

第十一章 线性方程组的求解及其在城市结构模拟中的应用

219

线性方程组应用很广，如常用的投入产出分析（Hewings, 1985；参见附录 11A 的简介）。本章以格瑞-劳韦模型(Grain-Lowry model)为例，介绍和讲解线性方程组的求解方法。格瑞-劳韦模型多用于城市内部用地结构分析，广受城市规划师和城市地理学家的青睐。本章通过一个假想的城市，探讨城市内部人口和就业分布的相互依存，研究二者的分布又是如何受交通网络和其他因素的影响。在本章的实例分析中，GIS 的应用涉及基于路网的通行时间计算，以及相关的数据准备工作。

线性方程组的求解是数字分析（numerical analysis）的基本算法，也是许多高级数字分析方法的基石，比如非线性方程组的求解和矩阵特征值的计算都要用到线性方程组的解法。附录 11B 简要讲解非线性方程组的求解，其中就用到了线性方程组的解法。

11.1 线性方程组求解

一个 n 元线性方程组有 n 个未知变量 x_1, x_2, \dots, x_n ，如下所示：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

上述方程组可用矩阵形式表示为：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

或简写为：

$$\mathbf{Ax} = \mathbf{b} \quad (11.1)$$

如果矩阵 \mathbf{A} 是对角矩阵，则方程组(11.1)变为：

220

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

对于这样简单的线性方程组，求解就很容易，其解为：

$$x_i = b_i / a_{ii} \quad (i = 1, 2, \dots, n)$$

如果 $a_{ii} = 0$ 并 $b_i = 0$ ，则 x_i 可以是任意实数；如果 $a_{ii} = 0$ 而 $b_i \neq 0$ ，则方程组无解。

还有两种线性方程组，其求解方法也比较简单。如果矩阵 \mathbf{A} 是个下三角形矩阵（即所有位于主对角线以上的元素都为 0），则方程组（11.1）式变为：

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

如果式中 $a_{ii} \neq 0$ ($i = 1, 2, \dots, n$)，上述方程组可用向前替代法求解。也就是说，通过第一个方程式可以求解 x_1 ；将 x_1 的值代到第二式，可求解 x_2 ；将 x_1 和 x_2 的值代入第三式，可求解 x_3 ；如次类推，可以求解其它变量。

类似地，如果矩阵 \mathbf{A} 具有上三角型的结构，即所有位于主对角线以下的元素都为 0，方程组（11.1）式变为：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

这种上三角型结构的方程组可以通过后向替代法来求解，其过程与向前替代法正好相反。

如果能将任何线性方程组转换为上述两种简单的线性方程组形式，也就是把矩阵 \mathbf{A} 转换成三角形矩阵，一般的线性方程组就可以得解了。操作上就是把矩阵 \mathbf{A} 分解成一个下三角形矩阵 \mathbf{L} 和一个上三角形矩阵 \mathbf{U} 的积（称为 LU 分解法），即 $\mathbf{A} = \mathbf{LU}$ 。其求解过程分为二个步骤：

(i) 先由 $Lz = b$, 求解 z ;

(ii) 再由 $Ux = z$, 求解 x 。

也就是说, 采用向前替代法可以从第一个方程组 ($Lz = b$) 中求出 z ; 然后将解得的 z 代入第二个方程组 ($Ux = z$), 采用后向替代法解得 x 。

实践中有多种 LU 矩阵分解法, 其中高斯法比较有效, 使用也广。高斯法由两个步骤组成, 第一步为分解, 或称向前消去, 第二步为求解, 用的是后向替代 (参见 Kincaid 和 Cheney, 1991, 第 145 页)。已经有许多现成的计算机程序, 用的都是高斯法, 如 FORTRAN 语言 (Press et al., 1992a)、C 语言 (Press et al., 1992b) 和 C++ 语言 (Press et al., 2002)。本书中的程序 SimuCity.for (见附录 11C) 是 FORTRAN 程序, 其中子程序 LUDCOMP 执行第一步, 子程序 LUSOLVE 执行第二步。这两个子程序都调用另外两个小的子程序 (SCAL 和 AXPY)。许多 FORTRAN 语言的编译软件可以免费从网站下载, 参见网络地址 <http://www.thefreecountry.com/compilers/fortran.shtml>, 作者使用的 g77 编译软件是从 <http://www.gnu.org/software/fortran/fortran.html> 下载的。本章第 3 节讲解这些程序如何应用于求解格瑞-劳韦模型。有条件的读者也可以用商用软件 MATLAB (www.mathworks.com) 或者 Mathematica (www.wolfram.com) 来求解线性方程组。

11.2 格瑞-劳韦模型

11.2.1 基本与非基本产业

研究城市问题的学者, 都熟悉城市内部人口与就业的分布是密切相关的。这里, 我们所说的“人口”, 是指居民住宅人口, 一般人口普查就是按住宅地址统计的; “就业”是指上班人员的就业地点, 是按工作地计算的。分析城市用地三大类 (商业用地、工业用地和住宅用地) 的分布其实最主要的就是要弄清楚城市人口与就业的分布, 因为人口代表了住宅用地、就业占用的是商业和工业用地。我们常常遇到的争论是: 究竟是人口跟随就业, 还是就业追随人口?

