

宏观经济学计算方法精要

Essential Macroeconomic Computational Methods

汤昕^①

武汉大学经济与管理学院
世界经济系助理教授

2016年4月19日

传播前烦请征得作者允许。
Please Do Not Distribute without Permission.

^①作者文责自负。请将对本讲义的任何建议发送至: zjutangxin@gmail.com, 作者不胜感激。

致所有在我的*Ph.D*阶段帮助过我的人

前言

这份讲义是为配合作者2016年春夏学期在武汉大学讲授的研究生课程《世界经济统计》而准备的。我们从长期增长、短期经济波动以及收入分配三个角度，学习经典宏观经济理论中对这些现象进行解释的模型的具体求解。

现代宏观经济学以及在其中得到广泛应用的动态优化理论的发展，对经济数据的解释提出了更高的要求。传统的基于时间序列的简化式统计分析已经越发无法满足学者们在挖掘统计数据背后的经济理论方面的需求。始于Kydland and Prescott (1982)和Long and Plosser (1983)的动态宏观经济数量分析极大的拓展了经济理论的适用范围，学者们不需要在理论的丰富程度和解析性质之间进行两难取舍（当然这并不是说数量分析可以求解的问题是没有边界的），从而可以以相对直观的方法对经济理论进行构建，并更加严格的在定量的意义上对构建的理论进行检验。分析方法在这个方向上的发展也对传统经济学在分析工具上的训练提出了新的要求，以求解解析解为目的的抽象推导在数量分析当中相当程度上让位于以求解数值解为目的的数值方法。很多在研究生阶段接受过第一年基于解析推导的宏观经济学训练的同学，在阅读前沿文献的时候不可避免的会感受到方法论上的差异，进一步的，方法论上的差异又使得学生在进行自己的研究的时候感到难于上手。本讲义的目的，便在于给熟悉基本的研究生水平的高级宏观经济理论、但是在经济理论的数值分析方面基础相对薄弱的同学提供一份入门的参考资源。

动态宏观经济理论已经发展了几十年，市面上优秀的参考资料不说是汗牛充栋，也可以用琳琅满目来形容，这里作者略举几例，比如：被誉为宏观经济学圣经的《经济动态中的递归方法》[Recursive Methods in Economic Dynamics, Stokey and Lucas (1989)]，以收录题材广泛而著称的《递归宏观经济理论》[Recursive Macroeconomic Theory, Ljungqvist and Sargent (2012)]，Adda and Cooper (2003)，以及关注于技术的Heer and Maußner (2005)等等。可能有的读者此时要问了，既然我们已经有了如此之多的教材，我们为什么还需要一份新的讲义呢？这里容我引用C++大师Stanley B. Lippman在其著作《Essential C++》的前言中评价自己的另一本鸿篇巨著《C++ Primer》时所说的话（中译本节选一并附上）：

“Eventually, I realized sadly that the third edition of C++ Primer could no longer fill a similar role for someone needing to learn C++. It had just become too big.”

“我终于在感伤的心境中明白，《C++ Primer》第三版其实无法扮演人们在初学C++时的导师角色，它太庞大了。”

相比于276页的《Essential C++》，对任何一个希望在较短的时间内能够让自己的程序乖乖动起来的人来说，1237页的《C++ Primer》（当然，第五版内容已压缩为936页）都难说是短时间内的一个最优选择。所以原因很简单，上面的鸿篇巨著，尽管它们出自名家之手，内容包罗万象，讲解清晰透彻，但是对于初学者来说，它们实在是过于庞大复杂了。这里容我再次引用《Essential C++》前言中（关于作者需要速成Perl语言）的另一段话：

“好吧，我想，我去找本书抱抱佛脚好了——前提是这本书不能太厚，起码此刻不能太厚，而且他最好不要告诉我太多东西。虽然我知道我应该知道每一样东西，不过暂且等等吧。……此刻我不需要全世界最棒的Perl大全，我需要的是一本能妥善引导我前进，并使我不致偏离正轨过远的小书。我找到了Randal Schwartz的《Learning Perl》，它让我立刻上手并进展神

速、而且颇具阅读趣味。不过，就像其他有趣的计算机书籍一样，它也略去了不少值得一读的内容——虽然在那个时间点，我并不需要了解所有内容，我只需要让我的Perl程序乖乖动起来。”

所以请允许我不自量力的向Stanley Lippman致敬一次，本讲义也试图达到《*Essential C++*》相同的目标：轻薄短小，易于上手。它没有试图靡无巨细的对使用经济理论分析实际问题进行总结的野心，它不追求数学上的严谨性，一切以实用为本。一言以蔽之，我的目标是：让一个接受过基本宏观经济学训练，但是几乎没有程序设计基础的人，能够在相对短的时间内，学会几种常用且稳健的方法，经济实用的让你的程序跑起来。在这个目标上，我想我也应该说是受到了Joshua D. Angrist与Jörn-Steffen Pischke《*Mostly Harmless Econometrics: An Empiricist's Companion*》一书的影响。

那么具体说来，读者面前的这份讲义将在哪些方面区别于以往的教材呢？

1. 本讲义将讨论的经济学题材严格限定在长期增长、真实经济周期与异质性消费者三个话题中。我们不会讨论诸如投资的调整成本、名义价格刚性、财政政策、最优税收、资产定价、不完全合约等问题，这些主题尽管有趣，但是对初学者来说并不是不可或缺的。并且对这些问题的求解方法是扎根于本讲义中所涵盖的经典方法之内的，读者在阅读本讲义之后，可以自行对照[Ljungqvist and Sargent \(2012\)](#)等教材进行学习。过多的引入不必要的细节对于初学者来说很可能起到“只见树木，不见森林”的反作用。
2. 我们假定读者没有高级语言（C/C++, Fortran等）的编程经验。所以贯穿本讲义的另一个主题，是帮助读者学习面向过程的程序设计思想，理解寻求数值解和解析解背后逻辑的不同，养成良好的编程习惯，使得撰写的代码易于理解、调试以及后期维护。
3. 我们放弃了很多在现代经济学当中的应用已经不那么广泛的方法，比如[Ljungqvist and Sargent \(2012\)](#)一书中花费了大量篇幅讨论的线性二次型最优化；我们同时还舍弃了很多目的在于提高程序执行效率的算法，比如政策函数迭代法（Policy Function Iteration）和参数化期望效用算法（Parameterized Expectation Algorithm）等等。对于无穷期问题，我们对不同的模型将反复使用值函数迭代法（Value Function Iteration）及其在有限期和生命周期问题中的变种逆向归纳法（Backward Induction）。这两种方法尽管执行效率上存在劣势，但是其在大量问题的应用中所能够被证明的理论上的收敛性，使得这种方法在实践中具有相当的稳健性。作者这里认为，对初学者（甚至大多数前沿科学的研究者）而言，有一种稳健的方法能够得到正确的答案应该是排在效率之前第一位的目标。事实上，很多其他算法在效率上的改善同样依赖于研究者能够在求解中引入在值函数迭代求解过程中发现的模型的特殊性质。
4. 我们使用一组逻辑上层级发展的现代宏观经济理论作为贯穿这份讲义的主线，而不单纯的以所要求解的模型在技术上的发展作为引导全文发展的脉络。根据作者本人在攻读博士学位时候的经验，能够成功的将数值方法应用到具体的经济分析中，仅仅掌握技术是不够的，对一个经济模型的求解实际上分为三个步骤：将科研问题转化为经济模型，将经济问题转化为数学问题，以及最终的，将数学问题转化为程序设计问题。这也是在计算机程序设计当中所谓的“自顶而下”(Top-down)的程序设计思想。很多时候，很多同学所谓的“程序写的不好”，与具体语言的实现无关，而是由对于算法理解的模棱两可造成的。反过来，对于实际经济研究问题的强调，能够帮助读者将抽象的理论和更加底层的计算与触手可及的实际经济问题联系起来，从而使读者能够时刻把握住“为什么要求解模型”这个主线。计算机的作用是更快的帮助我们解决我们能够通过（在不计时间成本的前提下）人工计算解决的问题，而不是如同变魔术一般解决我们自己不知道如何应对的问题。由

同样的理由，对基本数学基础的介绍，我们将采取与现有教材不同的方法进行处理。特别地，我们将不会把动态规划理论、马尔科夫过程及其离散化求解以及插值与拟合等数学问题归入一个或几个前置章节当中进行统一介绍。相反，只有当我们分析的问题需要我们引入相应的概念并进行求解的时候，我们才会分出相应的篇幅进行讲解。作者认为，尽管这会打断阐述经济理论时的流畅性，但是无疑将会更清晰的帮助读者理解这些数学工具是怎样被引入经济分析当中的，从而使读者在学习的过程中能够目标明确，有的放矢。

5. 作者试图根据中国大陆普通高等学校一般经济类专业课程的设置对材料进行组织，以适应我国经济、管理类学生本科教育所构建的知识体系，为我国的研究生教育与国际接轨做一点微小的贡献。

本讲义当中的字体使用规范如下。正文中采用宋体五号字，英文采用11号Times New Roman字体。重要的名词、术语以及作者重点强调的内容采用中文楷体与英文斜体(*Italic*)。正文中的英文键盘输入代码采用11号等宽字体Computer Modern Typewriter。正文中的长代码用淡绿色(■)背景高亮，并使用9号Computer Modern Typewriter字体书写，语法高亮与MATLAB的交互式开发环境保持一致。正文中的文献索引采用蓝色字体，公式、插图、表格、代码与角注的索引采用红色字体，在自动跳转的阅览器中浏览本文档时，点击上述交叉引用将自动跳转到正文的相应部分。自动跳转功能对于目录、插图、表格、代码索引以及文末的参考文献索引也是有效的。

这份讲义脱胎于我在纽约州立大学石溪分校选修的由我的Ph.D导师Eva Cáceles-Poveda所讲授的《计算宏观经济学与金融学》(Computational Macroeconomics and Finance)，以及我的答辩委员会成员Alexis Anagnostopoulos在我Ph.D一年级时开设的宏观经济学两门课程。两位老师多年的指导使我获益良多，这份讲义能够写成，绝大部分的功劳应属于他们。

目录

| | |
|---------------------------|-----------|
| 前言 | iii |
| 目录 | vi |
| 插图一览 | viii |
| 表格一览 | x |
| 代码一览 | xi |
| 第一章 MATLAB 简介 | 1 |
| 1.1 脚本文件 | 2 |
| 1.2 自顶而下的程序设计思想 | 4 |
| 1.3 模拟生命周期收入 | 5 |
| 第一部分 长期增长 | 11 |
| 第二章 增长理论 | 13 |
| 2.1 外生储蓄的增长模型 | 14 |
| 2.2 内生储蓄的增长模型 | 19 |
| 第三章 确定性动态优化问题 | 25 |
| 3.1 动态优化问题 | 25 |
| 3.2 值函数迭代法 | 30 |
| 3.3 应用模型进行增长核算 | 38 |

| | |
|-------------------------------------|------------|
| 目录 | vii |
| 第二部分 经济周期 | 44 |
| 第四章 真实经济周期理论 | 46 |
| 4.1 Hodrick-Prescott滤波 | 47 |
| 4.2 战后美国经济周期的特征事实 | 51 |
| 4.3 真实经济周期模型 | 55 |
| 第五章 马尔科夫过程与数值积分 | 64 |
| 5.1 平稳过程 | 64 |
| 5.2 马尔科夫过程 | 72 |
| 5.3 马尔科夫过程的离散近似 | 78 |
| 5.4 数值积分 | 85 |
| 第六章 不确定性动态优化问题 | 91 |
| 6.1 独立同分布生产率冲击 | 91 |
| 6.2 一阶自回归生产率冲击 | 99 |
| 6.3 真实经济周期模型 | 111 |
| 6.4 校准模型 | 122 |
| 第七章 扰动法求解动态随机一般均衡模型 | 127 |
| 7.1 扰动法的一般理论 | 128 |
| 7.2 应用DYNARE实现扰动法求解DSGE模型 | 134 |
| 第三部分 收入与财富分配 | 142 |
| 第八章 流动性约束与预防性储蓄 | 144 |
| 第九章 异质性消费者问题 | 145 |
| 第十章 经济周期中的异质性消费者 | 146 |
| 第十一章 结语 | 147 |
| 参考文献 | 147 |

插图一览

| | |
|--------------------------------|----|
| 1.1 模拟生命周期收入 | 9 |
| 2.1 投资与折旧 | 15 |
| 2.2 投资与人均收入水平 | 16 |
| 2.3 资本的收敛路径 | 18 |
| 3.1 使用数值解描述余弦函数 | 31 |
| 3.2 值函数迭代法求解动态优化问题 | 38 |
| 3.3 战后美国国内生产总值 | 39 |
| 3.4 平衡增长路径 | 40 |
| 4.1 战后美国经济波动 | 47 |
| 4.2 HP滤波的性质 | 50 |
| 4.3 战后美国经济周期波动 · 一 | 52 |
| 4.4 战后美国经济周期波动 · 二 | 53 |
| 4.5 战后美国经济周期波动 · 三 | 54 |
| 5.1 随机游走与一阶自回归过程 | 70 |
| 5.2 非遍历的平稳过程 | 72 |
| 5.3 马尔科夫链近似随机过程 | 79 |
| 5.4 黎曼积分 | 87 |
| 5.5 牛顿-科特斯公式 | 88 |
| 6.1 独立同分布随机冲击 · 值函数 | 96 |
| 6.2 独立同分布随机冲击 · 政策函数 | 96 |
| 6.3 独立同分布随机冲击 · 仿真 | 99 |

| | |
|--------------------|-----|
| 6.4 AR(1)随机冲击・值函数 | 106 |
| 6.5 AR(1)随机冲击・政策函数 | 106 |
| 6.6 AR(1)随机冲击・仿真 | 109 |
| 6.7 随机稳态 | 109 |
| 6.8 平稳分布 | 110 |
| 6.9 劳动决策函数 | 118 |
| 7.1 鞍点路径 | 130 |
| 7.2 DYNARE文件的一般结构 | 138 |
| 7.3 真实经济周期模型的脉冲反应 | 140 |

表格一览

| | |
|--------------------------------------|-----|
| 2.1 以2010年美元记的人均生活水平的国际比较 | 14 |
| 3.1 逻辑运算符与关系运算符 | 38 |
| 4.1 美国战后经济周期波动数据 | 55 |
| 5.1 利用马尔科夫链近似 $N(0, 2)$ 分布 | 81 |
| 5.2 利用马尔科夫链近似AR(1)分布 | 85 |
| 5.3 常见正交多项式函数族 | 89 |
| 6.1 真实经济周期模型仿真结果 | 122 |

代码一览

| | | |
|-----|--------------------|-----|
| 1.1 | 模拟生命周期收入 | 6 |
| 2.1 | 模拟外生储蓄率的索罗模型 | 17 |
| 3.1 | 值函数迭代法求解动态优化问题 | 35 |
| 5.1 | 利用马尔科夫链近似正态分布 | 80 |
| 5.2 | 利用马尔科夫链近似AR(1)过程 | 83 |
| 6.1 | 独立同分布随机冲击 | 93 |
| 6.2 | 独立同分布随机冲击仿真 | 97 |
| 6.3 | AR(1)随机冲击 | 101 |
| 6.4 | 马尔科夫链近似AR(1)过程·函数 | 103 |
| 6.5 | AR(1)随机冲击仿真 | 105 |
| 6.6 | 真实经济周期模型 | 112 |
| 6.7 | 真实经济周期模型稳态 | 115 |
| 6.8 | 真实经济周期模型仿真 | 119 |
| 7.1 | 应用DYNARE求解真实经济周期模型 | 136 |

第一章 MATLAB 简介

既然这是一份面向初学者的讲义，那么我们不妨从程序设计的基础开始。在开始写任何程序之前，你所遇到的第一个问题是，我应该使用哪一种程序设计语言，是C/C++，是Fortran，还是MATLAB？贯穿本书，我们将使用MATLAB作为我们对模型进行计算与求解的主要工具。对数值计算稍有了解的同学可能会对上述语言之间的差异略有耳闻，我们这里也做一下简单的介绍。与另外三者不同，MATLAB通常被称为解释性语言（Interpreted Language），反过来另外三者被称为编译语言（Compiled Language）。两者最大的区别是解释性语言的程序可以不经过编译的过程逐行执行，而编译语言必须将程序作为一个整体进行编译、与操作系统的库函数进行链接，之后才会生成可执行的程序。如果大家对统计软件包比较熟悉的话就会发现，R与STATA两者也都是解释性语言。编译语言相比于解释性语言最大的优势是其在编译的过程当中编译器能够根据程序的结构对变量在内存中物理地址的分配进行编译时优化，这能够很大程度上提高代码的执行效率。当然此类事前优化的代价就是，在编写代码的时候，不同于解释性语言可以对代码进行逐行调试，编译语言对任何程序内子段的调试都必须撰写专门的主程序函数进行调用。如果你有一个比较长的，并且无法被模块化的程序，那么编译语言的调试就是一个非常痛苦的过程。

不过对于本教程中所求解的那些模型来说，参照本教程中所描述的算法进行编写的程序在主流的计算机配置下，使用MATLAB都是能够在可以接受的时间内完成运行的。鉴于我们在前言中已经说过，我们的目的是用最简单经济的办法，让各位手上的程序给出我们想要的结果，我们这里将选择MATLAB作为我们阐述算法主要语言。

既然已经说到程序设计语言的选择，那么不妨多说几句。对于已有编程基础或者对MATLAB的使用比较熟练的同学，在阅读本教程的时候将本教程中的例子使用编译语言求解也不失为一个很好的练习。尽管很多时候我们使用MATLAB的一个重要原因是它内建了很多成熟的子程序，比如插值、拟合、最优化、以及图形化输出等等，可以很大程度上减轻我们编写底层实现代码的工作量，但是这并不是说使用其他高级语言的时候我们都必须在这方面亲力亲为。对于C/C++的用户而言，开源科学计算库GNU Scientific Library (GSL)是一个免费、配置简单、功能齐全且算法稳健的类库。类似的，在The Netlib上汇集了大量可供Fortran直接调用的子程序。事实上，很多GSL当中的函数都是The Netlib上Fortran版程序基于C/C++的实现。如果各位使用英特尔公司提供的Fortran编译器Intel Visual Fortran，那么当中还会附带英特尔自己研发的Intel Math Kernel Library (Intel MKL)。^①不过说GSL配置简单只是一个相对的概念，要熟练使用GSL或者在C语言环境下调用MKL，读者还是要对编译器的设置有一定的理解，同时因为很多库函数频繁的使用了指针和结构等概念，所以要真正让我们手上的程序乖乖跑起来，也并没有想象中那么容易。这里另一个可供选择方案是William H. Press, Saul A. Teukolsky, William T. Vetterling和Brian P. Flannery共同编写的《Numerical Recipe》，该书同时有针对Fortran于C/C++的两种版本。《Numerical Recipe》的好处是读者可以直接将需要的程序定义成子函数，使用相对简单的接口在自己的程序中调

^①很遗憾，该编译器在Microsoft Windows环境下的版本是收费的。但是Intel针对Linux发行了免费的版本，熟悉Linux的同学可以考虑。

用，而无需与GSL等类库为追求泛用性而采用的复杂函数接口打交道。

推荐之余，这里也容作者根据自己的经验说两样至少是对于初学者来说不推荐的事情。首先，作者不推荐使用C++作为主力编程语言。不要误读，这里不是说C++语言本身不好，C++是一门无愧伟大的语言，读者从我反复在前言里引用Stanley Lippman的话就能看出来作者对于C++本身是没有任何偏见的。但是C++语言编写的初衷并不是数值计算，C++当中对于面向对象与泛型程序设计的友好支持在绝大多数经济学所面临的主要问题是面向过程的数值计算问题当中并没有什么用武之地。相反，标准C++编译器对很多语法阐释上的灵活性对很多初学者来说在调试时将成为挥之不去的梦魇。另一方面，如果读者比较一下C与C++标准教材的厚度，就能直观的感受到学习C++与C语言在精力方面需求的巨大差异。花费巨大的精力学习C++，之后将它作为C语言使用，在作者看来是一件非常得不偿失的事情。事实上，即便是C语言，在经济学当中的应用也是十分有限的。很多时候，我们选择编程语言的时候不仅仅要考虑自己的偏好，很大程度上还要考虑行业内部通行的习惯，能否成功的找到合作者，自己的代码能否在最大程度上保持可移植性等等。基于这个考虑，作者这里谨慎的推荐读者在编译语言中选择Fortran。理由有二。一，Fortran诞生的初衷就是解决科学计算问题，因此该语言的基本结构有许多针对科学计算的优化。^①二是经济学中Fortran的应用十分广泛，熟练掌握Fortran语言，不仅有助于与别人交流自己的程序，也有助于读者理解其他科研工作者的代码。

其次，作者这里非常不推荐大家使用闭源的商业科学计算库函数，比如RogueWave公司推出的*IMSL Numerical Libraries*。尽管此类商业库应用简单，算法通常也经过广泛的检验而在稳健性上有保证，但是调用这些需要花费不菲的成本才能使用的库函数，将极大的局限程序代码在不同系统和不同用户之间的可移植性。^②同时，正如我们在前言当中提到的，计算机语言在科学计算领域的作用是帮助我们完成我们会做但是不愿意做的事情，而不是替我们做我们自己不会做的事情。以最优化为例，经济学科学计算当中所面临的函数很多情况下具有不连续、不可导、不单调等等非常“不好”的性质，盲目使用闭源的代码对这些问题求解，很可能在调试阶段给寻找程序代码的纰漏造成额外的困难，或者使得读者在毫不知情的情况下将错误的解当做正确的而接受。这对于MATLAB函数的调用，乃至更一般的对于任何研究方法的使用上都是一样的。读者要尽量减少乃至避免使用自己不理解的工具进行研究，使用别人写好的程序的目的应该是使得自己不需要将时间和精力花费在自己了解细节、但是实现起来比较繁琐的底层代码的撰写上，而不是把这个作为学习上投机取巧的手段。

1.1 脚本文件

现在我们言归正传。在撰写本教程的时候，我试图达到的一个目标是培养读者作为一个能够熟练使用数值方法的经济学工作者的研究习惯。对此目标来说，最重要的是我们在面对一个经济学问题的时候，如果从白纸开始，把最终能够解决这个经济学问题的代码编写出来。因此，这里将不采取其他教程当中通行的以介绍MATLAB基本元素比如四则运算、画图、调用函数等为开始的结构，我们试图让读者从第一个例子开始，就养成一个“自顶而下”思考设计编写程序的习惯。因此，我们从一个非常简单的经济学例子开始，读者在以后的学习中将会看到，这个只用不到50行程序（包括注释）就可以解决的问题，当中几乎包含了所有我们在后面的复杂问题中所要处理的元素。

不过在这之前，我们先来熟悉一下MATLAB的图形式用户界面（Graphical User Interface, GUI）。打

^①这从FORTRAN名字的来源——Formula Translation当中就能看出。注意到正文中我们使用的是首字母大写的Fortran，而角注中我们使用了全大写的FORTRAN。Fortran通常被用来指代Fortran 90之后在现代个人计算机上运行的现代Fortran语言，而FORTRAN则被用来指代早期在卡片机时代使用的语言，Fortran在一定程度上保持了对FORTRAN的向下兼容性。

^②至少我自己对这种代码是深恶痛绝的，在再现这种代码的时候需要花费许多额外的精力对系统环境进行调试。

开MATLAB，屏幕正中间闪烁着命令提示符的就是MATLAB的交互式环境（Interactive Environment）。顾名思义，在这里你可以直接键入命令对MATLAB进行操作，MATLAB会对你键入的命令给予即时的反馈。比如，当你在交互式窗口中输入`1 + 2`之后回车确认，MATLAB会立刻返回信息`ans = 3`。这个特性在用MATLAB处理小规模运算的时候是非常有吸引力的，但是如果要以交互式的方法处理一个哪怕只有100行的程序，交互式编程的弊端就不胜枚举了。正如导演一个五分钟的情景剧或许还可以骑驴看唱本，如果要导演一部90分钟或者120分钟的电影，如果你不是王家卫的话，显然还是老老实实的搭一个工作室，把脚本组织好比较靠谱。使用MATLAB编写程序也是如此，所以通常情况下我们都不会过度的依赖交互式环境，本课程中，我们所使用的主要编程环境是被称为M脚本文件（M-Script File）的文本编辑环境，以后我们简称脚本文件。一个脚本文件是一个包含了一系列MATLAB语句的文本文件，当使用MATLAB执行脚本文件时，MATLAB将从文件的第一行开始按顺序依次执行文件中的语句直到最后一行，效果上这与我们直接在交互式环境下手工输入命令是等价的。

那么一个典型的MATLAB程序应该包含哪些要素呢？换句话说，如果我们要写一段求解一个经济学问题的程序，一般而言，这段程序应该包含哪些组成部分呢？

在具体回答这个问题之前，我们先考虑一下，如果我们要拍个简单的舞台剧，作为一个导演（是的，在数值计算你就是这个导演），我们都需要做哪些准备？首先，我得有一块场地，显然无论是布景还是演员都不能放在天上。其次，我得有几个能差遣的演员和工作人员，除非我拍的是自导自演的独角戏……让我们假设我们拍的不是这种。第三，我需要一个剧本，这个剧本要告诉每个演员在什么时候做什么事情，同时这个剧本上除了台词以外，我可能还需要做一些额外的笔记。最后，也许我还会考虑给我的舞台剧配上音乐，但是这个我做不来，我是个导演，不是个作曲家……所以我得雇一个专业的作曲家，再给他雇一些帮手来做这件事情。

看上去应该这些就足够了，那么类比到数值计算里，这些东西都分别对应了什么呢？显然，剧本对应了一个脚本文件，程序设计中的“台词”对应的便是MATLAB的语句。我们所希望附加的说明，在这里被称为注释(Comments)，注释语句在MATLAB当中以一个百分号%开头，任何出现在百分号之后的同一行的字符都将被编译器忽略。换句话说，注释符仅对当前行起作用，如果要一次性注释掉一段代码，需要在代码的每一行前面添加%。MATLAB中可以用键盘快捷键Ctrl + R一次性注释所有选中行。因为代码注释事实上改变了MATLAB执行的语句的内容，所以很多时候代码注释的用途并不是如它的名字所暗示的为代码提供详细的解释，而是通常被用来快速改变代码内容。比如，如果我要分别计算函数 $y = x^2$ 在 $x = 1$ 和 $x = 2$ 的返回值，那么我可以在脚本文件中一次性的写入`x=1;`和`x=2;`（我们马上会看到这两句代码的意义）两行代码，随后在计算 $x = 1$ 的情形时，注释掉`x=2;`，反之亦然。类似地，快捷键Ctrl + T可以用来快速将注释掉的代码恢复。上述两组快捷键将是各位在本课程中使用频率最高的快捷键，没有之一。另外，继承自C语言，每一条可执行语句之后需要有一个分号标识该语句的结束。第二，场地对应了内存当中的一段物理地址，在MATLAB中这被称为工作空间（Workspace）。幸运的是无论我们是使用MATLAB还是其他编译语言，与操作系统打交道的工作都不需要我们亲力亲为，我们所需要做的只是在主程序开始的时候将这部分内存当中原本填写的内容清空。这是通过语句`clear`完成的。当然需要注意的是不要在程序的中间放入该命令，这会导致与实验室的电源被保洁员关掉同样的效果。对于任何MATLAB的运算符、语句、命令、函数，使用`help`加上相应的字符可以调用系统内部的帮助文档。比如在交互式窗口中执行命令`help clear;`就会显示MATLAB关于`clear`命令的内建帮助文档，通常还附有范例。MATLAB的内建帮助文档是最好的学习MATLAB的途径，本讲义当中不会对使用的函数命令等做详细的介绍，请读者务必活用帮助文档。第三，算法中的运算是通过变量进行承载的，所以对应了演员，我们需要定义一系列的变量。这些变量依据功能与目的的不同，可以代表一个具体的值（实型变量）、一列字符串（字符串变量）、一个函数（指针变量，在MATLAB中有时被称为函数句柄，*Function Handle*）等等。然后，正如一个演员需要手持不同的道具、身着不

同的服装，变量在使用之前也需要被赋值。赋值运算是通过赋值运算符`=`实现的，比如在下面的代码1.1当中，第6行的`a = -2.17`的意思就是“定义一个实型变量`a`，并对它赋值`-2.17`”。对于初次接触程序设计的同学来说，很重要的一点是区分等号作为赋值运算符与判断逻辑关系相等之间的区别，后者是用两个等号“`==`”来表达的，我们在后文具体碰到逻辑运算的时候会做展开解释。任何一个被定义并赋值的变量会出现在MATLAB的工作空间当中，同时可以经过主界面中的Workspace窗口查询，在主界面的默认布局中，这个窗口在右上方2点钟位置。最后，辅助主程序进行计算的子代码是通过函数脚本文件（*Function-M File*）实现的，以后我们简称其为函数。在本章这个简单的例子当中我们会调用MATLAB内建的随机数生成函数，但是不会涉及到函数的编写，因此我们将函数编写的语法介绍放到后续章节当中进行。

1.2 自顶而下的程序设计思想

了解了一个通常的MATLAB代码当中所应该包含的元素，现在我们来看一个具体的例子，顺便介绍一下一般一个易于维护的代码应该如何布局。特别的，我们来完成本章开始时候提出的例子，即模拟一个代表性工人在生命周期中的收入流。也许有程序设计经验的读者已经迫不及待的想要上手写代码了，不过这里容我们最后做一点准备，再“浪费”一些篇幅。万事开头难，许多没有程序设计经验的同学可能会对自己如何起头感到没有头绪，为此，我们在这里简单的介绍一个“自顶而上”（Top-Downed）的程序设计思想，并在本教程随后的内容当中贯穿始终，进行不断强化。

所谓自顶而上的程序设计思想，究其本质，是将程序设计的核心从代码的具体撰写转移到算法设计上。过度关注代码实现的细节，以至于一叶障目，是初学者经常犯的错误。很多初学者在学习程序设计语言的时候养成的一个不好的习惯是急于求成，不管三七二十一，先坐在终端前面把代码写起来再说。这种编程方法通常在应对教科书上的简单范例来说还勉强可行，但是在实际应用当中，面对那些要上千行代码和众多函数相互交叉引用的程序时，往往就显得力不从心了。同学们随着自己编程经验的不断累积，将会越发意识到把程序写出来其实是很简单直接的一件事情，难的是知道应该写一个什么样的程序。具体到求解经济学模型也是一样的，最困难的部分是理解你要求解一个具有什么性质的模型，为此应该使用一种什么样的算法，搞清楚这些以后，程序的编写才会游刃有余。

自顶而下的程序设计思想简单来说就是“模块化”，即将大型任务分解成子任务，每个子任务可以用一段功能明确而容易处理的短代码实现，子任务与子任务之间相互独立，从而单个子任务可以分别编译调试，最后通过主程序将这些能够正确工作的子程序组织起来，以完成完整的任务。具体来说，自顶而下的程序设计通常包括如下六个步骤：

1. 清楚的陈述要回答的经济学问题；
2. 理清需要回答上述经济学问题，主程序的输入输出变量（*Input/Output Variables*）都应该是什；
3. 根据问题的特征，将经济学问题转化为数学问题；
4. 设计求解上述一系列数学问题的算法，写出伪代码（*pseudo-code*）；
5. 将伪码转化为可执行的程序语句；
6. 对程序进行调试。

根据图灵奖得主弗雷德里克·布鲁克斯在《人月神话》一书中的说法，在一个完整的代码编写周期之内，三分之一左右的时间被花费在对于问题的理解和算法的设计上，大约有六分之一的时间是用在具体的代码书写上，而余下百分之五十的时间都是用在调试（*debug*）上的。作者自身的经验与之相比大体也是吻合的。所以，熟练掌握自顶而下的程序设计思想，将会极大的减少整个项目周期内最为耗费时间的步骤即调试的压力，从而极大的加快项目实现的效率。

1.3 模拟生命周期收入

下面我们来看本章唯一一个例子。

例子1.1. 模拟一个代表性劳动者在生命周期中的收入

宏观经济学与劳动经济学问题的求解中一个常见的步骤是要模拟一个劳动者在生命周期中可能的收入流。具体来说，我们想要预测一个代表性的美国高中毕业生直到退休为止（这里设为21岁到65岁），他每一年的劳动收入。特别的，[Storesletten, Telmer and Yaron \(2004b\)](#) 应用收入动态的面板数据（Panel Data of Income Dynamics, PSID）证明一个代表性美国劳动者生命周期中的收入流（以对数形式）可以用如下的随机过程很好的刻画：^①

$$y_{it} = f(t) + u_{it} + v_{it} \quad (1.1)$$

$$u_{it} = \rho u_{i,t-1} + \varepsilon_{it}. \quad (1.2)$$

在上述两式当中， y_{it} 是最后的可支配收入， $f(t)$ 是一个随年龄与工龄自然增长的趋势项， u_{it} 是 t 期的持续性冲击（*Persistent Shock*）， v_{it} 是 t 期的临时性冲击（*Temporary Shock*）， ε_{it} 是持续性冲击在本期中的随机扰动（*Innovation*）。^②具体来说，该过程背后的经济逻辑是一个劳动者在某个年龄时候的劳动收入是由三个因素决定的。首先，随着劳动者年龄的自然增长，劳动者作为一个整体，其平均工资会因为工龄增长、职务升迁、改善的匹配效率等原因提高， $f(t)$ 刻画了这部分与年龄相关的工资变化。注意到这也是 $f(t)$ 只与 t 相关，而与个体劳动者*i*无关的原因。第二，相比于劳动者群体的平均水平，有些人可能会更快的获得升迁，另一些人可能不那么走运，碰上企业大规模裁员或者在劳动市场上四处碰壁。一般而言，我们认为此类因素对于劳动者收入的影响会持续相对较长的一段时间，持续时间的长度是由自回归参数 ρ 控制的。此类个体上的异质性（*Idiosyncratic Heterogeneity*）是由持续性冲击 u_{it} 刻画的，注意到此时 u 同时与下角标*i*和*t*相关。最后，单个劳动者每年的劳动收入还会受年终的奖金分红、临时性大额开支等相对独立而不会影响到后续年份收入的因素影响，这些因素由 v_{it} 刻画。在上述三个因素中， $f(t)$ 是确定性的，即事先可以预期的，而 u_{it} 和 v_{it} 是随机而事先无法预期的。于是在我们的模拟中， $f(t)$ 是计算的，而余下两者是仿真的。

我们不妨来具体实践一下如何依照上一小节当中叙述的自顶而下的程序设计思想对上述问题进行分解。正如其他程序设计教材当中的简单范例一样，读者乍看之下可能会觉得有些分析步骤有画蛇添足之嫌，这里确有大炮打蚊子的意味。但是当读者由简单的例子上手，慢慢熟悉了这种编程思想之后会发现，这种思想在本教程后面章节中更加复杂的范例求解中将会起到莫大的作用。

本例所要回答的经济问题很简单，即一个代表性美国工人在生命周期中的收入流。该问题所需要的输入变量是能够反映美国工人作为一个整体其收入的统计特征的随机过程，输入则是这个随机过程

^① 注意到这是一个带有漂移的自回归移动平均过程，即ARMA(1,1)过程。

^② 这里请读者注意区分持续性（*Persistent*）冲击与持久性（*Permanent*）冲击的区别，在本课程中这个区分是不重要的，但是对于大家阅读文献将会有帮助。

在一段有限时间内的具体实现。这个问题对应的数学问题，概括的说，就是计算由方程(1.1)和(1.2)刻画的随机过程的一个实现。具体来说，求解这个问题的伪码如下：

1. 对上述随机过程赋参数，即选择与数据的统计特征相吻合的 $f(t)$ 的函数形式以及随机冲击 u 和 v 的分布特征。
2. 对 $t = 21$ ，设此时 $u_{i,20} = 0$ ，通过随机数生成器（*Random Number Generator*）抽取 ε 在本期的实现（*Realization*），然后利用方程(1.2)计算本期的 $u_{i,21}$ 。
3. 再通过随机数生成器抽取 v 在本期的实现，然后利用方程(1.1)计算本期的收入 $y_{i,21}$ 。
4. 计算 $t = 22$ 时的情形。此时， $u_{i,21}$ 的值由第2步中计算得出，通过随机数生成器抽取 ε 在本期的实现，然后利用方程(1.2)计算本期的 $u_{i,22}$ 。
5. 再通过随机数生成器抽取 v 在本期的实现，然后利用方程(1.1)计算本期的收入 $y_{i,22}$ 。
6. 重复3、4两步，直到 $t = 65$ 。
7. 将所计算的 45×1 向量 \mathbf{y} 作图输出。

上述伪码转化成的MATLAB代码包含在代码1.1中。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code simulates a life-cycle income profile of a representative
7 % U.S worker
8 % - Code 1.1 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Fall 2015
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 11/09/2015: Original Code
17 %
18 clear;
19 %
20 % DATA DICTIONARY
21 %
22 % Parameterizations
23 a = -2.17;
24 b1 = 0.1682;
25 b2 = -0.0323/10;
26 b3 = 0.0020/100;
27 age = 21:1:65;
28 rho = 0.95; % Persistency
29 sigmae = 0.0106; % Variance to Persistent Innovation
30 sigmav = 0.0738; % Variance to Transitory Shock
31
32 % Declare Variables
33 v = zeros(45,1);
34 epsilon = zeros(45,1);

```

```

35 u = zeros(45,1);
36 y = zeros(45,1);
37 feffect = zeros(45,1);
38
39 % -----
40 % SIMULATION
41 % -----
42 feffect = a + b1*age + b2*age.^2 + b3*age.^3; % Calculate the Trend
43 v = sigmav^.5 * randn(45,1); % Generate T.Shock
44 epsilon = sigmae^.5 * randn(45,1); % Generate P.Shock
45
46 y(1) = feffect(1);
47 u(1) = epsilon(1) ;
48 for indage = 2:1:45 % Calculate the log-earnings
49     u(indage) = rho * u(indage-1) + epsilon(indage);
50     y(indage) = feffect(indage) + u(indage) + v(indage);
51 end
52
53 % Plot the results
54 figure(1);
55 plot(age,y);
56 title('Simulated Earnings Profile');
57 xlabel('Age');
58 ylabel('Earnings');
59
60 figure(2)
61 hold on; % Not to erase previous curve when plot a new one
62 plot(age,y); % The individual profile
63 plot(age,feffect,'r'); % The trend profile
64 legend('Simulation','Trend','Location','SouthEast');
65 title('Simulated Earnings Profile');
66 xlabel('Age');
67 ylabel('Earnings');
68 hold off; % Turn off curve overlapping

```

代码 1.1.—模拟生命周期收入

现在我们来具体分析代码1.1。1至16行是代码的说明段，注意到这里所有的语句都是以注释的形式出现的。这一段代码在具体执行的时候不起任何作用，但是这能够概要的总结代码的功能、版本、所需要用到的外部函数库等等。也许你持续调试一段代码的时候能够对这段代码的种种细节了如指掌，但是如果你需要在若干个月甚至若干年之后重新对代码进行维护，从可执行的代码来反推代码的功能及算法就不容易了，此时，代码说明以及代码中其他部分的注释就显得尤为重要了。

第18-37行为变量初始化部分，如代码内的段落小标题（19-21行）所示，通常这部分内容被称为数据字典（*Data Dictionary*）。数据字典通常由两部分组成，第一部分记载了程序的参数（22-30行），即在程序运行过程中保持不变的部分；第二部分定义了程序的变量（32-37行），即在程序运行当中用来存储运算的中间结果和最终输出结果的变量。对于解释性语言来说，数据字典部分并不需要如某些编译语言一样放在第一条可执行语句之前，因为解释性语言并不需要在程序实际运行之前为所有变量分配内存空间。^①尽管将所有变量定义在程序开头会一定程度上降低代码的可读性（中等规模的程序一般要伴随数百个变量），不如使用时就近定义容易理解，但是作者这里依然推荐读者在编写程序的时候将变量

^① 这部分代码在编译语言当中被称为变量声明。在编译语言当中，Fortran和基于C89标准的C语言严格要求所有变量必须在第一条可执行语句之前声明。但是C99标准的C语言、C++和Java并没有这个要求，变量只需要在使用前被定义过即可。这源于C99和C++对于增强程序可读性的鼓励，此类编程风格通常被称为就近声明。

声明置于可执行程序之前。这样做有三个好处。第一，不同于软件开发，数值计算要频繁调试程序的参数，此时将所有的参数模块化的置于程序初始能够极大的减少后期维护时在几千行程序当中寻找一个变量声明语句所需要花费的时间。第二，就近定义尽管已经为很多现代编译器所支持，但是旧标准的C语言编译器和现代Fortran编译器并不支持这种编程风格，因此将变量声明前置可以极大的提高程序在不同平台和系统之间的可移植性与兼容性。不过为了尽可能的在兼顾泛用性的前提下提高程序的可读性，在数据字典当中将变量按照一定的逻辑关系进行归类以及进行详细的变量注释就显得十分重要了。读者将在本教程后半部分比较复杂的代码当中看到更加贴近现实的范例。

为具体刻画方程(1.1)和(1.2)的形式，我们需要设定 $f(t)$ 的函数形式，为自回归系数 ρ 赋值，以及设定 v 与 ε 所服从的概率分布。这里我们使用[Cocco, Gomes and Maenhout \(2005\)](#)的校准形式。具体来说，我们用一个 t 的三次多项式描述 $f(t)$:

$$f(t) = a + b_1 \cdot t + b_2 \cdot t^2 + b_3 \cdot t^3. \quad (1.3)$$

我们再假设 v_t 与 ε_t 服从i.i.d的均值为零的正态分布，其方差分别为 σ_v^2 和 σ_ε^2 。依照[Cocco, Gomes and Maenhout \(2005\)](#)，我们取 $a = -2.170, b_1 = 0.1682, b_2 = -0.0323/10, b_3 = 0.0020/100; \rho = 0.95, \sigma_v^2 = 0.0738$ 以及 $\sigma_\varepsilon^2 = 0.0106$ 。另外，我们定义了一个向量age用来标识工人的年龄，其代码age = 21:1:65的意思是“定义一个行向量，该向量的元素由一个等差数列给出，该数列的首项为21，末项为65，公差为1”。这些对应了第22-30行程序。

现在我们来看程序的结果应该存储在哪些变量当中，这部分任务是由32-37行完成的。因为我们要模拟的代表性工人在模型中的生命周期为45期（从21岁到65岁），因此我们需要一系列的 45×1 的向量来存储每一期与工人当期的收入相关的参数。具体来说，我们用向量y, u, v, epsilon和feffect保存 y, u, v, ε 和 $f(t)$ 的值。所有变量在这里都被定义为列向量，同时初始值赋为零。这里有两点需要提请读者注意。第一，向量并非一定要定义为列向量，这里统一定义为列向量是为了与Fortran语言的缺省设置一致。第二，变量定义之后并非一定要赋值为零，向量的初始值可以为任意实数。另外，读者除了可以应用命令zeros(m,n)生成m行n列的零矩阵，MATLAB还支持使用固有函数(built-in function)构造一矩阵(ones(m,n))，单位阵(eye(n))，以向量v为对角元的对角阵(diag(v))，以及不定结构的空矩阵(null matrix)A(A = [])。

做好全部准备之后，我们在39-50行进行程序的核心运算。注意到我们所要解决的问题的本质是一个动态问题（特别的，这是一个马尔科夫过程），为相邻两期的计算建立联系的变量就是持续性冲击 u_t ，原因是此时 u_t 的值与 u_{t-1} 相关。但是当我们在 t 期更新了 u_{t-1} 的取值以后， y_t 的计算方法与 $t-1$ 期是完全相同的，这提示我们可以使用循环体结构(Loop)来简化代码书写。另一方面，这个问题中其他一些变量，如feffect, v, epsilon的取值在各期之间又是相互独立的，对这部分变量，我们可以在循环体执行之前对他们进行统一处理。因此，我们一次性的计算出feffect的全部取值(42行)，并抽取所有45期当中i.i.d的v与epsilon的实现(43、44两行)。这里有两点需要指出，第一，42行中幂函数运算^前面的.表明该运算是以元素为单位的。具体的，对一个任意的 2×2 矩阵A，代码A.^2执行的运算为

$$\begin{bmatrix} a_{11}^2 & a_{12}^2 \\ a_{21}^2 & a_{22}^2 \end{bmatrix},$$

而A与另外一个 2×2 矩阵B元素相乘的结果A.*B为

$$\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}.$$

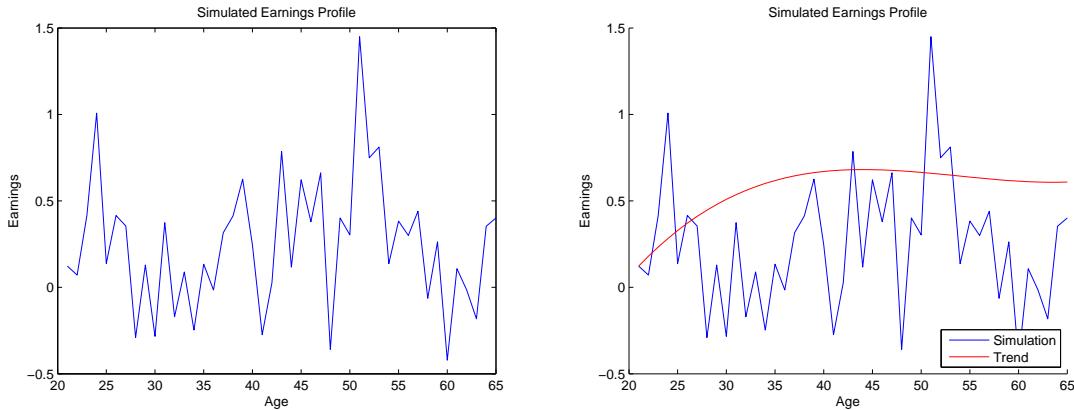


图 1.1—模拟生命周期收入

第二，在43、44两行中我们通过`randn(m,n)`调用固有随机数生成器生成 $m \times n$ 的矩阵，矩阵中的每个元素都是从标准正态分布中抽取的随机数，从这个标准正态分布出发，利用均值和标准差，我们可以构造出任意参数的正态分布。值得注意的是，我们可以把上述两组随机数的抽取放到循环体当中对每一期运算分别抽取，但是这样做会因为需要反复调用函数与传递参数降低算法的执行效率，故而这里将所有随机数统一抽取。

46-51行便是进行核心计算的循环体语句。特别的，任何循环体运算之前要对关键的动态变量进行初始化，循环体的实质运算从第二期开始。在这里，我们使用了MATLAB中两种循环体中的`for`循环，又称计数循环。`for`循环通常用于循环体执行次数已知的问题，后文当中我们将会介绍另一种用于条件判断循环体终止的`while`循环，又称迭代循环。如代码所示，一个`for`循环由循环体结构语句48、51和循环体语句49、50组成。其中49、50两个语句分别对应了方程(1.2)和(1.1)，这里不做进一步解释。循环体结构语句的终止句为51句，它仅由一个关键词`end`书写。循环体语句的起始句`for indage = 2:1:45`的意思是“以`indage`为循环控制变量，从`indage = 2`开始执行循环体语句，每次执行完循环体语句之后对循环控制变量`indage`加1，直到执行过`indage = 45`之后跳出循环体”。注意循环体起始句与终止句总是成对出现的。执行过循环体语句之后，我们就已经完成了计算。最后，我们通过53-68行代码图形化的输出程序运算的结果，图1.1当中左图对应`figure(1)`的输出，右图对应`figure(2)`的输出结果。这一段代码的细节是很容易掌握的，这里不做具体解释，对于他们的理解留作作业帮助读者熟悉`help`命令的运用，请读者活用`help figure`和`help plot`进行查询。

本教程随后章节将依照现代宏观经济学相关的主题分为三个部分：增长理论，经济周期理论，以及收入与财富分配理论。在计算方法上，上述三个部分恰好对应了基于代表性消费者的确定性动态优化、带有不确定性的动态优化，以及上述两种技术在异质性消费者问题中的应用。这同时也是现代宏观经济学自罗伯特·索罗之后的发展脉络。特别的，长期增长理论中，我们将学习Ramsey-Cass-Koopmans模型[Ramsey (1928), Koopmans (1963)和Cass (1965)]的求解，具有高年级本科生或研究生一年级高级宏观经济学基础的同学应该相当熟悉这个模型的连续时间版本。^①在短期经济波动当中，我们将要求解一个随机增长模型[Brock and Mirman (1972)]和一个真实经济周期模型[Hansen (1985), Rogerson (1988)]。同时，这一部分我们还将简要的介绍现代宏观经济学中数量分析当中模型的校准(Calibration) [Cooley and Prescott (1995)的第一章]。在收入与财富分配理论当中，我们首先将学习如

^①经典的教科书包括Blanchard and Fischer (1989), Romer (1996)以及最近的Acemoglu (2009)。

何求解一个存在收入不确定性的动态消费者决策模型，即Deaton-Carroll模型[[Deaton \(1991\)](#), [Carroll \(1997\)](#)和[Gourinchas and Parker \(2002\)](#)]。技术上，此时消费者决策模型中价格是给定的，因此我们不需要考虑价格的决定机制。掌握了个体消费者问题的求解技术之后，我们将要学习如何求解一个存在异质性消费者的稳态模型。此时，价格是在一般均衡当中内生决定的。因此我们需要刻画当存在不确定性的时侯，均衡当中消费者的财富和收入分布的性质。这是通过对于前述章节的消费者决策问题的加总来完成的。我们将主要关注Huggett-Aiyagari模型[[Huggett \(1993\)](#) 和[Aiyagari \(1994\)](#)]。注意到，Huggett-Aiyagari模型当中不存在生产率冲击，因此这类模型可以被看做是异质性消费者版本的增长理论。在本教程的最后，我们将要一起学习的是收入与财富分配在经济周期当中的波动，即一个异质性消费者版本的真实经济周期模型。这里，我们将要学习求解Krusell-Smith模型[[Krusell and Smith \(1998\)](#)]。本教程当中主要关注模型的全局解，即使用非线性方法对模型进行数值求解。非线性方法的求解较为精确，但是求解对技术要求比较高、同时也需要相当程度的计算时间。相对于全局解的这两个问题，在研究经济周期的学者中间存在另一种较为流行的算法，即扰动法(*Perturbation Method*)。该方法使用多项式对模型在稳态附近进行逼近。因为该方法的解只在稳态附近有较好的精度，所以此类解又被称为局部解。关于扰动法，我们将学习如何使用基于MATLAB和C++的DYNARE软件包对模型进行求解。一般而言，扰动法并不仅仅可以用来对经济周期模型进行求解，但是因为其在经济周期文献当中使用的最为广泛，而经济周期研究当中使用的模型通常被称为动态随机一般均衡模型（*Dynamic Stochastic General Equilibrium Model*, 即DSGE模型），所以扰动法通常都是伴随DSGE模型一起出现的。本节中我们将求解一个国际经济周期模型，并将局部解与全局解的性质进行比较[[Anagnostopoulos and Tang \(2015\)](#)]。

第一部分

长期增长

第二章 增长理论

从北美和西欧到撒哈拉以南非洲，世界各国的人均收入水平千差万别，我相信读者对此是没有异议的。以2014年为例，美国的人均国民生产总值（GDP）以2014年美元记，达到了53,042美元。而当年中国和印度的人均GDP分别只有6,807美元和1,499美元，只有美国的13%和3%不到。事实上，中国和印度远远不是所有国家人均收入的队伍的副班长，表2.1列出了世界银行统计的世界上人口总量最大的14个国家在2010年人均GDP的水平。当中我们可以看到，最贫穷的孟加拉国的人均收入水平只有美国的百分之一多一点。而全部有世界银行统计数据的国家中，最贫穷的马拉维共和国人均收入水平不到美国的0.5%。对造成世界各国人均收入存在如此巨大差异的原因的探求，有系统的讨论可以追溯到十五世纪中叶的重商主义时期。从配第、布阿吉尔贝尔、魁奈到古典经济学理论体系的创立者亚当·斯密，经济学几乎从她甫一降生，就对富国裕民的主题展开了大量的讨论。这从《国富论》一书的全称，即《国民财富的性质和原因的研究》，便可见端倪。而关于价值论和分配论的讨论则要到十九世纪拿破仑战争之后，才在主流经济学界引起重视，关于经济周期现象的研究更是要等到二十世纪三十年代的大萧条。所以说经济增长是宏观经济学乃至整个经济学中最古老和重要的议题，尽管可能略有夸张，但是想来也不为之太过。

近代宏观经济学关于长期增长的研究最早可以追溯到哈罗德—多玛模型[Harold (1939), Domar (1946)]。1987年诺贝尔经济学奖得主罗伯特·莫顿·索罗在其基础上发展出了作为新古典宏观经济学基础的外生增长模型[Solow (1956)]用来对国家间人均收入的水平与增长率的差异进行解释。^①简单来说，索罗的增长理论可以归纳为三个层次。资本积累决定稳态中的人均收入水平，人口的长期增长决定了经济总量的增长，而以人均产出衡量的技术进步造成了人均收入长期增长的驱动力。作为一份以计算方法为核心的教程，这里我们主要关注如何分析资本积累（亦即储蓄率）对稳态中人均收入水平的影响，对人均收入水平长期增长有兴趣的同学可以参考内生增长理论相关的文献。^②

具体来说，本章的第一节我们复习一个简单的外生储蓄率的增长模型，这个模型可以帮助我们理解索罗模型的核心机制，即由储蓄率决定的资本存量会导致人均收入的不同。^③在第二小节中，我们将储蓄率内生，讨论在一个简单的有限期消费储蓄模型当中，储蓄率决定的机制。由于此时储蓄率是由消费者行为内生求解的，这个版本的增长模型可以用来研究各种政策对储蓄率、进而对人均收入的影响。对无限期模型以及求解当中需要用到的动态优化理论的讨论，我们将放到下一章进行。

熟悉经典高级宏观经济学教材的同学可能会注意到在模型设置上，本教程与经典教科书的不同。

^① 外生增长模型有时候也称为索罗—斯旺模型，这源自澳大利亚经济学家特雷佛·斯旺在1956年独立提出了与相似的理论[Swan (1956)]。值得一提的是，索罗本人对于现代宏观经济学方法即动态一般均衡模型的态度是十分不以为意的，他本人对于后人将新古典增长模型的源头追溯到自己甚至是有些反感的。有兴趣的读者可以参看索罗2008年发表在Journal of Economic Perspectives上的一篇短评[Solow (2008)]。

^② 经典的文献见Romer (1986, 1990)，想要深入了解内生增长理论的读者也可以阅读罗伯特·巴罗与哈维尔·萨拉-伊-马丁的教科书[Barro and Sala-i-Martin (2003)]。

^③ 注意外生增长模型中的外生指的是技术进步率外生，以区别于内生增长模型，并不是说储蓄率是外生。

表 2.1
以2010年美元记的人均生活水平的国际比较

| 国家 | 人均GDP | 国家 | 人均GDP |
|-----|--------|-------|-------|
| 美国 | 47,140 | 印度尼西亚 | 2,580 |
| 德国 | 43,330 | 菲律宾 | 2,050 |
| 日本 | 42,150 | 印度 | 1,340 |
| 俄罗斯 | 9,910 | 尼日利亚 | 1,180 |
| 巴西 | 9,390 | 越南 | 1,100 |
| 墨西哥 | 9,330 | 巴基斯坦 | 1,050 |
| 中国 | 4,260 | 孟加拉国 | 640 |

[†] 数据来源：世界银行。

经典宏观经济学教科书为了与历史上的文献保持一致，通常采用连续时间的环境进行建模，但是绝大多数前沿文献现在使用的都是离散时间。作为一份目的在于向读者介绍前沿文献中常用的计算方法的讲义，我们这里选择与前沿文献一致的离散时间设置。尽管有此细微的差异，但是离散时间和连续时间在经济学直觉方面并无本质的不同，两者仅在数学推导的细节上存在形式上的不同，这里提请读者注意。

2.1 外生储蓄的增长模型

在具体进入模型细节之前，我们首先再明确一下我们想要回答的经济学问题。我们试图找到各国人均收入之间存在巨大的差异背后的经济学机理，也就是什么因素决定了一国的人均收入。从早先的资源禀赋理论到道格拉斯·诺斯的《西方世界的兴起》再到Acemoglu的《国家为什么会失败》，毋庸置疑，答案是多种多样的。在本节当中，我们试图研究的是储蓄这一因素，对于一国人均收入的影响。这便是我们想要回答的经济学问题。

2.1.1 储蓄与人均收入

既然谈到消费储蓄，那么我们不妨从消费者的消费储蓄决策讲起。一般来说，消费者一段时间内（比如一个月）的收入主要用作两个用途，通过消费提高当下生活质量和通过储蓄提高未来的生活质量。这两种使用对同一份资源即收入来说是竞争关系，如果要提高本期的生活水平，那么留作下一期使用的资源就会受到影响。一个国家收入的使用也是如此，生产资源到底用来生产当年使用的消费品，还是用来生产能够提高未来生产能力的资本，以提高未来的消费能力，便是索罗模型当中核心的经济权衡。最优的分配方式直觉上便是既要避免寅吃卯粮，也要避免裤腰带勒的太紧，能够在保证生活水平稳步增长的前提下，快速奔小康。下面我们来看一下数学上如何对上述简单的经济逻辑进行刻画。

我们考虑一个封闭经济。假设消费者的效用函数是线性的且不存在跨期贴现

$$U = \sum_{t=1}^{\infty} c_t,$$

c_t 是 t 期消费者的人均消费。人口总量保持在 \bar{N} 不变，同时储蓄率 s 外生给定。经济当中将生产要素转化为产出的技术由生产函数 $y_t = f(k_t)$ 给出，其中 y_t 和 k_t 分别为人均产出和人均资本。我们设 $f(\cdot)$ 是单增的凹函数，通过原点，且在原点处斜率为无穷大。产出有两种用途，用作当期消费 c_t 和投资 i_t 以提高下

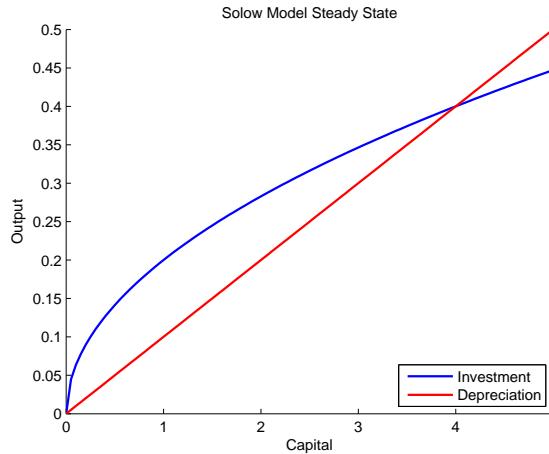


图 2.1.—投资与折旧

一期的生产能力，资源不存在浪费。再假设全部的储蓄都转化为投资品，即

$$i_t = sy_t. \quad (2.1)$$

那么，由国民收入恒等式 $y_t = c_t + i_t$ 我们得到

$$c_t = (1 - s)y_t. \quad (2.2)$$

注意到从方程(2.1)和(2.2)可以看出，在模型中投资和消费是被资本存量唯一决定的，给定储蓄率，两者都关于资本存量单增。于是，问题就转化到如何求解模型中的资本存量。此时，资本存量的动态受到两个因素影响：一方面，新的投资会增加经济中的资本存量；另一方面，已有的资本在使用中会自然折旧。我们假设一个简单的线性折旧率 δ ，那么连续两期之间的资本变化可以用下面的方程刻画：

$$k_{t+1} = (1 - \delta)k_t + i_t. \quad (2.3)$$

上式是说下一期的资本存量等于本期的已有的资本在下一期还能正常使用的那些，加上新添置的资本。注意到新添置的资本实际上在下一期会产生两种效果，首先更多的资本毋庸置疑会带来更多的产出，但是反过来更多的资本也意味着下一期将产生更多的折旧。由于生产函数假设为凹函数，所以当资本存量较低时，额外的资本投入会带来很高的产出，而线性折旧下并不会导致太多的维护成本，扣除折旧之后资本存量会得到提高。反过来，当资本存量较高时，继续增加投资能够带来的收益会变得十分有限，但是维护成本的增加却会保持线性，所以更多资本带来的未来产出的增加未必能够补足更大规模的折旧带来的损失，资本存量可能反而会因为入不敷出而降低。这提示我们资本存量应该存在一个额外产出恰好能够替换掉折旧的平衡点，我们称这个平衡点为稳态(*Steady State*)。资本存量在稳态当中保持不变，在方程(2.3)中带入 $k_{t+1} = k_t$ ，我们得到稳态中资本存量 k^* 的定义

$$sf(k^*) = \delta k^*, \quad (2.4)$$

即投资恰好补足了折旧。这里我们还可以进一步证明，当 $k_t \leq k^*$ 时， $k_{t+1} \geq k^*$ ，即长期来看，经济会收敛到稳态。这完成了我们对于模型环境的刻画。下面我们来回答本节开始的时候我们提出的经济学问题，即一国的储蓄率是如何影响一国稳态中的人均收入水平的。对应到模型当中的记号，我们要讨论的是 k^* 与 s 之间的关系。

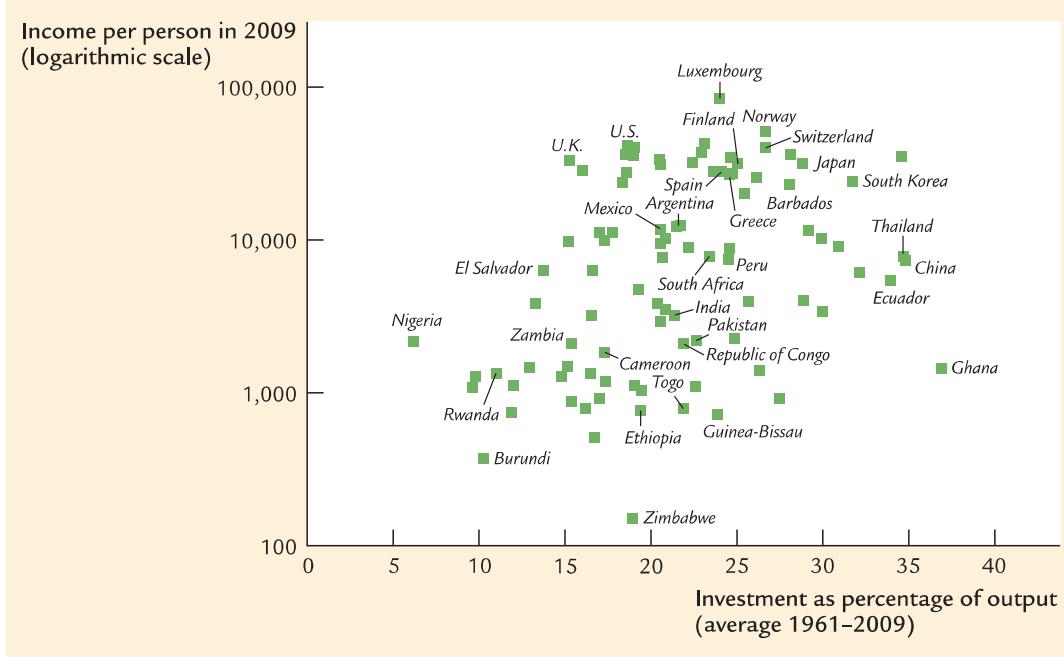


图 2.2—投资与人均收入水平

注意到方程(2.4)实际上定义了一个 k^* 关于 s 的隐函数 $k^* = k^*(s)$, 所以应用隐函数定理, 我们得到

$$f(k^*(s)) + sf'(k^*) \frac{dk^*}{ds} - \delta \frac{dk^*}{ds} = 0. \quad (2.5)$$

由生产函数的性质,

$$\lim_{k \rightarrow 0} sf'(k) = \infty > \delta,$$

且

$$\lim_{k \rightarrow 0} sf(k) = 0 = \lim_{k \rightarrow 0} \delta k$$

所以若 $sf(k^*) = \delta k^*, \forall k^* > 0$, 必有

$$sf'(k^*) < \delta,$$

否则两者不可能相交。图2.1表示了一个具体的例子, 该图应用了后文中例子2.1的参数设置。上式带入方程(2.5), 我们得到

$$\frac{dk^*}{ds} > 0, \quad (2.6)$$

即人均资本以及与之相对的人均产出随储蓄率增加而增加。故而索罗模型的一个预测是便是更高的储蓄率导致了更高的人均收入。如图2.2显示, 应用Penn World Table进行的跨国比较所观察到的现象是与模型预测相一致的。

下面我们来看一个简单的例子, 这个例子表明索罗模型在长期当中会收敛到稳态当中。

例子2.1. (索罗模型在长期当中的收敛) 假设生产函数的形式为

$$y = k^{\frac{1}{2}},$$

折旧率 $\delta = 0.1$, 储蓄率 $s = 0.2$ 。对上述索罗模型, 求解稳态中的资本、投资、消费与产出, 并计算从初始资本 $k_0 = 2$ 开始经济收敛的路径。

由方程(2.4)可以直接求出 $k^* = 4$, $y^* = 2$, $i^* = 0.4$ 和 $c^* = 1.6$ 。由于 $k_0 < k^*$, 所以在均衡路径当中, k_t 应该是渐增的。对于均衡路径的仿真, 在程序上的实现与上一章当中的例子1.1是类似的。在例子1.1中, 联系相邻两期收入动态的变量是持续性冲击的水平 u_t (当然收入关于外生变化的年龄的变动是平凡的), 对应了代码1.1的第49行。在这里, 核心的动态方程是下一期的资本与本期资本与投资之间的关系, 即方程(2.3)。只要计算出 k_t 的转移路径, 经济当中其余变量的路径便可由 k_t 的路径唯一求解。求解这个模型的伪码与例子1.1是类似的, 限于篇幅这里从略, 对应的MATLAB代码包含在代码2.1当中。这里核心的动态方程, 即方程(2.3)体现在代码2.1的第41行。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code simulates a Solow growth model.
7 % - Code 2.1 in the Lecture Notes
8 %
9 % Author:
10 % Xin Tang @ Wuhan University, Fall 2015
11 %
12 % Record of Revisions:
13 % Date: Description of Changes
14 % -----
15 % 12/12/2015: Original Code
16
17 clear;
18 %
19 % DATA DICTIONARY
20 %
21 % Parameterizations
22 t = 1:1:100;
23 alpha = 0.5;
24 s = 0.2;
25 delta = 0.1;
26 k = zeros(100,1); % Path: Capital
27 c = zeros(100,1); % Path: Consumption
28 y = zeros(100,1); % Path: Output
29 invs = zeros(100,1); % Path: Investment
30
31 % Initialization
32 k(1) = 2;
33 y(1) = k(1)^alpha;
34 invs(1) = s*k(1)^alpha;
35 c(1) = y(1) - invs(1);
36
37 %
38 % SIMULATION
39 %
40 for i = 2:1:100
41     k(i) = (1 - delta)*k(i-1) + s*k(i-1)^alpha; % Capital Evolution
42     y(i) = k(i)^alpha;
43     invs(i) = s*k(i)^alpha;
44     c(i) = y(i) - invs(i);
45 end
46

