gamma C^{-1-\gamma}, \quad C = f(K, A) - K \tag{1.2.46}$$

因此由式(1.2.44)~式(1.2.46), 可计算出 u' 、 u'' 、 f_K 、 f_A 、 f_{KK} 、 f_{KA} 的取值。在此基础上, 资本存量 K_t 的一阶条件(1.2.40)可写为

$$\frac{1}{\beta} - \left(1 + \frac{1}{\beta} + \beta \frac{u'}{u''} f_{KK} \right) g_K + g_K^2 = 0 \tag{1.2.47}$$

由于式(1.2.41)进一步可得

$$g_A = \rho \frac{u'' f_A - \beta u' f_{KA} \rho - \beta u'' f_A \rho f_K}{u'' + \beta u' f_{KK} + \beta u'' (f_K - (\rho + g_K)) f_K} \tag{1.2.48}$$

接下来, 将在 Matlab 中编程对上述计算逻辑进行数值求解^①, 参数的取值同原 RBC 模型: $\alpha=0.36$ 、 $\beta=0.99$ 、 $\delta=0.025$ 、 $\gamma=2$ 、 $\rho=0.9$ 。数值求解结果如下:

$$g_K = 0.9765, \quad g_A = 2.7695 \tag{1.2.49}$$

① 源文件地址: \Sources\Chap1_DSGE_basics\1.2_2nd_logic\first_order_perturbation_by_hand.m。

为了验证该求解结果的正确性，此处编写了该 RBC 模型对应的 Dynare 模型文件（以水平变量的形式）^①。在执行完模型文件后，可观察 Matlab 屏幕输出中的“政策函数和转换方程”部分，会发现资本存量 K_t 对应的一阶滞后系数 g_K 和技术变量 A_t 一阶滞后系数与此处手动计算的结果完全一致。此外，根据上述各参数的校准和计算可知

$$-\left[u'' + \beta u'' \cdot f_K (f_K - g_K - 1) + \beta u' \cdot f_{KK} \right] \neq 0 \quad (1.2.50)$$

因此由 (1.2.42) 可知 $g_\sigma = 0$ ，即确定性等价，具体请参考本书下一小节内容。

1.2.2 确定性等价和维数诅咒

扰动算法被广泛应用于经济模型的求解中。其优点在于不受维数诅咒 (Curse of Dimensionality) 的约束。维数诅咒是说当模型中状态变量的个数很多（即维数较大）时，求解将会耗费大量资源，甚至有时不可行。扰动算法最终归结为求解线性方程组，状态变量的多寡只和线性方程的维度或系数矩阵的维度相关，因而不受维数诅咒的影响。但是扰动算法有其缺点。特别是在一阶近似中，其解具有确定性等价 (Certainty Equivalence) 性质，因而限制了一阶近似求解方法的应用范围。但这种缺陷可以通过高阶近似加以解决。

上一节已经提及了确定性等价的概念，但为了使读者能够更加清楚地理解确定性等价的概念，此处介绍 Schmitt-Grohé & Uribe(2004) 关于确定性等价的阐述。Schmitt-Grohé & Uribe(2004) 关于变量表示方法区别于前述一阶和二阶求解逻辑中使用的方法，其将内生变量区分为内生状态变量和其他内生变量。但不管内生变量如何划分，确定性等价的本质是不变的：无条件均值和稳态相同，或者说，对称性外生冲击平均意义上没有作用^②。

1. 维数诅咒

维数诅咒是宏观经济学模型求解，特别是在动态规划求解中 (Dynamic Programming)，

① 源文件地址：\Sources\Chap1_DSGE_basics\1.2_2nd_logic\first_order_perturbation_by_hand.mod. Dynare 软件的使用、语法都会在后续章节详细介绍。如果尚未接触 Dynare 的读者，可忽略此部分。注意此处 mod 文件中的变量采取了水平形式，因此系数结果和对数水平对应的结果有较大差异。

② 稳态的求解是在模型没有不确定性，也就是说没有外生冲击时计算出来的内生变量的值。而无条件期望是指完全考虑到不确定性时计算出的内生变量的平均值，即考虑到了外生冲击的平均作用。当稳态和无条件期望相同时，意味着对称性外生冲击平均意义上没有作用，因此确定性等价也意味着外生冲击平均意义上没有作用。在 DynareWiki 中，关于脉冲响应的计算有这样一句注释：In Dynare, at order 1, we don't need to care about the possible values of shocks except for the impulsion, because in a linear model the average effect of symmetric shocks is always 0. This is not true for orders of approximation 2 and above.

遇到的一个难题。这也是动态规划方法无法求解中等或大型 DSGE 模型的重要原因。本部分将通过一个简单的示例，使用值函数递归算法求解一个简单的随机增长模型的政策函数，来解释维数诅咒的由来和背景。此处以本书“1.1.3 B&K 方法”一节中的随机增长模型为例，并使用值函数递归 (VFI) 算法求解政策函数。动态规划的方法要求求解模型对应的贝尔曼 (Bellman) 方程。

首先，确定模型的状态变量。该模型中含有两个状态变量：预先决定的状态变量，资本存量 K_{t+1} 和技术变量 A_t 。

然后，写出相应的 Bellman 方程：

$$V(K, A) = \max_C \left\{ \frac{C^{1-\sigma} - 1}{1-\sigma} + \beta EV(K', A') \right\} \quad (1.2.51)$$

受约束于

$$K' = AK^\alpha - C + (1-\delta)K \quad (1.2.52)$$

其中 E 表示期望算子，为了简化书写，去掉时间下标，使用未带时间下标的变量表示当期变量 (t 期)，带有上标一撇 (Prime) 的变量表示下一期变量即 $t+1$ 期。由资源约束条件，消除线性变量消费 C ，Bellman 方程可进一步写为

$$V(K, A) = \max_{K'} \left\{ \frac{(AK^\alpha - K' + (1-\delta)K)^{1-\sigma} - 1}{1-\sigma} + \beta EV(K', A') \right\} \quad (1.2.53)$$

求解 Bellman 方程是指在给定的状态 (K, A) 下以及技术变量 A 的随机过程，寻求最优参数值 K' 使得 (1.2.53) 得以成立。最优参数值 K' 选择是基于对 A' 的期望，因此对下一期值函数需要使用期望算子。求解 Bellman 方程的难点在于方程两边同时含有同一个值函数，而且其参数取值不同。对于这类问题，典型的求解方法就是递归法，其一般步骤如下：

第一，给出一个初始猜测值 $V^0(K, A)$ ；

第二，由 Bellman 方程计算

$$V^1(K, A) = \max_{K'} \left\{ \frac{(AK^\alpha - K' + (1-\delta)K)^{1-\sigma} - 1}{1-\sigma} + \beta EV^0(K', A') \right\}$$

第三，重复第二步 n 次，直到

$$\|V^n(K, A) - V^{n-1}(K, A)\| \leq \epsilon$$

其中， ϵ 为误差可容忍参数 (Tolerance)。

接下来，定义系统的状态空间。假设资本存量在如下的状态空间内变化：

$$[\bar{K} - 0.8\bar{K}, \bar{K} + 0.8\bar{K}]$$

其中, \bar{K} 表示资本存量的稳态值, 状态空间的选取以稳态值为中心, 两端长度相同, 为稳态值的 80%。系统的随机性来源于技术变量 A 。

为了简化分析, 此处不再假设技术变量服从经典的 AR(1) 过程, 而是假设其服从简单的 3 状态马尔可夫过程 (Markov Process): 假设 A 取如下 3 个状态值, 且定义其状态转换矩阵为 P :

$$A \equiv \begin{pmatrix} 0.8 \\ 1.0 \\ 1.2 \end{pmatrix}, P \equiv \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

在给定技术变量某个状态时, 其下一期状态有 3 种, 因而期望的值函数为

$$EV^0(K', A') = \left[\frac{1}{3}V^0(K', 0.8), \frac{1}{3}V^0(K', 1.0), \frac{1}{3}V^0(K', 1.2) \right]$$

最后, 进行编程求解^①。编程求解的基本逻辑如下: 首先校准模型结构参数, 在此基础上, 对资本存量的状态空间上的每一点, 递归求解值函数。在具体实施时, 则对状态空间进行均等划分成若干小区间, 使用 for 循环对每一端点求解下一期 (中间) 最优状态值。然后使用 while 循环实现递归求解, 确定最终最优状态值。运行 Matlab 代码, 即可得到图 1.7 和图 1.8 (区间数 kgrid 为 99, 端点数为 100)。

图 1.7 给出了不同技术状态下值函数的变化趋势: 第一, 给定资本存量水平, 技术水平越高, 值函数越大; 第二, 值函数是资本存量的下凹增函数, 资本存量越大, 值函数越大^②。

图 1.8 给出了不同技术状态下的政策函数, 即给定初始资本存量时最优的下一期资本存量。图 1.8 同时画出了 45° 线。可看出, 各政策函数的斜率都小于 1, 即比 45° 线更平坦。45° 线和 3 条线都相交, 说明模型存在稳态均衡。

① Matlab 源代码: \Sources\Chap1_DSGE_basics\1.2_2nd_logic\Val_fun_iter_stoc.m。直接运行该代码即可得到结果。此处非常感谢 Eric Sims 教授无私地提供和分享他的源代码。源代码中, 还牵涉很多实际问题。比如在求解最值时, fminbnd 内置函数要求目标函数必须是连续的, 而此处是离散数值求解, 因此必须进行处理。此处 Eric 教授采取了插值近似计算。

② 此处的值函数应为家庭终身贴现效用。

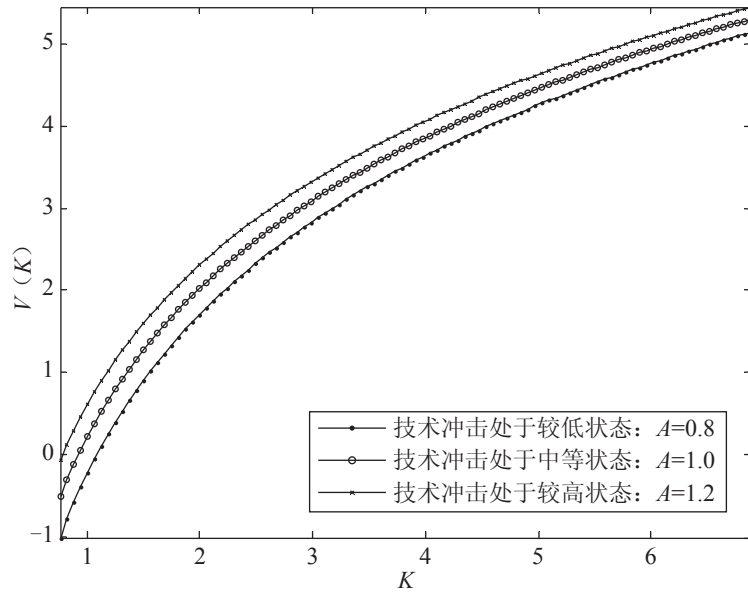
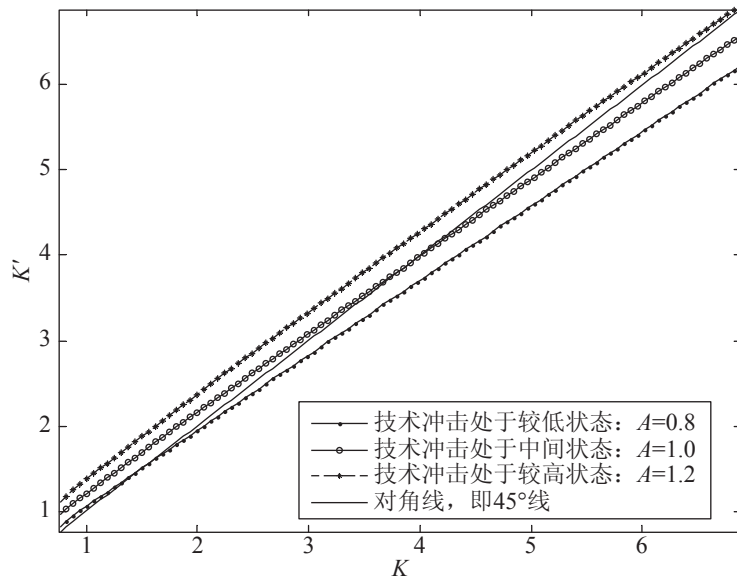


图 1.7 不同技术状态下的值函数

图 1.8 不同技术状态下的政策函数 (最优的资本存量 K')

其实，完全可以对技术变量采取和资本存量一样的处理方式，定义状态空间，并均

等划分成小区间。之所以简化处理，是因为如此处理，会节省不少计算资源。表 1.3 给出了资本存量状态空间两种划分下的耗时对比，细分程度加倍时，耗时变为原来的近 4 倍。如果将技术冲击的状态空间做相同的处理，取 100 个状态点，此时求解两变量 Bellman 方程的耗时将呈几何级数增长，从而变得不可接受。如果求解含有 3 个或更多状态变量的问题时，数值求解将会不再可行。这就是所谓的维数诅咒的由来。状态变量的个数越多，即维度越多，求解会越不可行。

表 1.3 资本存量状态空间细分程度与耗时比较

划分的区间数	$kgrid=99$	$kgrid=199$
耗时	dif = 11.733 6 its = 1	dif = 17.503 4 its = 1

	dif = 0.010 367 its = 84	dif = 0.010 126 its = 109
	dif = 0.009 848 5 its = 85	dif = 0.0096 198 its = 110
	Elapsed time is 201.195 085 seconds.	Elapsed time is 784.447 053 seconds.

数据来源：作者自行计算得到，不同平台下计算耗时会有所差异。dif 表示等式两端的误差；Elapsed time 表示耗时（秒）。

2. 确定性等价

前文已多次提及确定性等价的概念，此处以 Schmitt-Grohé & Uribe(2004) 为例进一步说明。Schmitt-Grohé & Uribe(2004) 假设系统均衡可写为如下的形式：

$$E_t f(\underbrace{y_{t+1}}_{n_y \times 1}, \underbrace{y_t}_{n_y \times 1}, \underbrace{x_{t+1}}_{n_x \times 1}, \underbrace{x_t}_{n_x \times 1}) = 0 \quad (1.2.54)$$

其中， y_t 为内生变量组成的 $n_y \times 1$ 维向量，该内生向量即为非预先决定的内生变量 (Non-predetermined)，包括前向变量和静态变量^①。 x_t 为 $n_x \times 1$ 维状态变量，包括预先决定的内生状态变量 x_{1t} 和外生状态变量 x_{2t} ：

$$x_t \equiv \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix} \quad (1.2.55)$$

外生状态变量满足 AR(1) 过程：

$$x_{2t+1} = Ax_{2t} + \underbrace{\tilde{\eta}}_{n_\epsilon \times n_\epsilon} \sigma \epsilon_{t+1} \quad (1.2.56)$$

其中，系数矩阵 A 的特征值都假设小于 1， $\tilde{\eta}$ 为 $n_\epsilon \times n_\epsilon$ 系数矩阵。模型 (1.2.54) 的解可表

① 通俗地说，非预先决定的内生变量是指只出现 t 和 $t+1$ 时间下标的内生变量；凡是出现 $t-1$ 时间下标的内生变量都认为是预先决定的内生变量，比如资本存量是预先决定的内生状态变量，虽然从形式上出现 $t+1$ 时间下标，但实际仍为 t 期变量，在 t 期末被确定。

示为

$$y_t = g(x_t, \sigma) \quad (1.2.57)$$

$$x_{t+1} = h(x_t, \sigma) + \eta\sigma\epsilon_{t+1} \quad (1.2.58)$$

其中, $\eta \equiv [0, \tilde{\eta}']'$ 为 $(n_x \times n_\epsilon)$ 为系数矩阵。

模型一阶和二阶近似解是指函数 g 、 h 的一阶和二阶泰勒近似, 在确定性稳态 $(x, \sigma) = (\bar{x}, 0)$ 附近分别进行一阶泰勒展开

$$g(x, \sigma) = g(\bar{x}, 0) + g_x(\bar{x}, 0)(x - \bar{x}) + g_\sigma(\bar{x}, 0)\sigma \quad (1.2.59)$$

$$h(x, \sigma) = h(\bar{x}, 0) + h_x(\bar{x}, 0)(x - \bar{x}) + h_\sigma(\bar{x}, 0)\sigma \quad (1.2.60)$$

和二阶泰勒展开

$$g(x, \sigma) = g(\bar{x}, 0) + g_x(\bar{x}, 0)(x - \bar{x}) + \frac{1}{2}g_{xx}(\bar{x}, 0)[(x - \bar{x}) \otimes (x - \bar{x})] \\ + g_{x\sigma}(\bar{x}, 0)\sigma(x - \bar{x}) + \frac{1}{2}g_{\sigma\sigma}(\bar{x}, 0)\sigma^2 \quad (1.2.61)$$

$$h(x, \sigma) = h(\bar{x}, 0) + h_x(\bar{x}, 0)(x - \bar{x}) + \frac{1}{2}h_{xx}(\bar{x}, 0)[(x - \bar{x}) \otimes (x - \bar{x})] \\ + h_{x\sigma}(\bar{x}, 0)\sigma(x - \bar{x}) + \frac{1}{2}h_{\sigma\sigma}(\bar{x}, 0)\sigma^2 \quad (1.2.62)$$

Schmitt-Grohé & Uribe(2004) 在定理 1 中指出 (1.2.59) ~ (1.2.62) 中关于随机调节变量 σ 的一阶项的系数矩阵均为 0, 即^①

$$g_\sigma(\bar{x}, 0) = 0, \quad h_\sigma(\bar{x}, 0) = 0, \quad g_{x\sigma}(\bar{x}, 0) = 0, \quad h_{x\sigma}(\bar{x}, 0) = 0 \quad (1.2.63)$$

这个定理表明, 一阶和二阶泰勒展开中, 关于随机调节变量的一次项的系数为 0 (即确定性等价); 因此在一阶泰勒和二阶泰勒展开中, 系数矩阵和随机调节变量无关, 也就是说外生冲击的标准差在一阶和二阶泰勒近似中没有任何影响 (二阶泰勒近似除常数项外)。这同时也意味着确定性模型和随机模拟在除常数项外, 一阶和二阶泰勒展开的系数应该相同。

特别地, 一阶近似解出现确定性等价的性质, 即此时内生变量无条件期望 (均值) 和确定性稳态相同:

$$E(y_t) = \bar{y}, E(x_t) = \bar{x} \quad (1.2.64)$$

其中, $\bar{y} = g(\bar{x}, 0)$ 为内生非预先决定变量的稳态值, $\bar{x} = h(\bar{x}, 0)$ 为外生变量的稳态值。事

① 为了获取上述泰勒展开中剩余的各项系数矩阵, 需要将 (1.2.57) 和 (1.2.58) 同时代入 (1.2.54), 可得 $0 = F(x, \sigma) \equiv E_t f \{g[h(x, \sigma) + \eta\sigma\epsilon_t, \sigma], g(x, \sigma), h(x, \sigma) + \eta\sigma\epsilon_t, x\}$, 然后关于状态变量 x 和随机调节变量 σ 求导, 并在稳态取值即可求解。

实上由 (1.2.60)，两边取无条件期望：

$$E(x_{t+1}) = E[h(x_t, \sigma)] = \bar{x} + h_x(\bar{x}, 0)[E(x_t) - \bar{x}] \quad (1.2.65)$$

如果内生和外生状态变量是平稳的，则 $E(x_{t+1}) = E(x_t)$ ，并注意到 $h_x(\bar{x}, 0)$ 为非单位矩阵，因此 $E(x_t) = \bar{x}$ 。从而由式 (1.2.59) 可推导出 $E(y_t) = \bar{y}$ 。

这种确定性等价的性质在二阶泰勒展开中不再具备，也就是说，内生变量的无条件均值和确定性稳态不再相等。事实上，相比确定性模型，内生非预先决定变量、内生状态变量对应的二阶展开公式则分别多出一个常数项：

$$\frac{1}{2}g_{\sigma\sigma}(\bar{x}, 0)\sigma^2, \frac{1}{2}h_{\sigma\sigma}(\bar{x}, 0)\sigma^2 \quad (1.2.66)$$

此时从二阶泰勒展开式 (1.2.62) 中，可知

$$E(x_t) = \bar{x} + \frac{1}{2}(I - h_x(\bar{x}, 0))^{-1}(h_{xx}(\bar{x}, 0)\Sigma_{\otimes x} + h_{\sigma\sigma}(\bar{x}, 0)\sigma^2) \quad (1.2.67)$$

$$E(y_t) = \bar{y} + g_x(\bar{x}, 0)(E(x_t) - \bar{x}) + \frac{1}{2}(g_{xx}(\bar{x}, 0)\Sigma_{\otimes x} + g_{\sigma\sigma}(\bar{x}, 0)\sigma^2) \quad (1.2.68)$$

此为式 (1.2.29) 的另外一种表现形式，因而无条件均值和确定性稳态不再相同。这也使得二阶近似具备了一阶近似不具备的统计性质：体现不确定性的影响。式 (1.2.66) 中的两个常数项恰恰说明了不确定性在起作用。也就是说，二阶近似在处理诸如资产定价问题时能够区别具有不同风险特征资产的平均回报率，或处理福利问题时能够捕捉不同风险水平下家庭的平均福利水平的差异 (Schmitt-Grohé & Uribe, 2004)。因此在处理某些特别的问题时，必须使用二阶或二阶以上的近似方法。

Schmitt-Grohé & Uribe(2004) 最大的理论贡献在于其发现扰动算法下，一阶和二阶泰勒近似解关于随机调节变量一次项的系数矩阵是 0(在泰勒展开基于确定性稳态时)，也就是说，一阶和二阶近似解均不含随机调节变量的一次项。换句话说，如果关于随机调节变量的一次项的系数矩阵不为 0，那么这个解就一定不是基于泰勒近似展开而得到的。Collard & Juillard (2001a, 2001b) 提出的“误差缩减 (Bias Reduction Procedure)”方法正是如此，此时的一次项和二次项的系数均依赖于随机调节变量，因此这种误差缩减方法和扰动算法并不等价。

接下来，以一阶和二阶近似解之间的关系为出发点，通过图示的方式来进一步解释确定性等价的概念，如图 1.9 所示。

图 1.9 给出了两种近似解下，对应的稳态和无条件均值之间的差异。虚线表示 45° 线，实线表示一阶近似解，曲线表示二阶近似解。点 A 表示系统的确定性稳态 (Deterministic

Steady State, 假设只有一个)。

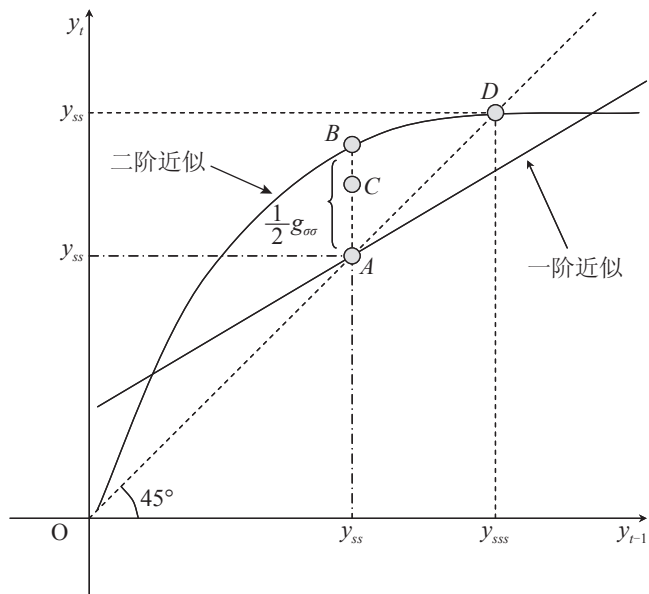


图 1.9 一阶和二阶近似解关于确定性稳态、不确定性稳态和无条件均值的差异

在一阶近似下，由确定性等价性质式 (1.2.64) 可知，点 A 也同时表示内生变量的无条件期望值 $E(y_t)$ ，也就是说如果当期选择稳态 A ，那么下一期最优决策仍然是点 A ，而不管风险如何，平均意义下风险都不起作用。

在二阶近似下，由决策规则 (1.2.28) 并考虑到风险和不确定性因素，可知最优决策不再是点 A 。如果当期选择稳态 A ，那么下一期最优决策不再是点 A ，而是点 B ；点 A 和点 B 之间的差异恰好为 $\frac{1}{2}g_{\sigma\sigma}$ ，此为风险作用的体现。由詹森不等式 (Jensen Inequality)^①，二阶近似下内生变量的无条件期望 $E(y_t)$ 不再是点 A ，也不位于二阶决策规则曲线上，而是位于二阶决策规则曲线之下的某个位置，比如位于点 A 和点 B 之间的某个 C 点位置。此外，点 D 则代表着不确定性稳态值 (Risky Steady State)^②，不确定性稳态概念会在本书下一节做简单介绍。

① 詹森不等式是以丹麦数学家约翰·詹森 (Johan Jensen) 命名的，是数学中的一个非常重要的不等式，也是有关凸性 (convexity) 的基本不等式。很多数学中重要的不等式都可以由詹森不等式推导或证明出来。

② 本书“1.3.1 确定性稳态值及其计算示例”一节会做简单的介绍。

1.3 DSGE 模型求解其他相关问题

1.3.1 确定性稳态值及其计算示例

本小节首先给出确定性稳态的定义及求解基本步骤，再通过一个简单的例子来说明如何求模型的确定性稳态值。

1. 确定性稳态的定义

确定性稳态 (Deterministic Steady State)，是指模型不存在任何外部冲击和不确定性，模型各内生变量都处于某一不变的稳定状态^①。假设模型的均衡可由如下的一般形式定义^②：

$$E_t f(y_{t+1}, y_t, x_{t+1}, x_t, z_{t+1}, z_t) = 0 \quad (1.3.1)$$

其中， y_t 为内生非预先决定变量； x_t 为预先决定的内生状态变量； z_t 为外生变量且满足

$$z_t = \Lambda z_{t-1} + \sigma \epsilon_t, \quad 0 < \Lambda < 1, \sigma > 0$$

其中，外生冲击 ϵ_t 的稳态被定义为 0。外生变量 z_t 的稳态也为 0 (一般为对数水平变量)。此处需要说明的是，对于一个 AR(1) 过程，比如典型的技术冲击过程：

$$\log A_t = \rho \log A_{t-1} + \epsilon_t, \quad 0 < \rho < 1$$

其中， ϵ_t 为外生冲击，稳态时其值取 0，因为其本身就被定义为独立同分布的白噪声过程。对于技术变量 A_t ，其为外生决定变量，稳态值一般设定为单位 1^③，也就是说其对数稳态为 0，事实上，根据技术变量 A_t 的 AR(1) 过程，设定外生冲击 ϵ_t 为 0，容易看到 $\log A_t = 0$ ，即水平变量 A_t 的稳态为 1。但这并不意味着所有的 AR(1) 过程对应的变量的稳态都为 1。比如对于常见的财政变量，如税收变量 (1.3.2)，其稳态值 $\bar{\tau}$ 一般应为小于 1 且大于 0 的某个值：

① 确定性稳态 (Deterministic) 有时也被称为非随机稳态 (Non-stochastic)。但某些文献中对非随机稳态有不同的定义，如 Groot, O. D. (2014). The risky steady state and multiple (spurious) equilibria: The case of the small open economy model，认为非随机稳态是指模型的无条件均值 (unconditional mean)。

② 此处关于一般形式的定义不同于 (1.1.1) 和 (1.2.54) 较为类似，但仍有区别。目的是为了后续关于不确定性稳态值的定义。

③ 在 Dynare 的变量声明中，仍然使用 var 命令声明，将其视为内生变量。一般对于那些 AR(1) 变量，如果其作为调节变量出现在均衡条件中，即乘法形式乘以其他变量，起到强弱调节作用，对系统均衡产生紧缩或扩张作用。其稳态值一般都被设置为 1。也就是说，稳态时此类变量不起作用。技术变量和资本利用率变量的作用都是如此，稳态值都是 1。有时也将技术水平变量 A 称为外生冲击。在给定初始状态外，其完全由白噪声过程决定。

$$\tau_t = (1 - \rho)\bar{\tau} + \rho\tau_{t-1} + \epsilon_t^r \quad (1.3.2)$$

式 (1.3.1) 的解可表示为

$$y_t = g(x_t, z_t, \sigma), \quad x_{t+1} = h(x_t, z_t, \sigma) \quad (1.3.3)$$

一个确定性的稳态值 $(y_t, x_t, z_t, \sigma) = (\bar{y}, \bar{x}, 0, 0)$, 被定义成满足如下等式的解:

$$f(\bar{y}, \bar{y}, \bar{x}, \bar{x}, 0, 0) = 0 \quad (1.3.4)$$

(1.3.4) 为非线性方程组, 注意该式中并未含有期望算子。求解上述非线性方程组, 即可求出相应的稳态值。然而有时直接求解 (1.3.4) 比较困难, 需要通过某些特殊的数值求解算法才能解出稳态值。此外, (1.3.4) 可能存在多个确定性稳态值。当模型存在多个稳态时, 扰动算法只能基于其中的一个进行泰勒近似, 而不能同时基于多个稳态近似求解。

围绕确定性稳态进行近似求解是文献中最常用的方法, 这是因为确定性稳态的求解和模型本身的随机设定没有关系, 稳态时已经假设模型没有随机性和不确定性, 因而求解相对简单方便。如果均衡存在, 在没有外生冲击或外生冲击逐渐消失时, 系统将逐渐收敛到确定性稳态值。

这里值得一提的是, 与确定性稳态相对应的一个概念是不确定性稳态 (Risky Steady State)。Coeurdacier, Rey & Winant (2011, *AER*) 将不确定性稳态定义为满足均衡条件 (1.3.1) 的这样一个点, 即当期 (t 期) 外生冲击为 0, 但未来各期 (包含 $t+1$ 期在内的以后各期) 仍然面临风险和不确定性, 即以后各期冲击都存在^①。Groot (2013, *EL*) 给出了不确定性稳态 $(y_t, x_t, z_t, \sigma) = (y^r, x^r, 0, \sigma)$ 的数学定义, 即满足如下方程的解:

$$E_t f(y_{t+1}, y^r, x^r, x^r, z_{t+1}, 0) = 0 \quad (1.3.5)$$

即当期冲击实现值为 0, 未来期的冲击仍存在。若用 (1.3.3) 的形式表示, 不确定性稳态满足:

$$x^r = h(x^r, 0, \sigma), \quad y^r = g(x^r, 0, \sigma) \quad (1.3.6)$$

比较式 (1.3.4) 和式 (1.3.5) 可知, 确定性和不确定性稳态的定义差异较大。不确定性稳态的计算较为复杂, 因为计算过程并不独立于系统的随机性, 虽然当期没有外生冲击, 但仍需要考虑未来的不确定性, 也就是说有期望算子出现^②。

不确定性稳态在某些情况下非常有用, 因为不确定性稳态本身包含了未来的不确

① ... point where agents choose to stay at a given date if the realization of shocks is zero at this date and expect future risk is stochastic... (or take into account of the distribution of future shocks).

② 在 Dynare 官方论坛中搜集了 9 个关于随机稳态计算的 Dynare 模型文件 (*.mod 文件), 请参考此处提供的 copy: \Sources\Chap1_DSGE_basics\1.3__other_issues_hp_filter\risky_steady_states_examples。感兴趣的读者可自行研究, 此处不再详细介绍。

定性信息。在考虑资产组合问题或不同宏观经济政策下的福利问题时，使用不确定性稳态较为契合 (Groot, 2013, P566)。关于不确定性稳态的计算，可参考 Coeurdacier, Rey & Winant (2011), Groot (2013), 此处不再详细介绍。其中 Groot (2013) 提出的算法较为简洁，并且容易操作，利用了 Schmitt-Grohé & Uribe(2004) 推导的二阶近似解，并给出了求解的 Matlab 代码。

2. 计算稳态的一般步骤

确定性稳态的计算是扰动算法得以实现的基础。对于简单的 DSGE 模型，确定性稳态的计算非常简单，甚至稳态都有解析表达式。但对于复杂的非线性模型，稳态值往往没有解析解，需要利用数值算法才能求出内生变量的稳态^①，比如带有金融加速器机制的 DSGE 模型。此外，确定性稳态计算的前提条件是所有结构参数都已知，即已被校准或估计。结合定义，确定性稳态的计算一般遵循如下的步骤^②：

第一，校准或估计所有结构参数；

第二，去除模型中所有外生冲击 (白噪声过程)；

第三，去除变量的时间下标；

第四，去除期望算子；

第五，编程求解。这往往是最困难的部分，需要对几个核心均衡方程和变量求解。有时需要复杂的循环或嵌套循环才能解决。在 Matlab 中，往往需要使用 `fzero` 等求根函数来寻找。

3. 两个稳态计算的例子

(1) 可分效用函数

这是一个含有资本存量和劳动的 RBC 模型，效用函数关于劳动和消费加性可分即 $U_{CN}=0$ ，且关于消费为对数形式，各内生变量的稳态有解析解，即可表示结构参数的解析表达式。

首先给出模型的均衡条件，分别为 Euler 方程 (1.3.7)、劳动供给方程 (1.3.8)、资本收

① 计算内生变量的稳态，对于某些中型或大型 DSGE 模型来说，的确是个问题，过程非常复杂。一般说来，DSGE 模型都要求是平稳的，即内生变量存在稳态，对于非平稳的模型需要经过处理，比如去趋势等变成稳态模型才能进行求解。Dynare 一般也要求模型是平稳的才能求解 (也能处理某些简单的非平稳模型，比如带趋势或 unit root 的模型)，并且能够计算某些简单模型的稳态值。但 Dynare 官方强烈建议使用者为 Dynare 提供模型的稳态，以提升求解速度和求解成功率。事实上，对于大多数模型 Dynare 无法计算其稳态。本书“3.8.2 稳态求解命令：steady”一节会进一步介绍 Dynare 稳态声明和处理逻辑。

② 上述计算步骤中的前三步可简称为“三去”处理：去冲击、去下标、去算子。

益率方程 (1.3.9)、劳动需求方程 (工资率决定方程) (1.3.10) 和资源约束条件 (或称为可行性约束, Feasibility Condition) (1.3.11) 和技术 AR(1) 外生过程 (1.3.12):

$$\frac{1}{C_t} = \beta E_t \left\{ \frac{1}{C_{t+1}} [r_{t+1} + (1-\delta)] \right\} \quad (1.3.7)$$

$$L_t^y = \frac{w_t}{C_t} \quad (1.3.8)$$

$$r_t = \alpha A_t K_{t-1}^{\alpha-1} (L_t)^{1-\alpha} \quad (1.3.9)$$

$$w_t = (1-\alpha) A_t K_{t-1}^{\alpha} (L_t)^{-\alpha} \quad (1.3.10)$$

$$K_t + C_t = (1-\delta)K_{t-1} + A_t K_{t-1}^{\alpha} (L_t)^{1-\alpha} \quad (1.3.11)$$

$$\log A_t = \rho \log A_{t-1} + \epsilon_t \quad (1.3.12)$$

其中, α 、 β 、 γ 、 δ 、 ρ 为模型结构参数, 且 $\gamma \neq -1$ 。注意, 此处资本存量的下标表示不同于以往模型, 具体可参考本书“3.2.2 Dynare 内生变量的书写规范”一节的阐述。根据稳态计算的一般步骤, 将上述均衡条件做“三去”处理, 可得如下的非线性方程组, 用不带时间下标的变量表示稳态值:

$$\frac{1}{C} = \beta \left(\frac{1}{C} (r + (1-\delta)) \right) \quad (1.3.13)$$

$$L^y = \frac{w}{C} \quad (1.3.14)$$

$$r = \alpha A K^{\alpha-1} (L)^{1-\alpha} \quad (1.3.15)$$

$$w = (1-\alpha) A K^{\alpha} (L)^{-\alpha} \quad (1.3.16)$$

$$K + C = (1-\delta)K + A K^{\alpha} (L)^{1-\alpha} \quad (1.3.17)$$

$$\log A = \rho \log A \quad (1.3.18)$$

首先技术变量 A_t 的稳态 A 为 1, 然后根据 Euler 方程可得资本收益率的稳态值为

$$r = \frac{1}{\beta} - (1-\delta) \quad (1.3.19)$$

再根据资本收益率稳态值和资本收益决定方程 (1.3.15), 可得资本劳动比稳态值为

$$\frac{K}{L} = \left(\frac{\alpha}{r} \right)^{\frac{1}{1-\alpha}} = \left(\frac{\alpha}{\frac{1}{\beta} - (1-\delta)} \right)^{\frac{1}{1-\alpha}} \quad (1.3.20)$$

由资本劳动比稳态值和劳动需求方程 (1.3.14)，可得工资稳态值为

$$w = (1 - \alpha) \left(\frac{\alpha}{\frac{1}{\beta} - (1 - \delta)} \right)^{\frac{\alpha}{1 - \alpha}} \quad (1.3.21)$$

最后来求劳动的稳态 L 。从劳动供给方程 (1.3.14) 求出消费稳态的表达式，并代入资源约束方程 (1.3.17)，两边除以劳动稳态 L 可得

$$\delta \frac{K}{L} + \frac{w}{L^{\gamma+1}} = A \left(\frac{K}{L} \right)^{\alpha} \quad (1.3.22)$$

$$L = \left(\frac{1 - \alpha}{1 - \frac{\alpha \delta}{\frac{1}{\beta} - (1 - \delta)}} \right)^{\frac{1}{\gamma+1}} \quad (1.3.23)$$

再由劳动供给方程 (1.3.14) 可得消费的稳态为

$$C = (1 - \alpha) \left(\frac{\alpha}{\frac{1}{\beta} - (1 - \delta)} \right)^{\frac{\alpha}{1 - \alpha}} \left(\frac{1 - \alpha}{1 - \frac{\alpha \delta}{\frac{1}{\beta} - (1 - \delta)}} \right)^{-\frac{\gamma}{\gamma+1}} \quad (1.3.24)$$

如果校准参数 $\alpha = 0.36, \beta = 0.99, \delta = 0.025, \gamma = 1$ ，可得内生变量的稳态如表 1.4 所示。

表 1.4 给定参数 γ 时各变量的稳态值

变 量	K	C	r	L	w
稳 态 值	35.243 7	2.555 3	0.035 1	0.927 7	2.370 6

数据来源：依据模型稳态计算逻辑，作者自行计算。

从上述模型稳态计算来看，劳动的稳态值为 0.927 7。如果认为劳动的总时间为单位 1，则此稳态值显然偏高，因此文献中往往采取相反的做法。即，先假定劳动的稳态值为 1/3，每天大约有 1/3 的时间用于劳动或工作，相当于 8 小时工作制，然后反向推定参数的

值^①。这里所指向的参数就是 γ 。参数 γ 有一个特殊的名字，叫劳动供给的Frisch弹性的倒数，会在RBC模型中做进一步介绍。在给定其他参数值不变的情况下，由(1.3.23)可知，

$$\gamma = \log \left(\frac{1-\alpha}{1 - \frac{\alpha\delta}{\frac{1}{\beta} - (1-\delta)}} \right) / \log L - 1 \approx -0.8634 \quad (1.3.25)$$

