强化学习第三章：基于模型的动态规划方法

马俏颖

2022-11-27

## 3 基于模型的动态规划方法

### 3.1 基于模型的动态规划方法理论

上⼀章我们将强化学习的问题纳⼊到马尔科夫决策过程的框架下解决。一个完整的已知模型的马尔科夫决策过程可以利用元组来表⽰。其中S为状态集，A为动作集，P为转移概率，也就是对应着环境和智能体的模型，r为回报函数，γ为折扣因子用来计算累积回报R。

累积回报公式为。当T为有限值时，强化学习过程称为有限范围强化学习，当T=∞时，称为无穷范围强化学习。我们以有限范围强化学习为例进行讲解。

强化学习的目标是找到最优策略π使得累积回报的期望最⼤。所谓策略是指状态到动作的映射π：s→a，用τ表⽰从状态s到最终状态的⼀个序列，则累积回报是个随机变量，随机变量无法进⾏优化，无法作为目标函数，我们采用随机变量的期望作为目标函数，即作为目标函数。

用公式来表示强化学习的目标：。强化学习的最终目标是找到最优策略为，我们看⼀下这个表达式的直观含义。

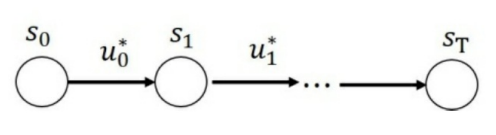


图3.1

如图3.1所⽰，最优策略的目标是找到决策序列，因此从⼴义上来讲，强化学习可以归结为序贯决策问题。即找到⼀个决策序列，使得目标函数最优。这里的目标函数是累积回报的期望值，累积回报的含义是评价策略完成任务的总回报，所以目标函数等价于任务。

强化学习的直观目标是找到最优策略，目的是更好地完成任务。回报函数对应着具体的任务，所以强化学习所学到的最优策略是与具体的任务相对应的。从这个意义上来说，强化学习并不是万能的，它无法利用⼀个算法实现所有的任务。

从⼴义上讲，强化学习是序贯决策问题。但序贯决策问题包含的内容更丰富。它不仅包含⻢尔科夫过程的决策，⽽且包括⾮⻢尔科夫过程的决策。

在上一节，我们已经将强化学习纳⼊到⻢尔科夫决策过程MDP的框架之内。马尔科夫决策过程可以利用元组来描述，根据转移概率P是否已知，可以分为基于模型的动态规划⽅法和基于无模型的强化学习⽅法，如图3.2所示。两种类别都包括策略迭代算法，值迭代算法和策略搜索算法。不同的是，在⽆模型的强化学习⽅法中，每类算法又分为online和offline两种。online和offline的具体含义，我们会在下⼀章中详细介绍。

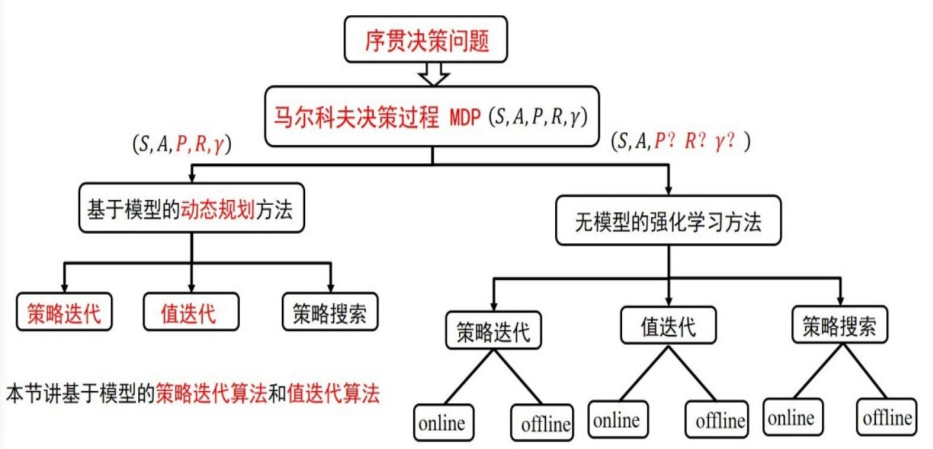


图3.2

基于模型的强化学习可以利用动态规划的思想来解决。顾名思义，动态规划中的“动态”蕴含着序列和状态的变化；“规划”蕴含着优化，如线性优化，二次优化或者非线性优化。利用动态规划可以解决的问题需要满足两个条件：⼀是整个优化问题可以分解为多个子优化问题；二是子优化问题的解可以被存储和重复利⽤。

