摘要

1. **问题重述**
   1. **问题背景**

习近平总书记就生态文明建设提出了一系列新思想新理念，深刻诠释了“山水林田湖草”是一个生命共同体的发展思路。林草作为地球的“健康色”，林草高质量发展对生态环境保护起着重要意义，在维护国家生态安全，推进生态文明建设中具有非常广泛作用，于社会发展、经济发展、生态发展都有重要的促进作用。因此，扎实推进林草高质量发展与保护，是践行习近平生态文明思想的具体行动，是实现“山水林田湖草”系统治理的必然选择。

黄土高原日照时间长，昼夜温差大，工业污染较少，是联合国粮农组织认定的苹果最佳优生区，也是红枣、核桃、花椒、葡萄等经济林果品的优生区域，具有发展经济林果产业优势。黄土高原地区社会经济的可持续发展是生态环境改善的根本保障，黄土高原一些县区的实践证明，经济林果产业的发展能够实现生态环境改善和社会经济发展的双赢。经济林是兼具生态、经济和社会效益的林种，国家新一轮退耕还林取消了营造生态林与经济林的比例限制，并明确提出大力发展特色经济林等林业产业。因此，我们需要对光照这一经济作物质量的重要影响因素进行分析，突出当地资源优势，因地制宜、合理布局、规模发展，加大对现有经济林的精细化科学管理，提高经济林果品的产量和质量，提升黄土高原经济林果品基地建设，以经济效益保障生态效益，巩固生态环境治理成效。[1][2]

* 1. **问题重述**

问题 1：试分析单棵植株树冠的受光及遮阴形成阴影与时间，季节变化的相关性。

问题 2：试分析不同林带行向间植株树冠的受光及遮阴而形成阴影与时间、季节变化的相关性。

问题 3：试分析林区中不同行列间树木树冠的受光及遮阴形成阴影与时间，季节变化的相关性。

问题 4：以西北坡向，坡度为 10°的陕北黄土高原地形重分析问题三。

问题 5：根据以上分析，对于林木栽植与林木结构管护提出合理建议

1. **问题分析**

考虑到真实树冠内部枝叶交错，结构复杂，将树冠层结构简化为三层纺锤结构，以树冠中心为坐标轴原点建立三维坐标系，并列出每层相应的椭球方程。

**问题1的分析：**

该题目可分割为受光与遮阴两个部分。

1. 针对单棵苹果树树冠的受光，根据设定时间的太阳高度角与太阳方位角，太阳高度平面作为面光源，在以树冠中心为坐标轴原点的三维坐标系中建立光线参数方程，与树冠层外层的椭球方程联立后求解出首个交点并进行标记，所有标记点拟合后形成曲面即为设定时间下单棵植株的受光面。

2. 针对单棵苹果树树冠遮阴形成阴影，以一定步长在光线上取点，判断点是否在树冠层内，在其中哪一层，通过对树冠层三层分别设立权重，得到每条光线所包含点的加权总数，设立标准判断阴影深度。

**问题2的分析：**

在问题一模型的基础上将受光目标拓展为单行林带，根据重要特征，按行列方向与树间距设计共计6组林带进行相关情况模拟。针对受光仅需考虑中间树冠层的情况，向整行推广即可；由于有邻木遮挡，针对阴影则须考虑整体阴影情况。

1. **模型假设**
2. **符号说明**

|  |  |  |
| --- | --- | --- |
| **符号** | **意义** | **单位** |
|  | 本地经度 | ° |
|  | 本地纬度 | ° |
|  | 太阳赤纬 | ° |
|  | 时角 | ° |
|  | 太阳高度角 | ° |
|  | 太阳方位角 | ° |

表1 符号说明

1. **模型建立与求解**

**5.1 问题一模型建立与求解**

**5.1.1 太阳高度角及方位角计算**

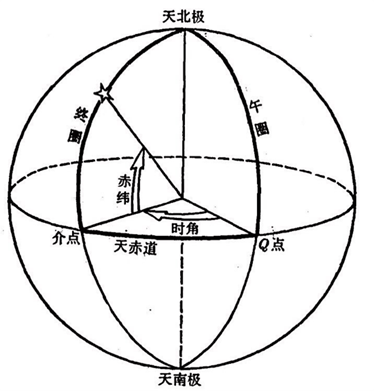
对于一个特定区域范围内的经济林带而言，不同时间下其受光照情况各不相同，其主要原因是当地太阳高度角及太阳方位角在随着时间季节变化而变化。在此引入这两个概念，并通过题目给出的延安经纬度，对太阳高度角及太阳方位角进行了计算。

图1 时角与赤纬

首先对另外两个会在计算过程中出现的值——时角与赤纬进行解释

时角是天文学专有名词，对于天球上一个点而言，其时角是两个平面之间的角度: 一个平面包含地轴和天顶 (子午面)，另一个平面是穿过该点与极点 (地轴) 的球面大圆切圆所形成的平面（赤经圈)，如图1所示。时角由子午线确定，遵循的规则是在子午线的东边则为负时角，在子午线的西边则为正时角。时角与赤纬结合使用，即可精确确定天球上一点在赤道坐标系下的坐标。其计算过程如下：

其中为北京时间，是本地经度。

赤纬与地球上的纬度相似，是纬度在天球上的投影。赤纬的単位是度，更小的单位是“角分”和“角秒”。太阳赤纬等于太阳入射光与地球赤道之间的角度，由于地球自转轴与公转平面之间的角度基本不变，因此太阳的赤纬随季节周期性变化，变化的周期等于地球的公转周期，即一年。由于地球公转轨道的偏心率非常低，可以看作是一个圆圈，太阳赤纬可用下面两个公式来计算：

余弦中的角度的单位是角度，是一年中的日数，例如月日的日数计为

通过时角与赤纬，可计算任意时刻的太阳高度角和太阳方位角。

太阳高度角，也称太阳高度，是指某地的太阳光线与当地地平面的所交的最小线面角，即为以太阳视平面的几何中心和理想地平线所夹的角度。其计算可以使用以下算式，得到良好的近似值：

此处 是太阳高度角。是地方恒星时系统下的时角。是当前太阳的赤纬,是当地的纬度。

太阳方位角则是太阳在方位上的角度，它通常被定义为从正北方向沿着地平残顺时针量度的角。其可以利用下面的公式，经由计算得到良好的近似值。注意到此处公式使用的是余弦函数，所以方位角永远是正值，因此，角度永远被解释为小于度，而必须依据时角来修正。

其中是太阳方位角, 是太阳高度角，是时角，是太阳赤纬，是当地纬度。

延安当地的经度为，纬度为，故由式可得一年中的春分、夏至、秋分、冬至四天的每个时段的太阳高度角及方位角，如表2、3所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时间 | 8时 | 11时 | 14时 | 17时 |
| 春分（3月21日） | 15.253° | 46.078° | 48.806° | 19.927° |
| 夏至（6月22日） | 28.868° | 64.460° | 68.721° | 33.650° |
| 秋分（9月23日） | 15.003° | 45.728° | 48.437° | 19.670° |
| 冬至（12月22日） | 0.551° | 25.411° | 27.300° | 4.643° |

表2 太阳高度角随季节、时间的变化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时间 | 8时 | 11时 | 14时 | 17时 |
| 春分（3月21日） | 102.306° | 141.640° | 210.452° | 253.746° |
| 夏至（6月22日） | 80.961° | 113.631° | 237.557° | 275.847° |
| 秋分（9月23日） | 102.634° | 141.929° | 210.204° | 253.415° |
| 冬至（12月22日） | 120.127° | 154.068° | 200.161° | 236.186° |