此时各变量的稳态如表 1.5 所示。可以发现，当稳态劳动为 1/3 时，资本存量和消费的稳态都下降了。

表 1.5 校准劳动稳态时各变量的稳态值

变 量	K	C	r	L	w
稳 态 值	12.663 1	0.918 1	0.035 1	0.333 3	2.370 6

数据来源：依据模型稳态计算逻辑，作者自行计算。

此外容易找到劳动稳态值 L 和参数 γ 之间的关系，如图 1.10 所示。图 1.10 显示了在其他参数给定的情况下，劳动供给的稳态 L 是参数 γ 的下凹增函数，随着参数 γ 的增加而逐步趋向于 1。

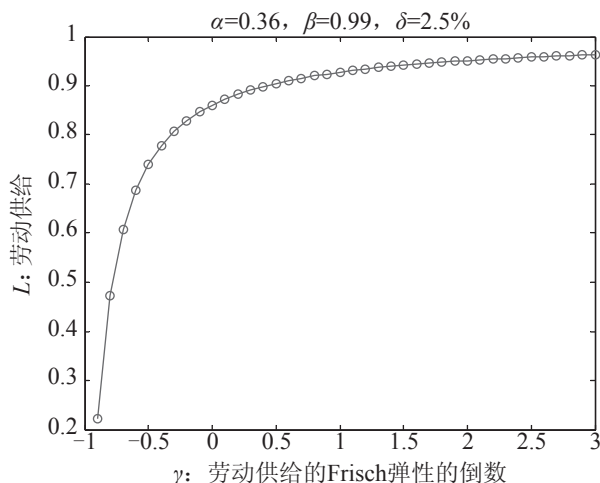


图 1.10 劳动供给和参数 γ 之间的关系^②

① 在给定某个内生变量的稳态值后，此时参数校准将失去一个自由度，也就是说必须有一个参数不能被自由校准或估计，必须由该内生变量的稳态值和其他参数联合决定。

② Matlab 源代码：\Sources\Chap1_DSGE_basics\1.3_other_issues_hp_filter\labor_gamma_relationship.m。

(2) 不可分效用函数

不可分 (Non-separable) 效用函数是指效用函数中消费和劳动不是加性可分的, 即 $U_{CN} \neq 0$ 。一个经典的不可分效用函数来源于 Greenwood, Hercowitz & Huffman (1988, *AER*)。在不可分效用函数假设下, 模型内生变量的计算变得复杂些。但对于简单的 RBC 模型而言, 内生变量的稳态值仍然可手动求出解析解。关于 Greenwood, Hercowitz & Huffman (1988) 模型的具体介绍和内生变量的稳态值计算, 请参考本书“4.1 RBC 模型及其拓展”一节的内容。

1.3.2 外生冲击持续参数和标准差的校准

外生冲击作为模型波动的主要来源, 对分析模型动态变化至关重要。在文献中, 一般假设外生冲击服从 AR(1) 过程。

AR(1) 过程由两个参数确定: 冲击方程 AR(1) 系数和冲击分布的标准差 (标准误)。AR(1) 系数即为持续性参数 (Persistence), 一般说来位于 0~1 之间, 表征冲击持续的时长和强度。持续性参数越大, 持续时间越长, 影响效果较远。

如下以技术冲击为例, 说明如何从数据中校准持续性参数和标准差参数。假设生产技术服从如下经典的 AR(1) 过程:

$$\log A_t = \rho_A \log A_{t-1} + \epsilon_t^A \quad (1.3.26)$$

其中, $0 < \rho_A < 1$, 技术外生冲击 $\epsilon_t^A \sim \text{i.i.d}(0, \sigma_A^2)$, 即服从均值为 0, 标准差为 σ_A 的高斯白噪声过程。考虑如下的柯布道格拉斯 (Cobb-Douglas) 生产函数:

$$Y_t = A_t K_t^\alpha N_t^{1-\alpha} \quad (1.3.27)$$

参考姚斌 (2007) 的做法, 对生产函数 (1.3.27) 两边取自然对数, 并做一阶差分可得:

$$\begin{aligned} \log A_{t+1} - \log A_t &= \log Y_{t+1} - \log Y_t - \alpha (\log K_{t+1} - \log K_t) \\ &\quad - (1-\alpha) (\log N_{t+1} - \log N_t) \end{aligned} \quad (1.3.28)$$

以我国 1992—2016 年度数据为例 (见表 1.6), 来估算生产技术水平。

表 1.6 国内生产总值、投资和就业人数

年 份	国内生产总值 (亿元, 现价)	投资总额 (亿元)	就业人数 (万人)
1992	27 194.50	8 080.10	66 152
1993	35 673.20	13 072.30	66 808
1994	48 637.50	17 042.10	67 455
1995	61 339.90	20 019.30	68 065
1996	71 813.60	22 913.50	68 950

续表

年 份	国内生产总值 (亿元, 现价)	投资总额 (亿元)	就业人数 (万人)
1997	79 715.00	24 941.10	69 820
1998	85 195.50	28 406.20	70 637
1999	90 564.40	29 854.70	71 394
2000	100 280.10	32 917.70	72 085
2001	110 863.10	37 213.50	72 797
2002	121 717.40	43 499.90	73 280
2003	137 422.00	55 566.61	73 736
2004	161 840.20	70 477.43	74 264
2005	187 318.90	88 773.61	74 647
2006	219 438.50	109 998.16	74 978
2007	270 232.30	137 323.94	75 321
2008	319 515.50	172 828.40	75 564
2009	349 081.40	224 598.77	75 828
2010	413 030.30	278 121.85	76 105
2011	489 300.60	311 485.13	76 420
2012	540 367.40	374 694.74	76 704
2013	595 244.40	446 294.10	76 977
2014	643 974.00	512 020.70	77 253
2015	689 052.00	561 999.80	77 451
2016	744 127.00	596 501.00	77 603

数据来源：中经统计网数据库和国家统计局网站；投资数据为全社会固定资产投资总额；就业数据为年末从业人员总数（包括城镇和农村从业人员总数）。

首先需要选取产出、资本存量和劳动的代理变量。以国内生产总值 GDP 作为 Y_t 的代理变量。由于资本存量的数据难以获取，而 (1.3.28) 中是关于资本存量的增长率，为了简化和计算的方便，使用投资的增长率近似替代。根据经典的资本积累方程，可得：

$$\log \frac{K_{t+1}}{K_t} = \log \left((1-\delta) + \frac{I_{t-1}}{K_t} \frac{I_t}{I_{t-1}} \right) \approx \frac{I_{t-1}}{K_t} \frac{I_t}{I_{t-1}} \approx \delta \frac{I_t}{I_{t-1}} \quad (1.3.29)$$

即以投资的总增长率乘以资本折旧率作为资本存量增长率的近似代理变量。而劳动则使用每年城镇就业人员总数作为代理变量。在进一步计算之前，需求对参数 α 、 δ 进行赋值。选择 $\alpha = 0.36$ 作为资本的产出份额。由于使用了年度数据，因此将折旧参数设定为 $\delta = 0.12$ 。

为了进行数据的拟合，需要获取对数技术变量的水平值，因此需要对初始对数技术变量进行假定。因为初始对数技术变量的大小只会产生水平效应（即截距会发生变化），

而不会产生斜率效应 [即 AR(1) 系数不会发生变化, 而且误差的标准差也不会发生变化], 因此理论上说可以假设为任何值。一般的情况下, 有两种做法: 第一, 假设初始对数技术变量为这样一个值, 它能够使得样本期内对数技术变量均值为 0, 即标准化的技术变量。第二, 为了计算的方便, 可以假设基期的对数技术变量为 0 (即技术变量的水平值为 1)。此处校准时使用第二种做法。

此外使用 Eviews 编程, 对外生冲击持续性参数和标准参数进行估计, 如源代码 6 所示。

源代码 6 外生冲击持续性参数和标准差的 Eviews 估计^①

```
'Define variables, genr is the built-in Eviews function to define new
variables
genr loggdpgrowth=log(gdp) -log(gdp(-1))
genr loginvgrowth = log(investment) - log(investment(-1))
genr loglabgrowth=log(labor)- log(labor(-1))

scalar alpha=0.4
scalar delta=0.12
genr productivity = loggdpgrowth - alpha*delta*loginvgrowth -
(1-alpha)*loglabgrowth
smpl 1994 2016
'recover the level of log productivity
productivity = productivity + productivity(-1)

'using hp filter to extract the trend, hpf is the built-in function of
Eviews to perform HP filtering.
productivity.hpf(lambda=100) hptrend_prod
'get the cycle of the productivity
genr hpcycle_prod= productivity - hptrend_prod

'the least square regression, ls is the built-in function of Eviews for
least square regression.
'eqhp is an Eviews object to store the LS regression results.
equation eqhp.ls hpcycle_prod hpcycle_prod(-1)
```

注: Eviews 工作表中的变量 `gdp` 代表国内生产总值, `investment` 代表固定资产投资总额, `labor` 代表就业人数。

在获取技术变量的 (对数) 水平变量后, 进行 HP 滤波处理, 获得波动成分。然后以 (1.3.26) 为回归方程, 进行最小二乘法回归, 一阶滞后项的系数即可作为持续性参数 ρ_A 的估计值, 回归的标准差 (S.E. of Regression) 可作为外生冲击标准差的估计值 σ_A , 回归

① Eviews 源代码: \Sources\Chap1_DSGE_basics\1.3_other_issues_hp_filter\persistence_vol_estimation.prg。即程序文件。数据存储于同目录下的 `productivity.wf1` 工作文件中。首先打开工作文件, 然后在打开 Eviews 程序文件, 运行该程序文件即可得到估计结果。

结果如表 1.7 所示: $\rho_A = 0.6867$, $\sigma_A = 0.0392$ 。

表 1.7 外生冲击持续性参数和标准差的 Eviews 估计结果

Dependent Variable: HPCYCLE_PROD				
Method: Least Squares				
Date: 04/19/17 Time: 13:09				
Sample (adjusted): 1995 2016				
Included observations: 22 after adjustments				
	Coefficient	Std. Error	t-Statistic	Prob.
HPCYCLE_PROD(-1)	0.686 656	0.158 542	4.331 068	0.000 3
R-squared	0.469 438	Mean dependent var		0.003 519
Adjusted R-squared	0.469 438	S.D. dependent var		0.053 785
S.E. of regression	0.039 177	Akaike info criterion		-3.597 075
Sum squared resid	0.032 231	Schwarz criterion		-3.547 482
Log likelihood	40.56 783	Hannan-Quinn criter.		-3.585 393
Durbin-Watson stat	0.755 138			

上述回归结果和 Galí(2008, P174) 的结果类似。

1.3.3 随机差分方程及其求解简析

随机差分方程在后续的分析中,特别是一阶差分方程经常用到,因为很多问题或均衡条件最后都可以归结为差分方程。因此,本节将对常见的几种差分方程及其求解进行简单介绍。

1. 确定性差分方程

最简单的确定性差分方程 (Deterministic Difference Equation, DDE) 当数一阶齐次单变量差分方程:

$$y_t = \alpha y_{t-1}$$

其中, α 为参数。很显然当 $|\alpha| < 1$ 时, $y_t \rightarrow 0, t \rightarrow \infty$ 。此处一阶是指只含有一阶滞后项 y_{t-1} ; 单变量是指 y_t 是一维的, 非多维变量; 确定性是指不带有随机扰动项, 如白噪声过程等; 齐次是指除了含有 y_t 及其滞后项或随机扰动项外, 不含有其他的项, 如常数项或其他非零项。如下的一阶单变量差分方程称为非齐次的 DDE, 即含有非零项 x_t :

$$y_t = \alpha y_{t-1} + \beta x_t$$

其中, x_t 为给定的非随机序列, $\alpha, \beta \neq 0$ 。在给定初始 (Initial Condition) 的条件下, 上述

DDE 可以前向迭代求解 (Forward Solution Method):

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta x_t \\ &= \alpha (\alpha y_{t-2} + \beta x_{t-1}) + \beta x_t \\ &\dots \\ &= \alpha^t y_0 + \beta \sum_{i=0}^{t-1} \alpha^i x_{t-i} \end{aligned}$$

此处 y_0 为给定的初始条件。

当给定终止条件 (Terminal Condition) 时, 可使用后向求解 (Backward Solution Method)。假设终止条件为 y_T , 即 T 期的取值, 此时需要对原始 DDE 做变形处理:

$$\begin{aligned} y_{t-1} &= \frac{1}{\alpha} y_t - \frac{\beta}{\alpha} x_t \\ &= \frac{1}{\alpha} \left(\frac{1}{\alpha} y_{t+1} - \frac{\beta}{\alpha} x_{t+1} \right) - \frac{\beta}{\alpha} x_t \\ &\dots \\ &= \frac{1}{\alpha^{T-t+1}} y_T - \beta \sum_{i=0}^{T-t} \frac{1}{\alpha^{i+1}} x_{t+i} \end{aligned}$$

在不确定性 (随机) 差分方程中, 很多变量往往只会给出终止条件, 而非初始条件, 如消费、资产价格这些前向变量 (Forward-looking)。这些变量的当期值往往受到外生冲击的影响, 没有初始值或者初始值并不重要, 更重要的是其未来的值, 受预期的影响, 因此往往给定终止条件而非初始条件。

2. 随机差分方程的几个例子

在不确定性模型中, 随机差分方程 (Stochastic Difference Equation, SDE) 是非常常见的。特别是一阶线性化的模型中能经常看到, 也就是说方程式中多出了一个随机扰动项或期望算子。下面举几个常见的例子。最经常看到的例子是一阶常系数非齐次差分方程。在特殊情况下, 能看到二阶常系数非齐次差分方程, 或者二元一阶常系数非齐次差分方程组。

(1) Euler 方程

RBC 模型中消费 C_t 的 Euler 方程, 其线性化后为一阶差分方程。在标准的 RBC 模型中, 消费的跨期替代方程为

$$\frac{1}{R_t} = \beta E_t \frac{C_t^\sigma}{\pi_{t+1} C_{t+1}^\sigma}$$

或者

$$\beta E_t \frac{C_t^\sigma}{C_{t+1}^\sigma} r_t = 1, \quad r_t \equiv \frac{R_t}{\pi_{t+1}}$$

其中, R_t 表示债券的名义利率 (Gross Rate); r_t 为形式定义; β 为贴现因子; σ 为消费的跨期替代弹性的倒数; $\pi_{t+1} \equiv P_{t+1}/P_t$ 为通胀 (Gross CPI Inflation)。若使用 Uhlig(1999) 的方法进行对数线性化, 即

$$\tilde{x}_t \equiv \log x_t - \log x$$

则上述消费的 Euler 方程可以改写为消费的一阶随机差分方程:

$$\tilde{c}_t = E_t \tilde{c}_{t+1} - \frac{1}{\sigma} \tilde{r}_t \quad (1.3.30)$$

(2) 新凯恩斯菲利普斯曲线

一般带有 Calvo(1983) 黏性价格设定的新凯恩斯模型, 都能推导出新凯恩斯菲利普斯曲线 (NKPC 曲线), 即当期通胀是预期通胀和边际成本的函数, 此为通胀的一阶随机差分方程:

$$\pi_t = \beta E_t \pi_{t+1} + \lambda m c_t \quad (1.3.31)$$

其中, $\beta < 1$ 为贴现因子, λ 为模型结构参数的函数, 关于参数的具体解释和 NKPC 曲线的推导可参考本书“4.2.7 ‘三方程’新凯恩斯模型”一节的内容。

(3) 投资学中的资产定价模型 (Asset-Pricing Model)

假设某风险证券的价格为 p_t , d_t 为其红利, 而无风险资产的投资回报率 r 为常值。根据资本定价理论, 在风险中性的假设下, 持有证券和无风险资产之间的套利会使得证券的期望回报率等于其无风险利率, 即

$$r = \frac{E_t p_{t+1} - p_t}{p_t} + \frac{d_t}{p_t}$$

整理后可得到关于资本价格的一阶随机差分方程:

$$p_t = \alpha E_t p_{t+1} + \alpha d_t, \quad \alpha \equiv \frac{1}{1+r} < 1$$

(4) 宏观货币模型: Cagan 模型

Cagan 模型是宏观经济学中一个著名的模型, 常被用来解释超级通胀问题。其核心观点是货币需求与期望的通胀负相关, 即期望的通胀越大, 实际货币余额需求越少:

$$\frac{M_t^d}{P_t} = \exp(-\alpha \pi_{t+1}^e) = \exp\left(-\alpha \frac{E_t P_{t+1} - P_t}{P_t}\right) \quad (1.3.32)$$

其中, $\pi_{t+1}^e \equiv \frac{E_t P_{t+1} - P_t}{P_t}$ 表示期望的通胀; M_t^d 表示货币需求; $\alpha > 0$ 为参数。考虑均衡状态, 货币需求 M_t^d 等于货币供给 M_t^s , 即

$$M_t^d = M_t^s = M_t$$

因此

$$\frac{M_t}{P_t} = \exp\left(-\alpha \frac{E_t P_{t+1} - P_t}{P_t}\right)$$

依据数学分析或高等数学中的经典结论: 当 x 很小时, 有 $\log(1+x) \approx x$ 或 $e^x \approx 1+x$; 若用小写字母表示大写字母的对数, 则上述等式可写为关于 (对数) 价格的一阶随机差分方程:

$$p_t = \lambda E_t(p_{t+1}) + (1-\lambda)m_t, \lambda \equiv \frac{\alpha}{1+\alpha} < 1$$

在上述 4 个例子中, 可看到其最终的方程都可以归结为如下形式的 SDE:

$$y_t = \alpha E_t(y_{t+1}) + \beta x_t \quad (1.3.33)$$

其中, x_t 为非零项或随机项, $\alpha, \beta \neq 0$ 。在第一个例子中, $\alpha=1$, 后 3 个例子中, 其前导项 (lead) 系数都满足: $0 < \alpha < 1$ 。通常情况下, y_t 被称为前向变量 (Forward-looking), 这类变量大都给定终止条件:

$$\lim_{t \rightarrow \infty} |y_t| < \infty$$

也就是说, 由未来的状态决定当前的行为, 从而存在均衡解或稳定解, 即施加了稳定性约束 (Stationarity)。这个终止条件非常类似于动态优化理论中的遍历性条件 (Transversality Condition), 从这个意义上讲, 施加遍历性条件, 其实就是施加了存在稳定解的条件。

(5) 二阶随机差分方程

在本书“7.1.2 基于福利损失的最优货币政策”一节中, 考虑带有成本推动型冲击 u_t 时, 规则承诺下的最优货币政策为价格的二阶随机差分方程:

$$p_t = \alpha p_{t-1} + \alpha \beta E_t p_{t+1} + u_t, \quad 0 < \beta < 1, \quad \alpha \equiv \frac{\omega}{\omega(1+\beta) + \kappa^2}, \quad \omega, \kappa > 0 \quad (1.3.34)$$

此为式 (7.1.54) 的推广形式。其中 p_t 为 (对数) 价格水平。此式说明, 当期最优的价格水平不仅是上一期价格水平, 也是预期价格水平的函数。其中外生冲击 u_t 满足 AR(1) 过程:

$$u_t = \rho_u u_{t-1} + \epsilon_t^u \quad (1.3.35)$$

假设外生冲击 u_t 满足 AR(1) 过程对于求解价格水平的解析解较为重要。

(6) 二元一阶常系数非齐次差分方程组

考察带有价格黏性的新凯恩斯三方程模型^①：

$$x_t = E_t x_{t+1} - \frac{1}{\sigma} (R_t - E_t \pi_{t+1} - \rho) + \epsilon_t \quad (1.3.36)$$

$$\pi_t = \beta E_t \pi_{t+1} + \kappa x_t + u_t \quad (1.3.37)$$

$$R_t = \rho + \phi_\pi \pi_t \quad (1.3.38)$$

其中, ϵ_t , u_t 分别为外生需求冲击和供给冲击 (成本推动型冲击); x_t , π_t 分别为产出缺口和通胀; R_t 表示名义利率; σ 、 ρ 、 κ 、 $\beta > 0$ 均为参数。将名义利率变量消去, 即可得到关于 x_t , π_t 的二元一阶常系数非齐次随机差分方程组:

$$\begin{cases} x_t = E_t x_{t+1} - \frac{1}{\sigma} (\pi_t - E_t \pi_{t+1}) + \epsilon_t \\ \pi_t = \beta E_t \pi_{t+1} + \kappa x_t + u_t \end{cases} \quad (1.3.39)$$

3. 求解方法

对于常系数差分方程, 无论是确定性还是随机差分方程, 一般都可以使用迭代法或待定系数法求解。

(1) 一阶差分方程

在上述4个例子中, 除第一个例子外, 其他3个例子中前导变量(lead)的系数均小于1。因此, 此处主要考虑(1.3.33)中当 $|\alpha| < 1$ 时的情形。当 $|\alpha| < 1$ 时, (1.3.33) 可使用后向迭代算法进行求解:

$$\begin{aligned} y_t &= \alpha E_t (E_{t+1} (\alpha y_{t+2} + \beta x_{t+1})) + \beta x_t \\ &= \underbrace{\alpha \beta E_t (x_{t+1}) + \beta x_t + \alpha^2 E_t (y_{t+2})}_{\text{迭代期望律, Law of Iterated Expectation}} \\ &= \dots \\ &= \beta \lim_{T \rightarrow \infty} \sum_{i=0}^T \alpha^i E_t (x_{t+i}) + \lim_{T \rightarrow \infty} \alpha^{T+1} E_t (y_{t+T+1}) \\ &= \underbrace{\beta \sum_{i=0}^{\infty} \alpha^i E_t (x_{t+i})}_{\text{遍历性条件}} \end{aligned} \quad (1.3.40)$$

① 具体请参考本书“4.2.7 ‘三方程’新凯恩斯模型”一节介绍。

式 (1.3.40) 中最后一个等号使用了终值约束条件。下面来看一个简单的例子，一阶非齐次随机差分方程：

$$\begin{aligned} y_t &= \alpha E_t y_{t+1} + \beta x_t \\ x_t &= \rho x_{t-1} + (1-\rho)\bar{x} + \epsilon_t \end{aligned}$$

其中， $\alpha=0.5$, $\beta=2$, $\rho=0.8$, $\bar{x}=3$, ϵ_t 为独立同分布的白噪声 (White Noise) 过程，并假设 x_t 的初始值为 2。使用上述后向算法，经过简单的迭代，可得到

$$y_t = \frac{\beta}{1-\alpha\rho} x_t + \frac{\alpha\beta(1-\rho)}{(1-\alpha)(1-\alpha\rho)} \bar{x} \quad (1.3.41)$$

上述求解也可使用待定系数法来求出，假定 $y_t = c_1 x_t + c_0$ ，然后代入即可求出式 (1.3.41)。使用上述的参数赋值，并对式 (1.3.41) 在 Matlab 中编程^①，可得到不确定性和确定性条件下 y_t 的轨迹图 [后向求解 100 期，由式 (1.3.41) 确定] 分别如图 1.11，图 1.12 所示。确定性条件下是指 x_t 中不存在白噪声过程，而不确定性是指 x_t 中存在白噪声过程。从图 1.11 中可以看到， y_t 围绕其均值 12 上下波动，而且从 y_t 的解析形式来看，白噪声被立即反映出来。

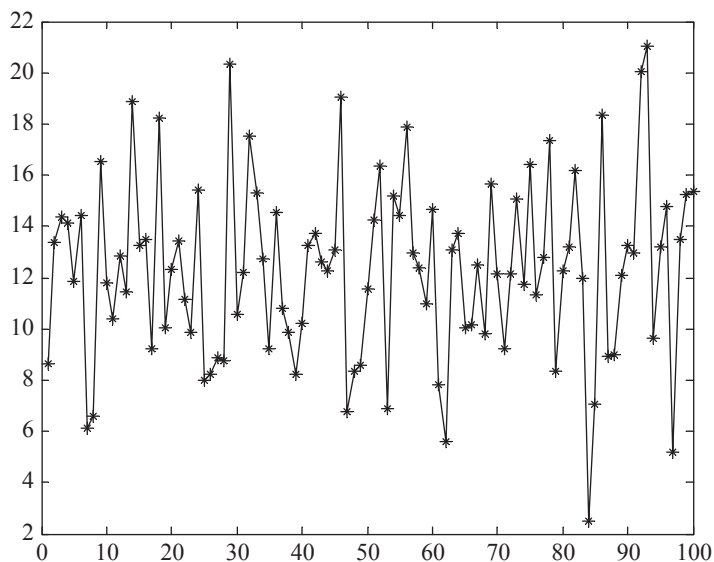
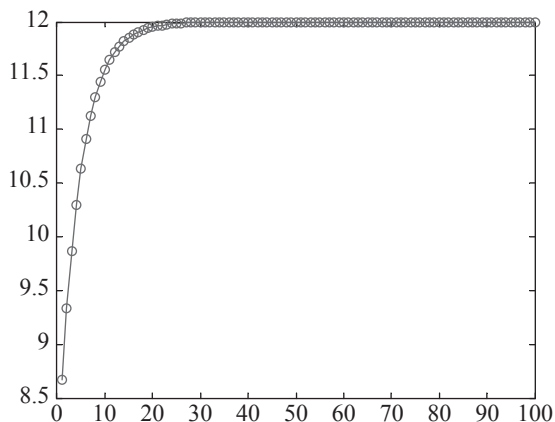


图 1.11 不确定性的情况下 y_t 的轨迹

从图 1.12 可看到， y_t 在经过短暂的上升后，一直稳定在其均值附近。如果将 x_t 的初始值设定为 3，即和 \bar{x} 相同，可以发现 y_t 将一直处于其均值上，不发生变化。

① 请直接运行 stochastic_diff_equ.m，即可得到上述两个图形。

图 1.12 确定性情况下 y_i 的轨迹

(2) 二阶差分方程

二阶差分方程的求解可同样使用待定系数法。以上述二阶随机差分方程 (1.3.34) 为例。假设其解可用如下形式的 AR(1) 过程表示：

$$p_t = \lambda p_{t-1} + \sigma u_t \quad (1.3.42)$$

其中， λ, σ 为待定系数。由 (1.3.42) 并注意到 u_t 服从 AR(1) 过程，容易得到：

$$E_t p_{t+1} = \lambda^2 p_{t-1} + \sigma(\lambda + \rho_u) u_t \quad (1.3.43)$$

然后将上述两式同时代入 (1.3.34) 中，可得关于 λ, σ 的二元二次方程组：

$$\alpha\beta\lambda^2 - \lambda + \alpha = 0 \quad (1.3.44)$$

$$\sigma = \frac{\alpha}{1 - \alpha\beta(\lambda + \rho_u)} \quad (1.3.45)$$

其中， α, β, ρ_u 均为已知参数。(1.3.44) 的两个根为

$$\lambda_{1,2} \equiv \frac{1 \pm \sqrt{1 - 4\alpha^2\beta}}{2\alpha\beta} \quad (1.3.46)$$

很显然两根之积、两根之和为

$$\lambda_1 + \lambda_2 = \frac{1}{\alpha\beta} > 1, \quad \lambda_1\lambda_2 = \frac{1}{\beta} > 1 \quad (1.3.47)$$

由 (1.3.34) 可知 $\alpha < \frac{1}{1+\beta}$ ，因此容易验证：

$$\lambda_2 \equiv \frac{1 + \sqrt{1 - 4\alpha^2\beta}}{2\alpha\beta} > \frac{1}{\beta} > 1 \quad (1.3.48)$$

因此另外一根必然小于 1，即

$$\lambda = \lambda_1 \equiv \frac{1 - \sqrt{1 - 4\alpha^2\beta}}{2\alpha\beta} < 1 \quad (1.3.49)$$

取稳定解，容易得到

$$\sigma = \frac{\lambda}{1 - \lambda\beta\rho_u} \quad (1.3.50)$$

因此最终将解写为

$$p_t = \lambda p_{t-1} + \frac{\lambda}{1 - \lambda\beta\rho_u} u_t, \lambda \equiv \frac{1 - \sqrt{1 - 4\alpha^2\beta}}{2\alpha\beta} \in (0, 1) \quad (1.3.51)$$

(3) 两元一阶差分方程组

两元一阶差分方程组也可以使用待定系数法来求解，以 (1.3.39) 为例。为了简化求解，此处假设外生冲击为独立同分布、互不相关的外生冲击，且产出缺口和通胀为外生冲击的线性函数。假设解具有如下形式：

$$\begin{cases} x_t = \lambda_1 \epsilon_t + \lambda_2 u_t \\ \pi_t = \gamma_1 \epsilon_t + \gamma_2 u_t \end{cases} \quad (1.3.52)$$

其中， λ_1 、 λ_2 、 γ_1 、 γ_2 为待定参数。通过简单的代数运算可得其解如下：

$$\begin{cases} x_t = \frac{\sigma}{\sigma + \kappa\phi_\pi} \epsilon_t - \frac{\phi_\pi}{\sigma + \kappa\phi_\pi} u_t \\ \pi_t = \frac{\kappa\sigma}{\sigma + \kappa\phi_\pi} \epsilon_t + \frac{\sigma}{\sigma + \kappa\phi_\pi} u_t \end{cases} \quad (1.3.53)$$

1.3.4 HP滤波的基本逻辑

数据处理在 DSGE 模型的估计以及后续分析中非常重要，特别是宏观经济数据尤其如此。从 DSGE 模型变量的构建和书写来看，大都将变量写为离差形式，即变量对其长期均衡值的偏离。严格地说，这些离差形式的变量都要求服从均值为零的协方差平稳随机过程 (Covariance Stationary Stochastic Process, CSSP)。因此，在寻求模型变量和观测统计数据之间的匹配时 (Take model to Data)，一般要求数据也是平稳的。

一般说来，时间序列数据大都含有某种趋势以及围绕这种趋势的某种波动。这种波动反映了经济波动的事实，具有 CSSP 性质，正是模型所需要的。从数据中去除趋

势,提取波动成分则成为经济分析的一个重要基础工作^①。因此,问题集中于如何去趋势。时间序列理论表明,不同的趋势特征需要使用不同的去趋势方法。当数据具有不变增长率的特征时(线性趋势),比如趋势平稳序列、差分平稳序列,可分别使用线性拟合和一阶差分的方法。当数据不具有线性增长趋势时,则需要考虑更复杂的处理方法。比如对于确定性的非线性趋势,可以考虑二次消除趋势处理(Quadratic Detrending),即使用二次多项式进行拟合;对于随机性的非线性趋势,可考虑使用诸如HP滤波、BP滤波等滤波工具^②。

本节将介绍HP滤波的基本逻辑及其Matlab实现。HP滤波算法是由Robert J. Hodrick & Edward C. Prescott于20世纪80年代初在分析美国战后的经济景气时首先提出的^③。后来HP滤波被广泛应用于对经济指标的分析中,其理论基础是时间序列的谱分析方法(Spectral Analysis),即将时间序列看作是不同的频率成分的复合。从直观上讲,HP滤波可从时间序列数据中抽出一条平滑曲线和一条波动曲线,即提取长期趋势(趋势性成分,Secular Component或随机性趋势,Stochastic Trend)和中短期波动成分(周期性成分,Cyclic Component)。因此,从这个意义上讲,可将HP滤波看作一个滤波器(Filter),能提取高频和低频成分。

简单地说,如果假设时间序列 $\{y_t\}$ 包含趋势项 $\{y_t^{Tr}\}$ 和波动项 $\{y_t^C\}$,即

$$y_t = y_t^{Tr} + y_t^C$$

那么HP滤波算法的功能就是从 $\{y_t\}$ 析出趋势项 $\{y_t^{Tr}\}$,因而也就析出了波动项 $\{y_t^C\}$ 。对于某个给定的有限长时间序列 $\{y_t\}_{t=1}^T$,HP滤波的基本想法是找到一个趋势序列(Trend) $\{g_t\}_{t=1}^T$,以最小化如下的目标函数:

$$\min_{g_t} V_T \equiv \sum_{t=1}^T (y_t - g_t)^2 + \lambda \sum_{t=2}^{T-1} [(g_{t+1} - g_t) - (g_t - g_{t-1})]^2, \lambda \geq 0 \quad (1.3.54)$$

目标函数中第一项是对波动项的考量,第二项则是对趋势项“平滑程度”的考量,参数 λ 用以调节两者的权重,被称为平滑参数。首先来关注参数 λ 的取值对结果的影响。

① 这并不意味着去趋势一定能够获取平稳的波动成分。但这往往是必要的,特别是对于那些波动位于6~40个季度的数据。

② 滤波(Filter)本质上是一种统计算法,其假定数据是由不同频率的成分组成(注意,线性趋势的波动频率为零)。HP滤波用来消除低频波动,BP滤波能够消除指定频率以外的所有波动成分,而季度滤波(Seasonal Filter)则用来消除高频波动(Seasonality)。注意,趋势和波动相比,趋势的频率更低。

③ 该文最早出现于1980年,University of Warwick, Discussion Paper No.451。后正式于1997年由JMCB杂志重印了这个未正式出版的discussion paper,1981.5修改版。Hodrick R. J., Prescott E. C. Postwar U.S. Business Cycles: An Empirical Investigation[J]. Journal of Money Credit & Banking, 1997(1):1-16。

如果当 $\lambda=0$ 时, 很显然 $g_t=y_t, t=1,2,\dots, T$ 是唯一解, 即趋势项即为序列本身。

如果当 $\lambda \rightarrow \infty$ 时, 此时序列的趋势项为线性时间趋势, 即 $g_t=\alpha t$, 其中 α 是常数。事实上, 当 $\lambda \rightarrow \infty$ 时, 如果对某个时间 $t, (g_{t+1}-g_t)-(g_t-g_{t-1}) \neq 0$, 此时目标函数的取值将趋于无穷大 ∞ , 无法实现最小化。因此对任意时间 $t, (g_{t+1}-g_t)-(g_t-g_{t-1})=0$, 即 $g_{t+1}-g_t=g_t-g_{t-1}$ 。也就是说, 每期趋势项的变化相同, 因此不妨假设 $\alpha=g_t-g_{t-1}, g_0=0$, 即可得到趋势项为线性时间趋势。

如果当 $0 < \lambda < \infty$ 时, 此时需要对目标函数关于变量 $g_t, t=1, \dots, T$ 求一阶偏导, 总共有 T 个一阶条件, 然后求解 T 元线性方程组即可。

首先将目标函数 V_T 的前 3 期展开:

$$\begin{aligned} V_T &= (y_1 - g_1)^2 + \lambda(g_3 - 2g_2 + g_1)^2 \\ &\quad + (y_2 - g_2)^2 + \lambda(g_4 - 2g_3 + g_2)^2 \\ &\quad + (y_3 - g_3)^2 + \lambda(g_5 - 2g_4 + g_3)^2 \\ &\quad \dots \end{aligned}$$

容易求得关于 g_1, g_2, g_3 的一阶条件:

$$\begin{aligned} \frac{\partial V_T}{\partial g_1} = 0 &\Leftrightarrow y_1 = (1 + \lambda)g_1 - 2\lambda g_2 + \lambda g_3 \\ \frac{\partial V_T}{\partial g_2} = 0 &\Leftrightarrow y_2 = -2\lambda g_1 + (1 + 5\lambda)g_2 - 4\lambda g_3 + \lambda g_4 \\ \frac{\partial V_T}{\partial g_3} = 0 &\Leftrightarrow y_3 = \lambda g_1 - 4\lambda g_2 + (1 + 6\lambda)g_3 - 4\lambda g_4 + \lambda g_5 \end{aligned}$$

特别地, 对于 $t=3, 4, \dots, T-2$, 有

$$\frac{\partial V_T}{\partial g_t} = 0 \Leftrightarrow y_t = \lambda g_{t-2} - 4\lambda g_{t-1} + (1 + 6\lambda)g_t - 4\lambda g_{t+1} + \lambda g_{t+2}$$

然后将目标函数 V_T 的后 2 期展开, 求出关于 g_{T-1}, g_T 的一阶条件:

$$\begin{aligned} V_T &= \dots \\ &\quad + (y_{T-2} - g_{T-2})^2 + \lambda(g_{T-1} - 2g_{T-2} + g_{T-3})^2 \\ &\quad + (y_{T-1} - g_{T-1})^2 + \lambda(g_T - 2g_{T-1} + g_{T-2})^2 \\ &\quad + (y_T - g_T)^2 \end{aligned}$$

容易求出:

$$\frac{\partial V_T}{\partial g_{T-1}} = 0 \Leftrightarrow y_{T-1} = \lambda g_{T-3} - 4\lambda g_{T-2} + (1+5\lambda)g_{T-1} - 2\lambda g_T$$

$$\frac{\partial V_T}{\partial g_T} = 0 \Leftrightarrow y_T = \lambda g_{T-2} - 2\lambda g_{T-1} + (1+\lambda)g_T$$

为了方便求解线性方程组, 定义如下的向量和矩阵:

$$\mathbf{Y}_T \equiv \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T-1} \\ y_T \end{pmatrix}, \mathbf{G}_T \equiv \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{T-1} \\ g_T \end{pmatrix}, \mathbf{A} \equiv \begin{pmatrix} 1+\lambda & -2\lambda & \lambda & 0 & \dots & \dots & \dots & 0 \\ -2\lambda & 1+5\lambda & -4\lambda & \lambda & 0 & \dots & \dots & 0 \\ \lambda & -4\lambda & 1+6\lambda & -4\lambda & \lambda & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & & \lambda & -4\lambda & 1+6\lambda & -4\lambda & \lambda \\ 0 & \vdots & & & \lambda & -4\lambda & 1+5\lambda & -2\lambda \\ 0 & 0 & 0 & 0 & \dots & \lambda & -2\lambda & 1+\lambda \end{pmatrix}$$

其中, \mathbf{A} 为 $T \times T$ 对称矩阵 (Symmetric), 其各元素为参数 λ 的线性函数, 即由一阶条件中各趋势变量的系数构成, 此矩阵为 HP 滤波问题的核心。容易看出, 当 $\lambda=0$ 时, \mathbf{A} 为单位矩阵 (Identity Matrix), 此时 $\mathbf{Y}_T = \mathbf{G}_T$, 即趋势项为数据序列本身, 波动项为 0。由上述定义, T 个一阶条件可写成矩阵形式:

$$\mathbf{Y}_T = \mathbf{A} \mathbf{G}_T$$

当矩阵 \mathbf{A} 可逆时 (一般情况下 \mathbf{A} 是可逆的), 即可求得解 $\mathbf{G}_T = \mathbf{A}^{-1} \mathbf{Y}_T$, 因此可以通过 Matlab 进行简单的循环编程加以实现。从上述分析中, 可看出要实现 HP 滤波, 其实非常简单: 在数据准备好后, 只要选择适当的 λ 值即可。一般说来, λ 取值应依据观测变量序列的取值频率确定。文献中大都使用如下的 λ 取值: 年度数据为 $\lambda=100$, 季度数据为 $\lambda=1\ 600$, 月度数据为 $\lambda=14\ 400$ ^①。此处亦然。

依据 (1.3.55), HP 滤波可以由源代码 7 实现 (hp_filter.m):

① 对于季度数据, 文献基本认可由原作者提出的取值: $\lambda=1\ 600$; 尤其是年度数据的取值争论比较大, 在此不再详述了。此处选取了 Eviews 软件的默认值 100, 由 Backus, D. K. and P. J. Kehoe (1992). *International Evidence on the Historical Properties of Business Cycles*. *American Economic Review* 82 (4): 864-888 提出。

源代码7 HP滤波的Matlab实现

```
function [ytrend,ycycle] = hp_filter(y,lambda)
%A SIMPLE code for implementing the HP filter
%Written By Xiangyang Li@UND
%y should be a column vector with T>=4 since the entries in LAMBDA is
hard coded.

%y should be a column vector or matrix with #rows>=4 , otherwise a error
%issued by Matlab will appear in Command window.
%number of rows
T= size(y,1);
if T<4
    error('rows of series must great than 3' );
end
%initialization of the Coefficient Matrix
mLAMBDA = zeros(T,T);

%the first 2 and last 2 rows of LAMBDA IS HARD CODED.
mLAMBDA(1,1) = 1+lambda;    mLAMBDA(1,2) = -2*lambda;
mLAMBDA(1,3) = lambda;

mLAMBDA(2,1) = -2*lambda;    mLAMBDA(2,2) = 1+ 5*lambda;
mLAMBDA(2,3) = -4*lambda;    mLAMBDA(2,4) = lambda;

for ii = 3:T-2
    mLAMBDA(ii,ii) = 1 + 6*lambda;    mLAMBDA(ii,ii+1) = -4*lambda;
    mLAMBDA(ii,ii+2) = lambda;        mLAMBDA(ii,ii-1) = -4*lambda;
    mLAMBDA(ii,ii-2) = lambda;
end

mLAMBDA(T,T) = 1 + lambda;        mLAMBDA(T,T-1) = -2*lambda;
mLAMBDA(T,T-2) = lambda;        mLAMBDA(T-1,T) = -2*lambda;
mLAMBDA(T-1,T-1) = 1 + 5*lambda; mLAMBDA(T-1,T-2) = -4*lambda;
mLAMBDA(T-1,T-3) = lambda;

ytrend = mLAMBDA\y;

ycycle = y - ytrend;
```

该代码定义了 `hp_filter` 函数，有两个输入参数：时间序列数据 (y) 和平滑参数 λ ，两个输出参数：HP 趋势项 (`ytrend`) 和波动项 (`ycycle`)^①，该函数可反复使用。注意此处对矩

① 感兴趣的读者，可以进一步验证该代码的正确性。首先选取一段时间序列数据，比如 GDP 数据；然后根据数据频率选取适当的 λ 值，在 Eviews 软件中进行 HP 滤波处理，并同时使用此处的 `hp_filter` 函数进行滤波处理，然后比较所得结果是否完全一致。

阵 $A(mLAMBDA)$ 的编程是整个函数的核心内容, 前两行和后两行的元素都直接加以赋值 (Hardcoded), 中间行的元素则通过 for 循环的方法, 动态赋值。因此, 这要求时间序列数据的长度不能小于 4 期, 否则会报错。

值得指出的是, HP 滤波的结果不仅依赖于平滑参数 λ 的选择, 而且也依赖于时间序列的长度 T 。不同的平滑参数值和不同长度的序列, 都会导致不同的 HP 滤波结果, 这一点在上述分析和编程中已能够清楚地看到。关于去趋势 (Detrending) 的更多文献, 可参考 Canova(1998, *JME*), 这是一篇非常不错的综述性文章, 发在《货币经济学》(*Journal of Monetary Economics, JME*) 上, 值得一读。

参 考 文 献

- [1] Andreasen M. M., Fernández -Villaverde J., Rubio-Ramírez J. F. *The Pruned State-Space System for Non-Linear DSGE Models: Theory and Empirical Applications*[J]. NBER Working Papers 18983,2013.
- [2] Aruoba S. B., Fernández-Villaverde J., Rubio-Ramírez J. F. *Comparing Solution Methods for Dynamic Equilibrium Economies*[J]. *Journal of Economic Dynamics and Control*,2006,30(12):2477-2508.
- [3] Blanchard O. J., Kahn C. M. *The Solution of Linear Difference Models Under Rational Expectations*[J]. *Econometrica*,1981,48(5):1305-1311.
- [4] Calvo G. A. *Staggered Prices in a Utility-Maximizing Framework*[J]. *Journal of Monetary Economics*,1983,12(3):383-398.
- [5] Canova F.*Detrending and Business Cycle Facts*[J]. *Journal of Monetary Economics*,1998,41(3):533-540.
- [6] Coeurdacier N., Rey H., Winant P.*The Risky Steady State*[J]. *American Economic Review*,2011,101(3):398-401.
- [7] Collard F., Juillard M.*Perturbation Methods for Rational Expectations Models*[J]. Manuscript, CEPREMAP,Paries, February,2001a.
- [8] Collard F., Juillard M.*Accuracy of Stochastic Perturbation Methods: The Case of Asset Pricing Models*[J]. *Journal of Economic Dynamics and Control*,2001b,25(6-7):979-999.
- [9] Fleming W. H.*Stochastic Control for Small Noise Intensities*[J].*SIAM Journal of Control*,1971,9(3):473-517.