前面已经讲过，强化学习可以利用马尔科夫决策过程来描述，利用贝尔曼最优性原理得到贝尔曼最优化⽅程：

在这里回顾一下上一章的知识：

状态值函数为：

状态-行为值函数为：

状态转移概率是包含动作的，即：

定义：状态最优值函数为在所有策略中值最大的函数值,即，最优状态-行为值函数为在所有策略中最大的状态—行为值函数。

将(3.3)式代入(3.2)便得到状态值函数的计算公式：

状态s处的值函数，可以利用后继状态的值函数来表⽰。可是有⼈会说，后继状态的值函数也是未知的，那么怎么计算当前状态的值函数，这不是自己抬自己吗？如图3.4所示。



图3.4

这正是bootstrapping算法(自举算法)！

如何求解(3.4)的⽅程？

首先，我们从数学的⾓度去解释⽅程(3.4)。对于模型已知的强化学习算法，⽅程(3.4)中的,和都是已知数，为要评估的策略是指定的，也是已知值。⽅程(3.4)中唯⼀的未知数是值函数，从这个⾓度理解⽅程(3.4)可知，⽅程(3.4)是关于值函数的线性⽅程组，其未知数的个数为状态的总数，⽤|S|来表⽰。

此处，我们使用高斯-赛德尔迭代算法进⾏求解。即：

下面我们给出策略评估算法的伪代码，如图3.5所示：

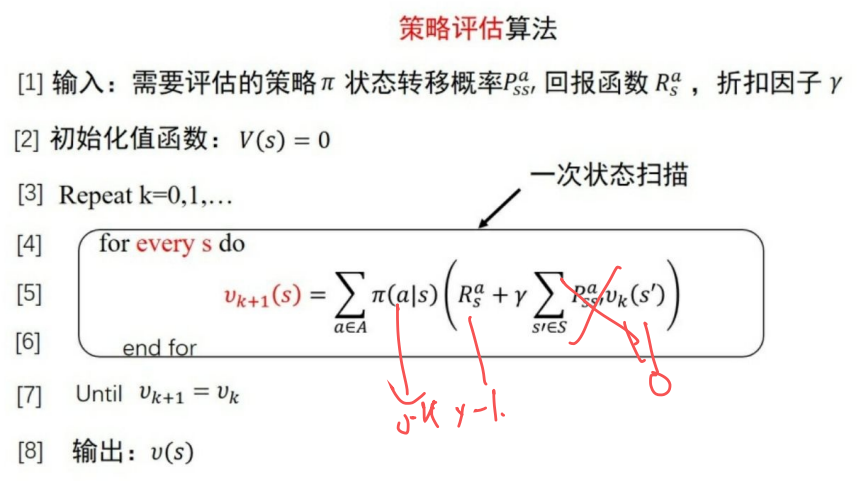


图3.5

需要注意的是，每次迭代都需要对状态集进⾏⼀次遍历（扫描）以便评估每个状态的值函数。

接下来，我们举个策略评估的例⼦。

如图3.6所示为网格世界，其状态空间为，动作空间为，回报函数为r≡-1，需要评估的策略为均匀随机策略：

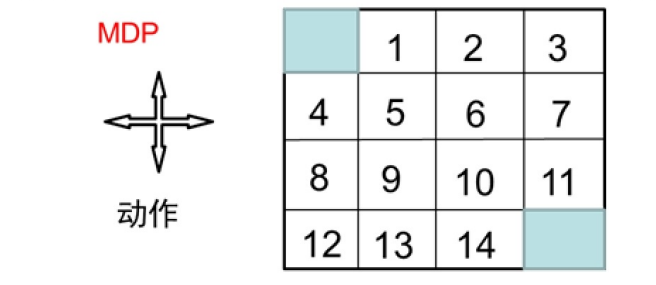


图3.6

图3.7为值函数迭代过程中值函数的变化。为了进⼀步说明，我们举个具体的例⼦，如从K=1到K=2时，状态1处的值函数计算过程。



图3.7

由公式（3.5）得到：