表3 太阳方位角随季节、时间的变化

**5.1.2 苹果树树冠模型**

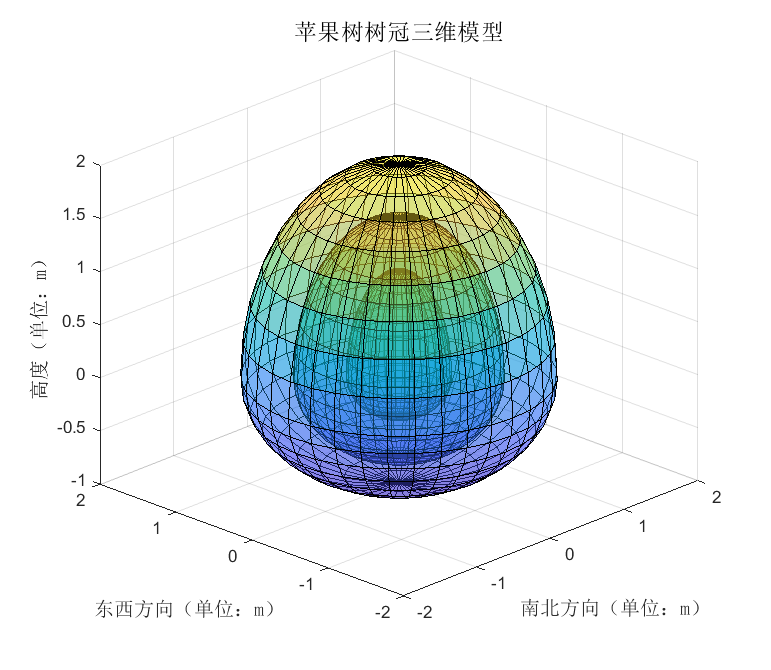
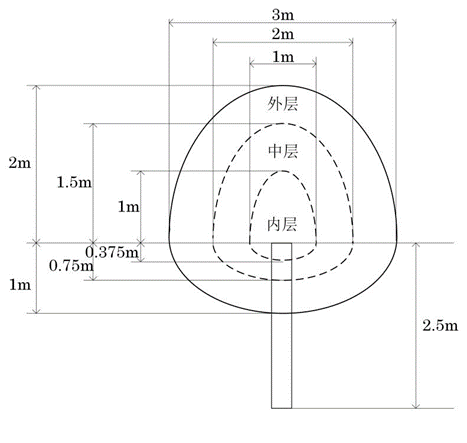
考虑到苹果树具体形态较难模拟，模型建立会非常复杂，经过资料查找与不同苹果树植株形态比对，现将苹果树形态简化为纺锤模型，以方便模型建立与分析。在文献查阅和实际观察中易得，树冠不同部分的枝叶疏密程度不同，整体呈现为内疏，中密，外次疏的特征。考虑到纺锤模型本身为均匀实心结构，需要对模型做出一定的优化调整。首先定义疏密参数，越接近1的其枝叶越稠密，越接近0的其枝叶越稀疏。此处使用叠套的方式来模拟苹果树的疏密特征，由内到外依次将疏密参数为共三层纺锤进行嵌套。具体结构和三维示意图如图3、4所示：

图3 苹果树树冠结构示意图

图4 苹果树树冠三维模型

以树冠中心为原点，南北方向为轴，东西方向为轴，树木生长方向为轴建立三维坐标系。此处即可使用椭球方程对树冠上下部分分别进行表示。

其中分别等于该层轴、轴、轴半径，由结构图易得：

内层：

中层：

外层：

**5.1.3 苹果树树冠受光情况分析**

在此情况下只需考虑树冠外层表面受光情况。具体分析分布如下：

**Step1:** 在其时太阳高度平面（,）上任意取点（,），在上述建立的三维坐标系中建立某一时间光线的空间直线参数方程：

其中)，

**Step2**：将经过该点的光线方程和树冠外层方程联立，判断其光线方程是否与树冠外层方程存在交点。如若存在两个交点，则只取第一个交点在树冠外层标记；如若只有一个交点，则直接标记该交点即可；如若不存在交点，则无需标记。

**Step3：**将所有标记点拟合成面并显示，即为该时间段树冠表面受光面积。

**Step4：**对受光面积进行计算，设空间曲面 的方程为 ，其中 为曲面 在 平面上的投影域，函数 在曲面上连续，如果 在 上有连续的一阶偏导数，则有

其中 是 在 上的投影域， 和 表示在 内某点 处的两个偏导数。由第一型曲面积分的定义，于是将第一型曲面积分化为二重积分的计算

在此以春分、夏至、秋分、冬至四个具有季节代表性的日期为例，分取当天8时、11时、14时，17时四个时间段进行光照情况模拟，具体情况如下述俯视图所示（均遵循“上北下南，左西右东”的方向原则）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **时间** | **8时** | **11时** | **14时** | **17时** |
| 春分 |  |  |  |  |
| 受光面积 |  |  |  |  |
| 夏至 |  |  |  |  |
| 受光面积 |  |  |  |  |
| 秋分 |  |  |  |  |
| 受光面积 |  |  |  |  |
| 冬至 |  |  |  |  |
| 受光面积 |  |  |  |  |

表4 单株苹果树树冠受光随季节、时间的变化情况

通过对表格横向与纵向的分析可得：

1. 随时间变化：树冠层受光面由早到晚呈自东向西的旋转趋势，早晚受光更集中在树冠层侧面上（图中显示为圆的边缘），靠近正午时受光面开始向顶层偏移（图中显示为圆的中心）。此时受光也更密集充分。
2. 随季节变化：夏季受光在树冠层顶层更为集中，受光面整体偏向北方一些，冬季受光在树冠层侧面更为集中，受光面整体偏向南方一些。总的趋势呈现为受光由侧到顶，再回到侧循环变化，受光面偏移自北向南，再回到南循环变化。