```

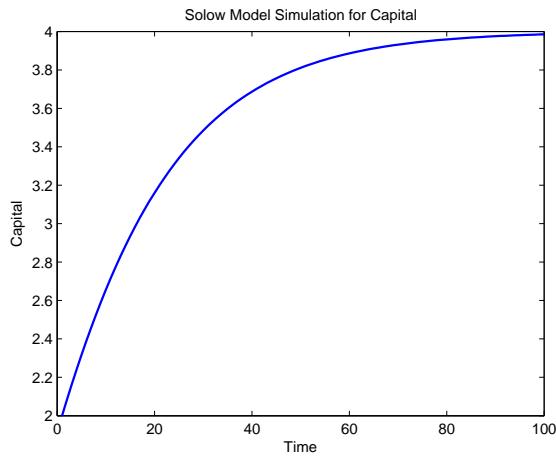


图 2.3.—资本的收敛路径

```

47 figure(1)
48 plot(t,k(1:100),'LineWidth',1.5);
49 title('Solow Model Simulation for Capital');
50 xlabel('Time');
51 ylabel('Capital');
52
53 figure(2)
54 plot(t,c,'LineWidth',1.5);
55 title('Solow Model Simulation for Consumption');
56 xlabel('Time');
57 ylabel('Consumption');
58
59 figure(3)
60 plot(t,invs,'LineWidth',1.5);
61 title('Solow Model Simulation for Investment');
62 xlabel('Time');
63 ylabel('Investment');
64
65 figure(4)
66 plot(t,y,'LineWidth',1.5);
67 title('Solow Model Simulation for Output');
68 xlabel('Time');
69 ylabel('Output');

```

代码 2.1.—模拟外生储蓄率的索罗模型

图2.3当中显示了资本的收敛路径，这对应了代码2.1的47-51行。

2.1.2 最优储蓄率

上一小节中我们用一个简单的索罗模型说明了储蓄率可以影响稳态当中的人均收入水平，尽管更高的储蓄率总是伴随着更高的稳态人均资本存量，这又会导致更多的人均产出与收入，但是这并不代

表储蓄率应该是越高越好。对 $s = 1$ 的情形而言，即一个经济所有的产出都用来进行投资，整个经济的产出确实可以在长期内达到最大化，但是对于经济增长的最终目标，即富国裕民而言，富国是达到了，裕民就未必了。这种不计消费的产出最大化政策通常能够在现代国家进入战争动员的状态时得到体现。在第二次世界大战当中，交战的各国都动员了最大的工业潜能，以产出记的产量远高于和平时期，但是即便是战火没有波及的美国本土，普通民众的生活水平与和平时期相比也是下降的。因此尽管产出与消费高度相关，更高的产出并不总意味着更高的消费，所以从长期经济发展的角度来看，如何刻画一个最优储蓄率，便是一个重要的问题。

如果我们同意经济增长的目标是提高一国国民的消费水平，那么我们可以就此定义一个最优储蓄率，即能够最大化稳态消费的储蓄率，这在文献中又被称为黄金律(*Golden Rule Rate*)。下面我们将具体求解黄金律。

将方程(2.4)带入(2.2)，得到

$$c^* = f(k^*) - \delta k^*.$$

将上式关于 s 求一阶条件，得到

$$\frac{dc^*}{ds} = [f'(k^*) - \delta] \frac{dk^*}{ds} = 0. \quad (2.7)$$

由方程(2.6)，上式等于零的充要条件是

$$f'(k^*(s)) = \delta, \quad (2.8)$$

即黄金律下，资本的边际产出恰好能够替换掉资本的自然折旧，从而使长期中消费水平不至下降。

我们从经济直觉的角度再深入理解一下黄金律的含义。我们考虑边际上一单位资本投资的用途，首先产出上，这能够为我们提供额外的 $f'(k)$ 的产品，这可以被用作消费与投资。注意到这一单位额外的资本在下一期会有 δ 部分自然折旧掉，如果我们不把这部分折旧的资本补足，那么长期而言这个资本水平就是不可持续的，也就不可能成为一个经济的稳态。所以，当 $f'(k) > \delta$ 时，用一部分额外的产出补齐折旧，剩下的便可以或者用来消费，或者用来投资，这时候追加的这一单位投资从长期增长的角度来看便是有益的。反过来，当 $f'(k) < \delta$ 时，追加一单位投资所带来的额外产出不足以补贴伴随投资产生的折旧，必须从已有的产出当中动用额外的资源才能保证产出水平长期的可持续性，此时，进行投资从提高消费的角度来看便不是最优的。因此，边际上，当额外的产出恰好能够补齐折旧时，投资的效益是最大化的。

2.2 内生储蓄的增长模型

索罗模型尽管揭示了储蓄率与人均收入之间的关系，但是作为核心机制的储蓄率，其取值是完全外生给定的。由于缺少一个结构化的对消费者消费储蓄决策的描述，索罗模型并不适合用来进行对经由储蓄率影响产出的政策进行分析。为此，本节放松索罗模型当中储蓄率 s 为外生的假设，来具体刻画一国在消费储蓄是所面临的具体权衡，即拉姆塞—卡斯—库普曼斯模型。本节中，我们首先通过一个两期的模型来理解卡斯—库普曼斯模型的核心机制，随后我们将两期模型扩展到一般的有限期，来讨论随着规划时间的增加，消费储蓄决策是如何变化的。特别的，我们将会讨论模型当时间 T 趋于无穷时的极限性质。这将为我们下一章的无穷期模型以及动态优化问题的学习打下基础。

为了更好的反应一国在消费储蓄决策当中所面临的权衡，我们需要对消费者偏好引入更多的结构。在上一节当中，消费函数是线性的且不存在跨期贴现。我们希望在我们的新模型当中刻画两个新古典

经济学当中广泛承认的消费者偏好的性质，边际效用递减与跨期消费贴现。不过这里我们依然保持消费者效用的跨期可加性，于时，消费者的偏好此时可以写作

$$U = \sum_{t=0}^{\infty} \beta^t u(c_t), \quad (2.9)$$

此时 $0 < \beta < 1$ 是贴现率，每期的效用函数 $u(\cdot)$ 是单增的凹函数，且在极限处满足 $\lim_{c \rightarrow 0} u'(c) = \infty$ 和 $\lim_{c \rightarrow \infty} u'(c) = 0$ 。经济当中余下的结构与外生储蓄率的索罗模型相同。

2.2.1 两期模型

让我们先从一个简单的两期模型上手。此时，效用函数是

$$U(c_0, c_1) = u(c_0) + \beta u(c_1).$$

资本的更新方程依然为

$$k_{t+1} = (1 - \delta)k_t + i_t, \quad t = 0, 1$$

同时每一期经济中的资源分配必须处在由资源约束定义的帕累托边界内部

$$c_t + i_t \leq f(k_t), \quad t = 0, 1.$$

设初始资本 k_0 已知，那么此时的社会最优规划问题 (*social planner's problem*) 写作

$$\begin{aligned} U &= \max_{c_0, c_1, k_1} u(c_0) + \beta u(c_1) \\ s.t. \quad & \\ c_0 + k_1 - (1 - \delta)k_0 &= f(k_0), \\ c_1 - (1 - \delta)k_1 &= f(k_1). \end{aligned} \quad (2.10)$$

注意到在上述问题当中我们将资源约束取等，排除了资源未被充分利用的情形。同时，我们隐含带入了 $k_2 = 0$ 。这两个条件通过库恩—塔克条件可以很容易的证明，读者应在研究生一年级的宏观经济学或者动态优化相关课程中学习过，限于篇幅这里从略。该问题所对应的拉格朗日函数为

$$\mathcal{L} = u(c_0) + \beta u(c_1) + \lambda_0 [f(k_0) - c_0 - k_1 + (1 - \delta)k_0] + \lambda_1 [f(k_1) - c_1 + (1 - \delta)k_1],$$

其中 λ_0, λ_1 分别为两期资源约束条件的拉格朗日乘子。^① 上述问题的一阶条件为：

$$\begin{aligned} c_0 : u'(c_0) &= \lambda_0, \\ c_1 : \beta u'(c_1) &= \lambda_1, \\ k_1 : \lambda_0 &= \lambda_1 [f'(k_1) + (1 - \delta)], \end{aligned}$$

三者相结合，我们就得到了关于跨期消费储蓄决策的欧拉方程

$$u'(c_0) = \beta u'(c_1) [f'(k_1) + (1 - \delta)]. \quad (2.11)$$

^① 这里简单指出，拉格朗日函数法求解带有约束的最优化问题数学上的目的是将一个有约束的最优化问题转化为一个无约束的最优化问题，后者可以通过一系列极值定理求解。其基本的直觉是对于在约束集合允许的范围之外取值进行一定的“惩罚”，比如在这里的例子里，如果资源约束不满足即 $c_0 + k_1 - (1 - \delta)k_0 > f(k_0)$ ，当 $\lambda_0 > 0$ 时，这一项为负，从而会降低 \mathcal{L} 函数的值。而问题的关键就是找准正确的“惩罚”程度。详见 Dixit (1990)。

注意到求解这个两期的模型，我们实际上需要求解的是本期的消费 c_0 ，本期的储蓄 $k_1 - (1 - \delta)k_0$ ，以及下一期的消费 c_1 一共三个变量。两期的资源约束加上欧拉方程(2.11)组成了一个三个独立方程的方程组，从而可以给出唯一解。

仔细来看这三个方程，我们发现其实当储蓄 k_1 确定以后， c_0 与 c_1 由相应的会计恒等式可以直接解出来。所以消费者实质上的决策在这里是由欧拉方程决定的。现在来看该方程的经济含义。方程左侧为边际上一单位储蓄的成本，根据现有的模型设置，一单位资源除了储蓄之外便是做消费使用，所以增加一单位储蓄的成本是放弃一单位消费，消费者对这一单位消费的估值是通过边际效用进行的。给定效用函数的性质，当消费水平较低的时候，放弃一单位消费对消费者来说福利上的损失是很大的，而反过来，当消费者已经有足够的水平的消费时，这个成本就不大。方程的右侧刻画了一单位储蓄的收益，这是由两部分组成的。首先，额外的一单位资本可以带来更高的产出，但是这同时也意味着更高的折旧。所以第二期当中，额外的资本所带来的产出中可以被用作净消费的就是 $f'(k_1) + (1 - \delta)$ 。消费者对这部分产品的估值同样是由边际效用进行的，不同之处是未来的消费以本期的产品计价时需要进行贴现。所以综合起来，欧拉方程是说，消费者的最优决策是使得分配到每一期的资源带来的净效用是相等的。注意到，在方程(2.11)当中带入上一节中索罗模型的假设 $u(c) = c$ 和 $\beta = 1$ ，我们能够得到之前的黄金律表达式

$$f'(k_1) = \delta.$$

2.2.2 有限期模型

现在我们考虑一个复杂一点的模型，模型的设置与上一小节大致相同，只是此时我们设时间是 $t = 0, \dots, T$ ，最优化问题(2.10)此时成为

$$\begin{aligned} U &= \max_{\{c_t, k_{t+1}\}_{t=0}^T} \sum_{t=0}^T \beta^t u(c_t) \\ s.t. \quad c_t + k_{t+1} - (1 - \delta)k_t &= f(k_t). \quad \forall t = 0, 1, \dots, T \end{aligned} \tag{2.12}$$

注意到此时共有 $T+1$ 个资源约束条件。上述问题的选择变量共有 $2T+2$ 个，因为最后一期 $k_{T+1} = 0$ （类比两期问题中 $k_2 = 0$ ），我们还剩下 $2T+1$ 个变量，所以需要 $2T+1$ 个相互独立的方程。于是，在 $T+1$ 个资源约束以外，我们需要另外 T 个独立方程。此时，优化问题(2.12)对应的拉格朗日函数为

$$\mathcal{L} = \sum_{t=0}^T \left\{ \beta^t u(c_t) + \lambda_t [f(k_t) + (1 - \delta)k_t - c_t - k_{t+1}] \right\}, \tag{2.13}$$

其中 λ_t 是 t 期资源约束的乘子，共有 $T+1$ 个。对方程(2.13)关于 c_t 和 k_{t+1} 求一阶条件，对于 $t = 0, 1, \dots, T-1$ ，我们有

$$\begin{aligned} \beta^t u'(c_t) &= \lambda_t, \\ \lambda_t &= \lambda_{t+1} [f'(k_{t+1}) + 1 - \delta]. \end{aligned}$$

两者综合，我们得到 $t = 0, 1, \dots, T-1$ 时的欧拉方程

$$u'(c_t) = \beta u'(c_{t+1}) [f'(k_{t+1}) + 1 - \delta], \tag{2.14}$$

这给了我们所需要的另外 T 个方程。于是，此时整个经济的动态路径可以被通过联立方程组求解。可以看到，尽管规划的时间维度变长了，但是消费者最优化的核心依然是在满足资源约束的前提下，使得资源在每一期的带来的收益是相等的。

注意到，在 $T+1$ 期模型当中，我们只能得到 T 个跨期的欧拉方程，所以整个问题能够求解依赖于 $k_{T+1} = 0$ 来消减变量的总数，否则整个方程组的解就是不定的。在下一章当中我们将会看到，无穷期问题之所以需要使用动态优化的方法求解，其原因就在于无穷期模型当中缺少一个终值条件，从而使得方程数量总是比未知数“少一个”。^①另外，除了最后一期，整个规划问题前面 T 期结构是完全一致的。特别的，我们这里定义，若一个问题可以以两期为单位进行分析，我们就称这样的问题是递归的(*recursive*)。

现在我们用一个具体的例子来分析一下有限期消费储蓄问题的性质。

例子2.2. (对数效用与柯布一道格拉斯约束) 设折旧是完全的，即 $\delta = 1$ 。再设生产函数是柯布一道格拉斯形式的

$$f(k) = k^\alpha, \quad 0 < \alpha < 1,$$

且消费者偏好由对数函数给出

$$u(c) = \ln c,$$

计算储蓄函数 $k_{t+1} = g(k_t)$, $\forall t = 0, 1, \dots, T$ 的形式。

我们用正文中同样的方法构造拉格朗日函数，带入 $\delta = 1$ ，对 $t = 0, 1, \dots, T-1$ 期可以得到欧拉方程

$$u'(f(k_t) - k_{t+1}) = \beta u'(f(k_{t+1}) - k_{t+2}) f'(k_{t+1}),$$

再带入 $u(\cdot)$ 和 $f(\cdot)$ 的函数形式，我们得到

$$\frac{1}{k_t^\alpha - k_{t+1}} = \frac{\alpha \beta k_{t+1}^{\alpha-1}}{k_{t+1}^\alpha - k_{t+2}}, \quad t = 0, 1, \dots, T-1.$$

将上式的左侧分子分母同时除以 k_t^α ，右侧除以 k_{t+1}^α ，我们得到

$$\frac{1/k_t^\alpha}{1 - k_{t+1}/k_t^\alpha} = \alpha \beta \frac{1/k_{t+1}}{1 - k_{t+2}/k_{t+1}^\alpha}. \quad (2.15)$$

定义储蓄率

$$z_t = \frac{k_t}{k_{t-1}^\alpha},$$

带入方程(2.15)之后我们得到储蓄率的递归表达式

$$\frac{z_{t+1}}{1 - z_{t+1}} = \frac{\alpha \beta}{1 - z_{t+2}}, \quad t = 0, 1, \dots, T-1. \quad (2.16)$$

重新定义一下角标，我们把上面的式子写成

$$\frac{z_t}{1 - z_t} = \frac{\alpha \beta}{1 - z_{t+1}}, \quad t = 1, \dots, T.$$

^①这个说法其实并不准确，因为比“无穷”少一个这种说法数学上是不成立的。

整理之后，得到

$$z_{t+1} = \alpha\beta \left(1 - \frac{1}{z_t}\right) + 1, \quad , t = 1, \dots, T. \quad (2.17)$$

注意到方程(2.17)实际上就是 $t = 1, \dots, T$ 时的欧拉方程。这时候，我们在上式中带入 $k_{T+1} = 0$ ，以储蓄率表达即 $z_{T+1} = 0$ ，我们得到

$$z_T = \frac{\alpha\beta}{1 + \alpha\beta}.$$

对 $T - 1$ 期递归使用方程(2.17)，又可以求出

$$z_{T-1} = \frac{\alpha\beta + (\alpha\beta)^2}{1 + \alpha\beta + (\alpha\beta)^2}.$$

T 和 $T - 1$ 期的函数形式提示我们猜想， t 期的储蓄率的函数形式为

$$z_t = \frac{\alpha\beta + (\alpha\beta)^2 + \dots + (\alpha\beta)^{T+1-t}}{1 + \alpha\beta + (\alpha\beta)^2 + \dots + (\alpha\beta)^{T+1-t}}, \quad t = 1, \dots, T + 1. \quad (2.18)$$

对 t 应用数学归纳法可以证明(2.17)中的函数形式是正确的，同时注意到在(2.17)当中， t 的取值可以包括 $T + 1$ 。应用等比数列求和公式，(2.17)可以写成

$$z_t = \alpha\beta \frac{1 - (\alpha\beta)^{T-t+1}}{1 - (\alpha\beta)^{T-t+2}}, \quad t = 1, \dots, T + 1.$$

于是，此时 $t = 0, 1, \dots, T$ 期的储蓄函数为

$$k_{t+1} = \alpha\beta \frac{1 - (\alpha\beta)^{T-t+1}}{1 - (\alpha\beta)^{T-t+2}} k_t^\alpha. \quad (2.19)$$

现在我们来看储蓄率函数 z_t 的性质。首先注意到数列 $\{z_t\}_{t=1}^{T+1}$ 是一个单减序列，即 $z_t > z_{t+1}$ 。其次，相邻两期之间的差异 $z_t - z_{t+1}$ 是一个单增序列，而当 $T \rightarrow \infty$ 时，可以验证 $\lim_{T \rightarrow \infty} z_1 - z_2 = 0$ 。事实上，若对方程(2.19)关于 T 取极限，应用极限的四则运算法则，我们得到

$$k_{t+1} = \alpha\beta k_t^\alpha. \quad (2.20)$$

对上述三点进行分析，我们发现距离终末期越远，储蓄率越高，同时相邻两期之间的储蓄率变化越小。在极限情况下，相邻两期的储蓄率是完全相同的。我们似乎可以猜想，终末期的存在，在当前这个例子当中，是储蓄率变化的唯一原因。当不存在终末期，也就是无穷期规划时，消费者固定的将 $\alpha\beta$ 的总产出备作未来使用，消费剩下的 $1 - \alpha\beta$ 。同时当 T 足够大时，消费者初始期的行为可以对无穷期当中消费者的行为进行很好的近似。在下一章的无限期间问题当中，我们将一起来验证上述猜想。

另外在进入下一章之前，我们再回来回顾一下为什么要把储蓄率内生化，作者希望读者在花费了大量时间理解技术细节之后，总是可以在合上书本以前从宏观的角度进行一下总结。如果我们的目的只是回答储蓄率如何经由投资和人均资本影响人均收入，那么我们是不需要将储蓄率内生的，外生储蓄的索罗模型就足以回答这个问题。但是如果我们想要讨论储蓄率的决定因素，索罗模型就不适合了。一般而言，如果我们要讨论一个以“若A则B”为形式的假说，这里的B必须是由模型所刻画的机制内生决定的。对应到本章的例子，索罗模型当中，我们要探讨的假说是“如果储蓄率变化，那么人均收入如何变化”，这里A是储蓄率的变化，B是人均收入的变化。回忆我们2.1节当中的模型，当中A是外生给定的，而B是由模型的机制——这里是由方程(2.3)刻画的——决定的。而如果我们要讨论的问题是“若某些政府政策发生变化，储蓄率会如何变化”，索罗模型显然就不适合了，这里我们就需要将储蓄内生

化，对卡斯—库普曼斯模型进行分析。比如如果我们要讨论提高消费税对人均收入的影响，在卡斯—库普曼斯模型中，我们只需要将最优化问题(2.12)中的资源约束改为

$$(1 + \tau_c)c_t + k_{t+1} - (1 - \delta)k_t = f(k_t),$$

而这在索罗模型当中是无法做到的。注意到这里我们分析是标准的索罗模型，不包括在索罗模型当中求解黄金律，细心的读者可能发现，黄金律中所对应的储蓄率与卡斯—库普曼斯模型当中的刻画是相同的，而索罗模型在求解黄金律时，最优储蓄率实际上已经是内生决定的了。事实上，从卡斯—库普曼斯模型的设置——在资源约束的情况下最大化消费——可以看出，其求解的实际上就是索罗模型中的黄金律。

第三章 确定性动态优化问题

回忆我们上一章的讨论，我们引入卡斯—库普曼斯模型的目的是讨论最优储蓄率是如何决定的。为此，我们从一个两期的模型上手，最后推导了一个 T 期的模型。我们从一个例子中看到，这个 T 的模型有一个很好的性质，那就是随着规划的时间变长，初始期之间储蓄率的变化是非常小的。那么这个 T 期的模型是否达到了我们的目的呢？不完全是。

之所以说不完全，是因为这里我们并没有得到一个唯一的储蓄率，而是对 T 中的每一个 t 都得到了一个对应的储蓄率，实际政策分析当中应该使用哪个储蓄率，这个问题一定程度上是悬而未决的。但是另一方面，在上一章的例子中，由于初始期的储蓄率关于规划时间 T 的极限存在，所以如果我们的目的是分析一项政策对一个进行长期规划的国家来说的长期效果，使用这个极限储蓄率似乎是合理的。不过现在这还只是我们的一个猜想，并没有得到证明。对应到索罗模型中的情况，我们实际上没有刻画的，是卡斯—库普曼斯模型当中的稳态。

为此，本章当中我们讨论无限期模型的求解。我们首先讨论如何将一个序列形式的无限期问题转化成以两期为周期的递归问题。然后我们来看例子2.2在无穷期问题下的版本（例子3.1），并讨论无穷期解与有限期解的极限之间的关系。这里我们将看到，与有限期问题中每期对应一个具体的储蓄率不同，无穷期情况下的储蓄率是一个与时间 t 无关的函数，以后称为政策函数（policy function）。例子2.2的一个好处是存在解析解，但实际问题中这样的好运气并不多见，所以在最后我们讨论如何使用值函数迭代法对无穷期的动态优化问题进行数值求解。卡斯—库普曼斯问题的数值解的算法是本教程当中所有算法的基础，请读者务必掌握。

3.1 动态优化问题

现在我们把2.2.2小节中的模型扩展到无穷期。最优化问题(2.12)此时成为

$$\begin{aligned} U &= \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{s.t.} \\ c_t + k_{t+1} - (1 - \delta)k_t &= f(k_t) \quad \forall t \\ k_0, \text{ 已知} & \end{aligned} \tag{3.1}$$

这里唯一的变化就是规划的时间由 T 变为无穷期了。为了陈述方便，让我们这里不严谨的使用一下极限的语言，设此时我们仍然在面对一个 T 期的问题，只不过 $T \rightarrow \infty$ 。下面我们来求解一下上面这个问题，首先让我们试试看上一章的解法这里是不是依然奏效。注意到此时我们有 $2T + 2$ 个变量需要求解，为

此需要 $2T + 2$ 个独立方程。像往常一样，预算约束为我们直接提供了 $T + 1$ 个方程，所以我们需要另外找到 $T + 1$ 个方程。构造最优化问题(3.1)所对应的拉格朗日函数：

$$\mathcal{L} = \sum_{t=0}^{\infty} \left\{ \beta^t u(c_t) + \lambda_t [f(k_t) + (1 - \delta)k_t - c_t - k_{t+1}] \right\}, \quad (3.2)$$

其中 λ_t 依旧是 t 期资源约束的乘子，这里共有 $T + 1$ 个。对方程(3.2)关于 c_t 和 k_{t+1} 求一阶条件，对于 $t = 0, 1, \dots, T - 1$ ，我们有

$$\begin{aligned} \beta^t u'(c_t) &= \lambda_t, \\ \lambda_t &= \lambda_{t+1}[f'(k_{t+1}) + 1 - \delta]. \end{aligned}$$

两者综合，我们再次得到 $t = 0, 1, \dots, T - 1$ 时的欧拉方程

$$u'(c_t) = \beta u'(c_{t+1})[f'(k_{t+1}) + 1 - \delta], \quad (3.3)$$

此时一共有 T 个。

现在问题来了，所有的方程加在一起，我们现在有 $T + 1 + T = 2T + 1$ 比变量数量少一个，方程组的解此时是不定的。跟(2.12)的求解过程相对照，我们发现出岔子的地方是这里我们缺少了 $k_{T+1} = 0$ 这个条件。在有限期问题当中，因为消费者只规划到 T 期，所以 $T + 1$ 期是不是洪水滔天在有限期问题当中是不被考虑的，在无穷期问题中如此的寅吃卯粮显然是不理性的，因为站在 T 期考虑，对有限期问题的消费者而言是不存在“明天”这个概念的，而对无穷期的消费者而言，依然要面对一个如下的最优化问题：

$$\begin{aligned} U &= \max_{\{c_t, k_{t+1}\}_{t=T}^{\infty}} \sum_{t=T}^{\infty} \beta^{t-T} u(c_t) \\ s.t. \\ c_t + k_{t+1} - (1 - \delta)k_t &= f(k_t) \quad \forall t \geq T \\ k_T &\text{ 已知,} \end{aligned}$$

此时 $k_{T+1} = 0$ 明显不是最优解。同时，因为 T 在这里是一个一般化的记号，所以这种“缺少一个方程”的问题无论我们考虑多大的 T ，应用有限期的求解方法都是没法解决的，面对这个死胡同，我们需要换个思路。注意到这里我们并不需要 $k_{T+1} = 0$ 本身，我们所需要的，只是 $T + 1$ 期资本的决定方式，也就是 k_{T+1} 关于 k_T 的函数。换句话说，我们其实只需要额外找一个方程 $k_{T+1} = Z(k_T)$ 。现在，我们就来使用动态优化的方法求解这个额外的方程。

对任意的 t ，让我们定义值函数 $V(k_t)$ 为

$$\begin{aligned} V(k_t) &= \max_{\{c_s, k_{s+1}\}_{s=t}^{\infty}} \sum_{s=t}^{\infty} \beta^{s-t} u(c_s) \\ s.t. \\ c_s + k_{s+1} - (1 - \delta)k_s &= f(k_s) \quad \forall s \geq t \\ k_t &\text{ 已知,} \end{aligned}$$

如果我们记给定 k_t 时约束集 $c_s + k_{s+1} - (1 - \delta)k_s = f(k_s), \forall s \geq t$ 为 $\Gamma(k_t)$, 那么上述问题等价于

$$V(k_t) = \max_{c_t, k_{t+1}} \left\{ u(c_t) + \beta \max_{\{c_s, k_{s+1}\}_{s=t+1}^{\infty} \in \Gamma(k_{t+1})} \sum_{s=t+1}^{\infty} \beta^{s-(t+1)} u(c_s) \right\}$$

s.t.

$$c_t + k_{t+1} - (1 - \delta)k_t = f(k_t)$$

k_t 已知,

带入 $V(k_t)$ 的定义, 上式可以简洁的写作

$$V(k_t) = \max_{c_t, k_{t+1}} \{u(c_t) + \beta V(k_{t+1})\}$$

s.t.

$$c_t + k_{t+1} - (1 - \delta)k_t = f(k_t)$$

k_t 已知。

(3.4)

注意到此时最优化问题(3.4)的结构是递归的。回想我们现在的目标是寻找一个一般的 k_{t+1} 关于 k_t 的方程, 这个方程应该是跟具体的 t 无关的, (3.4)看上去是一个满足我们要求的最优化问题: 这个问题跟序列形式表达的无穷期问题是等价的, 同时这个递归问题的结构与具体的 t 无关。正因为此, 动态优化问题的书写中我们通常省略下标 t 。在后文中, 我们约定使用不带下标的 c, k 等代表本期变量, 而使用带有'的变量 k' 等代表下一期变量。我们称具有与 $V(k) = \max_{c, k'} \{u(c) + \beta V(k')\}$ 类似形式的方程为贝尔曼方程 (*Bellman Equation*)。

我们这里不展开叙述如何将一个一般的序列问题写成动态优化形式, 细节请读者对照 Stokey and Lucas (1989) 的四、五两章进行学习。简单来说, 将一个序列问题转化为递归的动态优化问题, 读者需要仔细的分析原问题的那些部分是在每期当中周期性出现的, 这部分内容就构成了递归问题当中某一期的具体内容。之后, 读者需要分析相邻两期之间的问题是如何相互影响的。通常, 本期问题的求解会依赖于一些前期已经给定的变量的值, 我们称这些变量的为状态变量 (*State Variables*), 对应到(3.4)就是本期的资本存量 k 。另外, 本期问题的求解, 通常是指对于另外一些变量, 给定状态变量的取值, 它们的取值应如何选择, 这些变量我们称为控制变量 (*control variables*), 对应到(3.4)当中就是本期的消费 c 与储蓄 k' 。刻画控制变量与状态变量之间关系的函数就是本章开始时我们所说的政策函数, 注意到, 动态优化问题求解的最终目的就是求解政策函数。最后, 本期的储蓄在本期内是控制变量, 但是到下一期的会变成状态变量。动态优化名字中的动态两字, 指的便是这种本期行为对下期造成的影响, 对应到(3.4)当中, 便是资源约束 $c_t + k_{t+1} - (1 - \delta)k_t = f(k_t)$ 。同时, 具有这种跨期身份转变性质的状态变量我们又称其为内生状态变量 (*Endogenous State Variables*)。所以将一个序列问题写成递归问题, 核心就是正确的定义状态变量、控制变量以及刻画了本期控制变量与下期状态变量, 并找出定义了动态本质的内生状态变量。限于篇幅, 请读者在实际学习研究中对上述原则进行理解。

不过当我们打算具体求解的时候就发现问题来了。首先, 我们不知道 $V(k)$ 的形式, 无论我们打算用多么现代的数学工具, 不知道自己想要干什么都是绝对不行的, 所以我们得想办法把 $V(k)$ 的函数表达式找出来。其次, 这个函数表达式还要满足一个特定的条件, 那就是在(3.4)中花括号内定义的最优化运算之后, 成品必须等于他自己。(3.4)的这两个特征与我们目前为止所熟悉的经典数学分析问题有显著的不同, 如同上一章当中我们处理的那些问题, 经典的数学问题当中我们面对的未知量主要是一个或一列具体的数字, 无论他们是整数、实数还是复数, 我们所处理的函数其函数形式 (*Functional Form*) 本身一定是已知的, 实际的计算主要是对这些函数进行各种各样的运算。这里不一样的地方是,

我们所要处理的问题，其函数形式本身就是未知的。所以要求解(3.4)， $V(\cdot)$ 函数的函数形式必须在求解过程中一并求出来。数学上，此类一个函数其函数形式本身作为未知量的问题通常需要应用泛函分析(*Functional Analysis*)的知识进行分析。^①那么当未知量是一个形式未知的函数的时候，我们应该如何求解呢？或者说，更基本的问题是，在实际操作层面上什么叫做“函数形式未知”的问题呢？让我们用一个具体的例子来直观的感受一下。我们使用无穷期版本的例子2.2来进行说明。

例子3.1.（对数效用与柯布一道格拉斯约束）设折旧是完全的，即 $\delta = 1$ 。再设生产函数是柯布一道格拉斯形式的

$$f(k) = k^\alpha, \quad 0 < \alpha < 1,$$

且消费者偏好由对数函数给出

$$u(c) = \ln c,$$

求解动态优化问题

$$\begin{aligned} V(k) &= \max_{c,k'} \{u(c) + \beta V(k')\} \\ s.t. \\ c + k' &= f(k) \\ k \text{ 已知。} \end{aligned} \tag{3.5}$$

既然最优化问题(3.5)脱胎于例子2.2，我们不妨依样葫芦，投石问路。注意到我们这里求解的核心是要找到一个刻画 $V(k)$ 的方法。我们不妨先假设 $V_0(k) = 0$ ，如同我们在有限期问题当中所假设的那样，这里下标0不代表任何经济上的含义，只是用来标记这是我们所做的第几次尝试。此时将 V_0 带入(3.5)的最优化算子 \max 内部，我们得到

$$V_1(k) = \alpha \ln k.$$

如法炮制，再将 $V_1(k)$ 带入(3.5)，我们得到

$$\begin{aligned} V_2(k) &= \max_{c,k'} \{\ln c + \alpha \beta \ln k'\} \\ s.t. \\ c + k' &= k^\alpha. \end{aligned} \tag{3.6}$$

上式此时对应了拉格朗日函数

$$\mathcal{L} = \ln c + \alpha \beta \ln k' + \lambda [k^\alpha - c - k'],$$

按照标准方法求解一阶条件之后我们得到储蓄函数

$$k' = \frac{\alpha \beta}{1 + \alpha \beta} k^\alpha,$$

注意到这个式子与例子2.2中 z_T 的表达式是一致的。从储蓄函数中我们可以推出消费函数的表达式

$$c = \frac{1}{1 + \alpha \beta} k^\alpha,$$

^①当然我们这里不会应用很复杂的泛函分析知识，所以数学基础稍微薄弱的读者也不必紧张。

将两者带回(3.6), 得到

$$V_2(k) = \left[\ln \frac{1}{1 + \alpha\beta} + \alpha\beta \ln \frac{\alpha\beta}{1 + \alpha\beta} \right] + \alpha(1 + \alpha\beta) \ln k$$

对照 V_1 和 V_2 的形式, 我们发现他们看上去似乎都可以用 $\ln k$ 的线性表达式来描述, 那么我们不妨把一个一般的式子

$$V(k) = M + N \ln k$$

带入(3.5)求解。具体的, 此时(3.5)成为

$$\begin{aligned} V(k) &= \max_{c, k'} \{ \ln c + \beta(M + N \ln k') \} \\ s.t. \quad & \\ c + k' &= k^\alpha. \end{aligned} \tag{3.7}$$

对应的拉格朗日函数为

$$\mathcal{L} = \ln c + \beta(M + N \ln k') + \lambda[k^\alpha - c - k'],$$

通过一阶条件我们可以求出

$$\begin{aligned} k' &= \frac{\beta N}{1 + \beta N} k^\alpha, \\ c &= \frac{1}{1 + \beta N} k^\alpha. \end{aligned}$$

将最优决策带回(3.7), 我们得到

$$M + N \ln k = \ln \frac{1}{1 + \beta N} + \alpha \ln k + \beta [M + N \ln \beta N - N \ln(1 + \beta N) + \alpha N \ln k] \tag{3.8}$$

上式定义了一组基于待定系数法的关于 (M, N) 的方程组

$$\begin{aligned} M &= \ln \frac{1}{1 + \beta N} + \beta M + \beta N \ln \beta N - \beta N \ln(1 + \beta N), \\ N &= \alpha + \alpha \beta N. \end{aligned}$$

求解得到

$$\begin{aligned} M &= \frac{1}{1 - \beta} \left[\ln(1 - \alpha\beta) + \frac{\alpha\beta}{1 - \alpha\beta} \ln \alpha\beta \right], \\ N &= \frac{\alpha}{1 - \alpha\beta}. \end{aligned}$$

将 M, N 的取值带回一阶条件, 我们就得到了此时无穷期问题的政策函数

$$k' = \alpha\beta k^\alpha, \tag{3.9}$$

$$c = (1 - \alpha\beta)k^\alpha. \tag{3.10}$$

注意到(3.9)与有限期问题的极限(2.20)是一致的。这就完成了我们对整个问题的求解。这里还要提醒读者的是, 尽管在上面的计算中大量的工作是与 $V(k)$ 的求解相伴的, 但是我们的目的并不是求解 $V(k)$ 本身, 我们的目的在于求解消费与储蓄关于本期资本存量的函数(3.9)和(3.10), $V(k)$ 的求解只是手段。容

作者在此不厌其烦的啰嗦一下，在求解一个问题，特别是动态优化这种需要大量计算步骤的问题时，请读者务必时时提醒自己每一步计算的目标以及对于问题的求解最终的目的是什么，切勿在技术细节当中迷失。

最优化问题(3.5)还有一些可以进一步分析的地方。首先，对应到序列形式的最优化问题当中我们所遇到的困难，当时我们缺少一个定义了 k_{T+1} 与 k_T 之间关系的独立方程。在有限期问题当中这是由 $k_{T+1} = 0$ 提供的，在递归形式的无穷期问题中，这个方程是由泛函方程(3.8)给出的。注意到该方程实际上是通过求解 M, N 来获得函数 $V(k)$ 的表达式。其次，这里的泛函方程的求解能够简化成为通过待定系数法求解两个参数，其核心在于我们通过一些诊断性的试验猜想值函数 $V(k)$ 分类上属于对数线性函数，而这极大的简化了我们所需求解的变量总数。如果函数的分类上的形式是未知的，比如通过几次诊断性的迭代，我们无法归纳出 $V(k)$ 到底是对数函数、指数函数、多项式函数等等，甚至于 $V(k)$ 本身不存在能够用基本函数书写的解析表达式，那么泛函方程(3.8)的求解就是非常困难的。事实上，能够应用上述待定系数法求解的模型只有三种，分别为：对数效用函数与柯布一道格拉斯约束，CRRA（相对风险厌恶不变）效用函数与线性约束，以及二次型偏好与线性约束。显然我们不希望我们回答的问题被这三类函数形式限制住，因此，我们需要一种更加一般的方法来对动态优化问题(3.4)进行求解，这就是我们在下一小节当中要引入的值函数迭代法（Value Function Iteration）。第三，对很多优化问题来说，无穷期问题的解确实等于有限期问题 $T \rightarrow \infty$ 时的极限，但是这并不绝对。我们可以证明的是，当(3.4)定义了一个压缩映射时，这个结论是成立的。下一小节当中值函数迭代法的全局收敛性的证明便依赖于(3.4)是压缩映射。幸运的是，宏观经济学当中处理的很多问题都是满足这个性质的，大多数情况下值函数迭代法的求解是稳健的。读者在下一节中将会看到，值函数迭代法技术上的实现，实际上就是在求解有限期问题的极限。但是反例也同样存在，比如存在时间不一致性（Time Inconsistency）与递归合约（Recursive Contract）时的问题很多就不满足这个性质，关于这些前沿问题的讨论在本教程中我们不涉及，这里只提请读者注意。关于压缩映射原理的讨论在此从略，有兴趣的读者可以参阅[Stokey and Lucas \(1989\)](#)第三章中的相关讨论。

3.2 值函数迭代法

通过上一小节例子3.1中的讨论我们知道了，递归问题(3.4)中的泛函方程

$$V(k) = \max_{\{c, k'\} \in \Gamma(k)} \{u(c) + \beta V(k')\}$$

为我们提供了求解动态优化问题所需的那个“额外”的方程。同时，我们知道求解上述问题的核心在于对 $V(k)$ 进行刻画，并且除了几种特殊的情形之外，一般而言该问题不存在解析解。本节当中，我们介绍一种常用的求解 $V(k)$ 的方法，即值函数迭代法。

3.2.1 用函数表描述函数性质

首先我们来看如何对一个函数进行刻画。我们比较熟悉的对函数进行描述的方法是使用一个已知的表达式进行刻画，比如我们知道余弦函数的表达式是

$$y = \cos x,$$

这个表达式是说对于 x 的任何一个取值，我们都能够通过余弦函数的定义计算出相对应的 y 的值。所以这里的关键其实是对于任何一个 x ，我们都能够通过一定的方法找到对应的 y ，余弦表达式的作用只是

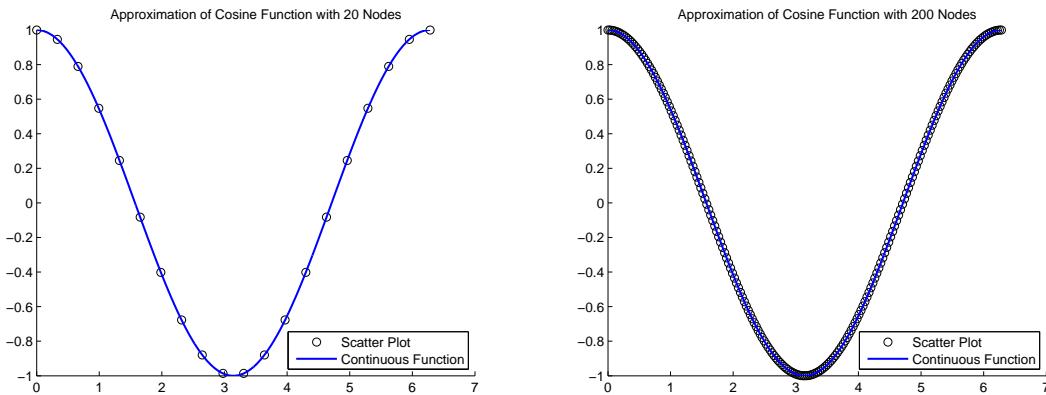


图 3.1.—使用数值解描述余弦函数

给我们提供了一种计算 y 的方法。也就是说，对于一个函数，如果我们能够知道对于每一个 x 的取值所对应的 y 的值，那么就算我们不知道 x 与 y 之间的具体表达式，我们也同样可以分析函数 $y = f(x)$ 的性质。特别的，对于后面的这种使用点对点的方式对函数进行描述，我们称为函数的数值解（*Numerical Solution*）。还是以余弦函数为例，在图3.1的两个小图当中，余弦函数用实线表示，左图中我们用20个点对余弦函数进行近似，右图中我们使用200个点。两者对比我们可以看出，当使用足够多的节点的时候，函数的数值解能够对由解析解获得的精确表达式进行非常好的近似。在 $V(k)$ 的求解中，我们就将使用基于节点（node）的数值解进行描述。具体来说，我们要找的，就是对于一组足够稠密的关于 k 的节点 $\{k_1, k_2, \dots, k_N\}$ ，对应的 $\{V(k_1), V(k_2), \dots, V(k_N)\}$ 是什么。图3.1的例子显示，当 N 足够大的时候，这些离散的 $\{k_i, V(k_i)\}_{i=1}^N$ 偶对可以很好对真实的 $V(k)$ 进行近似，这些偶对有时候也被称为函数表（*Function Table*）。在一般化叙述值函数迭代算法之前，我们先用一个简单的可以笔算的例子给读者一点直观上的感受。

例子3.2. 用数值方法求解例子3.1。特别的，设此时 $\beta = 0.96$, $\alpha = 1/3$ 。

为节省篇幅，例子3.1的函数形式就不在这里重复了。注意到我们的目的此时依然是求解政策函数，对于 $V(k)$ 的求解还是作为中间步骤进行。与例子3.1中不同的是，现在我们假设函数 $V(k)$ 是由函数表 $\{k_i, V(k_i)\}_{i=1}^N$ 刻画的。此时对于 $V(k)$ 的求解，指的是求解出一张满足贝尔曼方程的函数表。在这个简单的例子里，我们只用四个点来描述 $V(k)$ 。这当然是不精确的，但是我们此时的目的并不是求出精确解。同时这个不精确的近似还能够向我们指出一些数值方法当中所需要注意的问题。另外，由于 $V(k)$ 只在四个点上有定义，所以选择变量 k' 的取值也只有四个。我们在第六章当中将考虑线性插值来允许选择变量取更多的值，但是当下让我们把问题处理的简单一些。所以此时，任给本期的状态变量 k_i 的取值，选择变量 k'_i 的四个取值中哪个是最优的，可以用穷举法来计算。即我们把每种选择下对应的消费者效用都计算出来，然后把最大的那个挑出来，作为最优选择。注意到在例子3.1当中我们是通过具体求解一个最优化问题来完成的这个步骤。确定了最优选择之后，我们跟之前一样把最优选择带回贝尔曼方程，这就可以计算出新一轮迭代的值函数取值。

首先，我们用 $\mathbf{k} = (0.5, 3, 6, 9)$ 对状态空间（即函数的定义域）进行近似，也就是说我们假设 k 只能在这四个点上取值， $k_1 = 0.5, k_2 = 3, k_3 = 6, k_4 = 9$ 。我们依然保持在例子3.1当中使用的记号，即 $V_i(k)$ 的下标*i*用来标识第几次迭代。与之前的情况保持一致，我们设 $V_0 = \mathbf{0}$ （注意这是一个具有四个元素的向量）。注意到 V_0 的初始值的取值是任意的。可以证明，如果一个优化问题定义了一个压缩映

射，那么任意初始值开始的迭代最后都会收敛到函数真值，区别只是算法收敛的速度。本章中分析的所有问题都满足压缩映射原理，所以初始循环的赋值是任意的，不过对一般的问题来说这需要具体分析。当压缩映射的性质不满足时，不同的初始值有可能会收敛到不同的局部解，这里提醒读者注意。将 $V_0 = \mathbf{0}$ 带回优化问题，我们得到 $\forall i$,

$$\begin{aligned} V_1(k_i) &= \max_{k'_i} \{\ln(k_i^\alpha - k'_i) + \beta V_0(k'_i)\} \\ s.t. \\ k'_i &\in \mathbf{k}. \end{aligned}$$

注意到上式当中我们直接用资源约束消去了消费函数。为了刻画 $V_1(k_i)$ ，现在我们要对每一个 k_i ，通过穷举法计算其对应的最优选择。

先看 $k_1 = 0.5$ ，此时若 $k' = 0.5$ ，则消费者得到效用 $V^t = \ln(0.5^{1/3} - 0.5) + 0.96 \times 0 = -1.2252$ （我们用上标 t 表示临时变量）。而对 k' 的其他三个取值，因为 $0.5^{1/3} - k'_i < 0$ ，所以效用都设为负无穷。因为 $-1.2252 > -\infty$ ，所以 $k' = 0.5$ 在四个可能的取值当中给出的效用是最高的。于是此时得到本期资本为 k_1 时，储蓄也为 0.5。我们对 k_2, k_3, k_4 做同样的计算，可以求出本期资本为 k_2, k_3, k_4 时的储蓄函数，特别的，他们都为 $k'(k_i) = 0.5$ 。此时，我们可以更新值函数的迭代值。这是通过对每一个 k_i 的取值，带入选择变量的最优解 $k' = k'(k_i)$ 得到的。具体而言，这里我们有

$$V_1(k_i) = \ln(k_i^{1/3} - 0.5) + 0.96 \times 0.$$

注意到这里 $\ln(k_i^{1/3} - 0.5)$ 中的 0.5 是对应于 k_i 的最优储蓄。一般而言，如果我们的节点取的足够稠密，这里的取值对于不同的 k_i 是不同的。在这个例子当中他们都取 0.5，仅仅是由于我们取了很粗糙的近似而已。上述计算得到

$$V_1 = (-1.2252, -0.0595, 0.2754, 0.4575),$$

至此，我们便完成了一次迭代。

利用 $V_1(k_i)$ ，我们开始新一轮的迭代计算 V_2 。还是先看本期 $k_1 = 0.5$ ，此时若 $k' = 0.5$ ，则消费者得到效用 $V^t = \ln(0.5^{1/3} - 0.5) + 0.96 \times V_1(k') = -2.4014$ 。而对 k' 的其他三个取值，我们依然有 $0.5^{1/3} - k'_i < 0$ ，所以 $k' = 0.5$ 。同样可以验证，对 k_2, k_3, k_4 储蓄函数也都为 $k'(k_i) = 0.5$ 。此时，我们使用

$$V_2(k_i) = \ln(k_i^{1/3} - 0.5) + 0.96 \times V_1(0.5),$$

对值函数的取值进行更新，得到

$$V_2 = (-2.4014, -1.2357, -0.9007, -0.7187),$$

这完成了新一期的迭代。依次类推我们可以计算 V_3, V_4, \dots 。我们用 $\delta_t = \max |V_{t+1} - V_t|$ 来近似两次迭代之后值函数的差异。特别的，在这个例子中 $\delta_1 = 1.1762, \delta_2 = 1.1291, \delta_3 = 1.0840, \delta_4 = 1.0406$ ，构成一个单减序列。当我们进行足够多次的迭代之后，可以发现 $\lim_{t \rightarrow \infty} \delta_t = 0$ 。实际操作当中，我们设当 δ_t 小于一个足够小的 ε 之后，算法收敛，并取最后一次迭代得到的 $V_t(k)$ 为对 $V(k)$ 真值的近似。应用 $V_t(k)$ （再次提醒读者这是一个四个点组成的向量），我们可以求出此时政策函数的数值解

$$k' = 0.5, \tag{3.11}$$

$$c = (0.2937, 0.9422, 1.3171, 1.5801). \tag{3.12}$$

现在我们把数值解(3.11)与(2.20)中的解析解进行比较。对此时状态空间取的四个节点而言，其解析解下对应的储蓄水平分别为

$$\mathbf{k}' = (0.2540, 0.4615, 0.5815, 0.6656),$$

而数值解都取第一个节点 $k'_1 = 0.5$ 。这是因为对我们所考虑的状态空间而言，所允许的最大资本存量 k_4 下，对应的储蓄水平与第二个节点 $k'_2 = 3$ 之间的距离也远大于其与第一个节点 $k'_1 = 0.5$ 的距离。故而数值近似下，储蓄水平都取 $k'_1 = 0.5$ 。事实上，读者可以验证，若我们设 $k_2 = 0.6$ ，那么最优储蓄函数便不再是一个常量。这显示了要得到足够精确的数值解，状态空间的取点必须足够的稠密，当然这意味着更大的计算量与更长的计算时间，所以实践当中这里需要读者反复权衡。

3.2.2 算法实现

现在我们把上一小节当中直观的例子归纳成1.3节中用伪码表达的算法，同时复习一下“自顶而下”的程序设计思想在一个稍微复杂一些的问题中的应用，之后我们求解例子3.2的较高精度的数值解并分析代码。

此时，我们要回答的经济学问题是一个无限期的消费储蓄模型中储蓄率是如何决定的，这个问题的外生给定的是刻画了消费者偏好与企业生产技术的函数参数，模型当中内生求解出来的是刻画了消费与储蓄决策的关于本期资本存量的函数 $c = P_c(k)$ 与 $k' = P_k(k)$ ，这里我们用大写的 P 代表政策 policy。这两个函数是以数值方式描述的，所以我们实际上求的是两张由两个向量组成的函数表。函数表中自变量取值 k_i 是对于状态空间（实数轴）的近似，为外生给定，需要求解的是对于每个状态空间上的节点 k_i ，消费与储蓄的取值 $c_i = P_c(k_i)$ 和 $k'_i = P_k(k_i)$ 。这个问题在数学上对应了一个无穷期的序列问题，上一小节当中我们知道该问题可以转化成为一个动态优化问题(3.5)进行求解。优化问题(3.5)求解的关键是找到 $V(k)$ 数值上的近似，我们这里采用值函数迭代算法来求解 $V(k)$ 的近似解。有了 $V(k)$ 的数值解之后，我们就可以应用数值方法求解(3.5)所定义的最优化问题，注意到此时 $V(k)$ 已经是已知的，在这一步我们求解的是一个我们所熟悉的、已知函数形式的优化问题。当 $V(k)$ 有解析表达式的时候，我们一般使用拉格朗日函数法求解（如我们在例子2.2中所做的那样）。这时候的情况稍有不同的是我们没有 $V(k)$ 的解析表达式，所以不能使用传统的一阶条件方法来求解。这里我们采用穷举法来求解这个最优化问题。^①

具体描述算法之前，我们先用一个定理来叙述针对例子3.1的值函数迭代算法的理论基础。对于一般的动态优化问题的叙述请读者参阅Stokey and Lucas (1989)第三章关于压缩映射原理和Blackwell充分条件的叙述，这里从略。

定理3.1. (值函数迭代) 考虑一个最优化问题

$$\begin{aligned} V(k) &= \max_{c, k'} \{\ln c + \beta V(k')\} \\ s.t. \\ c + k' &= k^\alpha. \end{aligned}$$

^① 当然我们也可以采取数值微分的方法来计算一阶条件，不过在这里我们不过多引入新的内容，有兴趣的读者请参阅数值计算相关的文献。

对任给的 $V_0(k)$, 若定义函数序列 $V_t(k)$ 为

$$\begin{aligned} V_{t+1}(k) &= \max_{c,k'} \{\ln c + \beta V_t(k')\} \\ s.t. \\ c + k' &= k^\alpha, \end{aligned} \tag{3.13}$$