保留两位有效数字便是-1.7。

计算值函数的目的是利用值函数找到最优策略。第二个要解决的问题是：如何利⽤值函数进⾏策略改善，从而得到最优策略？

一个很自然的⽅法是当已知当前策略的值函数时，在每个状态采⽤贪婪策略对当前策略进⾏改善,即

如图3.9所示为⽅格世界贪婪策略的⽰意图。我们仍然以状态1为例得到改善的贪婪策略：

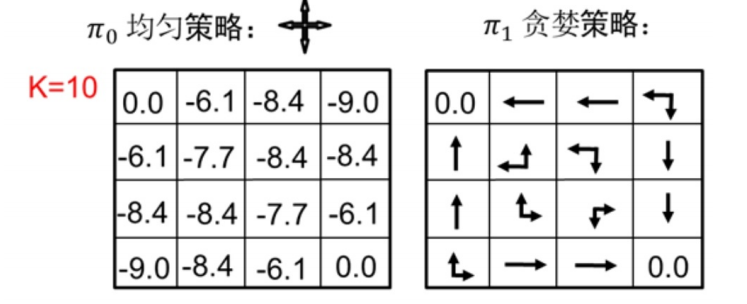


图3.9

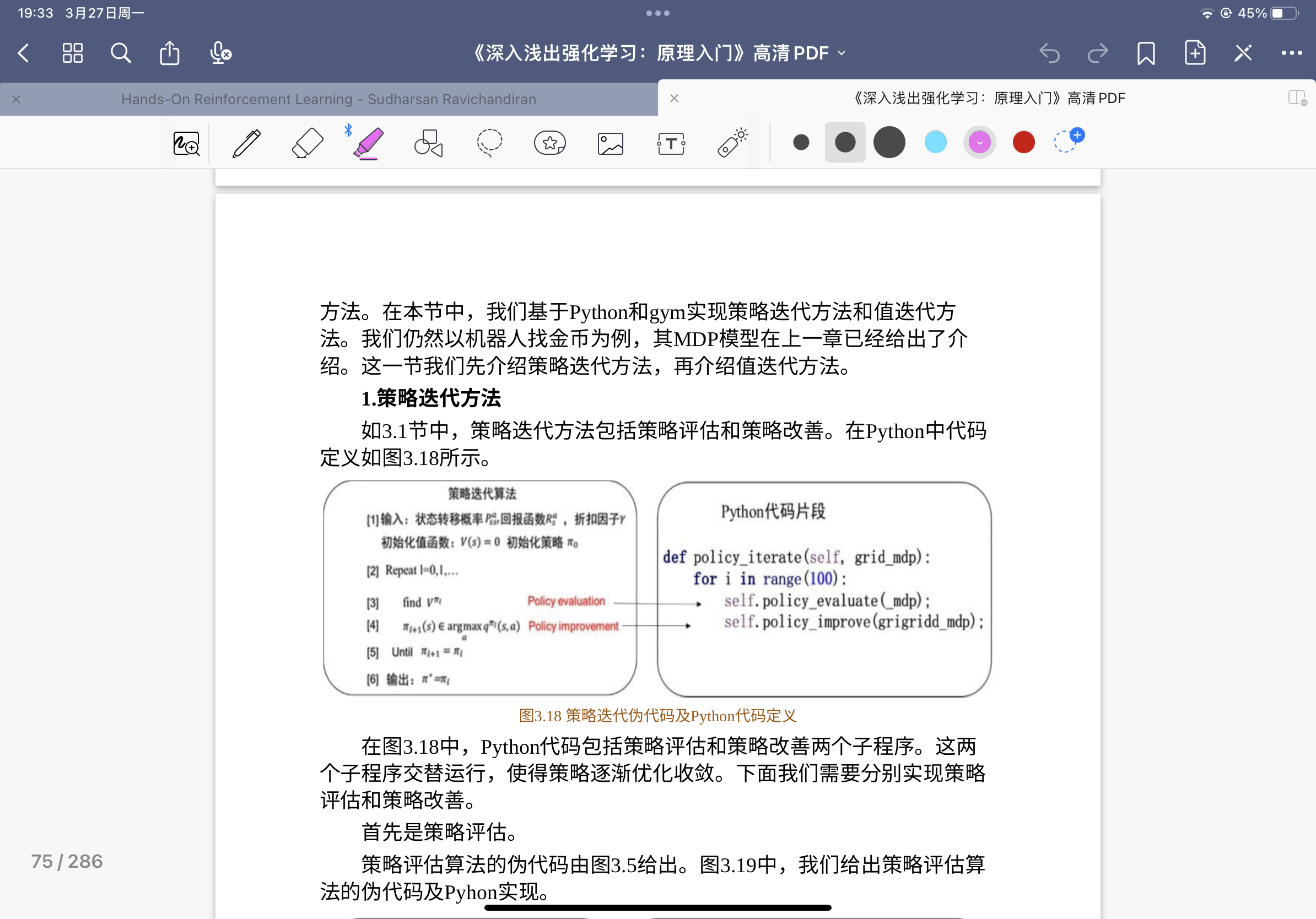
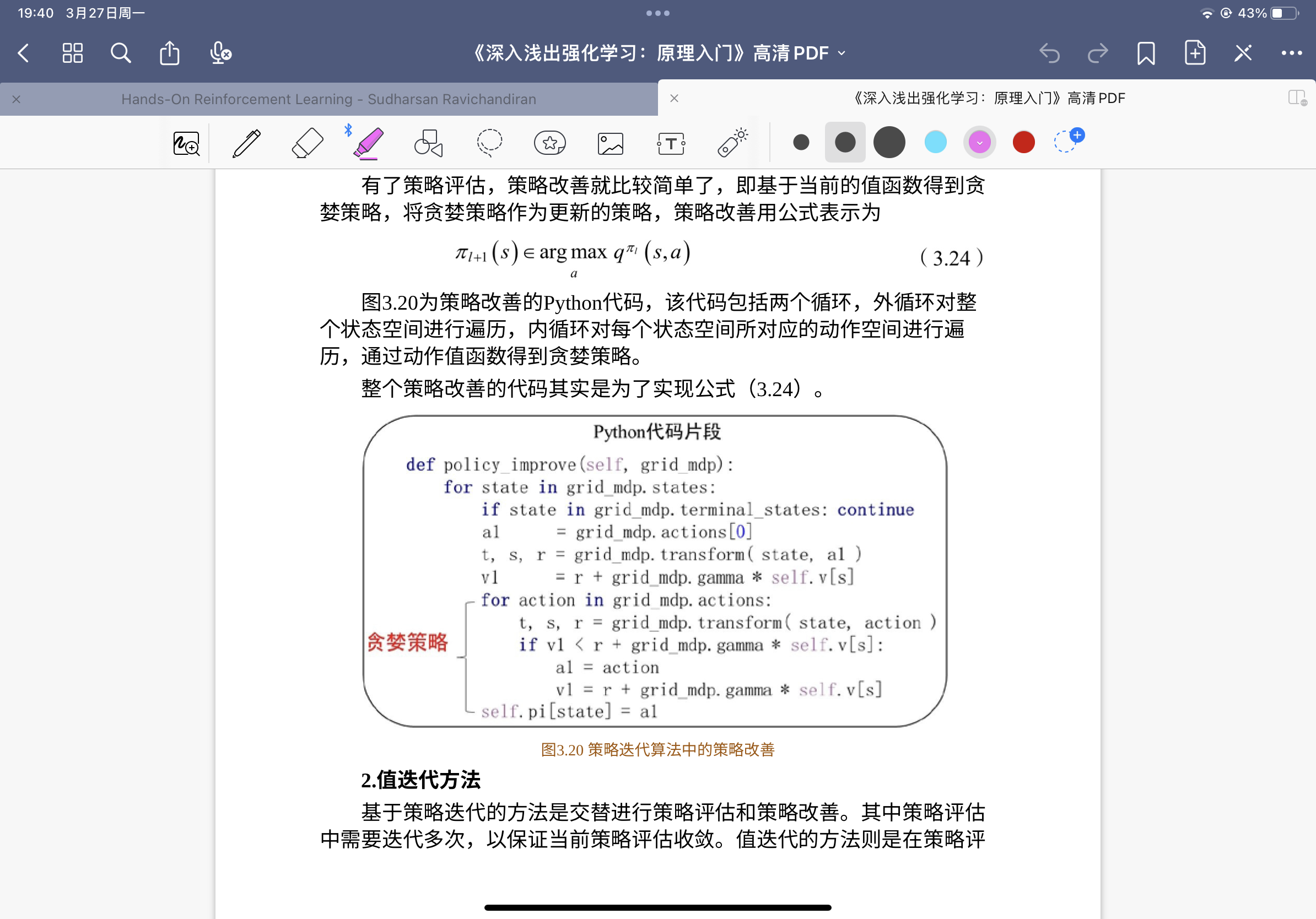
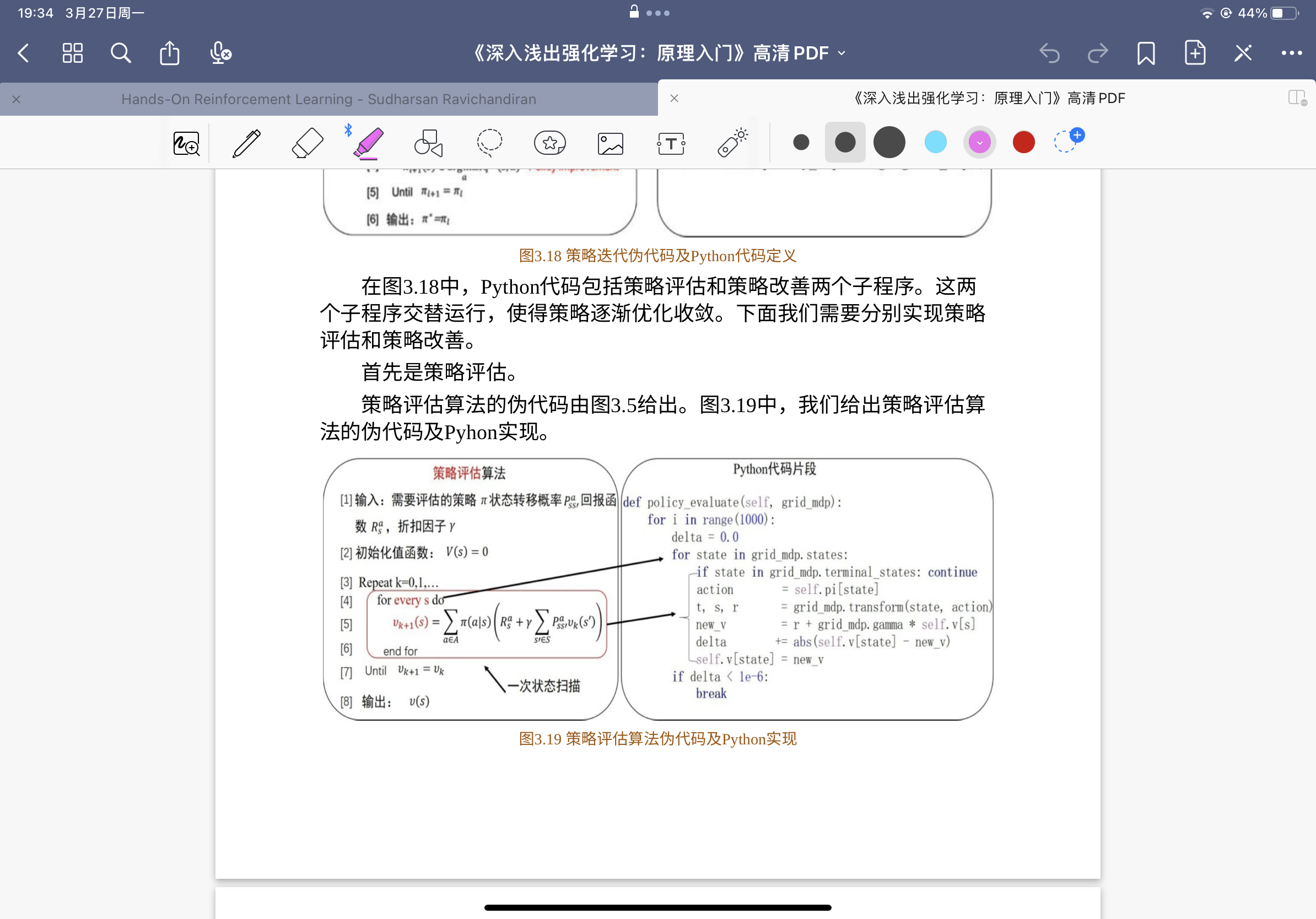
至此，我们已经给出了策略评估算法和策略改善算法。万事已具备，将策略评估算法和策略改善算法合起来便组成了策略迭代算法，如图3.10所示。