- [10] Greenwood J., Hercowitz Z., Huffman G. W. *Investment, Capacity Utilization, and the Real Business Cycle*[J]. American Economic Review, 1988, 78(3):402-417.
- [11] Groot O. D. *Computing the Risky Steady State of DSGE Models*[J]. Economics Letters, 2013, 120(3):566-569.
- [12] Hansen G. D. (1985). *Indivisible Labor and the Business Cycle*[J]. Journal of Monetary Economics, 1985, 16(3):309-327.
- [13] King R. G., Plosser C. I., Rebelo S. T. *Production, Growth and Business Cycles: II. New Directions*[J]. Journal of Monetary Economics, 1988, 21(2-3):309-341.
- [14] Kydland F. E., Prescott E. C. *Time to Build and Aggregate Fluctuations*[J]. Econometrica, 1982, 50(6):1345-1370.
- [15] McCallum B. G. *The ABCs of RBCs: An Introduction to Dynamic Macroeconomic Models*[M]. Harvard University Press, 2008: 448.
- [16] Paul K. *Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model*[J]. Journal of Economic Dynamics and Control, 2000, 24(10):1405-1423.
- [17] Schmitt-Grohé S., Uribe M. *Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function*[J]. Journal of Economic Dynamics and Control, 2004, 28(4):755-775.
- [18] Sims C. A. *Solving Linear Rational Expectations Models*. [J]. Computational Economics, 2002, 31(2):95-113.
- [19] Sims E. R. *News, Non-Invertibility, and Structural VARs*[M]. Balke N., Canova F., Milani F., Wynne M. A. DSGE Models in Macroeconomics: Estimation, Evaluation, and New Developments (Advances in Econometrics). Emerald Group Publishing Limited[D], 2012:81-135.
- [20] Uhlig H. *A Toolkit for Analyzing Nonlinear Dynamic Stochastic Models Easily* [D]. CentER, University of Tilburg, and CEPR, Unpublished Manuscript, 1999.
- [21] 姚斌. 基于福利分析的人民币汇率制度选择研究 [D]. 上海: 复旦大学经济学院国际金融系, 2007.



2

初识 Dynare

Dynare 是 Matlab 的一个预处理器 (Pre-processor)，Dynare 的模型文件将被 Dynare 预处理成 Matlab 可识别的文件，通常为 m 文件，然后依靠 Matlab 求解和运行，输出最终计算的数值结果和图形结果^①。

Dynare 能够求解一类广泛的宏观经济学模型，如 DSGE 模型和 OLG 模型，而且不管模型是否基于理性预期的假设。也就是说，Dynare 能够处理非理性预期假设的模型。此外，Dynare 能够在一定程度上解决异质性、变量非平稳等问题，但对于复杂的异质性模型的求解几乎无能为力。

Dynare 的编程语言比较简单易懂，初学者一旦掌握，能够在很短的时间内完成一个中等规模 DSGE 模型的编程和求解。在 Dynare 中，通过使用简单的命令，就能完成极其复杂的计算任务，从而大大减轻研究者的负担。Dynare 在编写的过程中，大量运用了文献中经典的数值求解和近似算法，例如：多元非线性求解、优化算法和矩阵分解，以及许多经典的统计算法如卡曼滤波、粒子滤波、贝叶斯估计中的 MCMC 算法，等等。

Dynare 主要由法国的经济研究与应用中心 (CEPREMAP) 的核心团队成员在业余时间内开发完成的，而且研究团队和合作伙伴^②在不断扩大，从而保证了 Dynare 更新和完善，及时将经典文献写入 Dynare。也正因此，成就了 Dynare 功能的强大，以及用户基础日益扩大，成为宏观经济研究中一个不可多得的免费工具软件。

2.1 安装 Dynare

Dynare 提供了多种操作系统下的安装软件包 (Packages)，包括 Windows，MacOS 和两种 GNU/Linux 操作系统。下面仅仅针对 Windows 操作系统讲述如何安装 Dynare，其

① 本书仅仅考虑 Matlab 软件，未考虑 Octave 软件。

② Dynare 项目的资金来源主要是 CEPREMAP、法国央行 (Banque de France) 和 DSGE-net。

他操作系统的安装请参考 Dynare 官网。

2.1.1 安装环境

以版本 4.5.0 为例，Dynare 支持 Windows XP/Vista/7/8 以及 Matlab 7.3/R2006b (或以后版本) 或 Octave (4.2.1)。其中 Matlab 和 Octave 软件选择其一。Octave 是一个类似于 Matlab 的免费 GNU 软件，语法大部分和 Matlab 兼容，因此可粗略地认为是 Matlab 的免费简化版本^①。因此，在 Matlab 不可获得的情况下，安装 Octave 是一个非常不错的选择。在 Windows 操作系统下，为了安装 Dynare，必须首先安装 Matlab 或 Octave，而且版本号必须达到上述要求。

2.1.2 下载安装

截至 2018 年 10 月，Dynare 最新版本为 4.5.6，提供了非常强大的功能。本书附属的源代码包中有 Dynare 历史版本的拷贝。

首先去 Dynare 的官方网站 (<http://www.dynare.org>) 下载最新版本的安装包。此处以 4.5.0 版本为例。Windows 下的安装包已经打包成 exe 可执行文件：dynare-4.5.0-win.exe，大约为 15MB，双击即可执行安装。图 2.1 为 Dynare 安装组件选择界面。

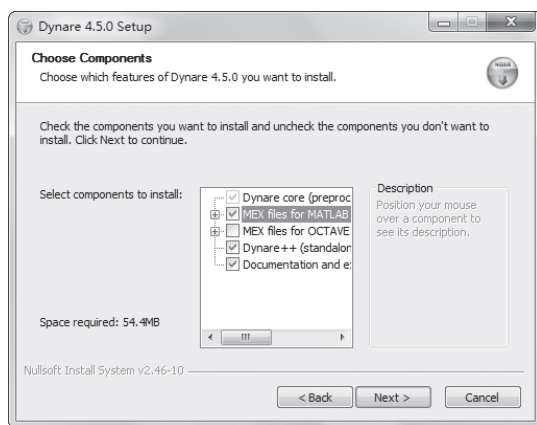


图 2.1 Dynare 安装组件选择界面

在选择组件的界面，Dynare 提供了 5 个组件：第一个是 Dynare 的核心，包括预处理器和重要的 Matlab 文件，为必选组件，对应安装目录下的 matlab 文件夹，包含核心

^① <https://www.gnu.org/software/octave/>. Octave 作为 GNU 计划的一部分，能够实现 Dynare 的几乎全部功能。

m 文件和以 exe 形式存在的预处理器；第二个是 Matlab 的 MEX 文件^①，可根据 Matlab 的版本（包括 32 位和 64 位，其中 32 位的版本包括 R2007b-R2015b，64 位的版本包括 R2007b-R2017a）选择需要的 MEX 文件；第三个是 Octave 的 MEX 文件，如果选择了 Matlab 的 MEX 文件，即可取消此选项；第四个是 Dynare++，这是一个由 C++ 语言编写的、可独立运行的简化版的 Dynare^②，专门用于求解 DSGE 的高阶近似解，感兴趣的读者可查看安装目录下的帮助文档；第五个是文档和例子（包括 Dynare 和 Dynare++），这个选项非常重要，强烈建议安装。如果勾选此项，会在安装目录下多出 doc 目录和 examples 目录。在 doc 目录下，会有当前版本的全套帮助文档，包括参考手册 Dynare Reference Manual，既有 PDF 版本，也有 html 版本、入门帮助 (UserGuide.pdf) 以及宏处理、预编译器等相关 PDF 帮助文档。

接下来，选择安装路径，如图 2.2 所示。这里有两点要注意：第一，Dynare 会自动创建安装路径，最后一层目录通常是版本号，建议大家不要更改此版本号；第二，如果更改安装路径，请避免在安装路径中出现中文字符，否则可能出现意想不到的问题。

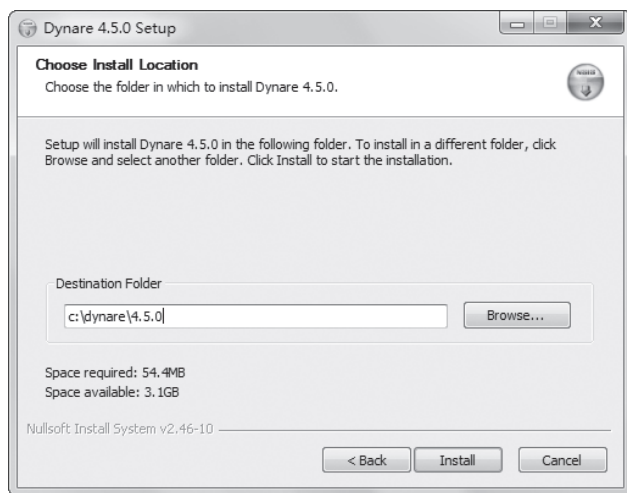


图 2.2 选择安装路径

选择好安装路径后，单击安装，等待文件抽取完毕即可完成 Dynare 安装。Dynare 安装完成后，必须要在 Matlab 或 Octave 中进行简单的配置，才能真正开始使用 Dynare。

① Mex 是 Matlab 与 Executable 的缩写，一般是由 C 或 Fortran 编写的外部程序，经 Matlab 编译后形成的二进制文件，能为 Matlab 自动装载并执行。这样做的好处是代码复用，提高效率。

② 其运行方式、mod 文件的语法等都和标准的 mod 文件有区别。Dynare++ 是一个独立的 exe 文件。

2.2 配置 Dynare

Dynare 的安装实际上只是把 Dynare 的基本组件拷贝到硬盘目录中，而真正的运行环境 Matlab 或 Octave 对其一无所知，因此必须告诉 Matlab 或 Octave，Dynare 软件的具体位置和必要的文件路径，以便 Matlab 或 Octave 搜寻，这称之为“配置 Dynare”。只有配置完成后，才能真正开始 Dynare 的使用。配置 Dynare 并不复杂，分为两种方法：单次配置与永久配置。此处以 Matlab 为例。

2.2.1 单次配置

单次配置是指在 Matlab 的命令行窗口 (Command Prompt) 直接输入诸如源代码 8 中的命令。

源代码8 单次配置

```
addpath c:\dynare\4.x.y\matlab
addpath c:\dynare\4.5.0\matlab
```

此处使用 Matlab 内置命令 `addpath`，将指定目录“`c:\dynare\4.5.0\matlab`”添加到 Matlab 搜寻路径列表 (Search Path) 的第一个位置^①，即最优先寻找的位置^②。注意，“`c:\dynare\4.5.0`”应为 Dynare 的真实安装目录。如果自定义了其他安装位置，请修改为正确的安装路径。这样做的缺点是，下次启动 Matlab 时，会丢失该次添加的目录，这也是为什么称之为“单次配置”的原因。但同时，这个缺点也是其优点，即 `addpath` 命令具有灵活性。也就是说，在编程时，能随时因需要用该命令来改变当前工作的 Dynare 版本^③，而无须担心改变原有目录配置设定。这是因为在下次启动 Matlab 时会自动恢复原有目录配置设定。

这里需要进一步解释 Matlab 搜寻目录的工作原理，才能更好地理解 Dynare 多版本管理的机制。假设 Matlab 搜寻目录中有两条记录，先后顺序如下：

```
c:\dynare\4.4.3\matlab
c:\dynare\4.5.0\matlab
```

如果调用 Dynare 中的相关命令，比如主命令 `dynare`，那么 Matlab 优先从第一个路

- ① 搜索路径列表除了在“设置路径”对话框中显示，还可以通过命令 `PATHDEF` 来查看。
- ② 在搜寻路径列表中，位置靠前，就意味着具有被优先搜寻到的机会。
- ③ 多版本需求经常碰到，特别是不同版本下编写的 `mod` 文件，可能会出现兼容性问题，因此运行某些别人或以前编写的模型文件时，很可能需要用到特定版本的 Dynare 软件包。

径中寻找该命令，即使用 4.4.3 版本的 Dynare，而非 4.5.0 版本。只有在第一条路径不正确或所要搜寻的命令不能在第一条路径中找到的情况下，才从第二条路径中寻找，依次类推。因此，直观上讲，哪个版本的路径在列表中靠前，就优先使用哪个版本。

2.2.2 永久配置

所谓永久配置，是相对于单次配置而言，一旦配置成功，就无须再次配置。如图 2.3 所示为 Matlab 设置搜寻路径对话框。

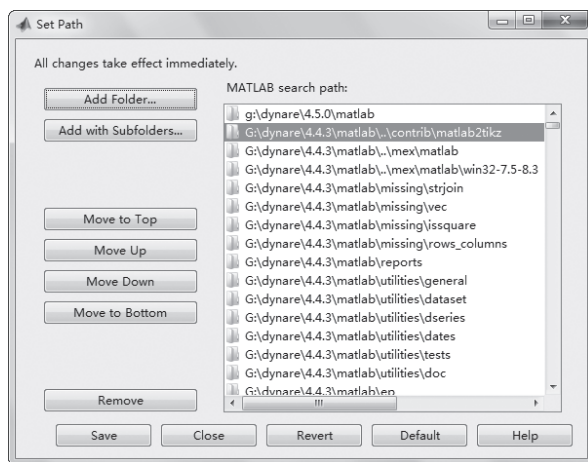


图 2.3 Matlab 设置搜寻路径对话框

永久性配置设定最简单的办法是通过 Matlab 的对话框来设置。以 Matlab R2012b 为例，在“Home”选项卡中找到“Set Path”按钮，单击打开“设置路径”对话框^①。单击添加目录“Add Folder...”按钮，定位到 Dynare 已安装目录，假设为“G:\dynare\4.5.0”，选择该文件夹，单击保存“Save”按钮并关闭，这样就完成了永久配置 Dynare 的任务。即使 Matlab 重新启动，该设定仍然有效，即仍然保证使用“G:\dynare\4.5.0”路径下的 Dynare 命令，而非其他路径或版本的命令。

此外，可以使用 Matlab 的内置命令“path”，即在命令行窗口中输入“path”命令来查看当前 Matlab 的搜寻目录，或者可使用“Set Path”对话框。Matlab 的“Set Path”对话框可以方便地管理当前搜寻目录列表，即移动或删除某条路径，设置完毕后保存并退出即可。

^① 也可以直接在命令行窗口中直接输入“pathool”命令，打开“设置路径”对话框。

2.3 执行和编辑 Dynare 文件

2.3.1 执行Mod文件

在编写好 Dynare 的模型文件 (*.mod, 也称之为 mod 文件) 之后, 就可以开始使用 Dynare 编译和求解了, 称之为执行模型文件。

执行模型文件非常简单, 只需两个简单的步骤: 第一, 首先确认 Matlab 当前工作目录 (Working Directory) 为该模型文件所在目录, 这是执行模型文件的第一步, 否则 Dynare 会编译报错, 无法找到当前模型文件。第二, 当模型文件无语法错误和运行时错误的情况下, 在命令行窗口使用 Dynare 的主命令 `dynare`, 即可完成执行。

1. 确认 Matlab 的当前工作目录

如果当前工作目录 (Working Directory) 不是模型文件所在的目录, 则需要将模型文件所在的目录设置为 Matlab 当前工作目录^①, 如图 2.4 所示。方法有两种: 一是单次设置; 二是永久设置。

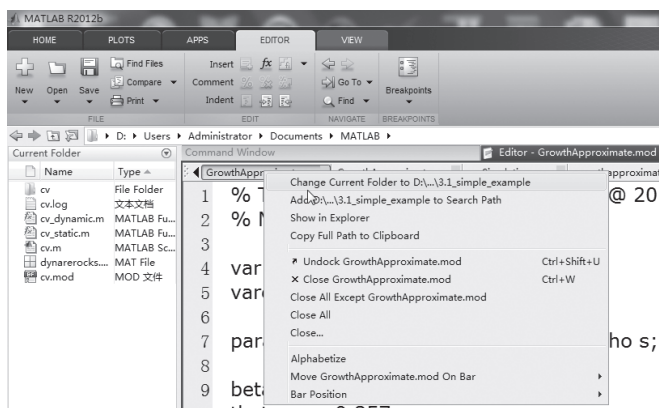


图 2.4 设置 mod 文件所在目录为 Matlab 当前工作目录

假设已编写好的一个模型文件: `GrowthApproximate.mod`, 位于目录 “D:\ABCD\Sources\Chap3_Dynare_Basics\3.1_simple_example” 下, 分别用以上两种方法操作如下。

方法一: 单项设置。即下次重新启动 Matlab 后, 需要重新设定, 当前设定不被保存到下次启动。

① 使用 Matlab 内置命令 `pwd` 来查看当前工作路径。

操作一：在命令行中直接使用 `cd` 命令，该方法需要准确输入路径名称，否则设置失败。`cd` 命令示例如下：

```
cd D:\ABCD\Sources\Chap3_Dynare_Basics\3.1_simple_example
```

操作二：使用鼠标操作，快捷方便。在编辑器(Editor)窗口，找到“GrowthApproximate.mod”文件对应的选项卡，右单击选项卡，选择“将当前目录设置为……”(Change Current Folder to ...)，即可将当前模型文件所在的目录设置为当前目录。

方法二：永久设置。一次设置，会一直被保存下来，重启 Matlab 仍然有效，直到下次更改位置：使用 `userpath` 命令和 `savepath` 命令，具体语法如下：

```
userpath('D:\ABCD\Sources\Chap3_Dynare_Basics\3.1_simple_example')
savepath
```

使用 `userpath` 命令会改变 Matlab 的搜寻路径列表 (PATHDEF)，使得当前路径被添加到搜寻路径列表第一行的位置。

2. 使用主命令：dynare

在工作路径设置完毕后，可在命令行窗口直接输入 `dynare` 命令后跟文件名即可（可省略 `mod` 后缀）。如果模型文件不存在语法错误和运行时错误，那么即可顺利执行。具体语法如下：

```
dynare GrowthApproximate.mod
dynare GrowthApproximate
```

关于 `dynare` 命令的选项和进一步说明，参见本书“3.5 Dynare 文件的预编译和运行原理”一节的内容。

2.3.2 编辑Mod文件

模型文件作为文本文件，可以在任何文本编辑器中编辑。推荐使用 Matlab 进行编辑，因为编辑完毕后，无须切换编辑平台即可立即执行和调试。在 Matlab 中编辑模型文件非常简单，只需要将当前目录切换到模型文件所在的目录，然后在当前目录 (Current Folder) 列表中双击该模型文件，或者在命令行窗口中输入“`edit filename.mod`”即可打开，进行各种编辑操作。

如果是新建模型文件，这里需要强调的是，Matlab 默认新建的是 `m` 文件。当保存或另存为时，在“保存类型”中选择“所有文件 (*.*)”，在“文件名称”中输入文件名时，一定要加后缀“`.mod`”。当保存不带后缀 (`.mod`) 的模型文件后，Dynare 编译此模型文件

时会提示当前目录中无法找到该文件。也就是说，模型文件一定要有“.mod”作为后缀。

模型文件作为 Dynare 的源文件，虽然有自己的语法和关键字，但很遗憾的是，到目前为止，Matlab 尚未完全支持模型 (*.mod) 文件的语法标记^①。而且 Dynare 官方或 Matlab 官方也未推出兼容 Dynare 模型文件的语法配置文件，使得在 Matlab 中编辑模型文件时，无法顺利实现全部智能语法标记和辅助编辑功能。但有一个替代的办法，能够使得 Matlab 部分支持模型文件的语法标记并继承着色方案，毕竟模型文件中的部分关键字来自于 Matlab。也就是说，通过简单的偏好设置，使得模型 (*.mod) 文件能够继承 m 文件 (*.m) 的着色方案以及智能输入或修改等功能。具体操作如下：单击 Preferences 按钮，打开对话框，找到“Editor/Debugger”下的“Language”选项，如图 2.5 所示。首先选择语言 MATLAB；然后添加扩展名 (File Extension)，“mod”，最后单击 OK 或 Apply 按钮后退出。关闭原来的模型文件，重新打开即可生效。

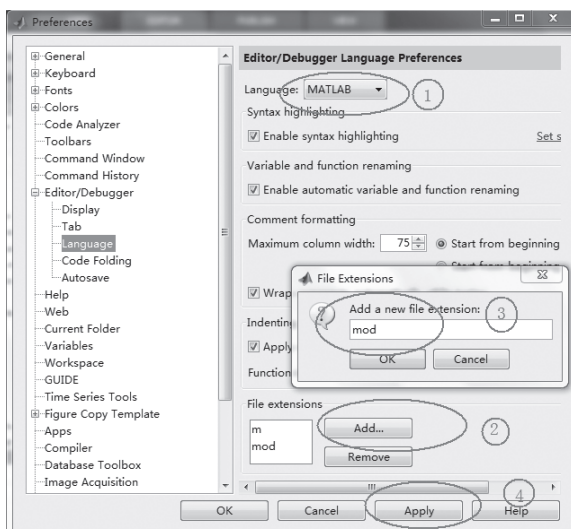


图 2.5 让 Matlab 部分支持模型文件的语法标记和智能修改功能

2.4 Dynare 多版本管理

Dynare 在不断完善和更新，版本号也不断变化，因此会面临安装多个 Dynare 版本

^① 相对比，m 文件会有各种类型的语法标记，包括着色、错误和警告提醒、智能输入等辅助编辑功能，极大地提升编辑效率和感受。

的尴尬。默认情况下，在安装新版本的 Dynare 时，会自动创建一个以当前版本号为名称的文件夹，并在此文件夹下进行新版本安装，因此无须担心会覆盖以前的旧版本。图 2.6 显示了 4 个不同版本 Dynare 的安装目录示意图。

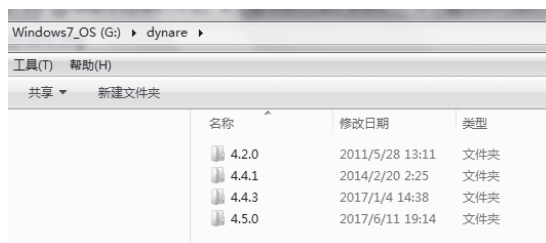


图 2.6 Dynare 的多版本管理

在某些情况下，老版本的某些功能可能会更加稳定或更符合使用习惯。某些运行于特定版本下的模型文件，即使安装了新版本也可能有使用老版本的需求。因此在使用时，如何在多版本之间进行切换呢？这个问题在本书“2.2 配置 Dynare”中已经涉及，答案非常简单，就是使用 `addpath` 命令将想要使用版本的 Dynare 的安装路径添加到搜寻路径列表即可（具体参考源代码 8）。注意，`addpath` 命令既可在命令行窗口使用，也可在编程时写入代码中，随时灵活切换版本。

2.5 获取 DYNARE 帮助

及时、准确地获得 Dynare 的使用帮助，无论是对初学者还是熟练的使用者来说，都是非常重要的。获取 Dynare 的使用帮助有多种渠道。

2.5.1 官方帮助文档

查阅官方帮助文档是解决问题最重要、最便捷、最权威的途径。Dynare 的官方帮助文档较多。其最有用的要数参考手册 (Dynare Reference Manual)、用户入门帮助 (UserGuide.pdf)。其中 Dynare Reference Manual 既有在线版本^①，也有本地版本。本地版本有 PDF 版本，也有 HTML 版本 (网页版)^②。Dynare Reference Manual 以功能为主题进行介绍，内容涵

① 在线版本的地址为：http://www.dynare.org/manual/index_toc.html#SEC_Contents。

② 本地 Dynare Reference Manual 网页版的地址是：安装盘符 \dynare\4.5.0\doc\dynare.html\index.html，其中 4.5.0 是对应的版本号，请根据自己安装的版本替换。

盖各个功能，甚至包括各种算法背后对应的参考文献、命令与变量的索引目录等，查阅起来非常方便。此外，Dynare 在安装目录中也提供了不少的例子，可供学习使用。值得注意的是，查阅的文档版本号应和使用的 Dynare 版本号一致，这样能避免因版本号差异造成的各种问题。

此外，Dynare 提供了一个在线查看其 Matlab 所有源命令的工具：M2HTML(即将 m 文件转换为网页文件的工具)^①。该工具提供了安装子目录 matlab 下所有文件的列表，以及这些源命令文件的代码和相互调用关系，对于理解源文件非常有帮助。

2.5.2 官方帮助论坛

Dynare 官方提供了一个在线论坛：Dynare Forum^②，供全世界的用户使用。该论坛由 Dynare 项目核心开发团队的成员 Johannes Pfeifer 负责维护，回答用户提出的问题。因此，一般用户的使用问题，都能够及时在这里得到最权威的解答。在碰到棘手的问题时，建议大家首先选此论坛。

此外，Dynare 论坛中还提供了很多经典文献对应的 Dynare 模型文件，并不断更新，可供用户下载学习和使用，非常难得。

2.5.3 其他在线方式

首先推荐大家去知名学者的个人网站获取相关的学习资料和 Dynare 的模型文件代码。如：Dynare 二阶扰动算法的创始人之一 Stephanie Schmitt-Grohé^③；三阶扰动算法的创始人之一 Jesús Fernández-Villaverde^④；2011 年诺贝尔经济学奖获得者、理性预期的创始人 Thomas Sargent^⑤；2011 年诺贝尔经济学奖获得者、VAR 理论的主要创始人 Sims Christopher^⑥ 等人的个人网站，都有大量的文献、源代码甚至视频资料可供下载学习使用。

Dynare 官方每年在不同的地方举办 Dynare 短期研讨班^⑦，由核心开发成员讲解 Dynare 的使用，提供讲义、代码等资料，是不可多得的学习机会。

① <http://www.dynare.org/dynare-matlab-m2html/>.

② Dynare 论坛的地址为：<https://forum.dynare.org/>，应首先注册，然后才可以登录提问。

③ Stephanie Schmitt-Grohé 的个人网页：<http://www.columbia.edu/~ss3501/>。

④ Jesús Fernández-Villaverde 的个人网页：<http://economics.sas.upenn.edu/~jesusfv/>。

⑤ Thomas Sargent 的个人网页：<http://www.tomsargent.com/>。

⑥ Sims Christopher 的个人网页：<http://www.princeton.edu/~sims/>。

⑦ 2013 年 11 月，Dynare 短期研讨班在上海财经大学经济学院举办，由 Dynare 核心团队成员 Michel Juillard 和 Sébastien Villemot 共同主讲。

国内著名的经济论坛(如经管之家论坛)活跃着一批研究 DSGE 模型的国内外学者和同人,提供了非常丰富的学习资料,并开设了 Dynare 的答疑专栏^①。

此外,本书“6.4 常见的 Dynare 运行错误”一节也提供了部分运行错误的解释,可供参考。

最后,鼓励初学者在遇到错误或其他困难时,应该首先“自救”,即根据 Dynare 的错误反馈信息,自行查阅官方提供的 Dynare Reference Manual 和 UserGuide 或在线搜索,这应该是最快捷的解决方式。在自己经过查阅后仍然无法解决的问题,可通过在线方式进行提问或其他方式寻求立即的帮助。

① 答疑专栏地址: <http://bbs.pinggu.org/forum.php?mod=forumdisplay&fid=114&filter=typeid&typeid=10943>。

3

Dynare 基本应用

本章介绍 Dynare 软件的基本语法及其简单应用。通过本章的学习，读者应该能够对 Dynare 变量的分类、排序与书写规范、模型 (*.mod) 文件的基本结构、模型文件的预编译和运行原理、确定性和随机模拟的基本原理、Dynare 中贝叶斯估计的基本方法、Dynare 的解表示和结果表示等，有较为深入的认识，能够自行编写简单的 DSGE 模型的模型文件，并能读懂 Dynare 的输出结果，为进一步深入应用 Dynare 做好铺垫。

3.1 DSGE 模型：一个简单的例子

本小节介绍一个简单的 RBC 模型^①，为后续章节做好铺垫，以示例“如何在 Dynare 中编写一个完整的模型文件”，熟悉 Dynare 的运行原理以及如何进行后续的各种分析。

一个代表性家庭选择消费 C_t 和劳动 L_t 来最大化其终身贴现效用：

$$\max_{\{C_t, L_t\}_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \frac{(C_t^\theta (1-L_t)^{1-\theta})^{1-\tau}}{1-\tau} \quad (3.1.1)$$

其中， θ, τ 为参数； $1 > \beta > 0$ 为贴现因子；即期效用函数关于消费和劳动不可分。最大化问题面临的资源约束为

$$C_t + I_t = \exp(Z_t) K_t^\alpha L_t^{1-\alpha} \quad (3.1.2)$$

其中， $0 < \alpha < 1$ 为产出的资本份额， I_t 为投资，资本存量 K_t 遵循经典的积累方程：

$$K_{t+1} = I_t + (1-\delta)K_t \quad (3.1.3)$$

其中， $0 < \delta < 1$ 为折旧率参数，外生冲击 Z_t 服从简单的 AR(1) 过程：

$$Z_t = \rho Z_{t-1} + s\epsilon_t, \quad \epsilon_t \sim N(0,1) \quad (3.1.4)$$

^① 该例子来源于 Practicing Dynare，由包括 Thomas Sargent 在内的 6 位作者编写的 notes，2010.12.4，可在 Sargent 的个人网站上下载到：http://www.tomsargent.com/rational_expectations.html。

其中, $s > 0$ 为扰动参数, $0 < \rho < 1$ 为外生冲击持续性参数, 越接近于 1, 持续性越强。效用最大化问题关于消费和劳动的一阶条件如下:

$$\frac{\left(C_t^\theta (1-L_t)^{1-\theta}\right)^{1-\tau}}{C_t} = \beta E_t \left[\frac{\left(C_{t+1}^\theta (1-L_{t+1})^{1-\theta}\right)^{1-\tau}}{C_{t+1}} (1 + \alpha \exp(Z_t) K_{t+1}^{\alpha-1} L_{t+1}^{1-\alpha} - \delta) \right] \quad (3.1.5)$$

$$\frac{1-\theta}{\theta} \frac{C_t}{1-L_t} = (1-\alpha) \exp(Z_t) K_t^\alpha L_t^{-\alpha} \quad (3.1.6)$$

因此, 模型的均衡由 4 个变量 $\{C_t, L_t, K_t, Z_t\}$ 和 4 个均衡条件 Euler 方程 (3.1.5)、劳动供给方程 (3.1.6)、资源约束方程 (3.1.2)^① 和技术变量的 AR(1) 过程 (3.1.4) 组成。

3.2 Dynare 内生变量的分类和书写规范

3.2.1 Dynare 内生变量的分类

在 Dynare 中, 内生变量根据时间下标的不同被分为 4 类, 分别为静态 (Static) 变量、前向 (Purely Forward-looking) 变量、后向 (Purely Backward-looking, Pre-determined) 变量和混合 (Mixed) 变量, 如表 3.1 所示。

表 3.1 Dynare 内生变量的分类

内生变量类型	定义	内生变量个数存储位置
静态变量	仅仅出现时间下标 t	M_.nstatic
前向变量	仅仅出现时间下标 t 和 $t+1$	M_.nfwr
后向变量	仅仅出现时间下标 t 和 $t-1$	M_.npred
混合变量	同时出现时间下标 t , $t-1$ 和 $t+1$	M_.nboth

数据来源: 作者自行总结。

针对第 1 节的简单模型, 和上述变量类型的定义, 可对模型中的 4 个变量进行归类。首先是消费和劳动。两者都出现时间下标 t 和 $t+1$, 而且没有出现 $t-1$ 期下标, 因此属于前向变量。对于资本存量则比较特殊。由资本积累方程可知, 虽然资本存量出现了 $t+1$

① 资源约束方程中的投资变量可由资本积累方程替换, 因此可消去投资变量。同时也可以称之为资本积累方程。

期下标，但是资本存量是在 t 期末决定，属于预先决定的变量，因此应归为后向变量^①。技术变量属于标准的后向变量。

当效用函数为消费和劳动的可加可分函数时，劳动通常为静态变量。但当效用函数关于消费和劳动不可分，劳动也将变成非静态变量，即动态变量。此外，如果在效用函数中引入消费习惯设定，则会使得消费变成混合变量。

通常情况下，内生变量属于且只能属于上述 4 种中的某一类，也就是说上述分类的划分是内生变量类型的一个完全划分。因此上述 4 类变量个数之和为模型所有内生变量的个数 (M_endo_nbr)，即

$$M_endo_nbr \equiv M_nstatic + M_nfwrđ + M_npred + M_nboth.$$

一般说来，模型中内生状态变量的个数为后向变量和混合变量的个数之和。在 Dynare 中内生状态变量的个数存储在 M_nspred 中，因此：

$$M_nspred \equiv M_npred + M_nboth.$$

此外，相对于静态变量而言，其余 3 类变量也被统称为动态变量 (Dynamic Variables)，其个数存储在 $M_ndynamic$ ，因此：

$$M_ndynamic \equiv M_nfwrđ + M_npred + M_nboth.$$

4 类不同的变量以及状态变量、动态变量之间的逻辑关系，如图 3.1 所示。

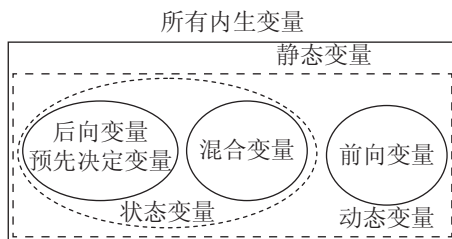


图 3.1 4 类不同的变量以及状态变量、动态变量之间的逻辑关系

3.2.2 Dynare 内生变量的书写规范

1. 命名规范

由于 Dynare 运行于 Matlab 之上^②，内生变量的命名规范一般情况下要遵循 Matlab 的

① 这一点需要特别注意，在书写 Dynare 的 mod 文件时，要滞后一期，即 K_{t+1} 用写成 K_t 的形式，而 K_t 要写成 K_{t-1} 的形式，这是很多初学者容易犯的错误。也正是因为如此，很多文献干脆在资本积累方程中直接写成滞后一期的形式，下一小节关于时间下标的说明会更详细地解释这一原因。