将模拟范围扩展至8时至17时，步长取为一个小时，对每一时间点的树冠层受光照面积进行统计，数据拟合曲线如下图所示。

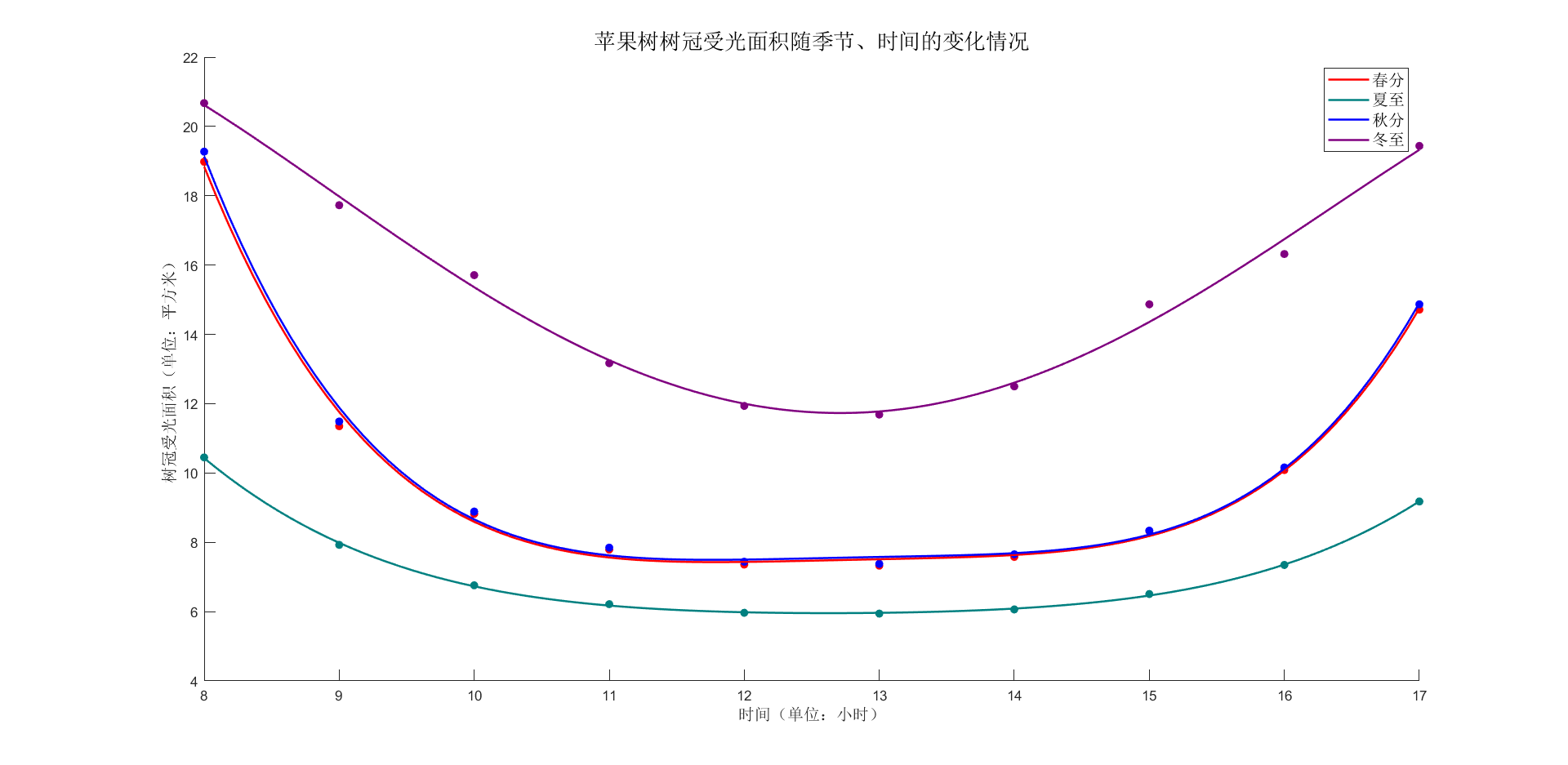


图5 苹果树树冠受光面积随季节、时间的变化情况

分析上图中曲线变化趋势可得：

1. 受光面积随时间变化情况：8时光照面积即处于峰值，随后受光面积随时间缩减，约在正午时分达到谷值，而后又随时间扩增，整体变化曲线为U形，即呈现先减后增的变化趋势。
2. 受光面积随季节变化情况：在一天之中，夏季树冠受光面积在谷值稳定持续时间较长，整体变化趋于平缓，冬季树冠受光面积处于时刻变化状态，峰值与谷值存在一定差距，而春秋两季属于树冠层受光面积早晚变化速度较快、变化幅度较大，中间时间段却较为平稳地维持在谷值。此外，夏季整体受光面积最小，冬日整体受光面积最大，春秋季整体受光面积相近，随四季更迭整体呈现“减—增—减”的循环变化趋势。

**5.1.4 苹果树树冠阴影情况分析**

通过观察比对可以发现，阳光下树木产生阴影在不同部分阴影深度不一，在此特引入加权阴影深度S对树冠遮挡产生阴影进行评估。

**Step1：**建立光线方程，具体步骤同上。

**Step2：**记内层内点个数集为，中层内点个数集为，外层内点个数集为，初始值均设为0

**Step3：**取步长为0.01，分别判断每一个点是否在树冠层的纺锤形状内，如不在则跳过该点，如在，则判断该点在树冠层内层、中层还是外层，并在相应的个数集内+1。

**Step4：**对分别设立权重，通过加权相加的办法，计算得出该条线的加权阴影深度值:

并根据值的大小对加权阴影深度进行分类:

如,则产生的阴影影深为淡影

如,则产生的阴影影深为暗影

如，则产生的阴影影深为浓影

**Step5：**重复上述过程，对所有值大于0的光线与地平面的交点进行标记，将所有标记点拟合成面，所得即当前时段树冠层遮挡阳光产生的阴影。

**Step6：**根据组成阴影的标记点个数计算该阴影面积。

在此以春分、夏至、秋分、冬至四日为例，分取当天8时、11时、14时，17时四个时间段进行阴影情况模拟，其中冬至日由于8时和17时太阳高度角过低，造成阴影过长，数据量较大，在此更改为9时与16时。具体情况如下述俯视图所示（均遵循“上北下南，左西右东”的方向原则）