那么可以证明 $V_t(k)$ 一致收敛到 $V(k)$

$$\lim_{t \rightarrow \infty} V_t(k) = V(k).$$

这里证明从略。不过关于该定理，有两点提请读者留意。第一，尽管我们只叙述了一个针对例子3.1中函数形式的版本，但是定理3.1对于很多宏观经济学当中常用的问题都是成立的。第二，注意到我们在上一小节当中的计算实际上就是对一个任取的 $V_0(k)$ (零向量)，反复的利用(3.13)构造函数序列 $\{V_t(k)\}_{t=1}^\infty$ 。此外，如果读者将 $V_0(k) = \mathbf{0}$ 看成是有限期问题中终末期效用为零，那么这个算法的实质就是在用一个时间足够长的有限期问题的初始期决策（如果将 $V_t(k)$ 的下标考虑成距离终末期的时间长度）来近似无穷期问题的解。直观上这就是说如果我们需要对现有的资源在比如一万年的时间中的使用进行规划，那么实质上规划出来的效果，至少在短期内，与我们考虑将这些资源应用在无穷期当中类似的，但是随着消费者越发接近一万年的结尾，消费者的规划会越发的“短视”，从而造成与无穷期问题更大的差异。当 $V_0(k)$ 取其他向量时，可以解释为存在遗产动机。所以定理3.1的另一个解读方法是说此时无穷期规划问题是有限期问题关于时间的极限。

下面我们将算法写成伪码。该代码的目的是求解(3.5)，解的定义是对每个节点 $\{k_i\}_{i=1}^\infty$ ，计算出对应的 c_i 与 k'_i 。为此，我们采用以下步骤：

1. 将状态空间离散化，即取向量 $\mathbf{k} = (k_1, k_2, \dots, k_N)$ 对 $V(k)$ 的自变量取值集合进行近似。
2. 使用值函数迭代法求解 $V(k)$ ，特别的，这包括下列步骤：

(a) 任取初始变量 $V_0(k)$ ，定义优化问题

$$\begin{aligned} V_1(k) &= \max_{c,k'} \{\ln c + \beta V_0(k')\} \\ s.t. \\ c + k' &= k^\alpha. \end{aligned}$$

(b) 对每一个节点 k_i ，计算对 $k'_j \in \mathbf{k}$ 的每一个取值， $\ln c_{ij} + \beta V_0(k'_j)$ 的值，记为 V_{ij} 其中

$$c_{ij} = k_i^\alpha - k'_j.$$

注意这里我们要求 k_i 与 k'_j 从同一个集合，即 \mathbf{k} 当中取值。 V_{ij} 记录的是，若本期资本存量为 k_i ，那么当储蓄决策为 k'_j 时，消费者的效用（本期加后续期）。这一步当中 k_i 保持不变，但是 k'_j 的取值遍历了 \mathbf{k} 。

- (c) 在 $(V_{ij})_{j=1}^N$ (向量) 的所有元素中选取最大元，记其指标是 m ，那么此时对于节点 k_i 而言，最优的储蓄决策就是 k'_m 。此时 $V_1(k_i) = V_{im}$ 。
- (d) 当(b)、(c)中的步骤对每一个 k_i 都计算过之后，我们就计算出了新的 $V_1(k)$ 。这一步当中 k_i 遍历了 \mathbf{k} 。

(e) 新的 $V_1(k)$ 可以使用同样的步骤计算 $V_2(k)$ 。一般的，对于任意的 t ，重复(b)至(d)，计算 $V_{t+1}(k)$ 。

(f) 定义 $\delta_t = \max |V_t - V_{t+1}|$ ，当 δ_t 足够小，即对于一个任取的 ε 有 $\delta_t < \varepsilon$ ，我们判断函数列 $\{V_t(k)\}$ 已经收敛，停止(e)中的迭代。^① 此时， $V_{t+1}(k)$ 为真实解 $V(k)$ 提供了良好的近似。这里需要提醒读者注意的是， ε 度量的是最后两次迭代时函数的差别，并不是 V_{t+1} 与函数真值之间的差别，后者当 $V(k)$ 的真值无法解析刻画时是不确定的（参考数学分析中关于数列的柯西收敛准则的讨论）。

3. 对于上一步最后得到的 $V_{t+1}(k)$ 应用步骤2.(b)与2.(c)，求解最优政策。具体来说，对每一个 k_i ，计算对 $k'_j \in \mathbf{k}$ 的每一个取值， $\ln c_{ij} + \beta V_{t+1}(k'_j)$ 的值，记为 V_{ij} 。其中

$$c_{ij} = k_i^\alpha - k'_j.$$

在 $(V_{ij})_{j=1}^N$ （向量）的所有元素中中选取最大元，记其指标是 m ，那么此时对于节点 k_i 而言，最优的储蓄决策就是 k'_m 。最优消费决策由

$$c_i = k_i^\alpha - k'_m.$$

给出。当 k_i 遍历了 \mathbf{k} 时，我们就得到了算法求解的目标，即函数表 $(k_i, k'_i)_{i=1}^N$ 与 $(k_i, c_i)_{i=1}^N$ 。

上述伪码转化的MATLAB代码包含在代码3.1当中。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code solves the dynamic programming problem with logarithmic
7 % utility function and Cobb-Douglas constraint.
8 % - Code 3.1 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Fall 2015
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 12/14/2015: Original Code
17
18 clear all ;
19 %
20 % DATA DICTIONARY
21 %
22 % Parameterizations
23 alpha = 0.3 ;
24 beta = 0.96 ;
25 N = 200 ;           % Number of Grids
26 tolv = 1e-9 ;       % Tolerance level
27 tol_n = 2500 ;      % Maximum Number of Loops in While-loop,
28                      % to avoid never-ending dead-loop
29 % Variables
30 V0 = zeros(1,N) ;    % Vt
31 V1 = zeros(1,N) ;    % Vt+1
32 VT = zeros(1,N) ;    % Vij

```

^① 实践中 ε 通常取 $\varepsilon \leq 1 \times 10^{-9}$ 。

```

33 polc = zeros(1,N) ; % Policy Function for Consumption
34 polk = zeros(1,N) ; % Policy Function for Saving
35
36 % Auxillary Variables
37 optim = -1 ; % Store the index of optimal choice
38 eps = 1 ; % Evaluate convergence of Vt
39 loopn = 1 ; % Calculate the number of loops
40
41 % Calculate the Steady State
42 kss = (1/(alpha*beta))^(1/(alpha-1)) ;
43 kmin = 0.8*kss;
44 kmax = 1.2*kss;
45 % Discretize the state space
46 kgrid = linspace(kmin, kmax, N);
47
48 % Calculate the consumption level when current capital is ki and
49 % future captial is kj
50 c = zeros(N,N);
51 for indi = 1:1:N
52     c(indi,:) = kgrid(indi)^alpha - kgrid ;
53 end
54 % Avoid evaluating negative consumption in log-utility
55 c(c <= 0) = 1e-11;
56
57 % -----
58 % VALUE FUNCTION ITERATION
59 % -----
60 % Iterate Vt until convergence
61 while (eps > tolv) && (loopn < tol)
62     for indi = 1:1:N
63         VT = log(c(indi,:)) + beta*V0 ;
64         V1(indi) = max(VT);
65     end
66     eps = max(abs(V1-V0)) ;
67     V0 = V1 ;
68     loopn = loopn + 1;
69 end
70
71 % -----
72 % OPTIMAL POLICIES
73 % -----
74 % Using the numerical approximation of V to solve the optimization problem
75 for indi = 1:1:N
76     VT = log(c(indi,:)) + beta*V1 ;
77     % Find the saving that leads to the highest utility
78     [xtemp optim] = max(VT) ;
79     polk(indi) = kgrid(optim) ;
80 end
81 polc = kgrid.^alpha - polk ;
82
83 % Calculate the Analytical Solution
84 kgrid_a = linspace(kmin, kmax*1.1, N*10);
85 polk_a = alpha*beta*(kgrid_a.^alpha);
86 polc_a = (kgrid_a.^alpha) - polk_a ;
87
88 % Compare the numerical solution with the analytical solution
89 figure(1)
90 hold on;
91 scatter(kgrid,polk,'k') ;
92 plot(kgrid_a,polk_a,'LineWidth',1.5) ;

```

```

93 legend('Numerical','Analytical','Location','SouthEast')
94 title('Numerical Solution of Saving')
95 xlabel('Capital') ;
96 ylabel('Next Period Capital')
97 xlim([0.14 0.21])
98 hold off;
99
100 figure(2)
101 hold on;
102 scatter(kgrid,polc,'k') ;
103 plot(kgrid_a,polc_a,'LineWidth',1.5) ;
104 legend('Numerical','Analytical','Location','SouthEast')
105 title('Numerical Solution of Consumption')
106 xlabel('Capital') ;
107 ylabel('Consumption')
108 xlim([0.14 0.21])
109 hold off;

```

代码 3.1.—值函数迭代法求解动态优化问题

代码3.1的内容与伪码中的陈述相同，这里不作逐行描述，作者再次只提请读者注意几个重要的问题。第一，与解析解不同，数值解永远是定义在某一段特定的自变量取值范围内的。比如本例当中，原问题的定义域是整个实数域，但是显然在实际计算的时候是不可能穷尽整个实数域的，所以通常情况下我们只计算所需要近似的函数与经济分析相关的那一部分的性质。如41-46行所示，在本例当中，我们选取的是稳态附近的一段区间进行近似。这里的稳态是在欧拉方程当中带入 $c_t = c_{t+1}$ 得到的。第二，在55行处我们将所有的负消费赋值为一个很小的实数 1×10^{-11} 。这是一个技术处理，为了防止在后续程序当中计算对数效用函数时出现调试错误。读者在学习的过程中将会看到，数值解相对于解析解，在很多技术的细节上需要进行处理以防止出现预期外的错误（*Unexpected Error*）。这种错误是很难穷尽的，很多情况下需要大量的编程经验进行指引，因此这就需要读者不断的学习当中慢慢体会。另外，这里我们应用了MATLAB对于向量进行逻辑运算的固有函数，55行程序的 $c(c \leq 0)$ 中，括号内的 $c \leq 0$ 在MATLAB中的含义是“找到 c 中所有小于等于零的元素，将它们的坐标赋值为1，对其余元素的坐标赋值为零”。比如，对向量 $c = (3, -1, 4, 2)$ ，命令 $c \leq 0$ 会返回向量 $(0, 1, 0, 0)$ 。之后， $c(c \leq 0)$ 会选定矩阵 c 当中坐标处取值为1的元素。之前的例子中，对向量 $c = (3, -1, 4, 2)$ 使用命令 $c(c \leq 0)$ 会返回-1。语句55也可以通过一个条件分支语句搭配计数循环来完成：