前者强调的是，就业区位定下来以后，就业人员设法就近寻找住宅，从而形成居民点；后者的逻辑是，居民点是先形成的，企业和公司只是设法办在邻近于居民区的地方，以求便捷地获取劳动力资源或便于服务当地居民。其实现实生活中，可能两种情况都存在。

格瑞-劳韦模型（Lowry, 1964; Garin, 1966）强调城市人口与就业的分布相互依存，但就业有基本就业与非基本就业之分，不同类型的就业影响居民人口分布所起的作用不同。基本就业的分布相对独立，可视为外在因素；服务性就业（或称非基本就业）的分布追随居民人口的分布。另外，居民人口的分布总是受就业分布的影响，无论是基本就业还是非基本就业。图 11.1 概括了它们之间的这种相互依存的关系。人口与就业分布的这种互相影响随其间的距离递减，而距离又是由城市交通网络所决定的。回顾第六章（特别是附录 6A）讨论过的城市经济模型（如米尔斯-穆斯模型），假定的是单中心，也就是所有就业机会都集中在市中心。格瑞-劳韦模型不同，具有很大的灵活性。在模拟城市人口与就业分布时，模型可以假定基本就业都集中在市中心，也可是其它分布，因此而检验就业对人口分布的影响。同时，该模型还可用于检验交通路网对人口和就业分布的影响。

222

将就业分为基本就业和非基本就业，源于城市或区域的经济可分为基本产业与非基本产业。前者指某一部门的产品或服务是面向市区以外的地区，是相对独立于当地经济的；后者指产品或者服务面向当地，销于当地，是依附于当地经济的。据此类推，基本就业就是指在基本产业部门的就业人口，非基本就业是指在非基本产业部门就业的人口。这种两分法是城市与区域规划中的一种有效手段（Wheeler et al., 1998，第 140 页），它明确指出了哪些经济部门对于城市的生存发展是至关重要的。由于基本产业的发展会带动整个城市经济包括非基本产业的发展，城市与区域规划师也常常从分析基本产业的变化入手，来预测整个城市的经济变迁。

区分基本和非基本就业的常用方法是“最小需求法”（Ullman and Dacey, 1962）。最小需求法是先把同等规模的城市归为一类，计算每个城市中各个部门的就业人口比重。对于某一产业，在同类城市中找到该产业比重最低的那个城市，这个比重就是这一产业的最低需求。最低需求可以近似为非基本就业，也就是服务于城市本身的就业数，多出的部分就是基本就业。

区分基本与非基本就业还可以通过分析出口资料 (Stabler and St. Louis, 1990) 来划分。实际操作中也可以粗略地把某一部门 (如工业) 全划为基本产业、某一部门 (如服务业) 全划为非基本产业。

这样看来, 格瑞-劳韦模型中分析的基本就业、非基本就业和居民人口就对应于城市用地的三大类 (工业用地、商业用地和住宅用地), 基本上解决了城市用地的分布问题。

11.2.2 格瑞-劳韦模型构建

在格瑞-劳韦模型中, 假设一个城市由 n 个区组成, 任何一个区如 j 区的人口分布受全市所有 n 个区的就业 (包括基本与非基本) 影响; 任何一个区如 i 区的服务 (非基本) 就业的分布由全市所有 n 个区的人口决定。两者之间相互影响的强度随距离递减, 由重力模型量化表达。格瑞-劳韦模型只需知道基本就业的分布和各区之间的距离 (时间), 就可模拟人口与非基本产业的分布情况。

首先, 任何 i 区服务就业 S_i 是全市每个区 ($k = 1, 2, \dots, n$) 人口 P_k 的影响之和, 其影响强度由重力轴系数 t_{ik} 调控:

$$S_i = e \sum_{k=1}^n (P_k t_{ik}) = e \sum_{k=1}^n [P_k (d_{ik}^{-\alpha} / \sum_{j=1}^n d_{jk}^{-\alpha})] \quad (11.2) \quad 223$$

式中常数 e 是全市服务就业人口与总人口之比 (用于平衡方程的左右两边), d_{ik} 是 i 区和 k 区之间的距离, α 是距离衰减系数 (用来刻画人们从居民点到服务业点的购物行为)。重力轴系数 t_{ik} 代表 i 区服务就业数受 k 区居民人口数影响的那一部分比例, 而 k 区居民人口又影响所有区的服务业。换言之, i 区服务就业数是每个 k 区 ($k = 1, 2, \dots, n$) 人口对其影响 (随距离递减) 的总和; 而反过来, 其中这个 k 区人口对 i 区服务就业的影响, 又仅仅是这个区对其它所有 j 区 ($j = 1, 2, \dots, n$) 服务就业影响的一部分。

类似地, 任何 j 区的人口 P_j 反过来是全市每个 i 区 ($i = 1, 2, \dots, n$) 就业数 E_i 的影响之和, 其影响强度也是由一个重力轴系数 g_{ij} 测算的。

$$P_j = h \sum_{i=1}^n (E_i g_{ij}) = h \sum_{i=1}^n [(B_i + S_i)(d_{ij}^{-\beta} / \sum_{k=1}^n d_{kj}^{-\beta})] \quad (11.3)$$

式中常数 h 是全市人口与就业数之比, β 是距离衰减系数 (刻画人们从居民点到就业点的通勤行为)。注意, 就业 E_i 包括基本和服务就业人员 ($E_i = S_i + B_i$)。同样地, 重力轴系数 g_{ij} 代表 j 区居民人口受 i 区就业影响的那一部分比例, 而这部分影响又只是 i 区总就业数影响所有 k ($k = 1, 2, \dots, n$) 区居民人口的一部分。