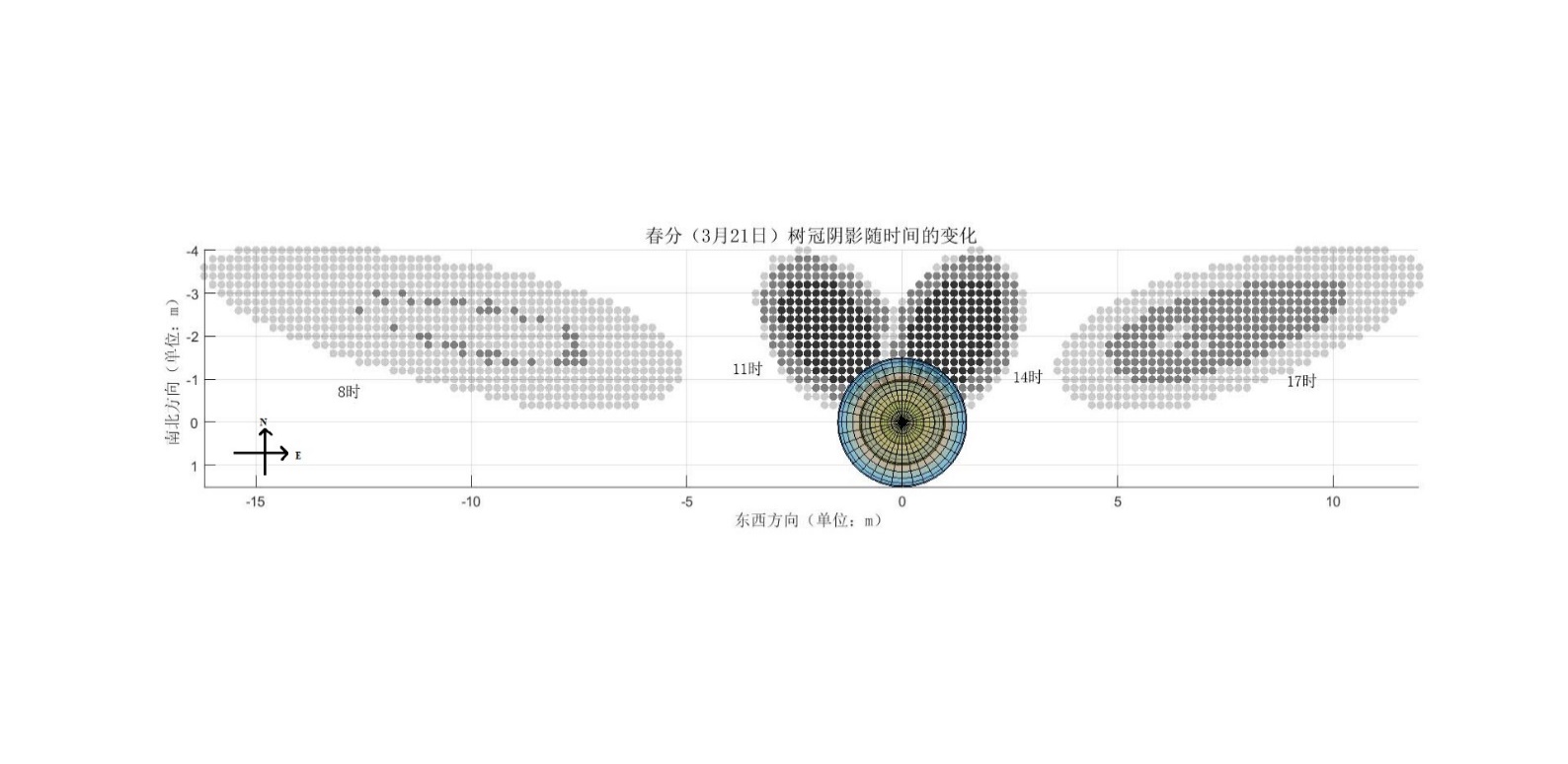


图6 苹果树树冠阴影在春分（3月21日）不同时间的情况

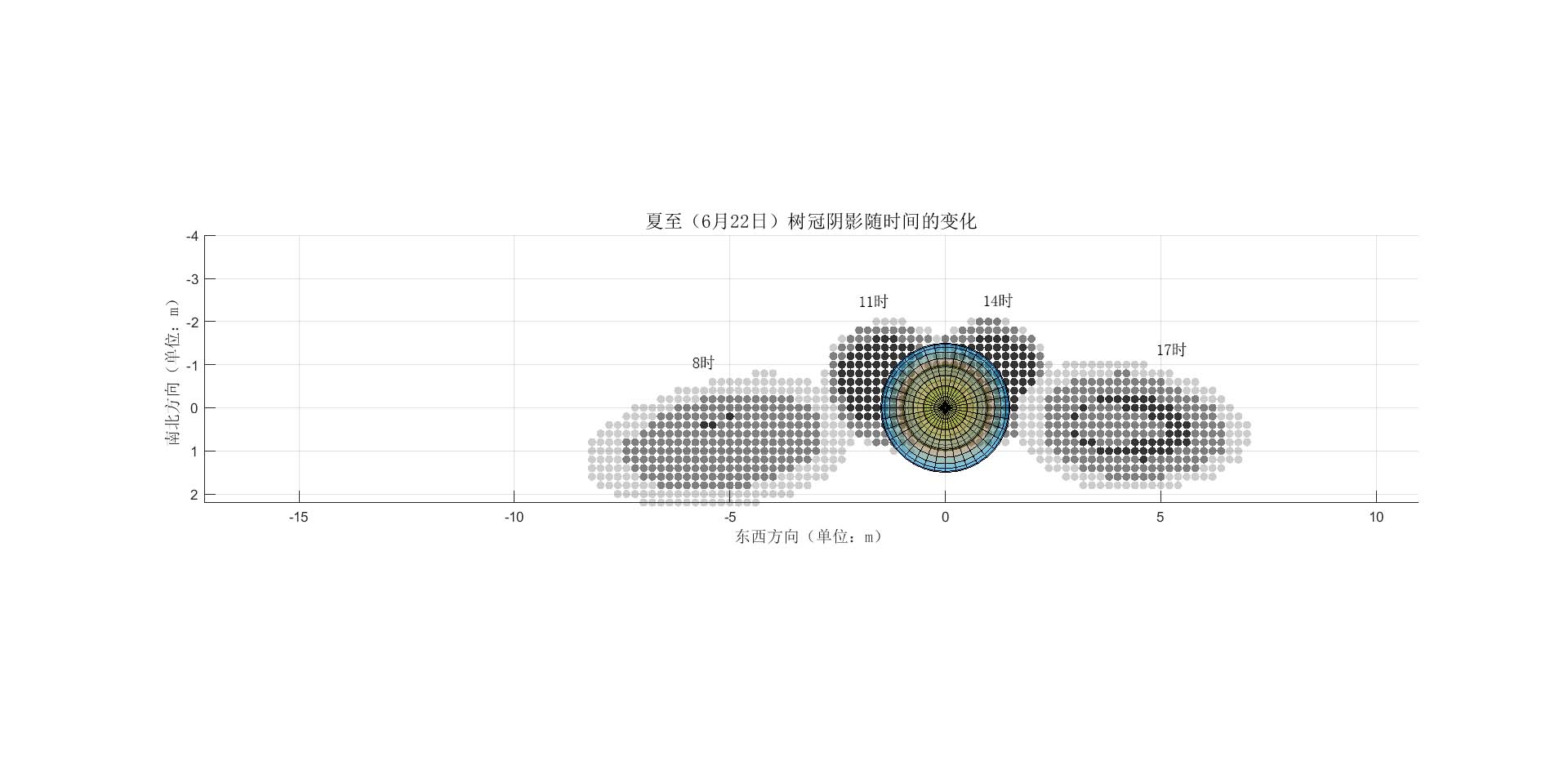


图7 苹果树树冠阴影在夏至（6月22日）不同时间的情况

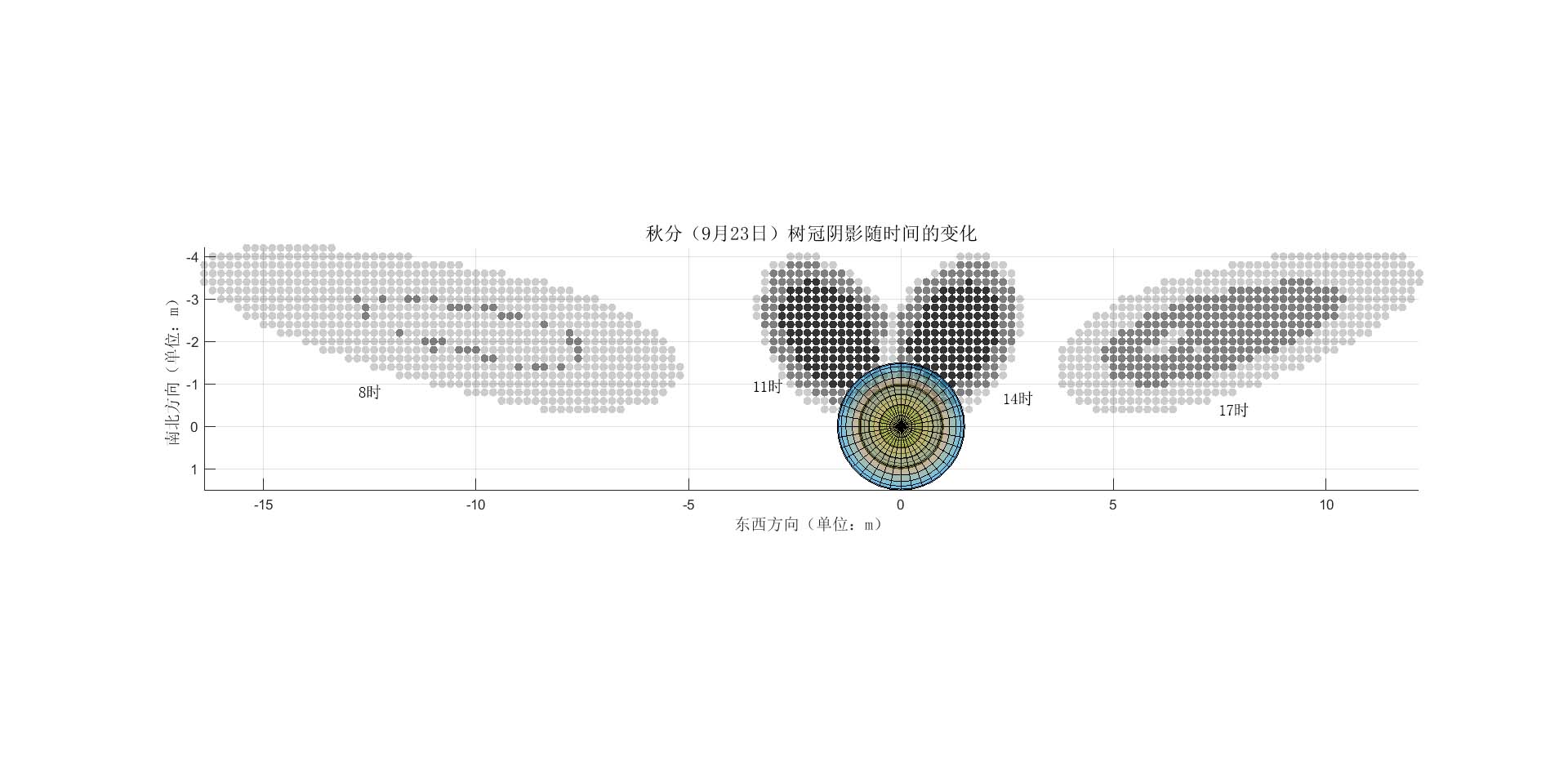


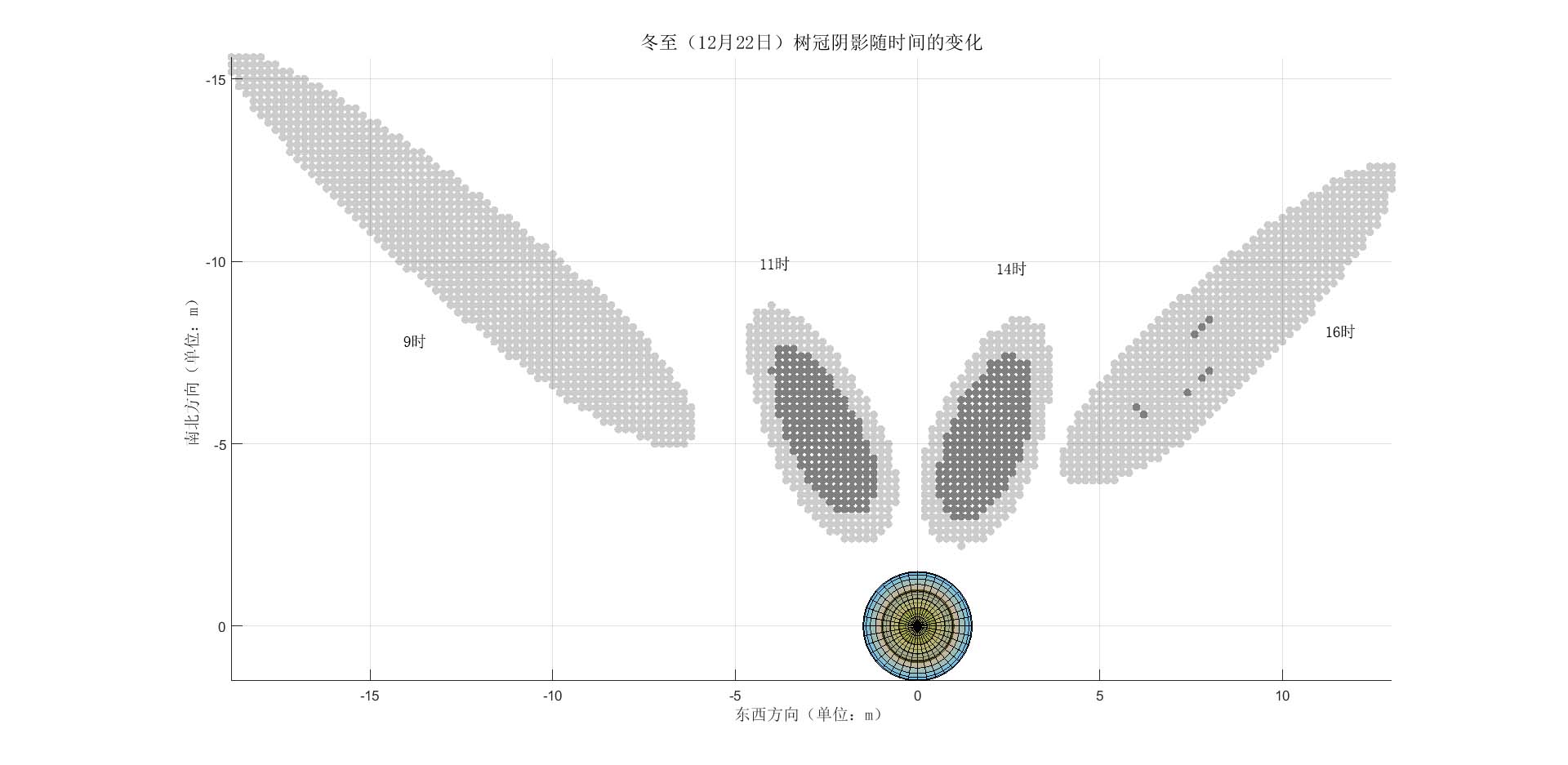
图8 苹果树树冠阴影在秋分（9月23日）不同时间的情况

图9 苹果树树冠阴影在冬至（12月22日）不同时间的情况

综合上面四图分析可得：阴影总体移动方向为自西向东顺时针旋转，在夏季整体方向偏南方一些，在冬季整体方向偏北方一些。无论对于哪一个时间点而言，树冠层阴影主要深度都集中在阴影中间，越靠近边缘影深越浅。时间越靠近早晚，日期越接近冬季，阴影越长，整体面积越大，影深越浅（以淡影和暗影为主）；时间越靠近正午，日期越接近夏季，阴影越短，整体面积越小，影深越深（以暗影与浓影为主）