② 此处未考虑其他运行平台，如 Octave。

命名规范。在变量命名时，尽量遵循如下规范，否则可能会造成意想不到的错误或后果。

第一，尽量不要使用 Matlab 和 Dynare 内置函数的名字作为变量或参数的名字。如 Dynare 使用“Sigma_e”作为存储外生变量方差的矩阵名称，应避免使用该名称。在自定义的变量名称前使用字母 v，在参数名称前使用字母 c，都是良好的编程习惯。比如，vConsump、vLabor、cAlpha、cBeta 等。注意变量不能以数字开头，比如“2Consump”是非法变量名^①。

第二，不要在变量或参数名称中包含特殊字符，如 @、#、! 等，但下划线“_”可以使用，而且提倡使用。

第三，Dynare 是字母大小写敏感的。比如 z 和 Z 是不同的名字，指代不同的对象。

第四，尽量不要使用字母 i，避免和 Matlab 中复数单位 i 混淆（但目前暂时不会造成技术上的问题）。在声明利率或投资变量时，可以使用 R、I 或 inv 等字母或字符串表示。

2. 时间下标约定

在 Dynare 中，变量的时间下标是根据该变量决定时期来确定。一般说来， t 期变量（流量）由 $t-1$ 期的变量（内生和外生变量）和 t 期初实现的外生冲击共同决定。因此，从这个意义上讲，存量变量（Stock Variables）在 t 期末被决定，或者说在流量变量决定后再决定。一个典型的例子就是资本存量。根据经典的资本积累方程：

$$K_{t+1} = I_t + (1 - \delta)K_t$$

投资 I_t 在 t 期中被决定，因此 K_{t+1} 形式上虽然是 $t+1$ 期变量，实则是 $t+1$ 期预先决定的变量，即 t 期末被决定。基于此原因，上述经典方程在某些文献中也被写作：

$$K_t = I_t + (1 - \delta)K_{t-1}$$

因此，时期 t 可虚拟地看成由 3 部分组成：期初（外生冲击实现）、期中（当期变量决定）、期末（存量变量决定）。

在 Dynare 文件的书写中，非 t 期的变量书写时，需要进行简单处理，具体参见表 3.2。

表 3.2 Dynare 内生变量书写规范

变量形式	书写形式
x_{t+2}	$x(+2)$, $x(2)$
x_{t+1}	$x(+1)$, $x(1)$
x_t	x
x_{t-1}	$x(-1)$
x_{t-2}	$x(-2)$

① 虽然某些希腊字母，如 alpha，在 Matlab 是内置函数，但在 Dynare 中将其声明为参数的名称，并不会造成问题。

只有少数情况下，变量会出现 $t+2$ 期或 $t-2$ 期下标的情况。这时为了技术处理上的便利，Dynare 会引入辅助变量 (Auxiliary Variables)，将 $t+2$ 期或 $t-2$ 期变量转换为 $t+1$ 期或 $t-1$ 期变量，方法如下：

$$y_t \equiv x_{t+1} \rightarrow y_{t+1} = x_{t+2}$$

$$y_t \equiv x_{t-1} \rightarrow y_{t-1} = x_{t-2}$$

因此，最终的模型文件只会出现 t 期、 $t+1$ 期和 $t-1$ 期共 3 种变量。

3.2.3 Dynare 内生变量的排序

在 Dynare 中，有两种内生变量的排序，分别用于不同的场合。

第一种为声明顺序 (Declaration Order)，即在模型文件书写时，使用 `var` 命令进行变量声明时，变量出现的先后次序。对于第一节中的简单模型，使用 `var` 命令声明变量如下：

```
var c k lab z;
```

此处，声明排序为消费、资本存量、劳动和技术变量。当然完全可以声明不同的顺序。在模型文件运行后，声明顺序被存储在 `M_endo_names` 中。

第二种为决策规则顺序 (Decision Rule Order, DR Order)。这种顺序有别于声明顺序，被用于决策规则中，即政策函数的表达式中。决策规则顺序的基本定义为：首先是静态变量，其次是后向变量，然后是混合变量，最后是前向变量。在每一类变量中，变量以声明顺序排列。针对 `var` 命令的声明，并结合第 1 节中的均衡条件，可知该模型中没有静态变量和混合变量，后向变量 (预先决定) 为资本存量 K_t 和技术冲击 Z_t 变量，然后是前向变量消费 C_t 和劳动 L_t ，此时决策规则顺序为资本存量、技术冲击、消费和劳动。其中，资本存量先于技术冲击，消费先于劳动，这是组内 (类别内) 按照声明顺序排列的缘故。

在 Dynare 中，声明顺序和决策规则顺序之间存在一个简单的映射 (置换, Permutation)。这个映射存储于数组 `oo_dr.order_var` 中，即决策规则顺序中的第 n 个变量，则对应着声明顺序中的第 `oo_dr.order_var(n)` 个变量。比如当 $n=1$ 时，`oo_dr.order_var(1) = 2`，决策规则顺序中的第一个变量为声明顺序中的第二个变量，即资本存量。

当内生变量的个数较多时，可以使用如下的命令列示出声明顺序、两者之间的置换以及决策规则顺序。第一列表示声明顺序，第二列表示声明顺序与决策规则顺序之间的映射，第三列表示决策规则顺序。

```
[M_.endo_names int2str(oo_.dr.order_var) M_.endo_names(oo_.dr.order_
var,:)]
ans =
c      2      k
k      4      z
lab    1      c
z      3      lab
```

注意：上述命令的使用需要在运行完模型文件之后，内存变量尚未清除之前，特别是结构变量 `M_` 未被清除之前。

3.3 Dynare 文件基本结构

由于 Dynare 语法规则较多，特别是某些关键命令的选项 (options) 较为专业且复杂，而且随着版本的更新，会新增或删除一些命令。本节并不打算详细介绍每一条语法，为了节省篇幅，此处仅简单介绍常用的语法规则，使大家能够编写可运行的、简单的 Dynare 文件^①。在本章和后续的章节中，也会陆续介绍相关的命令及其语法结构。本节介绍 Dynare 模型文件的基本结构，并以此为开端，介绍相关语法和命令。

一般说来，模型文件 (*.mod) 由 5 大部分组成：前导 (Preamble) 部分、模型 (Model) 声明部分、稳态值或初始值 (Steady state or initial values) 声明部分、外生冲击 (Shocks) 声明部分和模型计算 (Computation) 部分。

3.3.1 前导部分

前导部分主要完成模型声明前的准备工作，包括 3 个部分：内生变量声明 (var 命令)、外生变量声明 (varexo) 以及参数声明 (parameters) 与赋值。针对第 1 节中的模型，其对应模型文件的前导部分可编写为源代码 9，具体如下所述。

源代码9 Mod文件的前导部分

```
var c k lab z; % 内生变量的声明，使用var命令，使用空格或逗号隔开
varexo e; % 外生变量的声明，使用varexo命令
parameters beta theta delta alpha tau rho s; %参数的声明，使用parameters命令

%参数的赋值，直接书写表达式即可；
beta = 0.987;
theta = 0.357;
delta = 0.012;
```

① 若详细介绍 Dynare 的语法，将会花费整本书的篇幅。强烈推荐大家查阅 Dynare 的在线帮助文档，方便而且更新较为及时：http://www.dynare.org/manual/index_toc.html#SEC_Contents。

```
alpha    = 0.4;
tau      = 2;
rho      = 0.95;
s        = 0.007;
```

值得一提的是，前导部分的 3 个声明命令，在同一个模型文件中可多次使用，Dynare 会自动将其合并处理。多次重复使用的好处在于能提高模型文件的可读性，将不同类别或含义的变量或参数放在不同的声明下，以示区分。

3.3.2 模型部分

在前导部分声明完毕后，即可声明模型部分，这也是模型文件的核心部分，即模型均衡条件部分。该部分以“`model;`”开始，以“`end;`”结束^①，在中间输入模型的均衡条件^②，每个均衡条件以分号“`;`”结束。当均衡条件较长时，可以跨行，分号不能在行间使用，只能在结尾使用一次。源代码 10 列示了第一节中 RBC 模型的 4 个均衡条件。

源代码 10 Mod文件的模型声明部分

```
model;
% define model-local variables to improve readability;
# mar_c= (c^theta*(1-lab)^(1-theta))^(1-tau);
# mar_c1=(c(+1)^theta*(1-lab(+1))^(1-theta))^(1-tau);

%(1) Euler equation
[name = 'Euler Equation']
mar_c/c=beta*(mar_c1/c(+1))*(1+alpha*exp(z)*k(-1)^(alpha-1)*lab^(1-alpha)-delta);

%(2) labor supply equation
c=theta/(1-theta)*(1-alpha)*exp(z)*k(-1)^alpha*lab^(-alpha)*(1-lab);

%(3) resource constraint or capital accumulation equation
k=exp(z)*k(-1)^alpha*lab^(1-alpha)-c+(1-delta)*k(-1);

%(4) technology shock
z=rho*z(-1)+e;
end;
```

① 通常情况下，以模块 (block) 出现的结构，都由两个关键字加以界定首尾，并以 `end` 关键字加分号结束。如模型声明模块、初值与终值模块、外生冲击模块等。此外，每一个命令结束或表达式结束后，必须以分号结尾来告诉 Dynare 该命令或表达式已经结束，否则编译报错。

② 均衡条件的输入可以有两种方式：“`a=b;`”或“`a-b;`”，此时省略“`=0;`”。注意句末分号不可少。

为了提高源代码的可读性，建议在每个均衡方程前面添加注释和编号，以“%”或“//”开头表示单行注释，也是最常用的注释方式。此外，还可以使用系统标签 (tag) 对均衡条件进行标识。这样做的用处是，后续进行稳态计算调试时，resid 命令能够显示出该标签名称^①，将名称和均衡条件对应起来，从而提高调试效率，否则只会显示该均衡条件的编号。系统标签的语法非常简单，首先使用中括号 “[]”，然后使用关键字 name，后跟标签内容，并用单引号括起来，如源代码 10 中第一个均衡条件的示例。

在上述模型声明部分中，还定义了模型局部变量 (Model Local Variables)，即以“#”开头的两个变量，且名称和前导部分声明的变量与参数的名称不能相同：“mar_c”与“mar_c1”。在使用局部变量时，无须加“#”字符。定义局部变量的根本目的在于简化均衡条件的书写，提高可读性。特别当均衡条件中存在多个重复的表达式时，定义局部变量不仅能够提高书写效率，而且还能够提高可读性，减少差错。局部变量的唯一作用就是文本替换 (Text Substitution)。也就是说，均衡条件中的所有局部变量都被 Dynare 自动替换为其对应的字符串，除此之外，局部变量没有其他作用。这同时也意味着，局部变量无法在模型部分之外被使用，即局部可见。

此外，模型声明命令“model;”有 8 个选项，最常用的是线性 linear 选项，使用方式为“model(linear);”，即模型均衡条件被声明为线性，这也是初学时常遇到的选项。linear 表示线性，即当输入的均衡条件为线性化的均衡条件时使用该选项。需要注意的是，使用该选项必须“名副其实”。也就是说，均衡条件必须是手动或其他方式线性化后的结果，否则会出现编译错误。使用 linear 选项后，由于模型是线性化模型，可无须指定稳态值，因为稳态值本身就是 0。

3.3.3 稳态或初值部分与外生冲击

稳态值和初值的声明是模型文件的重要组成部分，这是因为 Dynare 采取的扰动算法的基本要求。初值的声明其最终目的是为了 Let Dynare 找到稳态值，然后进行下一步的运算。关于稳态和初值的声明会在本书“3.8.2 稳态求解命令：steady”一节中详细介绍，而外生冲击部分的设定会在本书“3.8.3 确定性模拟”一节中详细介绍。此处仅简单介绍随机模拟的情况，而确定性模拟的情况要稍微复杂。源代码 11 示例了如何简单地设定初始值、外生冲击的方差，计算稳态并同时计算各静态方程的残差，此时各内生变量取值稳态。

^① resid 命令是计算每个均衡条件对应的静态方程的残差。静态方程是抹去时间下标后对应的方程，resid 命令计算方程两端计算结果的差异 (残差)，因此在计算稳态时非常有用，残差为 0 时即找到稳态。

源代码11 Mod文件的初值、稳态与外生冲击部分

```

initval; %初值模块, 设定内生变量和外生变量的初值, 以此为起点计算稳态值
k = 1; c = 1; lab = 0.3; z = 0; e = 0;
end;

shocks; %外生冲击设定, 直接设定外生冲击的方差
var e=s^2;
end;

steady; %稳态计算
resid; %给定内生变量的取值(通常为稳态), 计算均衡条件对应的静态方程的残差
STEADY-STATE RESULTS:
c 1.49163
k 29.2885
lab 0.291593
z 0

Residuals of the static equations:
Equation number 1 : -1.5417e-07 : Euler equation
Equation number 2 : 3.0544e-06
Equation number 3 : 2.6682e-06
Equation number 4 : 0

```

在模型编译并运行后, Dynare 会显示稳态计算的结果, 以及静态方程的残差。此时, 第一个方程显示了“Euler Equation”, 这正是在模型声明时使用系统标签 (tag) 的结果。一般说来, 只有静态方程的残差为 0, 才能说明已经获取了稳态值。但上述静态方程的残差除第 4 个方程外, 都不为 0。这是为什么呢? 这是由于计算误差导致的。由于此处只设定了内生变量的初始值, 而此初始值本身离稳态值有一定的距离, 因此 Dynare 会使用内置非线性算法计算模型稳态, 并设定迭代容忍误差。当计算误差位于容忍度以内时, 即停止计算, 认为已经找到稳态值。因此, 静态方程两边会存在轻微误差。可看出前 3 个均衡条件对应的残差的数量级都在 $1E-6$, 比较接近于 0。

如果当内生变量个数较多时设定某一初值, 然后让 Dynare 自行计算稳态值, 往往是失败的, 要么因为初始值离稳态太远, 要么模型过于复杂, 在设定的初值下无法找到稳态 (如迭代次数、容忍度等)。因此在多数情况下, 建议手动编程计算稳态, 然后告诉 Dynare 即可, 这样效率较高。本书后序会详细介绍关于稳态计算语法设置的几种方式。

3.3.4 计算部分

计算部分是模型最核心的部分之一, 也是最复杂的部分。所谓计算, 就是在各参数

都已知，而且各内生变量的稳态都已求出的情形下^①，求解决策规则。常用的有 3 个计算命令：确定性模拟 (Simul)、随机模拟 (Stoch_simul) 与估计 (Estimation)。

3 种不同的计算命令，各有区别，在不同的场合下使用。粗略地说，确定性和随机性模拟是针对外生冲击而言的。确定性是指外生冲击的发生时刻和大小都已经预先知晓并确定，在 Dynare 中通过外生冲击模块设定。而随机性是指外生冲击的发生时刻或大小是未知的、随机的。通常的做法是从正态分布中随机抽取，每个时刻的大小是未知的。在 Dynare 中也是通过外生冲击模块设定。估计是指根据观测或统计数据，首先使用估计技术估计未知的参数值，然后使用估计后的参数值进行求解，此时其作用相当于随机模拟。确定性模拟将会在本书“3.8.3 确定性模拟”一节中详细介绍，随机模拟会在“3.9 随机模拟分析：stoch_simul”一节中详细介绍，估计命令会在“3.10 参数估计简介”一节中详细介绍。此处简单介绍随机模拟的设定。

源代码 12 列示了随机模拟命令 stoch_simul 及其 3 个常用的选项，然后使用 Dynare 内置命令将模拟结果保存为自定义名称的 Matlab 矩阵文件 (mat 文件)：simudata.mat，便于事后分析使用。

源代码 12 Mod 文件的计算部分

```
%if periods not specify, there will be no simulations.
stoch_simul(periods=1000,irf=40,order=1);

%save the simulated data to file
dynasave('simudata.mat');
```

然后来简单解释 stoch_simul 命令的 3 个选项。periods 表示随机模拟的期数；irf 用来指定脉冲响应的计算期数；order 表示求解的阶数，即指定一阶、二阶或三阶求解算法。因此，此处随机模拟命令的含义是使用线性求解算法求解模型，并模拟 1 000 期内生变量的样本，计算 40 期脉冲响应函数。

至此，已经完成了对模型文件的全部结构描述^②。相信读者已经能够独立地完成一个简单 RBC 模型的 Dynare 编程。

本书“6.5 Dynare 宏命令编程示例”将结合宏语言编程，讲解如何使用宏命令 (@#include) 来进行模块化编程，即将模型文件的组成部分分割成物理上独立的小模型文件，以处理复杂的模型编程问题。这样做的目的是提高代码的可读性和可维护性，并减少差错。

① 或指定内生变量在其他位置的情况下。

② 完整版 mod 源文件：\Sources\Chap3_Dynare_Basics\3.1_simple_example\GrowthApproximate.mod。

3.4 内生变量的表达形式：level or log-level

模型均衡条件的输入是 Dynare 编程的一个核心，因为均衡条件代表了整个模型。而多数情况下，均衡条件是参数和内生变量的非线性组合。不同的内生变量的表达形式，决定了运行结果的不同含义，因此要根据研究问题的实际需要选择不同的表达形式。

一般说来，内生变量的表达形式有 3 种：水平 (level) 形式、对数水平 (log-level) 形式和离差 (deviation) 形式。水平形式即变量本身，不做任何形式的变动；而对数水平即取对数，以对数形式进入均衡条件；离差形式是线性形式。

在本书“1.1.2 线性化与对数线性化”一节中，已经介绍了内生变量的离差形式。一阶线性是学习 DSGE 模型较好的出发点，形式简单，而且能够建立变量之间的直觉关系，有利于理解经济问题本身。但是，并不建议对每一个模型的均衡条件都进行线性化，然后输入线性化的均衡条件到模型文件中。原因有如下两点。

第一，手动线性化每一个均衡条件，很容易出错。而且在某些情况下，几乎是不可能完成的任务。对于中等规模或大型的 DSGE 模型，均衡条件的个数超过 100 个甚至 500 个以上，手动线性化是不大可能完成的任务。即使能够完成，不仅耗费大量的时间和精力，而且差错率也会较高，可谓“出力不讨好”。

第二，Dynare 可以帮你完成线性化。只须在计算命令中加入求解阶数选项 (order=1)，即可实现所有均衡条件的线性化，简单高效。

鉴于以上原因，推荐大家使用水平形式或对数水平形式。在源代码 10 中，各变量均使用了其水平形式，即变量本身。

下面着重介绍对数水平形式。这是因为对数水平形式，对于脉冲响应图形的显示和解释较为重要，因而推荐大家在实际研究中使用该种表示形式。首先通过第 1 节的例子来简单解释对数水平形式的具体内涵和使用方法。以均衡条件 (3.1.6) 为例，说明消费、劳动以及资本存量如何使用对数水平形式^①：

$$\frac{1-\theta}{\theta} \frac{C_t}{1-L_t} = (1-\alpha) \exp(Z_t) K_t^\alpha L_t^{1-\alpha}$$

进行如下的等价变形：

$$\frac{1-\theta}{\theta} \frac{\exp[\log(C_t)]}{1-\exp[\log(L_t)]} = (1-\alpha) \exp(Z_t) \{\exp[\log(K_t)]\}^\alpha \{\exp[\log(L_t)]\}^{1-\alpha}$$

① 技术变量已经是对数水平形式。

如果定义3个新变量为水平变量的对数,即对数水平变量(此处添加字母 v 表示对数变量)

$$vC_t \equiv \log(C_t), vL_t \equiv \log(L_t), vK_t \equiv \log(K_t)$$

那么上述均衡条件可进一步表示为如下形式:

$$\frac{1-\theta}{\theta} \frac{\exp(vC_t)}{1-\exp(vL_t)} = (1-\alpha)\exp(Z_t)\exp(\alpha \times vK_t)\exp(-\alpha \times vL_t)$$

在不引起混淆的情况下,可使用同一个变量名称,进而将上述对数水平形式的均衡条件表达为

$$\frac{1-\theta}{\theta} \frac{\exp(C_t)}{1-\exp(L_t)} = (1-\alpha)\exp(Z_t)\exp(\alpha \times K_t)\exp(-\alpha \times L_t)$$

因此,在实际操作时,只需要将相应的水平变量替换为 $\exp(\cdot)$ 形式,即可完成从水平形式到对数水平形式的转变。此时, Dynare 模型部分的输入形式如源代码 13 所示^①。

源代码13 Mod文件的模型声明部分

```

model;
.....
%(2) labor supply equation
exp(c)=theta/(1-theta)*(1-alpha)*exp(z)*exp(k(-1))^alpha*exp(lab)^(-
alpha)*(1-exp(lab));
.....
end;

```

需要注意的是,此时的变量虽然名称相同,但已经不是原来的变量,而是取对数以后的变量。因此,模型中变量的含义发生了变化,因此 Dynare 的运行结果也发生了变化。如: Dynare 输出的稳态值,是对数稳态值而非水平变量的稳态值;各变量之间的相关性变成了对数水平变量之间的相关性,而非水平变量;模拟的样本也将是对数水平变量,使用时要特别注意。最终决策规则中的变量也是对数水平变量。

为了进一步搞清楚使用对数水平变量带来的影响,有必要回顾 Dynare 决策规则的形式。以一阶为例, Dynare 的决策规则的形式可由(1.1.4)表示:

$$y_t = \bar{y} + g_y(y_{t-1} - \bar{y}) + g_u u_t$$

其中, \bar{y} 为稳态值。如果上述决策规则的变量 y_t 是对数水平变量,那么根据对数线性化变量的定义,有 $\hat{y}_t = y_t - \bar{y}$ 。此时上述决策规则中的变量其实已经变成了离差形式[回顾式(1.1.13)],即

$$\hat{y}_t = g_y \hat{y}_{t-1} + g_u u_t \quad (3.4.1)$$

① 对数水平形式源代码: \Sources\Chap3_Dynare_Basics\3.1_simple_example\GrowthApproximate_exp.mod。

这也正是为什么在模型文件中经常使用对数水平变量的原因所在。在此情况下，脉冲响应图形就变成了对稳态偏离的百分比，是相对偏离，而非绝对偏离^①，因而具有经济含义。此外，这也使得脉冲响应图形不因内生变量的量纲不同而出现图形比例不协调的问题，可谓“一举多得”。

图 3.2 和图 3.3 分别显示了第 1 节中的 RBC 模型，在水平和对数水平形式下，对一单位（一标准差）正向外生技术冲击的脉冲响应图。从图 3.2 中可看到，不同变量的脉冲响应的大小差异较大，最大差异接近 100 倍。而图 3.3 中，各变量的脉冲响应都在一个数量级上，表现匀称。

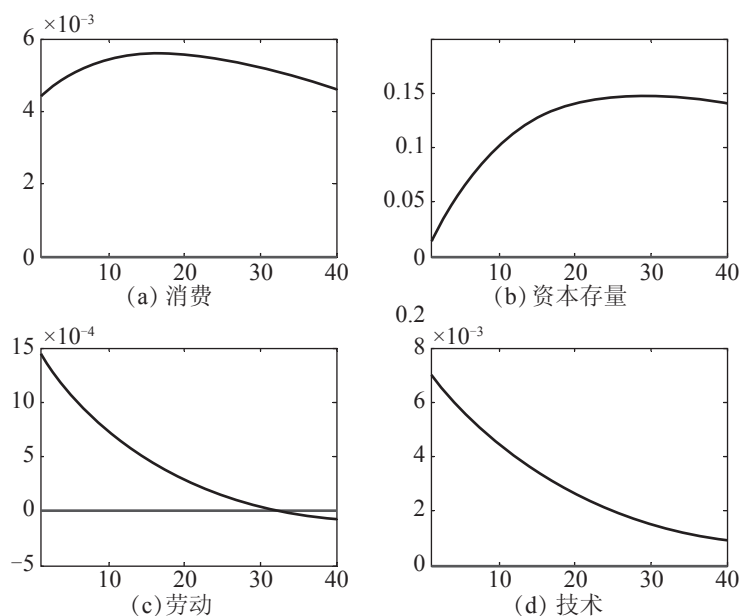


图 3.2 水平形式下的脉冲响应图

以消费为例来说明两者的定量差异。在图 3.2 中，第一期消费对于一单位外生冲击做出正向反应，绝对偏离稳态 0.004 409，而消费的稳态为 1.491 63，因此相对偏离为

$$\frac{0.004\ 409}{1.491\ 626} \times 100\% = 0.3\%$$

这恰是图 3.3 中消费对稳态偏离的百分比。

^① 绝对偏离是水平变量的情形，即水平变量对其稳态的偏离量。粗略地说，相对偏离和绝对偏离就差一个分母。

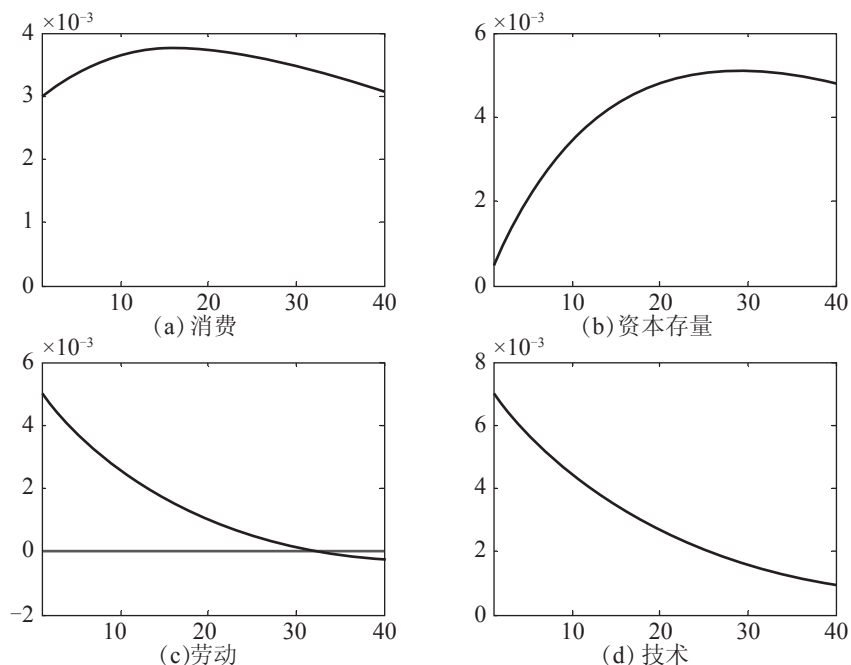


图 3.3 对数水平形式下的脉冲响应图

值得一提的是，虽然变量的表达形式发生了变化，但是均衡条件的结构没有发生任何变化，因此不会改变模型的内涵和变量的变化趋势（因为对数函数为增函数），只是变量本身和最终结果的解释方式发生了变化。此外，并不是所有的变量都可以或适合取对数，若某些变量的稳态为负，那么该变量一般不能取对数。再比如，某些比率变量如利率等，在净值 (net) 形式下不宜取对数，只有当比率变量为总值 (gross) 形式时才可以取对数，才有经济内涵。

3.5 Dynare 文件的预编译和运行原理

Dynare 的模型文件是用 Dynare 语言编写的文件 (*.mod)，只有 Dynare 才能读懂和处理，也就是说，Matlab 无法直接识别和处理。模型文件最终的执行和处理是依靠 Matlab，但必须经过 Dynare 预处理之后，翻译成 Matlab 可识别的文件（如 m 文件）才能最终执行。因此从这个意义上讲，Dynare 是外挂在 Matlab 上的一层外衣和中介，面向宏观经济学用户，以一种比较接近于经济学的语言编写程序，表征模型，然后由 Dynare 负

责中介处理和翻译，然后交由 Matlab 执行，并输出最终结果。

3.5.1 Dynare文件的预编译与运行原理

在编写完模型文件后，接下来是编译并运行模型，目标是求出决策函数。模型文件的预处理都交由 Dynare 的预处理器 (Preprocessor) 完成。预处理器是 Dynare 的编译器，包含了多个组成部分。模型文件的编译处理流程如下 (逻辑示意图如图 3.4 所示)：首先，若模型文件中含有宏命令^①，会先使用宏处理器，生成不含有宏命令的模型文件。其次，使用解析器对模型文件进行编译，形成较为抽象的、表征模型的 Matlab 文件 (一般为 m 文件)。这些 m 文件中包括了对 Dynare 和 Matlab 内置函数和命令的调用，并最终依靠 Matlab 输出各种结果。

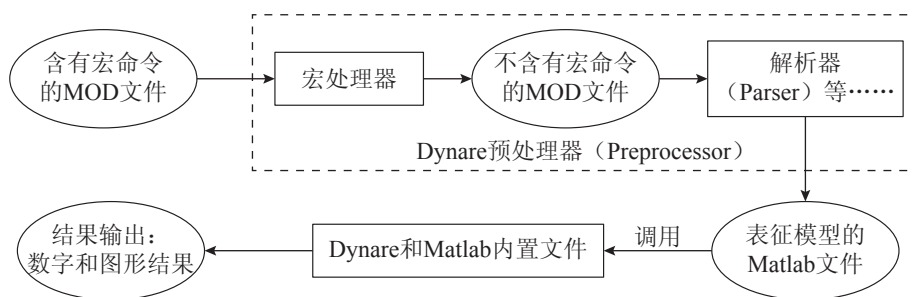


图 3.4 Dynare 的预编译和运行原理

在进一步介绍之前，有必要对 Dynare 的主命令 `dynare` 做进一步的介绍。`dynare` 命令在本书“2.3 执行和编辑 Dynare 文件”一节中已经提及，此处对其含义和选项做进一步说明。`dynare` 命令是预编译和求解的起点，即通过该命令启动 Dynare，自动完成预编译处理和求解两项任务。

`dynare` 命令使用非常简单，后面直接跟模型文件的全名，注意文件名大小写敏感，可加后缀名，也可省略，然后是以空格隔开的各种选项 (options)。具体语法参见源代码 14。

源代码14 `dynare`主命令示例

```
dynare GrowthApproximate.mod
dynare GrowthApproximate
dynare GrowthApproximate noclearall
```

- `dynare`命令有多达20余个选项，其中较为常用的有：

^① 宏命令将会在本书“6.5 Dynare 宏命令编程示例”一节中详细介绍。

- **noclearall**, 如果不使用该选项, Dynare会在编译成的主文件中(即filename.m, 比如GrowthApproximate.m)的第一行加入“clear all”命令, 以清除内存中的所有变量^①。如果使用该选项, 就不会清除。
- **debug**, 输出关于模型文件的解析和编译信息。即会在源代码 15中多出数千行的解析和编译信息, 供调试使用。
- **savemacro**, 当模型文件中含有宏语言时, 可使用此选项。该选项会命令宏处理器保存不含有宏语言的模型文件。本书“6.5 Dynare宏命令编程示例”一节中对此选项进行了示例。
- **nolog**, 指示Dynare不要保存预编译和计算过程的日志文件(即filename.log, 比如GrowthApproximate.log), 默认情况下保存日志文件。
- **nowarn**, 指示Dynare在编译和计算过程中不显示任何警告信息。
- **nograph**, 和计算命令, 比如stoch_simul的nograph选项具有相同的含义与作用, 即不画图(不显示也不保存)。

以第1节中的RBC模型为例子, 其预编译输出如源代码15所示。首先是Dynare的配置信息, 加载动态链接库mex文件。然后是模型文件的预处理信息, 包括均衡条件的语法检查、静态和动态模型的相关计算。最后开始运行编译后的Matlab文件(m文件)。

源代码15 Mod文件的预编译输出信息(Matlab屏幕输出)

```
Configuring Dynare ...
[mex] Generalized QZ.
[mex] Sylvester equation solution.
[mex] Kronecker products.
[mex] Sparse kronecker products.
[mex] Local state space iteration (second order).
[mex] Bytecode evaluation.
[mex] k-order perturbation solver.
[mex] k-order solution simulation.
[mex] Quasi Monte-Carlo sequence (Sobol).
[mex] Markov Switching SBVAR.

Starting Dynare (version 4.4.3).
Starting preprocessing of the model file ...
```

^① 当Matlab版本低于R2015b时。Matlab R2015b开始, 如果不使用该选项, 仅仅删除全局变量(global)和使用持久性(persistent)变量的函数, Matlab为两者都建立永久存储空间, 只不过persistent是函数内部可见的局部变量。Matlab R2015b开始如果使用该选项, 就不会删除全局变量和持久性变量。如果想在Matlab R2015b版本以前只清除全局和持久变量, 可使用onlyclearglobals选项。

```
Found 4 equation(s).
Evaluating expressions ... done
Computing static model derivatives:
- order 1
Computing dynamic model derivatives:
- order 1
Processing outputs ... done
Preprocessing completed.
Starting MATLAB/Octave computing.
```

3.5.2 表征模型的Matlab文件

在经过 Dynare 预处理后，即通过“dynare”命令后，将输入的模型文件，依具体情况生成如下的几个或全部 Matlab 的 m 文件（如下以 GrowthApproximate 为模型文件名来示例）^①：

- *GrowthApproximate.m*，表征模型的Matlab主文件，包含各种变量声明和各种计算命令。直接运行该文件，即可完成模型文件所要实现的全部目的。也就是说，该文件是主入口，因此在没有改动文件结构的前提下，直接运行该m文件，会节省大量的预编译时间。无须Dynare再次编译原有的模型文件，直接运行已编译好的m文件，会提升运行效率。特别是当循环调用时，此方法会节省大量的时间。
- *GrowthApproximate_static.m*，静态模型文件，包含长期静态均衡条件，用来计算稳态，会输出均衡条件的残差以及静态均衡条件的雅克比(Jacobian)矩阵。
- *GrowthApproximate_dynamic.m*，动态模型文件。
- *GrowthApproximate_steadystate2.m*，计算稳态时，如果使用内置模块命令“steady_state_model;”会生成该m文件。
- *GrowthApproximate_set_auxiliary_variables.m*，辅助变量文件，即模型均衡条件中出现多于一期的先导或滞后下标变量时，Dynare会生成此文件。

在随机模拟时，调用主文件会完成模型文件的执行。其基本的内部调用逻辑流程稍微复杂，牵涉多层调用，如图 3.5 所示。主文件依次嵌套调用 Dynare 的 3 个内置函数，进行模型求解：首先调用随机模拟函数，其次是 resol.m，再次是核心求解函数 stochastic_solvers.m，该函数实现了一阶和二阶扰动算法的核心逻辑，即由 Schmitt-Grohé & Uribe(2004) 提出的扰动算法；最后再调用模型的动态文件完成求解。

^① 此处仅仅列示了较为简单的情况。在更为复杂的模型中，比如 Ramsey 最优问题中，如果使用了诸如 ramsey_policy, planner_objective 等内置命令，会生成更多的辅助 m 文件，比如 filename_objective_static.m, filename_objective_set_auxiliary_variables.m 等 m 文件。

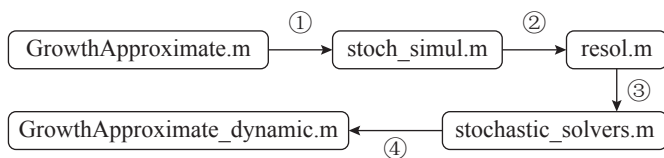


图 3.5 表征模型的 M 文件的调用逻辑 (随机模拟情况下)

3.6 Dynare 的解表示

求解完成后, Dynare 的最终计算结果在 Matlab 的主屏幕输出, 包含求解结果信息 (即政策函数与转换方程), 同时 Dynare 也会按照一定规则将求解结果存储到结构数组中。本节将使用本章第 1 节的 RBC 模型来分别讲述 Dynare 一阶和二阶求解表示和后续调用。

3.6.1 一阶解表示

Dynare 求解的最根本目的就是找到模型的政策函数 (决策规则)。正如前文所述, Dynare 关于一阶近似解有如下的经典表达形式:

$$y_t = g(y_{t-1}, u_t) \approx \bar{y} + g_y \hat{y}_{t-1} + g_u u_t \quad (3.6.1)$$

假设模型内生变量的个数为 n , 状态变量的个数为 m , 外生冲击的个数为 q , 根据 Dynare Reference Manual。

- \bar{y} 表示内生变量的稳态值, 行对应以声明顺序排列的内生变量, 其维度为 n 行 1 列, 存储于 `oo_dr.y` 结构数组中。
- g_y 表示状态变量的系数矩阵, 列对应状态变量, 行对应以决策规则顺序排列的内生变量, 其维度为 n 行 m 列, 存储于 `oo_dr.ghx` 中。
- g_u 表示外生冲击的系数矩阵, 列对应以声明顺序排列的外生变量, 行对应着以决策规则顺序排列的内生变量, 其维度为 n 行 q 列, 存储于 `oo_dr.ghu` 中。

根据上述的分析, 式 (3.6.1) 表示的经典一阶解形式, 其实并未考虑到变量的排序问题。也就是说, 其只是示意公式或形式逻辑。如果变量均以声明顺序排序, 那么 g_y 实际上对应着 `oo_dr.ghx(oo_dr.inv_order_var,:)`, 而 g_u 实际上对应着 `oo_dr.ghu(oo_dr.inv_order_var,:)`^①。

以第 1 节 RBC 模型为例, 内生变量的个数 $n = 4$, 状态变量的个数 $m = 2$, 外生冲

① 可在 Matlab 中进行验证, 即在 Matlab 命令行中输入 `oo_dr.ghx(oo_dr.inv_order_var,:)` 和 `oo_dr.ghu(oo_dr.inv_order_var,:)`, 会发现结果应和 (3.6.2) 是一致的。

击的个数 $q = 1$ 。模型定义中各变量以水平形式表示。根据内生变量的排序，再考虑到 Matlab 的运行结果，由式 (3.6.1) 可得到以声明顺序排列的解表示如下：

$$\begin{pmatrix} C_t \\ K_{t+1} \\ L_t \\ Z_t \end{pmatrix} = \begin{pmatrix} 1.4916 \\ 29.2885 \\ 0.2916 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.0282 & 0.5984 \\ 0.9779 & 1.9003 \\ -0.0019 & 0.1972 \\ 0 & 0.9500 \end{pmatrix} \begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix} + \begin{pmatrix} 0.6299 \\ 2.0004 \\ 0.2076 \\ 1.0000 \end{pmatrix} \epsilon_t \quad (3.6.2)$$

如果进一步观察 Dynare 在 Matlab 中输出的解表示 (在“政策函数和状态方程”一栏中) 会发现，上述由解表示的方法 (3.6.1) 得到的结果 (3.6.2) 与 Matlab 屏幕输出完全一致 (源代码 16)。

源代码16 一阶解表示：政策函数与转换方程(Matlab输出)				
POLICY AND TRANSITION FUNCTIONS				
	c	k	lab	z
Constant	1.491626	29.288520	0.291593	0
k(-1)	0.028175	0.977868	-0.001880	0
z(-1)	0.598385	1.900349	0.197182	0.950000
e	0.629879	2.000368	0.207560	1.000000

注意：第一列“Constant”表示内生变量的稳态值 (一阶近似时)。k(-1) 在形式上对应 K_t ，z(-1) 在形式上对应 Z_{t-1} ，e 表示外生冲击 ϵ_t ，即政策函数中的解释变量 (状态变量)。后 4 列分别表示各内生变量关于第一列各解释变量的系数向量。