以数列 \mathbf{P} 、 \mathbf{S} 、 \mathbf{B} 代表 P_j 、 S_i 、 B_i , 以矩阵 \mathbf{G} 、 \mathbf{T} 代表 g_{ij} (含常数 h) 和 t_{ik} (含常数 e), 方程 (11.2) 和 (11.3) 可写为:

$$\mathbf{S} = \mathbf{TP} \quad (11.4)$$

$$\mathbf{P} = \mathbf{GS} + \mathbf{GB} \quad (11.5)$$

合并(11.4)和(11.5), 重新整理为:

$$(\mathbf{I} - \mathbf{GT})\mathbf{P} = \mathbf{GB} \quad (11.6)$$

式中 \mathbf{I} 为 n 阶单位矩阵。(11.6)式就是一个 n 元线性方程组的矩阵形式, 未知数是矩阵 \mathbf{P} 。由于四个参数 (两个距离衰减系数 α 和 β 、人口与就业比 h 、服务就业与人口比 e) 是给定的常数, 距离矩阵 d 可以通过路网数据估算, 矩阵 \mathbf{G} 、 \mathbf{T} 就定义了; 基本就业数列 \mathbf{B} 代表市内基本就业的空间分布状态, 需要事先给定。

将 (11.6) 式的解 \mathbf{P} 代入到 (11.4) 式, 即可解出服务就业数列 \mathbf{S} 。也就是说, 给定市内基本就业的分布和各区之间的距离, 格瑞-劳韦模型就可以求得人口与服务产业的分布情况。参见有关文献 (Batty, 1983)。

下一节采用一个简单的实例, 讲解格瑞-劳韦模型的构建和求解过程。

11.2.3 简明实例

224

如图 11.2 所示, 假设城市由 5 个 ($n = 5$) 具有相同面积的区块组成, 图中虚线为连接各区道路。可以看出, 如果基本就业都集中在市中心(1 区), 从区位的角度来看, 其它区块

(2、3、4、5) 都是对称的, 也必然有同样的人口和服务就业数。这样, 只有 1 区和 2 区(图中阴影部分)需要区别, 设其人口分别为 P_1 、 P_2 , 非基本(服务)就业为 S_1 、 S_2 。假设基本就业总数标准化为 1, 那么, $B_1 = 1, B_2 = B_3 = B_4 = B_5 = 0$ 。这样标准化表示我们关心的是人口和就业分布在各区的相对人数, 也就是它们的地域变化。假设 1、2 两区距离为 1, 每区内部距离为 0.25 (即 $d_{11} = d_{22} = d_{33} = \dots = 0.25$), 区间距离以路网距离计算 (比如 $d_{23} = d_{21} + d_{13} = 1 + 1 = 2$)。为分析方便, 定义常数为: $e = 0.3$ 、 $h = 2.0$ 、 $\alpha = 1.0$ 、 $\beta = 1.0$ 。

据(11.2)式, 利用区位对称的特点 (即区 2、3、4 和 5 相对于区 1 来说区位等同), 则有:

$$S_1 = 0.3 \left(\frac{d_{11}^{-1}}{d_{11}^{-1} + d_{21}^{-1} + d_{31}^{-1} + d_{41}^{-1} + d_{51}^{-1}} P_1 + \frac{d_{12}^{-1}}{d_{12}^{-1} + d_{22}^{-1} + d_{32}^{-1} + d_{42}^{-1} + d_{52}^{-1}} P_2 * 4 \right)$$

代入距离值计算得到:

$$S_1 = 0.1500P_1 + 0.1846P_2 \quad (11.7)$$

同样地有:

$$S_2 = 0.3 \left(\frac{d_{21}^{-1}}{d_{11}^{-1} + d_{21}^{-1} + d_{31}^{-1} + d_{41}^{-1} + d_{51}^{-1}} P_1 + \frac{d_{22}^{-1}}{d_{12}^{-1} + d_{22}^{-1} + d_{32}^{-1} + d_{42}^{-1} + d_{52}^{-1}} P_2 + \right. \\ \left. \frac{d_{23}^{-1}}{d_{13}^{-1} + d_{23}^{-1} + d_{33}^{-1} + d_{43}^{-1} + d_{53}^{-1}} P_3 * 2 + \frac{d_{24}^{-1}}{d_{14}^{-1} + d_{24}^{-1} + d_{34}^{-1} + d_{44}^{-1} + d_{54}^{-1}} P_4 \right)$$

225

其中 3 区和 5 区相对于 2 区来说具有相同的区位。注意到 $P_4 = P_3 = P_2$, 则上式可以简化为:

$$S_2 = 0.0375P_1 + 0.2538P_2 \quad (11.8)$$

同理, 据(11.3)式可以得到以下两个方程:

$$P_1 = S_1 + 1.2308S_2 + 1 \quad (11.9)$$

$$P_2 = 0.2500S_1 + 1.6923S_2 + 0.25 \quad (11.10)$$

方程(11.7)-(11.10)组成一个四元一次线性方程组, 计算得到: $P_1 = 1.7472, P_2 = 0.8136; S_1 = 0.4123, S_2 = 0.2720$ 。从中可以看出, 人口和服务就业都是从市中心区向周边地区递减。

11.3 案例研究：基于一个假想城市的人口与服务就业分布模拟

如图 11.3 所示，假定城市路网由 10 个同心环路和 15 条放射状道路组成，所有区块围绕城市唯一的中心——中心商业区（CBD），整个城市被路网分割为 $1+9\times 15 = 136$ 个区。为了便于计算，假定任何除 CBD 外的区，其交通进出口就是放射状道路与环线的交点。这样一来，任何非 CBD 的区在空间上（GIS 中）都可以用这个点（环线与放射状线的交点）来表示，而不是传统的几何中心来表示。这都是为了简化各区之间交通距离或时间的计算，因为这些点正好是交通路网的结点，结点之间的交通距离就直接成了城市各区之间的交通距离。有兴趣的读者可以象第二章第 2.3.2 节讲到的那样，用各区的几何中心来表示各区的位置，寻找离该中心点最近的路网结点，再计算这些结点之间的路网距离，加上结点到起点区、结点到终点区的两段直线距离，三段总和成起点区到终点区的交通距离，计算复杂一些，但结论基本一致。这个假想的的城市没有具体的地理坐标系，距离或时间的值也就没有单位。