将模拟范围扩展至8时至17时，步长取为一个小时，对每一时间点的阴影面积进行统计，数据拟合曲线如下图所示。

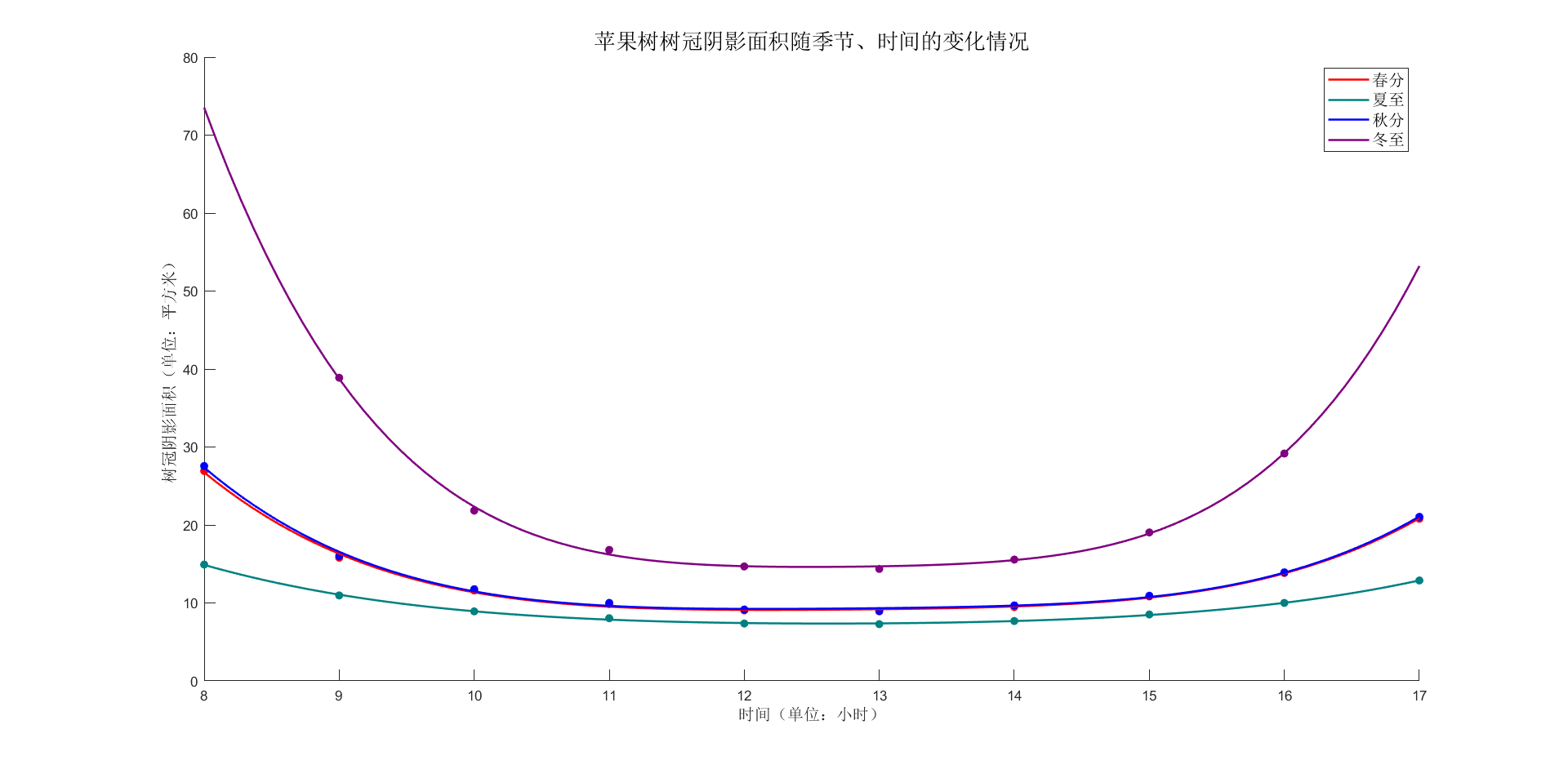


图10 苹果树树冠阴影面积随季节、时间的变化情况

通过对上图的观察分析不难得出：

1. 阴影面积随时间变化情况：8时（初始时间）阴影面积即处于峰值，从8时至11时面积随时间缩减，11时至14时基本维持在谷值，处于平稳状态，14时至17时面积随时间扩增，整体变化曲线为U形，即呈现先减后增的变化趋势。
2. 阴影面积随季节变化情况：春、秋、夏季阴影面积一天中变化不大，冬季早晚与正午的阴影面积存在较大差值。此外夏季阴影面积最小，冬日阴影面积最大，春秋季阴影面积相近，随四季更迭整体呈现“减—增—减”的循环变化趋势。