3.6.2 二阶解表示

Dynare 关于二阶近似解，在本书“1.2.1 基于扰动项的泰勒近似方法”一节中已有详细介绍，其经典表达形式如下：

$$y_t = \bar{y} + \frac{1}{2} g_{\sigma\sigma} + g_y \hat{y}_{t-1} + g_u u_t + \frac{1}{2} [g_{yy} (\hat{y}_{t-1} \otimes \hat{y}_{t-1}) + g_{uu} (u_t \otimes u_t) + 2g_{yu} (\hat{y}_{t-1} \otimes u_t)] \quad (3.6.3)$$

其中， \bar{y} ， g_y ， g_u 的含义如一阶解表示中的相同，所不同的主要是二阶项及其系数。

- $g_{\sigma\sigma}$ 表示关于扰动项的二阶系数列向量，由外生冲击造成的转移效应 (Shift Effect)，行对应以决策规则顺序排列的内生变量，存储于 oo_dr.ghs2 中。
- g_{yy} 表示关于状态变量二阶项的系数矩阵，行对应以决策规则顺序排列的内生变量，列对应以决策规则顺序排列的状态变量的克氏乘积 (Kronecker Product，即张量乘积)，存储于 oo_dr.ghxx 中。

- g_{uu} 表示关于外生冲击变量二阶项的系数矩阵，行对应以决策规则顺序排列的内生变量，列对应以决策规则顺序排列的状态变量的克氏乘积，存储于oo_dr.ghuu中。
- g_{yu} 表示关于外生冲击变量二阶项的系数矩阵，行对应以决策规则顺序排列的内生变量，列对应以决策规则顺序排列的状态变量的克氏乘积，存储于oo_dr.ghxu中。

同样以本章第1节中的RBC模型为例，模型文件中各内生变量以水平形式表示。

首先，模型的状态向量为资本存量与技术冲击组成的 2×1 的列向量，其克氏乘积为 4×1 列向量，可写为

$$\begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix}_{2 \times 1} \otimes \begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix}_{2 \times 1} \equiv \begin{pmatrix} K_t \times \begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix} \\ Z_{t-1} \times \begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix} \end{pmatrix}_{4 \times 1} = \begin{pmatrix} K_t^2 \\ K_t \times Z_{t-1} \\ Z_{t-1} \times K_t \\ Z_{t-1}^2 \end{pmatrix}_{4 \times 1} \quad (3.6.4)$$

其次，外生冲击只有技术冲击，为1维向量即标量，因此 $u_t \otimes u_t$ 是标量，即 ϵ_t^2 。

最后，由于外生技术冲击只有1维，因此模型状态变量与外生冲击的克氏乘积仍为 2×1 的列向量，即

$$\begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix}_{2 \times 1} \otimes u_t \equiv \begin{pmatrix} u_t K_t \\ u_t Z_{t-1} \end{pmatrix}_{2 \times 1} = \begin{pmatrix} \epsilon_t K_t \\ \epsilon_t Z_{t-1} \end{pmatrix}_{2 \times 1} \quad (3.6.5)$$

从存储结果中提取相关系数矩阵，并代入二阶解表示式(3.6.3)中，因此二阶解为^①

$$\begin{pmatrix} C_t \\ K_{t+1} \\ L_t \\ Z_t \end{pmatrix} = \begin{pmatrix} 1.4916 \\ 29.2885 \\ 0.2916 \\ 0 \end{pmatrix} + \underbrace{\frac{1}{2} 1.0E-05}_{\frac{1}{2} g_{\sigma\sigma}} \begin{pmatrix} -0.4503 \\ 0.8616 \\ 0.1085 \\ 0 \end{pmatrix} + \underbrace{\begin{pmatrix} 0.0282 & 0.5984 \\ 0.9779 & 1.9003 \\ -0.0019 & 0.1972 \\ 0 & 0.9500 \end{pmatrix}}_{g_y} \begin{pmatrix} K_t \\ Z_{t-1} \end{pmatrix} + \underbrace{\begin{pmatrix} 0.6299 \\ 2.0004 \\ 0.2076 \\ 1.0000 \end{pmatrix}}_{g_u} \epsilon_t \quad (3.6.6)$$

① 各系数矩阵均已调整为声明顺序，而非决策规则顺序，因此各系数矩阵的行位置有别于Dynare中存储的各系数矩阵的行位置。其中一阶项的系数和前一小节相同。

$$\begin{aligned}
 & + \frac{1}{2} \underbrace{\begin{pmatrix} -0.0004 & 0.0074 & 0.0074 & 0.4301 \\ -0.0002 & 0.0241 & 0.0241 & 2.4220 \\ 0.0001 & 0.0006 & 0.0006 & -0.0079 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix}}_{g_{yy}} \underbrace{\begin{pmatrix} K_t^2 \\ K_t \times Z_{t-1} \\ Z_{t-1} \times K_t \\ Z_{t-1}^2 \end{pmatrix}}_{4 \times 1} \\
 & + \underbrace{\begin{pmatrix} 0.0078 & 0.4527 \\ 0.0254 & 2.5494 \\ 0.0006 & -0.0083 \\ -0.0000 & -0.0000 \end{pmatrix}}_{g_{yu}} \underbrace{\begin{pmatrix} \epsilon_t K_t \\ \epsilon_t Z_{t-1} \end{pmatrix}}_{2 \times 1} + \frac{1}{2} \underbrace{\begin{pmatrix} 0.4765 \\ 2.6836 \\ -0.0087 \\ 0 \end{pmatrix}}_{g_{uu}} \epsilon_t^2
 \end{aligned} \tag{3.6.6}$$

仔细观察就会发现，上述解表示的方法得到的结果 (3.6.6) 与 Matlab 屏幕输出完全一致 (源代码 17)。

源代码17 二阶解表示：政策函数与转换方程(Matlab输出)				
POLICY AND TRANSITION FUNCTIONS				
	c	k	lab	z
Constant	1.491624	29.288524	0.291594	0
(correction)	-0.000002	0.000004	0	0
k(-1)	0.028175	0.977868	-0.001880	0
z(-1)	0.598385	1.900349	0.197182	0.950000
e	0.629879	2.000368	0.207560	1.000000
k(-1), k(-1)	-0.000184	-0.000080	0.000026	0
z(-1), k(-1)	0.007386	0.024104	0.000582	0
z(-1), z(-1)	0.215030	1.210985	-0.003943	0
e, e	0.238261	1.341812	-0.004369	0
k(-1), e	0.007775	0.025372	0.000613	0
z(-1), e	0.452695	2.549443	-0.008301	0

注意：其中 $k(-1)$ 、 $z(-1)$ 、 e 的含义同一阶解表示中的含义。而“Constant”则表示变量的稳态值加上 (correction)，此时 (correction) 表示 $1/2g_{\sigma\sigma}$ ，一般说来该值很小。 $k(-1)$ 、 $k(-1)$ 表示 K_t^2 ，其他类同。在 Matlab 屏幕输出中， K_t^2 的系数对应了最终的计算结果即 $1/2g_{yy}$ ，因此和 `oo_dr.ghxx` 对应的 g_{yy} 相差一个系数 1/2。

3.7 求解结果分析和调用

在上一节中介绍了政策函数的表示，其中已经涉及如何调用求解的部分结果，特别是政策函数中各项系数矩阵。本节继续介绍其他结果的调用，为后续分析所用。

3.7.1 屏幕输出结果

屏幕输出信息给出了运行结果的简要概括。其中大部分信息都被存储下来，后面会详细介绍。屏幕输出信息可分为两类：一类是预编译信息；一类是计算结果信息。预编译信息已经在本书“3.5.1 Dynare 文件的预编译与运行原理”一节中讨论过，此处主要介绍计算结果信息，并在本节第二部分介绍各计算信息的存储位置。

源代码 18 中显示了 8 个方面的结果信息，并都已经说明每个信息具体的存储位置：Dynare 专门设计的结构数组，前面的阐述中已经涉及。接下来，简单介绍这些重要结构数组的含义及其常用字段。

源代码18 计算结果信息(Matlab输出)

```
% 1. 内生稳态值, 存储于oo_.steady_state和oo_.dr_ys中
STEADY-STATE RESULTS:
c      1.49163
.....

% 2. 静态方程的残差
Residuals of the static equations:
Equation number 1 : -1.5417e-07 : Euler equation
.....

% 3. 模型变量概要, 存储于模型数组M_中
MODEL SUMMARY
Number of variables:      4
Number of stochastic shocks: 1
Number of state variables: 2
Number of jumpers:      2
Number of static variables: 0

% 4. 模型外生冲击的方差与协方差, 存储于模型数组M_.Sigma_e中
MATRIX OF COVARIANCE OF EXOGENOUS SHOCKS
Variables      e
e              0.000049

% 5. 政策函数与转换方程, 存储于oo_.dr中
POLICY AND TRANSITION FUNCTIONS

.....

% 6. 内生变量的各阶模拟矩, 存储于oo_.mean, oo_.var等数组中
MOMENTS OF SIMULATED VARIABLES
VARIABLE MEAN STD. DEV. VARIANCE
SKEWNESS KURTOSIS

c      1.467646    0.037010    0.001370    0.257294    0.239254
.....

% 7. 内生变量的相关系数, 存储于oo_.autocorr中
CORRELATION OF SIMULATED VARIABLES
```

```
VARIABLE      c      k      lab      z
c             1.0000  0.9585  0.4131  0.7532
k             0.9585  1.0000  0.1363  0.5345
lab          0.4131  0.1363  1.0000  0.9101
z            0.7532  0.5345  0.9101  1.0000

% 8. 内生变量的自相关系数, 存储于oo_.autocorr中
AUTOCORRELATION OF SIMULATED VARIABLES
VARIABLE      1      2      3      4      5
c             0.9921  0.9840  0.9746  0.9650  0.9550
.....
Total computing time : 0h00m06s
```

注：为了节省篇幅，此处省略了部分细节，其中关于政策函数和转换方程部分的仔细分析，请参考本书“3.6 Dynare 的解表示”一节的详细介绍。

3.7.2 存储结果

Dynare 会将大部分计算结果以结构数组的形式存储下来，以备后续分析和调用。Dynare 将计算结果分为 3 大类，分别存储于 `M_`、`oo_` 和 `options_` 这 3 个结构数组中^①。这 3 个结构数组的存储结果，会因不同的求解选项设定出现细微的差异。比如二阶求解比一阶求解会多出几项系数矩阵。

为了后续的分析，Dynare 会将这三个结构数组以 `mat` 文件的形式保存到硬盘上，以备后用。以本章第一节的 RBC 模型为例，Dynare 保存的这个 `mat` 文件的名字为 `GrowthApproximate_results.mat`，即以 `filename_results.mat` 的名字保存，其中 `filename` 表示原模型文件的名称。在 Matlab 中，使用 `load` 命令可加载此 `mat` 文件至内存，以作分析。

1. 模型信息数组 `M_`

数组 `M_` 包括了几乎所有和模型有关的基本解析信息，比如模型文件的名称、模型内生变量和参数的名称、参数的数值、4 种类型内生变量的个数、外生冲击及其方差和最大的先导或滞后的阶数等信息，因此不妨将 `M_` 称为模型数组。所有这些信息均可在模型运行完毕后进行调用，以作分析。在模型数组总共 40 余个数组中，比较常用的如表 3.3 所示。

^① 一般情况下，`mod` 文件编译运行后都会产生这 3 个数组。在不同的设定或计算命令下会产生其他的结构数组，如 `estimation_info`、`estim_params_`、`bayestopt_`、`dataset_oo_recursive_` 等字段，都会被存储到结果文件中。

表 3.3 模型数组中常用的字段

字 段	名 称	备 注
M_param_names	参数名称	多维字符矩阵
M_params	参数取值	数组
M_param_nbr	参数个数	整数
M_endo_names	内生变量名称	多维字符矩阵
M_endo_nbr	内生变量个数	整数
M_exo_names	外生冲击名称	多维字符矩阵
M_exo_nbr	外生冲击个数	整数
M_Sigma_e	外生冲击的方差协方差矩阵	方阵
M_sigma_e_is_diagonal	Sigma_e 是否为对角矩阵	0 或 1

此外，关于不同类型内生变量的个数也是较为常用的信息，此处不再列示，具体可参考本书“表 3.1 Dynare 内生变量的分类”。

2. 计算和模拟结果数组 oo_

模型计算的大部分结果都将被存储于结构数组 oo_ 中，因此不妨将 oo_ 称之为结果数组。结果数组中根据不同的设定情形，存储了大约近 30 个子数组^①，其中较为常用的如表 3.4 所示。

表 3.4 结果数组 oo_ 中的常用字段

字 段	名 称	备 注
oo_exo_simul	外生冲击的模拟样本矩阵	数值矩阵
oo_endo_simul	内生变量的模拟样本	数值矩阵
oo_dr	决策规则系数数组	结构数组
oo_exo_steady_state	外生变量的稳态	列向量 (或标量)
oo_steady_state	内生变量的稳态	列向量
oo_mean	内生变量的理论或模拟均值	列向量
oo_var	内生变量的理论或模拟方差协方差矩阵	方阵
oo_autocorr	内生变量的自相关矩阵	cell 数组
oo_irfs	脉冲响应函数	结构数组

除决策规则数组 (oo_dr.y) 除外，其余矩阵或向量的行都对应着以声明顺序排列

① 结果数组中依据不同情况，还存储着诸如预测、平滑、滤波、后验分布等相关信息。具体请参考 Dynare Reference Manual, http://www.dynare.org/manual/index_54.html#Variable-Index。

的内生变量。

`oo_exo_simul` 存储着外生冲击的模拟样本 (或称之为模拟路径)。该模拟样本一般是从正态分布中随机抽取, 其均值为 0, 方差或标准差由模型文件中的外生冲击模块指定^①。因此, 一般情况下, 外生冲击的其稳态为 0, 也就是说, `oo_exo_steady_state` 一般是零向量。`oo_endo_simul` 则存储着内生变量的模拟路径, 并同时以各内生变量的名称为数组名称, 将模拟路径直接列示在内存中。

`oo_dr` 结构数组存储着决策规则中各变量的系数矩阵, 在本书“3.6 Dynare 的解表示”一节中已经做过详细介绍, 此处不再赘述。

`oo_steady_state` 存储着内生变量的稳态值, 内容同 `oo_dr.ys`, 两者均以声明顺序排序。

`oo_mean` 和 `oo_var` 分别存储理论或模拟的一阶与二阶矩。所谓理论矩 (Theoretical Moments) 是指随机模拟的期数, 即 `periods` 选项未被指定时 (在随机模拟分析中将会进一步介绍), 没有进行模拟的条件下, 从均衡条件的关系式中计算出变量的各阶统计量。模拟矩 (Simulated Moments 或 Empirical Moments) 就从模拟出来的样本中计算出来的各阶统计量。

`oo_autocorr` 则存储着内生变量之间的自相关系数, 具体含义如下:

$$\text{oo_autocorr}\{i\}(m,n) \equiv \text{corr}\left(y_t^{(m)}, y_{t-i}^{(n)}\right) \quad (3.7.1)$$

其中, i 是自相关数组中的第 i 个自相关矩阵, 表示滞后阶数, 默认情况下最大滞后阶数为五阶, 也就是说自相关数组中默认情况下有 5 个自相关矩阵。 m, n 分别表示第 m, n 个内生变量 (以声明顺序排序)。

`oo_irfs` 存储各内生变量的脉冲响应, 其长度 (期数) 由计算命令的选项 `irf` 指定。脉冲响应的命名规则为 `Variable_Name_Shock_Name`。比如 `c_e` 表示变量 c 关于外生冲击 e 的脉冲响应。此外, 各变量的脉冲响应同时列示在内存中, 名称不变, 便于调用和查看。

在本书“3.9 随机模拟分析: `stoch_simul`”一节将结合第 1 节中 RBC 模型的例子, 示例如何调用模拟结果, 并画出模拟的内生变量的示意图。

此外, 在运行完毕模型文件后, 往往需要对内生变量的稳态做出分析, 如何快速地将所有内生变量和其稳态对应起来, 并动态加载到内存作为变量 (参数) 供分析使用呢? 比如, 消费 C , 假设其稳态值为 1, 分析时往往需要 $C=1$ 这样形式的变量。首先, 要弄明白的是稳态值被以声明顺序存储在 `oo_steady_state` 和 `oo_dr.ys` 两个数组中。其次, 可

^① 以本章第 1 节的 RBC 模型为例子, 可以验证抽取的外生冲击的样本, 的确是均值为 0, 标准差为 0.7% 的正态样本。

以编写内容为源代码 19 的 m 文件来实现此目的^①。

源代码 19 动态加载稳态值到内生变量

```
%loading steady states to endogenous variables
%how many endo. variables
Nendo = M_.orig_endo_nbr;

%dynamic loading steady states to endo. vars.
for indexvar = 1:Nendo
    %finding out the name of endogenous variables.
    varname = deblank(M_.endo_names(indexvar,:));
    eval([ varname '= oo_.steady_state(' int2str(indexvar) ')']);
end
```

源代码 19 首先从 M_ 中分别获取内生变量的个数以及内生变量的名称。然后动态加载，使用了 deblank、eval 和 int2str 共 3 个 Matlab 内置函数。其中 indexvar 为整数，需要使用 int2str 函数将其转换为字符串，使用中括号将各个字符串连接起来。最后使用 eval 动态执行字符串对应的命令。上述代码需要在模型文件运行完毕后再执行。

3. 选项数组 options_

选项数组 options_ 中存储着和模拟计算相关的各种细节性设定，多达近 200 项^②。比如求解牵涉的各种（迭代）算法的终止条件（误差容忍度）、随机模拟的求解阶数、模拟的期数，脉冲响应的计算期数，自相关系数计算的阶数，还包括结果输出中的各种选项：是否输出图像；是否保存输出的图形等。

由于模型文件中各大命令（如随机模拟命令 stoch_simul 和估计命令 estimation）的选项几乎都被写入选项数组，导致选项数组的字段比较繁多。虽然 Dynare Reference Manual 中并未给出具体的解答，但大多数选项的含义不言自明，可以反向追回到各大命令中去查找该选项的具体含义，比如随机模拟命令：“stoch_simul(periods=1000,irf=40,order=2);” 的 3 个选项，分别被存储在 options_.periods, options_.irf, options_.order 中，因此此处并不一一介绍每个选项的含义。

① 源文件地址：\Sources\Chap3_Dynare_Basics\3.7_results_analysis\loading_steady_states_into_endo_vars.m。

② 如果要查看选项数组的每个字段，可在运行完 mod 文件后，在 Matlab 的命令行中输入 options_ 即可列示所有相关的字段。

3.8 确定性求解和模拟：simul

确定性和随机模拟是模型分析的重要内容和方法，是在参数估计和模型求解之后必须要做的操作。不论是确定性还是随机模拟，都必须是模型结构参数已知（校准或估计后）并且模型已经被求解的基础上才能进行。

在进行两种模拟之前，还需要做一些准备工作。首先，在大多数的情况下，不论确定性模拟还是随机模拟，都需要提供初始条件 (Initial Condition)，某些情况下需要提供终止条件 (Terminal Condition)。具体说来，初始和终止条件一般是指内生变量和外生变量的初值和终值 (Initial and Terminal Value)。其次，还要依据问题分析需要，指定外生冲击的变化特征包括外生冲击大小、外生冲击的方差或标准差、持续周期等，选择模拟的类型即确定性还是随机模拟。下面结合 Dynare 的语法，首先介绍如何设定初值和终值。

3.8.1 初始、终止条件

一般说来，初值设定的目的有两个。第一个目的是为（随机或确定性）模型计算稳态值，提供一个初始的出发点。通常，如果模型的内生变量的稳态值容易求出，则可直接将其作为初值，这样会提升计算效率，节省时间，因为此时 Dynare 最多需要一次迭代。第二个目的就是为确定性模型模拟提供初值。终值的设定并不是专指设定模拟后各期的值，如果设定终值，它将覆盖初值设定的部分功能，下文进一步详述。

在 Dynare 中，模拟始于第一期 (period=1)，在模拟之前的那一期为第 0 期 (period=0)，在模拟之前的第 2 期为负 1 期 (period=-1)，如此类推（如图 3.6 所示）。初值即是第 0 期及以前各期的值，终值即是第 $T+1$ 期及以后各期的值。

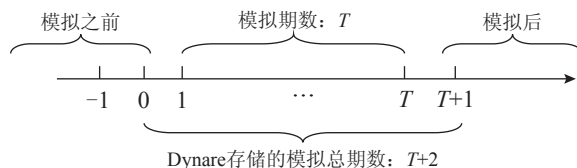


图 3.6 Dynare 模拟的时空设定

假设总模拟期数是 T 期，模拟期是指 1 到 T 期，模拟后的第一期是指 $T+1$ 期。一般情况下，Dynare 在存储模拟结果时（确定性模拟），不仅考虑模拟期，而且还考虑模拟前和模拟后的各一期，因此总共有 $T+2$ 期。如果画出此模拟路径，则往往在首尾位置出现

跳跃的情况。这是由于在 Dynare 中，模拟前和模拟后各变量值的设定通常独立于或先于模拟本身，因此设定值通常和模拟计算的结果存在较大的差异，于是在图形上表现为跳跃的情况。

在 Dynare 中设定初值和终值分别使用关键字：`initval`(或由 `histval` 修改) 和 `endval`，并使用 `end` 关键字结束初始和终止值的设定(见表 3.5)。然后在两个关键字之间设定内生和外生变量的初值和终值。一般来说，所有的内生变量和外生变量都要设定初值，否则默认值为 0。需要注意的是，初值设定一般来说是必需的^①，而终值设定则不是必需的(只有在确定性模拟中使用)。

表 3.5 初值和终止模块的设定

初值设定	终值设定
<pre>initval; endogenous_variables = initial_values; exogenous_variables = initial_values; end;</pre>	<pre>endval; endogenous_variables = initial_values; exogenous_variables = initial_values; end;</pre>

在 Dynare 中，初值设定的含义因模型类型不同而不同。确定性模拟相比随机模拟，初值和终值的设定稍微复杂。

在确定性模型中，使用 `initval` 模块设定初值是指对第 0 期及其之前的各期设定取值 (Before Simulation)^②，除非使用 `histval` 命令修改过。在没有 `endval` 模块设定终值时，`initval` 模块设定的初值将作为模拟后各期的取值 (After Simulation)，同时也作为模拟计算的初值。如果有 `endval` 模块，那么 `endval` 模块中设定的值将做为模拟后各期的值，`initval` 模块设定的初值仅为第 0 期的值，不再作为模拟后各期的值，这一点非常重要，特别是后续进行不同类型的确定性模拟时尤其如此。如果 `initval` 模块后紧跟 `steady` 命令，`initval` 模块的作用将发生变化：首先 `steady` 命令将使用 `initval` 模块提供的初值作为初始猜测值，来计算内生变量的稳态值，并将稳态作为初值。因此，`initval` 模块外加 `steady` 命令的设定，相当于只设定 `initval` 模块，并且各变量初值被设定为稳态值。

前述已经提及，使用 `endval` 模块设定终值时，将会覆盖部分 `initval` 模块的功能：终值优先用于设定模拟后各期的值，而不管 `initval` 的初始如何设定。如果某些变量在

① 尽管从理论上讲，初值的设定只有对后向变量 (backward-looking) 才有意义。虽然初值设定是必需的，但这并不意味着一定需要设定 `initval` 模块。

② 第 0 期之前期数的长度取决于模型中变量的最大滞后阶数。一般情况下，初值只针对第 0 期，也就说最大滞后阶数为 1。

endval 模块中没有被赋予终值，则其取值于上一个 initval 模块或 steady 命令稳态值（如果存在）^①。

此外，当 steady 命令结合 initval 和 endval 模块同时使用时，有更深刻的内涵。此时，Dynare 会输出两个稳态值，分别基于初值和终值的设定而分别计算出来的。如果两个模块中设定的外生变量的值有差异，那么这两个稳态值一般不同。initval 模块对应的稳态值被设定为初值，endval 模块对应的稳态值被设定为终值^②。

在随机模型中，初值的设定有着非常不同的含义，通常情况下是作为非线性求解算法的初始猜测值，用来求解模型的稳态值，而不管是否有 steady 命令紧随其后。如果进行随机模拟 (periods=xx，即设定 periods 选项) 或预测 (forecast)，这时会使用稳态值作为初值，即第 0 期及以前各期的值均为稳态值。

此外，可以使用 histval 模块来修改历史值，能和 initval 模块同时使用，但不能和 endval 模块同时使用，具体可参考 Dynare Reference Manual。

3.8.2 稳态求解命令：steady

计算稳态值是非线性模型求解面临的一个非常具有挑战性的问题，特别当模型较为复杂时尤其如此。在 Dynare 中，steady 命令被用来求解模型的稳态值。具体来说，steady 命令使用一个非线性的牛顿算法（递归算法），并使用 initval 或 endval 模块中声明的内生和外生变量的值作为初始值来求解稳态。通常情况下，对于较为复杂的模型甚至简单的模型，Dynare 都无法准确地求出模型的稳态值。这可能是因为初始值离稳态值太远，或默认求解的递归算法并不适合。这时需要重新猜测初始值或更改算法，但这往往耗时且效率不高。经验表明，完全依靠 Dynare 来计算模型的稳态值，几乎行不通。这也是初学者经常碰到问题的地方。因此，求解模型的稳态值，大部分只能靠手动编程来实现。在 Dynare 中具体编程时，稳态求解通常采取如下几种常见的处理方式，并且不受模型类型的限制，即不管模型是确定性还是随机模型都可以使用。

1. 自定义稳态参数

自定义稳态参数的方法是非常常用的方法，特别适用于中小规模的模型，而且要求

① 在“3.8.3 确定性模拟”一节中的例子：det_rbc_4_Unanticipated_Permanent.mod（未预期永久性冲击）和 det_rbc_5_Anticipated_Permanent.mod（预期永久性冲击）两个例子，请比较 oo_exo_simul 的差异，就能更加深刻理解 endval 模块设定的含义了。

② endval 模块后面紧跟的 steady 命令在计算稳态时，假设外生变量的指定值一直不变，并使用此值计算内生变量的稳态值。

稳态值能够方便地计算，具有解析解。这种方法的好处是简洁明晰，而且后续可扩展能力较强。使用这种处理方式，允许方便地循环调用模型文件来进行某个或某些参数的敏感性测试。该处理方式的基本逻辑如下：

首先，对每一个内生变量定义一个稳态参数。比如消费 C ，定义其稳态参数为 C_{ss} 。然后编程计算 C_{ss} 的取值。这要求内生变量的稳态值具有解析表达式，即稳态值要能够表示成结构参数、外生变量稳态值等变量的解析式。或要求内生变量稳态值能够数值求解。最后，在初值模块直接将 C_{ss} 赋予消费变量 C 即可。直接将稳态值作为初值赋予给内生变量，省去 Dynare 自行求解稳态值的烦琐过程，将在一定程度上提高 Dynare 运行的效率。

在源代码 20 的示例中，甚至可以省略 `steady` 命令，让 `stoch_simul` 命令来计算稳态值。但值得注意的是，如果加入 `steady` 命令，则需要放在 `initval` 模块后，否则会出现错误。

源代码20 自定义稳态参数求解稳态值

```

var C ...;
varexo epsA;
parameters alpha beta...;

%declare the steady state parameters for each endogenous variable
parameters Css ...;

% Compute the steady state of consumption C;
% This requires that steady states must have analytical forms of model
% structural parameters and already-known steady states of other
% endogenous and exogenous variables.
Css = ...;

model;
...; % model equilibrium conditions;
end;

initval;
C = Css;
...;
end;

stoch_simul;

```

2. 内置模块命令 `steady_state_model`

稳态计算的内置模块命令，提供了更加灵活的、可靠的稳态值计算方式，是第二种稳态计算方式。相比第一种自定义参数的办法，此种办法更为便捷，甚至不需要自行定义任何稳态参数，而直接在该模块内部使用内生变量本身进行稳态值的计算，而且可以不加声

明地使用自定义参数来辅助稳态值计算。在某些特殊的情况下，此命令非常有用。如在参数估计时，要对模型中的某个参数值在可能的参数空间内的每一点，重复计算稳态值。

和第一种自定义方法一样，此种处理方法具有一定的局限性。该方法同样要求内生变量稳态值具有解析解的形式或数值求解。也就是说，只适用于中小模型。对于不具有解析解或者求解稳态非常困难的模型，不推荐使用这种方法，而推荐使用第三种方法：使用外部 M 文件。

内置模块命令的语法非常简单。源代码 21 提供了一个简单的例子。在模块内部稳态计算上，使用了很多辅助变量，诸如 `Y_K`, `C_K`, `N_K` 等参数辅助计算，无须事前声明，比较方便。此外，该命令本身不直接计算稳态，需要辅以 `steady` 命令来具体计算和执行^①。如果不需要改变初值和终值，则无须使用 `initval` 和 `endval` 模块。如果确实需要使用 `initval` 和 `endval` 模块设定初值和终值，则 `steady` 命令须紧跟在 `initval` 和 `endval` 模块后来执行稳态计算。

源代码21 稳态计算内置模块命令

```
steady_state_model;
..... (具体稳态计算过程)
end;
steady;
```

Dynare 会根据内置模块命令内的信息，自动生成一个稳态计算的 M 文件：`filename_steadystate2.m`。

3. 使用外部 M 文件

对于复杂的模型，通常使用自定义的外部稳态计算文件是一种选择，也是稳态计算的第 3 种方式。这种处理方式的优点是能够较为灵活地处理稳态计算问题，但付出的代价就是繁重的编程任务，并耗时较多。

使用外部 M 文件，通常需要遵循一定的约定。首先，外部 M 文件的名称必须满足形式 `filename_steadystate.m`，其中 `filename` 为模型文件的名字，否则 Dynare 无法识别而出现运行错误；其次，M 文件的书写必须遵循一定的格式，否则 Dynare 无法读取计算的稳态值而出现错误并停止运行。

Dynare 提供了一个简单的附带例子，可以在 Dynare 的安装根目录下 `examples` 文件夹中找到：“`NK_baseline_steadystate.m`”。该稳态 M 文件，专门为 `NK_baseline.mod` 文件提供所需的稳态值，在模型文件编译中被调用。此外，本书“5.8 金融加速器与随机波

^① 否则会出现类似于如下的错误：Fatal error in bytecode: in Simulate_Newton_Two_Boundaries, the initial values of endogenous variables are too far from the solution. Change them!

动模型示例”一节中也给出了一个例子^①。

下面简单介绍一下该外部 M 文件的语法格式要求。该外部 M 文件其实是定义了一个 Matlab 函数，由 Dynare 自动调用。该函数有两个输入变量和两个输出变量，具体如下所示。

```
% function [ys,check] = NK_baseline_steadystate(ys,exo)
% 使用Matlab内置数值求解算法fsolve 计算 NK_baseline.mod 对应的内生变量的稳态值
% 输入:
%   - ys          [向量] 内生变量的初始值;
%   - exo         [向量] 外生变量的取值;
%
% 输出:
%   - ys          [向量] 内生变量的稳态值
%   - check       [标量] check=0 如果稳态计算无异常,正常工作; check=1 稳态
计算碰到问题;
global M_ options_
check=0;
...
%获取参数及其值,此处仅为示例;获取参数是稳态计算的必须步骤
%M_.param_names 存储所有参数名称
%M_.params       存储所有参数值
%M_.param_nbr    存储参数数量,为整数
Np = M_.param_nbr;
for I = 1:Np
    paramname = deblank(M_.param_names(I,:));
    eval([' paramname ' = M_.params(' int2str(I) ');']);
end
...
(稳态计算过程)
...
% 稳态赋值给稳态值ys变量,此变量名称为Dynare内部设定的名称,不得更改
Ne = M_.orig_endo_nbr;
ys = zeros(Ne,1);
for indexvar = 1:Ne
    varname = deblank(M_.endo_names(indexvar,:));
    eval(['ys(' int2str(indexvar) ') = ' varname ');']);
end
```

函数的主体结构中的第一部分，首先声明了全局变量 $M_$ ，从中获取模型结构参数的值，这是必需的步骤，否则无法进一步求解其他变量的稳态值。然后开始根据模型均衡条件求解稳态值，这是最重要的，也是最复杂的部分，依据模型不同而不同。最后返回稳态值向量 ys 以供 Dynare 继续求解使用。注意，此处至少两处使用了 Matlab 的内置函数 `eval`。

① 源文件地址：\Sources\Chap5_Financial_Friction\5.8_example\fin_acc_rbc_steadystate.m。

3.8.3 确定性模拟

1. 确定性模型求解的基本逻辑

确定性模拟 (Deterministic Simulation) 具体说来是指, 通过外生冲击的确定性变化过程来考察系统的变化, 如在某一期或几期施加一个外生的冲击来考察各内生变量在外生冲击发生后的变化路径和趋势, 画出脉冲响应图, 以直观观测外生冲击的影响。

确定性模型是指模型中不存在随机的外生冲击, 或者存在外生冲击, 但其是完全可见的, 也就是说通常假设完全信息 (Perfect Foresight), 指外生冲击的数值大小和持续周期被完全知晓, 不像随机冲击那样只知道外生冲击的分布, 在当期冲击实现后, 下一期样本取值完全不被知晓。确定性模拟是用来考察均衡系统在给定初始状态下, 对于给定的外生冲击, 系统如何返回原来的均衡或者如何到达另外一个新均衡。和确定性模型相反, 随机模型是指模型中存在随机的外生冲击, 形式上模型不仅存在期望算子和外生冲击的概率分布, 而且外生冲击的方差不能为 0。因此, 区别确定性和随机模型的一个关键的判断标准就是对外生冲击的定义。举例说明如下。假设模型中存在一个外生的技术冲击变量, $\{\epsilon_t\}_{t=0}^T$ 满足独立同分布。如果仅仅考虑 $T < \infty$ 的情形:

第一, 如果 $\epsilon_t = 0, t=0, 1, \dots, T$, 此时外生冲击完全消失, 可认为此时模型为确定性模型;

第二, 如果 $\epsilon_0 = 0, \epsilon_1 = 0.02, \epsilon_t = 0, t=2, \dots, T$, 此时除第 1 期外, 其他各期全为 0, 外生冲击的变化被完全预期, 因此模型为确定性模型;

第三, 如果 $\epsilon_t \sim N(0, 0.01^2), t=0, 1, \dots, T$, 即服从正态分布, 随机抽取样本作为其实现值, 不能被完全预期, 模型为随机模型。

2. 确定性模拟的 4 种类型

依据外生冲击的可预期性和变化长短类型, 确定性模拟可以分为 4 种情况: 预期的永久性变化、预期的暂时性变化、非预期的永久性变化和非预期的暂时性变化。直观来说, 预期和非预期是指外生冲击是否出现在模拟的第 1 期。如果出现在第 1 期, 通常是非预期性冲击, 否则是预期性冲击^①; 而永久性和暂时性是指变量的初值和终值是否相同, 如果两者不同, 则说明出现了永久性改变, 否则是暂时性变化。

上述确定性模拟需要 Dynare 中的几个命令组合才能实现: 初值命令 `initval` 和终值命令 `endval`、冲击命令 `shocks` 和模拟命令 `simul` 来实现, 如表 3.6 所示。

^① 直观上说, 外生冲击出现在第 1 期, 多为非预期性; 当使用冲击模块设定外生冲击的取值及其出现的期数, 这相当于说了外生冲击是确定出现的, 多为预期性。

表 3.6 确定性模拟的 4 种情况

变化长短 \ 预期性	预 期 的	非 预 期 的
永久性变化	初值 ≠ 终值 使用 <code>endval</code> 模块 外生冲击出现在其他期 需要使用 <code>shocks</code> 模块	初值 ≠ 终值 使用 <code>endval</code> 模块 外生冲击出现在第 1 期 不需使用 <code>shocks</code> 模块
暂时性变化	初值 = 终值 外生冲击出现在其他期 需要使用 <code>shocks</code> 模块	初值 = 终值 外生冲击出现在第 1 期 不需要使用 <code>shocks</code> 模块

注：永久性变化表示 Permanent Change；暂时性变化表示 Transitory Change；预期的表示 Anticipated；非预期的表示 Unanticipated。

后续会通过几个例子，对 4 种不同的组合实现不同的确定性模拟加以细致解释。接下来先对外生冲击模块、确定性模拟命令进行介绍，然后举例说明。

3. 外生冲击模块设定：Shocks

在 Dynare 中，外生冲击是指由 `varexo` 定义的变量，而非某个内生变量或者某个结构参数。确定性模拟使用外生冲击模块设定外生冲击的暂时性变化，而永久性变化需要使用 `endval` 模块定义。需要强调的是外生冲击模块只对外生冲击进行设定，而不能对参数或内生变量设定暂时性变化。

确定性模拟和随机模拟的外生冲击的设定方式在 Dynare 中不完全相同，如表 3.7 所示。

表 3.7 外生冲击模块的设定

	确定性模拟	随机模拟
语法	<code>shocks;</code> <code>var</code> 外生变量名称； <code>periods</code> 某一期或多期； <code>values</code> 数值或表达式； <code>end;</code>	<code>shocks;</code> <code>var</code> 外生变量名称； <code>stderr</code> 数值或表达式； <code>var</code> 外生变量名称 = 数值或表达式； <code>end;</code>
示例	<code>shocks;</code> <code>var e;</code> <code>periods 1;</code> <code>values 0.5;</code> <code>var u;</code> <code>periods 4:5;</code> <code>values 0;</code> <code>end;</code>	<code>shocks;</code> <code>var e ; stderr 0.01;</code> <code>var e ; stderr sigma-alpha;</code> <code>var e=sigma^2;</code> <code>var e=0.01^2/2;</code> <code>end;</code>

确定性模拟中，通常有 3 个关键字同时出现：var, periods 和 values。首行使用 var 定义发生变化的外生冲击变量的名称；然后使用 periods 指定变化出现在何时；最后使用 values 定义变化的大小。多个外生变量需要重新使用 3 个关键字加以指定。

在随机模拟中，外生冲击模块只使用 var 和 stderr 两个关键字，不使用 periods 和 values 两个关键字，因此语法比较简单。在外生冲击模块内直接指定外生冲击的标准差或方差，标准差或方差还可以是已知参数的表达式，此外还可以设定两个变量之间的协方差或相关系数，具体请参考 Dynare Reference Manual。

从语法来看，也能“悟出”随机模拟的内在含义。由于没有使用 periods 关键字指定外生冲击出现的时间，因此外生冲击在每一期都存在，而且是随机抽取，因而可能为正值，也可能为负值。理解这一点对理解随机模拟非常重要，这有别于脉冲响应函数。因为脉冲响应函数计算时，虽然也需要随机模拟，但外生冲击只出现在第 1 期，以后各期为 0^①。

4. Dynare 确定性模拟命令：Simul

Simul 命令根据既定设置，求解非线性联立方程，得到模拟值。首先来分析 simul 的求解逻辑，然后再分析 simul 命令的语法。

一般说来，确定性模型具有如下的形式：

$$f(y_{t+1}, y_t, y_{t-1}, u_t) = 0$$

其中， y_t 为内生变量（向量）， u_t 为确定性外生冲击（向量）^②。假设模型具有稳态 \bar{y} ，满足：

$$f(\bar{y}, \bar{y}, \bar{y}, \bar{u}) = 0$$

其中， \bar{u} 为外生变量的稳态值；Dynare 使用命令 steady 来计算模型的稳态值。

通常情况下，在模拟开始前即第 0 期，通常假设系统处于稳态，在接下来的第 1 期即模拟的开始，外生冲击已经完全确定。模拟的基本目的就是来分析系统在外生冲击下的变化路径，是回归原始均衡（如原始稳态值）还是到达一个新的均衡（如新的稳态值）。