```

for indi = 1:1:N
    for indj = 1:1:N
        if c(indi,indj) < 0
            c(indi, indj) = 1e-11 ;
        end
    end
end

```

第三，在61-69行我们使用了一种新的循环——条件循环。条件循环大体的作用与计数循环相同，只不过此时循环体执行的判断条件从计数循环中的固定次数变为由一组逻辑条件来判断，当61行中的逻辑运算为真时，执行循环体语句62-68。注意到计数循环是一种特殊的条件循环。事实上，条件循环在使用的时候，为了防止出现程序逻辑错误而导致的死循环，通常会增加一个计数器（语句68），当循环运行的次数过多时跳出运算，提醒程序员对程序的逻辑进行检查。本程序中，条件循环最后被执行了507次。另外，在语句61中我们应用了逻辑“与”运算符 $\&\&$ 。表3.1当中归纳了常用的逻辑运算符与二元

表 3.1
逻辑运算符与关系运算符

| 二元关系 | 运算符 | 逻辑关系 [†] | 运算符 |
|------|-----|-------------------|-----|
| 大于 | > | 与 | & |
| 小于 | < | 或 | |
| 等于 | == | 非 | ~ |
| 大于等于 | >= | | |
| 小于等于 | <= | | |
| 不等于 | ~= | | |

[†] 在新版本的MATLAB当中，&和|还有改进的版本&&和||。它们与表中逻辑运算符的区别是，对A && B，程序会判断B是否为真，当且仅当A为真。反过来，对A || B，程序仅当A为假时才会判断B是否为真。也就是说，如果整个表达式的值由第一个条件就可以判断，那么程序将不判断第二个表达式的真假。当判断表达式B非常耗时的时候，这能够提高程序执行的效率。

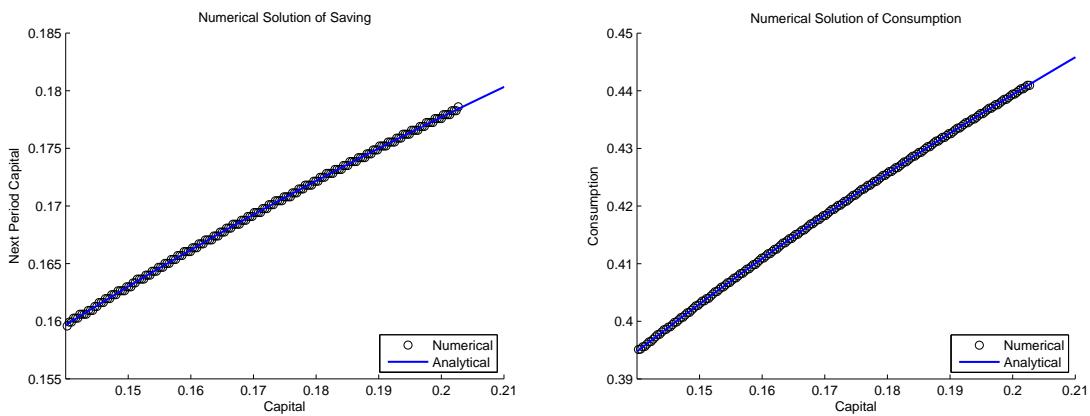


图 3.2—值函数迭代法求解动态优化问题

关系运算符。运行程序，结果在图3.2中显示，其中左图对比了储蓄函数的数值解与解析解，右图对比了消费函数的数值解与解析解。两图当中，蓝色曲线是例子3.1中求出的解析表达式，黑色的散点是由200个点近似的数值解。我们可以看出，此时，当我们选取了足够稠密的点对状态空间和函数进行近似，函数的数值解能够很好的逼近解析解。

3.3 应用模型进行增长核算

尽管在第二章当中，我们说索罗模型的作用之一是可以用来为跨国比较中投资与人均收入之间的正相关关系提供一个理论上的解释，不过索罗模型的作用并不止于此，其内在蕴含了远比上述比较静态更加丰富的理论含义。本节中我们应用增长模型来具体分析美国战后以来的经济的长期增长。具体而言，本节我们将关注两个方面。首先，我们将要检验增长模型与一系列战后美国经济数据的代表性事实(*Stylized Facts*)相符合；第二，我们将应用增长模型进行增长核算(*Growth Accounting*)，以分析增长模型当中能够导致人均收入提高的要素在实证上对于增长的具体贡献。

战后以来，美国的国内生产总值的季度数据如图3.3所示，数据取自美国商务部国民收入和产出账

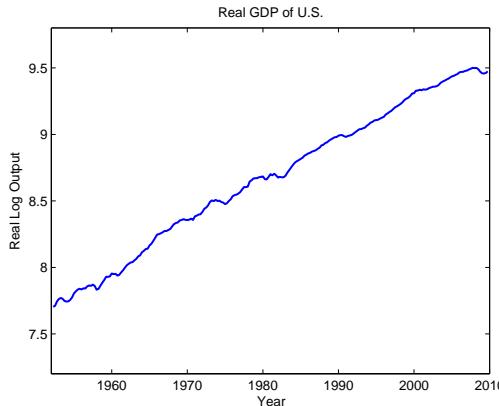


图 3.3.—战后美国国内生产总值

户(NIPA, National Income and Product Account)的表1.1.4。大体而言，尽管时有波动，美国经济维持了一个近似常数的增长率。如果我们进一步的查看NIPA中其他宏观经济变量的数据，我们会发现一个有趣的事，那就是尽管如消费、资本存量等其他宏观经济变量在战后也有显著的增长，但是很多变量与产出之间的比值是相对稳定的。因为这个发现最早是由Nicholas Kaldor观察到的[Kaldor (1957)]，所以通常在文献当中这由被称为Kaldor's Facts。这组事实的内容很丰富，不过读者马上将会看到许多内容之间是相互等价的，为节省篇幅，我们这里引用Solow (1970)所做的归纳如下：

1. 人均产出的增长率是一个常量。
2. 资本存量的增长率高于劳动投入的增长率。
3. 资本产出比是恒定的。
4. 资本回报率是相对稳定的。

我们在图3.4当中展示了另外几个代表性的时间序列。从上至下，我们依次发现劳动收入占总产出的比例在56%左右浮动，投资与消费关于产出的比例维持在一个比较稳定的比例，同时总劳动时间虽然体现出了增长的趋势，但是这个趋势与产出的增长相比要弱许多。我们称这种许多变量以与产出相同的速率增长的现象为平衡增长路径(Balanced Growth Path)。平衡增长路径的存在暗示我们经济当中许多变量的增长很可能是由一个或几个共同的因素推动的。下面我们就来证明存在劳动技术进步的增长模型蕴含了平衡增长路径。

考虑一个带有劳动技术进步的增长模型。设劳动生产率为 Z_t ，其增长率为 g ：

$$Z_{t+1} = (1 + g)Z_t.$$

这意味着在 t 期时，若劳动时间为 n_t ，那么有效劳动投入此时就是 $Z_t n_t$ 。若我们保持劳动供给无弹性的假设即 $n_t = 1$ ，那么此时的生产函数是

$$Y_t = F(K_t, Z_t \times 1),$$

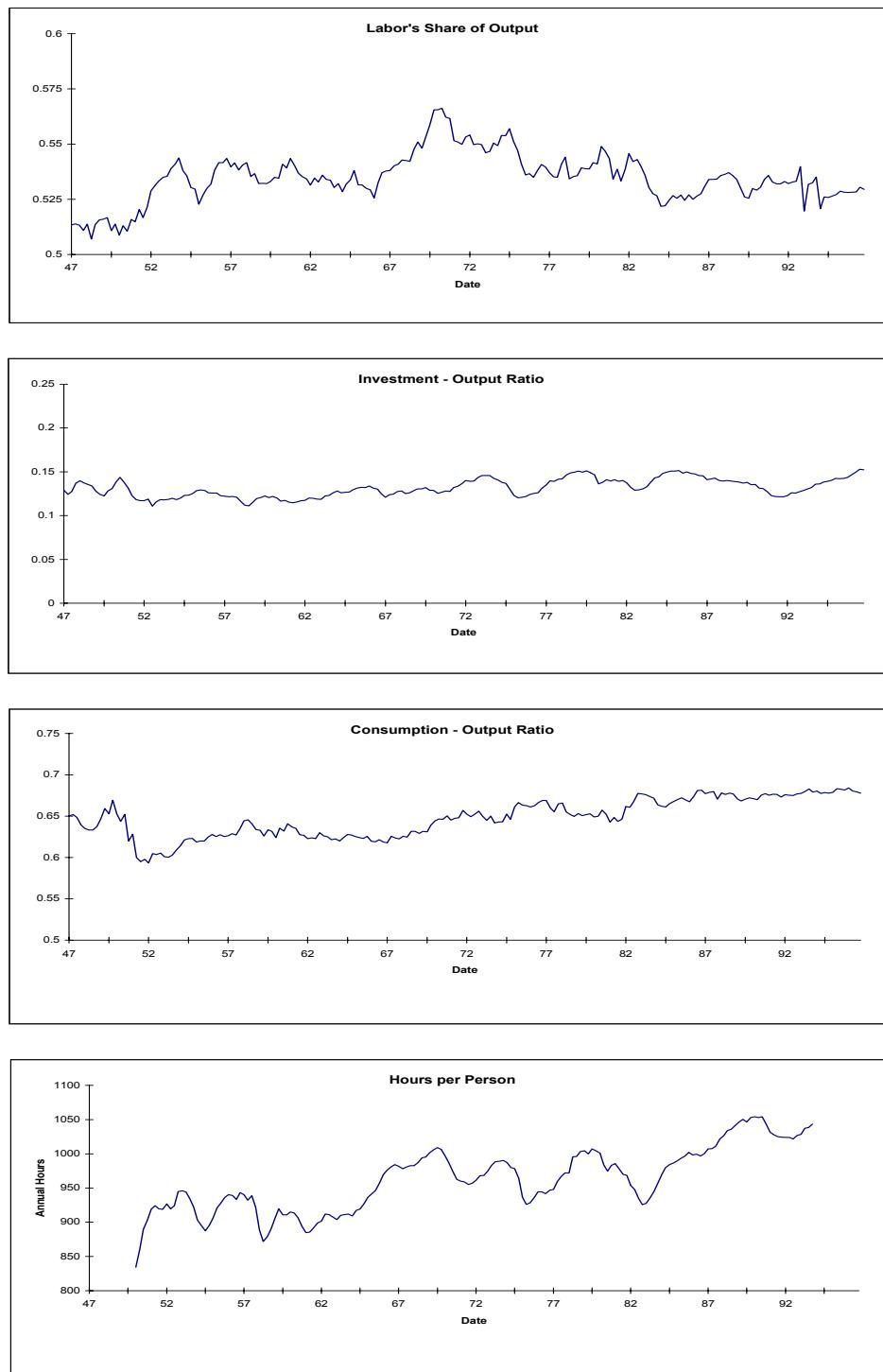


图 3.4—平衡增长路径

其中生产函数当中的1是消费者的劳动投入，同时我们假设 $F(\cdot)$ 是规模报酬不变的。这样，一个带有劳动技术进步的无穷期增长模型写作

$$\begin{aligned} & \max_{\{C_t, K_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \frac{C_t^{1-\sigma} - 1}{1-\sigma} \\ & \text{s.t.} \\ & C_t + K_{t+1} - (1-\delta)K_t = F(K_t, Z_t) \\ & K_0 \text{已知。} \end{aligned}$$

下面我们应用该模型，证明：1) C_t, I_t, Y_t, K_t 的增长率均等于 g ，即上述增长模型具有平衡增长路径；2) 资本回报率是常数。

首先记

$$\begin{aligned} C_{t+1} &= (1+g_c)C_t, \quad K_{t+1} = (1+g_k)K_t, \\ I_{t+1} &= (1+g_i)I_t, \quad Y_{t+1} = (1+g_y)Y_t, \end{aligned}$$

那么我们要证明 $g_c = g_k = g_i = g_y = g$ 。由资本积累方程，可以得到

$$\begin{aligned} K_{t+1} &= (1-\delta)K_t + I_t \\ \Rightarrow \frac{K_{t+1} - K_t}{K_t} &= -\delta + \frac{I_t}{K_t} \\ \Rightarrow g_k + \delta &= \frac{I_t}{K_t} \\ \Rightarrow I_t &= (g_k + \delta)K_t. \end{aligned}$$

将最后一式带入

$$g_i = \frac{I_{t+1} - I_t}{I_t},$$

得到

$$g_i = \frac{(g_k + \delta)(K_{t+1} - K_t)}{(g_k + \delta)K_t} = g_k,$$

即投资与资本存量的增长率相同。

再由资源约束条件得到

$$\begin{aligned} C_t + I_t &= Y_t \\ \Rightarrow \frac{C_t}{Y_t} + \frac{I_t}{Y_t} &= 1. \end{aligned}$$

注意到上式意味着 $g_c \leq g_y$ 与 $g_i \leq g_y$ 必须同时成立，否则随时间增长消费和投资会超过经济的资源总量，而这是无法达到的。同时，若 $g_c < g_y$ 与 $g_i < g_y$ ，那么长期来看经济当中将会有资源闲置，从而因与最优化问题解的性质不符被排除。两者结合，再与投资的增长率相结合，得到 $g_c = g_i = g_y = g_k$ 。

最后，由于生产函数规模报酬不变，我们有

$$Y_t = F(K_t, Z_t) \Rightarrow \frac{Y_t}{K_t} = F(1, \frac{Z_t}{K_t}).$$

因为已经证明 $g_k = g_y$, 故上式的左侧为常量。因此若要等式成立, 上式的右侧必须也是常量, 即 Z_t/K_t 为常量。因为 Z_t 的增长率为 g , 这意味了 K_t 的增长率为 g 。至此, 我们便证明了 $g_c = g_k = g_i = g_y = g$, 读者可以自行验证 Kaldor's Facts 当中与变量比相关的事。

下面我们证明资本回报率为常量。由规模报酬不变我们可以将生产函数写成

$$Y_t = K_t F(1, \frac{Z_t}{K_t}).$$

对上式关于 K_t 求导数得到

$$\begin{aligned} r_t &= \frac{\partial Y_t}{\partial K_t} = F(1, \frac{Z_t}{K_t}) + K_t \cdot \left(-\frac{Z_t}{K_t^2}\right) F_2 \\ &= F(1, \frac{Z_t}{K_t}) - \frac{Z_t}{K_t} F_2, \end{aligned}$$

其中 F_2 是 $F(\cdot)$ 对第二个变元的偏导数。因为 Z_t/K_t 为常量, 上式意味了 r_t 也是常量, 故新古典增长模型的平衡增长路径中利率是一个常量。进一步的, 应用同样的逻辑我们可以证明工资 w_t 的增长率为 g , 限于篇幅这里从略。

于是至此, 我们已经证明了新古典的增长模型的预测不仅与人均收入的跨国横截面数据相符合, 同时也与单个国家时间序列上的观测即平衡增长路径相符合。最后有三点评注。首先细心的读者可能发现, 上文的增长模型无法解释劳动投入的增长 (其被假设为常量)。这是因为大量的劳动经济学研究表明, 劳动时间的增加主要与战后女性劳动力参工率的提高有关, 与消费与闲暇替代率的变化关系较弱。第二, 索罗模型当中外生储蓄率的假设实际上是源自平衡增长路径当中 I_t/Y_t 是一个常量。最后, 这里我们假设技术进步是偏向劳动的。可以证明的是, 当生产函数是柯布一道格拉斯形式的, 无论技术进步是劳动偏向、资本偏向还是提高全要素生产率的, 均衡中的增长路径都是平衡的。但是对其他函数形式, 只有劳动偏向性技术进步的预测与平衡增长路径相吻合。关于这一点, 有兴趣的同学可以参阅 [Uzawa \(1965\)](#) 和 [Phelps \(1966\)](#)。

最后, 索罗模型的一个结构化的应用是用来进行增长核算。具体来说, 假设生产函数是柯布一道格拉斯形式的:

$$Y_t = A_t K_t^{1-\alpha} (N_t X_t)^\alpha,$$

那么应用 NIPA 当中产出、劳动投入和资本存量的数据, 我们可以计算索罗剩余 (*Solow Residual*) 为

$$\log SR_t = \log Y_t - \alpha \log N_t - (1 - \alpha) \log K_t,$$

其中

$$\log SR_t = \log A_t + \alpha \log X_t.$$

注意到尽管索罗剩余在模型的意义下的解释是技术水平, 但是因为实际上索罗剩余本身在数据当中是不可观测的, 所以它在现实经济当中的对应还是不清楚的。事实上, 关于以索罗剩余衡量的生产率差异的研究在现在还是一个非常活跃的领域。^①本质上, 索罗剩余就像回归分析当中的残差, 度量的是“我们无法解释的现象”, 这里提醒读者留意。

利用上述模型, 如果我们计算人均收入提高当中技术进步的贡献和资本存量增长的贡献之间的比例, 即

$$\log(Y_t/N_t) - \log(Y_{t-1}/N_{t-1})$$

^① 有兴趣的读者可以参考 [Hall and Jones \(1999\)](#), [Caselli \(2005\)](#), [Hsieh and Klenow \(2009\)](#) 和 [Adamopoulos and Restuccia \(2014\)](#) 等文章。

的平均值中

$$\log \text{SR}_t - \log \text{SR}_{t-1}$$

和

$$\log(K_t/N_t) - \log(K_{t-1}/N_{t-1})$$

分别占的比例，那么战后NIPA数据的估计显示前者约占2/3而后者贡献了1/3。增长核算的含义是，以一个满足平衡增长路径的模型为尺度，人均资本存量的增加对人均收入增长的贡献只是冰山一角，我们对于索罗剩余的意义需要更加深刻的理解。

第二部分

经济周期

第四章 真实经济周期理论

宏观经济的运行并不平稳，在短期内产出和就业常常会发生大幅波动。在大萧条之前，经济学家关于宏观经济的周期性波动就做了大量的工作，韦斯利·米切尔(Wesley Mitchell)、西蒙·库兹涅茨(Simon Kuznets)和弗雷德里克·米尔斯(Frederick Mills)等人整理了大量关于经济周期——即宏观经济当中众多变量同时波动的现象——的经验证据。随后经济学家对于经济周期提出了各种各样的理论解释，比如约瑟夫·熊彼特提出的读者耳熟能详的“创造性毁灭”理论。随着大萧条以及之后的凯恩斯主义革命，宏观经济学家将关注的重点放在总产出的决定上以期望在能够正确的预报(forecast)未来的产出水平，并据此设计相应的稳定性政策来阻止宏观经济再次陷入严重衰退。相当长的一段时间内，宏观经济学关于经济周期的研究从理论转向实证，从探索经济周期的理论解释转向对大型动态系统进行向量自回归与结构向量自回归模型的估计。战后三十年的宏观经济学研究的另一个特征是关于长期增长与短期波动的研究在很大程度上被割裂了。学界似乎达成了宏观经济的长期增长与短期波动需要使用两套理论分别进行分析的默契，这样的观点在当前的本科生宏观经济学教科书当中仍然占据主流地位。修读过曼昆《宏观经济学》的同学应该对长期增长中价格自由浮动而短期波动总是伴随价格粘性的设定不感到陌生，两部分所使用的分析工具，即新古典增长模型与IS-LM模型之间似乎也泾渭分明。

20世纪70年代发展起来的理性预期学派对宏观经济学的发展带来了重大的影响。^①著名的计量经济学家卢卡斯批评[Lucas (1976)]重新强调了提出一个结构化的能够解释经济波动的理论对于政策制定的重要性。^②始于Kydland and Prescott (1982) 的真实经济周期理论(*Real Business Cycle Theory*)文献在两个方面对现代宏观经济学的研究范式造成了极为深远的影响。首先，宏观经济学当中长期与短期分析不再是相互分割的两部分，而是有了一个统一的基于最优化与阿罗—德布鲁一般均衡的基本分析框架。第二，随着计算机科学的进步而带来的数值方法应用上门槛的降低，使得以校准和结构估计为标志的数量分析方法(*Quantitative Method*)（请读者注意数量分析与基于简化式回归(*Reduced-form Regression*)分析的实证方法(*Empirical Method*)之间的区别，数量分析的标志通常是一个显式的最优化模型）在宏观经济学的各个领域当中得到了非常广泛的应用。前者极大的降低了宏观经济学不同分支的学者之间交流的门槛，而后者大大的扩展了宏观经济学所能够分析的问题的疆域。因此，尽管原教旨意义下的真实经济周期理论已经渐渐褪色，但是对于真实经济周期理论的学习依然是理解现代宏观经济学极为重要的途径。而随着数量方法在经济学其他领域当中应用的普及，这还将帮助读者提高对如劳动经济学、国际贸易、实证产业组织理论等相关领域的理解。

在本教程的第二部分我们将学习真实经济周期理论模型的求解方法。本章中我们首先讨论战后美国经济周期的特征，并在上一章的卡斯—库普曼斯模型基础上构造一个适合于分析短期经济波动的一般均衡模型，这个模型于增长模型的主要区别是增加了一个生产率的一阶自回归AR(1)冲击。尽管我们在第一章的简介当中已经接触过比AR(1)更加复杂的随机过程，但是为了将AR(1)过程与不存在解析解

^①有兴趣的读者可以参考Muth (1961), Lucas (1972, 1975), Sargent and Wallace (1975)。

^②对于经济理论在经济周期分析当中的作用，见库普曼斯对于早期经济周期研究的一个批评[Koopmans (1947)]。

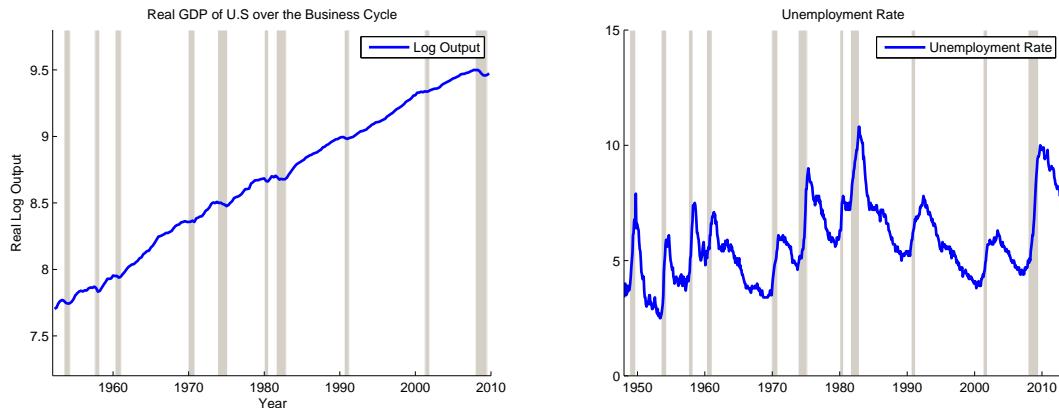


图 4.1.—战后美国经济波动

的动态优化过程相结合，我们需要新的工具。在下一章中，我们首先介绍马尔科夫过程的一些一般性质，之后我们会重点介绍对AR(1)过程进行离散化处理的计算方法[Tauchen (1986), Tauchen and Hussey (1991)]。随后，在第六章当中，我们将详细介绍如何将第五章中的离散马尔科夫过程与值函数迭代算法相结合来求解真实经济周期模型。我们将从服从独立同分布(*i.i.d*)生产率冲击的随机增长模型开始上手，随后我们将生产率冲击由*i.i.d* 改为AR(1)，最后我们将引入弹性劳动供给并分析模型解的性质，并将数值仿真结果与数据相对照。最后，在第七章当中我们将介绍另外一种求解经济周期模型常用的方法——基于稳态周围局部逼近的扰动法(*Perturbation Method*)，这里我们将使用DYNARE软件包辅助计算。

4.1 Hodrick-Prescott滤波

试图对经济周期进行分析时我们所遇到的第一个问题就是，数据而言，具体什么样的现象被我们定义为经济周期。图4.1当中包含了两个代表性的宏观经济指标的时间序列—产出（左图）与失业率（右图），其中灰色阴影部分为美国国民经济研究局(National Bureau of Economic Research, NBER)的经济周期委员会讨论决定的经济衰退的区间。^①乍看之下经济周期的定义似乎并不难回答。上一章当中我们学习了长期增长理论，如果我们认为实际经济数据是由长期趋势与短期波动组合而成的，那么从实际数据当中减去属于长期增长趋势的那部分显然就是答案。不过这么做的问题在于增长理论截止目前为止还没有得到类似的估计，所以此路暂时不通。既然不行，那么我们不妨从直觉开始，回想战后几次著名的经济衰退：两次石油危机、储贷危机、信息泡沫以及最近的金融危机，他们的共同点都是经济增长的停滞、失业率高攀、投资缩水等等，似乎这可以作为经济周期的一个备选定义。但是经济衰退的反面经济扩张又应该如何处理呢？注意到在增长理论的讨论当中我们知道战后美国经济总体来说是以强劲的势头在稳定的增长，如何将长期的增长与经济周期中过热的增长区分开，直观上答案好像就不那么显然。有些读者可能会提议将增长率高于平均增长率的时间段定义为扩张，反之则成为衰退。这当然不失为一种方法，但是线性的增长趋势似乎不能在增长核算中得到很有力的支持，同时我们知道美国经济大体的平均增长率是3.5%，很多时候当增长率是3%的时候（注意此时增长率低于平均值15%）我们并没有看到通常伴随衰退而来的种种令人担忧的现象，所以这个定义并不太令人满

^①注意到NBER并不是隶属于美国政府的政府机构，而是一个民间智库。

意。于是，我们需要用其他办法对原始数据进行处理，来分离出实际观测值当中分属于趋势(*trend*)和周期(*cycle*)的部分。如通信工程中的做法，我们这里使用滤波(*filter*)对原始数据进行处理。特别的，我们将主要介绍Hodrick-Prescott滤波[Hodrick and Prescott (1997)]。在正式介绍之前我们预先提请读者注意的是，HP滤波仅是许多具有类似功能的滤波的一种（比如卡尔曼滤波(*Kalman Filter*)，带通滤波(*Band-pass filter*)等），尽管它在宏观经济学当中的应用最为广泛，但这并不代表我们没有其他的选择。其次，滤波器对原始数据的过滤方法取决于预先设定的数据信号与噪声（这里对应了趋势与周期）的统计性质，在通信工程当中很多时候信号的统计特性是已知的，但是在经济周期的分析当中我们对于数据真实增长趋势的了解是极为有限的，所以滤波的结果在经济周期分析当中在一定程度上会受到先验假设的影响。幸运的是，经济周期数据在不同的滤波过滤之后的性质是相似的，并且在综合比照之后我们能够基本掌握过滤之后周期与趋势部分分别对应了原始数据中的哪些部分。不过由于情况并不总是如此，所以这里提请读者在实际应用滤波对数据进行前期处理的时候留意。

下面我们介绍HP滤波的概念。假设我们有 T 期的数据观测值 y_t ，它是由趋势项 g_t 和周期项 c_t 合成的：

$$y_t = g_t + c_t, \quad \forall t = 1, 2, \dots, T.$$

再定义趋势项的平滑性(*smoothness*)为其二阶差分的平方和

$$\sum_{t=1}^T [(g_t - g_{t-1}) - (g_{t-1} - g_{t-2})]^2,$$

那么若我们认为趋势项 g_t 在直觉上应该满足1)与观测值尽可能的相近和2)长期增长的变动尽可能平缓两个性质，记外生设定的平滑参数为 λ ，那么趋势项可以由下述最小化问题求解

$$\min_{\{g_t\}_{t=1}^T} \left\{ \sum_{t=1}^T (y_t - g_t)^2 + \lambda \sum_{t=3}^T [(g_t - g_{t-1}) - (g_{t-1} - g_{t-2})]^2 \right\}, \quad (4.1)$$

注意到 $y_t - g_t = c_t$ 。

方程(4.1)当中的一个重要参数是 λ ，直觉上看， λ 越大，那么相邻两个区间内经济增长率的差异对花括号内函数值的影响就越大。由于方程(4.1)定义了一个最小化问题，所以 λ 越大的时候，HP滤波所得到的趋势项的二阶差分越小，即相邻两期之间增长率的变化越小。特别的，在极限情形下，若 $\lambda \rightarrow \infty$ ，那么增长率的任何波动都会造成最小化函数的巨大变化，此时滤波得到的趋势项的二阶差分为零，也就是长期增长是线性的，意味着相邻两期之间增长率的波动完全为零。反过来，当 $\lambda = 0$ 时，趋势项 $g_t = y_t$ ，周期项 $c_t = 0$ ，经济当中观察到的所有波动都被解释为长期增长率的波动。

那么实际应用当中， λ 应做如何选择呢？ λ 能否与可以观测的真实经济数据相联系呢？答案是一定程度上可以。我们可以证明，如果周期项 c_t 和增长率二阶差分 $[(g_t - g_{t-1}) - (g_{t-1} - g_{t-2})]$ 都是独立同分布的均值为零的正态分布，其方差分别为 σ_1^2 和 σ_2^2 ，那么当 $\sqrt{\lambda} = \sigma_1/\sigma_2$ 时，所求解的 g_t^* 恰好等于 $E[g_t|y_t]$ 。换句话说，当 $\lambda \neq \sigma_1/\sigma_2$ 时，所求解出来的 g_t^* 的样本均值不等于给定 y_t 时候 g_t 的理论条件期望值，只有当 $\sqrt{\lambda} = \sigma_1/\sigma_2$ 时，两者才是相等的。所以对于增长率和周期波动相对幅度的先验假设可以用来给 λ 的取值提供经济解释。经验上，Hodrick and Prescott (1997)认为季度水平上周期项的波动幅度应该在5%上下，而增长率的波动应该是1/8个百分点左右，这意味着

$$\lambda = \frac{\sigma_1^2}{\sigma_2^2} = \left(\frac{5}{1/8} \right)^2 = 1600.$$

注意到无论是实际解等于条件期望这个标准，还是周期项与趋势项的相对波动幅度的赋值，他们都是有一定的随意性，并没有确实的理由来支持实践当中这样做，希望读者留意。不过实践证明实际滤

波的结果关于 λ 在一定的范围内是稳健的，所以尽管理论上对季度数据选取 $\lambda = 1600$ 并没有绝对客观的理由，但是在具体实践当中造成的影响并不如想象中大。[Ravn and Uhlig \(2002\)](#)建议对年度数据应用 $\lambda = 6.25$ ，对月度数据应用 $\lambda = 129,600$ 。

现在我们来分析一下HP滤波的性质。注意到方程(4.1)当中的参数就是 λ 与序列 $\{y_t\}_{t=1}^T$ ，又因为花括号内的优化函数是二次多项式，所以求解出来的趋势项的一般形式是 $\{y_t\}_{t=1}^T$ 的线性组合，权重受 λ 和 T 影响：

$$g_t = \sum_{i=1}^T \omega_{it}^T y_i.$$

进一步的，可以验证，当 $\lambda = 1600$ 且 $T \rightarrow \infty$ 时，

$$g_t = \sum_{i=-\infty}^{\infty} \omega_i^* y_{t+i},$$

其中

$$\omega_i^* = 0.8941^i [0.0562 \cos(0.01117i) + 0.0558 \sin(0.01117i)], \quad \forall i \geq 0, \quad (4.2)$$

同时 $\omega_i^* = \omega_{-i}^*, \forall i < 0$ 。^①方程(4.2)的表达式是说求解 g_t 的时候， y_t 的权重最大，而随着 i 增大， $t \pm i$ 距离 t 越来越远， $y_{t \pm i}$ 在 g_t 的计算中的权重逐渐减小，减小的幅度服从阻尼振动。这个解在直觉上的解释是对 t 的趋势项而言，距离 t 越近的观测值提供的信息越大，同时信息的衰减近似的服从一个阻尼振动。

下面我们以总产出为例来具体分析HP滤波在应用上的性质。尽管HP滤波的程序编写并不困难，但是因为针对各种语言的开放源代码版本的程序都已经经过广泛验证，并可以很方便的下载，所以本教程当中略过。^②图4.2的顶面板包括了美国战后的实际总产出与HP滤波对应的增长趋势。从中我们可以看到HP滤波趋势大体上从实际经济周期的曲线中间穿过，并且可以很明显的看到非线性的特征。进一步的，我们在图4.2的底面板中对HP滤波趋势与线性趋势进行比较。图中蓝色的实线是实际产出减去线性趋势（即增长率的均值）之后的周期项，粉色的虚线是HP滤波趋势与线性趋势之间的差。注意到线性增长对应的是一个周期“无限大”的波动（也就是没有波动），所以实际经济当中任何观察到的相对于线性趋势的偏离（无论周期多长）都被解释成周期波动。HP趋势与线性趋势之间的差，就是HP滤波趋势项当中包含的那些额外的低频趋势。从图中目测，这些低频趋势的最短周期约为8年（32）个季度。为了进一步的弄清HP滤波过滤掉的数据的特征，我们将HP滤波的周期项与另一种宏观经济学当中常用的滤波——带通滤波过滤的结果进行比较。带通滤波过滤后的周期项只包含原始数据当中周期在一定范围之内的信号，以通常的记号记，BP(6, 32)的结果就是一组时间序列数据当中所有周期在6季度至32季度之内的信号。图4.2的中面板当中比较了HP滤波的周期项与BP(6, 32)滤波的周期项。两者的相似性确认了我们之前的直觉，那就是相比于线性趋势，HP滤波的趋势项当中包含了更多的低频周期性波动，这些低频波动的周期大多在32季度以上[\[Prescott \(1986\)\]](#)。^③

所以简单来说，经过HP滤波处理以后的经济周期可以在直观上定义为经济变量时间序列数据当中周期小于32季度的那些波动。

^①物理上，傅里叶级数(4.2)定义了一个以 ω_i^* 为振幅的阻尼振动。

^②基于MATLAB 的实现可以从<http://www.mathworks.com/matlabcentral/fileexchange/3972-hodrick-prescott-filter>下载，基于Fortran的实现可以从<http://dge.repec.org/codes/prescott/hpfILTER.for>下载。

^③这里的发现与[Burns and Mitchell \(1946\)](#)使用完全不同的方法所观察到的结论是非常相似的，关于该方法的介绍与评论详见[Koopmans \(1947\)](#)。

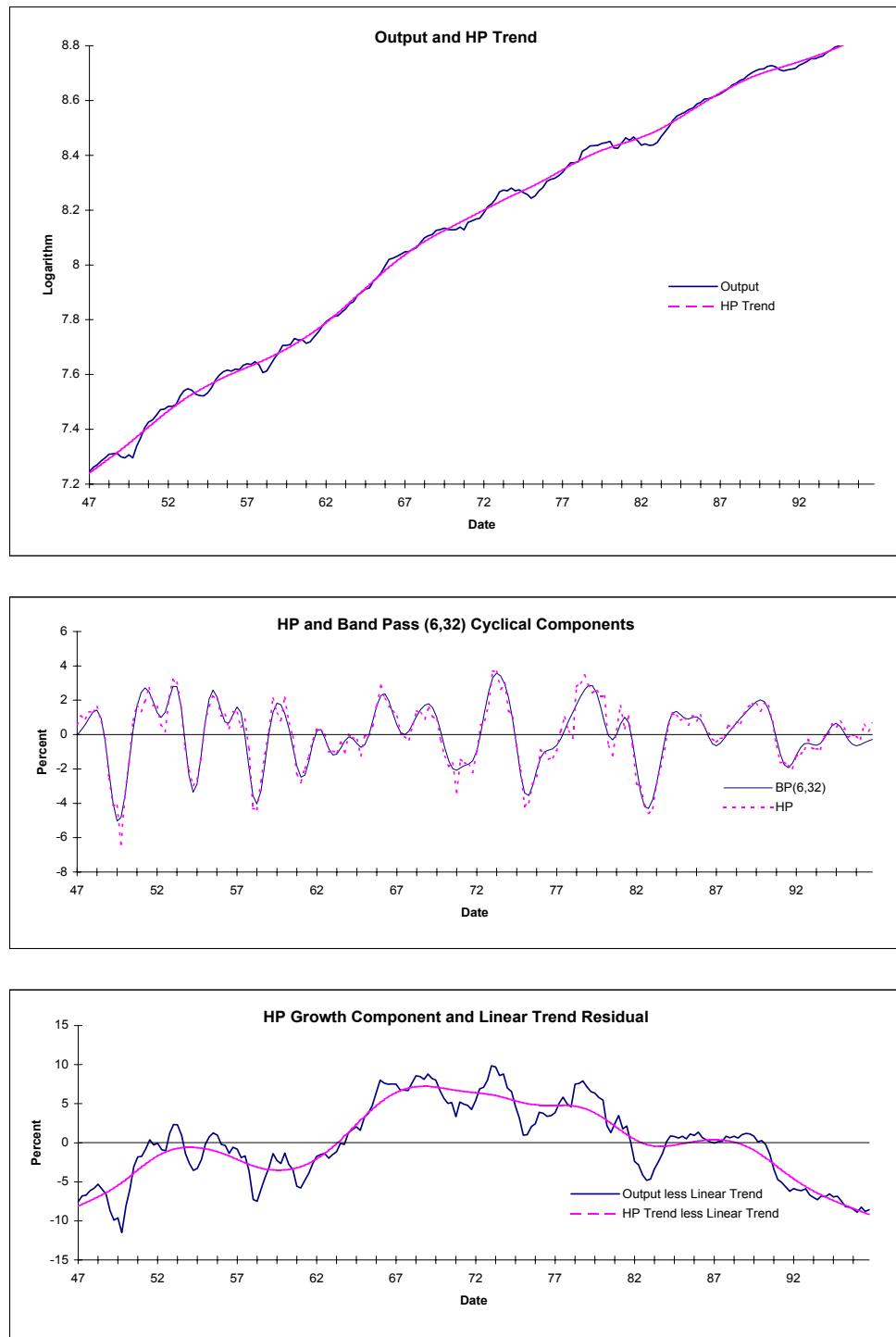


图 4.2—HP滤波的性质

4.2 战后美国经济周期的特征事实

下面我们具体分析战后美国经济周期的基本特征。这里简短的插一句，那就是尽管大量的关于经济周期的分析都是对于各种滤波处理之后的周期项进行的，但是近期宏观经济学也逐渐开始关注对长周期变化的趋势项的解释。较早的贡献见Abel (2003)和Geanakoplos, Magill and Quinzii (2004), Hansen and Ohanian (2016)是一篇最新的贡献。本小节当中所有的数据来源于NIPA 表格和美国劳动统计局(*U.S Bureau of Labor Statistics*) 的公开宏观数据，所有的时间序列的原始数据均为季度数据，周期性波动由 $\lambda = 1600$ 的HP滤波处理得到。图4.3、4.4 和4.5 当中描述了美国主要宏观经济变量的周期性波动（以百分比记），在每幅图当中总产出的周期性波动都用蓝色实线画出，以作为基准方便读者对其他宏观经济变量波动的幅度(volatility)、持续性(persistency) 及他们与产出之间的联动(comovement)进行比较。下面我们将分别对经济周期的这三个特征进行描述，在本小节当中，我们用图4.3-1指代图4.3的第一个面板。

幅度。—各个变量相对于产出波动的幅度特征可以总结如下：

- 非耐用消费品的波动低于总产出波动。（图4.3-1）
- 耐用消费品的波动高于总产出。（图4.3-2）
- 投资的波动是产出波动的三倍左右。（图4.3-3）
- 政府支出的波动低于产出波动。（图4.3-4）
- 总劳动时间的波动与产出波动的幅度大体相同。（图4.4-1）
- 资本存量的波动远低于产出的波动，但是制造业资本利用率的波动要高于产出的波动。（图4.4-2与图4.4-3）
- 总劳动雇佣的波动与产出波动幅度相似，但是人均劳动时间的波动要远低于产出波动。这意味着总工作时间的波动主要来自于外部边际即就业—失业的调整，而不是内部边际即工作时间的调整。（图4.5-1和图4.5-2）
- 人均劳动生产率的波动低于产出波动。^① （图4.5-3）
- 实际工资的波动低于产出波动。（图4.5-4）

联动与持续性。—从图4.3到图4.5可以看出大部分宏观经济变量是顺周期变化的(*procyclical*)，即他们在经济周期频率上随产出的提高而提高。另一方面，其他一些变量，比如实际工资率、政府支出和资本存量，是非周期的(*Acyclical*)。这是说他们的时间序列与产出的相关系数很小。另外我们可以看到的是，绝大多数宏观经济变量的取值在时间序列上有很强的持续性。事实上，具体的计算表明，图4.3到图4.5多数变量的一阶自回归系数都在0.7以上。值得注意的是，从图4.4-4可以看到，如上一章最后一小节定义的索罗剩余与总产出是高度相关的。

对经济周期的事实进行总结，首先我们可以看到的是对不同的经济周期之间比较，具体到每个经济周期的频率与波动幅度是没有规律的，比如有些衰退会长一些同时表现出更大的产出的下降，另外

^① 这个观察与总劳动的大幅波动放在一起，在文献当中又被称为Shimer's Volatility Puzzle，即较小的劳动生产率的波动在现有中的主流分析框架当中没法导致大幅度的总劳动供给的波动。这个问题在本讲义当中我们不涉及，有兴趣的同学可以参考Shimer (2005) 和Elsby, Michaels and Ratner (2015)。

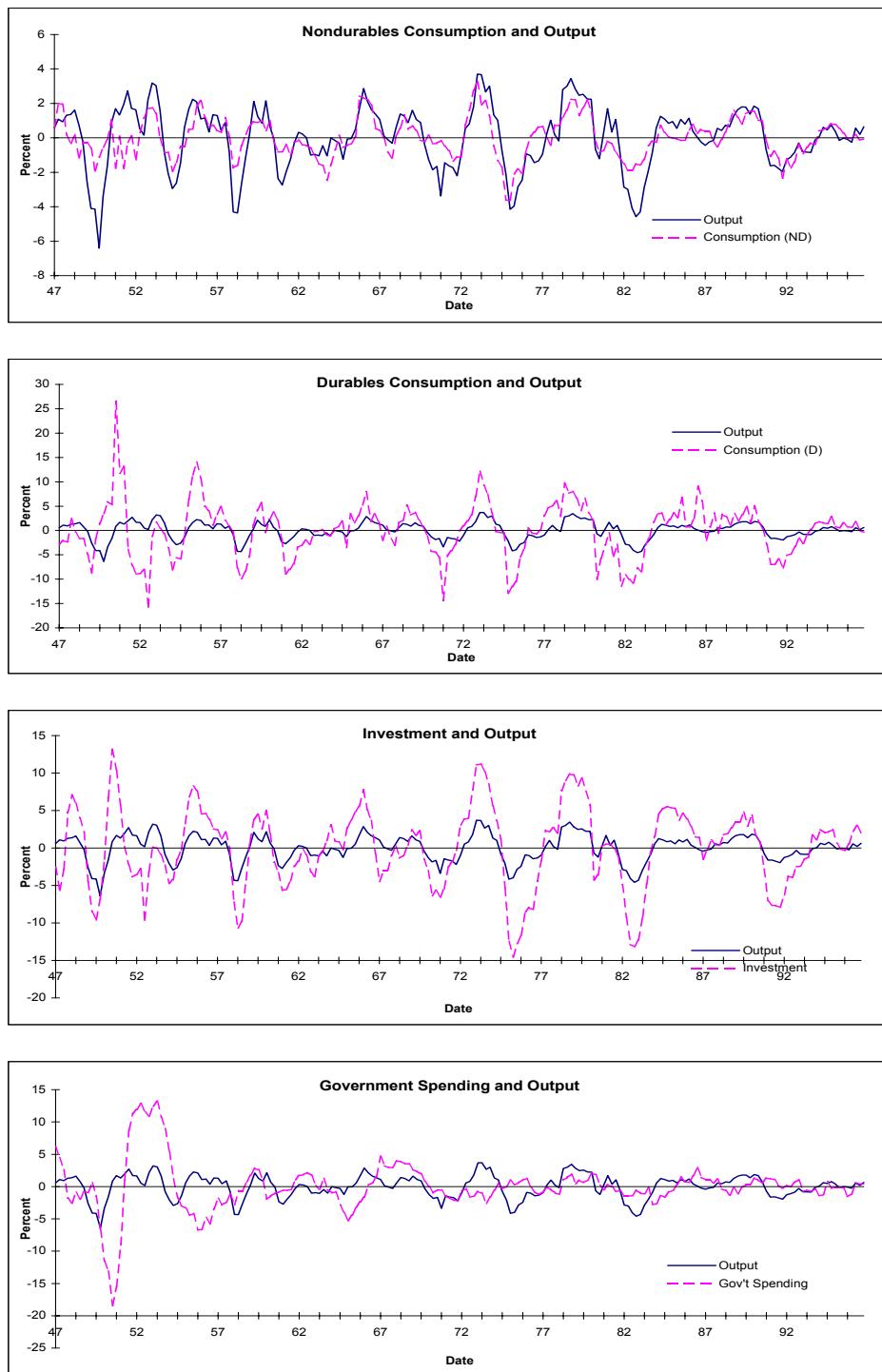


图 4.3.—战后美国经济周期波动 • 一

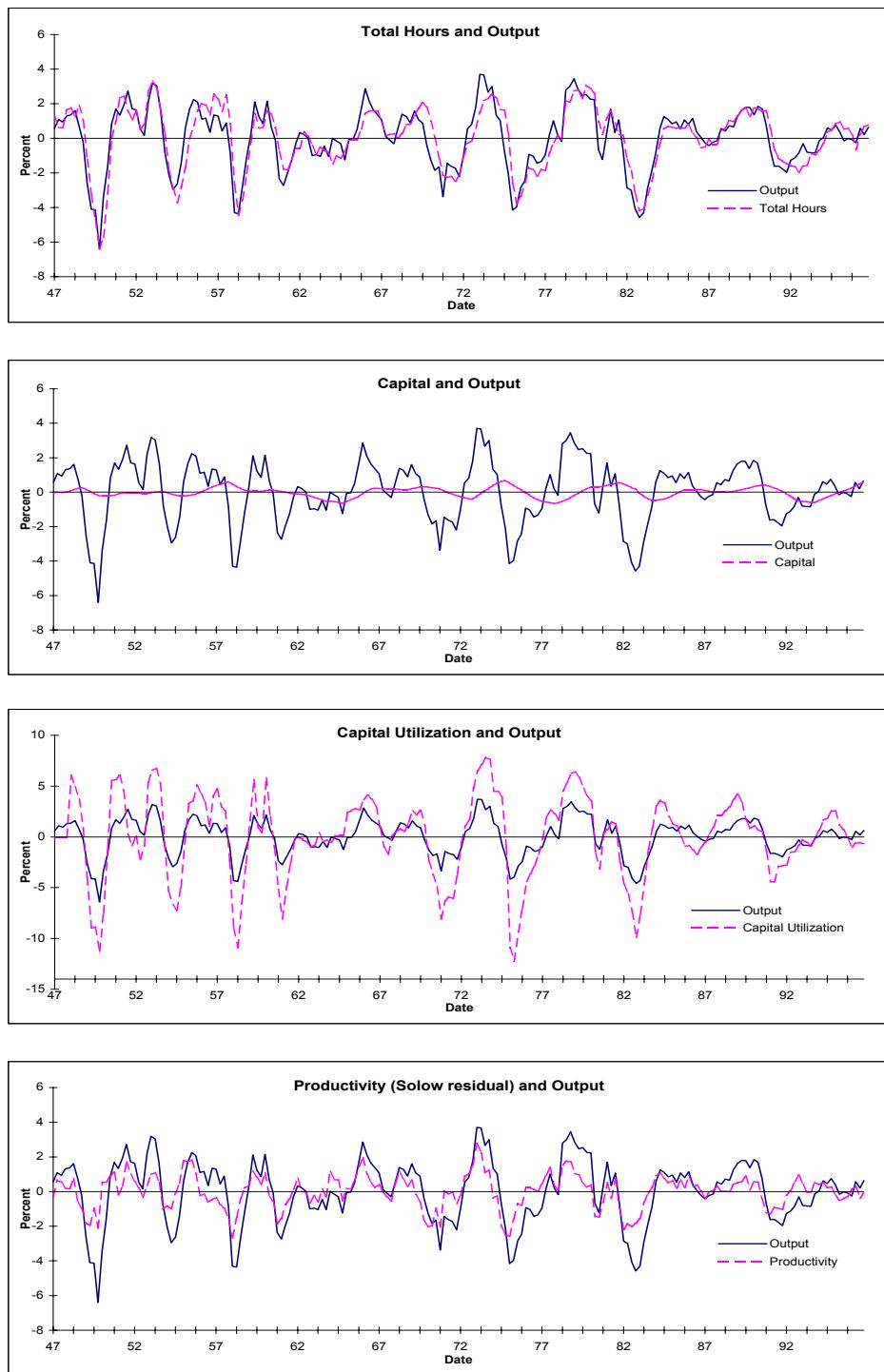


图 4.4—战后美国经济周期波动 • 二

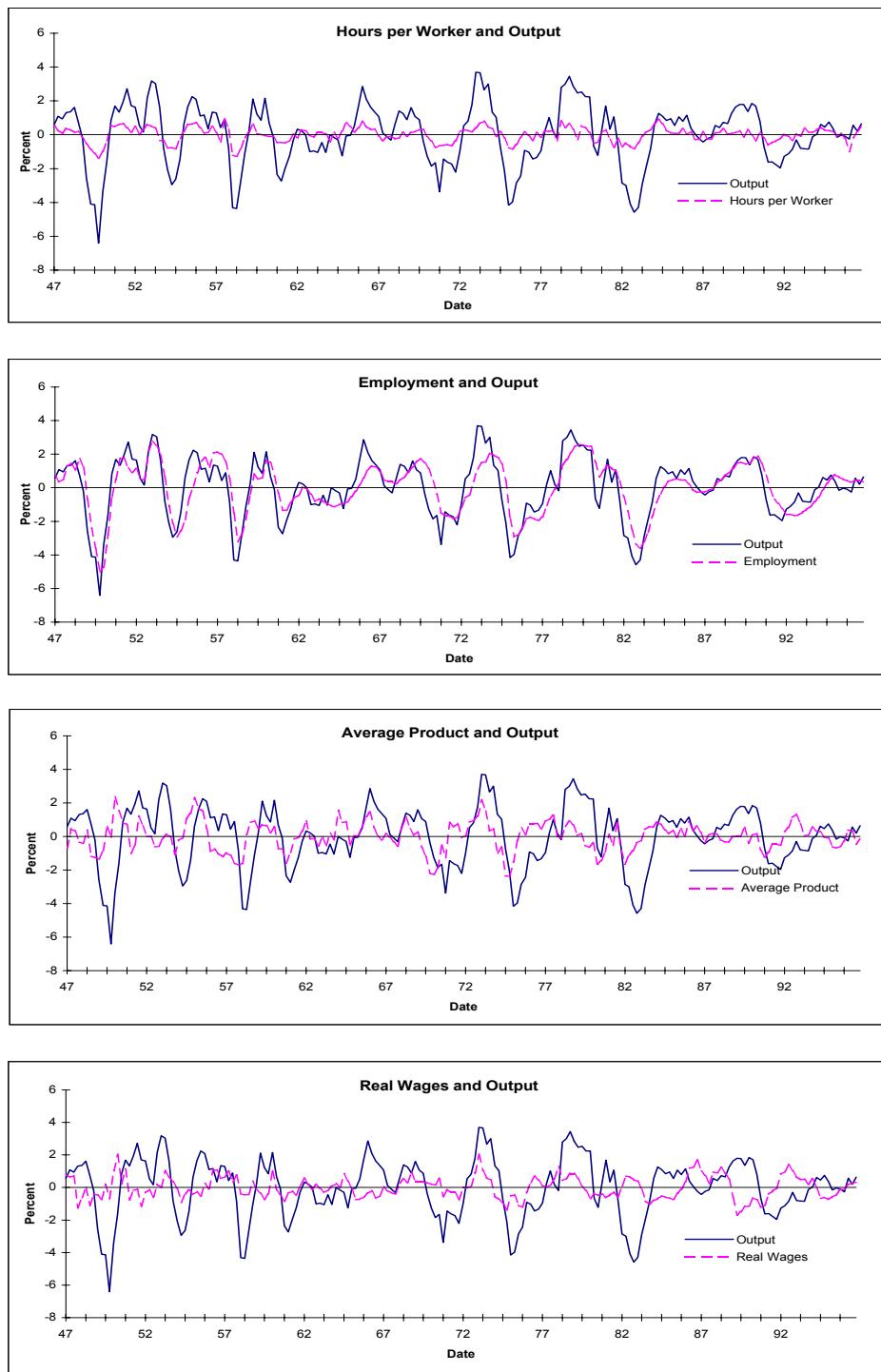


图 4.5.—战后美国经济周期波动 • 三

表 4.1
美国战后经济周期波动数据

| 变量 | 符号 | 标准差 | 与产出相对标准差 | 一阶自回归 | 与产出相关系数 |
|-------|-------|------|----------|-------|---------|
| 产出 | Y | 1.81 | 1.00 | 0.84 | 1.00 |
| 消费 | C | 1.35 | 0.74 | 0.80 | 0.88 |
| 投资 | I | 5.30 | 2.93 | 0.87 | 0.80 |
| 劳动时间 | N | 1.79 | 0.99 | 0.88 | 0.88 |
| 劳动生产率 | Y/N | 1.02 | 0.56 | 0.74 | 0.55 |
| 索罗剩余 | A | 0.98 | 0.54 | 0.74 | 0.78 |
| 实际工资 | w | 0.68 | 0.38 | 0.66 | 0.12 |
| 实际利率 | r | 0.30 | 0.16 | 0.60 | -0.35 |

† 所有变量均为人均水平的对数，数据源自NIPA和BLS。

一些则不然。对于经济周期波动幅度与时间的研究当然是重要的，只不过并不是真实经济周期理论的目标。真实经济周期理论的目标是研究那些在不同经济周期之间保持不变的共性特征，而不是某个具体经济衰退或扩张所对应的具体原因，比如油价上涨、信息泡沫或者次级抵押贷款等等。^①横跨不同的经济周期之间，我们观察到一般的规律有两个。第一，大量的宏观经济变量之间的波动是相互关联的。第二，尽管我们不解释单个经济变量在不同经济周期波动幅度不同的原因，但是不同宏观经济变量在经济周期内的相对波动幅度的特征却是具有一般性的。比如投资的波动在所有的经济周期当中都要高于产出波动，而消费都要低于产出波动等等。这两个现象反过来又让我们猜想，经济周期很可能是一些不同的宏观经济变量对某一个驱动力（很可能是不可观测的）的反应，具体到每一个经济周期，其波动的幅度与持续时间的长度可能是有这个驱动力的具体实现造成的，但是所有经济周期当中观察到的共性特征是，很多宏观经济变量都对同一个驱动力进行反应，并且某些变量的变化大一些，另一些变量的变化小一些。特别的，真实经济周期理论上述两个问题所提出的假说是：图4.4-4 所刻画的索罗剩余（不可观测的技术冲击）在经济周期频率上的波动是造成经济周期的驱动力，实际经济周期当中可观测的其他变量的值都是对于索罗剩余变化的反应。尽管具体到某个特殊的经济周期，索罗剩余的实现可能是由各种各样的现实经济因素引起的，但是宏观经济对这个实现的反应，即不同变量相对波动浮动的不同与众多宏观经济变量的变化的联动，是由真实经济周期模型所刻画的机制解释的。也正是因为这个原因，当我们的真实经济周期模型产生的结果与实际数据相对比的时候，我们所对比的并不是宏观经济变量的具体时间序列曲线，我们所对比的是在一组足够长的数值仿真当中1)模型当中不同变量的相对波动幅度的大小和2)各变量之间的相关系数。这些统计量在数据当中的估计罗列于表4.1当中。

4.3 真实经济周期模型

下面我们来构造真实经济周期理论的模型。简单来说，我们将使用一个带有生产率随机冲击和弹性劳动供给的卡斯—库普曼斯模型。不过注意到，卡斯—库普曼斯模型并不是一个均衡模型(*Equilibrium Model*)，我们之前讨论的主要是在经济当中资源的最优分配问题，并不是经济当中不同的参与者对于价格信号反应的结果，前者通常被称为规划者问题(*social planner's problem*)。尽管如果我们考虑的是一个长期增长的问题，规划者问题的假设能够一定程度上被认为是合理的，在短期经济波动当中假设存在一个老大哥（乔治·奥威尔《1984》）一样的人物对资源进行分配显然是一个站不住脚

^① 对具体经济周期研究的例子可以参考关于大萧条[Chari, Jehoe and McGrattan (2002)和McGrattan (2012)]与大衰退[Glover et al. (2014)]、战后经济衰退[Chari, Kehoe and McGrattan (2007)]以及一般事件对于宏观经济波动影响[Bloom (2009)]的研究。

的理论。所以真实经济周期模型当中我们需要引入竞争均衡(*Competitive Equilibrium*)的概念，使得实际均衡的结果是由经济中的参与人——这里为消费者与企业——对于经济当中价格信号的最优反应相综合决定的。限于篇幅，我们这里不详细对基于阿罗—德布鲁—麦肯锡的一般均衡理论进行详细讨论，市面上优秀的教科书数不胜数。需要对一般均衡理论进行复习的同学，可以参考[Ljungqvist and Sargent \(2012\)](#)的七八两章或[Mas-Colell, Whinston and Green \(1995\)](#)的十、十五、十六三章。本教程当中我们不对一般均衡理论做严格的要求，但是读者需要能够从直觉和计算的角度对均衡进行理解。

本节当中我们首先简单介绍竞争均衡的直觉与模型的组成元素，随后再分析为了从数值上对经济周期模型进行求解，我们需要解决哪些问题，需要引入哪些新的工具。

4.3.1 模型与竞争均衡

简单来说，所谓竞争均衡是说宏观经济被分为产品与要素两大类市场，每种产品和每种要素在各自的市场上进行交易。每个市场上存在买方与卖方，需求与供给均由市场价格决定。在均衡当中，所有市场当中的价格使得市场出清，即需求等于供给。应用竞争均衡模型分析经济周期的一个隐含假设就是我们观察的每一个时间点上，宏观经济都处在均衡当中。这当然是一个简化的假设，比如失业就是劳动供给超过劳动均衡的一个信号，不过暂时先让我们保留这个假设，这使我们能够最大限度的应用微观经济学发展的用来分析均衡的工具。具体来看，在真实经济周期模型当中，有三个独立的市场：产品市场、资本市场与劳动市场，后两者又可以统一归类为要素市场。在产品市场上，消费者是需求方，企业是供给方；而在要素市场上，两者地位反转。我们假设产品市场生产的复合产品(*composite good*)是计价物，这可以理解为产品市场观察到的是国内生产总值的货币化价值，于是产品市场价格恒为1。记 w 与 R 是工资与利率（也就是它们相对于复合产品价格的比值），分别为劳动市场和资本市场的价格。企业的要素需求和消费者的要素供给受 w 与 R 的影响，在均衡当中要素需求等于要素供给。这里我们假设企业是新古典的零经济利润企业，即消费者是资本的所有者，企业每一期需要按照要素市场价格向消费者租用生产资本，并雇佣工人。在竞争均衡当中，企业的经济利润为零。不熟悉宏观经济学的同学粗看之下可能认为这个假设不切实际，因为在实体经济当中企业是拥有资本的。这里稍作解释。新古典企业的另一个假设是消费者拥有企业，故现实生活中企业所拥有的资本在模型当中被解释为企业所有者（消费者）所拥有的资本，现实生活中企业的会计利润在模型当中等价于消费者通过出租资本获得的回报。这个处理方法的好处是我们可以将经济当中所有的动态行为都通过消费者来刻画，从而简化模型分析的难度。^①在产品市场上，企业根据之前要素市场当中的要素需求以及当期经济周期冲击的具体实现决定最终生产水平，消费者将通过出租资本与劳动获得的收入全部用来购买企业的产品，所有的产品当中一部分用于当期消费，另一部分转化为投资（这里等价于储蓄）用以提高下一期企业的生产能力。在均衡当中， w 和 R 调整使得两个要素市场出清，而由消费者的预算约束，产品市场自然出清（这个结论又称为瓦尔拉斯定律），整个经济处在均衡当中。

所以小节一下，为了刻画某一个时间点 t 上的竞争均衡，我们需要做的是首先对于任给的 t 期的价格 w_t 和 R_t ，消费者的在预算约束下的消费储蓄行为与企业的要素需求行为，这给了我们每个市场上的需求曲线和供给曲线，之后求解使得要素市场出清的 w_t 和 R_t ，并将 w_t 和 R_t 带回刚刚求出来的个体行为当中，求出消费、储蓄、要素与生产的具体分配。然后对整个时间轴上的所有 t ，求解上述价格与分配，最后获得价格与分配的时间序列实现。

下面我们具体来构造模型。我们首先介绍模型当中外生给定的技术参数，然后介绍企业行为，最

^①但是如果读者分析的问题需要考虑资本利得与分红，那么就需要在模型当中引入动态企业，两个近期的例子可以参考[Gourio and Miao \(2010\)](#)和[Anagnostopoulos, Cárcel-Poveda and Lin \(2012\)](#)。

后刻画消费者行为并给出一般均衡的规范定义。在这里，我们将考虑一般均衡的两种定义，第一种是如上文所叙述的序列定义(*Sequential Equilibrium*)，这个定义比较容易理解，同时在求解一阶条件分析模型解析性质的时候很容易使用，但是在使用值函数迭代的计算当中应用起来并不方便。^①所以我们随后会给出竞争均衡的另一种定义方式即递归定义(*Recursive Equilibrium*)，在第三部分讨论异质性消费者的时候，读者将会发现递归定义应用上的便利之处。本小节当中，我们用大写字母表示宏观加总之后的变量，小写字母表示其人均水平。

企业。—假设经济当中存在一个代表性企业，其生产函数是规模报酬不变的

$$Y_t = A_t F(K_t, N_t), \quad (4.3)$$

当中 Y_t 是总产出， K_t 和 N_t 是总资本与总劳动， A_t 是外生的生产率冲击。此处我们假设 $\log A_t$ 是自回归系数为 ρ 的AR(1)过程

$$\log A_{t+1} = \rho \log A_t + \varepsilon_{t+1}, \quad 0 < \rho < 1, \quad (4.4)$$

其中 ε_t 是服从以零为均值， σ_ε 为标准差的正态分布， $0 < \rho < 1$ 保证了上述过程是平稳的。^②再假设最终产品 Y_t 是计价物，于是最终产品价格为1。若记工资和利率分别为 w_t 与 R_t ，那么企业的利润最大化函数定义为

$$\max_{K_t, N_t} \{A_t F(K_t, N_t) - w_t N_t - R_t K_t\}, \quad \forall t \quad (4.5)$$

对(4.5)求一阶条件得到给定价格时的要素需求函数

$$w_t = A_t F_N(K_t, N_t), \quad (4.6)$$

$$R_t = A_t F_K(K_t, N_t), \quad (4.7)$$

注意上述要素需求函数同样是对任意 t 定义的。若 w_t 和 R_t 已知，方程(4.6)和(4.7)实际上定义了一组关于 K_t 和 N_t 的二元方程组。

消费者。—我们假设经济当中存在单位测度的无穷多个完全相同的代表性消费者，每个消费者在每一期拥有一单位时间禀赋。消费者可以将时间用于工作与闲暇，每个消费者在每一期的效用都是由消费 c_t 与闲暇 l_t 的水平决定的，这里 $l_t + n_t = 1$ 。再记效用函数为 $u(c_t, l_t)$ 。设每个消费者在 t 期拥有资本 k_t ， k_t 是继承自上期投资行为的状态变量，对消费者而言是给定的，那么若我们令消费者在 t 期的投资为 i_t ，消费者的预算约束此时成为

$$c_t + i_t = w_t n_t + r_t k_t.$$

我们再设资本存量每期会以线性比例 δ 自然折旧，于是资本的动态方程写作

$$k_{t+1} = (1 - \delta)k_t + i_t, \quad \forall t. \quad (4.8)$$

回想在索罗模型当中，我们假设消费者（当时我们解释为一个规划人）的目的是最大化未来效用的贴现。在经济周期模型当中，由于存在随机的生产率冲击 A_t ，企业在未来的产量与市场中的价格存在不确定性，即消费者未来收入的真实购买力是不确定的，所以这里消费者只能最大化未来期望效用的贴现。记贴现率为 β ，那么消费者的效用函数此时成为

$$U = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t, l_t), \quad 0 < \beta < 1,$$

^①不过后面我们将会看到，在使用扰动法计算的时候，序列定义用起来就非常方便了。另一种序列均衡用起来非常方便的场合是使用政策函数迭代法的情形，不过限于篇幅这里不涉及。

^②下一章当中我们将详细介绍平稳过程的概念。

其中 \mathbb{E}_0 的意义是在第零期关于 A_t 在整个时间序列（这里是无穷期的）的实现取期望。注意到由方程(4.6)和(4.7)，这里消费者实际上是在对未来的序列 $\{w_t\}$ 和 $\{R_t\}$ 形成期望。

现在我们可以定义消费者的最优化问题了，任给 w_t 与 R_t ：

$$\begin{aligned} & \max_{\{c_t, k_{t+1}, n_t\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t, 1 - n_t) \\ & \text{s.t.} \\ & c_t + k_{t+1} - (1 - \delta)k_t = w_t n_t + R_t k_t. \end{aligned} \quad (4.9)$$

这里再次提醒读者注意，若 w_t 和 R_t 已知，那么应用我们上一章的方法，我们可以求出消费者对于 c_t , k_{t+1} 和 n_t 的决策，这刻画了消费者的产品需求与要素供给。

具体来看，优化问题(4.9)所对应的拉格朗日函数为

$$\mathcal{L} = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \{u(c_t, 1 - n_t) + \lambda_t [w_t n_t + (R_t + 1 - \delta)k_t - k_{t+1} - c_t]\}$$

上式关于 c_t , n_t 和 k_{t+1} 分别求一阶条件得到

$$(c_t) : \frac{\partial u}{\partial c_t} = \lambda_t, \quad (4.10)$$

$$(n_t) : \frac{\partial u}{\partial l_t} = w_t \lambda_t, \quad (4.11)$$

$$(k_{t+1}) : \lambda_t = \beta \mathbb{E}_t [\lambda_{t+1} (R_{t+1} + 1 - \delta)]. \quad (4.12)$$

将(4.10)和(4.11)相结合，我们得到劳动供给曲线（有时候这又称为期内欧拉方程(*Intratemporal Euler Equation*)）

$$w_t \frac{\partial u}{\partial c_t} = \frac{\partial u}{\partial l_t}. \quad (4.13)$$

方程(4.13)刻画了消费者的劳动供给决策。这个决策是由劳动供给的边际收益和成本（分别由方程(4.13)的左右两侧描述）决定的。具体来说，因为此时消费者能够从闲暇当中获得效用，所以如果以效用为尺度来衡量，消费者在工作时所放弃的那部分闲暇带来的边际效用就是劳动供给的成本。通常我们会假设闲暇是边际效用递减的，即对一个终日无所事事的人来说，额外的一小时闲暇并不会带来太高的享受。反过来对一个日程表连轴转的人来说，几天的年假带来的改变就极为可观了。另一方面，劳动的边际收益是由工资和边际效用两部分组成的。消费者在决定劳动供给的时候不仅仅会考虑额外的劳动能够带来多少货币收入，还要考虑货币收入以效用记能够带来多大的福利的提高。如果我们继续假设消费的边际效用递减，那么对一个当前消费水平很高的人来说（边际效用很低），再高的工资也很难带来劳动供给较大幅度的提高，因为对这个消费者来说是否增加消费对于效用水平的影响是很微小的。注意到这里隐含的一个意义是消费与闲暇是相互替代的，事实上，如果我们将方程(4.13)略作处理，写成

$$\frac{\partial u / \partial l_t}{\partial u / \partial c_t} = \text{MRS}_{lc} = w_t,$$

我们就得到微观经济学当中所熟悉的闲暇对消费的边际替代率等于工资（市场替代率）。

再将(4.10)和(4.12)相结合，我们得到熟悉的刻画消费储蓄的跨期欧拉方程(*Intertemporal Euler Equation*)

$$\frac{\partial u}{\partial c_t} = \beta \mathbb{E}_t \left[\frac{\partial u}{\partial c_{t+1}} (R_{t+1} + 1 - \delta) \right]. \quad (4.14)$$

方程(4.14)有几个值得我们注意的地方。首先，这里的数学期望是针对 t 期时候取值不确定的变量，也就是边际效用 $\partial u / \partial c_{t+1}$ 和 R_{t+1} 取的。当然，因为两者实际上与生产率冲击 A_{t+1} 之间存在对应关系，所以这里的数学期望实质上是对以生产率冲击刻画的宏观经济状态取的。进一步的，如果我们假设经济不存在波动，并且带入企业的一阶条件 $R_t = \partial F / \partial K_t$ ，(4.14)式就回到了我们所熟悉的增长模型当中的情况。^①第二，由效用函数是凹函数的假设，应用延森不等式(*Jensen's Inequality*)我们有

$$\frac{\partial u}{\partial \mathbb{E}_t c_{t+1}} \leq \mathbb{E}_t \frac{\partial u}{\partial c_{t+1}},$$

当效用函数严格凹的时候取严格不等号。对照方程(4.14)，这是说相比于增长模型，当存在不确定性的时候，欧拉方程右侧的值会变大，于是为了确保等式依然成立，等式左侧的值（本期的边际效用）也需要相应的增大。在边际效用递减的假设下，这意味着存在不确定性的时候本期的消费水平下降，储蓄水平提高。在文献当中，我们将由收入不确定性带来的储蓄的增加称为预防性储蓄(*precautionary savings*)。^②

竞争均衡。—现在我们刻画竞争均衡，即市场出清条件。注意到我们假设了单位测度的无穷多个完全相同的代表性消费者，如果我们用下标 j 来标记单个消费者，那么任何人均水平变量 x_t 在宏观层面的加总水平 X_t 就可以定义为

$$X_t = \int x_{t,j} dj. \quad (4.15)$$

当然因为这里我们假设了代表性消费者，所以所有消费者的行为都是相同的即 $x_{t,j} = x_t$ 。又因为消费者总数测度为一，所以这里数值上我们有

$$X_t = x_t \int dj = x_t \times 1 = x_t, \quad (4.16)$$

即在当前的模型中，个体水平与加总水平变量的区分在数值上是不必要的（当然在概念上这个区分依然重要）。不过这里希望读者养成良好的理论思考习惯，因为在第三部分异质性消费者的讨论当中，由方程(4.15)定义的加总在概念上和数值计算上都是极为重要的。应用方程(4.16)，我们有

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

同时，消费者的预算约束加总之后得到

$$C_t + K_{t+1} - (1 - \delta)K_t = w_t N_t + R_t K_t, \quad (4.17)$$

因为生产函数是规模报酬不变的（一次齐次）且产品和要素市场是完全竞争的，上式的右侧等于总产出 $F(K_t, N_t)$ 。于是，我们有资源约束

$$C_t + K_{t+1} - (1 - \delta)K_t = Y_t. \quad (4.18)$$

现在我们可以定义竞争均衡了。我们首先给出序列版本的定义，这种方式的定义与我们之前在直觉上的叙述是一致的。之后我们再给出递归形式的竞争均衡。

^①值得一提的是，增长模型是一个规划者问题，而经济周期模型是一个均衡模型。我们可以证明的是当第一福利经济学定理成立（完备信息、完备市场、完全竞争、无外部性）的时候，两者所对应的均衡是等价的。实际的证明过程就是如我们这里所做的那样验证刻画两个均衡的一阶条件是相互等价的，本教程当中我们不对这个结论进行严格的证明，有兴趣的同学请参考Mas-Colell, Whinston and Green (1995)的第十六章。

^②在这里，收入的波动是由 R_{t+1} 和 w_{t+1} 造成的，其中后者经由 c_{t+1} 进入跨期欧拉方程。

定义4.1. (竞争均衡的序列定义) 我们定义竞争均衡为价格序列 $\{w_t, R_t\}$ 与个体分配序列 $\{c_t, k_{t+1}, n_t\}$ 和加总分配序列 $\{C_t, K_t, N_t, Y_t\}$ 使得

- (i) 企业最优化: 给定 $\{w_t, R_t\}$, $\{K_t, N_t\}$ 是(4.5)的解。
- (ii) 消费者最优化: 给定 $\{w_t, R_t\}$, $\{c_t, k_{t+1}, n_t\}$ 是(4.9)的解。
- (iii) 价格 $\{w_t, R_t\}$ 使得产品市场出清

$$C_t + K_{t+1} - (1 - \delta)K_t = Y_t.$$

注意到在(iii)当中严格的书写应该有劳动、资本与产品三个市场的出清条件, 我们这里只写了产品市场一个出清条件是因为在推导消费者预算约束与资源约束的等价性的时候 (上文(4.17)推导(4.18)式) 隐性的带入了消费者要素供给 ((4.17)式右侧) 等于企业的要素需求 ((4.6)和(4.7)两式), 即要素市场出清。此处的简化是源自代表性消费者的假设, 读者在第三部分将会看到, 对于异质性消费者问题, 上述三个市场的定义必须同时被显式的给出, 并且在数值求解的时候进行计算。

下面我们将上述模型写成递归形式。首先我们注意到企业的问题是完全静态的, 企业相邻的两期之间的决策是独立的。在任意 t 期, 消费者面临的状态变量即在当期消费者无法改变的变量为在上一期预先决定的资本存量 k_t 、本期外生的生产率冲击 $\log A_t$, 以及加总的资本存量 K_t 。这里有两点需要注意, 首先 t 期加总的均衡劳动并不是状态变量, 因为所有消费者本期都可以做出独立的劳动供给决定。其次, 消费者这里需要的实际上是价格 w 和 R 取值, 我们这里选 A_t 和 K_t 作为外生状态变量主要是因为由企业的最优化, (w, R) 可以表示成 A, K 的函数。于是应用第三章的记号, 消费者本期的值函数可以写为 $v(k; A, K)$, 我们通常用分号分隔内生状态变量 (即消费者可以影响的状态变量, 也就是那些在相邻两期之间会改变控制一状态变量身份的, 这里是消费者本期的自有资本) 和外生状态变量 (即消费者无法控制服从一个外生给定的动态更新方程的变量, 这里是生产率冲击和整个经济的总资本存量)。为了将问题写成递归形式, 消费者必须知道本期的决策会如何影响下一期 k, A, K 的取值。下一期的 k 是消费者本期的控制变量之一, 所以这个是可以确定的。现在消费者需要做的就是正确的判断下一期外生状态变量 A 和 K 的期望。回看模型设置, 我们发现方程(4.4)给了我们 A_t 动态的信息, 而方程(4.16)给了我们 K 的变化预测方式。我们提醒读者留意的是在这里由于代表性消费者的假设, 对总资本 K 的动态的预测是容易的。在第十章中读者将会看到, Krusell and Smith (1998) 的核心贡献就是提供了一个在异质性消费者环境下对资本的动态方程进行近似计算的方法。于是我们将消费者最优化问题(4.9)写成递归形式:

$$\begin{aligned} v(k; A, K) &= \max_{c, k', n} \{u(c, 1 - n) + \beta \mathbb{E}[v(k'; A', K')|A]\} \\ \text{s.t.} \\ c + k' - (1 - \delta)k &= w(A, K)n + r(A, K)k, \\ K' &= k', \\ \log A' &= \rho \log A + \varepsilon'. \end{aligned} \tag{4.19}$$

优化问题(4.19)和我们之前看到的(3.4)有两个主要区别, 一是值函数的维度现在从一维提高为三维, 二是在优化函数的内部有一个条件期望 (积分) 运算需要额外处理。在随后的三章当中我们将会看到, 对于这两个主要区别的处理, 是求解带有不确定性的动态优化问题的核心。不过先让我们把计算放在一边, 用定义4.1的递归形式结束我们对于模型的介绍。回想之前两章当中我们对于序列问题和动态优化问题的讨论, 序列问题当中我们求解的是一组消费和储蓄的序列, 而转化成递归形式的动态优化问

题当中我们求解的是消费和储蓄关于本期资本存量（状态变量）的函数形式。此时的情形也是大体相同的，在定义4.1当中我们将均衡定义为一系列价格与分配的序列，在递归版本的一般均衡定义当中我们要做的是求解价格与分配关于本期状态变量的函数。具体的，定义如下给出：

定义4.2. （竞争均衡的递归定义）我们定义竞争均衡为值函数 $v(k; A, K)$ ，价格函数 $w(A, K), R(A, K)$ ，消费者决策函数 $c(k; A, K), n(k; A, K), k'(k; A, K)$ 以及加总的分配函数 $C(A, K), K'(A, K), N(A, K), Y(A, K)$ 使得

- (i) 企业最优化： $w(A, K), R(A, K), N(A, K)$ 使得(4.6)和(4.7)成立。
- (ii) 消费者最优化： $v(k; A, K), w(A, K), R(A, K), c(k; A, K), n(k; A, K), k'(k; A, K)$ 是(4.19)的解。
- (iii) 加总一致性：

$$X(A, K) = \int x_j(k; A, K) dj = x(k; A, K), \quad X = C, K', N.$$

- (iv) 市场出清：

$$C(A, K) + K'(A, K) - (1 - \delta)K = Y(A, K).$$

这里同样提请读者注意的是在市场出清部分我们应该针对每个市场有单独的出清条件，这里由定义(4.1)中相同的原因被简化成为产品市场出清。另外在加总一致性条件中，下标 j 同样用于标识消费者，这样的处理在这里并不必须，但是读者将会在异质性消费者模型当中看到它的必要性。

至此，我们完成了对于模型的刻画。

4.3.2 求解模型

下面我们来求解由定义4.2刻画的均衡。这里我们再次重复一下，定义4.1和4.2是等价的，这里递归均衡的原因是其在计算当中应用起来比较方便。定义4.2中刻画的均衡与我们在上一章当中学习的技术有联系，也有区别。原则上来说，我们现在的目标就是求解定义4.2当中那一大堆具有特殊性质的函数。从一个机械的角度来看，定义4.2当中“最复杂”的式子（方程(4.19)）看上去跟上一章当中我们求解的动态优化问题至少长得像，所以一个合理的猜测是求解这个递归均衡应该跟之前的增长模型颇有相似之处，至少最麻烦的地方应该是差不多的。另一方面，定义4.2跟第三章的优化问题相比不同的地方在于现在多出来一大堆价格函数和加总一致性条件是我们之前所没有处理过的。于是，如果我们现在要求解这个均衡，一个合理的猜想就是我们是否能够进行一定的变换，把定义4.2的求解（也就是那些函数的求解）完全转化为另一个与第三章当中的动态优化问题类似的问题，毕竟在之前的动态优化当中，我们求解的其实也是一组满足了一定条件的函数。这个猜想是对的。所以本小节当中，我们首先将定义4.2的求解转化为一个等价的随机动态优化问题，然后再分析要求解这个等价的问题，我们需要使用哪些新的数学工具。

竞争均衡与规划问题。—优化问题(4.19)与之前的社会规划问题的不同主要有两点：1)此时人均资本 k 和加总资本 K 在概念上是不同的；2)消费者的选择约束此时是由受工资和利率影响的预算约束刻画的，而非之前受生产函数（技术）影响的资源约束刻画的。我们的目标是验证在均衡当中，上述两点与社会规划问题是等价的。

由于我们做了代表性消费者的假设，那么在均衡当中所有的消费者行为是一致的，又因为我们假设消费者的测度为一，所以所有的消费者相关的变量，其人均值都与加总值相等。这里我们强调在均

衡中是因为在概念上我们允许消费者做不同的选择，只不过因为所有消费者在均衡当中面临相同的约束面对相同的价格，从而作为模型在均衡当中内生的结果，会做出同样的选择。这里容作者不厌其烦的强调，人均变量与加总变量之间的等价性是内生的模型性质，不是假设。不过数值上，我们确实可以用将所有的 x 替换为 X 。这样做的一个重要的好处是我们现在可以在(4.19)的预算约束当中用企业的一阶条件替换掉价格（两者受加总变量影响的）。再应用生产函数是一次齐次的性质，我们可以将(4.19)的预算约束转化为我们熟悉的资源约束（事实上我们在上一小节已经验证过这个结论）

$$c + k' - (1 - \delta)k = AF(k, n).$$

于是，我们验证了满足递归均衡4.2的函数组同时也是下述社会规划问题的解

$$\begin{aligned} v(k; A) &= \max_{c, k', n} \{u(c, 1 - n) + \beta \mathbb{E}[v(k'; A')|A]\} \\ \text{s.t.} \\ c + k' - (1 - \delta)k &= AF(k, n), \\ \log A' &= \rho \log A + \varepsilon'. \end{aligned} \tag{4.20}$$

这样，我们就验证了任何使得定义4.2成立的函数组都是社会规划问题(4.20)的解。反过来我们同样可以验证，从社会规划问题(4.20)的解出发，总可以构造出一组价格函数使得该组价格函数与规划问题其他的函数作为一个整体满足递归均衡的定义。换句话说，我们可以证明的是此时递归均衡4.2与社会规划问题(4.20)的解在数值上是等价的（充分必要）。^①于是，应用竞争均衡的性质，我们便将递归均衡的求解转化为一个我们熟悉的社会规划问题的求解。需要提醒读者的是，这里的推导严格依赖于代表性消费者的假设。一般而言，存在异质性消费者的时候，由于人均变量与加总变量在数值上是不同的，所以对应着单个消费者的预算约束无法被转化为加总的资源约束，读者将在最后一章对于Krusell and Smith (1998)模型的讨论当中对这一点有更深刻的理解。^②

随机动态优化问题。—现在我们来分析如何从数值上求解社会规划问题(4.20)，首先来看规划问题(4.20)与(3.4)之间的区别。我们发现区别主要有两个，首先值函数从一维变成了二维；其次，此时两个状态变量当中的一个 A 是一个随机变量，同时在最优化问题内部的值函数外侧要进行期望运算 \mathbb{E} 。但是抛开具体的技术细节不提，如果我们知道 $\mathbb{E}[v(k; A)|A]$ 的表达式、或者有它的数值表达式，那么我们就可以求解最优政策函数。所以这里的关键把 $\mathbb{E}[v(k; A')|A]$ 数值的刻画出来。具体来说，我们将期望算子显式的写成积分，得到

$$\mathbb{E}[v(k; A'|A)] = \int v(k; A^\rho \varepsilon') d\Phi(\varepsilon), \tag{4.21}$$

其中积分是关于 ε' 进行的， $\Phi(\cdot)$ 是 ε' 的累积分布函数。从(4.21)可以看出， $\mathbb{E}[v(k; A')|A]$ 是一个关于 k 和 A 的二维函数，期望算子（积分运算）会将 ε 的信息加总，从而在最终表达式里消掉这个变量。^③所以从数值上刻画 $\mathbb{E}[v(k; A')|A]$ 就是对任意的 (A, k') 偶对（注意这里是 A 不是 A' ），我们要找到一个具体的数。那么现在我们的问题就转化为求(4.21)中的积分。对于积分运算，读者比较熟悉的做法可能是利用牛顿—莱布尼兹公式，将积分转化为原函数的差来求解。^④但是这要求我们知道被积函数（这里是 $v(k; A')\varphi(\varepsilon')$ ，其中 $\varphi(\varepsilon')$ 是密度函数）的解析性质，当我们只能数值的刻画被积函数的时候这条路走不通。这里我

^① 完整的证明有一些关于极限值的技术细节需要处理，故在此从略，但是实际上价格就是用企业的一阶条件进行构造。

^② 反过来，这里的经济周期模型也可以使用Krusell and Smith (1998)算法进行求解，只不过这么做难免有大炮打蚊子的感觉。同时，这里一个值得注意的一般原则是，在对模型进行数值求解之前，应该尽可能的应用模型的解析性质来对问题进行化简，以最大限度的降低数值解模型的复杂程度，提高算法的稳健性。

^③ 没有看出来这个联系的读者可以用一个 ε 取两个值的离散分布为例子验证。

^④ 读者这里不妨先将(4.21)想象成一个定积分，在实数轴上的不定积分理解为这个定积分关于上下限取极限。

们选择直接应用积分的定义来求解。回想数学分析当中的知识，求一个积分的四个步骤分别为分割、取点、求和、取极限，也就是说如果我们将积分区间划分为 n 个小段（并不一定要等分），每段 i 的中点为 ε_i ，那么(4.21)可以表示成

$$\int v(k; A^\rho \varepsilon') d\Phi(\varepsilon) = \lim_{n \rightarrow \infty} \sum_{i=1}^n v(k; \varepsilon_i) \varphi(\varepsilon_i). \quad (4.22)$$

现在方程(4.22)就可以被具体应用到计算当中了。因为随机冲击 ε' 的统计性质是已知的，即 $\varphi(\cdot)$ 的表达式已知，所以此时从任给的二维函数表 $v(k; A')$ 初始猜出发，我们都可以应用值函数迭代的算法对 $v(k; A')$ 进行求解，这里跟规划问题(3.4)唯一的区别，就是花括号内的计算因为存在积分而变得更加繁琐。^①

不过方程(4.22)在实际应用需要对效率进行一定的改进。注意到理论上，我们需要取大量的 ε_i 来对每一个 k 计算 $v(k; A')\varphi(\varepsilon_i)$ （ A' 是 ε' 的函数）之后进行求和，而这里的计算量是以几何级数提高的。比如，如果我们如上一章一样对 k 取300个点进行逼近，那么 ε' 取3个点的时候，每次迭代实际的计算量就会从确定性情形下的300次变成900次。如果我们同样对 ε' 取300个点进行逼近，那么单次迭代当中的计算量会变成 $300 \times 300 = 90,000$ 次！对于非常简单的模型，这样的计算或许还是可以接受的，但是如果读者想要具体的应用随机动态优化技术来分析现实问题，哪怕是使用编译语言应用计算机集群进行并行计算，很多情况下时间成本也是难以承受的。这种由于维度的提高而导致的算法效率的集聚下降，在数值计算当中被称为维度诅咒(*Curse of Dimension*)。幸运的是，(4.22)的本质是求一个数值积分，计算数学发展了大量应用少数几个点就可以对积分值进行极好的近似的技术。直觉上来看，假设 ε 是一个正态分布，由于正态分布本身是一个钟型曲线，取到极端值的概率很低，所以如果我们要近似正态分布的均值或者方差，显然更加靠近钟型曲线中间部分的点的性质更加重要。求积公式(*Quadrature*)的主要目的，就是找到在求数值积分过程当中比较重要的点。

于是在下一章当中，我们将首先学习如何能用比较经济的方式对一个随机过程进行近似。这里，我们将主要关注一类特殊的随机过程，即马尔科夫过程。而数学上我们需要引入的新的工具就是数值积分与求积公式相关的内容。当我们掌握了比较经济的求解(4.20)中的积分的技术之后，我们将在第六章当中对这个问题进行具体求解。

^①需要指出的是，在存在期望运算的时候，值函数迭代的收敛性需要更强的假设，这里证明从略，有兴趣的同学请参考Stokey and Lucas (1989)的第九章。

第五章 马尔科夫过程与数值积分

从概率论的角度出发，图4.3至4.5中刻画的宏观经济变量有几个特征，这些特征提示了我们应该使用何种数学工具对他们进行分析。第一，在任给一个时间点，变量的值是不确定的。此类现象我们知道适合使用随机变量进行描述。第二，在时间维度上，变量的波动发生在一个稳定的范围之内，同时这些波动围绕的“中间值”也不随时间变化。数学上我们使用随机过程(*stochastic process*)来描述这种随机变量随时间演变的性质，具有上面描述的这种稳定性过程，我们称为平稳过程(*stationary process*)。第三，为描述经济周期中宏观经济的波动，上一章当中我们构造了一个真实经济周期模型。尽管我们还没有对模型进行求解，但是读者应该能够看出来，模型的解给出的实际上就是一组随机变量的概率分布，只不过此时概率分布并不像一般概率统计教科书中那样由一个解析的函数表达式给出，而是由一个复杂的最优化问题间接描述的。特别的，该模型的随机性来自于一个一阶自回归过程，这个过程的一个重要特征是 $t+1$ 期的分布只与 t 期的值有关。具有这种特征的随机过程我们称为马尔科夫过程(*Markovian Process*)。本章当中，我们将对上述工具的理论与数值应用进行学习。

具体的，我们首先介绍随机过程的概念和平稳过程的数学特征。随后，我们讨论马尔科夫过程的一些简单特征。最后，我们学习数值上如何处理在动态优化问题当中出现的马尔科夫过程，即使用有限状态的马尔科夫链近似连续状态的马尔科夫过程。读者将会看到，这个问题的本质实际上是一个数值积分问题，于是在这部分内容中，我们还会穿插一些数值积分相关的基本知识。相对于本教程的其他章节，本章涉及的抽象概念比较多，希望读者在阅读的时候多从直觉的角度理解概念，同时留意各个概念之间的相互关系以及内容展开的逻辑框架，切勿死记硬背。

5.1 平稳过程

我们在第一章中已经初步接触了随机过程，在例子1.1中，方程(1.1)和(1.2)就是对应了刻画代表性劳动者生命周期收入的随机过程的数学表达式。直观上而言，在每一期 t ，劳动者的收入都是由一个分布函数已知的随机变量刻画的。从时间序列的角度来看，对某个特定的劳动者而言，对应的实际的收入流，就是随机变量每期具体抽取的随机数。现在我们对随机过程进行更加严格的数学描述。

5.1.1 随机过程的概念与数字特征

首先来看一些基本概念。

定义5.1.（随机过程）设 \mathcal{T} 是实数集，我们把依赖于参数 $t \in \mathcal{T}$ 的一族（无限多个）随机变量称为随机过程，记为 $\{X(t), t \in \mathcal{T}\}$ 。对每一个 $t \in \mathcal{T}$ ， $X(t)$ 是一个随机变量，我们称 \mathcal{T} 为参数集。我们常把 t 看

作时间，此时 $X(t)$ 称为 t 时刻过程的状态。若 $X(t_1) = x$ ，我们说这是 t_1 时刻过程处于状态 x 。对于一切 $t \in \mathcal{T}$ ， $X(t)$ 的所有取值 x 的全体称为随机过程的状态空间，记为 \mathcal{X} 。

在不引起混淆的情形下，我们一般用 $\{X_t\}$ 指代一个随机过程。依照参数集和状态空间的不同，我们可以对随机过程进行粗略的分类。具体来说，当 \mathcal{T} 是可数集合时， $\{X_t\}$ 称为离散参数随机过程或随机序列。我们在宏观经济学当中研究的主要是一类随机过程，此时 \mathcal{T} 是整数集。^① 反之，若 \mathcal{T} 为不可数集， $\{X_t\}$ 此时称连续参数随机过程。类似的，依照 \mathcal{X} 是否可数， $\{X_t\}$ 可以分为离散型随机过程与连续型随机过程。截止目前我们所看到的随机过程，无论是劳动者的收入，还是经济周期中的宏观经济变量，均为连续型随机过程。不过在本章的最后，我们将会学习如何用离散型过程去近似连续型过程。

定义5.2. (样本曲线) 对随机过程 $\{X(t)\}$ 进行一次试验称为随机过程的一条样本曲线。

在例子 1.1 中，假设我们仿真了 1,000 个不同的劳动者，那么某个劳动者 i 在生命周期中收入的具体实现就是一条样本曲线。在经济周期的例子中，一段时期内，对某个变量比如 GDP，实际观测到的 GDP 时间序列数据也是一条样本曲线。

由上面的定义可以看出来，随机过程实际上可以看作一个多维随机变量，所以对于随机过程的统计描述与对随机变量类似，可以利用分布函数进行刻画。具体的，给定随机过程 $\{X_t\}$ ，对于每一个固定的 $t \in \mathcal{T}$ ，随机变量 X_t 的分布一般与 t 有关，记为

$$F_X(x, t) = P(X_t \leq x), \quad x \in \mathcal{X},$$

此时 $F_X(x, t)$ 是随机过程 $\{X_t\}$ 的一维分布函数，而关于 t 将 $F_X(x, t)$ 汇总， $\{F_X(x, t), t \in \mathcal{T}\}$ 称为一维分布函数族。

一维分布函数族描述了随机过程在某时刻 t 时的统计特性，对该过程在不同时刻状态之间的联系，我们用 n 维分布函数来描述。设取样时间长度为 n ，对 n 个不同时刻 $t_1, t_2, \dots, t_n \in \mathcal{T}$ ， n 维分布函数记为

$$F_X(x_1, \dots, x_n, t_1, \dots, t_n) = P(X(t_1) \leq x_1, \dots, X(t_n) \leq x_n), \quad x_i \in \mathcal{X}.$$

类似的，给定 n ，关于 t_i 将 n 维分布函数汇总，我们得到 n 维分布函数族。正如分布函数是刻画了随机变量的本质特征，随机过程的本质特征也是由分布函数完全刻画的。

不过分布函数很多时候在应用中并不方便，在对于随机变量的分析中，我们经常关注以该变量的矩为代表的一些数字特征，比如均值、方差、偏度、峰度等。由于随机过程与多维随机变量类似，所以我们通常也对随机过程的这些数字特征更感兴趣。事实上，在宏观经济学当中，我们所关心的变量的分布函数绝大多数情形下是不存在解析表达式的，而我们通常也没有足够的数据应用非参数方法对整个分布函数进行分析，所以实践中必须依赖于利用数据样本对随机过程的这些数字特征进行近似。下面我们给出一些常用的数字特征的定义。

定义5.3. (均值函数) 对随机过程 $\{X_t, t \in \mathcal{T}\}$ ，固定 $t \in \mathcal{T}$ ，随机变量 X_t 的均值一般为 t 的函数，我们称这个函数为 $\{X_t\}$ 的均值函数，记为

$$\mu_X(t) = \mathbb{E}[X_t].$$

需要强调的是， $\mu_X(t)$ 是所有样本函数在 t 时刻的函数值平均值，不是一条足够长的样本曲线在时间维度上的均值。前者我们一般称为集平均或者统计平均，后者称为时间平均。用我们熟悉的例子，在生

^①事实上，由集合论的知识，任何可数集都与整数集同胚。

命周期收入的例子中，对某个 t ——比如 $t = 30$ ——如果我们仿真足够多的劳动者样本 i ——比如10,000个——那么关于这一万个样本在 $t = 30$ 时的均值，就是集平均。反过来，时间平均的例子可以看做我们观测了美国GDP在足够长的时间内的实现，对美国过去100年间的GDP取平均值，就是时间平均的一个例子。

仿照均值函数的定义，我们可以定义随机过程的高阶数字特征。

定义5.4. (高阶数字特征) 对随机过程 $\{X_t, t \in \mathcal{T}\}$ ，固定 $t \in \mathcal{T}$ ，我们定义随机变量 $X(t)$ 的二阶原点矩和二阶中心矩分别为

$$\begin{aligned}\Phi_X^2(t) &= \mathbb{E}[X_t^2] \\ \sigma_X^2(t) &= \mathbb{E}[(X_t - \mu_X(t))^2],\end{aligned}$$

他们分别称为均方值函数与方差函数。

均值函数反应了 X_t 各个时刻取样波动的中心，方差函数反应了取样波动对于均值的平均偏离程度。

以上的数字特征反应了随机过程在某个特定时刻的数字特征，对于随机过程在不同时刻取值的相互关系，我们有如下的概念。

定义5.5. (随机过程的跨时特征) 对随机过程 $\{X_t, t \in \mathcal{T}\}$ 和任意 $t_1, t_2 \in \mathcal{T}$ ，随机变量 X_{t_1} 和 X_{t_2} 的二阶混合原点矩定义为

$$R_X(t_1, t_2) = \mathbb{E}[X_{t_1} X_{t_2}],$$

称为自相关函数。而其二阶混合中心矩定义为

$$C_X(t_1, t_2) = \mathbb{E}[(X_{t_1} - \mu_X(t_1))(X_{t_2} - \mu_X(t_2))],$$

称为自协方差函数。

由上面的定义不难验证，上述数字特征之间存在如下的联系：

$$\begin{aligned}\Phi_X^2(t) &= R_X(t, t), \\ \sigma_X^2(t) &= R_X(t, t) - \mu_X^2(t) \\ C_X(t_1, t_2) &= R_X(t_1, t_2) - \mu_X(t_1)\mu_X(t_2),\end{aligned}$$

所以数字特征当中最重要的是均值函数与自相关函数。

下面来看一个简单的例子。

例子5.1. 设 A, B 是两个随机变量，对 $t \in \mathbb{R}$ ，定义随机过程 $X_t = At + B$

(i) 试写出 $\{X_t\}$ 的均值函数与自相关函数。

(ii) 若进一步假设 A 服从标准正态分布， B 服从 $[0, 2]$ 区间上的均匀分布且 A, B 相互独立，试求解此时 $\{X_t\}$ 的均值函数与自相关函数。

解

(i) X_t 的均值函数为

$$\mu_X(t) = \mathbb{E}[X_t] = \mathbb{E}[At + B] = t\mathbb{E}[A] + \mathbb{E}[B],$$

自相关函数为

$$\begin{aligned} R_X(t_1, t_2) &= \mathbb{E}[X_{t_1} X_{t_2}] = \mathbb{E}[(At_1 + B)(At_2 + B)] \\ &= t_1 t_2 \mathbb{E}[A^2] + (t_1 + t_2) \mathbb{E}[AB] + \mathbb{E}[B^2]. \end{aligned}$$

(ii) 当 A 服从标准正态分布时, $\mathbb{E}[A] = 0$, $\mathbb{E}[A^2] = 1$; 类似的, $\mathbb{E}[B] = 1$, $\mathbb{E}[B^2] = 4/3$ 。又因为 A, B 相互独立所以 $\mathbb{E}[AB] = \mathbb{E}[A]\mathbb{E}[B] = 0$, 带入上面的表达式得到

$$\mu_X(t) = 1, \quad R_X(t_1, t_2) = t_1 t_2 + 4/3.$$

□.

很多时候我们也关心两个随机变量之间的统计关系, 比如在上一章当中我们就关心经济周期频率下, 各种宏观经济变量与总产出波动之前的关系。我们以二维随机过程为例介绍几个基本概念。

定义5.6. (二维随机过程) 设 $\{X_t\}$ 和 $\{Y_t\}$ 是依赖于同一参数 $t \in \mathcal{T}$ 的随机过程, 对于不同的 $t \in \mathcal{T}$, (X_t, Y_t) 是不同的二维随机变量, 此时我们称 $\{X_t, Y_t\}$ 为二维随机过程。此时设 t_1, \dots, t_n 和 t'_1, \dots, t'_m 是 \mathcal{T} 中任意两组向量, 那么 $n + m$ 维随机变量

$$(X_{t_1}, \dots, X_{t_n}, Y_{t'_1}, \dots, Y_{t'_m})$$

的联合分布函数

$$P(X_{t_1} \leq x_1, \dots, X_{t_n} \leq x_n, Y_{t'_1} \leq y_1, \dots, Y_{t'_m} \leq y_m)$$

就是这个二维随机过程的 $n + m$ 维联合分布函数。

特别的, 当

$$\begin{aligned} P(X_{t_1} \leq x_1, \dots, X_{t_n} \leq x_n, Y_{t'_1} \leq y_1, \dots, Y_{t'_m} \leq y_m) &= \\ P(X_{t_1} \leq x_1, \dots, X_{t_n} \leq x_n) P(Y_{t'_1} \leq y_1, \dots, Y_{t'_m} \leq y_m) \end{aligned}$$

时, 我们说随机过程 $\{X_t\}$ 与 $\{Y_t\}$ 相互独立。

仿照一维随机过程中的定义, 我们可以定义二维随机过程的一些常用数字特征。

定义5.7. (联合数字特征) 对二维随机过程 $\{X_t, Y_t\}$, 称其二阶混合原点矩

$$R_{XY}(t_1, t_2) = \mathbb{E}[X_{t_1} Y_{t_2}]$$

为随机过程 $\{X_t\}$ 与 $\{Y_t\}$ 的互相关函数。类似的, 称其二阶混合中心矩

$$\begin{aligned} C_{XY}(t_1, t_2) &= \mathbb{E}\{[X_{t_1} - \mu_X(t_1)][Y_{t_2} - \mu_Y(t_2)]\} \\ &= R_{XY}(t_1, t_2) - \mu_X(t_1)\mu_Y(t_2) \end{aligned}$$

为互协方差函数。

如果对于任意 $t_1, t_2 \in \mathcal{T}$, 都有 $C_{XY}(t_1, t_2) = 0$, 那么我们说随机过程 $\{X_t\}$ 与 $\{Y_t\}$ 不相关。注意到两个随机过程的独立性要强于不相关性, 后者只反映两个随机过程间的线性关系。

5.1.2 平稳过程

宏观经济学所研究的时间序列数据有两个特征值得注意。第一是我们已经提到过的，图像上，这些变量在去掉趋势之后表现出了波动上的稳定性。第二点与我们刚刚提到的随机过程的数字特征有关。注意到上一小节当中所有的数字特征都是针对集平均定义的，这是说要对随机变量的分布函数或者数字特征进行分析，我们需要对同一个随机变量反复进行抽样得到的数据。这等于说如果我们要分析美国的经济周期数据，我们需要让一个假想中的美国反复经历战后50年的经济发展过程以收集足够多的样本曲线，显然这在现实生活中是不可能实现的。针对这两个特点，本小节中我们详细介绍一类特殊的随机过程——平稳(stationary)过程。顾名思义，平稳过程的随机性具有一定的稳定性。更重要的是，平稳过程具有一个重要的性质——遍历性(ergodicity)。遍历性是说对一条样本曲线足够长时间的取样可以替代多条样本曲线的反复取样，从而为宏观经济数据统计性质的分析提供了理论基础。

平稳过程的概念

我们首先给出平稳过程的定义。

定义5.8. (强平稳过程) 任给整数 n , 若对于任意的 $t_1, \dots, t_n \in \mathcal{T}$ 和 $h \in \mathcal{T}$, n 维随机变量的联合分布函数满足

$$P(X_{t_1} \leq x_1, \dots, X_{t_n} \leq x_n) = P(X_{t_1+h} \leq x_1, \dots, X_{t_n+h} \leq x_n),$$

我们就说 $\{X_t\}$ 是强平稳的。

由于强平稳过程在任何时候的一维分布函数都是相同的，显然其均值和自相关函数也是不随时间变化的。不过由于实际应用中我们很少将随机过程的分布函数完整刻画出来，而是通常只关注一些重要的数字特征，所以从数字特征出发我们可以定义一类条件较弱的平稳过程。

定义5.9. (弱平稳过程) 给定随机过程 $\{X_t, t \in \mathcal{T}\}$, 若对任意 $t, t+h \in \mathcal{T}$ 都有

$$\begin{aligned}\mathbb{E}[X_t] &= \mu_X \\ \mathbb{E}[X_t X_{t+h}] &= R_X(h),\end{aligned}$$

我们就称 $\{X_t\}$ 是弱平稳的。

由上述两个定义可以看出，任何强平稳过程当二阶矩存在的时候都一定是弱平稳的。不过由于随机过程的二阶矩未必存在，所以强平稳过程并不一定就是弱平稳过程。例子1.1 中的过程由于存在趋势项 $f(t)$ ，所以既不是强平稳也不是弱平稳的。那么去掉了趋势项之后的情况又是怎样的呢？下面我们来看一个重要的例子。

例子5.2. (随机游走与一阶自回归过程) 考虑两个相似的随机过程 y_t 与 z_t , 分别定义为

$$\begin{aligned}y_t &= y_{t-1} + u_t \\ z_t &= \rho z_{t-1} + u_t,\end{aligned}$$

其中 $\rho \in [0, 1)$, $u_t \stackrel{i.i.d.}{\sim} N(0, 1)$ 。此时我们称 y_t 是一个随机游走, z_t 是一个 $AR(1)$ 过程。

下面我们来分析这两个过程的平稳性。首先假设两个过程都可以回溯到无限期，即 $t_0 \rightarrow -\infty$ 。容易验证， y_t 和 z_t 此时都可以写成 u_t 的滑动平均函数

$$y_t = \lim_{n \rightarrow \infty} \sum_{i=-n}^t u_i$$

$$z_t = \lim_{n \rightarrow \infty} \sum_{i=-n}^t \rho^{t-i} u_i.$$

此时两者均值函数

$$\mathbb{E}[y_t] = \lim_{n \rightarrow \infty} \sum_{i=-n}^t \mathbb{E}[u_i] = 0$$

$$\mathbb{E}[z_t] = \lim_{n \rightarrow \infty} \sum_{i=-n}^t \rho^{t-i} \mathbb{E}[u_i] = 0.$$

y_t 的协方差函数为

$$\begin{aligned} \mathbb{E}[y_t y_{t+h}] &= \lim_{n \rightarrow \infty} \mathbb{E} \left[\left(\sum_{i=-n}^t u_i \right) \left(\sum_{i=-n}^{t+h} u_i \right) \right] \\ &= \lim_{n \rightarrow \infty} \mathbb{E} \left[\sum_{i=-n}^t u_i^2 \right] = \lim_{n \rightarrow \infty} (n + t). \end{aligned}$$

而此时 z_t 的协方差函数为

$$\begin{aligned} \mathbb{E}[z_t z_{t+h}] &= \lim_{n \rightarrow \infty} \mathbb{E} \left[\left(\sum_{i=-n}^t \rho^{t-i} u_i \right) \left(\sum_{i=-n}^{t+h} \rho^{t+h-i} u_i \right) \right] \\ &= \lim_{n \rightarrow \infty} \mathbb{E} \left[\sum_{i=-n}^t \rho^{t-i} u_i^2 \right] = \frac{\rho^h}{1 - \rho^2}. \end{aligned}$$

注意到 y_t 的协方差函数与 t 有关，而 z_t 的协方差函数只与时间间隔 h 相关。所以 AR(1) 过程当 $\rho < 1$ 时弱平稳，而随机游走是非平稳过程。进一步的，我们可以验证 $z_t \sim N(0, [1 - \rho^2]^{-1})$ ，所以当首项可以回溯到无限期之前时，AR(1) 过程还是强平稳的。但是另一方面，当首项只能回溯到有限期时，AR(1) 过程也是非平稳的。原因是在无限期问题中的等比级数现在要用一个有限项的等比数列替代，历史 u_t 的影响无法随 ρ 衰减为零。这也是我们在对平稳随机过程进行数值仿真时，需要取尽可能长的时间序列，并且通常会在样本矩的计算中丢弃初始项的原因。在图 5.1 当中，我们对 y_t 和 z_t 分别仿真了两条样本曲线，两图中蓝色曲线为 y_t ，红色曲线为 z_t ($\rho = 0.7$)。通过左右两图的对比，我们可以明显看到随机游走的波动范围随时间增加而增大，而一阶自回归过程的样本曲线稳定的在一个范围内波动。

遍历性

从图 5.1 当中我们可以从直觉上看出为什么对平稳过程来说，遍历性可以成立。对平稳过程 z_t ，我们抽取的两条样本曲线看上去区别不大，而且当中任何一条曲线都能够“取遍”波动范围内的所有取值

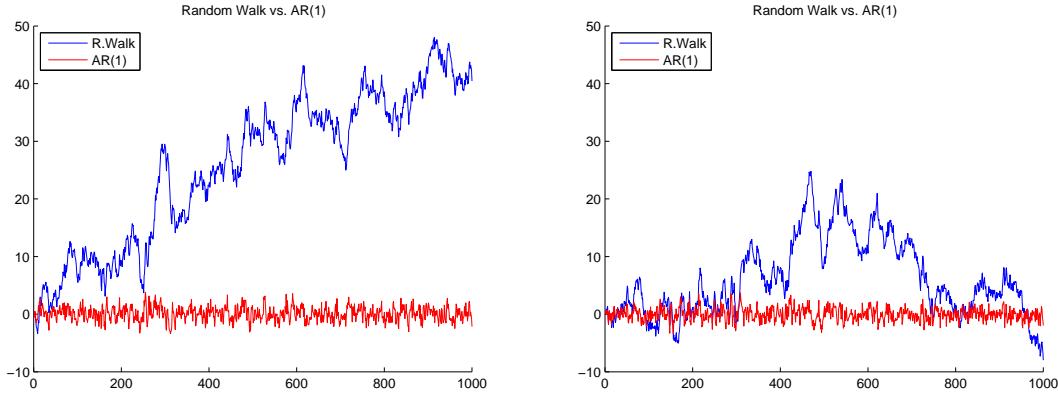


图 5.1.—随机游走与一阶自回归过程

(所以得名“遍历”)^①反过来对于非平稳的 y_t , 两条样本曲线所覆盖的值域是完全不同的, 因此一条较长的样本曲线无法代替多条较短的样本曲线。下面我们严格叙述遍历性的概念。

我们首先定义随机过程 $\{X_t\}$ 沿时间方向的数字特征。

定义5.10. (随机过程沿时间方向的数字特征) 给定随机过程 $\{X_t, t \in \mathcal{T}\}$, 定义积分

$$\mathbb{E}^t[X_t] = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T X_t dt$$

为 $\{X_t\}$ 的时间均值函数。类似的, 对于 $h \in \mathcal{T}$, 定义积分

$$\mathbb{E}^t[X_t X_{t+h}] = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T X_t X_{t+h} dt$$

为其时间相关函数。

注意到由于 X_t 本身是一个随机过程, 所以上述两个积分的取值一般来说是随机的。于是为了介绍遍历性, 我们需要引入依概率收敛(*convergence in probability*)和几乎处处收敛(*convergence almost everywhere*)的概念。

定义5.11. (依概率收敛) 设 $\{X_n\}_{n=1}^\infty$ 是一个随机变量序列, α 为一个常数。若对于任意正数 ε 都有

$$\lim_{n \rightarrow \infty} P\{|X_n - \alpha| < \varepsilon\} = 1,$$

那么我们说序列 $\{X_n\}_{n=1}^\infty$ 依概率收敛于 α , 记为

$$\text{plim}_{n \rightarrow \infty} = \alpha.$$

这里我们不详细展开依概率收敛的概念, 读者可以直观的将其理解为虽然由于随机性的存在, $\lim_{n \rightarrow \infty} X_n = \alpha$ 未必精确成立, 但是不成立的概率很小。如果这个不成立的概率小到可以忽略不计, 那么我们就说 $\{X_n\}_{n=1}^\infty$ 几乎处处收敛于 α 。严格的书写成为

^① 这里取遍打了引号是因为严格来说, AR(1)过程的值域依然是全体实数集, 只不过取极端值的概率很小而已。

定义5.12. (几乎处处收敛) 设 $\{X_n\}_{n=1}^{\infty}$ 是一个随机变量序列, α 为一个常数。若

$$P\left(\lim_{n \rightarrow \infty} X_n = \alpha\right) = 1,$$

那么我们说序列 $\{X_n\}_{n=1}^{\infty}$ 几乎处处收敛于 α , 记为

$$\lim_{n \rightarrow \infty} X_n \stackrel{a.e.}{=} \alpha.$$

不过除此以外, 依概率收敛与几乎处处收敛的直观意义与一般的极限没有明显区别。有了几乎处处收敛的概念, 我们现在来定义遍历性。

定义5.13. (遍历性) 对弱平稳过程 $\{X_t\}$

1. 如果时间均值函数与样本平均函数满足

$$\mathbb{E}^t[X_t] \stackrel{a.e.}{=} \mathbb{E}[X_t] = \mu_X,$$

则称 $\{X_t\}$ 具有均值遍历性。

2. 如果对任意 $h \in T$ 有

$$\mathbb{E}^t[X_t X_{t+h}] \stackrel{a.e.}{=} \mathbb{E}[X_t X_{t+h}] = R_X(h),$$

则称 $\{X_t\}$ 具有自相关函数遍历性。

3. 当 1 与 2 都成立时, 我们说 X_t 是弱遍历过程。

从定义中可以看出, 遍历性实际上就是大数定律在随机过程中的推广。上述遍历性的定义是针对弱平稳过程的, 其对二阶矩存在的强平稳过程同样成立, 但是一般强平稳过程不一定成立。不过由于宏观经济学中很少研究强平稳过程, 这里不做讨论, 有兴趣的同学可以参看随机过程的相关教材。^①

为叙述完整性考虑, 这里给出平稳过程具备遍历性的充要条件即遍历性定理。不过由于在宏观经济学当中我们很少能够得到随机过程的解析表达式, 所以在实际分析当中具体应用到遍历性定理的机会并不多。不过对于定理内容的叙述还是能够帮助我们了解遍历性的直观含义。另外我们略去定理的证明, 有兴趣的同学可以参考 [Karlin and Taylor \(1975\)](#)。对平稳过程的均值函数, 我们有

定理5.1. (均值遍历性定理) 平稳过程 $\{X_t\}$ 的均值具有遍历性的充分必要条件是

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^{2T} \left(1 - \frac{h}{2T}\right) [R_X(h) - \mu_X^2] dh = 0.$$

而对平稳过程的自相关函数, 我们有

定理5.2. (自相关函数遍历性定理) 平稳过程 $\{X_t\}$ 的自相关函数 $R_X(h)$ 具有遍历性的充分必要条件是

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^{2T} \left(1 - \frac{\tau}{2T}\right) [B(\tau) - R_X^2(h)] d\tau = 0,$$

其中

$$B(\tau) = \mathbb{E}[X(t)X(t+h)X(t+\tau)X(t+h+\tau)].$$

^① 读者可以参考 [Karlin and Taylor \(1975\)](#)。

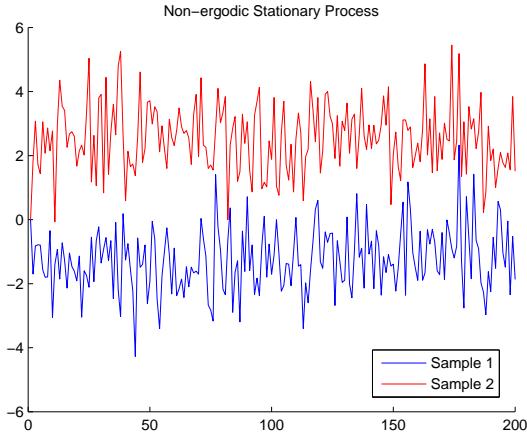


图 5.2.—非遍历的平稳过程

上述两个定理成立的条件其实是相当宽泛的，所以实践中我们碰到的随机过程绝大多数都是遍历的。直观上来说，遍历性是说自协方差函数 $C_X(h) = R_X(h) - \mu_X^2$ 的部分和序列渐进等于零。以均值遍历性定理为例，这也就是说，对一个遍历过程，如果刚开始随机过程更多的取到了比均值大的值，那么随时间的推移，随后一定会取到等量的比均值小的值，从而遍历随机过程的值域。由于前提假设是过程是弱平稳的，所以这个均值不随时间变化，当均值上下的取值量相近时，时间维度上的样本能够给出均值的渐进一致估计。

本小节的最后，我们来看一个非遍历平稳过程的例子。

例子5.3. (非遍历平稳过程) 设随机过程

$$y_t = \mu + u_t,$$

其中 $\mu \sim N(0, \lambda^2)$, $u_t \sim N(0, \sigma^2)$ 。

容易验证 $\mathbb{E}[y_t] = 0$ 和 $\mathbb{E}[y_t y_{t+h}] = \lambda^2$ ，所以 y_t 是弱平稳的。但是 y_t 在时间维度上的积分（这里是求和）为

$$\begin{aligned} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T y_t &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T (\mu + u_t) = \mu + \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T u_t \\ &= \mu \neq 0, \end{aligned}$$

于是该过程不是遍历的。直观上，这是因为 $\{y_t\}$ 的随机性来源于 μ 和 u_t 两个渠道，但是两者中只有 u_t 是与时间相关的， μ 是在初始期给定的。所以无论单条样本曲线有多长，来源于 μ 的那些随机性都不可被穷尽。因此要完整刻画该过程的数字特征，必须取多条样本曲线来反映蕴含于 μ 当中的随机性。图5.2以两条样本曲线为例阐述了这个原因。

5.2 马尔科夫过程

熟悉了随机过程的一般理论之后，下面我们来讨论一类特殊的随机过程——马尔科夫过程。我们

在第三章增长理论的讨论中看到，新古典的增长模型可以写成由值函数表示的递归形式，递归形式的一个重要结构是所有确定未来期决策所需要的信息都能够被一个状态变量——本期的资本存量——所概括，而与任何前期历史信息无关。上一章当中的经济周期理论中，我们对增长模型进行了扩展，引入了随机的生产率冲击。特别的，我们假设生产率冲击也具有未来期的分布与历史信息无关的性质，所以两者相结合，真实经济周期模型自然也满足这个特性。数学上，这种当随机过程在时刻 t_0 所处状态的已知的条件下，过程在时刻 $t > t_0$ 所处状态的条件分布与 t_0 之前过程所处状态无关的性质叫做马尔科夫性或无后效性。本节我们具体学习马尔科夫过程的性质。我们首先介绍一下马尔科夫过程的一般理论，随后我们介绍一种在宏观经济学当中常用的对马尔科夫过程进行数值逼近的方法——离散马尔科夫链近似[Tauchen (1986)]。

5.2.1 马尔科夫链的基本概念

首先我们来看马尔科夫性的数学定义。

定义5.14. (马尔科夫性) 设随机过程 $\{X_t, t \in \mathcal{T}\}$ ，若对于 n 个不同时刻 $t_1, \dots, t_n \in \mathcal{T}$ 和 $\{X_t\}$ 在这些时刻的取值 $x_1, \dots, x_n \in \mathcal{X}$ 的分布函数有

$$P(X_{t_n} \leq x_n | X_{t_1} \leq x_1, \dots, X_{t_{n-1}} \leq x_{n-1}) = P(X_{t_n} \leq x_n | X_{t_{n-1}} \leq x_{n-1}),$$

则称随机过程 $\{X_t, t \in \mathcal{T}\}$ 是一个马尔科夫过程。

容易验证，上一章当中我们假设的一阶自回归生产率冲击过程

$$\log A_{t+1} = \rho \log A_t + \varepsilon_{t+1},$$

就是一个马尔科夫过程。

当 \mathcal{T} 和 \mathcal{X} 都是可数集时，我们称对应的马尔可夫过程为马尔科夫链。在宏观经济学当中，我们主要处理的将是状态空间 \mathcal{X} 为有限集的情形。当状态空间由有限个状态组成时，显然马尔科夫链描述的是过程在各个状态之间的相互转移，这是通过转移概率矩阵(transition matrix)来描述的。

定义5.15. (转移概率矩阵) 设 $\{X_t\}$ 是马尔科夫链，状态空间 \mathcal{X} 包含 n 个元素。对任意 $x_i, x_j \in \mathcal{X}$ ，条件概率

$$P(X_{t+1} = x_j | X_t = x_i) \triangleq P_{ij}^{t,t+1}$$

称为马尔科夫链的一步转移概率。当这一概率与 t 无关时，该链称为平稳或齐次马尔科夫链，一步转移概率简记为 P_{ij} 。以 P_{ij} 为 i 行 j 列元素的 n 阶方阵 \mathbf{P} 称为一步转移概率矩阵。

本教程中我们只讨论平稳马尔科夫链，以后简称为马尔科夫链。由于概率是非负的，而过程总要转移到某一个状态当中去，所以一步转移概率矩阵的每行元素相加等于一，即

$$P_{ij} \geq 0, \quad \text{且} \quad \sum_{j=1}^n P_{ij} = 1.$$

对于一般的 n 步转移概率矩阵，有著名的切普曼—柯尔莫哥洛夫(Chapman-Kolmogorov)方程成立：

定理5.3. (CK方程) 设马尔科夫过程的一步转移概率矩阵为 \mathbf{P} ，则 n 步转移概率矩阵 $\mathbf{P}(n)$ 可以表示成一步转移概率矩阵的乘积

$$\mathbf{P}(n) = \mathbf{P}^n,$$

$\mathbf{P}(n)$ 中的元素记为 P_{ij}^n 。

这个定理是说马尔科夫链的任意转移概率矩阵都可以由一步转移概率阵出发构造，于是马尔科夫链的统计性质完全由初始分布和一步转移概率矩阵刻画。这说明一步转移概率矩阵是马尔科夫过程的本质性质，因此以后我们简称其为转移概率矩阵。另外，回忆上一小节平稳性的定义，容易验证当转移概率不随时间变化时，马尔科夫链是平稳的。

5.2.2 马尔科夫链的状态分类*

按照状态在过程不断转移当中被实现的特征，我们可以对马尔科夫链的状态进行分类，本节当中对这些类进行刻画。具体来说，我们将关注状态的两类特征。一是两个状态之间是否能够通过有限步相互转移，即互通性(*communicable*)。二是从一个状态出发经过有限步转移是否能够回到该状态，即常返性(*recurrence*)。预先指出的是，我们在宏观经济学当中遇到的马尔科夫链都是后文提到的性质较好的正则(*regular*)马尔科夫链，所以本节当中介绍的比较良好的那些性质在实际应用当中都是满足的，而所对应的那些具有不良性质的反例大多需要通过转移阵有分块零元素或者状态空间为可数无限集的特殊马尔科夫链来构造。尽管关于这些分类的讨论在实际问题当中并没有直接的应用，作者在此认为对一些基本概念的掌握还是能够加深对下一小节当中平稳分布的理解。如果读者对抽象概念的理解实在感到困难，跳过本节不会影响后文的阅读。

互通性

定义5.16.（互通性）若对整数 $n \geq 0$ ，有 $P_{ij}^n > 0$ ，则称状态 j 从状态*i*是可达的(*accessible*)，记作 $i \rightarrow j$ 。它表示从状态*i*起，经过有限步的转移可以到达状态*j*。两个相互可达的状态称为互通的，记作 $i \leftrightarrow j$ 。

互通性是一个数学上的等价关系，即它满足自反性 ($i \leftrightarrow i$)、对称性 ($i \leftrightarrow j \Rightarrow j \leftrightarrow i$) 和传递性 ($i \leftrightarrow j, j \leftrightarrow k \Rightarrow i \leftrightarrow k$)，这三个性质的含义与证明由互通性的概念都是显然的。由于传递性的存在，所以如同其他数学中的等价关系，互通性把马尔科夫链的状态分割成互不相交的子类。直观上，我们把能够相互转移的状态归为一类。如果马尔科夫链的所有状态之间都相互连通，我们就说这个马尔科夫链是不可约的(*irreducible*)。以下述矩阵为例，

$$\begin{bmatrix} \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

以上述矩阵为转移阵的马尔科夫链，状态{1, 2}和{3, 4, 5}构成了两个由互通关系刻画的等价类。

另一个相关的概念是周期性。

定义5.17.（周期性）设*i*是马尔科夫链的一个状态，使 $P_{ii}^n > 0$ 的所有正整数 $n \geq 1$ 的最大公约数称为状态*i*的周期，记作 $d(i)$ 。如果对所有 $n \geq 1$ ，都有 $P_{ii}^n = 0$ ，即从*i*状态出发永远无法返回，那么约定此时周期为 ∞ ；另一方面，若 $d(i) = 1$ ，我们说*i*是非周期的。

通常而言，我们所说的周期性是一个事件间隔多少时间必定发生一次，马氏链的周期性含义是相似的。不过由于马氏链本身具有随机性，所以事件的发生并不是必然的。在此处，周期性说的是从*i*状态出发，如果想要重复观测到*i*，至少要每隔多少时间回来观测一次。一个简单的例子是钟表。如果我

们记钟表上每一个数字为一个马氏链的状态（此时转移阵是近似对角的），那么任意一个状态的周期是12，这是说从某个时刻出发，每12小时回来观测一次，就可以保证不遗漏钟表再次经过这个时刻。再看一个略微复杂一些的例子，设四个状态($i = 1, \dots, 4$)的马氏链有如下转移阵

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix} \quad (5.1)$$

不难算出对于状态1而言，使得 $P_{11}^n > 0$ 的 $n = 4, 6, 8, \dots$ ，所以依照周期性的定义，状态1的周期是2。这是说，如果从状态1出发，想要不遗漏对于状态1的重复观测，每2期就要重复观测一次。反过来，这是说如果我们从状态1出发，在比如第4期没有观测到状态1，那么第五期一定不可能观测到状态1。而对于非周期的马氏链状态来说，在任何时间点，该状态都是有可能发生的。周期性的概念对于理解马氏链的极限性质是很有帮助的。不难看出，对由(5.1)刻画的马氏链而言， $\lim_{n \rightarrow \infty} P_{j1}^n$ 不存在。进一步的，我们有下面的命题。

命题5.1. 令 \mathbf{P} 为不可约、非周期的有限状态马尔科夫链的转移概率矩阵，那么一定存在 N ，使得当 $n \geq N$ 时， n 步转移概率矩阵 \mathbf{P}^n 的所有元素非零。

特别的，我们将存在 n 使得 \mathbf{P}^n 所有元素非零的马尔科夫链称为正则马尔科夫链。正则马尔科夫链的一个重要性质时其转移阵的极限 $\lim_{n \rightarrow \infty} \mathbf{P}^n$ 总是存在，我们将在下一小节当中详细介绍这个极限。最后，容易验证，对转移阵(5.1)，当 n 为奇数和偶数时，当 n 足够大时对应的转移阵分别为

$$\begin{bmatrix} 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \end{bmatrix} \quad \text{和} \quad \begin{bmatrix} \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \end{bmatrix},$$

于是对于周期性马尔科夫链，转移矩阵的极限不存在。从这个例子我们还可以看出，尽管我们只验证了状态1的周期为2，但实际计算显示，其他三个状态的周期也为2。注意到转移阵(5.1)是不可约的，所以我们猜想从属于同一个等价类的状态其周期应该是相同的。这个猜想是正确的，由下述命题给出：

命题5.2. (互通状态的周期性) 如果状态 $i \leftrightarrow j$ ，那么 $d(i) = d(j)$ 。

常返性

一个与周期性相联系的概念是常返性，常返性用来描述从一个状态出发时候是不是有可能在有限步之内返回。为严格叙述这个概念，我们首先定义首达概率 f_{ij}^n ，它表示从*i*状态出发，在*n*步时首次到达*j*状态的概率，用式子表达就是

$$\begin{aligned} f_{ij}^0 &= 0, \\ f_{ij}^n &= P(X_n = j, X_k \neq j, k = 1, \dots, n-1 | X_0 = i). \end{aligned}$$

约定当没有出发时， $f_{ij}^0 = 0$ 。从上面的定义可以看出， f_{ii}^n 表示从状态*i*出发，在第*n*次转移时首次回到状态*i*的概率。进一步的，我们可以定义

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^n,$$

f_{ij} 描述了从*i*出发最终达到状态*j*的概率。由互通性的定义，显然有 $i \leftrightarrow j \Leftrightarrow f_{ij} > 0$ 。现在我们可以给出常返性的定义：

定义5.18. (常返性) 如果 $f_{ii} = 1$ ，我们称状态*i*是常返的。反之，若 $f_{ii} < 1$ ，则称状态*i*为瞬时的(*transient*)。

从常返性的定义可以得出用转移概率对进行判别常返态的一个准则：

定理5.4. (常返态的判别准则) 状态*i*常返的充分必要条件是

$$\sum_{n=1}^{\infty} P_{ii}^n = \infty.$$

反过来，状态*i*瞬时，当且仅当

$$\sum_{n=1}^{\infty} P_{ii}^n < \infty.$$

从上述定理可以直接验证，若状态*j*与常返态互通，则*j*必常返。

对常返态，除了首达概率，另一个重要的概念是由*i*出发之后，首次回到*i*所需要的时间，即首达时间。定义首达时间 T_i 为

$$T_i = \min\{n \geq 1 : X_n = i\},$$

而首达概率用 T_i 表达，可以写成

$$f_{ii}^n = P(T_i = n | X_0 = i).$$

于是 T_i 的数学期望就是

$$\mathbb{E}[T_i] = \sum_{n=1}^{\infty} n f_{ii}^n \triangleq \mu_i.$$

当 $\mu_i < \infty$ 时，我们说状态*i*是正常返的，而若 $\mu_i = \infty$ ，我们说状态*i*是零常返的。可以证明，对有限马尔科夫链，若一个状态是常返的，那么它必然是正常返的。零常返状态仅在可数状态马尔科夫链中存在，而宏观经济学当中我们不会接触到后者。

5.2.3 马尔科夫链的极限定理与平稳分布

考虑一个只有两个状态的正则马尔科夫链 $\{X_t\}$ ，转移概率矩阵为

$$\mathbf{P} = \begin{bmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix}.$$

不难验证， \mathbf{P}^n 的极限存在：

$$\lim_{n \rightarrow \infty} \mathbf{P}^n = \begin{bmatrix} 2/3 & 1/3 \\ 2/3 & 1/3 \end{bmatrix}. \quad (5.2)$$

注意到矩阵(5.2)的行向量是完全相同的，回忆转移阵的定义，这是说无论初始分布是什么，经过足够长的时间之后，转移到状态1和2的概率固定为 $2/3$ 和 $1/3$ 。换句话说，无论初始分布中状态1和2分别有多少人，两者中总是有 $2/3$ 转移到状态1， $1/3$ 转移到状态2。由于初始分布中状态1和2的概率相加必然为

一，所以对(5.2)，任意初始分布经过足够长时间之后得到的极限分布都是 $[2/3, 1/3]$ 。直观上，如果我们设矩阵 \mathbf{Q} 代表极限矩阵

$$\mathbf{Q} = \lim_{n \rightarrow \infty} \mathbf{P}^n,$$

那么对任意初始分布 $\mathbf{v}_0 = [m_1, m_2]$ ，我们有

$$\mathbf{v}_0 \times \mathbf{Q} = \begin{bmatrix} 1 \rightarrow 1+2 \rightarrow 1 \\ 1 \rightarrow 2+2 \rightarrow 2 \end{bmatrix}' = \begin{bmatrix} 2/3 \\ 1/3 \end{bmatrix}'.$$

我们把矩阵(5.2)的行向量单独提取出来记为

$$\boldsymbol{\pi} = [\pi_1, \pi_2].$$

由于 $\boldsymbol{\pi}$ 是一个极限性质，所以当时间足够长的时候，开始阶段有限期的概率对样本总概率的影响会被冲淡，因此对一条样本曲线来说，总是有 $2/3$ 的样本取状态1， $1/3$ 取状态2。而对于大量样本曲线来说，任何时刻总是有 $2/3$ 的曲线处在状态1，另外 $1/3$ 的曲线处在状态2。回忆上一节中我们关于平稳性的讨论，这种横截面与时间序列数字特征的等价性正是平稳过程的遍历性。正因为这个原因，这个例子中由矩阵 \mathbf{P} 描述的马氏链是遍历的，同时 $\boldsymbol{\pi}$ 称为马氏链的平稳分布。进一步可以验证，对由首达时刻期望刻画的平均返回时间 μ_i 来说，我们有 $\mu_i = 1/\pi_i$ 。这个等式的含义是直观的，由于极限状态下，转移到状态1的平均概率是到状态2的两倍，所以状态1的平均常返时间是状态2的一半，所以在足够大的取样下，我们观测到的状态1的数量将是状态2的两倍。

现在我们将上述例子所反映的直觉进行严格的表达。注意本节当中的定理对可数状态的马尔科夫链也是成立的。我们首先看马尔科夫链的基本极限定理。

定理5.5. (马尔科夫链的基本极限定理) 对于一个马尔科夫链，

1. 若状态*i*是瞬时的或零常返的，那么

$$\lim_{n \rightarrow \infty} P_{ii}^n = 0.$$

2. 若状态*i*是以*d*为周期的常返态，那么

$$\lim_{n \rightarrow \infty} P_{ii}^{nd} = \frac{d}{\mu_i}.$$

3. 若状态*i*是非周期的正常返态，那么

$$\lim_{n \rightarrow \infty} P_{ii}^n = \frac{1}{\mu_i}.$$

这个定理刻画了各种不同的状态在极限当中的性质。换个角度来看，这也是在长期观测当中，各种状态被观测到的渐进概率。注意到定理第三款实际上刻画了平稳过程的遍历性，所以在马尔科夫链的情形下，非周期和正常返性可以用来判断状态的遍历性。

下面再看马尔科夫链的平稳分布。

定义5.19. (平稳分布) 设马尔科夫链有转移概率阵 $\mathbf{P} = (P_{ij})$ ，若概率分布 $\boldsymbol{\pi} = \{\pi_i\}$ 满足

$$\pi_j = \sum_{i=1}^N \pi_i P_{ij},$$

则称 $\boldsymbol{\pi}$ 是马尔科夫链的平稳分布。

平稳分布直观上是说当马氏链的状态进入平稳分布之后，系统的状态不发生改变。从横截面的角度出发，这是说处于每个状态中的样本曲线的比例总是保持在平稳分布所描述的水平上。此时初始分布对马氏链的后续变化不再起作用，与马尔科夫性相结合，我们知道 $\{X_t\}$ 的有限维联合分布函数族与 t 无关，于是我们得到一个重要的结论即从平稳分布出发的马尔科夫链是强平稳的。又因为马尔科夫链的二阶矩存在，所以同时它也是弱平稳的。定义5.19实际上还揭示了一种计算平稳分布的方法，由下面的定理给出。

定理5.6. (极限转移阵与平稳分布) 马尔科夫链的极限转移概率矩阵与平稳分布有如下的等价关系

1. 若一个不可约马尔科夫链中的所有状态都是遍历的，那么对于任意 i, j ，极限 $\lim_{n \rightarrow \infty} P_{ij}^n = \pi_j$ 存在，且 $\pi = \{\pi_i\}$ 即为该过程对应的平稳分布。即此时 π 满足平稳分布的定义

$$\begin{aligned}\sum_i \pi_i P_{ij} &= \pi_j, \\ \sum_j \pi_j &= 1.\end{aligned}\tag{5.3}$$

2. 反之，若一个不可约马尔科夫链存在唯一的平稳分布，且这个马尔科夫链的所有状态都是遍历的，那么该平稳分布就是这一马尔科夫链的极限分布。即对任何*i*都有

$$\lim_{n \rightarrow \infty} P_{ij}^n = \pi_j.$$

从定理5.6还可以看出，判断平稳分布是否唯一事实上等价于线性方程组(5.3)是否有唯一解。从计算负荷的角度出发，定理5.6提供了一个将大量矩阵乘法运算化简为一个简单线性方程组的途径，这在早期应用中能够极大的简化计算。由于计算机硬件的发展，求一个马尔科夫链的平稳分布可以直接将转移矩阵取高次幂得到，所以这个定理在计算负荷方面的用途在现代应用中不如早期那么重要。不过在判断马氏链是否对应唯一的平稳分布时，定理的第一款还是给我们提供了一个相当方便的方法。不过如我们马上要看到的，宏观经济学当中所使用的马尔科夫链通常是对另一个马尔科夫过程的近似，此时的转移概率矩阵一般而言不存在零元素，故而是正则的。对正则马尔科夫链而言，平稳分布存在且唯一。

5.3 马尔科夫过程的离散近似

系统的了解了随机过程相关的概念之后，让我们再来看上一章中的递归问题(4.20)，为叙述方便，在此重新写出该优化问题

$$\begin{aligned}v(k; A) &= \max_{c, k', n} \{u(c, 1 - n) + \beta \mathbb{E}[v(k'; A') | A]\} \\ s.t. \\ c + k' - (1 - \delta)k &= AF(k, n), \\ \log A' &= \rho \log A + \varepsilon'.\end{aligned}\tag{5.4}$$

如果我们将生产率冲击从AR(1)过程替换成一个简单的独立同分布二元过程：

$$A = \begin{cases} A_H, & \text{概率为0.5} \\ A_L, & \text{概率为0.5} \end{cases}\tag{5.5}$$

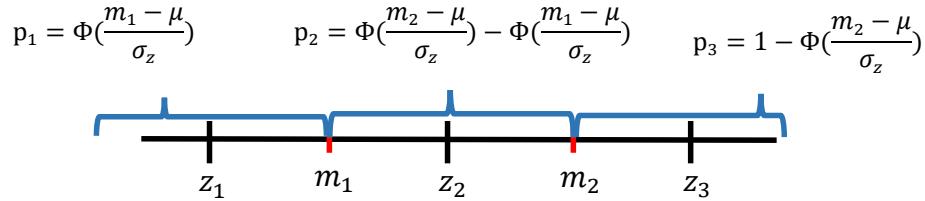


图 5.3.—马尔科夫链近似随机过程

那么优化问题(5.4)就可以用我们熟悉的值函数迭代的方法来求解。此时唯一的区别是值函数 v 成为二维的（或者理解为 v_H, v_L 两个值函数）。但是我们依然可以从零函数开始迭代，第一步时分别对 $A = A_H$ 和 $A = A_L$ 求解最优化问题。第二步递归计算时，对 k' 在每个节点的取值，期望算子无非就是把该节点对应的 v_H 和 v_L 的取值依照概率分布去算术平均。当我们得到 $\mathbb{E}[v(k'; A')]$ （注意i.i.d简化之后这里没有条件期望）的取值（这是一个数）之后，我们可以分别对 $A = A_H$ 和 $A = A_L$ 两种情形求解第二轮递归时的最优化问题，并对值函数进行更新。以此类推直到值函数收敛时，我们就可以对 $A = A_H$ 和 $A = A_L$ 求出两组政策函数。注意到(5.5)是一个马尔科夫链。所以如果我们能想办法将(5.4)中的AR(1)生产率冲击转化成一个马尔科夫链，优化问题(5.4)的求解就是一个步骤上略为繁琐的动态优化问题。但是其算法的本质与确定性的增长模型没有区别。本节当中我们就介绍一种经典的用离散马尔科夫链近似AR(1)过程的算法，该方法最早是由Tauchen (1986)提出的。

5.3.1 独立同分布随机过程的近似

我们不妨先考虑一个简单的独立同分布情形。我们的目标是使用 n 个状态的离散马尔科夫链近似随机过程 $\{X_t\}$ ，其中每一期 $X_t \sim N(\mu, \sigma_z^2)$ 。由于 X_t 是独立同分布的，所以马尔科夫链的转移阵的所有行向量相同，于是我们需要的就是确定马尔科夫链的状态 z_i 和每个 z_i 发生的概率 p_i 。直观上， z_i 的取值要尽可能的涵盖正态分布的定义域，至少要覆盖那些比较重要的值。而对这些选定的 z_i ， p_i 应能尽可能的反映正态分布的概率特征，比如对比较靠近均值 μ 的 z_i ， p_i 的取值就应该比较大，而对比较靠近尾部的 z_i ， p_i 最好比较小。一个直观的办法是对 $\{z_i\}_{i=1}^n$ ，将正态分布当中那些距离 z_i 比较近的点用 z_i 近似，而 z_i 的概率密度就是所有这些它代表的点的概率密度的和。下面的伪代码具体介绍了实施上述想法的步骤，图5.3提供了一个算法的图形表达。

1. 设置 n ，即用来近似的马氏链的状态总数。
2. 选一个 λ ，确定 z 取值的上下界为

$$\bar{z} = \mu + \lambda\sigma, \\ \underline{z} = \mu - \lambda\sigma.$$

3. 令 $z_1 = \underline{z}, z_n = \bar{z}$ ，余下的 z_i 等分 $[\underline{z}, \bar{z}]$ 。
4. 构造被 z_i 等分的区间中点（共 $n - 1$ 个）

$$m_i = \frac{z_{i+1} - z_i}{2}$$

5. 对 $z_i, i = 2, \dots, n - 1$, 设其被用来近似周围两个区间中点之间的点, 于是若我们记标准正态分布的累积密度函数为 $\Phi(\cdot)$, 那么对这些节点我们有

$$p_i = \Phi\left(\frac{m_i - \mu}{\sigma_z}\right) - \Phi\left(\frac{m_{i-1} - \mu}{\sigma_z}\right).$$

而对区间的两个端点 z_1 和 z_n , 我们除了临近区间的中点外, 所有尾部的点也被它们代表, 于是端点处的概率设为

$$p_1 = \Phi\left(\frac{m_1 - \mu}{\sigma_z}\right)$$

$$p_n = 1 - \Phi\left(\frac{m_{n-1} - \mu}{\sigma_z}\right).$$

至此我们就确定了节点 $\{z_i\}_{i=1}^n$ 和对应的概率 $\{p_i\}_{i=1}^n$ 。

关于这个算法我们有三点评注。一是这个算法当中, 节点的数目 n 和取值范围 λ 是可以自由选取的。显然, 节点数目越多计算的精度越高。但是注意到当应用与优化问题(5.4)时, 对于每个节点 z_i , 都需要求解一次值函数, 所以太多的节点即便对现代计算机也是很大的计算负担。实践中, 大多数研究者根据计算机处理速度与所求解的问题的复杂程度从3到15的奇数。经验上, 取15个点以上不会对计算精度造成明显影响。至于 λ , 通常对于正态分布我们取 $\lambda = 3$, 这源于概率论的知识, 对正态分布而言, 正负三个标准差之间涵盖了累积概率为99% 的值。第二是注意到 p_i 的取值源自正态分布概率的计算, 但是 z_i 的取值简单的取了等分点。细心的读者很自然的会联想到, 有没有可能选取一些特定的 z_i , 使得在 n 不变的情形下, 提高计算精度? 答案是肯定的, 这正是我们在下一节当中数值积分的介绍中将要讨论的 [Tauchen and Hussey \(1991\)](#) 求积公式算法。不过依作者本人的经验, 在大多数宏观经济学研究所要求的精度范围内, 近似方法的不同不会对文章的结论产生显著影响。如果读者的程序对于不同的近似算法给出显著不同的结果, 通常这更有可能是由于程序的其他漏洞而非近似误差造成的。第三, 上述算法并不局限于正态分布。实际上, 只要我们能够计算 p_i , 无论是通过概率分布的解析表达式还是非参数的数值表达式, 我们都可以应用类似的算法构造一个马尔科夫链进行对其进行近似, 从而使这个算法具有相当的泛用性。

下面我们来看一个简单的例题。

例子5.4. 试用马尔科夫链近似正态分布 $N(0, 2)$, 并比较不同 n 和 λ 对计算精度的影响。

对 λ 我们分别取2和3, 而对 n 我们分别取3和7。对每组 λ, n 的组合, 我们计算近似的马尔科夫链的均值和方差, 并与已知的正态分布的理论值进行对比。实现本节算法的代码包含在代码5.1当中。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 %
4 %
5 % Purpose:
6 % - This code approximates i.i.d normal distribution with a discrete
7 % Markov Chain.
8 % - Code 5.1 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
```