案例研究所需的数据集如下：

1. 一个包含有 136 个区块的城市多边形图层，名称为 `tract`。
2. 从多边形图层 `tract` 中抽取生成的交通路网图层，含有线和结点的拓扑关系，名称为 `road`。
3. 从交通路网图层的结点中抽取并生成的一个点图层，名称为 `trtp`，用来代表 135 个非 CBD 区块。
4. 一个名称为 `cbd` 的点图层（只有一个点，用来表示 CBD 区）。

路网距离（时间）的计算主要是利用点图层 `trtp` 和路网图层 `road` 来完成的。路网图层的属性数据文件 `road.aat` 中包括二项属性参数：参数 `length` 就是道路长度，用来定义路网的交通阻滞；而在参数 `length1` 中，除第 7 条环线 `length1` 的值被定义为实际长度的 1/2.5 外，其它值与参数 `length` 的值一样。定义这个新的交通阻滞参数 `length1`，是用来检验郊区环路（比如第 7 条环线）如果是高速公路的影响。也就是，这条环线的车速是其它路上的 2.5 倍时，其交通阻

滞就只有其它道路上的 1/2.5。点图层 `trtpt` 和 `cbd` 的属性文件中含有 `trtid` 和 `trt_perim` 两项属性参数，其中 `trtid` 为每个区的标识码，`trt_perim` 为每个区块的周长。区块周长将用于估算“区内交通距离”，定义为区块周长的 1/4。定义区内交通距离有两个目的，一是当起点区和终点区是同一区块时，其间距离不是 0（重力模型中距离不能是 0）；二是以点代面时，低估了区块之间的平均距离。下里我们计算各区块俩俩之间的交通距离时，总是先算点点间的路网距离，再把起点区和终点区的两段区内交通距离加上去。

11.3.1 任务 1：利用 ArcGIS 计算交通路网距离（时间）

如果所有路段的通行速度一样，时间也就等同于距离，路程距离代表了通行阻滞。由于采用路网结点来代表区块的位置，各区块俩俩之间的交通距离可以方便地在 ArcGIS 中计算出来。下面概略地列出了主要步骤，详细的操作指南参见第二章的第 2.3 节。 227

- (1) 计算非 CBD 区间的交通距离。首先计算起点区块（`trtpt`）到终点区块（也是 `trtpt`）通过路网图层 `road` 的路网距离；然后加上起点区和终点区的内部交通距离。
- (2) 计算 CBD 区到非 CBD 区之间交通距离。先是 CBD 区到非 CBD 区之间经过放射状道路的交通距离（就是其间的欧氏几何距离）；再加上起点区、终点区的区内交通距离。
- (3) 计算 CBD 区内的交通距离，即 CBD 区块周长的 1/4。

通过上述三步可以计算出所有区块俩俩间的距离，计算结果整合为 `odtime.txt` 的文本文件，共有 $136 \times 136 = 18,496$ 条记录，每个记录含有三个变量，即起点区块 ID，终点区块 ID，区间距离。将文件 `odtime.txt` 中的记录按起点区块 ID 和终点区块 ID 进行排序，中间用空格分隔，另存为 `odtime.prn` 文件。为了方便读者，随本书而带的 CD 光盘中提供了程序 `rdtime.aml` 和程序运行结果文件 `odtime.prn`，其中 `rdtime.aml` 是用来计算区块间交通距离的 AML 程序。

利用路网图层中的另一个属性参数 `length1` 来代替 `length`，重复上面的操作，可以得到一个新的文件 `odtime1.txt`。同样地，将这个文件按起点 ID 和终点 ID 进行排序生成一个

odtime1.prn 文件（CD 中也含有这个文件）。在这里，length1 考虑到了第 7 条环线的速度要比其它道路快，结果相当于用时间而不是距离来定义路网阻滞。

11.3.2 任务 2：模拟基本案例中人口和服务就业分布

我们设计一个基本案例，作为后面其它例子的对比对象。除了所有道路的通行速度一样外，基本案例还假定城市是单中心的，所有基本就业（比如 100 人）都集中在 CBD，同时，假定重力模型中的两个距离衰减系数 $\alpha = 1.0$ 和 $\beta = 1.0$ 。据美国人口普查局(Bureau of Census, 1993)的统计资料，模型中的 h 和 e 大致定为 $h = 2.0$ 、 $e = 0.3$ 。如果 P_T 、 B_T 、 S_T 分别是该市总人口数、总基本就业人数、服务就业人数，则可以得到：

$$S_T = eP_T$$

$$P_T = hE_T = h(B_T + S_T)$$

因此有

$$P_T = (h/(1-he))B_T$$

已知 B_T 为标准化值 100，就有 $P_T=500$ 、 $S_T=150$ 。在其后别的模拟例子时，我们保持 h 、 e 和 B_T 不变，也就是总人口和就业（包括基本与服务就业）保持不变，但是变化参数 α 和 β 的取值，改变基本就业的空间分布格局和交通路网（如市郊高速环路的修建），研究这些因素的影响。

附录 11C 中的 FORTRAN 程序 simucity.for（光盘中）流程主要有两步。首先，读取交通距离（时间）矩阵数据文件 odtime.prn；然后，利用 LU-分解法解格瑞-劳韦模型；最后，输出分析结果（人口和服务就业）到一个外部文件 basic.txt。由于同一个环上的区结果相似或是相同，表 11.1 中仅仅列出了沿放射状道路、由里到外的 10 个区的信息。图 11.4 和 11.5 分别表示人口和服务业从 CBD 往外的分布变化。