图3.10



策略迭代算法包括策略评估和策略改善两个步骤。在策略评估中，给定策略，通过数值迭代算法不断计算该策略下每个状态的值函数，利⽤该值函数和贪婪策略得到新的策略。如此循环下去，最终得到最优策略。这是⼀个策略收敛的过程。从策略迭代的伪代码我们看到，进⾏策略改善之前需要得到收敛的值函数。值函数的收敛往往需要很多次迭代，现在的问题是进⾏策略改善之前⼀定要等到策略值函数收敛吗？

对于这个问题，我们还是先看一个例⼦。如图3.12所示，策略评估迭代10次和迭代无穷次所得到的贪婪策略是一样的。因此，对于上面的问题，我们的回答是不一定等到策略评估算法完全收敛。如果我们在评估⼀次之后就进⾏策略改善，则称为值函数迭代算法。

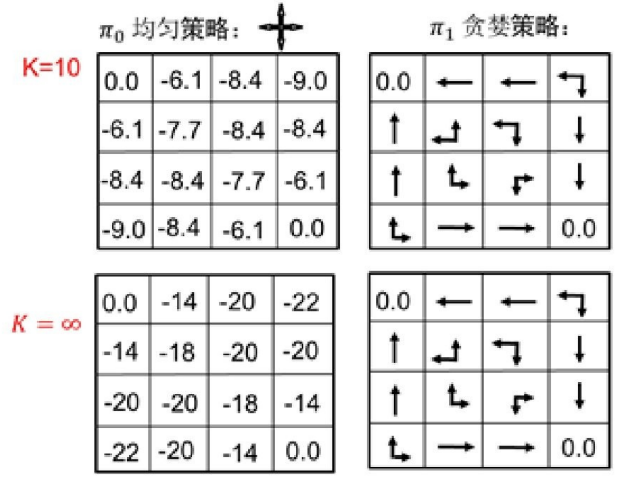


图3.12

值函数迭代算法（如图3.13所⽰）的伪代码为：

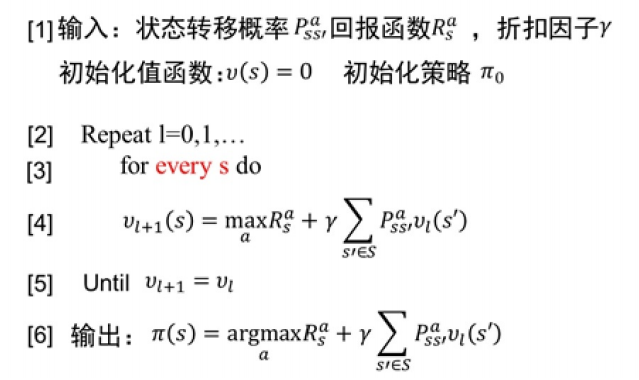
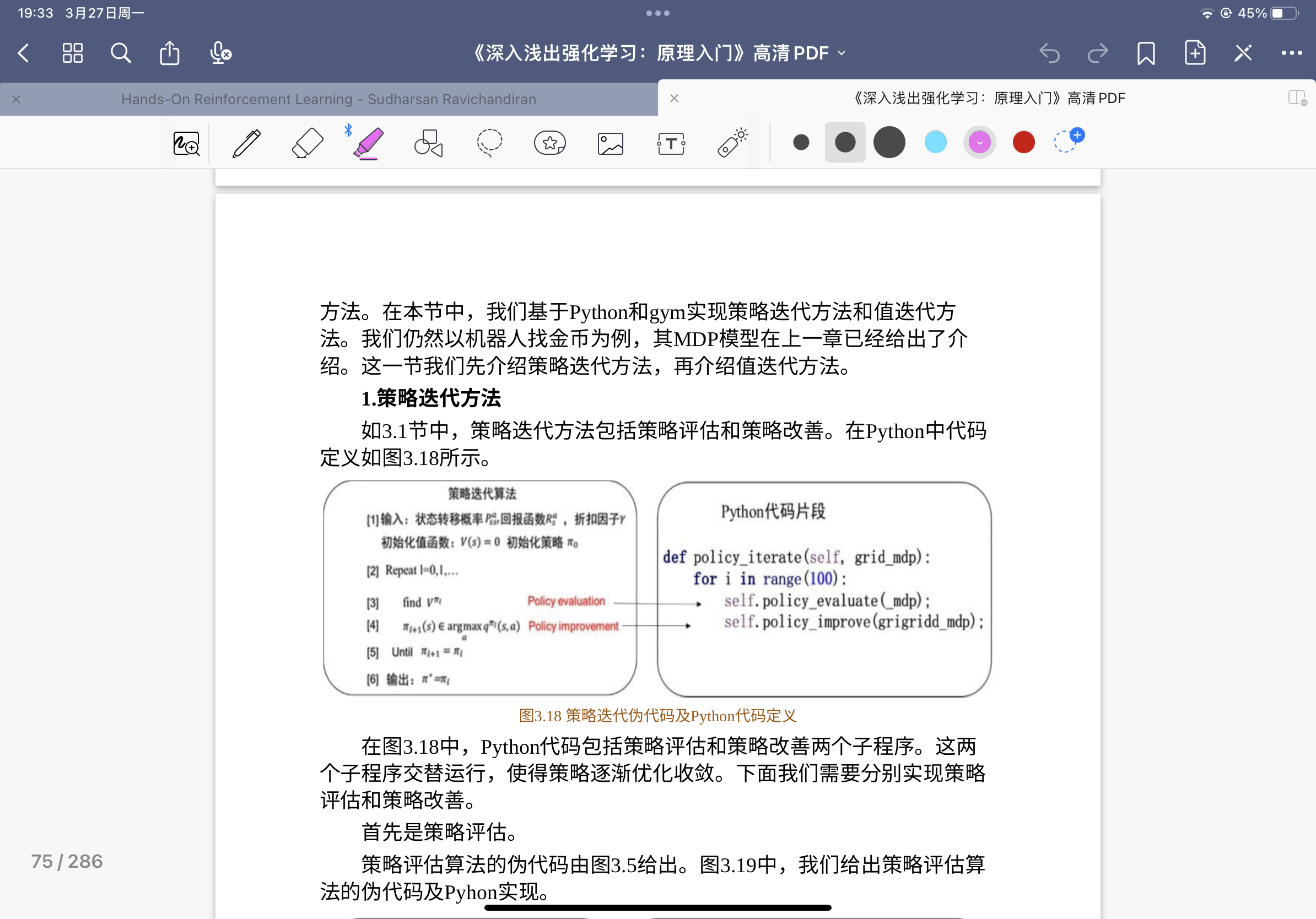