表 5.1
利用马尔科夫链近似 $N(0, 2)$ 分布

| | $\lambda = 2$ | | $\lambda = 3$ | |
|----|---------------|---------|---------------|---------|
| | $n = 3$ | $n = 7$ | $n = 3$ | $n = 7$ |
| 方差 | 2.5385 | 1.9277 | 2.4051 | 2.1600 |

```

14 % Date: Description of Changes
15 % =====
16 % 11/02/2016: Original Code
17
18 clear;
19 % -----
20 % DATA DICTIONARY
21 % -----
22 % Parameterizations
23 mu = 0;           % Mean of Normal Distribution
24 sigma = 2^.5;      % Std.dev of Normal Distribution
25 lambda = 3;        % Range of z
26 n = 7;            % Number of nodes
27
28 % Initialization
29 zlow = mu - lambda*sigma;    % Set endpoints of z
30 zhight = mu + lambda*sigma;
31 z = linspace(zlow,zhigh,n); % z nodes
32 p = zeros(n,1);             % z probability
33 m = zeros(1,n-1);          % Midpoints
34 for ind = 1:1:n-1
35     m(ind) = (z(ind) + z(ind+1))/2;
36 end
37
38 % -----
39 % CALCULATE PROBABILITY
40 % -----
41 p(1) = normcdf(m(1),mu,sigma);      % Endpoints
42 p(n) = 1 - normcdf(m(n-1),mu,sigma);
43 % Inner points
44 for ind = 2:1:n-1
45     p(ind) = normcdf(m(ind),mu,sigma) - normcdf(m(ind-1),mu,sigma);
46 end
47
48 % -----
49 % CALCULATE STATISTICS
50 % -----
51 mean_z = z*p ;
52 var_z = z.^2*p;
53 disp('Parameters: lambda n')
54 disp([lambda n])
55 disp('Mean:')
56 disp(mean_z)
57 disp('Variance')
58 disp(var_z)

```

代码 5.1.—利用马尔科夫链近似正态分布

程序运行的结果汇总在表5.1当中。从表中可以看出，随着节点数量的提高，方差的逼近精度有显著的提高。^①但是计算精度并不随 λ 的提高单调变化。这有两个原因，一是对正态分布而言， $\lambda = 2$ 也可以涵盖95%的取值，与 $\lambda = 3$ 具有同样的代表性。二是对同样数量的节点， λ 较大时，原分布概率密度较高的部分分到的节点数量会减少，这会导致计算精度的下降。一般来说，在实际应用当中，针对不同的问题 λ 和 n 的取值需要反复尝试。

5.3.2 一阶自回归过程的近似

一阶自回归过程的近似与独立同分布过程大体相同，不同之处主要是此时转移概率矩阵是一个确实的矩阵，而不是独立同分布情形下退化为一个向量。转移概率 p_{ij} 的计算也要相应的微调。具体来说，此时节点 z_j 处的概率还取决于过程是从哪一个节点 z_i 进行转移的。

具体的，我们现在的目标是用一个马尔科夫链近似一个AR(1)过程

$$z' = (1 - \rho)\mu + \rho z + \epsilon', \quad \epsilon' \stackrel{i.i.d.}{\sim} N(0, \sigma_\epsilon^2). \quad (5.6)$$

仿照例子5.2的处理，将(5.6)写成滑动平均的形式

$$z_{t+1} = \mu + \sum_{i=0}^{\infty} \rho^i \epsilon_{t+1-i}.$$

于是由正态分布的性质，我们知道 z_t 的平稳分布服从

$$z_t \sim N\left(\mu, \frac{\sigma_\epsilon^2}{1 - \rho^2}\right) \triangleq N(\mu, \sigma_z^2).$$

由于我们最终的目的是用马尔科夫链分析 z_t 的平稳分布，所以 z_i 的取点要尽量覆盖 z_t 平稳分布的值域。有了 z_t 的平稳分布，我们就可以写出适用于AR(1)过程的伪代码。

1. 设置 n ，即用来近似的马氏链的状态总数。
2. 选一个 λ ，确定 z 取值的上下界为

$$\bar{z} = \mu + \lambda \sigma_z, \\ \underline{z} = \mu - \lambda \sigma_z.$$

3. 令 $z_1 = \underline{z}, z_n = \bar{z}$ ，余下的 z_i 等分 $[\underline{z}, \bar{z}]$ 。
4. 构造被 z_i 等分的区间中点（共 $n - 1$ 个）

$$m_i = \frac{z_{i+1} - z_i}{2}$$

5. 对 $z_i, i = 2, \dots, n - 1$ ，还是设其被用来近似周围两个区间中点之间的点。注意到此时给定 z_t ，

$$\epsilon_{t+1} = z_{t+1} - (1 - \rho)\mu - \rho z_t$$

^①均值的逼近由节点构造方式必然为零，因此省略。

服从正态分布 $N(0, \sigma_\epsilon^2)$ 。记标准正态分布的累积密度函数为 $\Phi(\cdot)$ 。所以此时对内区间而言，当 t 期状态为 z_i 时， $t+1$ 期的 z_j 落在 $[m_{j-1}, m_j]$ 之间的概率就是

$$p_{ij} = \Phi\left(\frac{m_j - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right) - \Phi\left(\frac{m_{j-1} - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right),$$

此时 p_{ij} 恰好就是从 z_i 到 z_j 的转移概率。

6. 类似的，对区间的两个端点 z_1 和 z_n ，我们有

$$\begin{aligned} p_{i1} &= \Phi\left(\frac{m_1 - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right) \\ p_{in} &= 1 - \Phi\left(\frac{m_{n-1} - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right). \end{aligned}$$

至此我们就确定了节点 $\{z_i\}_{i=1}^n$ 和对应的转移概率矩阵 \mathbf{P} 。我们关于独立同分布情形下这个算法的评注在AR(1)的情形下也是适用的，这里不做重复。需要提请读者留意的是，(5.6)的滑动平均表示服从正态分布这个特征，源自于随机干扰项 ϵ 的分布本身是正态的。由于每个 z_t 都是正态分布的，所以(5.6)又称为高斯(Gauss)过程。因为正态分布的性质可以由均值和方差完全确定，所以弱平稳的高斯过程一定是最强平稳的。

下面我们重现一下Tauchen (1986)当中的例子。

例子5.5. [Tauchen (1986)] 使用马尔科夫链对一阶自回归过程

$$y_t = \rho y_{t-1} + \epsilon_t, \epsilon_t \stackrel{i.i.d.}{\sim} N(0, 0.01),$$

进行近似，并分析不同情形下的近似精度。

我们取 $\lambda = 3$ ，我们分析 $\rho = 0.1, 0.8, 0.9$ 三种情况，而对 n 我们分别取5和9。对每组参数组合的组合，我们计算近似的马尔科夫链的自回归系数与平稳分布的均值，并与原始分布的理论值进行对比。实现本节算法的代码包含在代码5.2当中。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 %
4 %
5 % Purpose:
6 % - This code approximates AR(1) process with a discrete Markov Chain.
7 % - Code 5.2 in the Lecture Notes
8 %
9 % Author:
10 % Xin Tang © Wuhan University, Spring 2016
11 %
12 % Record of Revisions:
13 % Date: Description of Changes
14 % -----
15 % 11/02/2016: Original Code
16
17 clear;
18 %
19 % DATA DICTIONARY
20 %
21 % Parameterizations

```

```

22 mu = 0; % Mean of Normal Distribution
23 sigma_e = 0.1; % Std.dev of Innovation
24 rho = 0.9; % AR(1) parameter
25 lambda = 3; % Range of z
26 n = 9; % Number of nodes
27
28 % Initialization
29 sigma = sigma_e/(1-rho^2)^.5;
30 zlow = mu - lambda*sigma; % Set endpoints of z
31 zhight = mu + lambda*sigma;
32 z = linspace(zlow,zhigh,n); % z nodes
33 P = zeros(n); % Transition Matrix
34 m = zeros(1,n-1); % Midpoints
35 for ind = 1:1:n-1
36     m(ind) = (z(ind) + z(ind+1))/2;
37 end
38
39 % -----
40 % CALCULATE TRANSITION MATRIX
41 %
42 for i = 1:n
43     for j = 2:n-1
44         P(i,j) = normcdf((m(j)-(1-rho)*mu-rho*z(i))/sigma_e) - ...
45             normcdf((m(j-1)-(1-rho)*mu-rho*z(i))/sigma_e);
46     end
47     P(i,1) = normcdf((m(1)-(1-rho)*mu-rho*z(i))/sigma_e);
48     P(i,n) = 1-normcdf((m(n-1)-(1-rho)*mu-rho*z(i))/sigma_e);
49 end
50
51 % -----
52 % CALCULATE STATISTICS
53 %
54 % Calculate the Stationary Distribution
55 % Using Theorem 5.6
56 % Using Cholesky decomposition to solve linear system
57 X = [P' - eye(n); ones(1,n)];
58 Y = X'*[zeros(n,1); 1];
59 R = chol(X'*X);
60 sd = (R\(\Y))'; % Probability of stationary distribution (row vector)
61
62 % Calculate the statistics of the stationary distribution
63 sm = sd*z'; % Mean
64 ssd = sqrt((z-sm).^2*sd'); % Standard Deviation
65 sac = sum(sum((z-sm)'*(z-sm).* (diag(sd)*P)))/ssd^2;
66
67 disp('Parameters: n rho');
68 disp([n rho]);
69 disp('Mean Std.dev Autocorrelation');
70 disp([sm ssd sac]);

```

代码 5.2.—利用马尔科夫链近似AR(1)过程

注意到58到60行应用了楚列斯基(*Cholesky*)分解求解线性方程组，这个算法在求解大规模线性方程组的时候比较稳定，不过在矩阵规模不大的时候，MATLAB 内建的矩阵求逆函数`inv`也可以得到良好的结果。最后，第65行应用了平稳分布的定义求解马氏链所对应的自回归系数。程序运行的结果汇总在表5.2 当中。不出意外，表5.2 中的结果与Tauchen (1986) 的表1 相同。值得注意的是，当 $\rho > 0.9$ 时，离

表 5.2
利用马尔科夫链近似AR(1)分布

| 节点数 | 理论值 | | 近似值 | |
|-----|--------|------------|--------|------------|
| | ρ | σ_z | ρ | σ_z |
| 9 | 0.1 | 0.101 | 0.0998 | 0.1027 |
| 9 | 0.8 | 0.167 | 0.7984 | 0.1762 |
| 9 | 0.9 | 0.229 | 0.8984 | 0.2533 |
| 5 | 0.9 | 0.229 | 0.9315 | 0.2912 |

散马尔科夫链近似的效果会显著下降。原因是当 ρ 接近1时，AR(1)过程的性质会越发与随机游走相似，由于上述算法是针对平稳分布的，由例子5.2中的计算我们知道随机游走过程不是平稳过程，也不存在平稳分布，其值域包括整个实数轴。使用有限个节点来近似一个相当分布相当广泛的过程，效果不佳也就可以理解了。不幸的是，在宏观经济学当中我们处理的自回归过程通常都具有很高的自回归系数，通常大于等于0.95。于是在具体应用当中，上述算法中对应的节点需要反复尝试，以获得与原始分布相近的统计性质。另一种做法常见于消费储蓄问题当中，此时模型当中直接应用随机游走过程，之后利用模型的齐次性将模型的解等价为一个带有独立同分布过程的相关问题，而对于独立同分布问题，上述算法能够得到很高的近似精度。对于高自回归系数自回归过程的处理的进一步讨论，详见Flodén (2008)。一个具体应用的例子可以参考Storesletten, Telmer and Yaron (2004a)。我们在后面关于消费储蓄问题的讨论中将会看到利用齐次性对模型进行化简的例子[Carroll (1997), Gourinchas and Parker (2002)]。

5.4 数值积分

回忆在优化问题(5.4)当中，马尔科夫过程出现在 $\mathbb{E}v(k'; A'|A)$ 当中，当 A 服从AR(1)过程时，该项实际上是

$$\mathbb{E}[v(k'; A'|A)] = \int v(k'; A') f(A'|A) dA', \quad (5.7)$$

其中若记 $\phi(\cdot)$ 是标准正态分布的密度函数，条件分布 $f(A'|A)$ 是

$$f(A'|A) = \phi\left(\frac{\log A' - \rho \log A}{\sigma_\epsilon}\right).$$

如果我们用 z_i 和 p_{ij} 表示用来近似AR(1)过程的离散马尔科夫链，那么积分(5.7)就成为

$$\sum_{i=1}^n v(k'; z_j) p_{ij}. \quad (5.8)$$

所以上一节中我们介绍的方法实际上最终的目的是求解一个数值积分。方程(5.8)实际上使用一列矩形的求和去逼近(5.7)中的积分。注意到在算法当中我们对于节点的选取是任意的，这在数值积分的常用算法中被称为牛顿—科特斯(Newton-Cotes)公式。在下一小节中，我们首先简单介绍一下牛顿—科特斯公式的基本结论。随后在下一节当中，我们介绍当求积节点选取考虑到利用正交多项式(orthogonal polynomial)的结论对计算效率进行提高时的求积方法，即高斯求积(Gaussian Quadrature)公式。最后，我们指出一种利用应用了高斯求积公式的上一节中算法的变种，Tauchen and Hussey (1991)算法。该算法的实现需要牵涉求正交多项式的零点，因此我们仅介绍其基本想法，具体的程序读者可以下载斯德哥尔摩大学Martin Flodén教授或加拿大西门菲沙大学Paul Klein教授发布的版本。^①

^①两位教授的主页截止本讲义编译的时间分别为`http://paulklein.ca/newsite/start/start.php`和`http://martinfoden.net/`。

5.4.1 牛顿-科特斯公式

回忆数学分析当中我们学习的黎曼(Riemann)积分的定义。^①

定义5.20. (分割) 设闭区间 $[a, b]$ 内有 $n - 1$ 个点, 依次为

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,$$

他们把 $[a, b]$ 分成 n 个小区间 $\Delta_i = [x_{i-1}, x_i], i = 1, 2, \dots, n$ 。这些分点于是构成 $[a, b]$ 的一个分割, 记为 $T = \{\Delta_i\}_{i=1}^n$ 。记每个小区间的长度为 $\Delta x_i = x_i - x_{i-1}$, 则分割 T 的模定义为

$$\|T\| = \max_{1 \leq i \leq n} \{\Delta x_i\}.$$

定义5.21. (黎曼积分) 设 f 是定义在闭区间 $[a, b]$ 上的一个函数。对于 $[a, b]$ 的一个分割 $T = \{\Delta_i\}$, 任取点 $\xi_i \in \Delta_i$, 那么和式

$$\sum_{i=1}^n f(\xi_i) \Delta x_i$$

就称为 f 在 $[a, b]$ 上的黎曼和。进一步的, 设 J 是一个确定的实数, 若对任意 ε , 总有正数 δ 使得对 $[a, b]$ 上的任意分割 T , 只要 $\|T\| < \delta$, 就有

$$\left| \sum_{i=1}^n f(\xi_i) \Delta x_i - J \right| < \varepsilon,$$

那么 J 就称为 f 在 $[a, b]$ 上的黎曼积分, 记为

$$J = \int_a^b f(x) dx.$$

从上面的定义可以总结出来, 黎曼积分的求解实际上分为四个步骤, 即分割、取点、作和、取极限, 图5.4对黎曼积分的定义从图像上作了一个表达。尽管我们对黎曼积分的定义是针对闭区间的, 但是相应的概念可以很容易的推广到广义积分情形, 这里从略。从黎曼积分的定义出发, 我们可以看出相对于牛顿-莱布尼兹公式, 我们上一小节当中看到以下式为形式的数值积分, 其实更加贴近黎曼积分的原始表达:

$$\sum_{i=1}^n f(\xi_i) \omega_i.$$

原则上, 黎曼积分的分割、取点是任意的, 若一个定积分存在, 那么在极限意义下其收敛性不受分割、取点方式不同的影响。不过正如我们所看到的, 实践当中多数需要对积分进行数值解的情形都是 $f(x)$ 的函数形式未知或者其表达式极为繁琐从而无法应用牛顿-莱布尼兹公式求解原函数的情形, 这种情况下分割出大量的节点进行求和运算很多时候是不现实的, 于是我们自然要考虑, 是否存在一种比较经济的选取矩形高度 $f(\xi_i)$ 和矩形宽度(权重) ω_i , 使得我们用较少次数的求和, 就能比较良好的近似原积分的性质。

对牛顿-科特斯类求积公式而言, 矩形宽度固定为对应小区间的长度, 而依照矩形高度 $f(\xi_i)$ 选取方式不同, 又分为不同的子类。设小区间为等分的 $[x_{i-1}, x_i]$, 这样的小区间共有 n 个, 每个长度 h 。那么若每个小区间上的 ξ_i 为区间中点

$$f(\xi_i) = f\left(\frac{x_i + x_{i+1}}{2}\right),$$

^① 数值积分的思想对于更加一般的勒贝格(Lebegue)和斯蒂尔切斯(Stieltjes)积分都是成立的, 不过在宏观经济学当中一般我们只处理常见的黎曼积分。

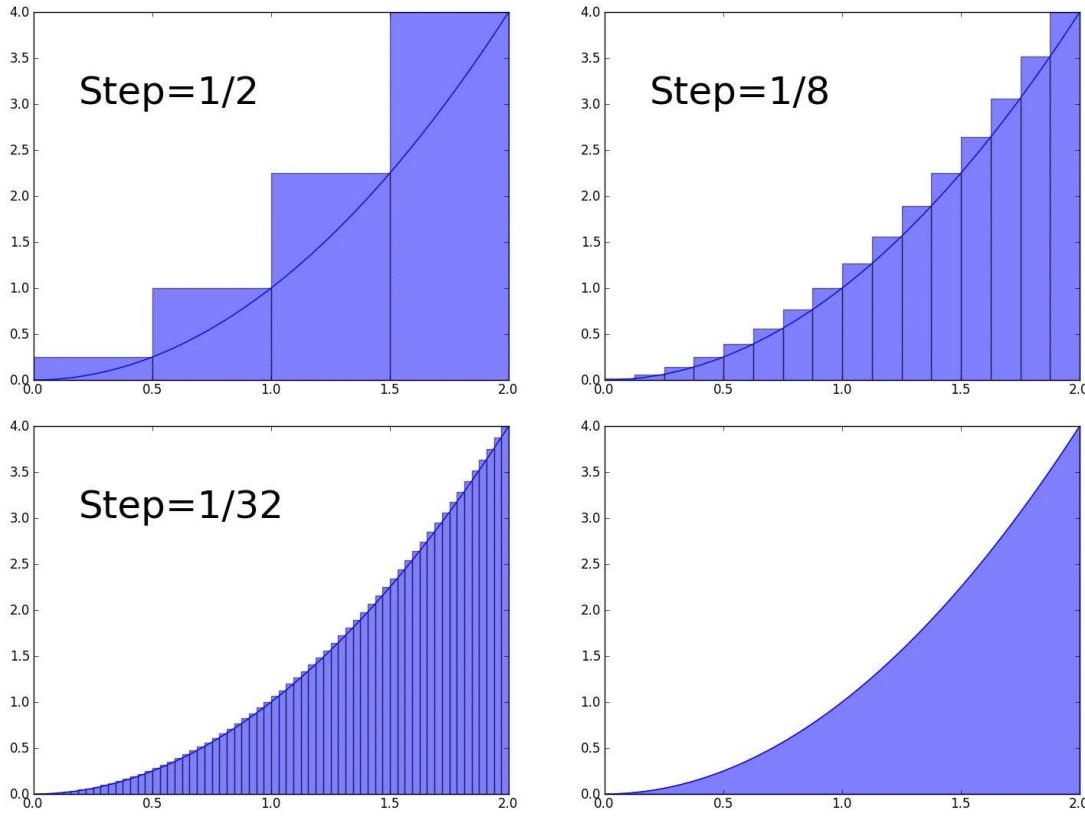


图 5.4.—黎曼积分

那么我们就得到了中点公式，其对应的数值积分为

$$\int_a^b f(x)dx = h \sum_{i=1}^n f(\xi_j) + \frac{h^2(b-a)}{24} f''(\xi),$$

其中 $\frac{h^2(b-a)}{24} f''(\xi)$ 为误差余项，可以由泰勒展开得到，具体计算从略，有兴趣的读者可以参考数值计算相关教材。类似的，如果我们取 $f(\xi_i)$ 为线性平均

$$f(\xi_i) = \frac{f(x_i) + f(x_{i+1})}{2},$$

我们就得到了梯形公式，其对应的数值积分与余项为

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right] - \frac{h^2(b-a)}{12} f''(\xi).$$

若我们用过 $f(x_i)$, $f\left(\frac{x_i+x_{i+1}}{2}\right)$, $f(x_{i+1})$ 三个点的抛物线对 $f(\xi_i)$ 进行近似，那么我们得到了辛普森(Simpson)公式，其对应的数值积分与余项为

$$\int_a^b f(x)dx = \frac{h}{6} \left[f(a) + 4 \sum_{i=0}^{n-1} f(x_{i+1/2}) + 2 \sum_{i=1}^{n-1} f(x_k) + f(b) \right] - \frac{(b-a)}{180} \left(\frac{h}{2} \right)^4 f^{(4)}(\xi).$$

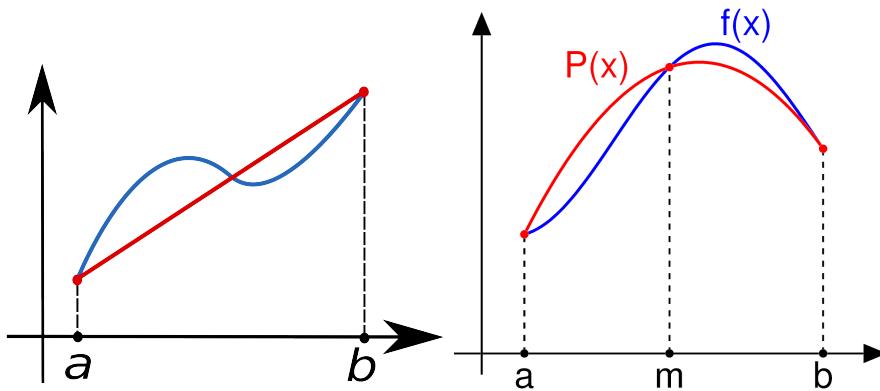


图 5.5—牛顿-科特斯公式

图5.5提供了对梯形公式和辛普森公式的图像表达。进一步的，对上述计算方法在低阶（ n 较少）和高阶处近似值进行特殊权重的线性组合，我们还可以得到龙贝格(Romberg)加速算法，本教程中不做讨论，有兴趣的读者可以参考相关教材。

5.4.2 高斯求积公式

牛顿-科特斯公式的核心是对任意选取的 x_i ，构造 ω_i ，使得黎曼和

$$\sum_{i=1}^n f(\xi_i) \omega_i$$

对于相应小区间上同阶数的多项式插值余项等于零，即黎曼和精确等于黎曼积分。上述三种牛顿-科特斯公式分别对应了阶梯函数、分段线性函数和分段抛物函数。一般而言，如果我们任取节点 x_i ，通过调整权重 ω_i 的取值，对于 $n+1$ 个节点，因为对应的 $n+1$ 个权重给我们提供了 $n+1$ 个自由度，所以由待定系数法可以直接验证，我们可以保证对所有 n 阶多项式黎曼和得到积分的精确解，此时我们称 n 是牛顿-科特斯公式的代数精度(algebraic accuracy)。同理，如果此时我们还可以同时自由调整 $n+1$ 个节点的取值，那么理论上我们可以达到的代数精度是 $2n+1$ 。一般的，当求积公式由 $n+1$ 个节点可以得到 $2n+1$ 阶代数精度时，我们称此时的求积公式为高斯求积公式。

高斯求积公式与正交多项式之间有密切的联系。我们首先定义正交多项式，为此我们需要首先定义权函数的概念。

定义5.22. (权函数) 对 $[a, b]$ 上几乎处处为正的函数 $\rho(x)$ ，若对任意 k 阶多项式积分存在且有限，那么我们称 $\rho(x)$ 为权函数。

定义5.23. (正交多项式) 设 $\varphi_n(x)$ 和 $\rho(x)$ 分别为 $[a, b]$ 上 n 次多项式和权函数，若多项式函数族 $\{\varphi_n(x)\}$ 的内积满足

$$\langle \varphi_j, \varphi_k \rangle = \int_a^b \varphi_j(x) \varphi_k(x) \rho(x) dx = \begin{cases} 0, & j \neq k \\ A_k > 0, & j = k, \end{cases}$$

此时我们称多项式族 $\{\varphi_n(x)\}$ 在 $[a, b]$ 上以 $\rho(x)$ 为权正交， $\varphi_n(x)$ 为 $[a, b]$ 上以 $\rho(x)$ 为权的 n 次正交多项式。

表 5.3
常见正交多项式函数族

| 函数族 | $\rho(x)$ | $[a, b]$ | 定义 | 适合积分 |
|------|------------------|---------------------|---|--|
| 勒让德 | 1 | $[-1, 1]$ | $P_n(x) = \frac{(-1)^n}{2^n n!} \frac{d^n}{dx^n} [(1-x^2)^n]$ | $\int_{-1}^1 f(x) dx$ |
| 切比雪夫 | $(1-x^2)^{-1/2}$ | $[-1, 1]$ | $T_n(x) = \cos(n \cos^{-1} x)$ | $\int_{-1}^1 f(x) \sqrt{1-x^2} dx$ |
| 拉盖尔 | e^{-x} | $[0, \infty)$ | $L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x})$ | $\int_0^\infty f(x) e^{-x} dx$ |
| 埃尔米特 | e^{-x^2} | $(-\infty, \infty)$ | $H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2})$ | $\int_{-\infty}^\infty f(x) e^{-x^2} dx$ |

对于正交多项式与高斯求积公式，我们有如下重要的定理[Davis and Rabinowitz (1984)]:

定理5.7. (正交多项式与高斯求积公式) 设 $\{\varphi_n(x)\}$ 是 $[a, b]$ 上以 $\rho(x)$ 为权的正交多项式， q_n 为其 n 次项前的系数。若记 $\{x_n\}$ 为 φ_n 的零点，即全部满足 $\varphi_n(x_n) = 0$ 的点，那么可以证明

1. $\{x_n\}$ 为区间 $[a, b]$ 的内点，即 $a < x_1 < \dots < x_n < b$ 。
2. 对 $[a, b]$ 上的二阶光滑函数 $f(x)$ ，若取

$$\omega_i = -\frac{q_{n+1}/q_n}{\varphi'_n(x_i)\varphi_{n+1}(x_i)}, \quad (5.9)$$

则对于定积分

$$\int_a^b f(x)\rho(x)dx$$

求积公式

$$\sum_{i=1}^{n+1} \omega_i f(x_i)$$

具有 $2n + 1$ 次代数精度，即该式为高斯求积公式。

注意到高斯求积公式权重的选取与权函数 $\rho(x)$ 本身没有必然联系，权函数的主要作用是针对不同类型的定积分选择相对应的不同正交多项式。我们在表 5.3 中归纳了几类常用的正交多项式、它们对应的积分区间、以及相应的权函数。从表中可以看出，对于一般函数的积分，高斯-勒让德积分最为常用。尽管勒让德多项式定义在 $[-1, 1]$ 上，但是应用定积分变换积分上下限的公式可以很容易的将其定义扩展到任意有限区间上，一般的这种操作对于一般的求积公式也是适用的。高斯-拉盖尔积分常用于连续时间金融当中对于未来现金流的符复合贴现。而对于我们上一小节中遇到的求正态分布随机变量期望的问题，由于正态分布 $N(\mu, \sigma^2)$ 的密度公式为

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\},$$

所以通过变换积分上下限的方法可以很容易转换为高斯-埃尔米特型积分。特别的，应用高斯-埃尔米特求积公式对前述 AR(1) 过程进行近似的算法在宏观经济学中称为 Tauchen and Hussey (1991) 算法。该算法的流程与我们上一节中看到的经典 Tauchen (1986) 算法基本一致，主要不同在于两点。第一，在插值节点的选择上，经典 Tauchen 算法是任取的等距节点，而 Tauchen-Hussey 算法取的是高斯-埃尔米特多项式的零点。第二，在插值权重的选择上，经典 Tauchen 算法应用了矩形公式，而 Tauchen-Hussey 算法应用

了依照方程(5.9)计算出来的权重。细心的读者已经看出，经典Tauchen算法与Tauchen-Hussey算法的区别实际上就是牛顿-科特斯公式与高斯求积公式之间的区别。

最后这里要提醒读者注意的是，尽管Tauchen-Hussey算法看上去更加复杂，也似乎具有更好的性质，但是在实际应用中它的效果并不总是比经典Tauchen算法要好。在对于独立同分布随机过程进行近似时，Tauchen-Hussey的近似效果通常好于经典Tauchen方法，对 $\rho < 0.95$ 的AR(1)过程这个结论基本上也有成立。但是如[Flodén \(2008\)](#)所指出的，对于 $\rho > 0.95$ 的高自回归系数的AR(1)过程，经典Tauchen算法当手工调整节点上下界时，对于平稳分布的方差近似要好于Tauchen-Hussey算法。

第六章 不确定性动态优化问题

上一章当中我们介绍了如何处理存在不确定性的动态优化问题当中随机过程的处理技术，同时通过平稳过程介绍了用时间序列分析理解模型统计性质的理论基础。有了这些准备，本章我们详细介绍第四章中真实经济周期模型的求解。简单比较一下这个模型与我们熟悉的新古典增长模型，由于我们在第四章已经说明对于代表性消费者的情形，竞争均衡与社会最优问题是等价的，所以两者的主要区别有三个。^① 第一，真实经济周期模型当中存在生产率的不确定性；第二，这种不确定性是AR(1)的；第三，每期在跨期消费储蓄决策之外，消费者还需要进行期间的劳动闲暇决策。所以本章我们也按照这几个步骤一步步的扩展增长模型。具体来说，我们首先考虑在增长模型当中引入独立同分布的生产率冲击随机过程，之后在这个模型的基础上将这个冲击修改为一阶自回归过程，最后我们将期间决策引入模型，求解完整的真实经济周期模型。前两部分的内容源自Brock and Mirman (1972)。截至本章为止，我们没有介绍如何确定数量宏观经济模型当中参数的设置。在本章的最后，我们将详细介绍在宏观经济学当中最为常用的参数设置方法，校准(calibration)。我们将同一个物理学的例子相对比说明，数量宏观经济模型从方法论的角度出发，是一种直接应用数据对理论进行结构化检验的方法。上一小节以及本节的内容主要取自Cooley and Prescott (1995)。

6.1 独立同分布生产率冲击

我们首先考虑最简单的扩展，即在第三章的增长模型当中对生产率引入独立同分布冲击。考虑如下的社会最优模型

$$\begin{aligned} & \max_{c_t, k_{t+1}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \log(c_t) \\ & s.t \\ & c_t + k_{t+1} \leq \theta_t k_t^{\alpha} \\ & \theta_t = \begin{cases} \theta_1, & P(\theta_t = \theta_1) = 0.5 \\ \theta_2, & P(\theta_t = \theta_2) = 0.5 \end{cases} \\ & k_0, \theta_0 \text{给定}, c_t, k_t \geq 0, \forall t. \end{aligned} \tag{6.1}$$

优化问题(6.1)与我们熟悉的(3.1)相比的唯一区别是产出并不是由资本存量唯一确定的，还受到当期生产率冲击 θ_t 的影响。这里 θ_t 没有序列相关性，每一期 θ_t 可能有两个取值， $\theta_1 > \theta_2$ ，两者概率相等。序列

^①当然我们这里再次强调一下这个等价性只在一些特殊的情形下成立，一般而言这需要代表性消费者或者完全线性的政策函数。关于异质性模型的加总，有兴趣的同学可以参考Azzimonti, de Francisco and Krusell (2008)

形式的优化问题(6.1)可以转化成一个等价的动态优化问题:

$$V(k; \theta) = \max_{k'} \{\log(\theta k^\alpha - k') + \beta \mathbb{E} V(k'; \theta')\}, \quad (6.2)$$

其中 k 是内生状态变量, θ 是外生状态变量。求解动态优化问题(6.2)的目标是求出储蓄的政策函数

$$k' = g_k(k, \theta),$$

之后通过资源约束可以推出消费的政策函数

$$g_c(k, \theta) = \theta k^\alpha - g_k(k, \theta).$$

注意到此时因为我们有了两个状态变量, 所以这两个政策函数的函数表都是矩阵。进一步的, 我们对参数做设定 $\alpha = 0.4$, $\beta = 0.9888$, $\theta_1 = 1.01625$, $\theta_2 = 0.98375$ 。下面我们来求解这个模型。

6.1.1 求解模型

我们整体的思路与普通增长模型相同, 也是先使用值函数迭代法求出 $V(k; \theta)$ 的函数表表达式, 之后再通过优化问题(6.2)求出储蓄的政策函数。尽管此时(6.2)中贝尔曼方程中包含了一个我们不熟悉的数学期望 $\mathbb{E} V(k'; \theta')$, 但是当对于每个节点的组合 (k', θ') 我们都已经找到了 $V(k', \theta')$ 的对应的取值时, 数学期望只不过是对这些取值进行代数平均而已。如之前一样, 在值函数迭代算法的第一步当中我们假设 $V(k; \theta) = \mathbf{0}$, 此时 $\mathbb{E} V(k'; \theta') = \mathbf{0}$, 而第一轮迭代的 $V(k; \theta)$ 就可以由

$$\max_{k'} \{\log(\theta_i k^\alpha - k')\}, i = 1, 2,$$

穷举求解。之后算法可以不断迭代, 直到 $V(k; \theta)$ 收敛。需要注意的是, 由于随机性的存在, 所以这里算法的收敛性需要更强的条件, 有兴趣的同学可以参考[Stokey and Lucas \(1989\)](#)的第九章相关讨论, 本教程不做涉及。我们还是先给出算法的伪代码。该代码的目的是求解(6.2), 解的定义是对每个网格节点 $\{k_i, \theta_j\}$, 计算出对应的 c_{ij} 与 k'_{ij} 。为此, 我们采用以下步骤:

1. 将状态空间离散化, 即取向量 $\mathbf{k} = (k_1, k_2, \dots, k_N)$ 对 $V(k; \theta)$ 的自变量取值集合进行近似。
2. 使用值函数迭代法求解 $V(k; \theta)$, 特别的, 这包括下列步骤:

- (a) 任取初始变量 $V_0(k; \theta)$, 定义优化问题

$$V_1(k; \theta) = \max_{k'} \{\log(\theta k^\alpha - k') + \beta \mathbb{E} V_0(k'; \theta')\}$$

- (b) 对每一个网格节点 (k_i, θ_m) , 计算对 $k'_j \in \mathbf{k}$ 的每一个取值

$$\log(\theta_m k_i^\alpha - k'_j) + \frac{\beta}{2} [V_0(k'_j; \theta_1) + V_0(k'_j; \theta_2)]$$

的值, 记为 V_{ijm} 。注意这里我们要求 k_i 与 k'_j 从同一个集合, 即 \mathbf{k} 当中取值。 V_{ijm} 记录的是, 当本期的生产率冲击为 θ_m 时, 若本期资本存量为 k_i , 那么当储蓄决策为 k'_j 时, 消消费者的效用(本期加后续期)。这一步当中状态变量 k_i 和 θ_m 保持不变, 但是选择变量 k'_j 的取值遍历了 \mathbf{k} 。

- (c) 在 $(V_{ijm})_{j=1}^N$ (向量) 的所有元素中选取最大元, 记其指标是 l , 那么此时对于网格节点 (k_i, θ_m) 而言, 最优的储蓄决策就是 k'_l 。此时 $V_1(k_i; \theta_m) = V_{ilm}$ 。

- (d) 当(b)、(c)中的步骤对每一个 (k_i, θ_m) 都计算过之后，我们就计算出了新的 $V_1(k; \theta)$ 。这一步当中 k_i 遍历了 \mathbf{k} ，而 θ 也取遍了 θ_1, θ_2 。注意到这里的计算其实就是对 θ_1 和 θ_2 分别应用一次增长模型的值函数迭代相应过程。
- (e) 新的 $V_1(k; \theta)$ 可以使用同样的步骤计算 $V_2(k; \theta)$ 。一般的，对于任意的 t ，重复(b)至(d)，计算 $V_{t+1}(k; \theta)$ 。
- (f) 定义 $\delta_t = \max \|V_t - V_{t+1}\|$ ，当 δ_t 足够小，即对于一个任取的 ε 有 $\delta_t < \varepsilon$ ，我们判断函数列 $\{V_t(k)\}$ 已经收敛，停止(e)中的迭代。此时， $V_{t+1}(k; \theta)$ 为真实解 $V(k; \theta)$ 提供了良好的近似。
3. 对于上一步最后得到的 $V_{t+1}(k; \theta)$ 应用步骤2.(b) 与2.(c)，求解最优政策。具体来说，对每一个 (k_i, θ_m) ，计算对 $k'_j \in \mathbf{k}$ 的每一个取值

$$\log(\theta_m k_i^\alpha - k'_j) + \frac{\beta}{2} [V_0(k'_j; \theta_1) + V_0(k'_j; \theta_2)]$$

的值，记为 V_{ijm} 。在 $(V_{ijm})_{j=1}^N$ （矩阵）的所有元素中中选取最大元，记其指标是 l ，那么此时对于网格节点 (k_i, θ_m) 而言，最优的储蓄决策就是 k'_l 。最优消费决策由

$$c_{im} = \theta_m k_i^\alpha - k'_l.$$

给出。当对于每个 θ_m ， k_i 遍历了 \mathbf{k} 时，我们就得到了算法求解的目标，即函数表 $(k_i, \theta_m; k'_i)$ 与 $(k_i, \theta_m; c_i)$ 。

上述伪码转化的MATLAB代码包含在代码6.1当中。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code solves the Brock and Mirman (1972,JET) stochastic dynamic
7 % programming problem with i.i.d productivity shocks.
8 % - Code 6.1 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 02/15/2016: Original Code
17
18 clc;
19 clear all ;
20 tic; % Initialize a stopwatch
21 %
22 % DATA DICTIONARY
23 %
24 % Parameterizations
25 alpha = 0.4 ;
26 beta = 0.9888 ;
27 theta = [1+1.625/100;1-1.625/100] ;
28 N = 300 ; % Number of Grids
29 tolv = 1e-9 ; % Tolerance level