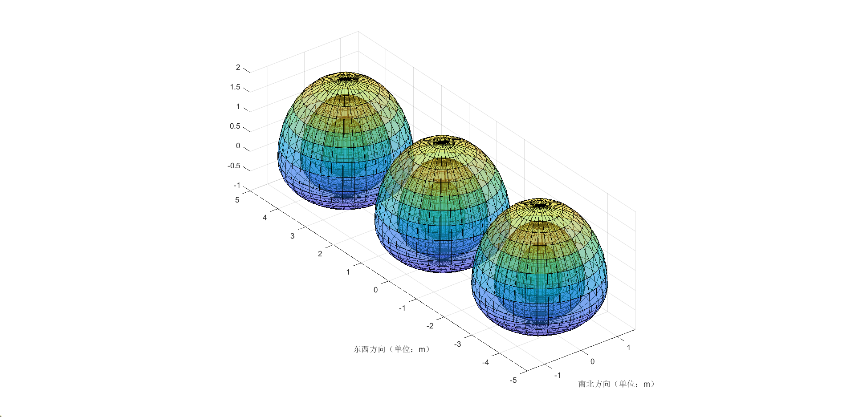
**5.2 问题二模型建立与求解**

**5.2.1 行向间苹果树树冠模型**

在不同林带行向间，由于相邻树木之间的相互遮挡，树木树冠的受光及遮阴而形成阴影情况也会发生相应变化，经过相关查询、研究与评估，在此只对较为重要的两个影响因素——行列方向以及树与树之间的距离进行影响程度分析。

关于行列方向，此处对东—西走向与南—北走向两种不同方向的单行林带做出区分，分别进行相关分析后再比对。

关于树与树间距，则只需在问题一模型的基础上进行扩展，将受光实体改成以相同的数棵苹果树沿直线等间隔排列构建而成的单行林带。此处取树木数量均为3，树与树间距分别为3.5，5，6.5的三种单行林带进行相关数据模拟。模型的具体结构如下图所示。



**5.2.2 行向间苹果树树冠受光情况分析**

由于排列的规律性以及太阳光的平行光性质，此处只需考虑中间树冠层的受光情况，最后再进行推广即可。

分东—西走向与南—北走向两组，每组以春分、夏至、秋分、冬至四日为例，以1小时为步长，在8时—17时的时间段内取点，使用5.1.3中所使用建立受光面的方式模拟该时间点该组单行林带受光情况，并计算受光面积。将数值拟合成曲线，结果按方向和间距分类，显示如下：

由对上六图的比对分析可得：

1. 不同于单个树冠层，在春、夏、秋三季的林带中，中间树冠层的受光面积在清晨和傍晚（8时—9时与16时—17时两个时间段）存在极小值，随后会在短时间内急速升高抵达较大值，中间时间段与同时间段单个树冠层的受光情况规律近似。
2. 对于东—西走向的单行林带，仅在树与树间距离较近时，会对春、夏、秋三季早晚的受光面积产生一定影响，具体规律为三季早晚中间树冠层的受光面积随树间距缩减而减小。冬季不受影响
3. 对于南北走向的单行林带，冬季正午左右中间树冠层的受光面积会随距离缩短而减小。春、夏、秋三季不受影响

**5.2.3 行向间苹果树树冠阴影情况分析**

考虑到三个树冠层的阴影可能存在相互重叠的影响，此处选择对单行林带整体阴影进行分析。

分出东—西走向与南—北走向两组，每组仍以春分、夏至、秋分、冬至四日为例，以1小时为步长，在8时—17时的时间段内取点，使用5.1.4中所使用建立阴影方式模拟该时间点该组单行林带阴影并计算面积。其中冬至日由于8时和17时太阳高度角过低，在此不做模拟分析。按组将阴影面积值点拟合成曲线后，生成结果按方向和间距分类，如下所示：

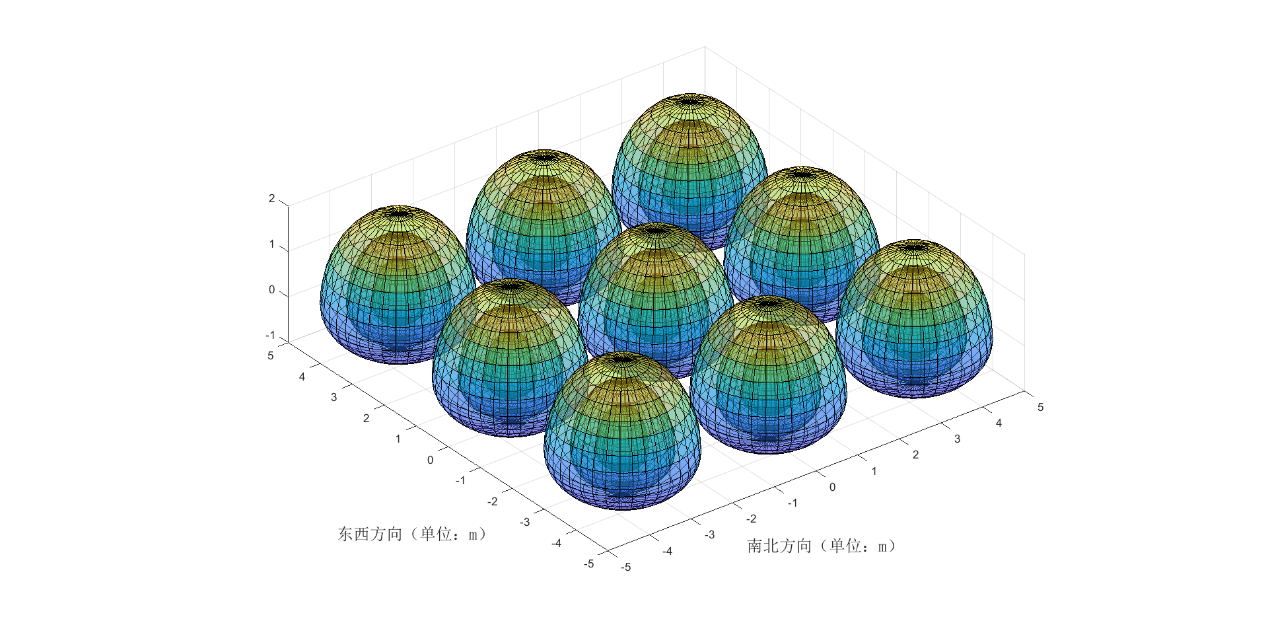
由对上六图的比对分析可得：

1. 无论间距大小与行列方向如何改变，单行林带整体阴影的变化趋势仍符合5.1.4中对于单个树冠层产生阴影的分析。详见上，在此不多赘述。
2. 对于东—西走向的单行林带而言，树与树间距离仅对日出与日落（即春夏秋三季8时与17时两个时间点，冬季9时与16时两个时间点）时的阴影面积产生一定影响。在一定限度内，树间距越大，单行林带所产生的整体阴影面积越大。
3. 对于南—北走向的单行林带而言，树与树间距离对其产生的整体阴影面积几乎完全没有影响
4. 单行林带的行列方向对于夏、冬两季其整体阴影面积影响不大，春、秋两季单行林带的整体阴影面积则呈现为南—北走向的大于东—西走向的。

**5.3 问题三模型建立与求解**

**5.3.1 行列间苹果树树冠模型**

相较起行向间树与树之间的遮挡关系而言，行列间的遮挡关系无疑更加复杂。在行向间苹果树树冠模型的基础上，将等距样本单行林带拓展为东南西北正方向上的的正方形林区，建立行列间苹果树树冠模型。具体结构如下图所示：



**5.3.2 行列间苹果树树冠受光情况分析**

**5.3.3 行列间苹果树树冠阴影情况分析**

与5.2.3类似，考虑区块内整体阴影面积。依旧对春分、夏至、秋分、冬至四日以1小时为步长在8时—17时时间段内取点，使用5.1.4中所使用建立阴影方式模拟特定时间点林区总阴影并计算面积。冬至日8时和17时在此不做模拟。将阴影面积值点拟合成曲线后，生成结果按日期分类，如下所示：