11.3.3 任务 3：检验基本就业分布的影响

229

上述的基本案例假定基本就业全部集中在市中心，现假定基本就业的分布平均分配在各个区，再利用模型测算人口和服务就业的分布情况。在本案例中，所有区块具有相同的基本就业数量，即 $100/136 = 0.8088$ 。只需将程序 `simucity.for` 中给基本就业分布赋值的代码修改一下，重新运行程序就可以获得新的分析结果。见表 11.1 和图 11.4、图 11.5。不难发现，即使基本就业在空间上是均匀一致的，人口和服务就业两者都是从市中心向外递减的。这说明，人口和就业从 CBD 往外随距离递减的趋势，反映了区位优势递减，这主要是因为交通路网的格局促成了 CBD 的易达性，而不只是因为 CBD 地区就业机会集中所致。当然 CBD 地区就业机会的集中自然会强化这种从市中心向外递减的效果。从图 11.4、图 11.5 中可以看出，基本就业均匀分布与全集中在 CBD 的基本案例相比，测算出的人口和服务就业的分布曲线都稍微缓一点。由此可见，服务就业的分布紧随人口分布，二者分布的趋势是相似的。

有兴趣者可以设计不同的基本就业分布模式，利用格瑞-劳韦模型分析检验其对人口和服务就业分布的影响。参见有关文献（Guldmann and Wang, 1998）。

11.3.4 任务 4：检验距离衰减系数的影响

保持基本案例中的所有参数不变，现在仅仅改变距离衰减系数 α 和 β 。基本案例中 $\alpha = 1$ 、 $\beta = 1$ ，在这里取 $\alpha = 2$ 、 $\beta = 2$ 。距离衰减系数表示人们出行（包括工作和购物）受交通距离（时间）的影响程度。当交通技术提高、路网更完善时，这两个系数通常会下降。在这里将 α 和 β 这二个参数设为 2，比起基本案例来，可以看作是城市发展更初期的交通条件。

230

修改程序 `simucity.for` 中相关的赋值语句，输入新的 α 和 β 值，生成新的人口和服务就业分布，结果也显示在表 11.1、图 11.4 和图 11.5 中。对应于较大的 α 和 β ($=2$)，曲线陡一些；对应于较小的 α 和 β ($=1$ ，也就是基本案例)，曲线缓一些。这个模拟结果很重要，印证

了第六章所述的人口密度方程随时间递变的趋势：随着时间的向前发展（距离衰减系数变小），城市人口密度从中心到边缘递减的递度趋于平缓。

11.3.5 任务 5：检验交通路网的影响

最后让我们来检验交通路网的影响。我们用这个案例分析郊区环线的影响，假定第 7 条环线是高速公路。也就是说，如果一般的城市道路的通行速度为 30 英里/小时，那么该环线是 75 英里/小时。在任务 1 中已经基于这个案例生成了另一个交通时间的文件 `odtime1.prn`。改变 `simucity.for` 程序中的输入文件名，将 `odtime.prn` 变为 `odtime1.prn`，运行程序可以生成新的计算结果，结果同样显示在表 11.1、图 11.4 和图 11.5 中。从中可以看出，人口和服务就业分布结果与基本案例相似，但比基本案例中的曲线稍稍平缓些（注意 CBD 区附近的值比基本案例的低一点，而最边缘区块的值比基本案例又高一点）。也就是说，郊区高速环线的建造，减小了 CBD 比起郊区来的区位优势，进而导致人口和服务就业分布的递度变缓。有关更多郊区高速环线影响的分析，参见有关文献（Wang, 1998）。

11.4. 讨论与总结

将整个经济分为基本与非基本产业，强调的是这两种产业在城市和区域经济发展中扮演的不同角色。格瑞-劳韦模型利用这个概念来刻画城市中人口和就业分布之间的相互作用。模型中，基本（出口导向）就业的分布被看作为一个相对独立的外在因素，而服务（内需导向）就业的分布则依赖于人口分布；反过来，人口的分布又受就业（包括基本与非基本就业）的影响。两者之间相互作用，其影响随着通行距离(时间)而衰减，衰减程度由重力模型量化。基于这些思想构建的格瑞-劳韦模型，就是一个线性方程组，求解比较简便。给定基本就业的分布，现成的交通路网就可定义市区各地段的通行距离(时间)，我们由此就可以算出市区内人口和服务就业的分布。

将格瑞-劳韦模型应用于实际城市用地分析时，需要就业、人口和路网分布的数据，美国城市这方面的数据都比较详尽。但是，要将就业分为基本和非基本两个部分，许多经济活动很难分清基本与非基本。通常一个城市的大多数经济活动既服务于城市本身（非基本），又服务于外部（基本），这点成为模型应用的最大难点。本章个案研究中采用一个假想城市，研究了基本就业分布、距离衰减系数、交通路网变化等要素的影响，有助于我们理解城市结构的变化机制，帮助解释了根据实证分析发现的城市人口密度方程变化的一些趋势。

231

例如，城市基本就业分布存在郊区化的趋势，导致人口分布的郊区化，同时服务业也往郊区分散，但是，这种分散不会改变总体上人口和就业还是在 CBD 附近最集中。随着交通技术和路网效率的提高，通行速度加快，人们用同样的时间可以抵达更远的地段，传统意义上城市中心与郊区的易达性差距在缩小，导致城市人口密度从中心向周边递减的坡度变缓。模拟结果解释了在城市人口密度方程中递度系数逐年趋缓的现象。郊区高速环线最初是为方便大都市间的联系而设计的，目的是促进城市间的交通便利，可结果是更帮助了城市内部的交通联系 (Taaffe et al., 1996, 第 178 页)。案例中的模拟结果已经显示，郊区高速环线促使人口分布变缓。研究表明（参见 Wang, 1998, 第 274 页），即使基本就业仍然集中在 CBD 区，只要多增加郊区高速环线，人口郊区化趋势就会加强，甚至会在环路附近出现局部人口密度峰值。