```

```

30 tolN = 10000 ;      % Maximum Number of Loops in While-loop,
31                      % to avoid never-ending dead-loop
32 % Variables
33 V0 = zeros(2,N) ;    % Vt
34 V1 = zeros(2,N) ;    % Vt+1
35 VT = zeros(1,N) ;    % Vi,j(theta)
36 polc = zeros(2,N) ;  % Policy Function for Consumption
37 polk = zeros(2,N) ;  % Policy Function for Saving
38 pindk = zeros(2,N); % Saving policy index, used in simulation
39
40 % Auxillary Variables
41 optim = -1 ;         % Store the index of optimal choice
42 eps = 1 ;             % Evaluate convergence of Vt
43 loopn = 1 ;           % Calculate the number of loops
44
45 % Calculate the Steady State
46 kss = (1/(alpha*beta))^(1/(alpha-1)) ;
47 kmin = 0.8*kss;
48 kmax = 1.2*kss;
49 % Discretize the state space
50 kgrid = linspace(kmin, kmax, N);
51
52 % Calculate the consumption level when current capital is ki and
53 % future capital is kj
54 c = zeros(2,N^2);
55 for indt = 1:1:2
56     for indi = 1:1:N
57         for indj = 1:1:N
58             c(indt,(indi-1)*N+indj)=theta(indt)*kgrid(indi)^alpha ...
59                         - kgrid(indj) ;
60         end
61     end
62 end
63 % Avoid evaluating negative consumption in log-utility
64 c(c <= 0) = 1e-11;
65 c = log(c);
66
67 % -----
68 % VALUE FUNCTION ITERATION
69 % -----
70 % Iterate Vt until convergence
71 while (eps > tolv) && (loopn < tolN)
72     for indt = 1:1:2
73         for indi = 1:1:N
74             for indj = 1:1:N
75                 EV = .5*V0(1,indj) + .5*V0(2,indj);
76                 VT(indj) = c(indt,(indi-1)*N+indj) + beta*EV;
77             end
78             V1(indt,indi) = max(VT);
79         end
80     end
81     eps = max(max(abs(V1-V0)));
82     V0 = V1;
83     loopn = loopn+1;
84     if mod(loopn,250) == 0
85         disp(['Loop=' ,num2str(loopn)])
86         disp(['eps=' ,num2str(eps)])
87     end
88 end
89

```

```

90 % -----
91 % OPTIMAL POLICIES
92 % -----
93 % Using the numerical approximation of V to solve the optimization problem
94 for indt = 1:1:2
95     for indi = 1:1:N
96         EVT = .5*V1(1,:) + .5*V1(2,:);
97         VT = c(indt,(indi-1)*N+1:indi*N) + EVT;
98         % Find the saving that leads to the highest utility
99         [xtemp optim] = max(VT) ;
100        pindk(indt,indi) = optim;
101        polk(indt,indi) = kgrid(optim);
102    end
103 end
104 polc = theta*kgrid.^alpha - polk ;
105 time = toc;      % Paired with tic, calculate total running time in between
106 disp(['Total Running Time=' ,num2str(time), ' secs'])
107
108 % Plot the value functions and policy functions
109 % Value functions for different shocks and Expected value functions
110 figure(1)
111 hold on;
112 plot(kgrid,V1(1,:),'LineWidth',1.5) ;
113 plot(kgrid,V1(2,:),'r','LineWidth',1.5) ;
114 plot(kgrid,.5*V1(1,:)+.5*V1(2,:),'--k','LineWidth',1.5) ;
115 legend('V(\theta_1)', 'V(\theta_2)', 'EV(\theta)', 'Location', 'SouthEast')
116 title('Value Functions of Different Shocks')
117 xlabel('Capital') ;
118 ylabel('Value Functions')
119 hold off;
120
121 % Saving functions for different shocks
122 figure(2)
123 hold on;
124 plot(kgrid,polk(1,:),'LineWidth',1.5) ;
125 plot(kgrid,polk(2,:),'r','LineWidth',1.5) ;
126 legend('k'(\theta_1)', 'k'(\theta_2)', 'Location', 'SouthEast')
127 title('Saving Functions of Different Shocks')
128 xlabel('Capital') ;
129 ylabel('Savings')
130 hold off;
131
132 % Consumption functions for different shocks
133 figure(3)
134 hold on;
135 plot(kgrid,polc(1,:),'LineWidth',1.5) ;
136 plot(kgrid,polc(2,:),'r','LineWidth',1.5) ;
137 legend('c(\theta_1)', 'c(\theta_2)', 'Location', 'SouthEast')
138 title('Consumption Functions of Different Shocks')
139 xlabel('Capital') ;
140 ylabel('Consumption')
141 hold off;
142
143 % save brockmirmaniid.mat;

```

代码 6.1.—独立同分布随机冲击

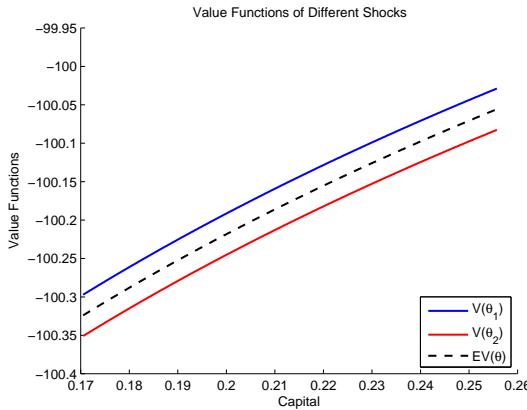


图 6.1.—独立同分布随机冲击·值函数

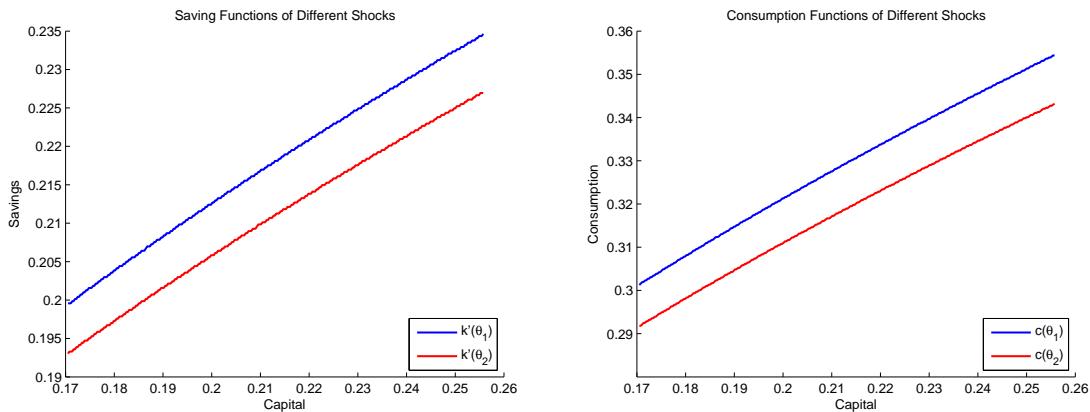


图 6.2.—独立同分布随机冲击·政策函数

代码6.1与代码3.1的结构基本相同，这里就不详细展开叙述了，我们只简单介绍一下当中应用的几个新的函数。第一，在第20行和第103行我们配对使用了 `tic` 和 `toc`，这是MATLAB内建的计时函数，其作用是分别记录下 `tic` 和 `toc` 对应的CPU时间，并在 `toc` 调用的时候自动返回以秒计数的两者的差异。比如，在代码6.1的精度要求下，在作者的个人电脑上，105行对应的变量 `Time` 的值为847秒。同时读者可以看出，尽管我们这里只加入了一个独立同分布且取值为两个的随机冲击，程序的运行效率已经比第三章中的增长模型大幅下降，这也是为什么我们在上一章当中要学习使用较少状态的离散马尔科夫链来近似连续状态空间的马尔科夫过程。即便是在现代计算机的辅助下，使用大量的节点来近似黎曼积分所带来的计算负担也是十分没有效率的。第二，注意到在三个主循环体55-62行、72-80行和94-103行，我们需要添加关于 θ_t 的外循环，其计数变量为 `indt`。第三，在运行时间较长循环体当中，适当的向显示终端输出程序运行的状态，比如84-87行当中执行循环的次数与值函数收敛的程度，可以帮助读者及时的判断程序是否存在明显的运行中错误，以便及时中断进行调试。最后，注意到75行和96行当中我们对于优化问题(6.2)内部数学期望的处理实际上就是简单的代数平均值。我们运行程序，结果在图6.1和图6.2中显示。图6.1中描绘了 $V(k, \theta)$ 和 $EV(k, \theta)$ ，读者可以再次看到，值函数的期望就是两种状态下值函数的代数平均。图6.2中描绘了储蓄与消费的政策函数。政策函数的性质是符合直觉的，即当生产率

或资本存量较高时，储蓄与消费都较高。

6.1.2 数值仿真

对带有随机性的模型而言，通常求解其政策函数只是分析模型性质的第一步，我们最终感兴趣的，是以随机动态优化问题定义的随机过程的统计性质。对绝大多数可以使用值函数迭代法求解的宏观经济模型来说，其随机性继承自外生的随机过程的特性，一般为生产率或劳动收入冲击，通常当这些过程是遍历的平稳过程时，最优化问题的内生解也是遍历平稳的。所以我们可以应用平稳过程的遍历性条件，通过分析一条较长的时间序列样本曲线的统计性质，来分析样本曲线的集平均性质。

应用模型政策函数的数值仿真与代码1.1的结构大体相同。我们首先需要通过随机数生成器确定外生状态变量即生产率冲击在时间序列上的取值。随后，我们从任给的内生状态变量（这里为资本存量）的初值开始，注意到任给(k_0, θ_0)的取值，我们都可以通过政策函数 $g_c(k; \theta)$ 和 $g_k(k; \theta)$ 求出本期消费与下一期资本存量 k_1 的值。而给定了 k_1 和第一步当中通过随机数生成器预先取好的 θ_1 ，又可以如法炮制获得 c_1 和 k_2 的值，于是整个仿真可以一直执行到最后一期。之后我们便可以分析样本函数的数字特征。这里与代码1.1不同的地方是，代码1.1是定义在连续状态空间上的，所以外生的冲击可以直接按照实数值记录。在本节的模型中，政策函数是定义在函数表之上的，所以如果我们直接按照政策函数所显示的实数值进行迭代，那么在下一期需要将按照实数值反推指标值。比如，给定 k_0 时，政策函数告诉我们 $k_1 = 0.22$ ，但是我们的函数表是定义在 $indk=1:N$ 之上的（37行），所以直接带入`polk(theta_1, 0.22)`是没有意义的。一般而言这有几个常用的做法，一是将 k 的取值按照`kgrid`对应回整数定义的指标集，比如如果假设向量`kgrid`的第45个元素值为0.22，那么我们需要定义一个函数将 $k_1 = 0.22$ 转换回45。第二是可以通过线性插值的方法，关于线性插值我们将推迟到第九章异质性消费者处再进行介绍。第三是如我们在代码6.1所做的那样，在38和100行处专门定义一个向量，来记录最优选择所对应的指标集。

由于伪代码与代码1.1大体相同，所以考虑到篇幅这里省略掉，我们直接在代码6.2当中给出对应的MATLAB程序代码。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 %
4 %
5 % Purpose:
6 % - This code simulates the Brock and Mirman (1972, JET) stochastic
7 % dynamic programming problem with AR(1) productivity shocks.
8 % - Code 6.4 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 02/15/2016: Original Code
17 %
18 clc;
19 clear all ;
20
21 load brockmirmaniid.mat ;
22 %
23 % DATA DICTIONARY

```

```

24 % -----
25 % Declaring variables
26 burnin = 50;
27 T = burnin + 201;
28 s_theta = ones(1,T);
29 s_k = zeros(1,T);      % kprime
30 s_c = zeros(1,T);
31 s_y = zeros(1,T);
32
33 % -----
34 % SIMULATION
35 % -----
36 % Draw random shocks
37 epsilon = rand(1,T);
38 s_theta(epsilon<0.5) = 2;
39
40 % Initialization
41 indk = N/2;           % Start from deterministic steady state
42 s_k(1) = kgrid(indk) ;
43
44 for indt = 2:T
45     indk = pindk(s_theta(indt-1),indk);
46     s_k(indt) = polk(s_theta(indt),indk);
47 end
48 s_y(1:T-1) = theta(s_theta(1:T-1)).*(s_k(1:T-1).^alpha) ;
49 s_c(1:T-1) = s_y(1:T-1) - s_k(1:T-1);
50
51 xt = burnin+1:T-1;
52 % Calculate coefficient of correlation between y,c and y,k
53 rho_yc = corrcoef(s_y(xt),s_c(xt));
54 rho_yk = corrcoef(s_y(xt),s_k(xt+1));
55
56 % Plot Time Series Sample Curve
57 figure(1)
58 hold on;
59 plot(xt,s_y(xt), 'LineWidth',1.5);
60 plot(xt,s_k(xt), '--r', 'LineWidth',1.5);
61 plot(xt,s_c(xt), '-.k', 'LineWidth',1.5);
62 legend('Output','Investment','Consumption','Location','NorthEast');
63 xlabel('Time')
64 ylim([0.15 0.6]);
65 title('Simulation of i.i.d Brock-Mirman')
66 hold off;

```

代码 6.2.—独立同分布随机冲击仿真

对于代码6.2我们有几点评注。首先，在21行中，我们读入了在代码6.1的143行存储的运行结果。宏观经济学当中我们经常要做的是对同一个模型进行许多次仿真，如果每次做仿真的时候都要把政策函数重新求解一次，这显然是过于劳民伤财了。于是我们通常的做法是将政策函数保存到一个文件当中，在仿真的时候只读取政策函数的结果。第二，在仿真随机冲击的时候，因为我们这里的 θ 只取两个值，且概率相等，所以我们在37行当中的随机数生成器中生成了[0, 1]区间上均匀分布的随机数，当随机数小于0.5时，我们就说马尔科夫链实现了状态1，反之则实现状态2。这是因为在[0, 1]区间上均匀分布抽取随机数小于0.5的概率恰好等于0.5，容易看出，这个技巧可以很容易的推广到状态数有限的结构更加丰富的马尔科夫链上。第三，注意到在循环体44-47行当中，我们应用了前述基于指标集的仿真方法。第

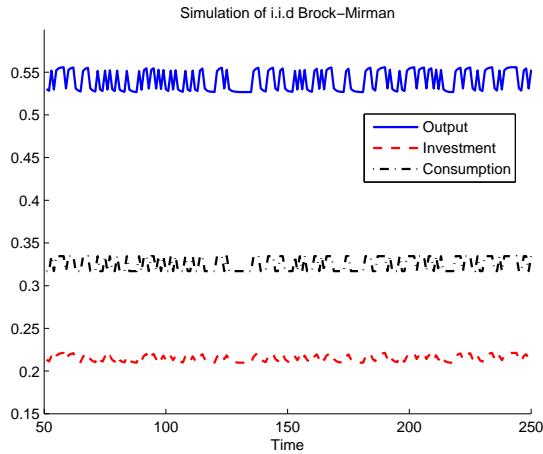


图 6.3.—独立同分布随机冲击·仿真

四，我们在26行处定义了变量`burnin`，回忆例子5.2中的结论，这是为了消除仿真初值对于平稳分布统计性质的影响。最后，我们在53-54行调用MATLAB的内建函数计算产出-消费与产出-投资的相关系数。由于在优化问题(6.2)当中我们假设了完全折旧 $\delta = 1$ ，所以注意到此处投资等于下期的资本。我们运行程序，结果在图6.3中显示。相关系数的计算显示， $\text{corr}(Y, C) = 0.9952$ ， $\text{corr}(Y, I) = 0.4041$ 。注意到这里我们只对经济仿真了200期（为了图像的可视性），所以前述两个相关系数在不同的仿真之间的变化可能会比较大。回忆上一章中我们关于平稳过程的讨论，读者可以自行尝试更长时间的仿真，来验证当模型仿真长度足够长的时候，对应的随机过程的数字特征会趋于稳定。

6.2 一阶自回归生产率冲击

下面我们将生产率冲击从独立同分布改为一阶自回归。我们同时还放松另外两个函数形式上的假设。第一，我们假设消费者的效用函数由更加一般的不变相对风险厌恶(*constant relative risk aversion, CRRA*)形式给出。顾名思义，CRRA效用函数的定义来源于相对风险厌恶系数[Pratt (1964)]。具体来说，对效用函数 $u(c)$ 而言，定义相对风险厌恶系数 γ 为

$$\gamma = -\frac{cu''(c)}{u'(c)}.$$

容易验证，当

$$u(c) = \frac{c^{1-\gamma} - 1}{1 - \gamma}, \quad (6.3)$$

时， γ 不随 c 的变化而变化。不变相对风险厌恶函数的一个含义是说，在一个资产组合问题中，消费者总财富中分配到风险资产的百分比不随财富水平变化而改变。对于CRRA效用函数(6.3)而言，另一个密切相关的概念是跨期替代弹性(*intertemporal elasticity of substitution, IES*)，IES的含义是在消费储蓄问题当中，利率变化的百分比所导致的储蓄行为变化的百分比。对CRRA函数而言，可以证明 $\text{IES} = 1/\gamma$ ，也就是跨期替代弹性是风险厌恶系数的倒数。这里的经济直觉是直观的，消费者的风

险厌恶主要体现在消费水平的波动会导致消费者期望效用的下降（由延森不等式证明），而跨期替代弹性体现了消费者

多大程度上愿意让消费水平对价格信号做出反应。所以，对于一个极度风险厌恶的消费者而言，轻微的消费水平的波动就会造成较大的福利水平下降，于是消费者此时对价格的反应会相对迟钝，反映出来的跨期替代弹性较低。^①不过这里希望读者注意的是，尽管这两个概念之间紧密相关，这并不代表他们是同一回事儿。在现代资产组合理论当中，对跨期替代弹性和风险厌恶系数的分离对于基于消费的资产定价理论有着重要的意义，不过由于对这些的讨论应用了比较深入的数学工具和复杂的异质性消费者模型，在本教程中我们对此不做涉及。^②另外，利用洛必达法则可以验证，

$$\lim_{\gamma \rightarrow 1} \frac{c^{1-\gamma} - 1}{1 - \gamma} = \log c,$$

即我们之前分析的对数效用函数广义上是一种特殊的CRRA效用函数。^③

第二，我们放松完全折旧的假设，设此时的折旧率 $\delta < 1$ 。不过读者马上将会看到，这两个假设的放松在程序上的处理是十分简单的，只需要修改相关部分的表达式即可。此时对应的社会最优化问题成为

$$\begin{aligned} & \max_{c_t, k_{t+1}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\gamma} - 1}{1 - \gamma} \\ & \text{s.t.} \\ & c_t + k_{t+1} - (1 - \delta)k_t \leq \theta_t k_t^\alpha \\ & \log(\theta_t) = \rho \log(\theta_{t-1}) + \varepsilon_t, \varepsilon_t \stackrel{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2) \\ & k_0, \theta_0 \text{给定}, c_t, k_t \geq 0, \forall t. \end{aligned} \quad (6.4)$$

而其对应的动态优化问题此时是

$$\begin{aligned} V(k; \theta) &= \max_{k'} \left\{ \frac{[\theta k^\alpha - k' + (1 - \delta)k]^{1-\gamma} - 1}{1 - \gamma} + \beta \mathbb{E}[V(k'; \theta') | \theta] \right\} \\ & \text{s.t.} \\ & \log \theta_t = \rho \log \theta_{t-1} + \varepsilon_t, \varepsilon_t \stackrel{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2). \end{aligned} \quad (6.5)$$

注意到由于生产率冲击是AR(1)的，所以此时贝尔曼方程内部的期望是条件期望。

求解(6.5)的算法与上节用来求解(6.2)的大同小异，这里不逐步列出求解的伪码。事实上，我们只需要对上节当中介绍的伪码做三处替换。第一，我们需要增添一步预处理程序段，将(6.5)中的马尔科夫过程转化为离散的马尔科夫链。第二，在2.(b)当中求值函数的条件期望时，我们需要将概率分布换成离散马氏链对应的转移概率矩阵。具体的，当本期的冲击是 θ_i 时，记转移阵是 $\mathbf{P} = \{P_{ij}\}$ ，那么此时的条件期望的概率就是矩阵的第*i*行，即本期状态为 θ_i 时，下一期当中生产率实现为各 θ_j 的对应概率。在上一节独立同分布的情形下，这个非条件概率固定等于0.5。第三，我们需要对效用函数和资源约束的表达式做相应的替换。

我们对模型做如下的参数假设： $\alpha = 0.36$, $\beta = 0.9$, $\gamma = 2$, $\delta = 0.1$, $\rho = 0.95$, $\sigma_\varepsilon = 0.00712$ 。我们于是直接在代码6.3当中给出求解该模型的MATLAB代码。

^① 关于更详细的讨论，请有兴趣的读者参考Mas-Colell, Whinston and Green (1995)第六章的相关讨论。

^② 有兴趣的读者可以参考Epstein and Zin (1989), Gomes and Michaelides (2005) 以及Guvenen (2009)。

^③ 对数效用函数下价格变化的替代效应和收入效应相等，所以价格变化会导致相消费的相同百分比变化，我们这里不做详细的展开，有兴趣的读者还请参考Mas-Colell, Whinston and Green (1995)的第六章。

```
1 % -----
2 % PROGRAM DESCRIPTION
3 %
4 %
5 % Purpose:
6 % - This code solves the Brock and Mirman (1972,JET) stochastic dynamic
7 % programming problem with AR(1) productivity shocks.
8 % - Code 6.3 in the Lecture Notes
9 %
10 % Author:
11 % Xin Tang © Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 02/15/2016: Original Code
17
18 clc;
19 clear all ;
20 tic; % Initialize a stopwatch
21 %
22 % DATA DICTIONARY
23 %
24 % Parameterizations
25 alpha = 0.36 ;
26 beta = 0.9 ;
27 gamma = 2;
28 delta = 0.1;
29 rho = 0.95;
30 sigmae = 0.00712;
31 mu = 0;
32 nshock = 9 ;
33 m = 3;
34 N = 350 ; % Number of Grids
35 tolv = 1e-9 ; % Tolerance level
36 tolw = 10000 ; % Maximum Number of Loops in While-loop,
37 % to avoid never-ending dead-loop
38
39 % Initialize Stochastic Shocks
40 % Using classical Tauchen (1986) implemented by Code 5.2
41 [theta,P,pi,sac,ssd] = Tauchen86(rho, mu, sigmae, m, nshock);
42 theta = exp(theta)';
43 theta_ss = pi*theta ;
44
45 % Declaring Variables
46 V0 = zeros(nshock,N) ; % Vt
47 V1 = zeros(nshock,N) ; % Vt+1
48 VT = zeros(1,N) ; % Vj(theta)
49 polc = zeros(nshock,N) ; % Policy Function for Consumption
50 polk = zeros(nshock,N) ; % Policy Function for Saving
51 pindk = zeros(nshock,N); % Saving policy index, used in simulation
52
53 % Auxillary Variables
54 optim = -1 ; % Store the index of optimal choice
55 eps = 1 ; % Evaluate convergence of Vt
56 loopn = 1 ; % Calculate the number of loops
57
58 % Calculate the Steady State
59 kss = ((1-beta*(1-delta))/(alpha*beta*theta_ss))^(1/(alpha-1)) ;
```

```

60 kmin = 0.85*kss;
61 kmax = 1.5*kss;
62 % Discretize the state space
63 kgrid = linspace(kmin, kmax, N);
64
65 % Calculate the utility level when current capital is ki and
66 % future captial is kj
67 c = zeros(nshock,N^2);
68 for indt = 1:1:nshock
69   for indi = 1:1:N
70     for indj = 1:1:N
71       c(indt,(indi-1)*N+indj)=theta(indt)*kgrid(indi)^alpha ...
72         - kgrid(indj) + (1-delta)*kgrid(indi) ;
73     end
74   end
75 end
76 % Avoid evaluating negative consumption in log-utility
77 c(c <= 0) = 1e-11;
78 c = (c.^(1-gamma)-1)/(1-gamma);
79
80 % -----
81 % VALUE FUNCTION ITERATION
82 % -----
83 % Iterate Vt until convergence
84 while (eps > tolv) && (loopn < tolv)
85   for indt = 1:1:nshock
86     for indi = 1:1:N
87       for indj = 1:1:N
88         EV = P(indt,:)*V0(:,indj);
89         VT(indj) = c(indt,(indi-1)*N+indj) + beta*EV;
90       end
91       V1(indt,indi) = max(VT);
92     end
93   end
94   eps = max(max(abs(V1-V0)));
95   V0 = V1;
96   loopn = loopn+1;
97   if mod(loopn,50) == 0
98     disp(['Loop=',num2str(loopn)])
99     disp(['eps=',num2str(eps)])
100   end
101 end
102
103 % -----
104 % OPTIMAL POLICIES
105 % -----
106 % Using the numerical approximation of V to solve the optimization problem
107 for indt = 1:1:nshock
108   for indi = 1:1:N
109     EVT = P(indt,:)*V1;
110     VT = c(indt,(indi-1)*N+1:indi*N) + EVT;
111     % Find the saving that leads to the highest utility
112     [xtemp optim] = max(VT) ;
113     pindk(indt,indi) = optim;
114     polk(indt,indi) = kgrid(optim);
115   end
116 end
117 polc = theta*kgrid.^alpha - polk + ones(nshock,1)*(1-delta)*kgrid ;
118 poli = polk - ones(nshock,1)*(1-delta)*kgrid ;
119 time = toc;    % Paired with tic, calculate total running time in between

```

```

120 disp(['Total Running Time=',num2str(time), ' secs'])
121
122 % Plot the value functions and policy functions
123 % Value functions for different shocks and Expected value functions
124 auxmid = (nshock + 1)/2;
125 indplot = [auxmid-2 auxmid auxmid+2];
126 figure(1)
127 hold on;
128 plot(kgrid,V1(indplot(1),:),'LineWidth',1.5) ;
129 plot(kgrid,V1(indplot(2),:),'r','LineWidth',1.5) ;
130 plot(kgrid,V1(indplot(3),:),'k','LineWidth',1.5) ;
131 legend('V(\theta_l)', 'V(\theta_m)', 'V(\theta_h)', 'Location', 'SouthEast')
132 title('Value Functions of Different Shocks')
133 xlabel('Capital') ;
134 ylabel('Value Functions')
135 hold off;
136
137 % Saving functions for different shocks
138 figure(2)
139 hold on;
140 plot(kgrid,polk(indplot(1),:),'LineWidth',1.5) ;
141 plot(kgrid,polk(indplot(2),:),'r','LineWidth',1.5) ;
142 plot(kgrid,polk(indplot(3),:),'k','LineWidth',1.5) ;
143 legend('k'(\theta_l)', 'k'(\theta_m)', 'k'(\theta_h)', 'Location', 'SouthEast')
144 title('Saving Functions of Different Shocks')
145 xlabel('Capital') ;
146 ylabel('Savings')
147 hold off;
148
149 % Consumption functions for different shocks
150 figure(3)
151 hold on;
152 plot(kgrid,polc(indplot(1),:),'LineWidth',1.5) ;
153 plot(kgrid,polc(indplot(2),:),'r','LineWidth',1.5) ;
154 plot(kgrid,polc(indplot(3),:),'k','LineWidth',1.5) ;
155 legend('c(\theta_l)', 'c(\theta_m)', 'c(\theta_h)', 'Location', 'SouthEast')
156 title('Consumption Functions of Different Shocks')
157 xlabel('Capital') ;
158 ylabel('Consumption')
159 hold off;
160
161 save brockmirmanar1_tmp.mat;

```

代码 6.3.—AR(1)随机冲击

在代码6.3的41行，我们通过函数调用了代码5.2。这需要对代码5.2进行一些修改，使其成为一个函数脚本(function-script)文件，代码6.4即为修改后的结果。

```

1 %
2 % -----%
3 % PROGRAM DESCRIPTION
4 %
5 % Purpose:
6 % - This code approximates AR(1) process with a discrete Markov Chain.
7 % - Function code to be called in code 6.3.
8 % - Code 6.4 in the Lecture Notes.
9 %

```

```

10 % Author:
11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 02/15/2016: Original Code
17
18 function [z,P,pi,sac,ssd] = Tauchen86(rho, mu, sigma_e, lambda, n)
19 % Syntax:
20 % Function Name: Tauchen86
21 % -----
22 % Output: z - Markov Chain states, column vector
23 % P - Transition Matrix Pij
24 % pi - Stationary Distribution, row vector
25 % sac - Autocorrelation, approximated
26 % ssd - Standard Deviation, approximated
27 % -----
28 % Input: rho - Autocorrelation, theoretical
29 % mu - Innovation mean
30 % sigmae - Innovation Standard Deviation
31 % lambda - Range of states
32 % n - Number of States
33 % End of Syntax
34 % -----
35
36 % Initialization
37 sigma = sigma_e/(1-rho^2)^.5;
38 zlow = mu - lambda*sigma; % Set endpoints of z
39 zhight = mu + lambda*sigma;
40 z = linspace(zlow,zhigh,n); % z nodes
41 P = zeros(n); % Transition Matrix
42 m = zeros(1,n-1); % Midpoints
43 for ind = 1:1:n-1
44     m(ind) = (z(ind) + z(ind+1))/2;
45 end
46
47 % -----
48 % CALCULATE TRANSITION MATRIX
49 % -----
50 for i = 1:n
51     for j = 2:n-1
52         P(i,j) = normcdf((m(j)-(1-rho)*mu-rho*z(i))/sigma_e) - ...
53             normcdf((m(j-1)-(1-rho)*mu-rho*z(i))/sigma_e);
54     end
55     P(i,1) = normcdf((m(1)-(1-rho)*mu-rho*z(i))/sigma_e);
56     P(i,n) = 1-normcdf((m(n-1)-(1-rho)*mu-rho*z(i))/sigma_e);
57 end
58
59 % -----
60 % CALCULATE STATISTICS
61 % -----
62 % Calculate the Stationary Distribution
63 % Using Theorem 5.6
64 % Using Cholesky decomposition to solve linear system
65 X = [P' - eye(n); ones(1,n)];
66 Y = X'*[zeros(n,1);1];
67 R = chol(X'*X);
68 sd = (R\Y)'; % Probability of stationary distribution (row vector)
69 pi = sd;

```

```

70
71 % Calculate the statistics of the stationary distribution
72 sm = sd*z'; % Mean
73 ssd = sqrt((z-sm).^2*sd'); % Standard Deviation
74 sac = sum(sum((z-sm)'*(z-sm).* (diag(sd)*P)))/ssd^2;
75
76 disp('Parameters: n rho std.dev');
77 disp([n rho sigma]);
78 disp('Mean Std.dev Autocorrelation');
79 disp([sm ssd sac]);

```

代码 6.4—马尔科夫链近似AR(1)过程・函数

这里我们简单介绍一下函数脚本文件的结构。注意到代码6.4与5.2唯一的区别在于增添了18行的函数句法(syntax)说明段，余下部分与代码5.2完全相同。其中，关键词**function**提示MATLAB，该文件定义了一个函数脚本。第一个向量[z,P,pi,sac,ssd]定义了函数的输出值，Tauchen86定义了函数名，函数名之后括号内的向量为函数的输入值rho, mu, sigma_e, lambda, n。注意到MATLAB并不是通过函数名调用函数，而是通过包含这个函数的文件名来调用的。所以对应到代码6.3来说，当我们的文件名和函数名都为Tauchen86时，显然函数能够被正确的调用。如果我们在代码6.4中将函数名修改为比如Tauchen1986，那么只要文件名依然是Tauchen86.m，代码6.3依然能够正确执行。但是反过来，如果我们将文件名改为Tauchen1986.m而保持函数名不变，那么执行代码6.3时将会产生无法找到函数的错误。这里建议读者将函数名与文件名取同样的名字。另一个良好的编程习惯是如19-34行那样，清楚的写明函数的句法结构，即名称与输入输出变量，尽管MATLAB在一定程度上允许语法的灵活解释，也就是当读者没有严格的定义某个变量时，MATLAB可以通过“揣测”可能的意图使得程序运行下去。但是读者在实践中将会体会到，这种处理会造成很多难以调试的错误，所以这里提请读者注意，在定义函数的时候请务必注意语法的严格性，特别是向量和矩阵的结构。

另外读者可能注意到了，在函数文件代码6.4有很多变量的名称与主程序代码6.3当中是相同的，不熟悉高级语言但是细心的读者可能会疑问，这样是否会造成程序运行的错误。答案是不会，这牵涉到程序设计语言中变量名作用域(namespace)的概念。所谓变量名作用域，指的就是在哪部分程序段当中，变量的名称必须是唯一的。不同的作用域之间变量名称可以是相同的。对MATLAB来说，一个函数脚本就构成一个作用域。所以尽管主程序和子程序中都包含了变量rho，但是MATLAB在运行的时候，会专门对子程序构造一个变量名列表，从而避免冲突。用一个通俗一点的类比，如果我们认为一个函数对应了一个教学班，那么尽管可能两个教学班当中都有一名同学（一个变量）叫做张三，但是因为他们属于两张不同的花名册，所以我们并不会搞混。我们运行程序，模型的解（值函数与两个政策函数）在图6.4和图6.5中显示。两图当中我们可以看出消费和储蓄同样随着生产率冲击单调变化。

与上一小节相同，由于我们求解带有随机冲击模型的目的最终是为了计算数值模型仿真序列中变量的统计相关关系，所以我们这里也对模型(6.5)进行仿真，对应的程序包含在代码6.5当中。

```

1 %
2 % PROGRAM DESCRIPTION
3 %
4 %
5 % Purpose:
6 % - This code simulates the Brock and Mirman (1972, JET) stochastic
7 % dynamic programming problem with AR(1) productivity shocks.
8 % - Code 6.5 in the Lecture Notes
9 %
10 % Author:

```

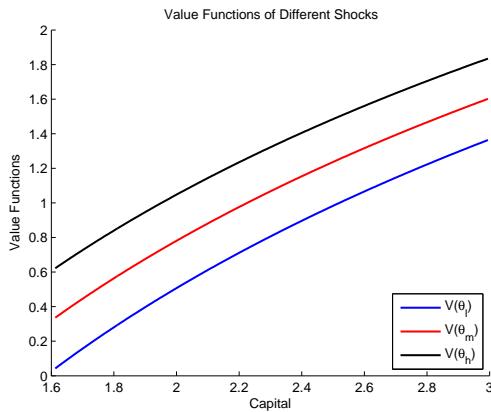


图 6.4.—AR(1)随机冲击 • 值函数

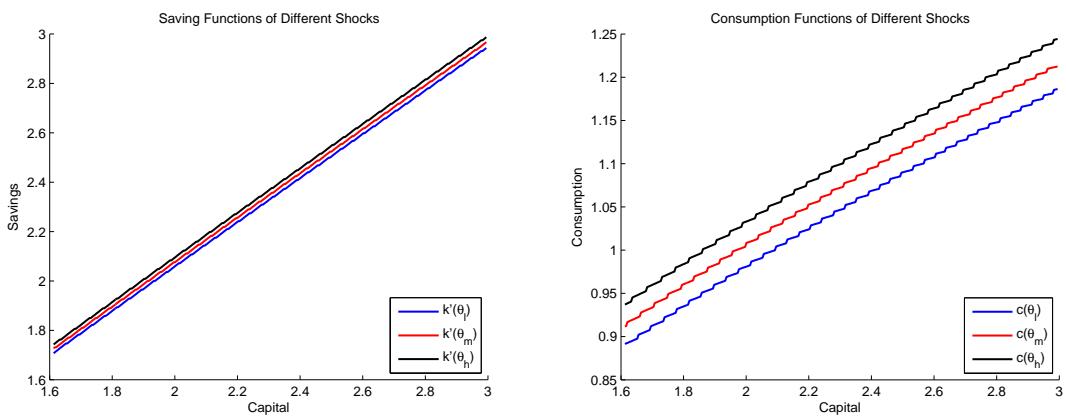


图 6.5.—AR(1)随机冲击 • 政策函数

```

11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 03/10/2016: Original Code
17
18 clc;
19 clear all ;
20
21 load brockmirmanar1_tmp.mat ;
22 % -----
23 % DATA DICTIONARY
24 % -----
25 % Declaring variables
26 burnin = 150;
27 T = burnin + 201;
28 s_theta = zeros(1,T);
29 s_k = zeros(1,T);

```

```

30 s_c = zeros(1,T);
31 s_y = zeros(1,T);
32 s_i = zeros(1,T);
33 s_shock = zeros(1,T);
34
35 % -----
36 % SIMULATION
37 % -----
38 % Draw random shocks
39 epsilon = rand(1,T);
40 s_theta(1) = (nshock+1)/2;
41 for indt = 2:1:T
42     for indj = 1:1:nshock
43         cump = sum(P(s_theta(indt-1),1:indj));
44         if epsilon(indt) < cump
45             s_theta(indt) = indj;
46             break;
47         end
48     end
49 end
50
51 % Initialization
52 indk = N/2;           % Start from deterministic steady state
53 s_k(1) = kgrid(indk) ;
54
55 for indt = 2:1:T
56     indk = pindk(s_theta(indt-1),indk);
57     s_k(indt) = polk(s_theta(indt),indk);
58     s_i(indt) = poli(s_theta(indt),indk);
59 end
60 s_shock(1:T-1) = theta(s_theta(1:T-1));
61 s_y(1:T-1) = theta(s_theta(1:T-1)').*(s_k(1:T-1).^alpha) ;
62 s_c(1:T-1) = s_y(1:T-1) - s_i(1:T-1);
63
64 xt = burnin+1:T-1;
65 % Calculate coefficient of correlation between y,c and y,k
66 rho_yc = corrcoef(s_y(xt),s_c(xt));
67 rho_yi = corrcoef(s_y(xt),s_i(xt));
68
69 % -----
70 % STATIONARY DISTRIBUTION
71 % -----
72 % Lower Bound
73 i = 1;
74 while polk(1,i) >= kgrid(i)
75     i = i+1;
76     if i >= N
77         break;
78     end
79 end
80 kslow = kgrid(i);
81
82 % Upper Bound
83 i = 1;
84 while polk(nshock,i) >= kgrid(i)
85     i = i+1;
86     if i >= N
87         break;
88     end
89 end

```

```

90 kshigh = kgrid(i-1);
91 ksmean = (kslow+kshigh)/2;
92
93 % Plot Time Series Sample Curve
94 figure(1)
95 hold on;
96 plot(xt,s_y(xt), 'LineWidth',1.5);
97 plot(xt,s_i(xt), '--r', 'LineWidth',1.5);
98 plot(xt,s_c(xt), '-.k', 'LineWidth',1.5);
99 legend('Output', 'Investment', 'Consumption', 'Location', 'NorthEast');
100 xlabel('Time')
101 ylim([0.15 1.8]);
102 title('Simulation of AR(1) Brock-Mirman')
103 hold off;
104
105 % Plot Stochastic Deviation from Steady State
106 % For understanding the stochastic behavior of the model
107 % TFP Shock
108 figure(2)
109 hold on;
110 plot(xt,s_shock(xt)-theta_ss,'LineWidth',1.5);
111 plot(xt,0,'-.r','LineWidth',2);
112 xlabel('Time')
113 ylim([-0.08 0.08]);
114 title('Deviation from Deterministic Steady State: TFP Shock')
115 hold off;
116
117 % Capital
118 figure(3)
119 hold on;
120 plot(xt,s_k(xt)-kss,'LineWidth',1.5);
121 plot(xt,s_k(xt)-ksmean,'r','LineWidth',1.5);
122 legend('Deterministic', 'Stochasitc')
123 plot(xt,0,'-.r','LineWidth',2);
124 xlabel('Time')
125 ylim([-0.3 1.0]);
126 title('Deviation from Steady States: Capital')
127 hold off;
128
129 % Stationary Distribution
130 figure(4)
131 hold on;
132 plot(kgrid,polk(1,:),'LineWidth',1.5);
133 plot(kgrid,kgrid,'r','LineWidth',1.5);
134 plot(kgrid,polk(nshock,:),'k','LineWidth',1.5);
135 legend('theta_n', '45-Degree', '\theta_1', 'Location', 'SouthEast')
136 xlabel('Capital')
137 xlim([2 3.4])
138 title('Stationary Distribution')
139 hold off;

```

代码 6.5.—AR(1)随机冲击仿真

代码6.5与6.2的结构基本相同，不同之处在于38-49行处对于随机冲击的模拟，这里主要有两处不同。首先，由于此时生产率冲击是一个AR(1)过程，所以对应的马氏链的转移阵每行是不同的，也就是说，本期的生产率水平会影响下一期不同水平生产率出现的概率。这体现在42行处，针对上一期

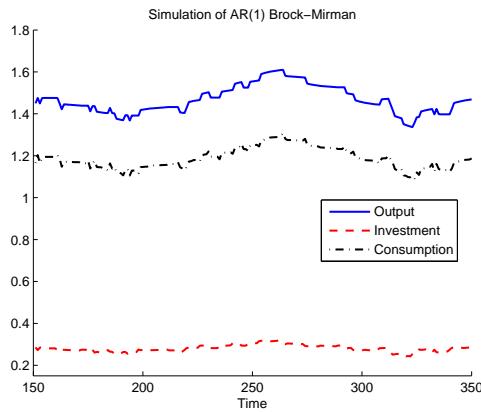


图 6.6.—AR(1)随机冲击・仿真

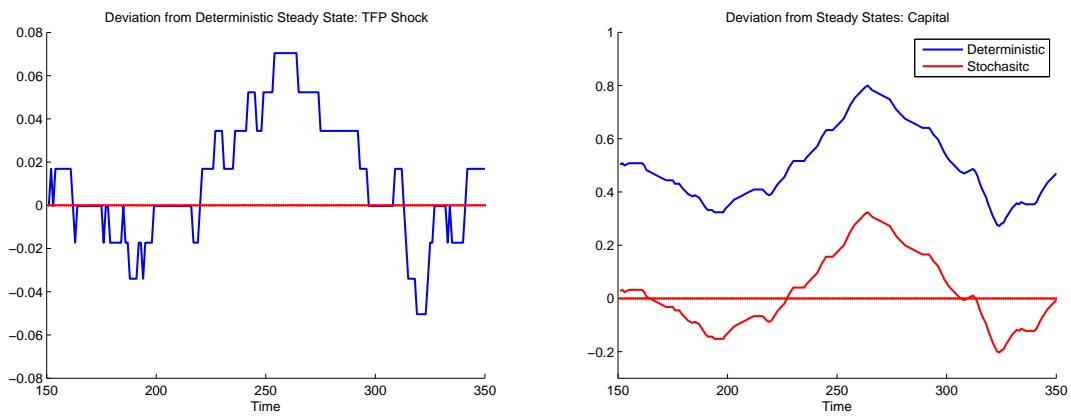


图 6.7.—随机稳态

不同的`s_theta(indt-1)`, 本期要应用不同的转移概率`P(s_theta(indt-1),1:indj)`。其次是由于此时马氏链的状态多于两个, 所以我们不能简单的用0.5作为分界, 同时又因为各个状态的概率不同, 所以也不能直接用1除以状态的总数。不过代码6.2 中的基本想法依然适用。比如我们设转移阵对应的概率为 $\mathbf{P} = [0.3, 0.2, 0.5]$, 这是说给定当前的状态, 下一期生产率为状态1、2和3的概率分别为0.3, 0.2和0.5。所以我们需要一个简单的算法, 能够让我们根据抽取的随机数定义三个条件, 分别以0.3, 0.2和0.5为概率成立。一个简单的做法就是在 $[0, 1]$ 均匀分布上抽取任意随机数, 容易看出随机数落在 $[0, 0.3], [0.3, 0.5]$ 和 $[0.5, 1]$ 的概率恰好为0.3, 0.2和0.5。用程序语句表达, 就是代码43-47行, 即我们首先检验随机数是否在 $[0, 0.3]$ 中, 如果在, 那么就表示下一期的生产率冲击为状态1 (45行), 46行的`break`语句是说跳出当前的`for`循环 (内层循环42-48行)。如果随机数不在 $[0, 0.3]$ 中, 再检验其是否小于 $0.3 + 0.2 = 0.5$, 以此类推。注意这里只需要检验随机数是否小于0.5是因为前一次循环我们已经检验过该随机数大于等于0.3。运行该程序, 结果在图6.6中显示。

这里有一个值得读者留意的地方。读者可能会认为, 实际仿真当中的资本、产出等变量应该是围绕稳态周围波动的, 这个猜想这里是不成立的。我们在图的左右两图中分别描绘了对应了图6.6的仿

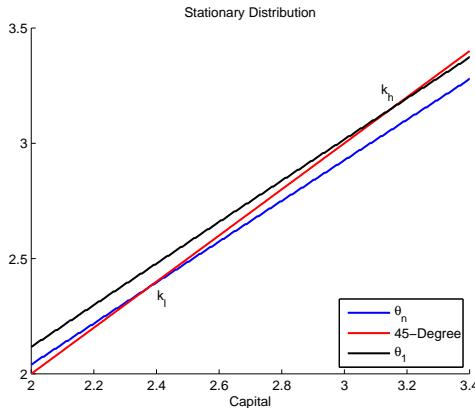


图 6.8—平稳分布

真，生产率与资本存量分别关于他们稳态水平的偏离。图中我们可以看出，尽管生产率是围绕稳态均值波动的，资本的水平要系统性高于稳态水平（右图中的蓝线）。这是由储蓄函数的非对称性造成的。尽管生产率冲击对应的平稳分布是一个对称分布，但是资本的平稳分布并不对称。这是由于储蓄率随生产率的非线性造成的（图6.5）。也就是说，如果生产率1%的增加能够造成比如1.5%的储蓄增加，但是相应的1%的生产率减小只会造成0.5%的储蓄增加，那么一个对称的生产率分布就会对应一个非对称的资本分布，此时资本的平稳分布更多的密度集中在上分布处。这里的非线性主要是由CRRA效用函数的非线性（进而是值函数的非线性）所导致的。不过读者从图6.6可以看出，尽管资本并不围绕确定性稳态波动，但是看上去资本依然是围绕某个均值进行波动，那么我们能否求出这个均值——有时候也称为随机稳态(*stochastic steady state*)呢？答案是肯定的。代码6.2的67-89行的作用就在于此。直观上来说，资本路径在仿真中的平稳分布是一个区间，只要资本的水平落入这个区间，那么生产率处于最高水平，在区间上界处下一期的资本也不会高于当前水平；同理，即便生产率处于最低水平，在区间下界处下一期资本也不会低于当前水平。于是求平稳分布实际上就是求区间的上下界。容易验证，区间的上下界分别为生产率冲击最低与最高水平对应的资本政策函数与45°线的交点，如图6.8所示。当 $k < k_l$ 时，由于 $g_k(k, \theta) > k$ 恒成立，所哪怕生产率水平最低也有 $k_{t+1} > k_t$ ；反过来，当 $k > k_h$ 时，由于 $g_k(k, \theta) < k$ 对最高的生产率冲击也成立，故 $k_{t+1} < k_t$ ，所以资本存量处在这两个范围内的时候是不稳定的，特别的，资本序列会单调的向中心区域收敛。而当资本存量处于 $[k_l, k_h]$ 当中时，资本的波动是双向的。对应了第五章的概念，我们知道 $k < k_l$ 和 $k > k_h$ 时，对应的 k 是瞬时的，而当 $k \in [k_l, k_h]$ 时， k 是常返的。67-89行算法所需要的性质是生产函数必须满足稻田条件，且与45°线交点的邻域附近斜率小于等于1，这里的原理与索罗模型当中稳态相似，请读者自行思考。回到我们的仿真序列图6.6中，回忆我们在画图的时候舍弃了前面一部分有限期的样本，这里的原因就是要舍弃样本曲线没有进入平稳分布的那部分，对应到代码6.2中，由于我们在51行出设定了仿真从确定性稳态出发，所以读者可以自行验证，在仿真刚开始的部分，资本序列实际上构成一个上鞅(*upper-martingale*，资本序列围绕一个单增的趋势波动)，直到资本水平进入平稳区间之后，资本序列由一个平稳分布刻画。反过来，如果我们从一个较高的初值出发，资本序列的初始部分将是一个下鞅(*lower-martingale*)。

6.3 真实经济周期模型

现在我们在模型(6.4)中引入劳动供给决策，这对应了第四章中的模型(4.20)，也就是我们引入诸多数学工具和启发性例子的最终目的。此时我们在消费和资本的政策函数以外，还要求解消费者闲暇的政策函数 $l = g_l(k; \theta)$ 。具体来说，此时我们要求解的模型的序列形式为

$$\begin{aligned} & \max_{\{c_t, l_t, k_{t+1}\}_0^\infty} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \frac{(c_t^\sigma l_t^{1-\sigma})^{1-\gamma} - 1}{1-\gamma} \\ & \text{s.t.} \\ & c_t + k_{t+1} - (1-\delta)k_t = \theta_t k_t^\alpha (1-l_t)^{1-\alpha} \\ & \log(\theta_t) = \rho \log(\theta_{t-1}) + \varepsilon_t, \varepsilon_t \stackrel{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2) \\ & k_0, \theta_0 \text{已知}, c_t, k_t \geq 0, 0 \leq l_t \leq 1. \end{aligned}$$

其中 σ 是对消费者而言消费与闲暇之间的替代弹性。上述问题对应的动态优化问题为

$$\begin{aligned} V(k; \theta) &= \max_{c, k', l} \left\{ \frac{(c^\sigma l^{1-\sigma})^{1-\gamma} - 1}{1-\gamma} + \beta \mathbb{E}_\theta[V(k'; \theta')] \right\} \\ & \text{s.t.} \\ & c + k' - (1-\delta)k = \theta k^\alpha (1-l)^{1-\alpha} \\ & \log \theta' = \rho \log \theta + \varepsilon', \varepsilon' \stackrel{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2). \end{aligned} \quad (6.6)$$

对比优化问题(6.6)与(6.5)，除了生产函数和效用函数的计算此时需要带入 l 的最优解以外，我们发现最大的区别在于我们需要求解一个额外的控制变量 l 的最优值。由于消费者总的时间禀赋 $l \in [0, 1]$ 是一个有界集，于是我们可以用一列节点 $\mathbf{l} = [l_1, \dots, l_n]$ 对 $[0, 1]$ 区间进行近似。与资本存量的节点相组合，此时我们就有了一组二维节点 (k_i, l_j) ，容易看出，给定 (k_i, l_j) 的取值，由资源约束我们可以直接求出消费者的消费，进而求出当期的效用与未来值。所以我们可以继续应用值函数迭代的思路，对控制变量偶对 (k_i, l_j) 所对应的效用进行穷举，之后选择最大的那个，所得的 k_i 与 l_j 就是当前资本存量与生产率冲击水平下的最优选择，本节当中我们便采用这个算法。

不过这并不是唯一的求解上述模型的办法。回忆我们在第四章中得到的方程(4.13):

$$w_t \frac{\partial u}{\partial c_t} = \frac{\partial u}{\partial l_t},$$

我们在当中带入(6.6)中的函数形式，可以得到

$$(1-\alpha)\theta k^\alpha (1-l)^{-\alpha} = \frac{1-\sigma}{\sigma} \cdot \frac{c}{l}.$$

此时若已知 k'_j 的取值，上式与资源约束一起构成了一个关于 (c, l) 的二元非线性方程组，从而可以应用MATLAB的固有函数`fsolve`求解。这两种方法的解应该是等价的，由于第一种方法与我们截止目前所学的方法联系更为紧密，同时穷举法的结果也来得更加稳健一些，所以本节我们使用第一种算法。第二种解法只需要在第一种解法程序的基础上做很简单的修改，在此便留给读者作为习题练习。由于程序的结构与前两节我们遇到的模型并没有太大的区别，于是我们省略伪代码。我们在此采用Cooley and Prescott (1995)中的参数，具体来说，我们设 $\gamma = 1$ ，于是效用函数此时成为对数线性的

$$u(c, l) = \sigma \log c + (1-\sigma) \log l,$$

此时 $\sigma = 0.36$ 。再设贴现率 $\beta = 0.987$, 资本份额 $\alpha = 0.40$, 折旧率 $\delta = 0.025$ 。^① 我们依然保持上一节中关于生产率冲击的参数假设, 即 $\rho = 0.95, \sigma_\varepsilon = 0.00712$ 。

代码6.6包含了求解模型(6.6)政策函数的代码。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code solves the Hansen (1985,JME) model.
7 % - Code 6.6 in the Lecture Notes
8 %
9 % Author:
10 % Xin Tang @ Wuhan University, Spring 2016
11 %
12 % Record of Revisions:
13 % Date: Description of Changes
14 % ===== =====
15 % 03/11/2016: Original Code
16
17 clc;
18 clear all ;
19 tic; % Initialize a stopwatch
20 %
21 % DATA DICTIONARY
22 %
23 % Parameterizations
24 alpha = 0.40 ;
25 Asigma = 0.36 ;
26 A = (1-Asigma)/Asigma;
27 beta = 0.987 ;
28 delta = 0.025;
29 rho = 0.95;
30 sigmae = 0.00712;
31 mu = 0;
32 nshock = 9 ;
33 m = 3;
34 N = 1000 ; % Number of Capital Grids
35 NL = 300 ; % Number of Labor Grids
36 tolv = 1e-9 ; % Tolerance level
37 toln = 10000 ; % Maximum Number of Loops in While-loop,
38 % to avoid never-ending dead-loop
39 increase = 1; % If == 1, previous solution is used to accelerate
40
41 % Initialize Stochastic Shocks
42 % Using classical Tauchen (1986) implemented by Code 5.2
43 [theta,P,pi,sac,ssd] = Tauchen86(rho, mu, sigmae, m, nshock);
44 theta = exp(theta)';
45 theta_ss = pi*theta ;
46
47 % Calculate the Steady State
48 aux_para = [alpha A beta delta theta_ss];
49 xguess = [1.0 15 1/3];
50 xsolve = fsolve(@(x) code6_6_ss(x,aux_para),xguess);
51 css = xsolve(1);