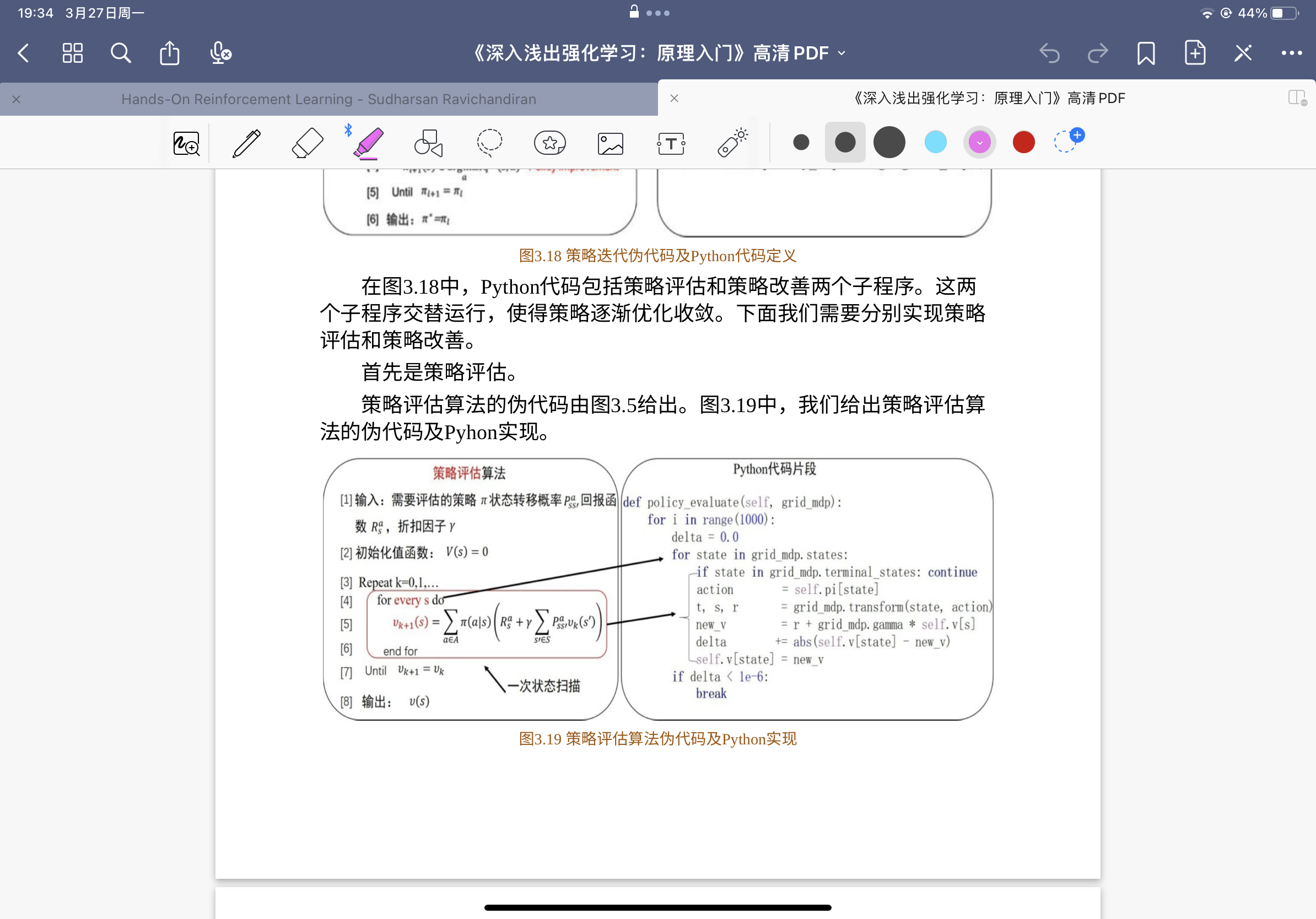
图3.13

需要注意的是在每次迭代过程中，需要对状态空间进行一次扫描，同时在每个状态对动作空间进行扫描以便得到贪婪的策略。

值函数迭代是动态规划算法最⼀般的计算框架，我们接下来阐述最优控制理论与值函数迭代之间的联系。解决最优控制的问题往往有三种思路：变分法原理、庞特里亚金最⼤值原理和动态规划的⽅法。三种方法各有优缺点。

基于变分法的⽅法是最早的⽅法，其局限性是无法求解带有约束的优化问题。基于庞特里亚金最⼤值原理的⽅法在变分法基础上进⾏发展，可以解决带约束的优化问题。相⽐于这两种经典的⽅法，动态规划的⽅法相对独立，主要是利用贝尔曼最优性原理。





### 3.2 动态规划中的数学基础讲解

#### 3.2.1 线性方程组的迭代解法。

利用（3.4）计算策略已知的状态值函数时，⽅程（3.4）为⼀个线性⽅程组。因此策略的评估就变成了线性⽅程组的求解。线性⽅程组的数值求解包括直接法（如高斯消元法，矩阵三角分解法，平⽅根法、追赶法等）和迭代解法。策略评估中采用线性⽅程组的迭代解法。

1. 何为迭代解法

不失⼀般性，用⽅程（3.17）表示⼀般的线性⽅程组。

所谓迭代解法是根据（3.17）式设计⼀个迭代公式，任取初试值，将其代⼊到设计的迭代公式中，得到，再将代⼊迭代公式中得到 ，如此循环最终得到收敛的X。

那么，根据（3.17）式如何设计迭代公式？