确定性模拟，事实上是基于两边界（初值和终值）的数值求解问题。在 Dynare 的具体实现上，确定性模拟其实是使用了一个有限期模型来近似逼近无限期模型，从而数值求解模拟值。具体来说，Dynare 求解如下的非线性联立方程 (Stacked System) 来完成确定性模拟计算：

① Dynare 在计算脉冲响应函数时，使用的算法有别于此处，具体请参考本书“6.2 脉冲响应函数和自定义编程”一节。

② 这里必须满足条件 (identification rule)：方程个数与内生变量个数相同。

$$\begin{cases} f(y_2, y_1, y_0, u_1) = 0 \\ f(y_3, y_2, y_1, u_2) = 0 \\ \vdots \\ f(y_{T+1}, y_T, y_{T-1}, u_T) = 0 \end{cases} \quad (3.8.1)$$

其中, T 为模拟的总期数; y_0, y_{T+1} 为给定的初值和终值。如果写成更紧凑的形式, 方程 (3.8.1) 可表达为

$$F(Y) = 0$$

其中, $Y = (y'_1, y'_2, \dots, y'_T)$ 。Dynare 将使用牛顿算法来求解该联立方程系统。其对应的 Jacobian 矩阵 $\partial F/\partial Y$ 是一个 $n \times T$ 的矩阵, n 是内生变量的个数。可以看出, 当模拟总期数 T 很大时, 雅克比 (Jacobian) 矩阵将是一个非常大的矩阵, 因此大规模模拟会给 Dynare 的计算带来挑战。

在明白确定性模拟计算逻辑后, 接下来看 `simul` 命令的语法。相比随机模拟 `stoch_simul`, 确定性模拟命令 `simul` 要简单很多, 选项相对较少, 具体参见源代码 22。

源代码22 simul命令的基本用法

```
simul ;
simul (options...);
```

确定性模拟的选项只有 6 项, 最常用的就是 `periods`, 用来指定模拟的总期数 T 。一般情况下, 需要指定该选项的值。此外还有一个比较重要的选项用来指定求解算法: `stack_solve_algo`。此选项的值一般不需要更改。在 4.4.3 版本中, Dynare 提供了 7 种不同的算法来处理雅克比矩阵^①, 包括稀疏矩阵 LU 分解 (Sparse LU)、块分解 (Block decomposition)、松弛技术 (Relaxation Technique) 等方法。此处不再深入介绍背后艰深的数学算法问题。其他几个选项也不在一一介绍, 请参考 Dynare Reference Manual。

模拟后, 内生变量和外生变量被存储在不同的位置。内生变量的模拟值被存储在 `oo_endo_simul`, 每个内生变量占一行, 每列表示一期, 存储顺序为声明顺序 (Declaration Order), 即在命令 `var` 后出现的先后顺序。外生变量的模拟值被存储在 `oo_exo_simul`, 每个外生变量占一列, 每行表示一期, 这一点和内生变量截然相反, 但存储顺序为声明顺序, 和内生变量的存储方式相同。一般情况下, `oo_endo_simul` 和 `oo_exo_simul` 存储的模拟结果比 `periods` 指定的多两期, 这是由于包括初值和终值的原因。初值被存储于模

^① 选项 `stack_solve_algo=0,1,2,...,6`, 用来确定不同的算法。默认值为 0, 即稀疏矩阵算法。具体请参考 Dynare Reference Manual, 一般情况下使用默认值即可。

拟结果的第一个元素中，终值被存储在模拟结果的最后一个元素中。

5. 两个简单的示例

首先通过第一个简单的示例来说明初值和终值模块具体设定的内涵，具体参见源代码 23。这个例子的目的是，在给定外生变量变化的情况下，模拟消费和资本存量的最优路径。此处同时设定了 `initval` 和 `endval` 模块，但没有 `shocks` 模块，根据表格 3.6，意味着外生冲击将在模拟第一期就会取值 $\text{eps}=0.1$ ，并且在模拟期内保持不变，而非 $\text{eps}=0$ ，因此外生冲击为永久性冲击。当然可以使用 `shocks` 模块对外生冲击的取值和持续时长进行灵活设定。从资本积累方程中可看到，资本存量是后向变量 (Backward-looking)，因此需要设定初值，即第 0 期取值 k_0 ，并在 `initval` 模块中指定。而从 Euler 方程中，可知消费是前向变量 (Forward-looking)，因此需要指定终值即 c_{T+1} ，并在 `endval` 模块中设定。因此，设定消费的初值和资本存量的终值，其实在模拟中并没有用到，改变消费的初值和资本存量的终值对模拟并没有影响。这说明从理论上，后向变量一般只需要指定初值，前向变量只需要指定终值。但问题是为什么还要指定这些值呢？虽然并未用到这些值，但 Dynare 认为这些值是在第 0 期和第 $T+1$ 期的均衡值或优化的结果，并将其当作给定，并包括在模拟结果中。

源代码23 初值、终值模块与稳态命令示例

```
var c k ...;
varexo eps;
.....
model;
exp(c)^(-sigma)=beta*exp(c(+1))^(sigma)*(R(+1)+(1-delta));
exp(k)=exp(i)+(1-delta)*exp(k(-1));
exp(y)=exp(eps)*exp(k(-1))^(alpha)*exp(n)^(1-alpha);
.....
end;

initval;
c = 0.6;
k = 15;
eps = 0;
end;
steady;

endval;
c = 1;
k = 30;
```

```

eps = 0.1;
end;
steady;
simul( periods=198);

```

然后来看一个完整的例子：带有投资的新古典增长模型。假设家庭选择消费、劳动和资本存量来最大化如下的终身贴现效用：

$$\max_{\{C_{t+j}, N_{t+j}, K_{t+j}\}_{j=0}^{\infty}} E_t \sum_{j=0}^{\infty} \beta^j u(C_{t+j}, N_{t+j})$$

s.t.

$$Y_t = C_t + I_t$$

$$Y_t = A_t F(K_t, N_t)$$

$$K_{t+1} = I_t + (1 - \delta)K_t$$

$$\log A_t = \log A + \rho \log A_{t-1} + \epsilon_t^A, \epsilon_t^A \sim N(0, \sigma^2)$$

其中， C_t 、 Y_t 、 I_t 、 K_t 、 A_t 分别为消费、产出、投资、资本存量和技术冲击； $A=1$ 为技术冲击的稳态值； $\delta > 0$ 为资本存量的折旧率； $\rho > 0$ 为技术冲击的自相关系数。假设效用函数和生产函数分别具有如下的解析形式：

$$u(C_t, N_t) = \frac{[C_t^\theta (1 - N_t)^{1-\theta}]^{1-\tau}}{1-\tau}$$

$$F(K_t, N_t) = [\alpha K_t^\psi + (1 - \alpha) N_t^\psi]^{\frac{1}{\psi}}$$

其中， $\theta > 0$ 为效用函数中消费的权重； $\tau > 0$ 为风险厌恶参数； $\alpha > 0$ 为资本存量的产出份额； ψ 为生产函数中资本存量和劳动的替代弹性参数，生产函数为 CES 形式，即常替代弹性形式。效用函数为消费和劳动的不可分函数。

家庭选择消费、劳动和资本存量以最大化效用函数，可得如下的一阶条件：

$$u_C(C_t, N_t) = \beta E_t \{ u_C(C_{t+1}, N_{t+1}) [A_{t+1} F_K(K_{t+1}, N_{t+1}) + 1 - \delta] \} \quad (3.8.2)$$

$$-\frac{u_N(C_t, N_t)}{u_C(C_t, N_t)} = A_t F_N(K_t, N_t) \quad (3.8.3)$$

模型均衡由 5 个内生变量 C_t 、 K_t 、 N_t 、 Y_t 、 A_t 和 5 个均衡条件：Euler 方程 (3.8.2)、劳动供给方程 (3.8.3)、生产函数、资源约束条件以及技术冲击的 AR(1) 过程^① 组成。从形

① 为了增强 Dynare 源代码 mod 文件的可读性，在编程时，引入了另外两个辅助变量，具体请参考源码。

式上看，模型是随机模型，但可通过对外生冲击的确定性定义，来进行不同的确定性模拟。

此处共给出了 5 个模型文件，第一个模型文件使用 histval 模块修改内生变量的初值；其余 4 个模型文件“名副其实”，具体如表 3.8 所示。

表 3.8 确定性模拟示例 -Mod 文件说明

文件名	说明
det_rbc_1.mod	示例 histval 模块的用法
det_rbc_2_Unanticipated_Transitory.mod	未预期的暂时性冲击
det_rbc_3_Anticipated_Transitory.mod	预期的暂时性冲击
det_rbc_4_Unanticipated_Permanent.mod	未预期的永久性冲击
det_rbc_5_Anticipated_Permanent.mod	预期的永久性冲击

注：模型文件位于本节对应的目录下：\Sources\Chap3_Dynare_Basics\3.8_simulation。

源代码 24 给出了预期永久性冲击的模型文件的部分源代码。在模型雅克比矩阵处理时选择了块分解的方式进行而非默认情况，并且使用了 steady_state_model 模块来计算模型的稳态值，并配合初值和终值设定模块以及 shocks 模块来完成预期永久性冲击的确定性模拟。最后使用了 Dynare 内置画图命令 rplot，画出了消费和资本存量的模拟轨迹图。

源代码24 确定性模拟(预期的永久性冲击)

```
%Mod file: det_rbc_5_Anticipated_Permanent.mod
%Anticipated Permanent shock
%Adapted by Xiangyang Li@SCC, 2017-3-4 ;provided by Sébastien Villemot.
close all; //close all existing figures
var
K, //capital stock
Y, //output
N, // labour
C, //consumption
A, // level technology
a, // log technology
STerm//simplifying variables
;
%usually, exogenous variables has steady state zero
varexo epsA;

parameters beta, theta, tau, alpha, psi, delta, rho, Astar;

beta    = 0.99; //discount factor
theta   = 0.35; //weight of consumption in utility
tau     = 2; //risk aversion
alpha   = 0.35; //share of capital in production
psi     = -0.1; //elasticity of substitution of captial and labor
```

```

delta = 0.025; //depreciation rate
rho = 0.85; //autocorrelation of productivity, i.e. technology
persistence
Astar = 1; // steady state of level of productivity or technology;

%using a divide-and-conquer technique for the Jacobian matrix and
compact
%representation of the modle in a binary file instead of the M files;

model(block,bytecode,cutoff=0);
.....(equilibrium conditions ommited for saving space, see source code
for details)
end;

steady_state_model;
a = epsA/(1-rho);
A = Astar*exp(a);
Y_K=((1/beta-1+delta)/alpha)^(1/(1-psi));
C_K=Y_K-delta;
N_K=((Y_K/A)^psi-alpha)/(1-alpha)^(1/psi);
Y_N=Y_K/N_K;
C_N=C_K/N_K;

% Compute steady state of the endogenous variables.
N=1/(1+C_N/((1-alpha)*theta/(1-theta)*Y_N^(1-psi)));
C=C_N*N;
K=N/N_K;
Y=Y_K*K;
STerm=beta*(((C^theta)*((1-N)^(1-theta)))^(1-tau))/C
      *(alpha*(Y/K)^(1-psi))+1-delta);
end;

%the economy starts from the inital steady states
%immediately followed by a steady command and this is equivilent to
%initval block with endogenous variables are equal to steady states;
initval;
epsA = 0;
end;
steady;

%in period 5, technology increases by 8% permanently
%and this is anticipated as defined by shocks block.
%followed immediately by a steady command and this is equivilent to
%endval block with endogenous variables are equal to steady states;
endval;
epsA = (1-rho)*log(1.08);
end;
steady;

```

```
%a shocks block is used to maintain technology remains at
%its initial level during periods 1-4;
%uncomment for anticipated shock.
%comment for unanticipated shock.
shocks;
var epsA;
periods 1:4;
values 0;
end;

simul(periods=98);

%built-in Dynare function: rplot
%Plots the simulated trajectory of one or several variables.
rplot C;
rplot K;
```

注意：如果将冲击模块注释或删除，那么此时冲击为未预期的永久性冲击，因为此时无法预期外生何时发生，此时缺少外生冲击模块的明确定义，此即 `det_rbc_4_Unanticipated_Permanent.mod`。

在模型 `model` 模块使用了几个选项，其中之一是 `bytecode`，该选项表示使用二进制文件来存储模型 (*.bin)，而不用 M 文件来存储。这样做的好处是提高运行的效率，相比 M 文件，bin 文件中模型存储的形式更加紧致 (Compact)^①，更有利于执行。在模型的雅克比矩阵处理方式上选择了 `block` 选项，即块分解技术。关于 `block` 选项以及 `cutoff` 选项的具体含义，请参考 Dynare Reference Manual。

在运行完模型文件后，此时 Dynare 会给出两个稳态值，分别基于初值和终值而计算出的稳态。如表 3.9 所示，第 1 列基于初值，第 2 列基于终值。可以发现在永久性冲击下，各变量的稳态值均发生变化，包括技术变量的水平值和对数值。

表 3.9 初值与终止分别对应的两个稳态值

STEADY-STATE RESULTS(与初值对应)		STEADY-STATE RESULTS(与终值对应)	
K	6.936 19	K	7.777 96
Y	0.857 369	Y	0.961 419
N	0.325 829	N	0.328 191
C	0.683 964	C	0.766 97
A	1	A	1.08
a	0	a	0.076 961
STerm	2.157 77	STerm	1.852 85

数据来源：Dynare 输出结果。

① 使用 `bytecode` 选项，Dynare 将生成 `filename_dynamic.bin` (动态) 和 `filename_static.bin` (静态) 两个二进制文件；如果不使用 `bytecode` 选项，则会生成两个对应的 M 文件。

从图 3.7 可看出, 对于消费的第 0 期和第 1 期 (第 1 列), 和资本存量在第 99 期和第 100 期之间 (第 2 列), 图形中存在明显的跳跃情况。这就是在第一个示例中提及的问题。对于消费而言, 其初值为 0.684 0(由 `initval` 模块设定而得的稳态值确定), 而模拟的第 1 期值为 0.701 4(参考 `oo_endo_siuml`), 因此存在跳跃情况; 而对于资本存量而言, 其第 99 期的模拟值为 7.709 1(`oo_endo_siuml`), 而终值为 7.778 0(由 `endval` 模块设定而得的稳态值确定), 因而也存在跳跃情况。

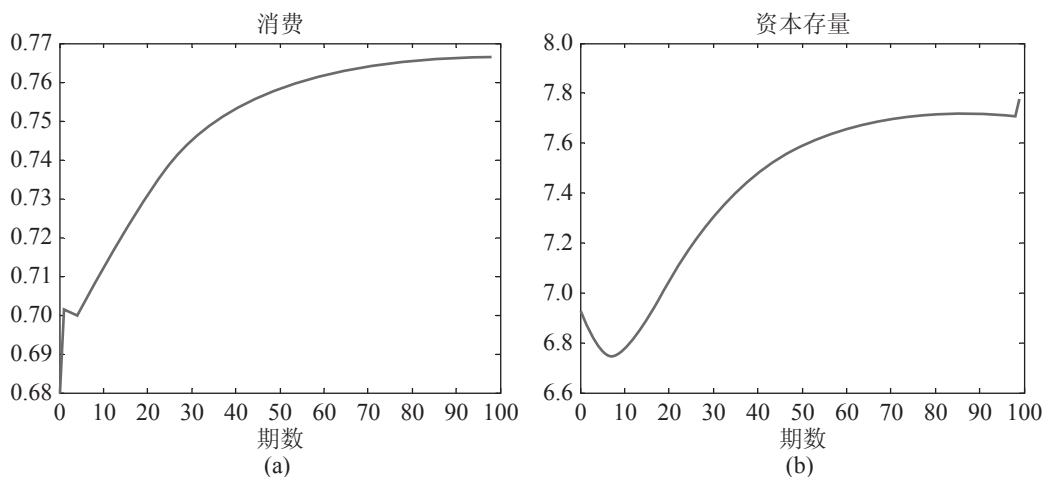


图 3.7 预期的永久性技术冲击下的消费和资本存量的轨迹图

注: (a) 表示消费 (C); (b) 表示资本存量 (K); 水平轴表示模拟期数 (periods)。

此外可以看出, 永久性的冲击使得模型收敛于新的稳态, 也就是说, 从旧的稳态逐步向新的稳态过渡。正向的技术冲击使得消费稳步增长, 而资本存量先下降后逐步升高。因此, 如果模拟的周期足够长, 资本存量对应的变化轨迹则不会出现跳跃的情况。而消费对应的图形中的跳跃情况不会消失, 从表面上看, 这是因为初值和终值不同而造成的。而其本质原因在于消费是前向控制变量, 在外生冲击下能立即跳跃到新的状态 (即新的鞍点路径), 并最终收敛到模型的新稳态, 这也是为什么前向内生控制变量得名跳跃变量 (Jump Variable) 的原因。此外, 资本存量在初期并未发生跳跃, 这是因为资本存量为预先决定的状态变量, 资本形成需要时间, 因而只能缓慢变化, 不会发生跳跃的情况。接下来, 以未预期的暂时性冲击为例来介绍确定性模拟的另外一种情况。相比永久性冲击, 暂时性冲击在图形上最大的不同在于变量最终回归到原始的稳态。具体参见源代码 25。

源代码25 未预期的暂时性技术冲击

```
%Mod file: det_rbc_2_Unanticipated_Transitory.mod
.....
steady_state_model;
.....
end;
steady;

%the economy starts from the steady state
%unanticipated negative transitory shock during period 1:3.
%the economy will eventually return back to initial steady state.
shocks;
var epsA;
periods 1:3;
values -0.1;
end;

simul(periods=98);
.....
```

注：省略部分和源代码 24 相同。

此处并未设置初值模块，此时外生变量的初值被认为 0，内生变量的初值仍被 Dynare 赋予为其稳态值。该处设定了在模拟的前 3 期使用了负向的外生技术冲击，然后以后各期的外生冲击变为 0。

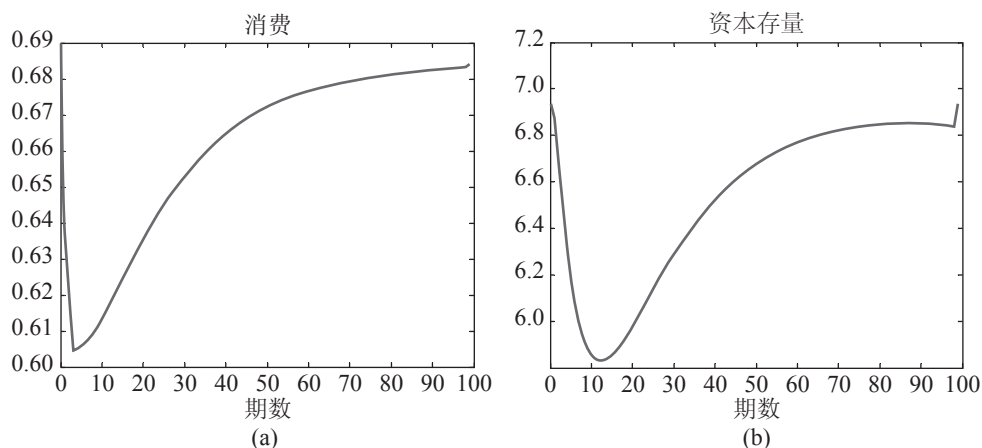


图 3.8 未预期的暂时性冲击下的消费和资本存量的轨迹图

注：(a) 表示消费 (C)；(b) 表示资本存量 (K)；水平轴表示模拟期数 (periods)。

消费和资本存量的初始稳态值同表 3.9。图 3.8 显示了暂时性负向 (Negative) 技术冲击下, 消费 (第 1 列) 和资本存量 (第 2 列) 出现剧烈的下降, 随着外生冲击的消失, 而后缓慢上升, 消费和资本存量最终回归到原来的稳态值, 而非新的稳态。同样地, 消费在初期和资本存量在末期存在跳跃的情况, 具体可参考 `oo_endo_simul` 进行验证。

关于其他几种确定性模拟的示例, 请参考提供的 Dynare 模型文件, 此处不再一一介绍。在介绍完确定性模拟之后, 下一节来聚焦随机模拟分析。

3.9 随机模拟分析: `stoch_simul`

随机模拟分析是最常用的 Dynare 计算命令之一。顾名思义, 随机模拟首先使用扰动算法完成基本的模型求解任务, 然后在求解的基础上计算内生变量的脉冲响应、各阶矩, 最后进行随机模拟, 即根据随机抽取的外生冲击的样本, 模拟出内生变量的增长路径。在本书“1.1.7 脉冲响应和随机模拟”一节中, 已经简单介绍了随机模拟的概念, 并手动编程实现随机模拟, 即实现了 `stoch_simul` 的部分功能。手动编程能够帮助初学者理解 `stoch_simul` 命令背后的基本原理, 但每次手动编程工作繁重而且容易出错, 因此使用 `stoch_simul` 命令不失为一个好办法。

3.9.1 随机模拟选项简介

此处简略介绍 `stoch_simul` 命令及其选项。在 Dynare Reference Manual 中, `stoch_simul` 命令的选项多达 44 项, 此处仅选取常用的选项加以介绍。

- **drop = xx**, 在计算内生变量的各阶矩时, 需要首先指定丢弃的期数(burn-in): xx期, 但这并不影响模拟变量本身, 即`oo_endo_simul`不受影响。默认情况下xx=100。
- **hp_filter =xx**, 在计算内生变量的各阶矩时, 首先进行HP滤波, 同样地, 这并不影响模拟变量本身, 即`oo_endo_simul`。此处的xx代表通常的lambda值, 具体取值可参考本书“1.3.4 HP滤波的基本逻辑”。
- **irf=xx**, 计算xx期的脉冲响应, 默认情况下xx=40, 即计算40期的脉冲响应。关于脉冲响应的计算会在本书“6.2 脉冲响应函数和自定义编程”一节中进一步探讨。
- **irf_shocks = (shock1,shock2, ...)**, 针对给定的外生冲击计算脉冲响应, 多个外生冲击以逗号隔开。默认情况下, 即不指定该选项, 计算全部外生冲击的脉冲响应。
- **irf_plot_threshold = xx**, 指定画出脉冲响应图形的门槛大小。当某变量的脉冲响应的各期值的绝对值的最大值小于xx, 那么该变量将不会在脉冲响应图中画出,

默认情况是百亿分之一，即1E-10。

- **nocorr**, 指定不要在Matlab命令行窗口显示相关系数矩阵，默认显示，不影响计算，即仍然计算相关系数并存储起来。
- **nofunctions**, 指定不要在Matlab命令行窗口显示政策函数和转换方程，默认显示，不影响计算。
- **nomoments**, 指定不要在Matlab命令行窗口显示内生变量的各阶矩，默认显示，不影响计算。
- **nograph**, 指定不要显示和保存图形，即不画图，默认为显示和保存。
- **nodisplay**, 指定不要显示图形，但保存图形到硬盘。
- **noprint**, 指定不要显示任何计算信息，对于循环调用非常有帮助，能节约不少时间，但启动和预编译信息仍然显示。
- **order =xx**, 指定Taylor近似的阶数，可接受的值为1, 2, 3。默认为2，即二阶近似求解。
- **periods=xx**, 指定随机模拟的期数，同时也是决定计算理论矩还是模拟矩的关键所在。xx为非零时，计算模拟矩，否则计算理论矩。在外生变量随机抽取样本时，如果外生冲击之间是相关的，即M_.Sigma_e是非对角矩阵，那么Dynare会使用类似于VAR文献中的处理办法来对M_.Sigma_e进行Cholesky分解，此时外生变量的次序则依据varexo的声明顺序进行分解。默认情况下periods=0。
- **pruning**, 指定使用剪枝算法，即在迭代计算过程中丢弃高阶项。如果指定该选项，Dynare在二阶计算时，将使用由Kim, Schaumburg & Sims (2008)提出的剪枝算法；在三阶计算时，使用由Andreasen, Fernández-Villaverde & Rubio-Ramírez (2013)^①提出的剪枝算法。本书“6.3 二阶随机模拟中的一些问题”中会进一步介绍剪枝算法。

3.9.2 随机模拟示例

以本章第一节中的RBC模型为例。首先进行随机模拟，然后示例如何提取模拟结果中的水平变量进行画图分析。具体语法参见源代码26。

^① Andreasen, Martin M., Jesús Fernández-Villaverde, and Juan Rubio-Ramírez.: *The Pruned State-Space System for Non-Linear DSGE Models: Theory and Empirical Applications*, NBER Working Paper, 2013,18983 (Jesús Fernández-Villaverde 的网站上提供了更新的版本, 2017-7).

源代码26 随机模拟冲击示例

```

stoch_simul(periods=1000,irf=40,order=1) c k lab z;
.....
figure(1)
subplot(2,2,1)
plot(oo_.endo_simul(1,:), '-b', 'Linewidth', 1) %调用oo_.endo_simul
title('Consumption ')

subplot(2,2,2)
plot(oo_.endo_simul(2,:), '-b', 'Linewidth', 1)
title('Capital Stock ')

subplot(2,2,3)
plot(oo_.endo_simul(3,:), '-b', 'Linewidth', 1)
title('Labor')

subplot(2,2,4)
plot(oo_.endo_simul(4,:), '-b', 'Linewidth', 1)
title('Technology Shock')

```

此处，随机模拟命令中指定了3个选项，即模拟1 000期、计算40期脉冲响应图、进行一阶泰勒近似求解。在结果显示时只针对列示的4个内生变量（实际上是全部变量），未被列示的变量将不予显示（包括政策函数、脉冲响应图、各阶矩等不予显示），但不影响计算。

根据前文的分析，随机模拟的结果存储于 `oo_.endo_simul` 结构数组中，其行对应着以声明顺序排序的内生变量，列为模拟的期数，因此 `oo_.endo_simul` 为 $4 \times 1\,000$ 的矩阵。接下来，从 `oo_.endo_simul` 数组中提取4个内生变量的模拟值，如图3.9所示^①。从图3.9可看出，在模拟期内，技术冲击实际上就是白噪声过程，其均值为0，标准差为

$$\text{std}(Z_t) = \frac{s}{\sqrt{1-\rho^2}} \approx 0.0224$$

其中，参数 s ， ρ 的含义参见本书“3.1 DSGE 模型：一个简单的例子”。可以验证模拟样本的标准差与此值非常接近^②。在此冲击驱动下，其他内生变量也呈现出不同幅度的变化。

① 具体 Matlab 源代码请参考源代码 26。对其他存储结果的调用可如法炮制。具体请参考本章第一节中的模型文件：`\Sources\Chap3_Dynare_Basics\3.1_simple_example\GrowthApproximate.mod`。

② 在 Matlab 的命令行中输入 `std(oo_.endo_simul(4,:))`，即外生技术冲击模拟样本的标准差，可发现此值与理论值非常接近。

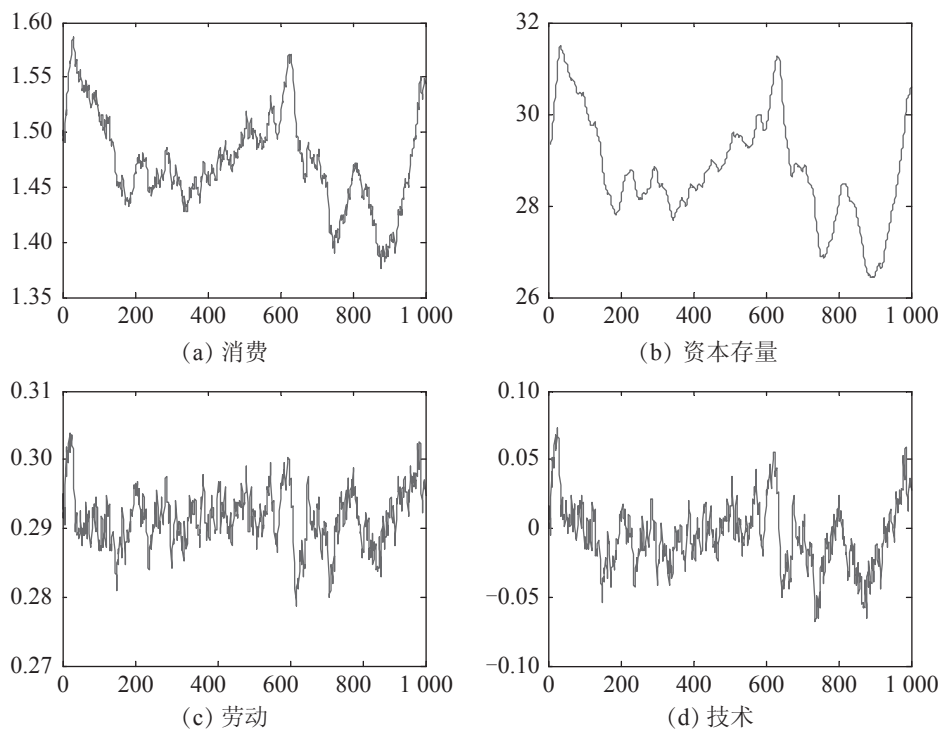


图 3.9 随机模拟 (消费、资本存量、劳动和技术冲击)

3.10 参数估计简介

模型中结构参数的赋值有两种方法：第一种为校准 (Calibration)；第二种为估计 (Estimation)，即使用实际统计或观测数据对参数进行估计，从而获取估计值。简单来说，Dynare 对参数进行估计就是使用某些基本方法，比如贝叶斯 (Bayesian) 估计如 Schorfheide(2000)、Rabanal & Rubio-Ramirez (2005)、Smets & Wouters (2003; 2007) 等，极大似然估计 (MLE)，如 Ireland (2004) 等，并在某些假设下 (比如假定参数服从某些概率分布)，从数据中“发现”参数的“真实”取值^①。

^① 由于真实值是无法知道的，只能估计。此处“真实”二字加了引号，这表明估计得到的值，不是绝对意义上的真实值，而是相对意义或者有条件的真实值。之所以称之为相对或有条件的真实值，是因为估计是在一定的前提下进行的，这包括估计方法的选择，先验分布的选择，数据序列本身和序列长度的选择等，这些因素的差异将会导致估计值的差异。

Herbst & Schorfheide (2016) 是一本专门介绍 DSGE 模型贝叶斯估计的专著，不仅介绍了线性化 DSGE 模型的估计，更深入地讨论了使用粒子滤波 (Particle Filter) 估计非线性 DSGE 模型，并给出具体的例子，值得仔细阅读。

本节结合 Dynare 实践，对极大似然估计和贝叶斯估计进行简单介绍，然后对贝叶斯估计中的马尔可夫链—蒙特卡洛 (MCMC) 方法进行阐述，最后给出一个例子。

3.10.1 极大似然估计与贝叶斯估计的基本逻辑

在本小节中，主要阐述最大似然估计和贝叶斯估计的基本原理和逻辑，但并不打算阐述两者背后复杂的数理知识，只是点到为止。在继续阐述之前，熟悉表 3.10 中各符号的含义，将会非常有助于理解接下来的内容。

表 3.10 本节定义的几个符号与其含义

符 号	含 义
θ	待估参数，一般为向量
$Y^{\text{data}} \equiv \{Y_1^{\text{data}}, \dots, Y_T^{\text{data}}\}$	观测数据，用于估计的事实统计数据
$p(\theta)$	先验分布，Prior Density Function, PDF
$p(Y^{\text{data}} \theta)$	似然函数，Likelihood Function
$p(Y^{\text{data}})$	边际似然函数，用于估计模型的拟合度
$p(\theta Y^{\text{data}})$	后验分布密度函数，Posterior Density Function, PDF
$Ker(\theta Y^{\text{data}})$	Kernel, 核密度函数
$\theta^* = \arg \max_{\theta} \log Ker(\theta Y^{\text{data}})$	众数，mode

1. 极大似然估计

极大似然估计 (Maximum Likelihood Estimation, MLE) 就是选择参数 θ 来最大化如下的似然函数 (Likelihood)，其原理较为简单：

$$\begin{aligned}
 p(Y^{\text{data}} | \theta) &\equiv p(Y_1^{\text{data}}, Y_2^{\text{data}}, \dots, Y_T^{\text{data}} | \theta) \\
 &= \underset{\text{条件概率}}{p(Y_1^{\text{data}} | \theta)} \times p(Y_2^{\text{data}} | \theta, Y_1^{\text{data}}) \\
 &\quad \times \dots \times p(Y_t^{\text{data}} | \theta, Y_1^{\text{data}}, \dots, Y_{t-1}^{\text{data}}) \\
 &\quad \times \dots \times p(Y_T^{\text{data}} | \theta, Y_1^{\text{data}}, \dots, Y_{T-1}^{\text{data}})
 \end{aligned}$$

其中，条件概率 $p(Y_t^{\text{data}} | \theta, Y_1^{\text{data}}, \dots, Y_{t-1}^{\text{data}})$ 使用 Kalman 滤波进行计算。关于 Kalman 滤波的详细介绍，请参考 Hamilton(1994)，此处不再详细介绍。后续会举例说明，如何在 Dynare 中使用极大似然估计的方法进行参数估计。

2. 贝叶斯估计

贝叶斯估计 (Bayesian Estimation) 的基本原理，直观地说非常简单，就是在参数先验分布 (Prior) 的基础上，结合数据的信息 (Y^{data})，找到参数的后验分布 (Posterior)。也就是说，贝叶斯估计的实质就是一个简单的映射，依据数据信息，将先验分布映射成后验分布。由于先验分布是依据经验而假设的分布，可能存在“偏误”，而数据的作用其实是“纠偏”，将其中有用的信息融入，以“纠正偏误”，从而形成后验分布，如图 3.10 所示。

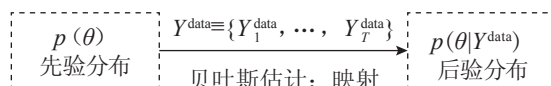


图 3.10 贝叶斯估计的基本原理

从逻辑推导上说，贝叶斯估计背后就是概率论中著名的贝叶斯法则 (Bayes' Rule)，即参数的后验分布由条件概率公式：

$$p(Y^{\text{data}} | \theta) \times p(\theta) = p(Y^{\text{data}}, \theta) = p(\theta | Y^{\text{data}}) \times p(Y^{\text{data}}) \quad (3.10.1)$$

推导出来：

$$p(\theta | Y^{\text{data}}) = \frac{p(Y^{\text{data}} | \theta) \times p(\theta)}{p(Y^{\text{data}})} \quad (3.10.2)$$

从数据使用和先验假设的角度来看，贝叶斯估计位于极大似然估计和参数校准 (Calibration) 之间。参数校准就是依据经验假设，给参数指定一个值，其实是在指定先验分布，只不过这个先验分布比较特殊，其对应的方差或标准差为 0，也就是说是一个固定的点。而极大似然估计直接使用数据，没有给定先验分布 (或者说给定的先验分布的方差为无穷大)，然后极大化似然函数，从而找到使得似然函数取值最大的参数值作为估计值。而贝叶斯估计恰恰结合了这两点：假定先验分布并使用数据，正如贝叶斯法则 (3.10.2) 那样，将两者联系在一起。Hamilton(1994) 提供了一个非常直观的例子来理解参数校准、极大似然估计与贝叶斯估计之间的联系^①。

贝叶斯估计中先验分布则是一般意义上的分布，即标准差不为 0，不再是一个点，

① PP352, Chapter 12, Hamilton, J. D. *Time Series Analysis*. Princeton, NJ, Princeton University Press, 1994.