```

^① 注意到此处折旧率要大于Cooley and Prescott (1995)中的参数, 原因是Cooley and Prescott (1995)当中考虑了人口与资本增长, 在当前的例子中我们不考虑平衡增长路径。我们将在下一节对于校准的介绍中详细解释两者的不同。事实上, Cooley and Prescott (1995)去掉平衡增长路径之后就是Hansen (1985)的模型。

```

52 kss = xsolve(2);
53 nss = xsolve(3);
54 clear aux_para; % Remove unnecessary auxillary variable(s)
55
56 % Load previous solutions to speed up algorithm
57 if increase
58     load hansen1985_700 V0 kgrid;
59     Vold = V0;
60     oldgrid = kgrid;
61     clear V0 kgrid;
62 end
63
64 % Set Grids
65 kmin = 0.80*kss;
66 kmax = 1.30*kss;
67 kgrid = linspace(kmin, kmax, N);
68 ngrid = linspace(1e-6, 1-1e-6, NL);
69
70 % Declaring Variables
71 V0 = zeros(nshock,N); % Vt
72 V1 = zeros(nshock,N); % Vt+1
73 VT = zeros(1,N); % Vtj(theta)
74 polc = zeros(nshock,N); % Policy Function for Consumption
75 polk = zeros(nshock,N); % Policy Function for Saving
76 poln = zeros(nshock,N); % Policy Function for Labor
77 poli = zeros(nshock,N); % Policy Function for Investment
78 pindk = zeros(nshock,N); % Saving policy index, used in simulation
79
80 % Use linear interpolation to initialize V0 using previous solution
81 if increase
82     for indt = 1:1:nshock
83         V0(indt,:) = interp1(oldgrid,Vold(indt,:),kgrid);
84     end
85 end
86
87 % Auxillary Variables
88 optim = -1; % Store the index of optimal choice
89 eps = 1; % Evaluate convergence of Vt
90 loopn = 1; % Calculate the number of loops
91 aux_poln = zeros(nshock,N^2); % Store intermediate optimal labor
92 aux_u = zeros(NL,1); % Used in Calculating intermediate labor
93 u = zeros(nshock,N^2); % Store Raw Utility
94
95 % Calculate the utility level when current capital is ki, future capital
96 % is kj, shock is thetaj and labor is lm
97 for indt = 1:1:nshock
98     disp(indt)
99     for indi = 1:1:N
100         for indj = 1:1:N
101             for indl = 1:1:NL
102                 aux_c = theta(indt)*kgrid(indi)^alpha...
103                     *ngrid(indl)^(1-alpha) - kgrid(indj) ...
104                     + (1-delta)*kgrid(indi);
105                 if aux_c <= 0
106                     aux_c = 1e-11;
107                 end
108                 aux_u(indl) = log(aux_c) + A*log(1-ngrid(indl));
109             end
110             [xtemp optim] = max(aux_u);
111             aux_poln(indt,(indi-1)*N+indj) = ngrid(optim);

```

```

112         u(indt,(indi-1)*N+indj) = xtemp ;
113     end
114 end
115
116 % -----
117 % VALUE FUNCTION ITERATION
118 % -----
119 % Iterate Vt until convergence
120 while (eps > tolv) && (loopn < tol_n)
121     for indt = 1:1:nshock
122         for indi = 1:1:N
123             for indj = 1:1:N
124                 EV = P(indt,:)*VO(:,indj);
125                 VT(indj) = u(indt,(indi-1)*N+indj) + beta*EV;
126             end
127             V1(indt,indi) = max(VT);
128         end
129     end
130     eps = max(max(abs(V1-VO)));
131     VO = V1;
132     loopn = loopn+1;
133     if mod(loopn,50) == 0
134         disp(['Loop=' ,num2str(loopn)])
135         disp(['eps=' ,num2str(eps)])
136     end
137 end
138
139 % -----
140 % OPTIMAL POLICIES
141 % -----
142 % Using the numerical approximation of V to solve the optimization problem
143 for indt = 1:1:nshock
144     for indi = 1:1:N
145         EVT = P(indt,:)*V1;
146         VT = u(indt,(indi-1)*N+1:indi*N) + EVT;
147         % Find the saving that leads to the highest utility
148         [xtemp optim] = max(VT) ;
149         pindk(indt,indi) = optim;
150         polk(indt,indi) = kgrid(optim);
151         poln(indt,indi) = aux_poln(indt,(indi-1)*N+optim) ;
152     end
153 end
154
155 polc = theta*kgrid.^alpha.*poln.^(1-alpha)...
156     - polk + ones(nshock,1)*(1-delta)*kgrid ;
157 poli = polk - ones(nshock,1)*(1-delta)*kgrid ;
158 time = toc;    % Paired with tic, calculate total running time in between
159 disp(['Total Running Time=' ,num2str(time), ' secs'])
160
161 % Plot the value functions and policy functions
162 % Value functions for different shocks and Expected value functions
163 auxmid = (nshock + 1)/2;
164 indplot = [auxmid-2 auxmid auxmid+2];
165 figure(1)
166 hold on;
167 plot(kgrid,V1(indplot(1),:),'LineWidth',1.5) ;
168 plot(kgrid,V1(indplot(2),:),'r','LineWidth',1.5) ;
169 plot(kgrid,V1(indplot(3),:),'k','LineWidth',1.5) ;
170 legend('V(\theta_l)', 'V(\theta_m)', 'V(\theta_h)', 'Location', 'SouthEast')
171 title('Value Functions of Different Shocks')

```

```

172 xlabel('Capital') ;
173 ylabel('Value Functions')
174 hold off;
175
176 % Saving functions for different shocks
177 figure(2)
178 hold on;
179 plot(kgrid,polk(indplot(1),:),'LineWidth',1.5) ;
180 plot(kgrid,polk(indplot(2),:),'r','LineWidth',1.5) ;
181 plot(kgrid,polk(indplot(3),:),'k','LineWidth',1.5) ;
182 legend('k' '(\theta_l)', 'k' '(\theta_m)', 'k' '(\theta_h)', 'Location', 'SouthEast')
183 title('Saving Functions of Different Shocks')
184 xlabel('Capital') ;
185 ylabel('Savings')
186 hold off;
187
188 % Consumption functions for different shocks
189 figure(3)
190 hold on;
191 plot(kgrid,polc(indplot(1),:),'LineWidth',1.5) ;
192 plot(kgrid,polc(indplot(2),:),'r','LineWidth',1.5) ;
193 plot(kgrid,polc(indplot(3),:),'k','LineWidth',1.5) ;
194 legend('c' '(\theta_l)', 'c' '(\theta_m)', 'c' '(\theta_h)', 'Location', 'SouthEast')
195 title('Consumption Functions of Different Shocks')
196 xlabel('Capital') ;
197 ylabel('Consumption')
198 hold off;
199
200 % Labor functions for different shocks
201 figure(4)
202 hold on;
203 plot(kgrid,poln(indplot(1),:),'LineWidth',1.5) ;
204 plot(kgrid,poln(indplot(2),:),'r','LineWidth',1.5) ;
205 plot(kgrid,poln(indplot(3),:),'k','LineWidth',1.5) ;
206 legend('n' '(\theta_l)', 'n' '(\theta_m)', 'n' '(\theta_h)', 'Location', 'SouthEast')
207 title('Labor Functions of Different Shocks')
208 xlabel('Capital') ;
209 ylabel('Labor')
210 hold off;
211
212 % Investment functions for different shocks
213 figure(5)
214 hold on;
215 plot(kgrid,poli(indplot(1),:),'LineWidth',1.5) ;
216 plot(kgrid,poli(indplot(2),:),'r','LineWidth',1.5) ;
217 plot(kgrid,poli(indplot(3),:),'k','LineWidth',1.5) ;
218 legend('n' '(\theta_l)', 'n' '(\theta_m)', 'n' '(\theta_h)', 'Location', 'SouthEast')
219 title('Labor Functions of Different Shocks')
220 xlabel('Capital') ;
221 ylabel('Investment')
222 hold off;
223
224 save hansen1985_1k.mat;

```

代码 6.6.—真实经济周期模型

其中在47-54行求解模型稳态时，需要调用函数code6_6_ss，该函数代码由代码6.7给出。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code defines the steady state of Hansen (1985, JME)
7 % - Function code to be called in code 6.6.
8 % - Code 6.7 in the Lecture Notes.
9 %
10 % Author:
11 % Xin Tang @ Wuhan University, Spring 2016
12 %
13 % Record of Revisions:
14 % Date: Description of Changes
15 % =====
16 % 03/11/2016: Original Code
17
18 function y = code6_6_ss(x,par)
19 % Syntax:
20 % Function Name: code6_6_ss
21 %
22 % Output: y - R.H.S of equations system
23 %
24 % Input: x(1) - Consumption
25 % x(2) - Capital
26 % x(3) - Labor
27 % par - Model Parameters [alpha A beta delta theta_ss]
28 %
29 % End of Syntax
30 %
31 c = x(1);
32 k = x(2);
33 l = 1 - x(3);
34 alpha = par(1);
35 A = par(2);
36 beta = par(3);
37 delta = par(4);
38 theta_ss = par(5);
39
40 y = zeros(1,3);
41 y(1) = theta_ss*k^alpha*(1-l)^(1-alpha) - c - delta*k;
42 y(2) = A/l - ((1-alpha)*theta_ss*k^alpha*(1-l)^(-alpha))/c;
43 y(3) = beta*(alpha*theta_ss*k^(alpha-1)*(1-l)^(1-alpha)+1-delta)-1;

```

代码 6.7.—真实经济周期模型稳态

前述两个代码中有一些细节值得我们仔细分析。首先，在47-54行中求解模型稳态时，我们无法得到稳态消费、资本与劳动的解析解。具体的，若用 c, k 和 l 分别代表稳态中的消费、资本与闲暇，那么容

易验证，此时模型的稳态是由下述三个非线性方程组成的方程组描述的：

$$\begin{aligned} c + \delta k &= \theta k^\alpha (1 - l)^{1-\alpha} \\ \frac{A}{l} &= \frac{(1 - \alpha)\theta k^\alpha (1 - l)^{-\alpha}}{c} \\ \frac{1}{\beta} &= \alpha \theta k^{\alpha-1} (1 - l)^{1-\alpha} + 1 - \delta. \end{aligned}$$

上述方程组的参数是由模型的参数 $\alpha, \sigma, \beta, \delta$ 和生产率冲击的均值 θ 给定的。^① 上述方程组的表达式在代码6.7中给出。代码6.6的50行调用了固有函数`fsolve`对该方程组进行求解。`fsolve`利用了牛顿-拉夫森(*Newton-Raphson*)算法对非线性方程组进行求解，这里我们忽略该算法的技术细节，有兴趣的同学可以参考数值计算方法相关的教材。不过该算法是一个局部收敛(*local convergence*)算法。这是说该算法需要从一个比较接近真实解的位置出发，最后才会收敛到真实解。从任意初始值出发并不能保证算法给出正确的解。因为这个原因，我们需要在49行中定义初始猜，读者在实际应用`fsolve`的时候可能需要反复尝试不同的初始猜。之后在50行调用`fsolve`的时候，输入的第二组变量就是对应了每个未知量的初始猜，注意到此时初始猜的顺序必须与代码6.7中定义的顺序(31-33行)完全相同。细心的读者可能会会想到我们反复应用的值函数迭代算法，我们知道对值函数迭代算法，初始猜 V_0 可以任意选取，序列 $\{V_i\}$ 最后总会收敛到 V 的真值。对于这样的算法，我们称为全局收敛(*global convergence*)算法。^② 第50行中`fsolve`的第一个变元`@(x) code6_6_ss(x, aux_para)`是说此时要求解的方程是以 x 为变量，`aux_para`为参数的方程组。读者参考MATLAB内建的`fsolve`帮助文档中的例子可以发现，如果我们所要求解的方程组参数是确定的(比如我们知道方程组是由 $x + y = 1$ 和 $x - y = 0.5$ 两个方程刻画的， x, y 前的参数是不变的)，那么此时第一个变元处只需要以字符串形式输入函数文件名`'code6_6_ss'`即可。此处我们将自变量与参数变量分开，主要是因为通常在宏观经济学当中我们要对同一个模型在不同参数组合下的性质进行分析，我们对于每组参数组合如果都要专门写一个对应的稳态函数`code6_6_ss`显然不是一个很聪明的做法，因此50行处我们应用的处理方法适用于方程组的函数形式一定，但是参数值在不同次运行时不同的情形。^③

其次，我们在56-62行和79-84行两个循环体中应用低精度(节点数较少)时的解对求解高精度(节点数较多)解时的算法执行进行了加速。运行模型，我们在图6.9当中显示经济周期模型的劳动决策函数在资本取350个节点、劳动为100个节点(左图)和资本取1000个节点、劳动取350个节点(右图)时的性态。从图6.9中我们可以发现，为了得到较为平滑的解，对真实经济周期模型，我们需要应用非常多的节点进行近似，而这意味着计算时间几何倍数的提高。为获得右图中的解，在作者的个人计算机上，加速处理过之后，依然需要18,981秒即五个多小时的运行才能得到结果，在没有加速的情况下将会需要将近8小时的运行时间。所以通常我们在检验程序的时候，会首先从节点数较少的情形出发，得到一个比较粗糙的解，确认这个粗糙的解的定性性质符合经济直觉之后，再利用较多的节点数求精确解。不过这个过程通常需要尝试许多组的参数组合，来确定一个比较合适的计算精度。而我们知道，值函数迭代法尽管是一个全局算法，但是如果选取与真实解比较相似的初始猜，无疑会大大的加速算法的执行效率。此处采取的加速策略便是如此。具体来说，我们在56-62行中读入低精度的

^① 在实际计算中，我们通常应用 θ 的马氏链近似所对应的平稳分布的均值，不过这对计算结果的影响非常小。原因就是我们在上一节中看到的，确定性稳态与对应的仿真序列的稳态分布的均值通常是不同的。

^② 本教程中不涉及的另一种常用的求解动态优化问题的算法，即政策函数迭代算法就是一个局部收敛算法。政策函数迭代算法的收敛速度通常要快于值函数迭代，但是这需要一个比较准确的初始猜。实践中我们通常采用比较粗糙的值函数迭代算法的解作为初始猜，应用政策函数迭代法求解节点较为密集的参数下的模型。有兴趣的同学可以参考Coleman (1990)。

^③ 这里也可以应用全局变量来传递参数，不过由于全局变量在使用中非常容易造成预期外的错误，所以在本教程中不做介绍。有兴趣的读者可以自行参考其他教科书学习。

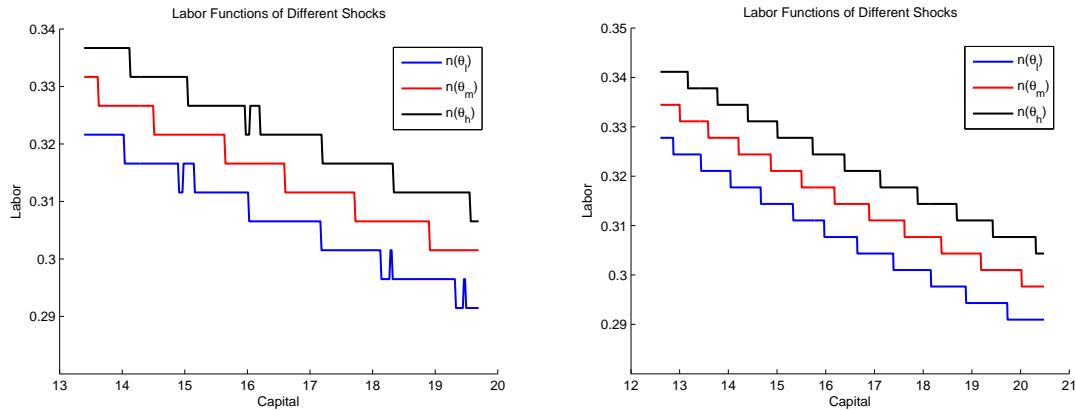


图 6.9.—劳动决策函数

解（由于显而易见的原因，在第一次求解低精度解的时候与加速相关的两段代码需要被注释掉）。由于低精度所对应的函数表在许多高精度节点上并没有对应的取值，所以我们这里采用分段线性函数对这些取值进行近似（79-84行），这里我们应用了MATLAB的进行一维线性插值的固有函数`interp1`，关于`interp1`的具体句法请读者参考MATLAB的内建文档。简单来说，假设我们在低精度的解中得到了 $(k_1, V(k_1))$ 和 $(k_3, V(k_3))$ 的取值，在高精度解的求解中我们需要知道 $(k_2, V(k_2))$ ， $k_1 < k_2 < k_3$ 的近似值。函数`interp1`的作用，就是求出过 $(k_1, V(k_1))$ 和 $(k_3, V(k_3))$ 两点的直线方程 $V(k) = a + b \cdot k$ ，之后用 $a + b \cdot k_2$ 作为 $V(k_2)$ 的近似值。读者可以自行验证，这个加速的效果是极为明显的。不使用加速算法时，经过50次迭代之后值函数的收敛精度是 10^{-1} ，但是使用了低精度解进行加速时，经过50次迭代之后，值函数将以 10^{-6} 精度收敛。另外，从图6.9中可以看出，劳动供给只在1/3左右波动，所以在实际求解当中我们可以只对 $1 - l$ 在1/3的一个邻域进行近似。

第三，对比代码6.1，6.3和6.6，我们发现三者的主结构是完全相同的。代码6.3相比于6.1转移概率阵发生了变化，而代码6.6相对于6.3其实只是增加了109-111行求最优劳动供给一步，其余的代码都是完全重复利用的。这里给我们的一个提示是，当求解一个比较大的问题的时候，前沿的顶级宏观经济学家一般也并不是一开始就把整个模型求解。通常学者们都会首先将问题简化为一个已知解法的熟悉的问题，之后慢慢向基础模型当中添加元素，最后再求解一开始想要求解的较为复杂的模型。宏观经济学要处理的模型通常是由一个步骤很多的线性过程组成的，所以通常而言适用于“面向对象”编程思路的将一个大的任务拆解成多个相互独立的小任务的方法并不使用。但是我们可以像本章几个模型之间的递进关系一样对任务进行拆解。具体来说，就是从一个大型任务出发，构造出一系列元素递增的不同版本，各个版本之间有相同的计算过程。这样我们可以通过在较为简单版本的模型当中对程序的某个模块进行测试，从而在求解较为复杂版本的模型时，我们就可以比较确定的认为程序的某一个部分至少是没有错误的。比如对比代码6.6和6.3，两者在值函数迭代部分的代码是完全相同的，所以当我们在代码6.3的执行当中确定了值函数迭代部分没有错误，那么如果我们调试代码6.6的时候遇到错误，我们就能够判断，代码的错误有非常大的可能是出在求解最优劳动——也就是新添加的那部分上。所以，本章我们之所以通过三个步骤来求解真实经济周期模型，按照由简到难的顺序进行教学是一方面考虑，更重要的是以一个具体的例子向读者解释在实际研究的时候，应该以何种方式将程序模块化，从而将一个大型任务分解成几个相对容易处理的小规模任务。

我们用代码6.8对真实经济周期模型进行仿真，注意到代码6.8与代码6.5是非常相似的。主要的改变

有三个，第一是要应用劳动的政策函数计算劳动投入在经济周期中的波动，第二是对于仿真的序列应用HP滤波进行处理以与真实经济数据进行比较，下一小节当中我们会具体的解释这样做的原因。第三，这里我们重复100次仿真以消除由于不同仿真之间随机数抽取造成的统计矩的随机波动。

```

1 % -----
2 % PROGRAM DESCRIPTION
3 % -----
4 %
5 % Purpose:
6 % - This code simulates the Hansen (1985,JME) RBC model.
7 % - Code 6.8 in the lecture notes
8 %
9 % Author:
10 % Xin Tang @ Wuhan University, Spring 2016
11 %
12 % Record of Revisions:
13 % Date: Description of Changes
14 % ===== =====
15 % 03/14/2016: Original Code
16
17 clc;
18 clear all ;
19
20 load hansen1985_1k.mat
21
22 % -----
23 % DATA DICTIONARY
24 % -----
25 % Declaring variables
26 burnin = 350;
27 T = burnin + 151;
28 xt = burnin+1:T-1;
29 nsim = 100;           % Simulate 100 times to remove random errors
30 lambda = 1600;        % HP-filter weight
31 % Sample series
32 s_theta = zeros(nsim,T);
33 s_k = zeros(nsim,T);
34 s_c = zeros(nsim,T);
35 s_y = zeros(nsim,T);
36 s_i = zeros(nsim,T);
37 s_n = zeros(nsim,T);
38 s_shock = zeros(nsim,T);
39
40 % HP-Trend
41 s_c_t = zeros(nsim,T);
42 s_y_t = zeros(nsim,T);
43 s_i_t = zeros(nsim,T);
44 s_n_t = zeros(nsim,T);
45 s_shock_t = zeros(nsim,T);
46
47 % HP-Cycles
48 s_c_c = zeros(nsim,T);
49 s_y_c = zeros(nsim,T);
50 s_i_c = zeros(nsim,T);
51 s_n_c = zeros(nsim,T);
52 s_shock_c = zeros(nsim,T);
53
54 % Statistics
55 % Standard Deviations

```

```

56 std_y = zeros(nsim,1);
57 std_c = zeros(nsim,1);
58 std_i = zeros(nsim,1);
59 std_s = zeros(nsim,1);
60 std_n = zeros(nsim,1);
61
62 % Correlation with output
63 r_y = zeros(nsim,1);
64 r_c = zeros(nsim,1);
65 r_i = zeros(nsim,1);
66 r_s = zeros(nsim,1);
67 r_n = zeros(nsim,1);
68
69 % -----
70 % SIMULATION
71 % -----
72 % Draw random shocks
73 epsilon = rand(nsim,T);
74 s_theta(:,1) = (nshock+1)/2;
75
76 % Run 100 Simulations
77 for indsim = 1:1:nsim
78 % Individual simulations
79 for indt = 2:1:T
80     for indj = 1:1:nshock
81         cump = sum(P(s_theta(indsim,indt-1),1:indj));
82         if epsilon(indsim,indt) < cump
83             s_theta(indsim,indt) = indj;
84             break;
85         end
86     end
87 end
88
89 % Initialization
90 indk = N/2;           % Start from deterministic steady state
91 s_k(indsim,1) = kgrid(indk) ;
92
93 for indt = 2:1:T
94     indk = pindk(s_theta(indsim,indt-1),indk);
95     s_k(indsim,indt) = polk(s_theta(indsim,indt),indk);
96     s_n(indsim,indt) = poln(s_theta(indsim,indt),indk);
97     s_i(indsim,indt) = poli(s_theta(indsim,indt),indk);
98 end
99 s_shock(indsim,1:T-1) = theta(s_theta(indsim,1:T-1));
100 s_y(indsim,1:T-1) = theta(s_theta(indsim,1:T-1))'...
101    .*(s_k(indsim,1:T-1).^alpha).*(s_n(indsim,1:T-1).^(1-alpha)) ;
102 s_c(indsim,1:T-1) = s_y(indsim,1:T-1) - s_i(indsim,1:T-1);
103 s_shock(indsim,1:T-1) = s_y(indsim,1:T-1)./s_n(indsim,1:T-1);
104
105 % HP-Filter Simulated Data
106 [s_shock_c(indsim,xt),s_shock_t(indsim,xt)] =...
107    hp2(log(s_shock(indsim,xt))',lambda);
108 [s_y_c(indsim,xt),s_y_t(indsim,xt)] =...
109    hp2(log(s_y(indsim,xt))',lambda);
110 [s_i_c(indsim,xt),s_i_t(indsim,xt)] =...
111    hp2(log(s_i(indsim,xt))',lambda);
112 [s_c_c(indsim,xt),s_c_t(indsim,xt)] =...
113    hp2(log(s_c(indsim,xt))',lambda);
114 [s_n_c(indsim,xt),s_n_t(indsim,xt)] =...
115    hp2(log(s_n(indsim,xt))',lambda);

```

```
116 % -----
117 % CALCULATE BUSINESS CYCLE STATISTICS
118 % -----
119 % Standard Deviations in percentage
120 std_y(indsim,1) = std(s_y_c(indsim,xt),1);
121 std_c(indsim,1) = std(s_c_c(indsim,xt),1);
122 std_i(indsim,1) = std(s_i_c(indsim,xt),1);
123 std_s(indsim,1) = std(s_shock_c(indsim,xt),1);
124 std_n(indsim,1) = std(s_n_c(indsim,xt),1);
125
126 % Correlation with output
127 temp = corrcoef(s_y_c(indsim,xt),s_y_c(indsim,xt));
128 r_y(indsim,1) = temp(1,2);
129 temp = corrcoef(s_y_c(indsim,xt),s_c_c(indsim,xt));
130 r_c(indsim,1) = temp(1,2);
131 temp = corrcoef(s_y_c(indsim,xt),s_i_c(indsim,xt));
132 r_i(indsim,1) = temp(1,2);
133 temp = corrcoef(s_y_c(indsim,xt),s_shock_c(indsim,xt));
134 r_s(indsim,1) = temp(1,2);
135 temp = corrcoef(s_y_c(indsim,xt),s_n_c(indsim,xt));
136 r_n(indsim,1) = temp(1,2);
137
138 end
139
140 %
141 % -----
142 % CALCULATE MEAN STATISTICS
143 % -----
144 % Standard Deviations
145 m_std_y = mean(std_y);
146 s_std_y = std(std_y);
147 m_std_c = mean(std_c);
148 s_std_c = std(std_c);
149 m_std_i = mean(std_i);
150 s_std_i = std(std_i);
151 m_std_s = mean(std_s);
152 s_std_s = std(std_s);
153 m_std_n = mean(std_n);
154 s_std_n = std(std_n);
155
156 % Correlations
157 m_r_y = mean(r_y);
158 s_r_y = std(r_y);
159 m_r_c = mean(r_c);
160 s_r_c = std(r_c);
161 m_r_i = mean(r_i);
162 s_r_i = std(r_i);
163 m_r_s = mean(r_s);
164 s_r_s = std(r_s);
165 m_r_n = mean(r_n);
166 s_r_n = std(r_n);
167
168 % Display on screen
169 disp('Standard Deviations of Variables:')
170 disp('y    c    i    n    shock:')
171 disp('=====')
172 disp([m_std_y, m_std_c, m_std_i, m_std_n, m_std_s])
173
174 disp('Correlation of Variables with Output:')
175 disp('y    c    i    n    shock:')
```

```

176 disp('=====')
177 disp([m_r_y, m_r_c, m_r_i, m_r_n, m_r_s])
178
179 save hansen85_simu.mat;

```

代码 6.8.—真实经济周期模型仿真

表 6.1
真实经济周期模型仿真结果

| 变量 | Cooley and Prescott (1995) | | 代码6.6与6.8 | |
|-------|----------------------------|------|-----------|------|
| | 标准差 | 相关系数 | 标准差 | 相关系数 |
| 产出 | 1.35 | 1.00 | 1.54 | 1.00 |
| 消费 | 0.33 | 0.84 | 0.54 | 0.86 |
| 投资 | 5.95 | 0.99 | 4.57 | 0.98 |
| 劳动时间 | 0.77 | 0.99 | 0.80 | 0.96 |
| 劳动生产率 | 0.61 | 0.98 | 0.81 | 0.96 |

[†] 标准差的单位为百分比。

上述代码中，我们在77行和138行之间定义了一个循环体，进行多次仿真。循环体内的语句与代码6.5基本相同。104-114行中对仿真序列进行了HP滤波处理，注意到这里我们调用了用户自定义程序hp2，此处的语法依照读者下载的代码版本不同可能会有所区别。此处作者使用了芝加哥大学教授Harold H. Uhlig编写的版本，基于著作权考虑，此处不在讲义中给出。另外，这里我们对仿真序列取了对数，这主要是为了应用对数函数的性质相对于趋势项的偏离做百分比解释。此处的结果与直接应用原始序列之后用均值标准化是一致的，读者可以自行验证。回忆我们在第四章中的介绍，真实经济周期模型的目的是检验给定生产率冲击的波动，由真实经济周期模型所刻画的机制所导致的宏观经济变量的波动幅度与相关性是否与数据当中观察到的一致。于是如果我们同样用HP滤波对真实数据的对数序列进行处理，那么得到的结果在表6.1中显示。表中左面板为Cooley and Prescott (1995)中计算的结果（表1.2），右面板为执行代码6.6和6.8的结果。我们可以看出，本节中的代码得到了与文中非常相似的结果。之所以没有得到与Cooley and Prescott (1995)中完全相同的结果，主要有两个原因。第一，本节中求解的模型与文中并不完全相同，特别的，我们在这里抽象掉了长期增长，于是模型运行的结果应有细小的差别。第二，由于Cooley and Prescott (1995)文中没有具体提到使用了何种算法对模型进行求解，而任何数值算法都是对模型解的真值的近似，所以不同的算法通常而言会给出不同的近似值。事实上，读者在未来的学习中将会体会到，对于大多数数量分析的文献，在没有作者原始代码的情况下，要精确重现结果都是困难的。这也是为什么，现在主流的经济学期刊要求所有的数量分析研究都必须附带有完整的可重现的代码，供其他研究者重现结果使用。

6.4 校准模型

6.4.1 校准的一般含义

截至目前为止，我们对于模型所有数值算法的讨论都是基于参数已知的情形，那么读者可能要问了，这些模型的参数到底是怎么确定的呢？在数量经济学当中，我们称这种参照数据对模型进行参数化的过程为校准(calibration)。本节我们就来简单介绍一下校准的基本思想，以及如何对本章中的真实

经济周期模型进行校准。在开始讨论之前，我们需要指出的一点是校准是一种非常依赖于具体模型、数据以及所要回答的问题的参数化方法，所以不存在一般的流程化的方法对模型进行校准。尽管学会正确的校准数量模型以及正确的理解文献当中模型校准的思路，是阅读现代宏观经济学前沿文献的关键，但是遗憾的是这是一个需要经过大量文献阅读慢慢积累的过程。本小节中我们只以真实经济周期模型为例，对校准的基本思路进行介绍。至于对模型校准更加深刻的理解，还有赖于读者在未来的学习当中通过文献阅读慢慢积累。而校准的概念之所以在实践中很难把握，其根本原因是校准适用于两种完全不同的场合，而其中一种又与劳动经济学当中常用的结构估计(*structural estimation*)密切相关。我们马上将会看到，校准在实践中通常应用与两种场合。第一，是在已知机制在某些特定的环境下成立的前提下，校准模型到参数使得模型反映了该机制成立的参数环境。之后，应用校准过的模型——也就是该模型所蕴含的科学原理对其他对象的某些特征进行度量。第二种场合，是在不确定某个在特定环境中成立的性质是否能为观察到的现象提供解释时，通过调整模型参数使得模型反映该机制成立的环境，之后应用建筑内过的模型对该机制数量上的解释力进行检验。本节的讨论基于[Cooley and Prescott \(1995\)](#)，如果读者希望更深入的理解校准，也可以参考[Dawkins, Srinivasan and Whalley \(2001\)](#)。

我们先看第一种校准使用的环境，这里预先提醒读者注意的是，真实经济周期模型对校准的应用主要是为第二种目的。首先需要指出的是，校准实际上并不是一个经济学的专有名词，其最早的起源可以追溯到工程上。工程上通常称对于一个度量工具基准值和刻度进行选择以使该工具可以用来对其他对象的特征进行定量测量的过程为校准。考虑一个简单的生活中的例子，即对一支水银温度计依照摄氏温标进行校准。由热力学知识可知，在一个标准大气压下，当温度介于冰水混合物与沸水温度之间时，水银的受热膨胀大致是线性的。于是我们定义一个标准大气压下将冰水混合物与沸水之间的温差一百等分，每一等分即为一摄氏度。于是依次概念，对水银温度计依照摄氏温标进行校准的方式就是首先将水银球置于冰水混合物当中，记录下水银的读数，再将水银球置于沸水当中，记录下另一个读数。之后将两个读数之间的区间一百等分，得到的便是一支依照摄氏温标进行校准过的温度计。随后我们便可以利用这支温度计去度量其他温度在 $0^\circ \sim 100^\circ$ 之间的对象的温度。

如果我们用上面的例子跟我们所熟悉的模型进行一下类比，那么我们发现温度计的例子中，我们所处理的模型是描述水银体积与温度关系的热力学性质，特别的，此时理论模型的性质——也就是水银体积膨胀与温度之间的线性关系是已知的。此时，我们需要校准的参数是摄氏温标温度计的刻度。而我们校准的目的，是应用校准之后的温度计去度量其他对象的温度。对应到经济学的例子当中，如果以我们本节介绍的真实经济周期模型为例，那么此时的情景应该是我们已知以索罗剩余度量的生产率冲击在经济周期中的变化能够解释宏观经济变量在经济周期频率上的波动与联动（水银的体积线性膨胀），我们的目标是应用这个模型来衡量比如反周期的财政政策对于宏观经济运行平稳程度的效果（度量其他对象的温度），那么我们校准模型的策略就是调整模型的参数，使得模型所蕴含的经济周期频率上变量的波动与联动与数据当中观察到的相一致（摄氏温标）。

另一方面，校准在我们当前真实经济周期模型中的应用，则是为了另外的目标。真实经济周期模型的目标是检验在数量上，由索罗剩余刻画的生产率冲击是否能够在一个符合美国宏观经济长期增长特征的模型中引起与数据相吻合的宏观经济变量的波动与联动。注意这个情形与第一种情形间显著的不同是我们只知道某个机制可能起作用，但是不确定在定量上的效果到底如何。类比到温度计的例子里，我们知道在一个标准大气压下，当温度在 $0^\circ \sim 100^\circ$ 之间时，水银体积随着温度升高会膨胀，但是膨胀的程度是否为线性，我们并不清楚。为了检验这一点（注意到这是热力学定律的一个预测），此时我们需要对水银温度计进行的校准是确定气压为1atm，以及确定温度计水银柱在 0° 和 100° 时候的读数。随后，我们需要检验的是，随着温度的升高，水银柱的读数上升是否是线性的。对应到经济周期模型当中，我们对模型的校准此时要保证在不存在生产率冲击的时候，模型的性质与美国宏观经济长期增长的Kaldor Facts吻合，此处相当于保证室内气压为1atm以及确定水银柱在 0° 和 100° 时候的读数。随后

我们在模型当中引入符合数据的统计特征的生产率冲击（令温度升高），求解并对模型进行仿真以检验模型所刻画的宏观经济变量的波动与联动是否与数据当中观察到的相符合（水银体积的膨胀是否为线性）。

这里我们简单提一下另一种在劳动经济学与实证产业组织理论当中非常常用的数量分析方法，即结构估计。与校准类似，结构估计通常也用于上述两种用途。对应了校准第一类用途的结构估计在方法论上与校准并没有太大的不同，只不过在应用的场合上略有不同。直观上，对模型进行校准的参数通常只得到一个参数值，而没有读者所熟悉的统计学中通常伴随统计分析存在的标准误。这是由于对模型进行校准通常是针对数据中某个变量的确定值进行的，比如美国战后经济的平均增长率。于是此时在数学上我们实际上只是在求一个特定方程或方程组的解。不过，当我们需要模型的统计性质与数据中的某些统计指标相吻合的时候，由于可以证明对同一个模型进行多次仿真并计算统计矩得到的结果本身是服从大数定律和中心极限定理的，所以在多次仿真的意义上我们可以得到此时模型参数的渐进分布[Pakes and Pollard (1989), Duffie and Singleton (1993)]。对应到此时的真实经济周期模型中，如果我们使用校准的方法，那么我们将会选择生产率冲击的随机过程参数 ρ 和 σ_ε ，使得模型多次仿真得到的统计矩的样本间均值与数据中观察到的相等，这能够给我们 ρ 和 σ_ε 确定的值。而如果我们应用结构估计进行处理，那么此时我们还能够得到 ρ 和 σ_ε 估计量的标准误。读者在此可能会产生一个印象，那就是乍看之下似乎结构估计应该是优于校准的一种方法，因为似乎两者唯一的区别似乎就是结构估计中我们还能够得到估计量的标准差，如果不考虑它可能用起来会更麻烦这个成本的话。这个直觉部分上是正确的。之所以说是部分上，是因为如果我们处理的是如同增长理论中的确定性模型，那么基于蒙特卡洛仿真的估计量的渐进分布此时就无法构造，所以在此类场合中校准是唯一的选择。另外，由于应用在此类目的时，通常校准或估计过的模型会被用来进行关于一些政策效果的反事实仿真，此时模型参数的统计性质将会如何传导到新的仿真变量的统计矩上，一般而言是不确定的。对应到经济周期模型的例子中，就是哪怕假设我们知道贴现率 β 服从一个参数已知的正态分布，一般而言投资序列 I_t 的统计性质如何随 β 的统计性质变化也是不清楚的，读者可以想象当存在多个参数的时候这个问题处理的难度会进一步的上升。所以如果仅仅是为了进行反事实的政策仿真的话，结构估计所额外带来的模型参数的标准差在应用中的意义并不大。事实上这也是在早期方法论辩论中，极大似然估计最终让位于校准和基于仿真的结构估计的原因——一般均衡模型中内生变量的统计性质非常难于构造。^①

结构估计实际上更常用于校准的第二类目的当中，此时两者之间的区别更加微妙。以经济周期的例子来说，校准的逻辑是当生产率冲击与数据相一致时，检验宏观经济仿真变量的统计性质是否与数据相吻合。结构估计的逻辑则是硬币的反面，即若要模型仿真变量的统计性质与数据相吻合，那么生产率冲击应该符合怎样的分布？从这个对比可以看出，尽管两种方法实际上应用了同样的数据，但是他们背后所回答的经济学问题、或者说研究者的先验假设是完全不同的。如果说校准的目的，从方法论的角度出发是为了检验生产率冲击这个因素是否能为经济周期提供一个解释，那么结构估计更适合应用于当我们确定的知道生产率冲击作为一个不可观测的变量是解释经济周期现象的核心时，以一个结构模型作为辅助的度量工具，来对生产率冲击本身进行度量的过程。简单来说，前者的目的在于检验一个机制，而后者的目的则是估计一个变量。不过，在实践中，结构估计也常被灵活应用于估计出某个变量的值或分布之后，根据已有的数据判断如果模型刻画的机制是正确的，那么所需要的核心变量的统计性质是否与已观测到的水平相吻合的情形。结构估计应用于此类场合时，通常核心变量的值是可以观测的。事实上，当这个变量估计出来的值与观测到的事实相吻合时，结构估计和校准在方法论上是一回事儿。

^① 这里我们不展开解释结构估计，对基于仿真的结构估计有兴趣的读者可以参考Gouriéroux and Monfort (1996)和Stern (1997)，而对于早期方法论辩论有兴趣的同学请见Hansen and Heckman (1996)。

正是由于校准和结构估计之间存在非常紧密的联系，其应用场合具有相当的灵活性，所以这两种方法在实践中极易被误用。另一方面，由于文献中通常不会明确的表明对于该文章而言，对模型校准或结构估计主要是为了何种目的，所以很多读者在刚开始阅读数量分析文献的时候经常不得要领，无法理解作者行文的目的与逻辑。所以这里提醒读者，读者在刚开始接触前沿文献的时候，请务必弄清作者校准模型的目的，这通常是与作者所要回答的科学问题直接相关的。事实上，如果读者感觉无法弄清文章作者对模型进行校准的目的，通常读者可能也并没有搞清楚整篇文章的逻辑体系，即想要回答什么样的问题，做了何种假设，理想情况下应该得到何种结果才能对提出的问题进行证实或证伪。不过正如在本小节开头所说，这是一个非常依赖于经验积累的过程，所以读者在上手阶段务必耐心。

6.4.2 校准真实经济周期模型

下面我们来看如何对真实经济周期模型进行校准。首先让我们明确，真实经济周期模型的目的是检验对于一个符合美国经济增长特征事实的模型而言，在其中引入以索罗剩余来度量的生产率冲击，能否产生于数据相符合的经济周期波动。所以这里我们对模型的校准，主要的目标有两个。一是使得模型的参数使得模型在长期上的含义（这里就是稳态）与数据相吻合，二是刻画生产率冲击的随机过程要与数据中观察到的索罗剩余的特征相一致。

一般而言，校准主要分为三个步骤。第一是确定模型中各种函数的函数形式，比如CRRA的效用函数或者柯布-道格拉斯的生产函数等。第二是从数据中构造与模型变量相对应的度量。比如在我们当前的模型中不存在政府部门，那么我们就需要思考数据当中政府部门进行的投资、采购、公共储蓄等需要进行如何的处理，才能使得模型与数据之间可以相对应。第三，从数据中构造了与模型相一致的度量之后，我们就可以选择模型的参数使得模型在这些维度上的特征与数据相符合。下面我们分步骤来对本节的真实经济周期模型进行校准。

回忆图3.4的顶面板，我们知道长期而言，资本与劳动份额在国民收入中所占的比重是相对固定的，于是生产函数我们采用柯布-道格拉斯形式[Cobb and Douglas (1928)]。^①另一方面，图3.4的底面板中劳动时间占总时间禀赋的比例大体是稳定的，于是我们选择对数线性的形式[King, Plosser and Rebelo (1988)]。而利用柯布-道格拉斯生产函数，从数据当中计算出来的索罗剩余的对数表现出了非常明显的持久性，于是我们设生产率冲击服从一个一阶自回归过程。这就完成了对于模型中函数形式的刻画。

由于在新古典模型中，整个系统动态的核心机制是消费储蓄的跨期替代，所以为了使模型的结论能够与实际数据相对比，我们需要确保模型中的资本与数据中的“资本”描述的是同一个概念。注意到，概念上我们的模型是高度抽象的。比如，模型当中不包括政府部门、没有家庭生产部分、我们只考虑了一个封闭经济没有对外贸易，同时我们也没有显式的对存货进行处理。上面这些概念于是在模型当中都被抽象到资本存量 K 当中了。所以相应的， Y 也需要包含由这部分资本带来的产出。这些数据的处理通常是非常繁琐的，本教程当中我们不逐步讲解，希望了解细节的读者请参考Cooley and Prescott (1995)或Prescott (1986)。现在我们假设我们已经对数据进行过相应的处理，得到了与模型概念相吻合的度量，现在我们来看模型的参数应该如何进行校准。

我们一共需要校准的参数有六个，分别是贴现率 β ，消费在决定效用时所占的份额 σ ，资本折旧率 δ ，国民收入中资本所占的份额 α ，生产率冲击一阶自回归过程的自回归系数 ρ 与随机扰动的标准差 σ_ϵ 。首先，我们在数据中可以计算出国民收入中资本的比例，注意由上一自然段的讨论，此处的数据包含了政府投资等我们在模型中抽象掉的部分。于是对照数据，可以直接得到 $\alpha = 0.40$ 。随后，

^① 不过最近有一股新的文献在讨论劳动份额近些年持续下降的原因，有兴趣的读者可以参考Boppart (2014), Karabarounis and Neiman (2014)。

应用 $\alpha = 0.40$, 我们可以从数据当中计算索罗剩余。计算出来的索罗剩余可以被用来估计 ρ 和 σ_ε , 这给我们 $\rho = 0.95$ 和 $\sigma_\varepsilon = 0.00712$ 。注意到资本的动态方程在稳态当中意味着

$$\delta k = i.$$

NIPA数据中计算出年度投资占总资本的比率 $k/y = 10\%$, 这意味着模型中的年度折旧率为10%。由于模型每一期对应了现实经济中的一个季度, 所以我们设 $\delta = 0.025$ 。在稳态中, 跨期欧拉方程成为

$$\frac{1}{\beta} = \alpha \cdot \frac{y}{k} + 1 - \delta,$$

那么由数据中得到的资本产出比 $k/y = 3.32$, 带入年度折旧率 $\delta = 0.10$, 我们得到年度贴现率等于0.9128, 这意味着季度贴现率 $\beta = 0.977$ 。最后, 稳态中消费闲暇之间替代的期内欧拉方程为

$$(1 - \alpha) \cdot \frac{y}{c} = \frac{1 - \sigma}{\sigma} \cdot \frac{1 - l}{l},$$

那么由微观计量经济学估计出来的平均劳动时间占总时间的 $1/3$ 和稳态中产出消费比为1.33可以得出 $(1 - \sigma)/\sigma = 1.78$ 。于是我们就完成了所有模型参数的校准。

注意到我们校准出的参数与Cooley and Prescott (1995)中并不完全一样, 原因正如上一节所述, 我们进一步的抽象掉了长期人口增长和技术进步。注意到虽然这会改变模型的参数, 但是我们在上一节中看到了, 模型数值仿真得到的统计性质在定量上与Cooley and Prescott (1995)中的区别并不大。这告诉我们模型参数的校准通常情况下是与模型结构直接相关的, 但是由于很多情况下参数的改变实际上只相当于参照系的改变, 所以模型的定量性质未必会因此发生变化。另外, 注意到尽管我们的模型是季度的, 但是我们在通过稳态校准的时候是先假设稳态为年度, 之后再推算出相关参数在季度中所应有的水平。这主要是由于NIPA数据中很多变量是年度的。比如在校准折旧率的时候, 我们在数据中只能观察到年度投资对资本的比例, 所以校准出的 δ 反映的是年度的信息。如果我们此时有季度水平上投资对资本的比例, 此处我们可以直接校准季度所对应的折旧率 δ 。

第七章 扰动法求解动态随机一般均衡模型

在上一章当中我们应用动态优化的方法对第四章中构造的真实经济周期模型进行了求解，并且发现在定量上，一个经过良好校准的模型所蕴含的宏观经济变量的波动与联动能够在一定程度上与数据当中观察到的想符合。不过在对模型的求解当中我们也发现值函数迭代法用起来至少有四个方面不是很方便。第一，模型求解的过程非常耗时，反复调试的试错时间成本很高。第二，我们在第五章当中看到，当 ρ 接近1也就是生产率冲击近似随机游走的时候，马尔科夫链近似的性质并不是很好。同时注意到，我们在第五章当中判断近似效果的好坏依据的是由马尔科夫链刻画的随机过程的统计性质与原始一阶自回归过程的统计性质是否相似。但是我们在第六章求解模型的时候，实际需要的是积分 $EV(k'; \theta')$ 的值与真值比较接近，马氏链的良好统计性质能否直接对应到相应的数值积分的良好精度至少直观来看并不是那么直接。从这个意义出发，我们希望有另一种算法，能够保留生产率冲击的原始结构，以另一种方法对模型的解进行近似，从而与我们通过值函数迭代得到的解进行对照。第三，值函数迭代法实际上提供了一个一般的分析方法，但是具体到不同的经济学问题，程序上具体的实现细节可以相差很大。于是我们希望知道在宏观经济学领域中，是否有类似于计量经济学当中的回归方法一样，能够以非常标准的步骤应用软件包对不同结构的模型进行求解。第四，尽管函数的数值表达也能够很好的反映政策函数的特征，但是解析解在分析模型性质的时候，无疑还是能够以更为清晰的方式传递信息。

本章我们介绍一种求解动态随机一般均衡(*Dynamic Stochastic General Equilibrium, DSGE*)模型的常用方法——扰动法(*Perturbation Method*)。简单的说，扰动法就是应用函数的泰勒展开对模型在稳态附近进行局部近似。相比于值函数迭代给出了较高精度的全局解，扰动法的精度通常只在稳态周围较高，当真实政策函数是非线性的时候，离开稳态越远，扰动法得到的政策函数精度就越低，这是应用扰动法的成本。但是扰动法相对于值函数迭代，在上一段中提到的四个方面能够提供一定的改善。具体来说，扰动法的求解非常迅速，同时有成熟的DYNARE软件包支持，程序编写远远比值函数迭代来的容易。另外，模型中的不确定性冲击（比如生产率冲击）可以直接按照原始随机过程的形式带入，这使得我们可以容易的分析更加复杂的冲击结构，而后者要使用马氏链进行近似就要复杂的多。最后，扰动法得到的政策函数为控制变量关于状态变量的多项式函数的解析表达式，从中我们可以比较清晰的看出状态变量对控制变量进行影响的渠道，从而使模型的经济机制更加清晰。

本章的主要内容如下。我们首先以一个简单的增长模型为例，介绍用线性化求解动态模型的方法。随后我们将会看到，线性化（或对数线性化）是扰动法的一个特例。最后我们介绍如何使用DYNARE软件包求解上一章中的真实经济周期模型。本章的内容主要参考自Judd (1998)的第十三章，Schmitt-Grohé and Uribe (2004)，以及DYNARE的官方用户指南。

7.1 扰动法的一般理论

回忆我们在第三章当中求解的增长模型：

$$\max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

s.t.

$$c_t + k_{t+1} = (1 - \delta)k_t + f(k_t), \quad (7.1)$$

其一阶条件由下述欧拉方程和资源约束给出

$$u'(c_t) = \beta u'(c_{t+1})[f'(k_{t+1}) + 1 - \delta], \quad (7.2)$$

$$c_t + k_{t+1} - (1 - \delta)k_t = f(k_t). \quad (7.3)$$

我们知道一种求解优化问题(7.1)的方法是将其转换为一个动态优化问题，之后求解政策函数。此时，我们所谓的模型的解，实际上是控制变量 c_t 和 k_{t+1} 关于状态变量 k_t 的函数，而值函数迭代法便是求解这样一组函数关系的算法中的一种。仔细看方程(7.2)和(7.3)，若我们记向量 $\mathbf{z}_t = [c_t, k_t]'$ ，那么我们发现两者实际上定义了一个 \mathbf{z}_{t+1} 关于 \mathbf{z}_t 的非线性差分方程组。一般而言，非线性差分方程组的分析是困难的，但是对于线性差分方程组有很成熟的理论用比较一般的理论对其进行分析。我们首先对一阶线性差分方程组的基本理论进行简单的介绍。

7.1.1 一阶线性差分方程组及其稳定性理论

不妨先从单变量的情形开始考虑，设差分方程为

$$x_{t+1} + ax_t = c, \quad (7.4)$$

其中 a, c 为参数。回忆在差分方程或常微分方程当中学到的知识，我们知道方程(7.4)是一个一阶非齐次(*non-homogeneous*)线性差分方程，其相对应的齐次(*homogeneous*)方程为

$$x_{t+1} + ax_t = 0. \quad (7.5)$$

方程(7.5)的解称为所有对应了该齐次方程的非齐次方程的通解(*general solution*)。由差分方程相关理论[Elaydi (2005)]，一个非齐次方程的解等于齐次方程的通解加上另一个使非齐次方程成立的特解(*specific solution*)。若记 $x_{t+1} = x_t = x^{ss}$ ，那么容易验证

$$x^{ss} = \frac{c}{1 + a},$$

是满足非齐次方程(7.4)的一个特解。注意到此处我们求解稳态处的特解的原因是对确定性的增长模型，由第三章的知识知道长期中模型会收敛到稳态。另一方面，对齐次方程(7.5)我们可以验证对于任给的常数 A ，

$$x_t = A(-a)^t$$

是它的通解。^① 注意到为了确定常数 A 的值，我们需要一个初值条件或边值条件，不妨令初值 x_0 已知。那么由于非齐次方程的解的一般形式为

$$x_t = \frac{c}{1 + a} + A(-a)^t,$$

^① 这里我们不具体解释上述通解的求解过程。熟悉微分或差分方程理论的同学应该能回忆起 $-a$ 其实就是方程(7.4)对应的特征多项式的根，这种求解微分或差分方程的方法又称特征线(*Characteristic Equation*)法。详见Hirsch, Smale and Devaney (2004) 或 Elaydi (2005)。

带入 x_0 , 可以得到

$$A = x_0 - \frac{c}{1-a}.$$

于是我们就得到了非齐次方程(7.4)的解为

$$x_t = \frac{c}{1+a} + \left(x_0 - \frac{c}{1+a}\right)(-a)^t. \quad (7.6)$$

注意到此时方程(7.6)在极限处的性质取决于 $|a|$ 的值。当 $|a| < 1$ 时, $t \rightarrow \infty$ 时系统收敛到稳态 x^{ss} ; 反之, 若 $|a| > 1$, 则 $\lim_{t \rightarrow \infty} x_t$ 是发散的。

下面我们将上面的理论推广到多个变量的一阶线性差分方程组的情形, 我们以两个变量为例, 但是分析的逻辑可以很容易推广到多个变量的情形。具体的, 设此时 $\mathbf{X}_t = [x_{1t}, x_{2t}]'$ 和 \mathbf{C} 是 2×1 的列向量, \mathbf{A} 为 2×2 维矩阵, 那么此时差分方程组为

$$\mathbf{X}_{t+1} + \mathbf{AX}_t = \mathbf{C}. \quad (7.7)$$

非齐次一阶线性差分方程组(7.7)同样对应了一个齐次方程组

$$\mathbf{X}_{t+1} + \mathbf{AX}_t = \mathbf{0}. \quad (7.8)$$

非齐次方程(7.7)的求解与单变量的情形类似, 我们也是先求齐次方程组(7.8)所对应的通解, 之后再与特解相加。设 $\mathbf{X}_{t+1} = \mathbf{X}_t = \mathbf{X}^{ss}$, 带入(7.7)式得到特解

$$\mathbf{X}^{ss} = (\mathbf{I} + \mathbf{A})^{-1} \mathbf{C},$$

其中 \mathbf{I} 是单位矩阵。注意到上式与标量情形的相似性。下面我们求通解, 矩阵的情形与标量情形相比要稍微复杂一些。首先设矩阵 \mathbf{A} 的若尔当(Jordan)分解为

$$\mathbf{A} = \mathbf{VDV}^{-1},$$

其中 \mathbf{V} 是由特征向量组成的矩阵, \mathbf{D} 是由特征值构成的对角阵。那么若定义 $\mathbf{Z}_t = \mathbf{V}^{-1} \mathbf{X}_t$, (7.8)式此时成为

$$\mathbf{Z}_{t+1} + \mathbf{DZ}_t = \mathbf{0}. \quad (7.9)$$

记 $\mathbf{V} = \text{diag}(\lambda_1, \lambda_2)$, 那么可以验证对于任意常数 m_1, m_2 , 齐次方程(7.9)的通解为

$$\begin{aligned} z_{1t} &= m_1(-\lambda_1)^t \\ z_{2t} &= m_2(-\lambda_2)^t, \end{aligned}$$

其中 z_{1t} 和 z_{2t} 为 \mathbf{Z}_t 的两个元素。若进一步令特征向量矩阵

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix},$$

那么带回 \mathbf{Z}_t 的定义, 我们有原齐次方程(7.8)的通解

$$\mathbf{X}_t^G = \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} = \begin{bmatrix} v_{11}m_1(-\lambda_1)^t + v_{12}m_2(-\lambda_2)^t \\ v_{21}m_1(-\lambda_1)^t + v_{22}m_2(-\lambda_2)^t \end{bmatrix}. \quad (7.10)$$

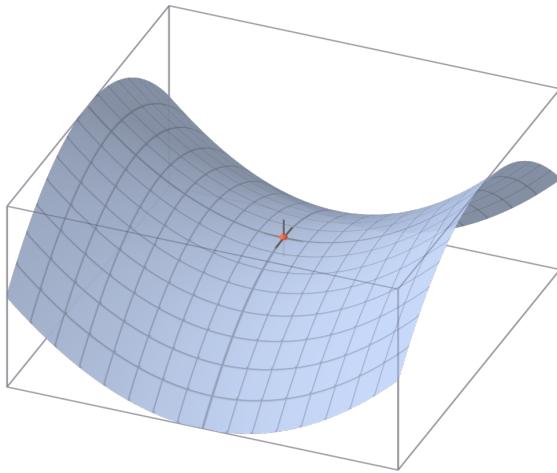


图 7.1.—鞍点路径

此时我们有两个自由参数 m_1, m_2 需要确定，于是我们需要两个条件。在单变量的情形下，自由参数 A 是由一个初值条件给出的，所以如果我们有初值 \mathbf{X}_0 ，那么将该条件带入方程(7.7)的解

$$\mathbf{X}_t = \mathbf{X}^{ss} + \mathbf{X}_t^G,$$

我们就得到了一个关于 m_1, m_2 的二元一次方程组。不过回忆本节开始时候的例子， \mathbf{X}_t 的两个元素对应了 c_t 和 k_t 。通常我们只有 k_0 的初值，而没有 c_0 的初值，所以差分方程理论中通常能够确定解的唯一性的初值条件在我们通常处理的问题当中只能成立一半，所以我们还需要另外一个条件才能把 m_1, m_2 唯一的确定下来。我们在单变量情形下提到过，整个系统的性质取决于特征根 a 的绝对值。在多变量情形下我们可以从(7.10)中看出，整个系统的稳定性受两个特征根 λ_1, λ_2 共同影响。具体来说，对应到二维的情形，我们有以下三种情形：

- 两个特征根的模都大于1。此时除非我们有 $m_1 = m_2 = 0$ ，不然整个系统在极限处是发散的。同时由于当 $m_1 = m_2 = 0$ 时，事实上我们有 $\mathbf{X}_t^G = 0$ ，所以很多时候这会与初值条件相矛盾。此时，我们称整个线性系统是发散的(*explosive*)。
- 两个特征根的模都小于1。此时无论 m_1 和 m_2 取何值，整个系统在极限处都收敛到特解，也就是稳态。此时除非有另一个初值或者边值条件，不然 m_1 和 m_2 是不确定的。此时我们称线性系统是不定的(*indeterminate*)。
- 一个特征根的模大于1，另一个小于1。不妨设 $|\lambda_1| < 1$ 而 $|\lambda_2| > 1$ 。此时若要整个系统在极限处收敛到稳态，需要 $m_2 = 0$ 成立。此时，一个初值条件就可以唯一的确定参数 m_1 ，从而唯一的确定整个系统的性质。由于此时整个系统只有在沿着 λ_1 所定义的方向才是稳定的，而任何 λ_2 方向上的偏移都会导致整个系统发散，所以此时我们称线性系统沿鞍点路径(*saddle path*)稳定。对应到图7.1中，直观上就是说整个系统必须严格按照水平方向运行，任何纵深方向的偏移（即 $m_2 \neq 0$ ）都会使得系统发散，即图像上滑落到马鞍外面去。

经验上来说，在宏观经济学当中我们遇到的绝大多数情形都是鞍点路径均衡。注意到，我们需要鞍点路径条件的原因是我们只有一个初值条件，从而需要另一个条件来确定另一个自由参数的值。对应到

增长模型里，这里的初值条件是初始资本 k_0 。我们在动态优化问题当中知道， k 是作为状态变量存在的，于是资本并不只是在第零期可以作为初值使用，对应到任何的 t ， k_t 都是已知的。所以这提示我们无论我们考虑哪一期，总是有与状态变量数目相同的初值条件可以用来确定系统的参数，如果我们有与余下的参数数量相等的大于1的特征值，我们就可以应用鞍点路径的条件刻画线性系统。写成数学记号，如果我们有 n 个变量组成的线性系统，那么我们有 n 个特征值以及 n 个自由参数。设内生状态变量数量为 m ，那么任给时期 t ，我们都有 m 个初值条件。于是当模大于1的特征值数量为 $n - m$ 时，我们可以求解线性系统沿鞍点路径的解。这个条件在文献当中又称为Blanchard-Kahn条件[[Blanchard and Kahn \(1980\)](#)]。另外这里需要特别提醒读者注意的是，从解

$$\mathbf{X}_t = \mathbf{X}^{ss} + \mathbf{X}_t^G$$

出发反向构造出的差分方程

$$\mathbf{X}_{t+1} = \mathbf{F}_1 \mathbf{X}_t + \mathbf{F}_0$$

中一般来说 $\mathbf{F}_1 \neq \mathbf{A}$ ，且 $\mathbf{F}_0 \neq \mathbf{C}$ 。这是因为 $\mathbf{F}_0, \mathbf{F}_1$ 中带入了鞍点路径均衡所需要的唯一解的条件。

7.1.2 线性化与对数线性化

有了上一小节的线性差分方程组一般性质的讨论，我们考虑如果能够用一个线性差分方程组对由(7.2)和(7.3)刻画的非线性差分方程组近似，那么我们就可以用上一小节中的一般理论对这个线性方程组进行求解，从而得到模型的一个具有解析表达式的近似解。^①数学分析当中我们常用函数在某一个点邻域附近的泰勒展开来对函数进行近似，而对应了(7.2)和(7.3)的稳态可以很容易的求解，于是我们便可以通过将(7.2)和(7.3)在稳态周围进行一阶泰勒展开来构造非线性系统的线性近似。若定义集合 \mathcal{C}^n 为全部 n 阶光滑——即 n 阶导数存在且连续——的函数所组成的集合，回忆数学分析中的知识，我们有如下的关于单变量和多变量函数的泰勒中值定理。

定理7.1. (单变量函数泰勒中值定理) 若 $f \in \mathcal{C}^{n+1}[a, b]$ 且 $x, x_0 \in [a, b]$ ，那么

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \\ &\quad + \cdots + \frac{1}{n!}f^{(n)}(x_0)(x - x_0)^n + R_{n+1}(x), \end{aligned}$$

其中对某个处于 x 和 x_0 之间的 ξ ，拉格朗日余项

$$R_{n+1}(x) = \frac{1}{n!} \int_{x_0}^x (x-t)^n f^{(n+1)}(t) dt = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-x_0)^{n+1}.$$

定理7.2. (多变量函数泰勒中值定理) 若函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 是 \mathcal{C}^{k+1} 的，那么对于 $\mathbf{x}, \mathbf{x}^0 \in \mathbb{R}^n$ ，

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}^0) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\mathbf{x}^0)(x_i - x_i^0) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}^0)(x_i - x_i^0)(x_j - x_j^0) \\ &\quad + \cdots + \frac{1}{k!} \sum_{i_1=1}^n \cdots \sum_{i_k=1}^n \frac{\partial^k f}{\partial x_{i_1} \cdots \partial x_{i_k}}(\mathbf{x}^0)(x_{i_1} - x_{i_1}^0) \cdots (x_{i_k} - x_{i_k}^0) \\ &\quad + \mathcal{O}(\|\mathbf{x} - \mathbf{x}^0\|^{k+1}). \end{aligned}$$

为简化记号，我们将余项 $\mathcal{O}(\|\mathbf{x} - \mathbf{x}^0\|^{k+1})$ 写成佩亚诺形式。

^① 英文文献中通常称此时模型具有closed-form solution。

由于泰勒展开通常只在展开点周围有较好的精度，所以由函数泰勒展开构造的线性动态系统也只有在波动范围在稳态周围时才有较好的精度，这里提请读者注意。

下面我们将(7.2)和(7.3)进行线性化。首先记稳态处的消费和资本分别为 \bar{c} 和 \bar{k} 。首先来看资源约束(7.3)。由于方程左边本身就是线性的，所以对其应用泰勒展开是平凡的，不过这里为了处理的一致性，也将其写成“泰勒展开”的形式：

$$\bar{c} + (c_t - \bar{c}) + \bar{k} + (k_{t+1} - \bar{k}) - (1 - \delta)[\bar{k} + (k_t - \bar{k})].$$

注意到上式是对一个多元函数进行泰勒展开，而不是将(7.3)中的每一项分别线性展开。(7.3)式的右侧的一阶泰勒展开是

$$f(\bar{k}) + f'(\bar{k})(k_t - \bar{k}).$$

两式相等，注意到稳态中我们有

$$\bar{c} + \bar{k} - (1 - \delta)\bar{k} = f(\bar{k}),$$

简单的代数运算得到(7.3)的线性展开式

$$c_t - \bar{c} + k_{t+1} - \bar{k} - (1 - \delta)(k_t - \bar{k}) = f'(\bar{k})(k_t - \bar{k}). \quad (7.11)$$

对欧拉方程(7.2)如法炮制，我们得到方程的左侧为

$$u'(\bar{c}) + u''(\bar{c})(c_t - \bar{c}),$$

而方程的右侧成为

$$\beta \left\{ \underbrace{u'(\bar{c})[1 - \delta + f'(\bar{k})]}_{\text{常数项}} + \underbrace{u''(\bar{c})[1 - \delta + f'(\bar{k})](c_{t+1} - \bar{c})}_{c_{t+1} \text{的一次项}} + \underbrace{u'(\bar{c})f''(\bar{k})(k_{t+1} - \bar{k})}_{k_{t+1} \text{的一次项}} \right\}.$$

再带入稳态中

$$u'(\bar{c}) = \beta u'(\bar{c})[1 - \delta + f'(\bar{k})],$$

我们得到(7.2)的线性展开式

$$u''(\bar{c})(c_t - \bar{c}) = u''(\bar{c})(c_{t+1} - \bar{c}) + \beta u'(\bar{c})f''(\bar{k})(k_{t+1} - \bar{k}), \quad (7.12)$$

其中等号右侧第一项前的系数我们应用稳态当中 $\beta[1 - \delta + f'(\bar{k})] = 1$ 进行了化简。于是方程(7.11)和(7.12)就构成了刻画优化问题(7.1)解的线性差分方程组。注意到这是一个非齐次的方程组，可以用上一小节的知识就行求解。

注意到，方程(7.11)和(7.12)刻画的是 c_t 和 k_t 的具体值，而如我们在前几章的经济周期模型中看到的，很多时候我们对某个变量的绝对值并不太感兴趣，我们想要研究的是变量相对于稳态值偏离的百分比。手工计算 c_t 和 k_t 偏移的百分比当然是一种方法，但是更加常用的方法是对(7.2)和(7.3)进行对数线性化，其得到的线性系统刻画的此时就是相对于稳态均值偏离的百分比。下面我们就来看如何对(7.2)和(7.3)进行对数线性化。

回忆数学分析当中学到的知识，关于对数函数我们有极限

$$\lim_{x \rightarrow 0} \ln(1 + x) = x,$$

若定义变量 x_t 相对于稳态偏移的百分比为

$$\hat{x}_t = \frac{x_t - \bar{x}}{\bar{x}},$$

那么当 x_t 在 x^* 周围小幅度波动即 $\hat{x}_t \approx 0$ 时，我们有

$$\ln(1 + \hat{x}_t) = \ln\left(\frac{x_t}{\bar{x}}\right) = \hat{x}_t.$$

我们的目标是将(7.2)和(7.3)中的变量全部替换为 \hat{x}_t ，由上式知道这等同于先将全部变量关于稳态均值做商，之后再取对数，对数线性化的名称便由此而来。我们还是先看资源约束(7.3)，在其中带入

$$x_t = \bar{x}(1 + \hat{x}_t),$$

之后将方程关于 \hat{x}_t 进行一阶泰勒展开，注意到此时稳态中 $\hat{x}_t = 0$ ，方程左侧于是成为

$$\bar{c}(1 + \hat{c}_t) + \bar{k}(1 + \hat{k}_{t+1}) - (1 - \delta)\bar{k}(1 + \hat{k}_t).$$

类似的，方程的右侧成为

$$f[\bar{k}(1 + \hat{k}_t)] \approx f(\bar{k}) + f'(\bar{k})\bar{k}\hat{k}_t.$$

将左右两侧联立，再应用稳态条件消去常数项，我们得到(7.3)的对数线性展开

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_{t+1} - (1 - \delta)\bar{k}\hat{k}_t = f'(\bar{k})\bar{k}\hat{k}_t. \quad (7.13)$$

同理对于欧拉方程(7.2)，我们得到方程左侧

$$u'(\bar{c}) + u''(\bar{c})\bar{c}\hat{c}_t,$$

以及方程右侧

$$\beta \left\{ u'(\bar{c})[1 - \delta + f'(\bar{k})] + u''(\bar{c})[1 - \delta + f'(\bar{k})]\bar{c}\hat{c}_{t+1} + u'(\bar{c})f''(\bar{k})\bar{k}\hat{k}_{t+1} \right\}.$$

再次将两者联立，消去稳态条件，我们得到对数线性化之后的欧拉方程

$$u''(\bar{c})\bar{c}\hat{c}_t = u''(\bar{c})\bar{c}\hat{c}_{t+1} + \beta u'(\bar{c})f''(\bar{k})\bar{k}\hat{k}_{t+1}. \quad (7.14)$$

容易看出，由方程(7.13)和(7.14)组成的线性系统是齐次差分方程组，同样可以应用上一小节中的方法求解。

7.1.3 扰动法的一般理论

前述的例子实际上可以用如下的结构来表达。设 x 是变量， c 是参数，我们已知 x 和 c 之间有函数形式已知的关系

$$f(x, c) = 0, \quad (7.15)$$

读者可以将 x 理解为上面例子中的资本与消费，将 c 理解为资本与消费分别相对于各自稳态均值的偏离。我们要求解的实际上是 x 关于 c 的一个函数 $x = x(c)$ 。由数学分析当中的知识，我们知道此处需要应用隐函数定理，作为复习，我们这里给出多变量情形下的隐函数定理。

定理7.3. (隐函数定理) 如果函数 $H(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ 是 C^k 的, $H(\mathbf{x}_0, \mathbf{y}_0) = \mathbf{0}$, 且雅各比矩阵 $H_{\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0)$ 非奇异 (即行列式不为零), 那么存在唯一的 C^0 函数 $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 使得 $\mathbf{y}_0 = h(\mathbf{x}_0)$ 且在 \mathbf{x}_0 的邻域内有 $H(\mathbf{x}, h(\mathbf{x})) = \mathbf{0}$ 。进一步的, 若 H 是 C^k 的, 那么 h 也是 C^k 的, 且 h 的导数可以通过对 $H(\mathbf{x}, h(\mathbf{x})) = \mathbf{0}$ 求隐式微分得到。

回忆在上一节中的例子中, 我们能够求解 x 在稳态处的值 $x(0)$ 。为简化记号, 我们以 x 和 c 均为标量的情形为例。此时, 对(7.15)做隐式微分, 我们得到

$$f_x(x(c), c)x'(c) + f_c(x(c), c) = 0. \quad (7.16)$$

其中 f_x 和 f_c 分别为 f 关于变元 x 和 c 的偏导数。由于 f 的函数形式已知, 所以当 $x(0)$ 已知的时候, 在 $(x(0), 0)$ 处的偏导数 $f_x(x(0), 0)$ 和 $f_c(x(0), 0)$ 的值都是可以求解, 于是由隐函数定理

$$x'(0) = -\frac{f_c(x(0), 0)}{f_x(x(0), 0)}$$

当 $f_x(x(0), 0) \neq 0$ 时可以求出。于是, 我们就可以构造 $x(c)$ 的线性近似

$$x(c) \doteq x^L(c) \equiv x(0) - \frac{f_c(x(0), 0)}{f_x(x(0), 0)}c.$$

我们还可以进一步的构造二次近似, 具体来说, 若我们继续对(7.16)求隐式微分, 可以得到

$$f_x x'' + f_{xx}(x')^2 + 2f_{xc}x' + f_{cc} = 0,$$

注意到当 $c = 0$ 时, 上式可以整理成

$$x''(0) = -\frac{f_{xx}(x(0), 0)(x'(0))^2 + 2f_{xc}(x(0), 0)x'(0) + f_{cc}(x(0), 0)}{f_x(x(0), 0)}. \quad (7.17)$$

由于 $x'(0)$ 在线性近似的时候已经求出, 而其他偏导数由于 f 函数形式已知都可以求解, 所以(7.17)可以用来构造 $x(c)$ 的二次近似

$$x(c) \doteq x^Q(c) \equiv x(0) - x'(0)c + \frac{1}{2}x''(0)c^2.$$

注意到上述问题的递归结构, 即 $x^{(n)}$ 的计算只需要用到截至 $x^{(n-1)}(0)$ 的信息。因此, 上一小节当中介绍的线性化与对数线性化方法, 可以应用到任意 n 阶多项式的近似上。

7.2 应用DYNARE实现扰动法求解DSGE模型

上一节中我们看到, 从模型对应的一阶条件出发, 应用隐函数定理和泰勒中值定理, 我们可以构造政策函数的任意阶多项式近似。但是由定理7.2我们也可以看出, 对于高次多项式近似, 其计算的繁琐程度是以几何级数提高的, 如果高阶近似全部采用手工计算, 效率和容错程度都很低。幸运的是, 由于扰动法的求解过程是高度标准化的, 同时对一个函数进行泰勒近似无论是数值解还是计算机化的符号运算, 其算法都是十分成熟的。本节就介绍如何应用DYNARE软件包, 实现扰动法求解DSGE模型。

DYNARE可以从其官方网站上免费下载。^① DYNARE是一个开源的、以处理动态随机一般均衡(DSGE)模型与世代更替(OLG)模型的软件平台。DYNARE的主要功能有如下四个:

^① DYNARE官方主页: <http://www.dynare.org/>.

- 求解一个DSGE模型的稳态。
- 求解DSGE模型稳态周围的一阶与二阶近似解。
- 应用模型近似解对模型进行仿真。
- 应用极大似然估计或贝叶斯方法对DSGE模型的参数进行估计。

标准的DYNARE需要搭配MATLAB使用，这是由于DYNARE在计算过程中需要调用一些外部程序进行辅助。对于没有MATLAB的用户，DYNARE也可以搭配开源的GNU Octave使用。Octave与MATLAB类似，也是一个以数值计算为主要目的高级解释性语言，其语法与MATLAB基本一致，一些良好编写的程序在MATLAB和Octave之间可以相互移植，不过相对来说代码的执行效率要略低于MATLAB。对GNU Octave有兴趣的读者请参考其官方网站，本教程当中依然以MATLAB作为载体对DYNARE进行介绍。^①在DYNARE的四个功能当中，本教程不会涉及最后一个，即对DSGE模型进行参数估计，这属于宏观计量经济学(*Macroeometrics*)和时间序列分析(*Time-series Analysis*)的主要研究领域，与我们教程中所涉及的研究问题在目的和方法论上有本质上的不同，对宏观计量经济学感兴趣的读者可以参考DeJong and Dave (2011)。

在新版本的DYNARE在安装的时候附带了一个可以独立使用（无需MATLAB或Octave辅助）的C++版本，DYNARE++。与标准DYNARE相比，DYNARE++不能对DSGE模型进行参数估计。但是DYNARE++在求解模型政策函数的近似解的时候支持二阶以上的近似。同时，由于DYNARE++的算法实现是基于C++的，同时算法进行了高度并行化处理，所以其计算效率要高于标准DYNARE。DYNARE++的计算结果保存在.mat文件当中，这些文件可以通过MATLAB或Octave直接读取进行进一步的分析。不过由于调用DYNARE++需要读者熟悉在Windows或Unix/Linux下进行命令行操作，同时需要对系统环境变量进行一些简单的配置，本教程当中不涉及，有兴趣的读者可以参考DYNARE++的官方主页。^②

另外，由于本教程是一本初级教程，同时由于教程中涉及到的研究问题在文献中通常并不适合应用DYNARE进行求解，所以本节中对DYNARE的介绍非常的简略，仅以帮助读者上手为目的。应用DYNARE进行宏观计量经济学分析在整个宏观经济学当中可以说占据了半壁江山，有不计其数的优秀教程和活跃的技术论坛，如果读者对应用DYNARE求解与估计DSGE模型有特别的需求，请参考DYNARE官方主页上的文档、参考手册、范例以及常见问题。^③

标准DYNARE的配置非常简单，读者在网站上下载了安装文件在系统当中安装之后，只需要打开MATLAB，进入“File”选项卡下，选择“Set Path”，之后选择第一项“Add Folder...”（不要选择第二项“Add with Subfolders...”），之后将DYNARE在系统中的安装文件夹添加到列表当中即可。比如在Windows系统下，DYNARE 4.4.3版的默认安装设置下，该文件夹路径为C:/dynare/4.4.3/matlab。配置完成后，在MATLAB的交互式主界面当中执行命令dynare会显示如下的错误：