对上四图曲线进行比对分析可得：

1． 与单个苹果树树冠层阴影情况类似，不同季节与不同间距均呈现阴影面积先减后增的变化趋势。春秋两季变化曲线近似甚至重合，相较于春、秋、冬三季，夏季变化较为平稳。

2． 不同树间距仅对早晚阴影面积产生影响，树间距越小阴影面积越少，正午时分不同树间距产生的阴影的面积均会处于一个相同的谷值。此值以夏季最小，春秋两季相同。而对于冬季而言，不同树间距对应谷值均不同，树间距越大，谷值越大。

**5.4 问题四的模型建立与求解**

**5.4.1斜坡苹果树林区树冠模型**

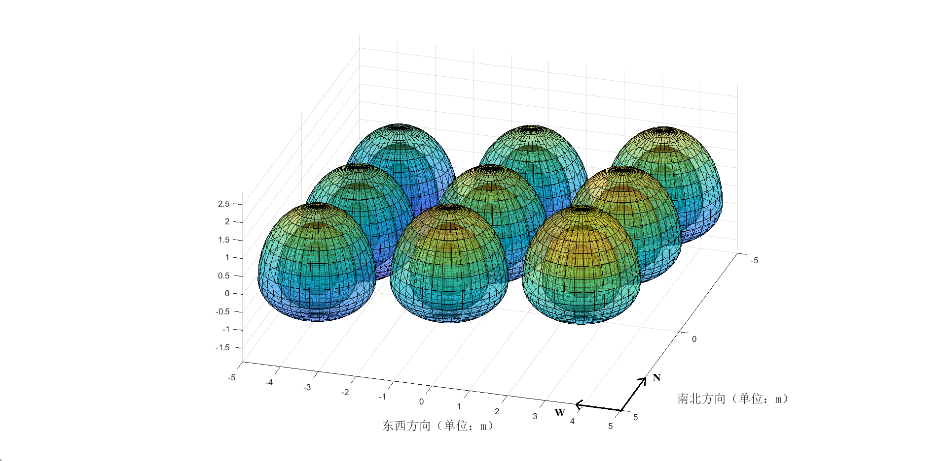
此问要求将问题三所建立模型运用在西北坡向，坡度为10°的陕北黄土高原地形之上。由于地面此时不再为水平面，需要对先前建立的行列间苹果树树冠模型进行一定改良。

仍考虑考虑的正方形林区，以中间树冠层的中心坐标轴原点，将坡面西北方向抬高，东南方向压低，使得整体坡面与水平面呈10°夹角。则正东—西方向上与正南—北方向上坡度（比）为：

根据设定的树间距d可计算方向上相邻树冠层的高度差，具体方程为：

将正方形林区内9个树冠层的z轴坐标用矩阵表示，结果为：

以此为基准，对行列间苹果树树冠模型中的各个样本进行位置上的改变，具体结构如下图所示



仿照问题3中的方式，对此新模型重新进行模拟分析。

附录

附件1 apple\_tree\_model.m（苹果树树冠三维模型，MATLAB代码）

clc,clear

%树冠可视化模型

[theta,phi] = meshgrid(0:10:360,0:10:90);

X = 1.5\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y = 1.5\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z = 2\*sin(phi\*pi/180);

ss1=surf(X, Y, Z);

ss1.FaceAlpha=0.4;

hold on;

[theta,phi2] = meshgrid(0:10:360,-90:10:0);

X2 = 1.5\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y2 = 1.5\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z2 = 1\*sin(phi2\*pi/180);

ss2=surf(X2, Y2, Z2);

ss2.FaceAlpha=0.4;

hold on;

[theta,phi] = meshgrid(0:10:360,0:10:90);

X3 = 1\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y3 = 1\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z3 = 1.5\*sin(phi\*pi/180);

ss3=surf(X3, Y3, Z3);

ss3.FaceAlpha=0.4;

hold on;

X4 = 1\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y4 = 1\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z4 = 0.75\*sin(phi2\*pi/180);

ss4=surf(X4, Y4, Z4);

ss4.FaceAlpha=0.4;

hold on;

X5 = 0.5\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y5 = 0.5\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z5 = 1\*sin(phi\*pi/180);

ss5=surf(X5, Y5, Z5);

ss5.FaceAlpha=0.4;

hold on;

X6 = 0.5\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y6 = 0.5\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z6 = 0.375\*sin(phi2\*pi/180);

ss6=surf(X6, Y6, Z6);

ss6.FaceAlpha=0.4;

axis equal

附件2 single\_apple\_tree\_model\_light.m（单株苹果树树冠受光模型，MATLAB代码）

clc,clear

%树冠可视化模型

[theta,phi] = meshgrid(0:10:360,0:10:90);

X11 = 1.5\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y11 = 1.5\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z11 = 2\*sin(phi\*pi/180);

ss11=surf(X11, Y11, Z11);

ss11.FaceAlpha=0.4;

ss11.FaceColor = [0.5 0.5 0.5];

hold on;

[theta,phi2] = meshgrid(0:10:360,-90:10:0);

X12 = 1.5\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y12 = 1.5\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z12 = 1\*sin(phi2\*pi/180);

ss12=surf(X12, Y12, Z12);

ss12.FaceAlpha=0.4;

ss12.FaceColor = [0.5 0.5 0.5];

hold on;

%树冠模型参数

a11=1.5;b11=1.5;c11=2;

a12=1.5;b12=1.5;c12=1;

center1=[0,0,0];

%两个角度

elevation\_angle=[15.253 46.078 48.806 19.927 28.868 64.460 68.721 33.650 15.003 45.728 48.437 19.670 10.438 25.411 27.300 14.029];

azimuth\_angle=[102.306 141.640 210.452 253.746 80.961 113.631 237.557 275.847 102.634 141.929 210.204 253.415 129.748 154.068 200.161 225.982];

for s=1:1:16

azimuth\_angle(s)=180-azimuth\_angle(s);

end

% 将角度转换为弧度

elevation\_rad = deg2rad(elevation\_angle(1));

azimuth\_rad = deg2rad(azimuth\_angle(1));

% 光线方向向量

direction\_vector = [cos(azimuth\_rad) \* cos(elevation\_rad);

sin(azimuth\_rad) \* cos(elevation\_rad);

sin(elevation\_rad)];

%光线参数

point1=[0,0,-2.5];

point22=point1+direction\_vector';

p=1;

while point22(3)<5

point22=point22+direction\_vector';

p=p+1;

end

direction=point22-point1;

num=0;

%步长计算

S=0;

numIntersectionPoints1=0;

for x =-10:0.1:10

for y =-40:0.1:20

point1=[0,0,-2.5];

point1=point1+[x,0,0]+[0,y,0];

point2=point22+[x,0,0]+[0,y,0];

%plot3([point1(1),point2(1)],[point1(2),point2(2)],[point1(2),point2(2)])

Bool = 0;

for t = 0:0.01:1 % 步长为0.01

point = point2 - t \* direction;

% 判断点是否在半椭球内

if point(3)>=0

if (((point(1)-center1(1))/a11)^2 + ((point(2)-center1(2))/b11)^2 + ((point(3)-center1(3))/c11)^2 <= 1) & Bool == 0

numIntersectionPoints1 = numIntersectionPoints1 + 1;

Bool = 1;

scatter3([point(1)], [point(2)],[point(3)],'filled', 'MarkerFaceColor','y','MarkerFaceAlpha', 1);

A = sqrt((point(1)/a11^2)^2 + (point(2)/b11^2)^2 + (point(3)/c11^2)^2);

sintheta=(point(2)/b11^2)/A;

sintheta=sqrt(1-sintheta^2);

S=S+0.01\*abs(sintheta);

end

elseif point(3)<0

if (((point(1)-center1(1))/a12)^2 + ((point(2)-center1(2))/b12)^2 + ((point(3)-center1(3))/c12)^2 <= 1) & Bool == 0

numIntersectionPoints1 = numIntersectionPoints1 + 1;

Bool = 1;

scatter3([point(1)], [point(2)],[point(3)],'filled', 'MarkerFaceColor','y','MarkerFaceAlpha', 1);

A = sqrt((point(1)/a12^2)^2 + (point(2)/b12^2)^2 + (point(3)/c12^2)^2);

sintheta=(point(2)/b12^2)/A;

sintheta=sqrt(1-sintheta^2);