而是指定一个区域，或更严格地说，是整个参数空间中的一个子区域，作为候选区域。先验分布可以理解为施加于似然函数的“权重”或“偏好”，使得参数的选择范围集中于均值附近，而非离均值较远。换句话说，先验分布使得“巨大的参数选择空间”变小，从而使似然函数集中于更“有意义”的子区域去寻找合适的参数估计值。

Dynare 中贝叶斯估计可分为两步：第一步，使用数值求解算法，求出使得似然函数取最大值时的众数 (Mode)；第二步，从众数开始，使用 MCMC 方法模拟后验分布，然后计算各种感兴趣的矩 (Moments)。在下一节中，将对此进行简要介绍。

3.10.2 马尔可夫链——蒙特卡洛(MCMC)方法

由于参数的后验分布 (密度函数) 在多数 DSGE 模型中 (甚至某些 RBC 模型中) 无法用解析形式来表达，不可能通过贝叶斯法则计算出来，只能通过随机抽样的方法，即所谓的蒙特卡洛 (Monte Carlo) 方法来近似逼近后验分布。马尔可夫链—蒙特卡洛方法 (Markov Chain-Monte Carlo, MCMC) 是一类方法的统称，是贝叶斯估计的逻辑方法和框架，而非具体估计的算法。此处使用的 MCMC 的具体方法是 MH 算法 (Metropolis-Hastings Algorithm)，后续会有进一步介绍。

1. 核密度函数

假设已经获得待估参数 θ 的抽样 $\theta^{(1)}$ 、 $\theta^{(2)}$ 、 \dots 、 $\theta^{(T)}$ (即马尔可夫链, Markov Chain)，那么后验分布的密度函数 $p(\theta|Y^{\text{data}})$ 可以通过如下的方式获得^①：

$$p(\theta|Y^{\text{data}}) = \lim_{T \rightarrow \infty} \text{Frequency}(\theta^{(i)} \text{ draw using MCMC})$$

其中，待估参数抽样通过 MCMC 方法抽取，以频率的极限值作为后验分布概率密度。此外，在获取待估参数 θ 的抽样 $\theta^{(1)}$ 、 $\theta^{(2)}$ 、 \dots 、 $\theta^{(T)}$ 后，可以计算或估计各种关于 θ 的后验分布的统计量，比如期望和方差 $E(\theta)$ 、 $\text{var}(\theta)$ ，以及边际后验分布

$$q(\theta_i|Y^{\text{data}}) \equiv \int_{\theta_{j \neq i}} p(\theta|Y^{\text{data}}) d\theta_{j \neq i}$$

和边际似然函数

$$p(Y^{\text{data}}) = \int_{\theta} p(Y^{\text{data}}|\theta) \times p(\theta) d\theta$$

因此，MCMC 方法的核心是在后验分布没有解析解的情况下，如何获取后验分布及其抽样。

^① 在 Matlab 中，可以使用命令 hist 画出抽样的直方图来直观地看出后验分布概率密度函数的近似形状。

由 (3.10.2) 可得

$$p(\theta|Y^{\text{data}}) \propto p(Y^{\text{data}}|\theta) \times p(\theta) \equiv \text{Ker}(\theta|Y^{\text{data}}) \quad (3.10.3)$$

其中, $\text{Ker}(\theta|Y^{\text{data}})$ 被称为后验分布的核密度函数 (Posterior Kernel), 与后验分布 $p(\theta|Y^{\text{data}})$ 相差一个常数 [取对数后, 并假设数据的边际似然函数 $p(Y^{\text{data}})$ 为常数, 即不依赖于待估参数]。在贝叶斯估计中, 同样使用 Kalman 滤波对似然函数 $p(Y^{\text{data}}|\theta)$ 进行计算, 该似然函数是待估参数的复杂函数, Dynare 将使用数值求解方法来找到最大值对应的参数^①, 被称为众数:

$$\theta^* \equiv \arg \max_{\theta} p(\theta|Y^{\text{data}}) = \arg \max_{\theta} \text{Ker}(\theta|Y^{\text{data}})$$

这个众数是 Dynare 的一个重要的输出结果。一般说来, 对于单众数 (单峰) 对称分布, 均值和众数是一致的 (Hamilton, 1994)。在某些情况下, 均值和众数存在差异, 但随着样本的增大, 众数将趋近于均值 (DeGroot, 1970, PP236)。

2. MH 算法

在获得似然函数的众数后, 接下来就是寻找后验分布自身。在大多数 DSGE 模型中, 核密度函数本身没有解析解, 只能通过 Kalman 滤波算法进行数值计算和估计, 从而后验分布也没有解析解。在对后验分布知晓甚少的情况下 (此时知晓后验分布的众数、由 Kalman 滤波估计的似然函数), 如何才能找到其密度函数呢? 由于其没有解析表达式, 因而只能通过随机模拟来抽样, 近似逼近密度函数。这正是 MH 算法所做的事情。MH 算法是文献中公认的一个较为高效的算法, 思想类似于常见的接受拒绝算法 (Acceptance-Rejection Algorithm, AR), 对抽取的样本选择性接受, 因而存在一个适意的接受率问题 (Acceptance Rate) 后续会有进一步介绍。MH 算法通过对核密度函数进行随机模拟, 抽取一系列的样本 (Markov Chain), 然后画出样本直方图来近似逼近后验分布的密度函数。当样本足够大时 (比如抽样十万或百万次), 直方图的锯齿状会逐渐消除, 趋于平滑和连续。

了解 MH 算法, 将有助于 Dynare 贝叶斯估计编程, 特别是估计命令 `estimation` 选项的设置。因而此处做简单介绍, 其基本的步骤如下:

第一步: 选择一个初始点, $\theta^{(1)} = \theta^*$, 通常情况下选择众数作为初始点;

第二步: 抽取 $\theta^{(t)}$, $t > 1$ 。首先从跳跃分布中 (Jumping Distribution) 抽取一个候选样本 (Candidate) s :

^① 该数值求解方法, 由估计命令 `estimation` 的选项 `mode_compute=xx` 来确定, 具体见后续关于该命令详细介绍。

$$s \sim \theta^{(t-1)} + c \times N(0, V_{\text{mode}}) = N(\theta^{(t-1)}, c^2 V_{\text{mode}}) \quad (3.10.4)$$

其次计算接受比率 r :

$$r \equiv \frac{p(Y^{\text{data}} | s) p(s)}{p(Y^{\text{data}} | \theta^{(t-1)}) p(\theta^{(t-1)})} = \frac{\text{Ker}(Y^{\text{data}} | s)}{\text{Ker}(Y^{\text{data}} | \theta^{(t-1)})}$$

最后根据如下的规则，以确定是否接受或拒绝候选样本 s :

$$\theta^{(t)} = \begin{cases} s & \text{with probability } \min(r, 1); \\ \theta^{(t-1)} & \text{otherwise.} \end{cases}$$

通常情况下，会从 $U(0, 1)$ 一致分布中抽取样本，和接受比率 r 比较大小，用于判断是否接受还是拒绝候选样本。

第三步：不断重复第二步，直到获取足够的样本。从数学上可证明，经过这样抽取的样本最终收敛于后验分布。因此只要样本足够大，就可以获得足够的近似。

第二步中，首先要对跳跃分布做出简单说明。关于跳跃分布的详细背景、定义以及如何选择的问题，已经超出了本书讨论的范围。简单地说，如果一个分布很难或无法知道如何从中抽样，则此时需要借助于另一个分布来辅助抽样，这个分布就是跳跃分布。跳跃分布的选择，其实很难说有一个统一的标准。在 AR 抽样中，如果从原始分布 $f(x)$ 直接抽取样本很困难，则一般会选择跳跃分布 $g(x)$ 和某个合适的常数 λ ，满足 $f(x) \leq \lambda g(x)$ ，然后从 $g(x)$ 抽样，经过选择后留下合适的样本。而此处由于对后验分布知之甚少，因此无法确定一个合适的跳跃分布。通常情况下要求这个跳跃分布是关于均值对称 (Symmetric)。于是，一个最自然的选择就是正态分布^①。

An & Schorfheide(2007) 指出使用众数作为该正态分布的均值，使用一个“调整”的 (Scaled) 的方差—协方差矩阵作为该正态分布的协方差矩阵。这样，至少在众数周围，可以对后验分布做非常好的近似^②。这正是 Dynare 的做法。初始值选择众数作为均值，并使用：

$$V_{\text{mode}} \equiv \left[-\frac{\partial^2 L}{\partial \theta \theta'} \right]_{\theta=\theta^*}^{-1}$$

作为正态分布的方差—协方差，其中 $L \equiv L(\theta) \equiv \log p(Y^{\text{data}} | \theta) + \log p(\theta)$ 。这里“调整”

① 一般情况下，参数可看作是渐进服从正态分布 (asymptotically normal)。

② An, S. and F. Schorfheide (2007). *Bayesian Analysis of DSGE Models*. *Econometric Reviews* 26 (2-4): 113-172. “... The algorithm constructs a Gaussian approximation around the posterior mode and uses a scaled version of the asymptotic covariance matrix as the covariance matrix for the proposal distribution. This allows for an efficient exploration of the posterior distribution at least in the neighborhood of the mode...”

是指式 (3.10.4) 中的系数 c ， c 值的变化直接影响跳跃分布的方差和抽样。Dynare 的估计命令中有一个专门的选项，`mh_jscale`，用以设定 c 。作为关键参数， c 的大小还直接影响随机模拟样本的接受率。在方差和协方差矩阵 V_{mode} 给定的情况下，较小的 c 值将导致方差较小，从而使得样本选择过于集中在均值附近，从而导致接受率较高，很难遍历较大的参数空间，而且有可能“困在”某个“局部最大值”的地方，从而导致马尔可夫链 (Markov Chain) 收敛较慢。反之，当 c 值较大时，方差较大，从而会使得样本选取范围更大，同时这也意味着样本接受率较低，往往会使得模拟“困在”后验分布的尾部，即概率密度较小的地方。因此适意的 c 值较为重要，后续会做进一步介绍。

虽然样本最终收敛于后验分布，但在模拟的初期却面临一个尴尬：初期抽取的样本无法知晓其分布，也就说是初期抽取的样本可能根本就不是来自于后验分布，或者说与后验分布差异很大。此时，一个标准的做法就是丢弃一定数量前期抽取的样本，这在统计中被称为预迭代期 (Burn-in Period)。Dynare 中使用选项 `mh_drop=xx` 来定义丢弃的样本数量，默认值为丢弃 50%。

在第二步计算接受比率时，可看到接受比率是密度函数之比。在给定上一期抽取的样本 $\theta^{(t-1)}$ 后，当前抽取的样本 s 的似然值越大，接受比率越大，因而其被接受的概率越高，这也符合直觉。

最后是关于重复次数的问题。在实际模拟时，重复 10 000 次算是比较小的数值，理想的情况多达 1 000 000 次，甚至更多。但这种情况下将会非常耗时 (Time Intensive)，特别是多个并行的马尔可夫链同时进行 (默认情况下是两条链同时进行模拟，下文会提及，由估计命令的选项 `mh_nblocks` 决定)。因此，可根据问题的需要和实际计算资源，选择合适的重复次数。

为了直观地理解 MH 算法，来看一个简单的例子。在该例子中，使用 MCMC 来近似逼近均值为 1 的对数正态分布 (Log Normal) 的密度函数，也就是说将对数正态分布看作后验分布。此后验分布的均值已知 (因而方差)，无须估计，此处反过来使用 MCMC 算法逼近后验分布，检验其效果。因此，在计算跳跃分布的方差时，直接使用了对数正态分布的密度函数 (Matlab 的内置函数 `lognpdf`)，而且在计算接受比率时也同样直接使用了该函数。同时为了对比分析，使用了拉普拉斯 (Laplace) 算法 (见本节第三部分) 逼近对数正态分布的密度函数。

源代码 27 给出了所有关于模拟的细节和代码解读，应该仔细研读，以掌握 MCMC 算法。整个代码可分为如下几个部分：第一部分是模拟前的准备，寻找跳跃分布的均值和方差 (此处跳跃分布选择正态分布，但需要确定此正态分布的均值和方差)，即找出给定对数正态分布的众数作为均值，然后使用二阶导数的近似计算公式计算出方差。第二

部分是进行 MCMC 模拟。此模拟部分的代码是和前面 MH 算法的步骤一一对应的，并力争做到符号对应，以方便理解和阅读。最后是模拟后的简单处理。首先对直方图进行标准化处理，即将直方图中的各个小矩形的面积之和标准化为 1，然后画出相应的图形。对于 Laplace 近似算法，则直接使用了本节第三部分的结果，在画图函数 plot 中直接调用 Laplace 近似的结果，即近似服从正态分布。

源代码27 对数正态分布的MCMC和Laplace近似^①

```
%% Log-Normal Approximation: MCMC_Laplace_Approximation.m
%This m file is to illustrate the MCMC and Laplace approximation
%to a lognormal distribution important in Financial Accerlerator Model

clear all
close all

%parameter in the jump distribution(mh_jscale):
c=8.5;
%number of total simulation, you could change this value depends on need
number_of_sim=20000;

%parameters of the Lognormal distribution with expected value equal to 1.
%see “5.2 对数正态分布的基本概念” for more details.
lognormal_mean=-1/2;
lognormal_sigma=1;

%find the mode of lognormal
x_interval=0:.001:3; %desired or defined domain of lognormal
Y = lognpdf(x_interval,lognormal_mean,lognormal_sigma);
[maxY,i]=max(Y);
lognormal_mode=x_interval(i);

epsilon=1e-6; %a small number for 2nd derivatives
s=lognormal_mode;

%get the variance required for the Laplace approximation and the MCMC:
f_right=log(lognpdf(s+2*epsilon,lognormal_mean,lognormal_sigma));
f_left=log(lognpdf(s-2*epsilon,lognormal_mean,lognormal_sigma));

%approximation for 2nd derivatives of lognormal distribution
f_2nd_der=(f_right-2*log(maxY)+f_left)/(2*epsilon*2*epsilon);

%the standard deviation (variance) for the jump distribution;
%in our text, we use Vmode as the variance
Vmode=sqrt(-1/f_2nd_der);
```

① 源文件地址：\Sources\Chap3_Dynare_Basics\3.10_estimation\MCMC_Laplace_Approximation.m。

```

randn('seed',1); %set seed, for replication

theta(1)=lognormal_mode; %starting simulation from mode

acceptance=0; %count how many samples are accepted

%how long its takes for simulation
tic
%codes for MCMC simulation
for ii = 2:number_of_sim
    %draw from [0,1] uniform distriubtion
    u=rand;

    %draw from the jump distribution starting from mode
    s=theta(ii-1)+c*randn*Vmode;

    x0=theta(ii-1); %previous draw

    %the lognormal random variable is non-negative, so a negative
    candidate has density zero.
    if s < 0
        theta(ii)=theta(ii-1); %reject the draw, i.e. discard negative
        draws
    else
        fn=lognpdf(s,lognormal_mean,lognormal_sigma);
        fd=lognpdf(x0,lognormal_mean,lognormal_sigma);
        %acceptance ratio
        r=fn/fd;
        %accept the new draw with prob. min(r,1)
        if u < r
            theta(ii)=s;
            acceptance=acceptance+1;
        else
            theta(ii)=theta(ii-1); %reject the draw
        end
    end
end
toc
%burn-in periods
burn_in=min([1000,round(.01*number_of_sim)]);
%the histogram for the sample
%N, the vector of numbers in each bins which correspondes the height
of
    %each bin in the histogram
%X, the centers of each bin
[N,X]=hist(theta([burn_in:number_of_sim]),150);

```

```

%the width of each bin should be the same
width1=max(diff(X));
width2=min(diff(X));
width=(width1+width2)/2;

%to ensure the sum of the area of all bins equal to unity since we
are
%approximating the probability density function using histogram
weight=width*sum(N);
%you will find sum(n*width) ==1 where n*width represents the area of
each
%bin, this implies the Riemann integral over the histogram is unity.
n=N/weight;

%restrict the samples to the x_interval, the defined domain

upper_=max(find(X<max(x_interval)&X>min(x_interval)));
lower_=min(find(X<max(x_interval)&X>min(x_interval)));
rsi=[lower_:upper_];

%plot the lognormal, Laplace approximation and MCMC simulation
% Laplace approximation:see(3.10.6).
plot(x_interval,Y,X(rsi),normpdf(X(rsi),lognormal_mode,Vmode),'x-
',X(rsi),n(rsi),'ro-')

t1=['Lognormal distribution, mode = ',num2str(lognormal_mode),' \mu =
',num2str(lognormal_mean), ' \sigma = ',num2str(lognormal_sigma)];
t2=['MCMC, c = ',num2str(c),' acceptance ratio=
',num2str(100*acceptance/number_of_sim),'%'];

legend(t1,'Laplace approximation',t2)
axis tight

```

有一点需要提及，在随机模拟中由于后验分布即对数正态为非负，而跳跃分布为正态分布，可以为负，因此对抽取的负值样本应舍弃，因此代码中多出了一个 if 判断语句：“if s < 0”来处理这个问题。

上述源代码同时画出图 3.11。在直方图标准处理时，首先计算了所有小矩形的面积，将其设定为总权重 (weight)，然后将每个小矩形的高度 N 标准化为 N/weight 。从而保证了总面积之和为 1，满足了概率密度函数在定义区间上积分为 1。此处的标准化并非严格意义上的标准化，因为此处对数正态分布的定义区间仅仅考虑了 $[0,3]$ ，而非 $[0, \infty)$ 。

图 3.11 给出了 MCMC 和 Laplace 近似方法的结果。可以看出 MCMC 的逼近结果较好，而 Laplace 近似在众数 (mode=0.223 0) 附近取值较大，给予了较多的权重，因此尾部

的权重较低，总体而言偏离较大。其实可以在更苛刻的环境中考察两者的近似情况，比如可考察具有双峰 (bimodal) 的概率分布^①，结果是同样的，而且随着模拟次数的增长，MCMC 几乎和后验分布重合，而 Laplace 算法逼近的效果较差，这说明 MCMC 算法是一种比较可靠的近似算法。

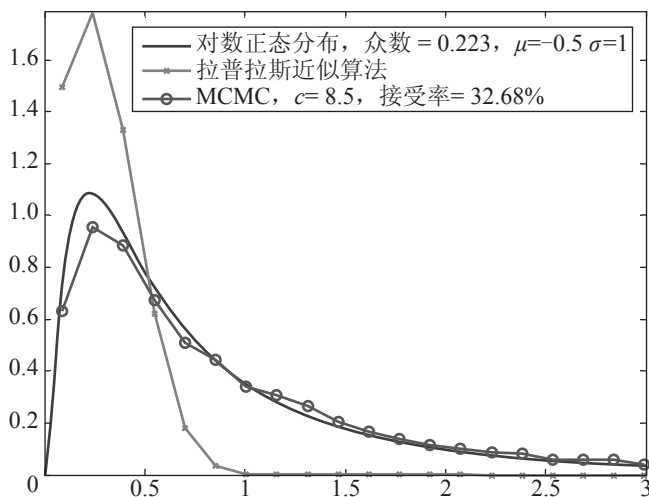


图 3.11 对数正态分布的 MCMC 和 Laplace 逼近

3. Laplace 近似算法

Dynare 在贝叶斯估计后，一般会给出如下的输出信息：

```
Log data density [Laplace approximation] is xxx.xxx.
```

此信息即对数数据密度值，常常用于模型的拟合度比较。该数量值越大，意味着拟合度越高。比如：同一个模型的一个参数，给定两种不同的先验分布，在其他条件不变，对应较大数据密度的先验假设被认为更合适。

从输出结果中可看到，数据密度的计算使用了 Laplace 近似算法。从本质上讲，该算法就是在某个给定点处的二阶 Taylor 展开。通常情况下，该给定点为众数。核密度函数的 (对数) $L(\theta)$ 的定义为

$$L(\theta) \equiv \log p(Y^{\text{data}} | \theta) + \log p(\theta) \quad (3.10.5)$$

① 两个均值不同的正态分布密度的线性组合，即一个双峰的密度函数。这区别于两个正态随机变量之线性组合仍为正态分布的结论。

在众数处对 $L(\theta)$ 进行二阶 Taylor 展开, 可得:

$$L(\theta) \approx L(\theta^*) + L_{\theta}(\theta^*)(\theta - \theta^*) - \frac{1}{2}(\theta - \theta^*)' V_{\text{mode}}^{-1}(\theta^*)(\theta - \theta^*), V_{\text{mode}}^{-1}(\theta^*) \equiv - \left. \frac{\partial^2 L(\theta)}{\partial \theta \partial \theta'} \right|_{\theta = \theta^*} \quad (3.10.6)$$

其中, θ^* 为后验分布的众数, 因此 $L_{\theta}(\theta^*)=0$, $V_{\text{mode}}^{-1}(\theta^*)$ 为正定矩阵 (Positive Definite)。于是由 (3.10.6) 可得:

$$p(Y^{\text{data}} | \theta) p(\theta) \approx p(Y^{\text{data}} | \theta^*) p(\theta^*) \exp\left(-\frac{1}{2}(\theta - \theta^*)' V_{\text{mode}}^{-1}(\theta^*)(\theta - \theta^*)\right)$$

利用正态分布的基本性质, 可推导出在 Laplace 近似下^①, 后验分布近似为如下的正态分布

$$\theta | Y^{\text{data}} \sim N(\theta^*, V_{\text{mode}}) \quad (3.10.7)$$

以及数据密度函数满足

$$p(Y^{\text{data}}) \approx \frac{p(Y^{\text{data}} | \theta^*) p(\theta^*)}{\frac{1}{(2\pi)^{\frac{T}{2}}} |V_{\text{mode}}^{-1}|^{\frac{1}{2}}} \quad (3.10.8)$$

4. Dynare 贝叶斯估计的核心逻辑

综上所述, 在贝叶斯估计中, Dynare 首先使用 Kalman 滤波, 结合观测数据, 估计似然函数, 然后通过数值求解算法求出似然函数的最大值即众数, 最后使用 MH 算法 (MCMC) 模拟, 抽取马尔可夫链, 通过直方图近似逼近后验分布的密度, 这就是 Dynare 贝叶斯估计最核心的逻辑 (如图 3.12 所示)。

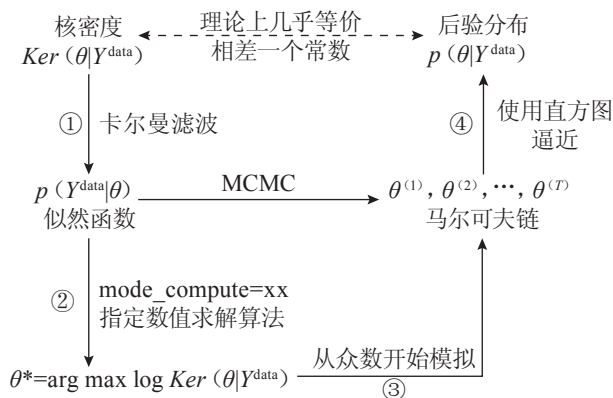


图 3.12 Dynare 贝叶斯估计的核心逻辑

① 由于篇幅限制, 此处没有提供推导的过程, 感兴趣的读者可向作者索要推导过程。

3.10.3 Dynare 参数估计和一个例子

由于篇幅的限制，本节并不打算介绍 Dynare 对参数估计所使用的求解后验众数的数值算法和 MH 算法背后的各种细节，而是着重介绍如何使用 Dynare 内置的估计命令，对一个或多个参数进行编程估计。

1. Dynare 参数估计的基本逻辑

Dynare 参数估计是使用模型 (3.10.9) 的一阶线性化系统进行估计^①：

$$E_t \{ f(y_{t+1}, y_t, y_{t-1}, u_t; \theta) \} = 0 \quad (3.10.9)$$

其中， θ 为待估参数向量； u_t 为外生的冲击； y_t 为内生变量组成的向量。

(1) Dynare 参数估计的基本逻辑

- 第一，计算内生变量 y_t 的稳态值 y ；
- 第二，线性化系统均衡条件；
- 第三，求解线性化模型；
- 第四，使用 Kalman 滤波算法计算对数似然函数 (Log-Likelihood)；
- 第五，找到极大似然值和后验分布的众数 (Mode)；
- 第六，使用 Metropolis-Hastings 算法模拟后验分布。

(2) Dynare 参数估计的语法

Dynare 参数估计的基本语法包括 3 部分：首先是使用内置关键字以声明参数估计模块 (block)；然后是待估参数本身的声明；最后是使用 estimation 命令，完成语法设置。接下来一一介绍。

Dynare 中声明待估参数模块的内置命令是以 “estimated_params;” 开头，以 “end;” 结尾，中间输入待估参数声明。待估参数有两种常见的类型：一种是结构参数，使用 “方法一” 声明，一种是外生冲击的标准差参数，使用 “方法二” 声明，具体如下所示。

```
estimated_params;
…… (加入相应的语法声明，见方法一和方法二)
end;
```

方法一：

```
PARAMETER NAME [, INITIAL_VALUE [, LOWER_BOUND, UPPER_BOUND]], PRIOR_SHAPE,
PRIOR_MEAN, PRIOR_STANDARD_ERROR [, PRIOR_3RD_PARAMETER [,
PRIOR_4TH_PARAMETER [, SCALE_PARAMETER ] ] ];
```

^① 这里的线性化方法包括对数线性化，但需要在 estimation 命令中指定选项 loglinear，才能使用对数线性化，具体参考 Dynare Reference Manual。

方法二:

```
stderr VARIABLE_NAME [, INITIAL_VALUE [, LOWER_BOUND, UPPER_BOUND]],
PRIOR_SHAPE,
PRIOR_MEAN, PRIOR_STANDARD_ERROR [, PRIOR_3RD_PARAMETER [,
PRIOR_4TH_PARAMETER [, SCALE_PARAMETER ] ] ];
```

实例如下:

```
estimated_params ;
gamma, normal_pdf, 1, 0.05;
alpha, uniform_pdf, , , 0, 1;
end;
estimated_params ;
stderr e, inv_gamma_pdf, 0.01, inf;
end;
```

初始值和上下界的其他指定方式

```
estimated_params_init; (初始值)          estimated_params_bounds; (上下界)
stderr eps_a, 0.015;                      stderr eps_a, 0.001, .2;
rho, 0.89;                                 rho, .001, .95;
end;                                        end;
```

注: 此处仅列示了 Bayesian 估计下的简单、常用的命令, MLE 估计的声明方式略有不同, 即不需要指定 Prior Shape, 只需要参数的初始值、上下限 (lower & upper bound); SCALE_PARAMETER 用于设定跳跃分布的方差与协方差矩阵的调节系数, 还可以通过 estimation 命令的选项 mh_jscale 设定该条件系数, 更为详细的使用方法请参考 Dynare Reference Manual。

结构参数的声明以该参数的名称开头, 然后是该参数服从的先验分布的名称 (Prior Shape), 该名称是由 Dynare 预先定义的常见概率分布, 如表 3.11 所示。

表 3.11 Dynare 预先定义的 Prior Shape

先验分布名称	符号形式	分布取值范围
beta_pdf	$B(\mu, \sigma, p_3, p_4)$	$[p_3, p_4], p_3, p_4 > 0$
gamma_pdf	$\Gamma(\mu, \sigma, p_3)$	$(p_3, \infty), p_3 > 0$
inv_gamma_pdf	$I\Gamma(\mu, \sigma)$	$(0, \infty)$
normal_pdf	$N(\mu, \sigma)$	$(-\infty, \infty)$
uniform_pdf	$U(p_3, p_4)$	(p_3, p_4)

数据来源: Dynarev4 User Guide, P.48。

表格 3.11 中参数 μ 、 σ 分别表示分布的均值和标准差; 参数 p_3 、 p_4 仅用于 Beta 分布、Gamma 分布和 Uniform 分布, 可以为无穷大。此外 Uniform 分布因为不需要声明均值和

标准差，因此需要在参数声明中留出两个逗号隔开的空格，然后声明 p_3 、 p_4 参数。对于标准差参数的声明，需要使用关键字 “stderr” 后跟外生冲击的名称，而不是外生冲击对应的标准差的名称，然后是指定其服从的先验分布，其声明语法同方法一。

如果在 “estimated_params;” 声明中没有给定参数的初始值、上下界或已给定，可以在 “estimated_params_init;” 或 “estimated_params_bounds;” 中另外单独指定或重新给定，以覆盖前面指定的值（参见上述实例）。

对于先验分布的选择，的确是非常困难的。在没有足够信息的情况下，选择先验分布往往具有较强的主观性，这也成为贝叶斯估计被质疑和批评最多的地方。但文献中有如下几条不成文的共识，可供参考：

- 对于标准差参数 σ 的估计，通常选择 Inverse Gamma 分布；
- 对于 AR(1) 过程的滞后项系数 ρ ，即 Persistence 参数，通常选择 Beta 分布；
- 其他参数，大都选择正态 (Normal) 分布。

在进行估计之前，还需要注意几个问题。

第一，数据问题。 参数估计的前提是模型中某些内生变量是可以观测的或者由观测值经过处理得到。使用数据首先需要确定选择哪些数据的问题。比如模型中有产出、消费、投资、通胀、就业等多个可观测的内生变量，但模型中只有 3 个外生冲击，由于不确定性问题 (Indeterminacy Problem) 的存在^①，从而只能选择不超过 3 个观测数据项，作为数据输入。选定数据项后，需要选择适当的时间序列长度和时间频率。一般说来，时间频率的选择可根据家庭贴现因子 β 的大小来选择，这是因为名义利率的稳态值 $R = \pi/\beta - 1$ ，如果稳态的通胀 $\pi = 1$ ， $\beta = 0.99$ 时，名义利率的稳态为 1%，年化名义利率为 4%，因此选择季度为数据频率较为合理。当 $\beta = 0.95$ 时，名义利率的稳态值为 5%，因此可选择年度作为频率。在时间频率选定后，就是序列长度问题。从直觉上讲，时间序列越长越好，因为其包含的有用信息越多，估计的准确性越高^②。

第二，观测数据与内生变量的匹配问题。 在估计时，要求模型文件中的变量形式应和观测数据相匹配。模型文件中的内生变量可能为水平变量、对数水平变量，也可以为增长率形式。比如对于产出 Y_t ，如果模型文件中为对数水平变量即 $y_t = \log Y_t$ ，则观测数据也需要取对数。若产出为增长率变量即 $dy_t = \log Y_t - \log Y_{t-1}$ ，那么数据也要做相应的处理。

① 不确定性问题是指观测变量的个数不能超过模型中外生冲击 (shock) 的个数，否则无法计算模型的似然函数 likelihood，从而无法估计和求解。观测变量个数少于或等于外生冲击的个数是能使用 Kalman filter 的充分条件。不确定性问题也称之为随机奇异问题 (Stochastic Singularity)。

② 其前提条件是待估参数本身是相对稳定的或随时间不发生较大变化。

第三, 观测变量的声明问题。内生变量如果声明为可观测变量, 需要使用如下的命令:

```
命令
varobs VARIABLE_NAME...;
例如
varobs pi tau a; //声明三个内生变量为观测变量
```

Dynare 会在 estimation 命令中调取外部数据 (datafile 选项指定外部数据文件名), 并加载到内存中去。观测变量声明命令 varobs 会告诉 Dynare 去内存中寻找变量名为 pi、tau 和 a 的 3 个序列, 如果找不到, 则会提示相应错误。

第四, Estimation 命令及其选项问题。估计命令 estimation 是 Dynare 中最重要也最复杂的命令之一^①。此处仅仅选取几个关键的选项加以介绍。

在参数估计之后, Dynare 将使用后验分布的均值 (Posterior Mean) 作为参数的估计值, 分别存放于 M_params 及 M_Sigma_e (外生冲击的方差与协方差矩阵) 中。然后使用这些值继续求解模型^②。

estimation 常用的选项 (options) 如下 (第 8 项开始为 Bayesian 估计独有选项)。

① **datafile = xx**, 可以是一个 Matlab 的 m 文件, mat 矩阵或外部 CSV 文件等。也可以使用 xls_sheet 选项加载 xls 文件等。此处推荐使用 m 文件, 这样可以对已有的数据文件, 如 mat 文件、csv 或 xls 文件先进行简单的自定义处理, 然后加载到当前模型文件中。在源代码 28 中, 使用了 Matlab 文件 load_Gali_est_data.m 文件加载数据。

② **conf_sig = xx**, 定义置信区间的宽度 (比如 xx = .95 定义 “95% 的置信区间”)。

③ **first_obs=xx**, 定义由 load_Gali_est_data.m 加载的数据, 第一个开始用于估计的数据位置, xx 必须是一个数字 (正整数), 不能为预先定义的变量^③, 缺省值为 1。

④ **nobs=yy**, Dynare 估计中使用的观测变量序列长度: t=xx to t=yy+xx-1, 同样的 yy 必须是一个数字 (正整数), 缺省值为全部观测变量。

⑤ **mode_check**, 画出后验分布众数 (posterior mode) 附近的后验分布密度图。

① 在 4.4.3 版本中, 估计命令 estimation 的选项 (options) 多达 75 项, Dynare Reference Manual 中也有多达 20 余页的相关解释。

② Dynare Reference Manual: After running estimation, the parameters M_params and the variance matrix M_Sigma_e of the shocks are set to the mode for maximum likelihood estimation or posterior mode computation without Metropolis iterations. After estimation with Metropolis iterations (option mh_replic > 0 or option load_mh_file set) the parameters M_params and the variance matrix M_Sigma_e of the shocks are set to the posterior mean. 因此极大似然估计使用众数, 而贝叶斯估计使用后验分布均值作为参数的估计值, 进而求解。

③ 如果使用预先定义的变量, 会产生错误: syntax error, unexpected NAME, expecting INT_NUMBER or '['。

⑥ **mode_compute=xx**, 选择优化算法 (Optimizer), 取值 0 ~ 10, 默认使用 4, 即 Chris Sims's csminwel 程序。(注意, 优化算法在不断的更新中)

⑦ **forecast=n**, 在观测变量最后一期之后进行预测 n 期值。

⑧ **smoother**, 计算内生变量和外生冲击的后验分布的相关矩, 即给定所有信息 (至最后一期), 计算期望值 $E_T y_t$ (Expected Values, Dynare 中称为 Smoothed Variables)。同时也计算另外一种期望值, 即给定所有至当期的信息, 计算期望值 $E_t y_t$ (Dynare 中称为 Updated Variables), 并存储到 oo_ 结构 (struct) 的相应位置。

⑨ **mh_replic=xx**, 定义计算后验分布时 MCMC 重复的次数, 缺省值为 2 000, 一般大于 1 200。

⑩ **mh_jscale=xx**, 定义 MCMC 中的乘法因子 (Scale Factor)。如果合适, 理想情况下, 乘法因子的选择应该使得接受率 “acceptation rate” 在 25% ~ 33% 附近。如果太高, 请增大乘法因子的值。该参数一般需要多次调试后获取比较合适的值, 很难一次获取理想的值, 缺省值为 0.2。此乘法因子还可以通过 “estimated_params;” 命令进行声明。

⑪ **mh_nblocks=xx**, 定义 Metropolis-Hastings 算法中平行 (Paralleled) 进行的马尔可夫链 (Markov Chain) 的个数, 缺省值为 2。链条数越多, 耗费资源越多。

⑫ **filtered_vars**, 计算滤波变量, 即一步前向预测, 是指 $E_{t-1} y_t, t=1, \dots, T$, 即提前一期预测 (One-step-ahead Forecast), 相关结果存储在 oo_FilteredVariables 中。

⑬ **moments_varendo**, 计算内生变量的后验分布的理论矩, 结果存储在结构 oo_PosteriorTheoreticalMoments 中。

2. Dynare 参数估计的例子

在 Dynare 中使用贝叶斯对参数进行估计的一个非常经典的例子就是 Smets & Wouters (2007, *AER*)。估计时, Smets & Wouters (2007) 首先对原有 DSGE 模型的均衡条件进行了线性化, 然后使用了 7 个宏观经济序列变量 (GDP、消费、投资、工资、劳动时间、GDP 平减通胀、联邦基金利率), 对 19 个结构参数和 17 个外生冲击 AR(1) 过程参数 (标准差和持续性参数) 进行了估计^①。由于该模型较为复杂, 不再详细讲解, 此处转而估计一个简单的模型。

为了更加清楚地看到估计结果的准确性, 此处不打算使用事实统计数据进行估计, 而是使用一个完全校准的线性化模型进行随机模拟, 将模拟的数据用于参数估计。因为

① 此处收集了该论文的数据和代码: \Sources\Chap3_Dynare_Basics\3.10_estimation\SW2007。包括论文技术附录、模型文件、数据文件、数据使用说明和其他技术细节等。

待估参数此时已经完全知晓，这样就能清楚看到 Dynare 估计结果的准确性和某些估计选项的适用性。

考虑如下一个简单的新凯恩斯模型，不含有投资但含有黏性价格和产出缺口，该模型主要来自于 Galí(2008)。此处为对数线性化后的均衡系统，其多数均衡方程都是文献中标准的结论。

(1) 模型

假设家庭的即期效用函数如下：

$$U(C_t, N_t) \equiv \frac{C_t^{1-\sigma} - 1}{1-\sigma} - \exp(\tau_t) \frac{N_t^{1+\varphi}}{1+\varphi} \quad (3.10.10)$$

其中， $\sigma > 0$ 为消费跨期替代弹性的倒数； τ_t 为劳动负效用的外生冲击，服从 AR(1) 过程： $\tau_t = \lambda \tau_{t-1} + \epsilon_t^\tau$ ， $\epsilon_t^\tau \sim \text{i.i.d}$ ； $\varphi > 0$ 为劳动供给的 Frisch 弹性的倒数。

首先是新凯恩斯 IS 曲线 (NKIS) 和新凯恩斯菲利普斯曲线 (NKPC)^①：

$$x_t = E_t(x_{t+1}) - \frac{1}{\sigma} (i_t - E_t(\pi_{t+1}) - i_t^*) \quad (3.10.11)$$

$$\pi_t = \beta E_t \pi_{t+1} + \kappa x_t \quad (3.10.12)$$

其中，NKIS 曲线由标准的消费跨期替代的 Euler 方程得到： $-\sigma c_t = i_t - \sigma E_t c_{t+1} - E_t \pi_{t+1}$ ^②，此处假设了所有产出用于消费，即没有投资。 x_t 为产出缺口，定义为 $x_t \equiv y_t - y_t^*$ ； y_t 为产出的对数； y_t^* 为自然水平产出的对数^③； i_t 为名义利率； i_t^* 为自然利率 (Natural Real Rate)； π_t 为通货膨胀； β 为贴现因子； $\kappa \equiv \frac{(1-\beta)(1-\beta\theta)}{\theta}$ ^④； θ 为标准的 Calvo 黏性价格定价参数， $\Theta \equiv \frac{1-\alpha}{1-\alpha+\alpha\epsilon}$ ； $1-\alpha$ 为产出的劳动份额，由中间品厂商的生产函数决定： $Y_{i,t} = A_t N_{i,t}^{1-\alpha}$ ； ϵ 为最终品生产的不同中间品之间的替代弹性。为了后续分析简单起见，假设 $\alpha=0$ ，即除技术外，产出由劳动完全确定。

① 本例子中各变量均为对数离差形式，即 log deviation form。为了简化处理，此处暗含了假设：净通胀稳态值为 0，即总通胀 (gross) 的稳态值为 1。因此，价格离散变量 [Price Dispersion，具体分析见本书“4.2.2 黏性价格设定与价格离散核 (Price Dispersion)”一节] 的稳态为 1，进而在稳态时自然产出和实际产出水平相同，因此产出缺口在稳态时为 0，在后续的产出缺口分析中不再显性说明。

② IS 曲线由标准的 Euler 方程减去自然水平下的 Euler 方程得到。

③ 自然产出水平，Natural Output，是不存在无效率（如没有价格黏性）时经济的产出水平。此外，有学者将自然产出认为是潜在产出 (Potential Output，有学者将其称为产能)。关于自然产出水平的解释，请参考本书“4.2.4 有效均衡、弹性价格均衡和实际均衡”一节。

其次是自然水平产出和自然利率方程：

$$y_t^* = \frac{1+\varphi}{\sigma+\varphi} a_t - \frac{1}{\sigma+\varphi} \tau_t \quad (3.10.13)$$

$$i_t^* = \sigma \frac{1+\varphi}{\sigma+\varphi} (\rho-1) a_t + \sigma \frac{1-\lambda}{\sigma+\varphi} \tau_t \quad (3.10.14)$$

技术冲击 A_t 的对数 a_t 服从 $a_t = \rho a_{t-1} + \epsilon_t^a, \epsilon_t^a \sim \text{i.i.d.}$ 。自然水平产出方程由加总的生产函数和家庭劳动供给方程共同确定^①。利率方程则由消费的 Euler 方程、自然水平产出方程和技术的 AR(1) 过程共同确定。此外，假设货币政策为 Taylor 规则如下：

$$i_t = \rho_i i_{t-1} + (1-\rho_i)(\phi_\pi \pi_t + \phi_x x_t) + \epsilon_t^i, \epsilon_t^i \sim \text{i.i.d.} \quad (3.10.15)$$

除滞后项外，名义利率对通胀和产出缺口做出反应。其中 $\rho_i, \phi_\pi, \phi_x > 0$ 。

综上所述，模型有 6 个内生变量： x_t 、 i_t 、 i_t^* 、 π_t 、 a_t 、 τ_t 和 6 个均衡条件：NKIS 曲线 (3.10.11)、NKPC 曲线 (3.10.12)、自然利率方程 (3.10.14)、Taylor 规则 (3.10.15)、技术和偏好冲击的 AR(1) 过程。所有参数的校准请参考前面的模型文件：源代码 28。

接下来，对上述模型文件进行随机模拟，模拟 5 000 个样本^②作为下一步估计的数据。然后，使用上述模拟的样本作为观测数据，对模型中的 3 个结构参数进行估计，如表 3.12 所示，并示例如何在 Dynare 中分别使用极大似然估计和贝叶斯估计此 3 个参数。虽然 3 个参数已经有真实值，并且样本数据是基于这 3 个真实值模拟出来的，因此反过来假设 3 个参数未知，使用数据估计参数值，能做到有的放矢，发现问题。比如，通过选取不同观测序列或序列长度对参数估计结果的影响。

表 3.12 3 个待估结构参数

待估参数	含 义
$\rho=0.8$	技术冲击 AR(1) 过程的持续参数
$\lambda=0.5$	劳动负效用的外生冲击 AR(1) 过程的持续参数
$\sigma_a=0.02$	技术冲击 AR(1) 过程中的外生冲击的标准差