```
??? Input argument "fname" is undefined.
Error in ==> dynare at 36
if strcmpi(fname,'help')
```

此时表示DYNARE已经正确的在系统中进行了配置，否则MATLAB会提示函数或变量未定义：

```
??? Undefined function or variable 'dynare'.
```

^① GNU Octave 的官方主页：<http://www.gnu.org/software/octave/>.

^② DYNARE++的官方主页：<http://www.dynare.org/documentation-and-support/dynarepp>.

^③ 上述参考资料的网址为：<http://www.dynare.org/documentation-and-support>.

我们以上一章中的真实经济周期模型为例，对DYNARE的应用进行讲解。我们的目的依然是首先求解模型，之后再利用模型的解进行仿真，之后将仿真序列的数字特征与数据进行比照。为方便读者，这里重新给出以序列形式写出的上一章中的模型：

$$\max_{\{c_t, l_t, k_{t+1}\}_0^\infty} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t [\log c_t + A \log l_t]$$

s.t.

$$c_t + k_{t+1} - (1 - \delta)k_t = \theta_t k_t^\alpha (1 - l_t)^{1-\alpha}$$

$$\log(\theta_t) = \rho \log(\theta_{t-1}) + \varepsilon_t, \varepsilon_t \stackrel{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2)$$

$$k_0, \theta_0 \text{ 已知}, c_t, k_t \geq 0, 0 \leq l_t \leq 1.$$

模型的参数取值与6.3节当中一致，我们知道这些参数是通过6.4节的校准获得的。回忆上一小节当中，应用扰动法对模型进行求解不需要将上述问题转化为动态优化问题，扰动法是直接作用在模型的约束与一阶条件之上的。我们知道，上述模型的解是由下面的非线性方程组刻画的：

$$\begin{aligned} \frac{1}{c_t} &= \beta \mathbb{E}_t \left\{ \frac{1}{c_{t+1}} [\alpha \theta_{t+1} k_{t+1}^{\alpha-1} + 1 - \delta] \right\}, \\ \frac{A}{l_t} &= \frac{1}{c_t} (1 - \alpha) \theta_t k_t^\alpha (1 - l_t)^{-\alpha}, \\ y_t &= \theta_t k_t^\alpha (1 - l_t)^{1-\alpha}, \\ \log \theta_t &= \rho \log \theta_{t-1} + \varepsilon_t, \\ c_t + k_{t+1} - (1 - \delta)k_t &= y_t. \end{aligned}$$

可以看出上面的方程组对应了五个变量 c, k, l, y, θ ，将方程组线性化之后，我们就可以应用一阶差分方程组的一般理论进行求解。由于DYNARE在执行的过程当中将一部分运算通过MATLAB与C语言进行对接的接口MEX进行了混合编程，所以DYNARE的用户指南中推荐使用C语言相关的句法书写代码，因此下面的代码中有些句法与我们所熟悉的MATLAB程序的句法略有不同。另外，由于DYNARE的执行过程当中会生成数量众多的辅助文件，所以这里建议读者对于每个DYNARE的工程都建立一个单独的文件夹进行存储。我们首先在代码7.1当中给出求解上述模型的DYNARE程序，之后我们将对代码进行详细的解释。

```

1  /* -----
2           PROGRAM DESCRIPTION
3  -----
4
5  Purpose:
6      - This code solves the RBC model specified in Chapter 1 of Cooley and
7        Prescott (1995)
8      - Code 7.1 in the Lecture Notes
9
10 Author:
11     Xin Tang @ Wuhan University, Spring 2016
12
13 Record of Revisions:
14     Date:          Description of Changes
15     =====       =====
16     03/27/2016:    Original Code
17
18 */

```

```

19          PREAMBLES
20  -----
21 /* Variables and Parameters */
22 var y c k i n y_n z; /* z = exp(theta) */
23 varexo e;
24 parameters beta delta rho sigmae A alpha;
25
26 /* Calibration */
27 beta = 0.987;
28 delta = 0.025;
29 rho = 0.95;
30 sigmae = 0.00712;
31 A = 1.778;
32 alpha = 0.40;
33
34 /*
35          MODEL
36  -----
37 model;
38     1/c = beta*(1/c(+1))*  

39         (alpha*exp(z(+1))*(k^(alpha-1))*(n(+1))^(1-alpha) +1-delta );
40     A/(1-n) = (1/c)*(1-alpha)*exp(z)*(k(-1)^alpha)*(n^(-alpha));
41     c+i = y;
42     y = exp(z)*(k(-1)^alpha)*(n^(1-alpha));
43     i = k - (1-delta)*k(-1);
44     y_n = y/n;
45     z = rho*z(-1)+e;
46 end;
47
48 /*
49          STEADY STATES AND INITIAL VALUES
50  -----
51 initval;
52     k = 16;
53     c = 1.15;
54     n = 0.31;
55     z = 0;
56     e = 0;
57 end;
58 steady; /* Start simulation from the Steady State */
59 check; /* Check the stability of the system, Blanchard and Kahn (1980) */
60
61 /*
62          SHOCK STRUCTURE OF THE MODEL
63  -----
64 shocks;
65     var e = sigmae^2;
66 end;
67
68 /*
69          SIMULATE THE MODEL
70  -----
71 stoch_simul(hp_filter = 1600, order = 1, periods = 2100);

```

代码 7.1.—应用DYNARE求解真实经济周期模型

下面我们来详细解释代码7.1。DYNARE的代码通常保存在一个.mod文件当中，这个文件一般来说可

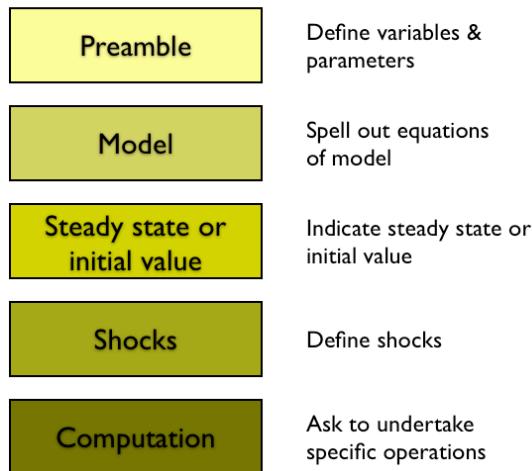


图 7.2.—DYNARE文件的一般结构

以由图7.2中所包含的五个部分组成：变量定义（18-32行）、模型设置（34-46行）、初值与稳态（48-59行）、冲击结构（61-66行）和模型仿真（68-71行）。首先我们注意到在代码7.1，程序注释不再使用%标识，而是使用一对/**/将注释语句包括起来。同时注意到，分号此时的作用不是用来隐藏语句执行的结果，而是用来显式的标识一条语句的结束。此时每一个语句之后必须以分号结尾，来标识语句的完成。同时如38-39行所示，对于需要跨行书写的长语句，我们不再使用...来断行，而是直接换行，并以分号标识长语句的结尾。熟悉C/C++语言的同学已经看出，两者都是C/C++的句法特征。另外，注释行也可以如C++当中采取//标识。

第二，变量定义部分。首先我们要区分三种变量。模型的内生变量，刻画随机冲击的外生变量，以及模型的参数，三者分别以var, varexo和parameters开头的语句定义。在22行当中变量输入的顺序在DYNARE后续的运行过程当中将会得到保留。具体来说，DYNARE在计算的时候会将同一类的变量，比如方程(7.7)线性项前面的系数阵A。显然A中元素的排列是与X_t中变量的顺序直接相关的。当前版本的DYNARE处理的方式就是在X_t保留22行当中顺序。不过在早期的DYNARE当中，变量会按照字母顺序排列，所以如果读者碰巧在一台装有早期版本DYNARE的电脑上执行程序，那么结果呈现的顺序与更新过的DYNARE会有所不同（不过这很少会发生）。另外这里我们定义了投资和劳动生产率两个变量，这两个变量在本节开始处的方程组当中我们隐性的将他们带掉了。之所以这里要列上，是因为只有当变量被显式的定义之后，DYNARE才会在相关系数等仿真的数字特征当中对它们进行计算。不过由于读者可以读取DYNARE计算的结果之后自行计算，所以这里并不必要。在26-32行的校准当中，我们使用了与第六章相同的参数值。

第三，模型设置部分。首先注意到刻画模型所需要的方程被包含在37和46行以model;开始以end;结束的结构当中。这些方程在模型当中的对应都是直接的，这里不在赘述。这里读者可以看到DYNARE的一个重要特征，那就是模型的方程可以以相对直观的方式输入，从而减少了学者之间交换代码的成本。在书写模型的时候，最为重要的是弄清模型的时间安排。注意到此处有些变量后面带有(+1)和(-1)，这就是标识动态变量所属的时间的后缀。DYNARE变量时间的约定是以变量取值被确定的那一期为准。具体来说，在模型中的t期来说，在38-39行的欧拉方程当中，变量c_{t+1}和n_{t+1}的取值因为会在t + 1期被确定，所以在代码中我们分别将它们写作c(+1)和n(+1)。但是由于下一期的资

本 k_{t+1} 是在本期(t)决定的，所以在代码中我们写为k。同理，在42行的生产函数当中，由于本期的资本 k_t 是在上一期决定的，所以此处我们写作k(-1)。这个时间上的约定非常重要，请读者在撰写代码的时候务必留心！

第四，在初值与稳态部分。这部分主要有两个作用，取决于在结构之后是否加入steady;语句。首先，无论后面是否跟着steady;语句，51-57行内的值都会被DYNARE用作求解稳态时候的初值（回忆我们上一章中对fsolve的介绍）。而如果存在steady;语句，那么在后续的仿真当中，DYNARE会使用具体的稳态作为仿真时第零期的初值，如果没有steady;语句，那么在结构initval;中包含的值就会被当做仿真的初值。由于线性近似通常只在稳态周围有较好精度，所以除非读者的目的是分析确定性模型两个稳态之间的转移路径，作者在此强烈建议读者在仿真的时候从稳态开始出发。关于转移路径的分析这里我们不做介绍，感兴趣的读者可以参考DYNARE用户指南中的讨论，King and Rebelo (1993)提供了一个具体的例子。另外这里并不需要为所有变量赋初值，只要为能够给整个系统所有变量赋值所需要的最小数量的变量赋值即可，比如这里不需要给 i 与 y 赋值，因为通过 k, n, c 的初值，他们可以被直接计算出来。

在冲击结构一部分，我们需要输入的是对应了随机过程的随机扰动项所服从的分布的参数。换句话说，对于一个AR(1)过程来说，我们需要输入的是每期i.i.d的扰动项的分布，而不是合成之后的 $\log \theta_t$ 的分布。注意到DYNARE的缺省设置下，冲击被默认为服从正态分布的，对使用其他类型分布有需求的读者请参考进阶的教程或者DYNARE的参考手册，这里不做讨论。对真实经济周期模型而言，这个冲击的结构非常简单，就是一个i.i.d的正态分布，其标准差为sigmae。不过在DYNARE的缺省设置下，我们可以很容易的引入相互关联的多个随机冲击。比如，假设模型中有两个AR(1)的冲击：

$$\begin{bmatrix} a_t \\ b_t \end{bmatrix} = \begin{bmatrix} \rho & \tau \\ \tau & \rho \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ \nu_t \end{bmatrix},$$

再假设此时 (ε_t, ν_t) 的协方差矩阵为

$$\begin{bmatrix} \sigma_\varepsilon^2 & \psi\sigma_\varepsilon\sigma_\nu \\ \psi\sigma_\varepsilon\sigma_\nu & \sigma_\nu^2 \end{bmatrix},$$

那么我们只需要将65行改写成

```
var e = sige^2;
var u = sigu^2;
var e,u = psi*sige*sigu;
```

就可以对上述冲击进行刻画。

仿真部分，在71行，我们指示DYNARE，对模型的解进行线性近似(order=1)，应用近似解进行长为2100期的仿真，并对仿真序列应用 $\lambda = 1600$ 的HP滤波进行处理，之后对周期项计算相关的随机过程数字特征。注意到这里如果使用order=2，那么DYNARE会计算模型的二次近似。标准的DYNARE最高只能进行二阶的近似，需要更高阶近似的情形请读者使用DYNARE++进行计算。在缺省设置下，前100期的仿真会被舍去，于是对periods=2100来说，最后用于计算的数据就是2000期。舍弃的期数可以通过在括号内添加选项drop=N来设置。71行的仿真命令还有很多其他可以设置的参数，本教程当中不做具体解释，有需要的读者可以参考DYNARE的参考手册或用户指南。

现在我们来执行这个文件，我们假设这个文件的名字为RBC.mod，那么我们只需要将MATLAB的工作文件夹切换到该文件所在的地址，之后在MATLAB的主窗口键入命令

```
dynare RBC.mod
```

MATLAB就会调用DYNARE求解模型，并给出计算结果。这里限于篇幅我们不具体介绍DYNARE运行

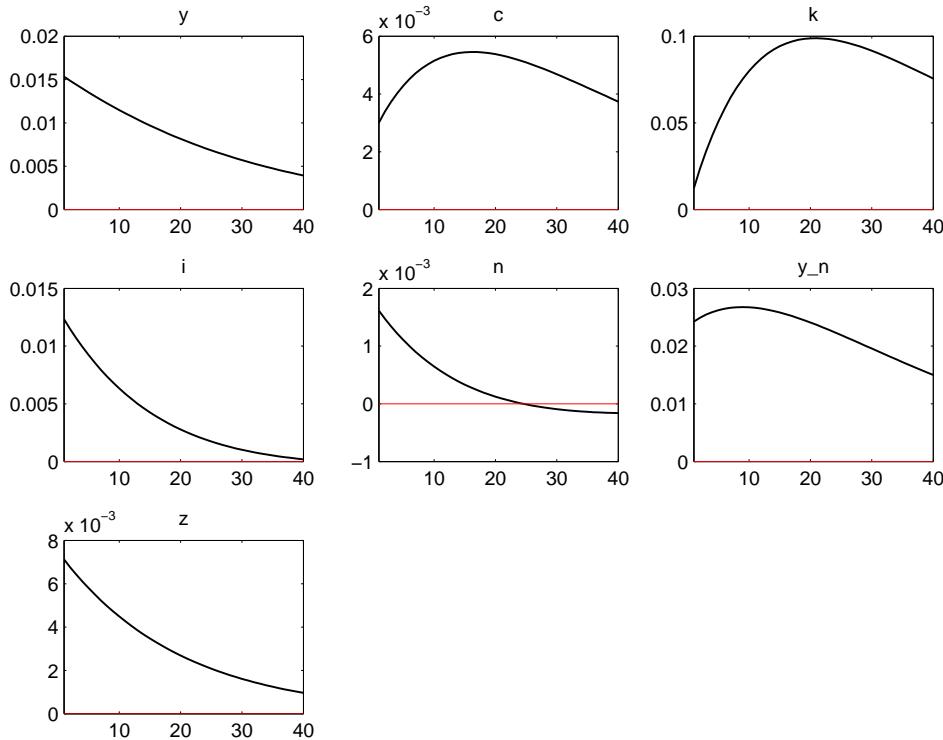


图 7.3.—真实经济周期模型的脉冲反应

的结果，这些结果所对应的内容是直观的，这里仅举几例。对应了代码7.1，产出的政策函数为

$$y_t = 1.503 + 0.024k_t + 2.046z_t + 2.154\varepsilon_t.$$

此时在表MATRIX OF CORRELATIONS之下，我们可以看到产出与消费、投资、劳动时间和劳动生产率之间的相关系数分别为0.88, 0.99, 0.98和0.98，这与表6.1中我们通过非线性解所得到的值是非常相似的。另外，在缺省设置下，DYNARE运行代码7.1之后还会输出如图7.3所示的脉冲反应(Impulse Response)。脉冲反应的概念非常简单，实际上就是假设第零期的时候系统运行在稳态当中 ($\varepsilon_0 = 0$)，在 $t = 1$ 时对整个系统施加一个大小等于一个标准差的冲击 ($\varepsilon_1 = \sigma_\varepsilon$)，之后令 $t \geq 2$ 开始冲击再恢复为零 ($\varepsilon_t = 0$)。在上述冲击结构下，判断各变量对于冲击的反应。脉冲反应通常被用来刻画模型对于某个冲击反应的机制，注意到脉冲反应使用我们第六章得到的非线性解也可以很容易的实现。DYNARE运行之后的结果保存在生成的.mat文件当中，关于变量具体的存储规则，由于介绍起来比较繁琐，而且写在本教程中无非是为DYNARE的参考手册进行中文翻译，这里从略，有需要的读者请自行查阅相关文档。

至此我们就完成了对于扰动法和DYNARE软件平台的简要介绍。有的读者可能会想，既然利用DYNARE进行线性近似求解迅速，代码撰写方便，而得到的模型的解在精度上似乎相比第六章的非线性解也并没有很显著的下降。那么，实践当中我们为什么还需要学习非线性的全局解呢？作为初级教程，这里我们不展开叙述。注意到第六章中我们得到的政策函数的非线性并不明显，真实的政策函数原本就近似线性，于是线性解给出较好的近似效果并不意外。但是对于很多宏观经济模型来说，政策函数可能具有很强的非线性，这时候应用线性近似就会导致较大的误差。也许有的读者会考虑使用高阶近似来刻画政策函数的曲率，但是注意扰动法本质上是对政策函数在稳态附近的一个近似，这个近似的过程当中，模型在边界处由于其他的约束条件（比如下一章我们要看到的流动性约

束)造成的政策函数的弯曲是无法被高阶多项式保持的,因为在使用扰动法进行近似的时候,这些约束条件被处理成不存在的。所以当模型存在偶尔紧的约束(*occasionally-binding constraints*)的时候,扰动法就是不适用的。注意到约束必须是时而紧的,像资源约束一类的总是紧(*always binding*)的约束在扰动法当中总是被遵守的。关于这一点的详细讨论,请参考Anagnostopoulos and Tang (2015)。具体来说,该文的结果表明,高阶多项式无法近似在边界处因为存在偶尔紧的约束造成的政策函数的弯曲。同时,在数值仿真当中,非线性的全局解对应了一个平稳分布,而扰动法的政策函数进行的仿真对足够长的序列而言总是发散的。在本教程的最后一部分我们将会看到,由于对于不完全市场下的异质性消费者的Bewley-Aiyagari-Huggett模型而言[Bewley (1977), Huggett (1993)和Aiyagari (1994)],偶尔紧的约束几乎一定存在的,所以DYNARE无法用来处理此类模型。而此类模型,是当前数量宏观经济学用来分析消费者问题的工作模型(*workhorse model*),于是这很大程度上限制了扰动法在前沿宏观经济问题中的应用。另一方面,DYNARE本身的句法结构也并不适合处理具有大量消费者的问题。同时即便是对代表性消费者的问题而言,如果模型存在更加复杂的结构,比如Klein and Ríos-Rull (2003)、Klein, Krusell and Ríos-Rull (2008)的马尔科夫精炼均衡(*Markovian Perfect Equilibrium*),那么也不能直接应用DYNARE进行求解。但是DYNARE在分析具有复杂冲击结构的货币经济学模型以及结构宏观计量经济学模型当中得到了非常广泛的应用,有兴趣的读者请参考相关的教材。^①

^① 读者可以参考Woodford (2003),Walsh (2010), DeJong and Dave (2011), 以及Galí (2015)。

第三部分

收入与财富分配

第八章 流动性约束与预防性储蓄

第九章 异质性消费者问题

第十章 经济周期中的异质性消费者

第十一章 结语

参考文献

- Abel, Andrew B.** 2003. "The Effects of a Baby Boom on Stock Prices and Capital Accumulation in the Presence of Social Security." *Econometrica*, 71(2): 551–578. 51
- Acemoglu, Daron.** 2009. *Introduction to Modern Economic Growth*. New Jersey:Princeton University Press. 9
- Adamopoulos, Tasso, and Diego Restuccia.** 2014. "The Size Distribution of Farms and International Productivity Difference." *American Economic Review*, 104(6): 1667–1697. 42
- Adda, Jérôme, and Russell Cooper.** 2003. *Dynamic Economics: Quantitative Methods and Applications*. Cambridge, Massachusetts and London, England:The MIT Press. iii
- Aiyagari, S. Rao.** 1994. "Uninsured Idiosyncratic Risk and Aggregate Saving." *The Quarterly Journal of Economics*, 109(3): 659–684. 10, 141
- Anagnostopoulos, Alexis, and Xin Tang.** 2015. "Evaluating Linear Approximations in a Two-Country Model with Occasionally Binding Borrowing Constraints." *The B.E. Journal of Macroeconomics*, 15(1): 43–91. 10, 141
- Anagnostopoulos, Alexis, Eva Cárceres-Poveda, and Danmo Lin.** 2012. "Dividend and Capital Gains Taxation under Incomplete Markets." *Journal of Monetary Economics*, 59(7): 599–611. 56
- Azzimonti, Marina, Eva de Francisco, and Per Krusell.** 2008. "Aggregation and Aggregation." *Journal of the European Economic Association Papers and Proceedings*, 6(2-3): 381–394. 91
- Barro, Robert J., and Xavier Sala-i-Martin.** 2003. *Economic Growth*. . 2nd ed., Cambridge, Massachusetts:The MIT Press. 13
- Bewley, Truman.** 1977. "The Permanent Income Hypothesis: A Theoretical Formulation." *Journal of Economic Theory*, 16(2): 252–292. 141
- Blanchard, Olivier Jean, and Charles M. Kahn.** 1980. "The Solution of Linear Difference Models under Rational Expectations." *Econometrica*, 48(5): 1305–1311. 131
- Blanchard, Olivier Jean, and Stanley Fischer.** 1989. *Lectures on Macroeconomics*. Cambridge, Massachusetts; London, England:The MIT Press. 9
- Bloom, Nicholas.** 2009. "The Impact of Uncertainty Shocks." *Econometrica*, 77(3): 623–685. 55
- Boppert, Timo.** 2014. "Structural Change and the Kaldor Facts in A Growth Model with Relative Price Effects and Non-gorman Preferences." *Econometrica*, 82(6): 2167–2196. 125
- Brock, William A., and Leonard J. Mirman.** 1972. "Optimal Economic Growth and Uncertainty: The Discounted Case." *Journal of Economic Theory*, 4(3): 479–513. 9, 91
- Burns, Arthur F., and Wesley C. Mitchell.** 1946. *Measuring Business Cycles. NBER Book Series Studies in Business Cycles*, New York:National Bureau of Economic Research. 49
- Carroll, Christopher D.** 1997. "Buffer-Stock Saving and the Life Cycle/Permanent Income Hypothesis." *The Quarterly Journal of Economics*, 112(1): 1–55. 10, 85
- Caselli, Francesco.** 2005. "Accounting for Cross-Country Income Differences." In *Handbook of Economic Growth*. Vol. 1A, , ed. Philippe Aghion and Steven N. Durlauf, Chapter 9, 679–741. North Holland:Elsevier B. V. 42
- Cass, David.** 1965. "Optimum Growth in an Aggregative Model of Capital Accumulation." *The Review of Economic Studies*, 32(3): 233–240. 9
- Chari, V. V., Patrick J. Jehoe, and Ellen R. McGrattan.** 2002. "Accounting for the Great Depression." *American Economic Review*, 92(2): 22–27. 55
- Chari, V. V., Patrick J. Kehoe, and Ellen R. McGrattan.** 2007. "Business Cycle Accounting." *Econometrica*, 75(3): 781–836. 55
- Cobb, Charles W., and Paul H. Douglas.** 1928. "A Theory of Production." *American Economic Review*, 18(1): 139–165. 125

- Cocco, João F., Francisco J. Gomes, and Pascal J. Maenhout.** 2005. "Consumption and Portfolio Choice over the Life Cycle." *The Review of Financial Studies*, 18(2): 491–533. 8
- Coleman, Wilbur John II.** 1990. "Solving the Stochastic Growth Model by Policy-Function Iteration." *Journal of Business & Economic Statistics*, 8(1): 27–29. 117
- Cooley, T. F., and Edward C. Prescott.** 1995. "Economic Growth and Business Cycles." In *Frontiers of Business Cycle Research*. , ed. T. F Cooley, Chapter 1, 1–38. New Jersey:Princeton University Press. 9, 91, 111, 112, 122, 123, 125, 126
- Davis, Philip J., and Philip Rabinowitz.** 1984. *Methods of Numerical Integration*. . 2nd ed., New York:Academic Press. 89
- Dawkins, Christina, T.N. Srinivasan, and John Whalley.** 2001. "Calibration." In *Handbook of Econometrics*. Vol. 5, , ed. James J. Heckman and Edward Leamer, Chapter 58, 3653–3703. Amsterdam:North-Holland. 123
- Deaton, Angus.** 1991. "Saving and Liquidity Constraints." *Econometrica*, 59(5): 1221–1248. 10
- DeJong, David N., and Chetan Dave.** 2011. *Structural Macroeconomics*. . 2nd ed., Princeton, NJ:Princeton University Press. 135, 141
- Dixit, Avinash K.** 1990. *Optimization in Economic Theory*. . 2nd ed., Oxford:Oxford University Press. 20
- Domar, Evsey.** 1946. "Capital Expansion, Rate of Growth, and Employment." *Econometrica*, 14(2): 137–147. 13
- Duffie, Darrell, and Kenneth J. Singleton.** 1993. "Simulated Moments Estimation of Markov Models of Asset Prices." *Econometrica*, 61(4): 929–952. 124
- Elaydi, Saber.** 2005. *An Introduction to Difference Equations*. . 3rd ed., New York:Springer. 128
- Elsby, Michael W. L., Ryan Michaels, and David Ratner.** 2015. "The Beveridge Curve: A Survey." *Journal of Economic Literature*, 53(3): 571–630. 51
- Epstein, Larry G., and Stanley E. Zin.** 1989. "Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework." *Econometrica*, 57(4): 937–969. 100
- Flodén, Martin.** 2008. "A Note on the Accuracy of Markov-Chain Approximations to Highly Persistent AR(1) Process." *Economic Letters*, 99(3): 516–520. 85, 90
- Galí, Jordi.** 2015. *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework and Its Applications*. . 2nd ed., Princeton, NJ:Princeton University Press. 141
- Geanakoplos, John, Michael Magill, and Martine Quinzii.** 2004. "Demography and the Long-Run Predictability of the Stock Market." *Brookings Papers on Economic Activity*, 2004(1): 241–307. 51
- Glover, Andrew, Jonathan Heathcote, Dirk Krueger, and José-Víctor Ríos-Rull.** 2014. "Intergenerational Redistribution in the Great Recession." *Manuscript*. 55
- Gomes, Francisco, and Alexander Michaelides.** 2005. "Optimal Life-Cycle Asset Allocation: Understanding the Empirical Evidence." *The Journal of Finance*, 60(2): 869–904. 100
- Gouriéroux, Christian, and Alain Monfort.** 1996. *Simulation-Based Econometric Methods. CORE Lectures Series*, New York:Oxford University Press. 124
- Gourinchas, Pierre-Olivier, and Jonathan A. Parker.** 2002. "Consumption over the Life Cycle." *Econometrica*, 70(1): 47–89. 10, 85
- Gourio, François, and Jianjun Miao.** 2010. "Firm Heterogeneity and the Long-Run Effects of Dividend Tax Reform." *American Economic Journal: Macroeconomics*, 2(1): 131–168. 56
- Guvenen, Fatih.** 2009. "A Parsimonious Macroeconomic Model for Asset Pricing." *Econometrica*, 77(6): pp. 1711–1750. 100
- Hall, Robert E., and Charles I. Jones.** 1999. "Why Do Some Countries Produce So Much More Output Per Worker Than Others?" *The Quarterly Journal of Economics*, 114(1): 83–116. 42
- Hansen, Gary D.** 1985. "Indivisible Labor and the Business Cycle." *Journal of Monetary Economics*, 16: 309–327. 9, 112
- Hansen, Gary D., and Lee E. Ohanian.** 2016. "Neoclassical Models in Macroeconomics." *NBER Working Paper No. 22122*. 51
- Hansen, Lars Peter, and James J. Heckman.** 1996. "The Empirical Foundations of Calibration." *Journal of Economic Perspectives*, 10(1): 87–104. 124
- Harold, Roy F.** 1939. "An Essay in Dynamic Theory." *The Economic Journal*, 49(193): 14–33. 13
- Heer, Burkhard, and Alfred Maußner.** 2005. *Dynamic General Equilibrium Modelling: Computational Methods and Applications*. Berlin, Heidelberg, Germany:Springer. iii
- Hirsch, Morris W., Stephen Smale, and Robert L. Devaney.** 2004. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. . 2nd ed., California:Academic Press and Elsevier. 128
- Hodrick, Robert J., and Edward C. Prescott.** 1997. "Postwar U.S. Business Cycles: An Empirical Investigation." *Journal of*

- Money, Credit and Banking*, 29: 1–16. 48
- Hsieh, Chang-Tai, and Peter J. Klenow.** 2009. “Misallocation and Manufacturing TFP in China and India.” *The Quarterly Journal of Economics*, 124(4): 1403–1448. 42
- Huggett, Mark.** 1993. “The Risk-Free Rate in Heterogeneous-Agent Incomplete-Insurance Economies.” *Journal of Economic Dynamics and Control*, 17: 953–969. 10, 141
- Judd, Kenneth L.** 1998. *Numerical Methods in Economics*. Cambridge, Massachusetts and London, England:MIT Press. 127
- Kaldor, Nicholas.** 1957. “A Model of Economic Growth.” *The Economic Journal*, 67(268): 591–624. 39
- Karabarbounis, Loukas, and Brent Neiman.** 2014. “The Global Decline of the Labor Share.” *The Quarterly Journal of Economics*, 129(1): 61–103. 125
- Karlin, Samuel, and Howard M. Taylor.** 1975. *A First Course in Stochastic Processes*. . 2nd ed., San Diego, California:Academic Press. 71
- King, Robert G., and Sergio T. Rebelo.** 1993. “Transitional Dynamics and Economic Growth in the Neoclassical Model.” *The American Economic Review*, 83(4): 908–931. 139
- King, Robert G., Charles I. Plosser, and Sergio T. Rebelo.** 1988. “Production, Growth and Business Cycles: II. New Directions.” *Journal of Monetary Economics*, 21(2-3): 309–341. 125
- Klein, Paul, and José-Víctor Ríos-Rull.** 2003. “Time-Consistent Optimal Fiscal Policy.” *International Economic Review*, 44(4): pp. 1217–1245. 141
- Klein, Paul, Per Krusell, and José-Víctor Ríos-Rull.** 2008. “Time-Consistent Public Policy.” *The Review of Economic Studies*, 75(3): 789–808. 141
- Koopmans, Tjalling C.** 1947. “Measurement without Theory.” *The Review of Economics and Statistics*, 29(3): 161–172. 46, 49
- Koopmans, Tjalling C.** 1963. “On the Concept of Optimal Economic Growth.” *Cowles Foundation for Research in Economics No. 163*. 9
- Krusell, Per, and Jr. Smith, Anthony A.** 1998. “Income and Wealth Heterogeneity in the Macroeconomy.” *The Journal of Political Economy*, 106(5): 867–896. 10, 60, 62
- Kydland, Finn E., and Edward C. Prescott.** 1982. “Time to Build and Aggregate Fluctuations.” *Econometrica*, 50(6): 1345–1370. iii, 46
- Ljungqvist, Lars, and T. J. Sargent.** 2012. *Recursive Macroeconomic Theory*. . 3rd ed., Cambridge, Massachusetts and London, England:MIT Press. iii, iv, 56
- Long, John B., and Charles I. Plosser.** 1983. “Real Business Cycles.” *Journal of Political Economy*, 91(1): 39–69. iii
- Lucas, Robert E.** 1972. “Expectations and the Neutrality of Money.” *Journal of Economic Theory*, 4: 103–124. 46
- Lucas, Robert E.** 1975. “An Equilibrium Model of the Business Cycle.” *Journal of Political Economy*, 83(6): 1113–1144. 46
- Lucas, Robert E.** 1976. “Econometric Policy Evaluation: A Critique.” *Carnegie-Rochester Conference Series on Public Policy*, 1: 19 – 46. 46
- Mas-Colell, Andrew, Michael D. Whinston, and Jerry R. Green.** 1995. *Microeconomic Theory*. New York:Oxford University Press. 56, 59, 100
- McGrattan, Ellen R.** 2012. “Capital Taxation During the U.S. Great Depression.” *The Quarterly Journal of Economics*, 127(3): 1515–1550. 55
- Muth, John F.** 1961. “Rational Expectations and the Theory of Price Movements.” *Econometrica*, 29(3): 315–335. 46
- Pakes, Ariel, and David Pollard.** 1989. “Simulation and the Asymptotics of Optimization Estimators.” *Econometrica*, 57(5): 1027–1057. 124
- Phelps, Edmund.** 1966. *Golden Rules of Economic Growth*. New York:Norton. 42
- Pratt, John W.** 1964. “Risk Aversion in the Small and in the Large.” *Econometrica*, 32(1/2): 122–136. 99
- Prescott, Edward C.** 1986. “Theory Ahead of Business Cycle Measurement.” *Federal Reserve Bank of Minneapolis Quarterly Review*, 10: 9–22. 49, 125
- Ramsey, Frank P.** 1928. “A Mathematical Theory of Saving.” *The Economic Journal*, 38(152): 543–559. 9
- Ravn, Morten O., and Harald Uhlig.** 2002. “On Adjusting the Hodrick-Prescott Filter for the Frequency of Observations.” *The Review of Economics and Statistics*, 84(2): 371–376. 49
- Rogerson, Richard.** 1988. “Indivisible Labor, Lotteries and Equilibrium.” *Journal of Monetary Economics*, 21(1): 3–16. 9
- Romer, David.** 1996. *Advanced Macroeconomics. Advanced Series in Economics*. 2nd ed., McGraw-Hill. 9
- Romer, Paul M.** 1986. “Increasing Returns and Long-Run Growth.” *Journal of Political Economy*, 94(5): 1002–1037. 13

- Romer, Paul M.** 1990. "Endogenous Technological Change." *Journal of Political Economy*, 98(5): S71–S102. 13
- Sargent, Thomas J., and Neil Wallace.** 1975. "'Rational' Expectations, the Optimal Monetary Instrument, and the Optimal Money Supply Rule." *Journal of Political Economy*, 83(2): 241–254. 46
- Schmitt-Grohé, Stephanie, and Martín Uribe.** 2004. "Solving Dynamic General Equilibrium Models using a Second-Order Approximation to the Policy Function." *Journal of Economic Dynamics and Control*, 28(4): 755–775. 127
- Shimer, Robert.** 2005. "The Cyclical Behavior of Equilibrium Unemployment and Vacancies." *American Economic Review*, 95(1): 25–49. 51
- Solow, Robert M.** 1956. "A Contribution to the Theory of Economic Growth." *The Quarterly Journal of Economics*, 70(1): pp. 65–94. 13
- Solow, Robert M.** 1970. *Growth Theory: An Exposition*. New York:Oxford University Press. 39
- Solow, Robert M.** 2008. "The State of Macroeconomics." *Journal of Economic Perspectives*, 22(1): 243–249. 13
- Stern, Steven.** 1997. "Simulation-Based Estimation." *Journal of Economic Literature*, 35(4): 2006–2039. 124
- Stokey, Nancy L., and Robert E. Lucas.** 1989. *Recursive Methods in Economic Dynamics*. Cambridge, Massachusetts:Harvard University Press. iii, 27, 30, 33, 63, 92
- Storesletten, Kjetil, Christopher I. Telmer, and Amir Yaron.** 2004a. "Consumption and Risk Sharing over the Life Cycle." *Journal of Monetary Economics*, 51(3): 609–633. 85
- Storesletten, Kjetil, Christopher I. Telmer, and Amir Yaron.** 2004b. "Cyclical Dynamics in Idiosyncratic Labor Market Risk." *Journal of Political Economy*, 112(3): 695–717. 5
- Swan, Trevor W.** 1956. "Economic Growth and Capital Accumulation." *Economic Record*, 32(2): 334–361. 13
- Tauchen, George.** 1986. "Finite State Markov-chain Approximations to Univariate and Vector Autoregressions." *Economic Letters*, 20: 177–181. 47, 73, 79, 83, 84, 89
- Tauchen, George, and Robert Hussey.** 1991. "Quadrature-Based Methods for Obtaining Approximate Solutions to Nonlinear Asset Pricing Models." *Econometrica*, 59(2): pp. 371–396. 47, 80, 85, 89
- Uzawa, Hirofumi.** 1965. "Optimum Technical Change in An Aggregative Model of Economic Growth." *International Economic Review*, 6(1): 18–31. 42
- Walsh, Carl E.** 2010. *Monetary Theory and Policy*. . 3rd ed., Cambridge, Massachusetts:The MIT Press. 141
- Woodford, Michael.** 2003. *Interest and Prices: Foundations of a Theory of Monetary Policy*. New Jersey:Princeton University Press. 141