格瑞-劳韦模型还可以用于检验城市结构研究中的更多问题。例如，通过比较不同城市的人口分布，分析大城市是否比小城市具有更为平缓的人口密度递减梯度 (McDonald, 1989, 第 380 页)；通过输入更密集的城市环路和放射状道路，分析路网密度的影响；通过设计不同的路网格局（如方格路网和半圆形路网），可以研究路网结构的影响。

附录 11A. 投入—产出模型

投入—产出模型被政府各级部门广泛使用于分析经济发展与规划。模型中，每一部门的产出又成为所有部门（包括这个产出部门自身）的投入，每一部门的投入也是来源于所有部门（包括这个投入部门自身）的产出。模型的关键假设是：连接各部门投入—产出关系的系数，

标志着某个时期的经济结构和生产技术，在相当一段时期内不变。投入—产出模型常见的应用问题有：某一部门产量的调整变化，将如何影响其他部门的生产？如何调整所有部门的生产，以适应市场需求的变化？

我们用一个只有两个工业部门的简单例子来讲解投入—产出模型。假设整个经济由汽车和钢铁工业两部门组成， X_1 为汽车工业的产量， X_2 为钢铁工业的产量。对于每个单位汽车产出，用于汽车工业自身的那部分消耗为 a_{11} ，用于钢铁工业的消耗为 a_{12} 。也就是说，汽车总产出 X_1 中，汽车工业自身消耗为 $a_{11}X_1$ ，用于钢铁工业的消耗为 $a_{12}X_2$ 。除此之外，最终销往市场的汽车为 d_1 。同样的，钢铁产量 X_2 也有三种去向： $a_{21}X_1$ 表示用于汽车工业的消耗量， $a_{22}X_2$ 是用 232
于钢铁工业自身的消耗量， d_2 是最终销往市场的钢铁销售量。数学模型如下：

$$\begin{cases} X_1 = a_{11}X_1 + a_{12}X_2 + d_1 \\ X_2 = a_{21}X_1 + a_{22}X_2 + d_2 \end{cases}$$

其中 a_{ij} 就是投入产出系数。

上述模型写成矩阵形式，就是：

$$\mathbf{IX} = \mathbf{AX} + \mathbf{D}$$

其中 \mathbf{I} 是单位矩阵，移项后重新将方程改写为：

$$(\mathbf{I} - \mathbf{A})\mathbf{X} = \mathbf{D}$$

上式就是一个线性方程组，只要给定最终需求 \mathbf{D} 和投入-产出系数 \mathbf{A} ，即可以求解方程组，得到各部门的产量 \mathbf{X} 。

附录 11B. 求解非线性方程组

首先，我们介绍单个非线性方程的解法——牛顿法。比方说，函数 f 是一个非线性函数，寻找满足 $f(x) = 0$ 的 x 值就是非线性方程的求解。假如 r 为方程未知的实数解，并设 x 为暂时找到的接近这个解 r 的值。用泰勒式展开 $f(x+h)$ ，但仅保留泰勒展开式的线性项作为近似值，我们可以得到：

$$0 = f(r) = f(x+h) \approx f(x) + hf'(x) \quad (\text{B11.1})$$

这里 h 是一个微小的增量，其值为 $h = r - x$ 。由 (B11.1) 式可得：

$$h \approx -f(x)/f'(x)$$

如果 x 接近方程的解 r ，则 $x - f(x)/f'(x)$ （也就是 x 加上微小增量 h ）将更接近 r 。牛顿法就是利用这个微小增量 h 的定义，逐步逼近求解的。开始时，取一个初值 x_0 作为方程的解，代入 $x - f(x)/f'(x)$ 作为方程新的解，用这个新的解计算方程 $f(x)$ 的值，然后判断这个新的值是否在求解误差以内（很接近于 0）。若是，则求解完毕，否则用这个新的解代替前一个进行迭代计算。迭代算法的数学定义如下：

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

设定合适的初值(x_0)，对成功地利用牛顿法求解方程至关重要，该值必须足够接近方程的真实解(Kincaid and Cheney, 1991, 第 65 页)，另外，牛顿法要求函数的一阶导数必须存在。这个方法可以扩展到非线性方程组。

假定方程组有两个变量，如下式：

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases} \quad (\text{B11.2})$$

类似于(B11.1)式，选取泰勒展开式的线性项部分有：

$$\begin{cases} 0 = f_1(x_1 + h_1, x_2 + h_2) \approx f_1(x_1, x_2) + h_1 \frac{\partial f_1}{\partial x_1} + h_2 \frac{\partial f_1}{\partial x_2} \\ 0 = f_2(x_1 + h_1, x_2 + h_2) \approx f_2(x_1, x_2) + h_1 \frac{\partial f_2}{\partial x_1} + h_2 \frac{\partial f_2}{\partial x_2} \end{cases} \quad (\text{B11.3})$$

把上式中的 h_1 、 h_2 看作变量，其它为常数，(B11.3) 就是一个二元一次线性方程组，其系数矩阵是函数 f_1 和 f_2 的雅可比矩阵，即：

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

这样的话，非线性方程组(B11.2)的牛顿替代法就可定义为：

$$\begin{aligned} x_{1,n+1} &= x_{1,n} + h_{1,n} \\ x_{2,n+1} &= x_{2,n} + h_{2,n} \end{aligned}$$