注：此处将参数真实值列示出来，以和估计值对比。

(2) 最大似然估计 (MLE) 及 Dynare 实践

在源代码 28 中，使用了“estimated_params;”设定了初始值，使用“estimated_params_bounds;”设定了上下边界。此外使用“varobs”定义了 3 个观测变量。

① 在没有黏性价格设定下，加总的产出为 $Y_t=A_t N_t$ ，劳动供给方程为 $\exp(\tau_t) C_t^\sigma N_t^\sigma = A_t W_t / P_t$ 。

② 模拟文件地址：\Sources\Chap3_Dynare_Basics\3.10_estimation\Gali_sim.mod。具体请参考本小节第四部分关于 Dynare 程序文件的说明。

源代码28 极大似然(MLE)估计

```

// Maximum Likelihood estimation of the Galí-Christiano Model
// Gali_est_mle.mod, 默认设定
//we are going to estimate two persistence parameters, rho and lambda
// and one standard deviation parameter
//Written by Xiangyang Li@SCC

//declaration of the endogenous variables, log-deviation form
var a // technology
pi //CPI inflation
i //nominal rate
istar //nature real rate
tau //labor disutility shock
x //output gap = output - natural output
dy //growth rate of output
;
varexo eps_a eps_tau eps_i;

parameters sigma phi beta kappa phi_x phi_pi rhoi theta;
parameters rho lambda; //to be estimated

// Parameter Values
sigma = 1; //inverse of inter-temporal elasticity of substitution of
consumption
beta = 0.99; //discount factor of households
phi_x = .15; //coefficient of output gap in monetary policy rule
phi_pi = 1.5; //coefficient of inflation in monetary policy rule
rhoi = 0.8; //persistence of nominal interest rate;
//rho = 0.8; //persistence of technology, to be estimated
//lambda = 0.5; //persistence of labor disutility shock, to be
estimated;
phi = 1; //inverse of Frisch elasticity of labor supply
theta = 0.75; //Standard Calvo stickiness parameter
kappa = ((1-theta)*(1-beta*theta))/theta; //simplifying parameter

// The model in log-linear
model(linear);
//(1) Philips Curve - Calvo Pricing Equation
beta*pi(+1) + kappa*x = pi;

//(2) New Keynesian IS curve
sigma*(i - pi(+1)-istar) = x(+1) - x;

//(3) natural real rate definition
istar = sigma*(1+phi)*(rho-1)/(sigma+phi)*a + sigma*(1-lambda)/
(sigma+phi)*tau;

```

```

// (4) Taylor Rule
i = rhoi*i(-1) + (1-rhoi)*(phi_pi*pi + phi_x*x) + eps_i;

// (5) Technology shock
a = rho*a(-1) + eps_a;

// (6) Labor disutility shock
tau = lambda*tau(-1) + eps_tau;

// (7) Growth rate of output - by output gap
dy = x - x(-1) + (1+phi)/(sigma+phi)*(a-a(-1)) - (tau - tau(-1))/(sigma+phi);
end;

shocks;
var eps_tau;   stderr 0.01;
var eps_i;     stderr 0.01;
end;

// maximum likelihood estimation:
// initial values, pre-set to optimum
estimated_params;
    stderr eps_a, 0.01;
    rho, .80;
    lambda, .50;
end;

estimated_params_bounds;
stderr eps_a, 0.001, .2;
rho, .001, .95;
lambda, .001, .95;
end;

// Observables used in the estimation
varobs pi tau a;

estimation(datafile=load_Gali_est_data, conf_sig = .95, first_obs=101, nobs=4000,
mode_check, mode_compute=4) a pi i istar tau x dy;

```

在默认情况下 (观测数据选择通胀、技术和劳动负效用的外生冲击^①，序列长度为 4 000，算法选择 mode_compute=4)，极大似然估计结果令人满意，非常接近于真实值 (表 3.13 第一列)。但选择不同的观测数据，会影响参数的点估计，如表 3.13 所示。

① 假设观测数据合理。事实上，劳动负效用的外生冲击在实际中不可观测或很难观测到。

表 3.13 观测数据选择的影响

变 量		varobs pi tau a	varobs pi dy a
$\rho=0.8$	估计值	0.813 5	0.804 4
	标准差	0.009 2	0.002 8
$\lambda=0.5$	估计值	0.484 4	0.893 1
	标准差	0.006 9	0.004 4
$\sigma_a=0.02$	估计值	0.019 6	0.019 6
	标准差	0.000 2	0.000 2

数据来源：Dynare 估计结果，经由作者自行计算，其算法 mode_compute=4 和观测数据 nobs=4 000(第一列的设置是真实值^①)。

第二列设定了不同的观测数据，将劳动负效用的外生冲击的数据 τ_t 换成了产出的增长率 dy。此时，劳动负效用冲击的 Persistence 参数 λ 发生了很大变化。直观上讲，由于观测数据 τ_t 和参数 λ 密切相关，即数据 τ_t 中包含了参数 λ 最准确的信息。如果不使用数据 τ_t 势必会损失有用信息，而结果也恰恰说明了这点。因此，在选择观测数据时，要尽量选择 and 待估参数关系密切的数据序列，包括尽可能多的数据项，以获取更多有用信息来准确估计参数。

同样，观测数据序列的长度也会影响估计。表 3.14 列示了序列长度分别是 4 000 和 40 这两种情况。可看出，当序列长度缩小到 100 倍时，标准差增大约 10 倍。除 λ 参数外，其他两个参数的估值变化很小，这说明极大似然估计有一定的稳健性。

表 3.14 序列长度选择的影响

变量 (真实值)		nobs=4 000	nobs=40
$\rho=0.8$	估计值	0.813 5	0.792 4
	标准差	0.009 2	0.094 8
$\lambda=0.5$	估计值	0.484 4	0.280 7
	标准差	0.006 9	0.090 0
$\sigma_a=0.02$	估计值	0.019 6	0.018 4
	标准差	0.000 2	0.002 1

数据来源：Dynare 估计结果，经由作者自行计算，其算法 mode_compute = 4 和观测数据 varobs pi tau a。

① 在极大似然估计中，默认情况下，优化方法的选择并未对结果产生影响。我们尝试了 mode_compute = 4、7、10 等，发现估计结果没有发生变化。但这并不意味着优化方法的选择不会对所有情况产生影响。

(3) 贝叶斯估计及 Dynare 实践

在贝叶斯估计 (源代码 29) 中, 指定先验分布时, 设定技术标准差参数的均值为 0.02, 和真值相同, 标准差为 10; 设定 ρ 、 λ 两个参数的均值为其真值, 标准差都为 0.04; 此外还设定了 3 个参数的上下界和初始值。在默认情况下^①, 从图 3.13 和表 3.15 中可看出, 贝叶斯估计的结果同样令人满意。

源代码29 贝叶斯(Bayesian)估计

```
// Bayesian estimation of the Gali-Christiano Model
// Gali_est_Bayesian.mod : 默认设定
//we are going to estimate two persistence parameters, rho and lambda
// and one standard deviation parameter
//Written by Xiangyang LiaSCC

..... (此处省略, 代码同最大似然估计中的源代码28)
// Set up the priors on the parameters
// Bayesian estimation:
estimated_params;
    stderr eps_a, inv_gamma_pdf,0.02,10;
    rho, beta_pdf, 0.80, .04;
    lambda, beta_pdf, 0.50, .04;
end;
estimated_params_bounds;
stderr eps_a, 0.001, .2;
rho, .001,.95;
lambda, .001,.95;
end;

// the initial value block is optional, pre-set to the optimum.
estimated_params_init;
stderr eps_a, 0.019;
rho, 0.82;
lambda, 0.51;
end;

// Observables used in the estimation
varobs pi tau a;

// Bayesian estimation
estimation(datafile=load_Gali_est_data,conf_sig=.95, first_
obs=101,forecast=40,nobs=40, mode_check, mode_compute=4, mh_
replic=1200, mh_jscale=1.4,mh_nblocks=2) a pi i istar tau x dy;
```

① 默认设定: mode_compute=4; mc_replic=1 200; mc_nblocks=2; nobs=40; varobs pi tau a。

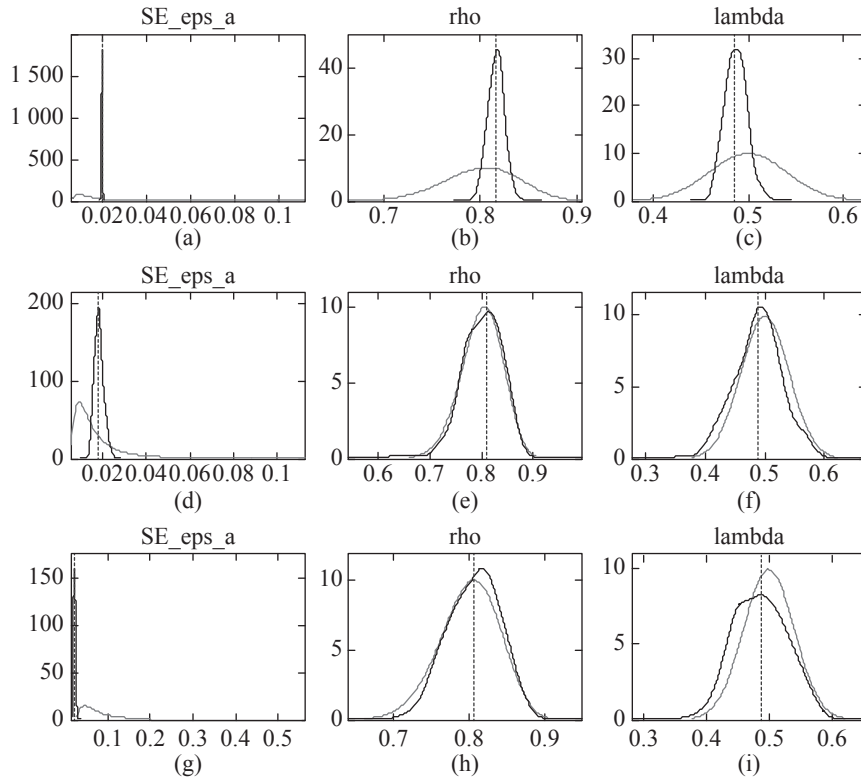


图 3.13 贝叶斯估计下的先验 (灰线) 和后验分布图 (黑线)

第一行: nobs=4 000; 第二行: nobs=40; 第三行: nobs=40, stderr eps_a, inv_gamma_pdf,0.1,1。

表 3.15 序列长度选择的影响

变量 (真实值)		nobs=4 000	nobs=40
$\rho=0.8$	估计值	0.816 2	0.801 0
	标准差	0.040 0	0.040 0
$\lambda=0.5$	估计值	0.487 1	0.488 1
	标准差	0.040 0	0.040 0
$\sigma_a=0.02$	估计值	0.019 6	0.018 3
	标准差	10.000 0	10.000 0

数据来源: Dynare 估计结果, 经由作者自行计算; 除序列长度外, 其他选项为默认设定。

图 37 则给出了 3 种不同情况下的先验分布和后验分布的情况。黑线表示后验分布,

灰线表示先验分布。

第一行和第二行表示在默认设定下，两个不同序列长度下的估计结果，两者相差 100 倍。在序列长度为 4 000 时，后验分布则呈现尖峰状 (Spike)，而且比先验分布更加紧凑。在序列长度为 40 时，除标准差参数外，后验和先验分布的较为接近。由于先验分布的假设和真实值相同 (源代码 29)，可以看出后验分布和先验分布差异并不大，也就是说数据中的信息和先验分布的假设差异不大。**有一个不成文的 (事后判断) 经验法则就是先验分布和后验分布的均值不能差异太大。**从图形上讲，两者在水平方向上不能偏离太远。偏离太远，说明先验分布的假设离数据中包含的信息差异太远；靠得太近或重合，说明数据中的信息没有被利用上，或者说先验分布中已经包含了数据中的大部分信息。

如果假设参数的均值和真实值不同，会有什么情况发生呢？为此，假设标准差参数 σ_a 不再取真实值 0.02，而是取值 0.1，是真值的 5 倍，标准差从 10 变为 1，缩小 10 倍，其他参数假设不变。第三行就是此假设下的估计结果。可以看出，标准差后验分布则严重偏离先验分布，向真实值靠拢 (左偏离，第三行第一个图)。这说明，数据中的确包含了关于真实值的信息，而不是完全依靠先验假设。

和极大似然估计相比，序列长度和观测数据的选择似乎对参数点估计的影响不大。表格 3.15 显示了序列长度为 4 000 和 40 时两种情况，结果相差不大。序列长度为 4 000 时，标准差参数 σ_a 的点估计有所提高，其他两个参数点估计的结果甚至差于序列长度为 40 时的估计结果。从表 3.16 中看到，观测数据选择影响不大，选取 pi tau a 这 3 个观测数据比 pi dy a 这 3 个观测数据结果略有改善，但改善不大。

相比极大似然估计，贝叶斯估计似乎更加稳健，因为其受到序列长度和观测数据的选择影响较少 (如表 3.15 和表 3.16 所示)。

表 3.16 观测数据选择的影响

变 量		varobs pi tau a	varobs pi dy a
$\rho=0.8$	估计值	0.801 0	0.806 8
	标准差	0.040 0	0.040 0
$\lambda=0.5$	估计值	0.488 1	0.487 0
	标准差	0.040 0	0.040 0
$\sigma_a=0.02$	估计值	0.018 3	0.018 1
	标准差	10.000 0	10.000 0

数据来源：Dynare 估计结果，经由作者自行计算；除观测变量定义外，其他选项为默认设定。

选项 `mh_jscale` 取值默认为 0.2, 但该默认值很少能够满足要求, 因为该参数的取值要使得 MCMC 算法的接受率在 25% ~ 33% 附近。也就是说需要不断尝试才能取得合适的值。根据接受率的合意取值, 从表 3.17 可看出, 本例在两条马尔可夫链上, 取值 1.2 都导致接受率较高, 高于 33%; 而取值 1.8, 接受率都低于 25%。因此, 比较理想的乘法因子的取值介于 1.2 和 1.8 之间, 经过反复试验, 发现取值 1.4 较为合适。

表 3.17 乘法因子的选择 (scale factor, `mh_jscale`) 与接受率 (acceptance ratio)

接受率 / 后验均值	<code>mh_jscale</code> =1.2	<code>mh_jscale</code> =1.4	<code>mh_jscale</code> =1.8
Chain 1: 接受率	36.681 7%	30.784 6%	22.188 9%
Chain 2: 接受率	36.082%	29.785 1%	23.538 2%
运行耗时	6min22s	5min57s	6min38s

数据来源: Dynare 估计结果, 经由作者自行计算; 除乘法因子外, `nobs`=3 000, 其他为默认设定。

至此, 基本完成了 Dynare 中关于参数估计的简单介绍。此外, 为了更加清楚地理解 Dynare 中贝叶斯估计的各种输出结果, 在接下来的部分, 以产出缺口为例来对此加以说明。

3. 产出缺口 (Output Gap) 分析

通俗来讲, 产出缺口是衡量实际产出和自然产出 (或潜在产出) 之间差异的一个经济指标。然而产出缺口的准确计算比较困难。自然产出水平是经济在其效率最高时的产出水平。然而现实中的自然产出水平几乎是无法观测的, 其效率最高, 无法量化, 而且经济增长受到的制约因素太多, 比如制度约束、市场因素等。在宏观经济模型构建时, 一般认为自然产出水平是模型中不存在无效率设定, 如不存在黏性价格等设定时的产出水平。

本部分利用 Dynare 在贝叶斯估计后的输出关于后验分布的相关结果, 对产出缺口做一个简单的分析^①。作为贝叶斯参数估计的副产品, Dynare 会给出内生变量的各种估计和预测, 包括平滑估计 (Smoothed)、滤波估计 (Filtered)、实时估计 (Updated)、预测 (Forecast) 等。本部分将结合 HP 滤波对产出缺口进行比较分析^②。

(1) 基本概念

在实际的计算中, 常使用 HP 滤波对 GDP 序列 $\{Y_t\}$ 进行处理, 以获取产出缺口 Gap_t ,

① 很多书籍对产出缺口的介绍非常有限。本书试图从简单的新凯恩斯模型入手, 通过利用 Dynare 的输出, 进行简单的图示分析, 但并未进行深入探讨。

② 关于 HP 滤波的详细逻辑, 请参考前述本书“1.3.4 HP 滤波的基本逻辑”一节分析。

其计算公式如下：

$$Y_t = Y_t^T + Y_t^C \quad (3.10.16)$$

$$Gap_t \equiv \frac{Y_t - Y_t^T}{Y_t^T} \times 100 = \frac{Y_t^C}{Y_t^T} \times 100 \quad (3.10.17)$$

其中， Y_t^T 为趋势项， Y_t^C 为波动项。公式 (3.10.17) 为产出缺口的实际计算公式，为标准化后的产出缺口，不妨称作相对产出缺口；但有时也将波动项 Y_t^C 直接解释为产出缺口，而无须将其标准化，此时不妨称作绝对产出缺口。在本小节前述的模型中，产出缺口的定义为

$$x_t \equiv y_t - y_t^* = \log \frac{Y_t}{Y} - \log \frac{Y_t^*}{Y} = \log \frac{Y_t}{Y_t^*} \approx \frac{Y_t - Y_t^*}{Y_t^*} \quad (3.10.18)$$

其中， Y_t^* 为自然产出水平变量。

在模型线性化时，前文已有阐述，即假定通胀的稳态为 0，此时容易推导出实际产出水平和自然产出水平的稳态值相同， $Y = Y^*$ ，因此 (3.10.18) 和 (3.10.17) 具有相同的形式^①，但这并不意味着能将自然产出水平等同于趋势项： $Y_t^* = Y_t^T$ ，实际上两者往往在数量上存在一定的差异，如图 3.14 所示。

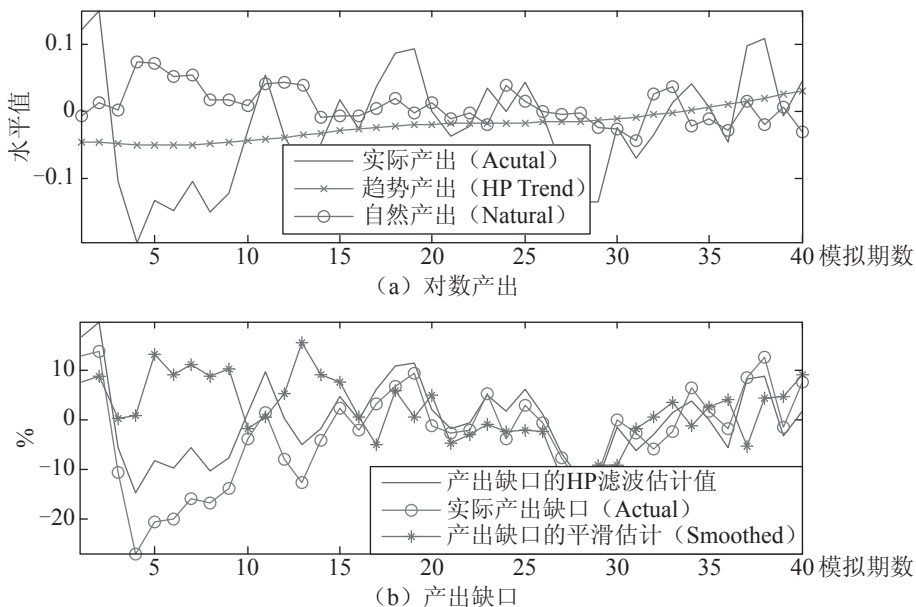


图 3.14 (对数) 产出和产出缺口

① 在后续的图示中，已将模型中的产出缺口 x_t 转换成了百分点 (%) 表示。

然后，考虑 Dynare 进行贝叶斯估计后，关于内生变量的各种估计和预测值。

- **滤波变量(Filtered Variables)**。滤波估计从本质上讲是计算期望值，具体说来是指 $E_{t-1}y_t$, $t = 1, \dots, T$ ，即提前一期预测(One-step-ahead Forecast)。当 $t = 1$ 时，对应第一个观测变量，此时的预测可获得的信息集是空集，即没有可获得的数据，因此可看作该序列的无条件均值 $E_0 y_1$ 。当 $t = T$ 时对应最后一个观测变量。当 $t = T + 1$ 时，此时为样本外一期预测值， $E_T y_{T+1}$ 。Dynare 将滤波结果存储在 oo_.FilteredVariables 中。
- **平滑变量(Smoothed Variables)**。平滑估计和滤波估计一样，都是计算期望值，只不过计算方法或期望基于的信息集不一样，具体可参考估计命令 estimation 选项的解释部分。如果在估计命令 estimation 后加入选项 smoother，则会计算各内生变量的平滑变量。结果将存储在 oo_.SmoothedVariables 中。
- **实时变量(Updated Variables)**。同样的，实时变量的计算，其实质也是计算期望值，具体可参考估计命令 estimation 选项的解释部分。实时变量的计算也依赖于选项 smoother，即 smoother 选项同时会计算平滑和实时变量。结果将存储在 oo_.UpdatedVariables 中。
- **预测变量(Forecast)**。直观来说，预测是样本外预测。Dynare 计算预测会根据估计 estimation 的选项不同而不同。如果选项没有使用 MH 算法，则基于后验分布的众数进行预测，结果存储于 oo_.forecast 结构中；否则依据 MH 模拟结果，进行点预测(Point Forecast，考虑到参数和外生冲击的不确定性)和均值预测(Mean Forecast，仅仅考虑到参数的不确定性，外生冲击被平抑，Averaged Out)，结果分别存储于 oo_.PointForecast，oo_.MeanForecast 中。

(2) 比较分析

此部分所画的图形中，产出缺口的各种估计使用了 Dynare 输出的中位数 (Median) 序列^①。为了节省篇幅，不再列示相关的 Matlab 源代码，具体请参考本书提供的相关 m 文件^②。

由于使用的是模拟数据，因此产出序列是对数水平变量^③，进而 HP 滤波估计时使用了对数水平变量而非水平变量。图 3.14 显示了对数产出及产出缺口。第一行为产出图，

① Dynare 会对各内生变量的各种估计计算中位数 (median)、均值 (mean)、方差 (var)、分位数 (deciles) 和置信区间的上下界 HPDinf 与 HPDsup (如果估计选项中设置了 conf_sig)。

② 地址：\Sources\Chap3_Dynare_Basics\3.10_estimation\output_gap_analysis.m。

③ Dynare 源文件中的产出变量 y_t 是对数离差形式，需要将其转换为对数水平变量然后进行 HP 滤波处理，即将产出变量稳态对数加回对数离差序列中，就能将其转换为对数水平变量。具体请参考源文件。

实线表示模拟的实际产出，带有 x 的标识线表示 HP 滤波后的趋势项，带有 o 标识的是自然产出^①。可见，自然产出水平和 HP 趋势项存在较大差异，趋势项则呈现出缓慢增长的态势。此外，自然产出水平虽然和实际产出水平存在数量差异，但两者的增减趋势大致相同。

图 3.14 第二行显示了 3 种不同的产出缺口的估计，分别是实线表示的 HP 滤波估计值 [由 (3.10.17) 定义]、带 o 表示的实际值 (x_t) 和带 * 表示的平滑值 (x_t 的平滑值)。可看出，三者虽然存在一定的数量差异，但增减趋势大致相同，但 HP 滤波估计值更贴近真实值。

图 3.15 中画出了产出缺口的滤波值 (Filtered, 非 HP 滤波) 和实际产出缺口，以及两者的差异。可以发现，滤波值与实际值差异较大。

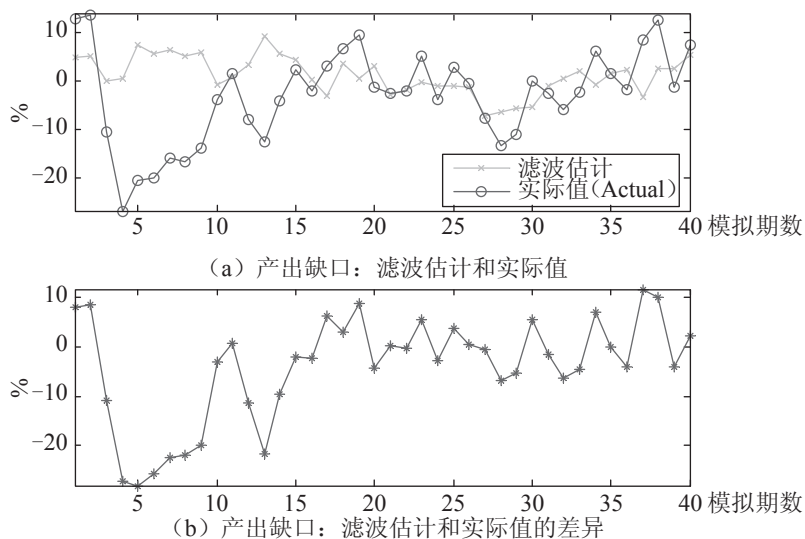


图 3.15 滤波后的产出缺口

图 3.16 关于产出缺口的实际值 (x_t)、实时变量和点预测。此处点预测为 40 期的样本外预测。产出缺口的点预测较为平滑，和实际值与实时变量的差异较大。表 3.18 列示了各种产出缺口的估计值与真实值之间差异的描述性统计，包括差异的均值和标准差 (误)。可以看出，各估计值的确和真值之间存在不小的差异，其中 HP 滤波与真实值之间差异的绝对值平均值最小，标准差 (误) 最小。

① 自然产出水平为模拟值。

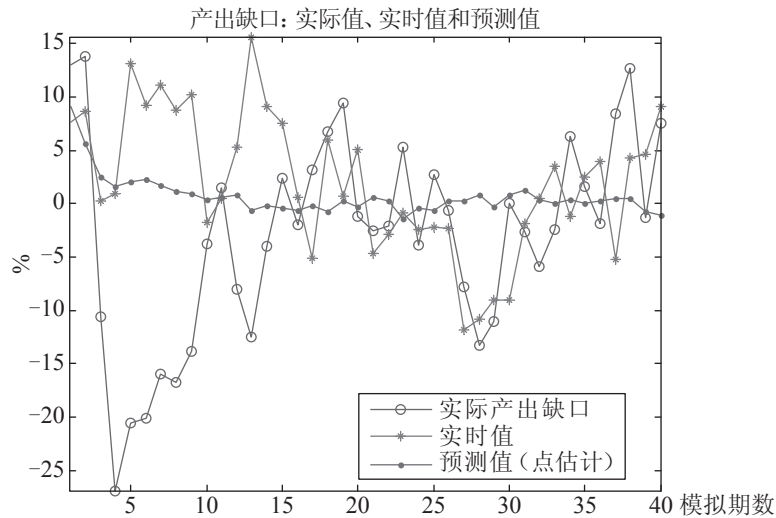


图 3.16 实时产出缺口和预测产出缺口

表 3.18 各种产出缺口估计与真实值 (Actual) 之间差异的统计描述

统计量 \ 变量	平滑变量	实时变量	滤波变量	HP 滤波
差异的标准差	0.123 7	0.123 7	0.108 1	0.044 5
差异均值	-0.048 5	-0.048 5	-0.041 1	-0.029 2

数据来源: Dynare 贝叶斯估计结果和作者自行计算; 中位数 median 和均值 mean 的统计量相同。具体计算源代码请参考本节的 m 文件。

4. Dynare 程序说明

(1) 观测数据模拟文件——Gali_sim.mod

Gali_sim.mod 用于模拟本节中的模型, 并保存模拟的数据, 作为可观测数据用于后续的估计。因此在极大似然和贝叶斯估计之前, **请先运行此文件**, 以获取 data.mat。其最核心的代码如下:

```
... (此处省略, 请参考源代码)
set_dynare_seed=1; //set a fixed seed for random number generators for replications.
stoch_simul(periods=5000, irf=7, nograph); //stochastic simulation
gcsim = oo_.endo_simul; //create a variable gcsim to store the data
save data gcsim; //save gcsim into data.mat file.
```

(2) 其他文件说明

观测数据加载文件——`load_Gali_est_data.m`，以加载上述模拟文件保存的 `data.mat`。该观测数据文件在 `estimation` 命令的选项中被调用，具体可参考源代码 28 和源代码 29。

```
load_Gali_est_data.m

load data; //load data.mat file saved by Gali_sim.mod

datatobeloaded = {'a' 'pi' 'i' 'istar' 'tau' 'x' 'dy'};

for i=1:length(datatobeloaded)
    eval(strcat(datatobeloaded{i}, '=', 'gcsim(', num2str(i), ',, :)' ));
end
```

此处使用了两个 Matlab 的内置函数：字符串估计函数 `eval` 和字符串连接函数 `strcat`。第一个 `eval` 函数是一个非常有用的 Matlab 函数，在动态编程中经常用到，已经多次提及。

- **Gali_est_mle.mod**: 最大似然估计模型文件。
- **Gali_est_Bayesian.mod**: 贝叶斯估计的模型文件。

以上文件都位于 `\Sources\Chap3_Dynare_Basics\3.10_estimation` 目录下。

参考文献

- [1] An S., Schorfheide F. *Bayesian Analysis of DSGE Models*[J]. *Econometric Reviews*,2007, 26(2-4):113-172.
- [2] Andreasen M. M., Fernández-Villaverde Jesús., Rubio-Ramírez J. F. *The Pruned State-Space System for Non-Linear DSGE Models: Theory and Empirical Applications*[J]. NBER Working Papers 18983,2013.
- [3] Cooley T. F., Prescott E. C. *Economic Growth and Business Cycles*[M].Cooley T. F. *Frontiers of Business Cycle Research*. Princeton: Princeton University Press, 1995:1-38.
- [4] DeGroot M. H. *Optimal Statistical Decisions*[M]. New York: McGraw-Hill,1970.
- [5] Hamilton J. D. *Time Series Analysis*[M]. Princeton, NJ: Princeton University Press: 799,1994.
- [6] Herbst E. P., Schorfheide F. *Bayesian Estimation of DSGE Models*[M]. Princeton University Press,2016.

- [7] Ireland P. N. *A Method for Taking Models to the Data*[J]. *Journal of Economic Dynamics and Control*,2004,28(6):1205-1226.
- [8] Kim J., Kim S., Schaumburg E., Sims C. A. *Calculating and Using Second-Order Accurate Solutions of Discrete Time Dynamic Equilibrium Models*[J]. *Journal of Economic Dynamics and Control*,2008,32(11):3397-3414.
- [9] Rabanala P., Rubio-Ramírez J. F. *Comparing New Keynesian Models of the Business Cycle: A Bayesian Approach*[J]. *Journal of Monetary Economics*,2005,52(6):1151-1166.
- [10] Schorfheide F. *Loss Function-Based Evaluation of DSGE Models*[J]. *Journal of Applied Econometrics*,2000,15(6):645-670.
- [11] Smets F., Wouters R. *An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area*[J]. *Journal of the European Economic Association*,2003,1(5):1123-1175.
- [12] Smets F., Wouters R. *Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach*[J]. *The American Economic Review*,2007,97(3):586-606.

4

RBC 模型和 NK 模型

Kydland & Prescott(1982, *Econometrica*) 开创性地提出了真实经济周期模型 (Real Business Model, RBC)。RBC 模型不仅在理论上有所创新，而且在方法论上更有创新。

首先来看其在理论上的创新。其理论创新和贡献现在看来虽然有争议，但仍然值得回顾，具有启发性：第一，技术的趋势性进步是经济长期增长的主要推动力；从短期看，技术冲击是经济波动 (Aggregate Fluctuation) 的主要因素，回答了为什么技术冲击能够使得消费、投资和劳动产生顺周期特性的问题。第二，货币政策因素是不重要的，即货币中性。第三，弗里德曼规则 (Friedman Rule) 是最优的，即名义利率为 0。第四，经济波动 (Business Cycle) 是对供给冲击的反应，因而使我们重新认识到不仅需求冲击重要，而且供给冲击同样重要。第五，供给冲击的传播机制 (Propagation Mechanism) 解释了经济短期波动的内在机理，以及宏观统计变量与产出的顺周期特性。

其次来看其在方法论上的创新。具体表现为四点：第一，抛弃了以往简化形式的联立方程或结构方程的分析方法，从一般均衡的角度进行分析，即求解家庭和厂商最优化行为，设定目标函数和预算约束。第二，用理性预期理论来处理预期形成及其影响，而非适应性预期理论。第三，求解 RBC 模型时使用了以前未被使用的数值求解和模拟技术，并创新性地使用了一种特殊的参数估计方法即校准法来对参数赋值。第四，强调模型分析的数量特征以及和实际统计数据的对比与吻合程度，如参数校准、模拟以及经济变量之间的统计特征分析 (顺周期和逆周期)。这些方法论上的创新，得到了进一步发扬光大。如今 DSGE 模型的方法论几乎完全继承了 Kydland & Prescott(1982) 方法论的创新。

Prescott(2016) 对 RBC 方法论 (Methodology) 和新古典增长理论进行了系统综述，值得仔细阅读。在过去的几十年里，Kydland & Prescott(1982) 在理论上的贡献引起诸多质疑和批评，从而使得宏观经济分析模型的发展开始转向新凯恩斯模型 (New Keynesian, NK)。NK 模型区别于 RBC 模型的两个明显的、重要的特征：模型是否包含垄断竞争和名义刚性或黏性 (Rigidity or Stickiness)。名义刚性 (或黏性) 是指名义变量，比如价格或

工资在短期内不易发生变化，具有一定的黏性或刚性。名义刚性使得短期内货币政策不再无效，反而会发挥重要作用。本章将通过简单的 NK 模型来说明垄断竞争、名义刚性的作用及其对应的不同均衡^①以及货币政策的非中性问题等。

4.1 RBC 模型及其拓展

4.1.1 RBC模型与新古典增长模型

作为 RBC 模型的基本框架，新古典增长模型 (Neoclassical Growth Model) 在 Kydland & Prescott(1982) 以及后来的研究中起到了重要的作用。因此，在介绍 RBC 模型之前，首先来介绍新古典增长模型的一些基本常识；然后以 Kydland & Prescott(1982) 的研究为出发点，介绍 RBC 模型。

经济周期现象 (Business Cycle Phenomena) 是指 GDP 围绕其长期变化趋势周期波动的现象，同时重要的宏观经济变量之间会呈现规律性的波动 (Co-movements)，比如 GDP 与消费、投资、就业以及生产率等。

1. 新古典增长模型

新古典增长模型，最早可追溯到 Slow(1956, *QJE*) 和 Swan(1956) 提出的 Slow-Swan 增长模型。Slow-Swan 增长模型通过分析资本积累、劳动或人口增长以及技术进步来解决经济的长期增长问题。该增长模型中采取了被后来广泛使用的生产函数即 Cobb-Douglas 生产函数，这使得 Slow-Swan 增长模型具备了和微观经济学连接的桥梁。由于其具备良好的数学性质 (一次齐次)，使得 Slow-Swan 增长模型成为进一步研究和拓展的基础，比如 Ramsey-Cass-Koopmans 增长模型，通过求解消费优化问题，使得储蓄率内生。也就是说，储蓄率可变，从而改变了 Slow-Swan 增长模型关于储蓄率为常值的基本假设。

接下来，介绍一个简单的新古典增长模型，及其相关的几个基本概念：相位图 (Phase Diagram) 和鞍点路径 (Saddle Point Path)。

经济中存在大量的具有无限生存期限 (Infinitely Lived) 的相同个体，个体通过消费获得效用，并储蓄以积累资本存量，模型均衡可通过求解如下无穷期贴现效用最大化问题：

^① 本章的写作受到了美国圣母大学 Eric Sims 教授、Timothy S. Fuerst 教授、Nelson C. Mark 教授的启发，并借鉴了他们的讲义。聆听他们的授课和教诲让我受益匪浅，在此一并表示感谢，当然文责自负。

$$\max_{C_t, K_{t+1}} E_0 \sum_{t=0}^{\infty} \beta^t \frac{C_t^{1-\sigma} - 1}{1-\sigma}$$

预算约束为

$$C_t + K_{t+1} - (1-\delta)K_t = A_t F(K_t, Z_t N_t) \quad (4.1.1)$$

其中, $A_t F(K_t, Z_t N_t)$ 为标准的厂商生产函数, 满足 $F_1, F_2 > 0$ (表示一阶偏导数) 即关于要素投入为增函数, $F_{11}, F_{22} < 0$ (表示二阶偏导数) 即关于要素投入具有边际产出递减性质; A_t 为中性技术进步 (Neutral Technological Progress), (对数) 服从平稳的随机过程; Z_t 为劳动增强型技术进步 (Labor-Augmenting Technological Progress), 为趋势平稳序列 (Trend Stationary), 满足:

$$Z_t = (1+z)^t Z_0 \quad (4.1.2)$$

其中, Z_0 为给定, z 为复合增长率。 N_t 表示人口数量, 同样假设其为趋势平稳序列:

$$N_t = (1+n)^t N_0 \quad (4.1.3)$$

其中, N_0 为给定, n 为人口复合增长率。很显然, 模型存在两个不平稳变量: Z_t 和 N_t 。因而在后续均衡条件的处理中需要消除不平稳因素。生产函数采取 Cobb-Douglas 形式:

$$F(K_t, Z_t N_t) \equiv K_t^\alpha (Z_t N_t)^{1-\alpha} \quad (4.1.4)$$

根据本书“1.1.3 B&K 方法”一节中的求解方法 (拉格朗日问题), 很容易获取上述效用最大化问题的一阶条件, 即 Euler 方程:

$$C_t^{-\sigma} = \beta E_t C_{t+1}^{-\sigma} (A_{t+1} \alpha K_{t+1}^{\alpha-1} (Z_{t+1} N_{t+1})^{1-\alpha} + 1 - \delta) \quad (4.1.5)$$

为了消除模型的不平稳性, 重新定义变量如下:

$$c_t \equiv \frac{C_t}{Z_t N_t}, k_t \equiv \frac{K_t}{Z_t N_t} \quad (4.1.6)$$

因而式 (4.1.5) 可重新写为

$$c_t^{-\sigma} = \beta \gamma^{-\sigma} E_t c_{t+1}^{-\sigma} (A_{t+1} \alpha k_{t+1}^{\alpha-1} + 1 - \delta) \quad (4.1.7)$$

其中, $\gamma \equiv (1+z)(1+n) \geq 1$, 此时可行性条件可写为

$$\gamma k_{t+1} - (1-\delta)k_t + c_t = A_t k_t^\alpha \quad (4.1.8)$$

模型均衡由式 (4.1.7) 和式 (4.1.8) 共同组成。和本书“1.1.3 B&K 方法”一节中的均衡条件相比, 此处的均衡条件仅仅多出了一个参数 γ , 从数值求解和均衡分析来看, 其影响并不大, 这也是为什么常常考虑平稳模型的原因所在, 不平稳模型可以通过去趋势的办法消除不平稳性。因此, 通常情况下该参数被忽略, 仅仅考虑平稳的情况。

接下来, 为了简化分析, 仅仅考虑 $\gamma=1$ 。新古典模型一个典型的定性求解分析工具