（1）⽅法⼀：雅克⽐（Jacobi）迭代法。 雅克比迭代法假设系数矩阵的对角元素。从（3.17）的第i个⽅程分离出，以此构造迭代⽅程：

⽅程（3.18）写成矩阵的形式为:

其中：

若记,则迭代公式为

在进行迭代计算时，(3.18)式变为

利用迭代公式（3.20）求解线性⽅程组（3.17）的⽅法称为雅克比迭代法。矩阵B称为迭代矩阵。

雅克⽐迭代法解线性⽅程很快，还能不能更快？

答案是肯定的。我们可以观察（3.21），不难发现第k+1次迭代计算分量时，分量都已经求出来了，但是在计算时，雅克⽐迭代⽅法没有利用这些新计算出来的值。如果这些新计算出来的值能够被利用，计算速度肯定会提高，这就是高斯-赛德尔迭代法。

2.高斯-赛德尔迭代法

当求得新的分量之后，马上用来计算的迭代算法称为高斯-赛德尔迭代法。对于线性⽅程组，对应于雅克⽐迭代过程（3.21）的高斯-赛德尔迭代过程为

用矩阵的形式可表示为

写成迭代方程为

其中

#### 3.2.2 压缩映射证明策略评估的收敛性

首先，我们先从数学上了解什么是压缩映射（Contraction Mapping）。ContractionMapping中⽂可以译为压缩映射或压缩映像。这个概念来自于数学中的泛函分析。内容涉及不动点理论。不动点和压缩映射常用来解决代数⽅程，微分⽅程，积分⽅程等，为⽅程解的存在性、唯⼀性和讨论迭代收敛性证明提供有⼒的⼯具。本⽂用来证明迭代的收敛性。