这里，增量 $h_{1,n}$ 和 $h_{2,n}$ 是线性方程组(B11.3)的解，也就是：

$$\begin{cases} h_{1,n} \frac{\partial f_1}{\partial x_1} + h_{2,n} \frac{\partial f_1}{\partial x_2} = -f_1(x_{1,n}, x_{2,n}) \\ h_{1,n} \frac{\partial f_2}{\partial x_1} + h_{2,n} \frac{\partial f_2}{\partial x_2} = -f_2(x_{1,n}, x_{2,n}) \end{cases}$$

或者

$$J \begin{bmatrix} h_{1,n} \\ h_{2,n} \end{bmatrix} = - \begin{bmatrix} f_1(x_{1,n}, x_{2,n}) \\ f_2(x_{1,n}, x_{2,n}) \end{bmatrix} \quad (\text{B11.4})$$

利用第 11.1 节中介绍的线性方程组解法，给定一对初始值 $x_{1,n}$ 和 $x_{2,n}$ ，求解方程组(B11.4)

234

得到增量 $h_{1,n}$ 和 $h_{2,n}$ ，然后利用牛顿替代法逐步求解非线性方程组(B11.2)。较多变量的非线性方程组的解法采用同样的策略，不同之处在于雅可比矩阵需要扩展。例如，对于一个三元非线性方程组，其雅可比矩阵为：

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}$$

由此可见，线性方程组的求解的确是非线性方程组求解的基石。

附录 11C. 求解格瑞-劳韦模型的 FORTRAN 程序

```

*****
*      SimuCity.FOR simulates the distributions of population      *
*      and service employment given a basic employment pattern    *
*      using the Garin-Lowry model:      (I-A*B)*POPU=A*BEMP.      *
*      Version on 4/26/2005, by Fahui Wang.                      *
*****

*      Variables defined:
*      ALPHA, BETA: distance friction coefficeints
*      H: population / employment ratio
*      E: service employment / population ratio
*      N: total number of tracts the city is divided into
*      D(i,j): distance between tracts i and j
*      A(i,j), B(i,j): matrices G, T in the Garin-Lowry model
*      BEMP(i): basic employment vectors [known]
*      POP(i), SEMP(i): population & service employment vectors
*                      [variables to be solved]
*      other variables: intermediates for computational purposes,
*                      defined where they appear first

      PARAMETER (N=136)
      DOUBLE PRECISION D(N,N),A(N,N),B(N,N),DOMA(N),DOMB(N),IAB(N,N)
      DOUBLE PRECISION BEMP(N),POP(N),SEMP(N),AA(N,N),BB(N)
      REAL ALPHA,BETA,H,E
      INTEGER IPVTV(N),INFO,I,J,OZONE(N),DZONE(N)

*      Step 1. Define parameters & Input data

*      Input the values of ALPHA, BETA, H, E
      DATA ALPHA /1.0/
      DATA BETA /1.0/
      DATA H /2.0/
      DATA E /0.3/

*      Input the distribution of Basic Employment
*      In the basic case, all employment (100) is assumed to be at
CBD
      DO 1 I=1,N
1      BEMP(I)=0.0
      BEMP(1)=100.0

*      Input the distance matrix, pre-sorted by OZone & DZone
      OPEN (2, FILE='c:/g77/ODTIME.PRN', STATUS='OLD')
      DO 10 I=1,N
      DO 10 J=1,N
      READ (2,*) OZONE(I), DZONE(J), D(I,J)
10      CONTINUE
      CLOSE(2)

*      Step 2. Build matrices A, B, I-A*B, based on D(i,j)

```

```

*      Derive matrix Aij, Bij first

      DO 20 I=1,N
*        DOMA(i) & DOMB(i) are the dominators in matrix formula.
        DOMA(I)=0.0; DOMB(I)=0.0
      DO 20 J=1,N
        DOMA(I)=DOMA(I)+D(J,I)**(-BETA)
        DOMB(I)=DOMB(I)+D(J,I)**(-ALPHA)
20    CONTINUE
      DO 30 J=1,N
      DO 30 I=1,N
        A(J,I)=H*D(J,I)**(-BETA)/DOMA(I)
        B(J,I)=E*D(J,I)**(-ALPHA)/DOMB(I)
30    CONTINUE

*      Derive the matrix I-A*B, represented by IAB here

      DO 70 I=1,N
      DO 70 J=1,N
        IAB(I,J)=0.0
      DO 80 M=1,N
        IAB(I,J)=IAB(I,J)+A(I,M)*B(M,J)
80    CONTINUE
        IF (I.EQ.J) THEN
          IAB(I,J)=1.0-IAB(I,J)
        ELSE
          IAB(I,J)=-IAB(I,J)
        ENDIF
70    CONTINUE

*      Step 3. Prepare the data for solving the model.
*      Since it is a given-employment problem, in the
*      system of linear equations: AA*X=BB, AA is IAB,
*      BB is A*BEMP, and X is POP.

      DO 90 I=1,N
        BB(I)=0.0
      DO 90 J=1,N
        BB(I)=BB(I)+A(I,J)*BEMP(J)
        AA(I,J)=IAB(I,J)
90    CONTINUE

*      Step 4. Solve the system of linear equation, AA*X=BB

*      Factor AA matrix, print out INFO (indicating if the solution
*      exists).

      CALL LUDCOMP(AA,N,N,IPVT,INFO)
      PRINT *,INFO

*      Solve for x, which is vector POP

      CALL LUSOLVE(AA,BB,N,N,IPVT)