定义：设是度量空间，其度量用表⽰。映射，若存在a,，使得，则称T是X上的⼀个压缩映射；

若存在使得，则称是T的不动点。

定理1：完备度量空间上的压缩映射具有唯⼀的不动点。

定理1是说，从度量空间任何⼀点出发，只要满⾜压缩映射，压缩映射的序列必定会收敛到唯⼀的不动点。因此证明⼀个迭代序列是不是收敛，只要证明该序列所对应的映射是不是压缩映射。

我们已经知道了什么是压缩映射，那么策略评估是如何跟压缩映射扯上关系的呢？

回答这个问题，我们还要追根溯源，看看基于模型的策略评估到底是什么东西。前面已经提过，从数学的角度来看，若将值函数看成是未知数，值函数的求解其实是解⼀组线性⽅程。为讲述⽅便，我们在这里再次列⼀下：

对这个线性方程组，我们使用的方法是高斯-赛德尔迭代：

高斯-赛德尔解线性⽅程组迭代收敛条件是右面的未知数矩阵的谱半径⼩于1，这个条件也是由压缩映射定理得来的。在这里，我们不去求系数矩阵的谱，⽽是找⼀个特殊的度量以便简化证明，这个度量我们选为⽆穷范数，即

(只与元素中绝对值最大的元素有关)

从当前值函数到下一个迭代值函数的映射可表示为

下面，我们证明该映射是一个压缩映射。

证明

因为，所以是⼀个压缩映射，该迭代序列最终收敛到相应于策略π的值函数。上面是利用向量形式进⾏的推导，可能不是很直观，下面我们从值函数的定义出发进⾏证明。

值函数公式为:

此时映射函数为：

证明：

如果将值函数看成是⼀个向量，⽆穷范数是该向量中绝对值最大的那个。

### 3.4 最优控制与强化学习⽐较

当模型已知时，强化学习问题可转化为最优控制问题。本节我们给出最优控制的计算⽅法。一般而言，最优控制的数值计算⽅法分为间接法和直接法。其分类如图3.18所⽰。

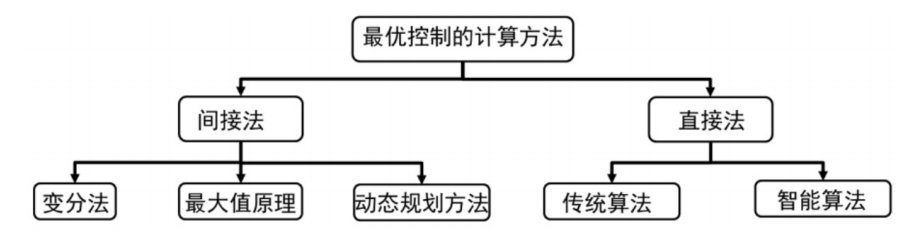


图3.18

1.什么是间接法

所谓间接法，是指首先利用变分法、最大值原理或者动态规划⽅法得到求解最优问题的一组微分⽅程（如本章3.1节利用动态规划的⽅法得到了⼀组偏微分⽅程），之后，利用数值求解⽅法求出此微分⽅程组的解，此解即为原最优问题的解。如本⽂介绍的微分动态规划的⽅法就属于间接法。

2.什么是直接法

直接法与间接法不同，它不需要首先利用最优控制理论（如变分原理，最大值原理或动态规划⽅法）得到一组微分⽅程，而是直接在可⾏控制集中搜索，找到最优的解。直接⽅法也分为两类，一类是将状态变量和控制变量参数化，将最优控制问题转化为参数优化问题；第二类是引⼊函数空间中的内积与泛函的梯度，将静态的优化⽅法推⼴到函数空间中。

在直接法中,最常用的是伪谱的⽅法。伪谱的⽅法是指在正交配置点处将连续最优控制问题离散化，通过全局差值多项式逼近状态量和输⼊控制量，直接将最优控制问题转化为⾮线性规划问题，再利用⾮线性规划问题的各种优化⽅法求解。常用的伪谱⽅法有Gauss（⾼斯）伪谱法、Legendre（勒让德）伪谱法、Radau（拉道）伪谱法和Chebyshev（切⽐雪夫）伪谱法。

最优控制⽅法在那些模型已知的序贯决策问题中已经取得了很好的结果。若是你面对的问题可以用精确的模型来描述，便可直接采用最优控制的⽅法。至于采用最优控制的哪种⽅法，可根据具体问题选用。

可能有⼈疑惑，这本书讲的是强化学习，怎么又讲开了最优控制，是不是跑题了？

没有跑题。

最优控制经过几十年的发展，已有很多优秀的成果。在模型未知的强化学习算法中，这些优秀的成果可直接拿来应用。如何应用？在基于模型的强化学习算法中，智能体会先利用交互数据拟合⼀个模型，有了模型，我们就可以利用最优控制的计算⽅法计算当前最优解，产生当前的最优控制率。智能体会利用当前的最优控制率与环境交互，进⼀步优化⾏为。结合最优控制的强化学习算法在机器⼈等领域取得了很多优秀的成果，最典型的是引导策略搜索的算法。本书第10章会详细介绍该⽅法。