```

```

        DO 100 I=1,N
        POP(I)=BB(I)
        SEMP(I)=0.0
100    CONTINUE

*      Solve for the vetcor SEMP (=B*POP)
        DO 200 I=1,N
        DO 200 J=1,N
            SEMP(I)=SEMP(I)+B(I,J)*POP(J)
200    CONTINUE

*      Step 5. Output the results

        OPEN(12,FILE='Basic.TXT')
        DO 500 I=1,N
        WRITE(12,501) I,OZONE(I),BEMP(I),POP(I),SEMP(I)
501    FORMAT(1X,2(1X,i4),3(1X,f12.6))
500    CONTINUE
        CLOSE(12)

        STOP
        END

        subroutine scal(c,x,n)
c      =====
c      Scales a vector by a constant
        integer i,n
        double precision c,x(*)
        do 10 i=1,n
            x(i) = c*x(i)
10    continue
        return
        end

        subroutine axpy(c,x,y,n)
c      =====
c      Constant times a vector plus a vector
        integer i,n
        double precision c,x(*),y(*)
        do 10 i=1,n
            y(i) = y(i) + c*x(i)
10    continue
        return
        end

        subroutine ludcomp(a,n,m,ipvt,info)
c      =====
c      integer m,n,ipvt(m),info
c      double precision a(m,m)
c
c      ludcomp computes the L-U factors of a square

```

237

238

```

c      matrix by Gaussian elimination with pivoting
c
c      adapted from linpack
c
c      on entry
c
c          a          double precision(m,m)
c                     matrix to be factored
c
c          n          integer
c                     order of matrix a
c
c          m          integer
c                     maximum order of matrix a
c
c      on return
c
c          a          upper triangular matrix U and, subdiagonally,
c                     the multipliers in the L-U factorization of
c                     the original matrix a
c
c          ipvt       integer(m)
c                     vector of pivot indices
c
c          info        integer
c                     = 0  normal value,
c                     > 0  indicates singularity
c
c      calls: scal, axpy
c
c      double precision t,dmax,dtmp
c      integer ip,j,k
c
c      info = 0
c      do 60 k = 1, n-1
c          Find pivot index
c          ip = k
c          dmax = dabs(a(k,k))
c          do 40 i = k+1,n
c              dtmp = dabs(a(i,k))
c              if(dtmp.le.dmax) go to 40
c              ip = i
c              dmax = dtmp
40      continue
c          ipvt(k) = ip
c          Zero pivot implies column already triangularized
c          if (a(ip,k) .eq. 0.0d0) then
c              info = k
c              go to 60
c          endif
c          Interchange if necessary
c          if (ip .ne. k) then
c              t = a(ip,k)
c              a(ip,k) = a(k,k)

```

```

        a(k,k) = t
    endif
c    Compute multipliers
    t = -1.0d0/a(k,k)
    call scal(t,a(k+1,k),n-k)
c    Row elimination with column indexing
    do 30 j = k+1, n
        t = a(ip,j)
        if (ip .ne. k) then
            a(ip,j) = a(k,j)
            a(k,j) = t
        endif
        call axpy(t,a(k+1,k),a(k+1,j),n-k)
30    continue
60 continue
    ipvt(n) = n
    if (a(n,n) .eq. 0.0d0) info = n

    return
end

```

240

```

subroutine lusolve(a,b,n,m,ipvt)
=====
c    integer n,ipvt(m)
c    double precision a(m,m),b(m)

c
c    lusolve solves the double precision system
c    a * x = b using factors computed by ludcomp
c
c    adapted from linpack
c
c    on entry
c
c        a        double precision(m,m)
c                  output from ludcomp
c
c        n        integer
c                  order of matrix a
c
c        m        integer
c                  maximum order of matrix a
c
c        ipvt     integer(m)
c                  pivot vector from ludcomp
c
c        b        double precision(m)
c                  right hand side vector
c
c    on return
c
c        b        solution vector x
c
c    error condition

```

```

c
c      division by zero will occur if the input factor contains a
c      zero on the diagonal; technically this indicates
singularity
c
c      calls: axpy
c
c      double precision t
c      integer k,l

c      First solve  $l*y = b$ 
do 20 k = 1, n-1
    l = ipvt(k)
    t = b(l)
    if (l .eq. k) go to 10
    b(l) = b(k)
    b(k) = t
10  continue
    call axpy(t,a(k+1,k),b(k+1),n-k)
20  continue

c      Now solve  $u*x = y$ 
do 40 k = n, 1, -1
    b(k) = b(k)/a(k,k)
    t = -b(k)
    call axpy(t,a(1,k),b(1),k-1)
40  continue

    return
end

```

表 11.1 格瑞-劳韦模型关于人口和服务就业分布的模拟

区块位置	人口				服务就业			
	基本案例 ¹	基本就业 各区相同	$\alpha, \beta=2$	带郊区高 速环路	基本案例	基本就业 各区相同	$\alpha, \beta=2$	带郊区高 速环路
1	8.3188	5.5322	16.8392	8.1525	1.8900	1.7046	3.4875	1.8197
2	7.1632	5.3157	12.3649	7.0032	1.8069	1.6356	3.1947	1.7391
3	5.2404	4.5684	6.2167	5.1022	1.5003	1.3923	1.8949	1.4407
4	4.2291	4.0251	3.9251	4.1066	1.2883	1.2171	1.2844	1.2346
5	3.5767	3.6616	2.7623	3.4981	1.1300	1.0985	0.9394	1.0948
6	3.1075	3.4179	2.0684	3.0942	1.0050	1.0177	0.7202	0.9979
7	2.7465	3.2629	1.6099	2.8178	0.9022	0.9649	0.5691	0.9323
8	2.4551	3.2398	1.2843	2.6641	0.8150	0.9522	0.4585	0.9069
9	2.2113	2.8620	1.0397	2.3538	0.7389	0.8407	0.3734	0.8012
10	2.0014	2.5659	0.8465	2.1094	0.6712	0.7533	0.3047	0.7181

¹ 基本案例：所有基本就业集中在 CBD， $\alpha = \beta = 1$ ，每条道路的通行速度相同。