S=S+0.01\*abs(sintheta);

end

end

end

hold on;

end

end

disp(numIntersectionPoints1)

disp(S)

xlabel('南北方向（单位：m）','FontSize',12,'FontName','宋体');

ylabel('东西方向（单位：m）','FontSize',12,'FontName','宋体');

view(90,90)

axis equal;

附件3 single\_apple\_tree\_model\_light\_area.m（单株苹果树树冠受光面积曲线拟合模型，MATLAB代码）

y1 = [18.9801

11.3502

8.8233

7.7895

7.3577

7.3253

7.5792

8.3211

10.0851

14.7158];

y2 = [10.4491

7.9258

6.7594

6.2143

5.9664

5.9419

6.0624

6.5068

7.3459

9.1783];

y3 = [19.2739

11.4861

8.8863

7.8453

7.4391

7.3775

7.653

8.3368

10.1594

14.8667

];

y4 = [20.6738

17.7271

15.7093

13.1685

11.9353

11.6851

12.4989

14.8676

16.3198

19.4376];

x = [8 9 10 11 12 13 14 15 16 17];

x2=[8 9 10 11 12 13 14 15 16 17];

y1=y1';

y2=y2';

y3=y3';

y4=y4';

% 设置多项式阶数

n = 4;

% 进行四条曲线的曲线拟合

p1 = polyfit(x, y1, n);

p2 = polyfit(x, y2, n);

p3 = polyfit(x, y3, n);

p4 = polyfit(x2, y4, n);

% 生成一系列 x 值

x\_fit = linspace(min(x), max(x), 100);

% 计算四条拟合曲线对应的 y 值

y\_fit1 = polyval(p1, x\_fit);

y\_fit2 = polyval(p2, x\_fit);

y\_fit3 = polyval(p3, x\_fit);

y\_fit4 = polyval(p4, x\_fit);

% 绘制原始数据和拟合曲线

figure;

hold on;

scatter(x, y1, 'o','filled','MarkerFaceColor','r');

scatter(x, y2, 'o', 'filled','MarkerFaceColor',[0,0.5,0.5]);

scatter(x, y3, 'o', 'filled','MarkerFaceColor','b');

scatter(x2, y4, 'o', 'filled','MarkerFaceColor',[0.5,0,0.5]);

plot(x\_fit, y\_fit1,'color','r','LineWidth',1.5);

plot(x\_fit, y\_fit2,'color',[0,0.5,0.5],'LineWidth',1.5);

plot(x\_fit, y\_fit3,'color','b','LineWidth',1.5);

plot(x\_fit, y\_fit4,'color',[0.5,0,0.5],'LineWidth',1.5);

hold off;

xlabel('时间（单位：小时）','FontSize',12,'FontName','宋体');

ylabel('树冠受光面积（单位：平方米）','FontSize',12,'FontName','宋体');

title('苹果树树冠受光面积随季节、时间的变化情况','FontSize',16,'FontName','宋体');

legend('','','','', '春分', '夏至', '秋分', '冬至','FontSize',12,'FontName','宋体');

附件4 single\_apple\_tree\_model\_dark.m（单株苹果树树冠阴影模型，MATLAB代码）

clc,clear

%树冠可视化模型

[theta,phi] = meshgrid(0:10:360,0:10:90);

X = 1.5\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y = 1.5\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z = 2\*sin(phi\*pi/180);

ss1=surf(X, Y, Z);

ss1.FaceAlpha=0.4;

hold on;

[theta,phi2] = meshgrid(0:10:360,-90:10:0);

X2 = 1.5\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y2 = 1.5\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z2 = 1\*sin(phi2\*pi/180);

ss2=surf(X2, Y2, Z2);

ss2.FaceAlpha=0.4;

hold on;

[theta,phi] = meshgrid(0:10:360,0:10:90);

X3 = 1\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y3 = 1\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z3 = 1.5\*sin(phi\*pi/180);

ss3=surf(X3, Y3, Z3);

ss3.FaceAlpha=0.4;

hold on;

X4 = 1\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y4 = 1\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z4 = 0.75\*sin(phi2\*pi/180);

ss4=surf(X4, Y4, Z4);

ss4.FaceAlpha=0.4;

hold on;

X5 = 0.5\*cos(phi\*pi/180).\*cos(theta\*pi/180);

Y5 = 0.5\*cos(phi\*pi/180).\*sin(theta\*pi/180);

Z5 = 1\*sin(phi\*pi/180);

ss5=surf(X5, Y5, Z5);

ss5.FaceAlpha=0.4;

hold on;

X6 = 0.5\*cos(phi2\*pi/180).\*cos(theta.\*pi/180);

Y6 = 0.5\*cos(phi2\*pi/180).\*sin(theta\*pi/180);

Z6 = 0.375\*sin(phi2\*pi/180);

ss6=surf(X6, Y6, Z6);

ss6.FaceAlpha=0.4;

%树冠模型参数

a11=1.5;b11=1.5;c11=2;a21=1;b21=1;c21=1.5;a31=0.5;b31=0.5;c31=1;

a12=1.5;b12=1.5;c12=1;a22=1;b22=1;c22=0.75;a32=0.5;b32=0.5;c32=0.375;

center=[0,0,0];

elevation\_angle=[10.43771148

10.43771148

18.92008084

25.41121023

29.24919056

29.9130712

27.30034527

21.79243076

14.02900813

14.02900813

];

azimuth\_angle=[129.7484118

129.7484118

140.9571165

154.0680759

168.9518354

184.7639009

200.1609288

214.0333748

225.9820887

225.9820887

];

elevation\_angle=elevation\_angle';

azimuth\_angle=azimuth\_angle';

for s=1:1:10

azimuth\_angle(s)=180-azimuth\_angle(s);

end

for n=1:1:10

disp(['阴影点个数为：',num2str(n)])

% 将角度转换为弧度

elevation\_rad = deg2rad(elevation\_angle(n));

azimuth\_rad = deg2rad(azimuth\_angle(n));

% 光线方向向量

direction\_vector = [cos(azimuth\_rad) \* cos(elevation\_rad);

sin(azimuth\_rad) \* cos(elevation\_rad);

sin(elevation\_rad)];

%光线参数

point1=[0,0,-2.5];

point22=point1+direction\_vector';

p=1;

while point22(3)<5

point22=point22+direction\_vector';

p=p+1;

end

direction=point22-point1;

num=0;

%步长计算

for x =-50:0.2:50

for y =-200:0.2:20

point1=[0,0,-2.5];

point1=point1+[x,0,0]+[0,y,0];

point2=point22+[x,0,0]+[0,y,0];

numIntersectionPoints1=0;

numIntersectionPoints2=0;

numIntersectionPoints3=0;

direction=point2-point1;

for t = 0:0.01:1 % 步长为0.01

point = point1 + t \* direction;

% 判断点是否在半椭球内

if point(3)>=0

if ((point(1)-center(1))/a31)^2 + ((point(2)-center(2))/b31)^2 + ((point(3)-center(3))/c31)^2 <= 1

numIntersectionPoints3 = numIntersectionPoints3 + 1;

elseif ((point(1)-center(1))/a21)^2 + ((point(2)-center(2))/b21)^2 + ((point(3)-center(3))/c21)^2 <= 1

numIntersectionPoints2 = numIntersectionPoints2 + 1;

elseif ((point(1)-center(1))/a11)^2 + ((point(2)-center(2))/b11)^2 + ((point(3)-center(3))/c11)^2 <= 1

numIntersectionPoints1 = numIntersectionPoints1 + 1;

end

elseif point(3)<0

if ((point(1)-center(1))/a32)^2 + ((point(2)-center(2))/b32)^2 + ((point(3)-center(3))/c32)^2 <= 1

numIntersectionPoints3 = numIntersectionPoints3 + 1;

elseif ((point(1)-center(1))/a22)^2 + ((point(2)-center(2))/b22)^2 + ((point(3)-center(3))/c22)^2 <= 1

numIntersectionPoints2 = numIntersectionPoints2 + 1;

elseif ((point(1)-center(1))/a12)^2 + ((point(2)-center(2))/b12)^2 + ((point(3)-center(3))/c12)^2 <= 1

numIntersectionPoints1 = numIntersectionPoints1 + 1;

end

end

end

sum=numIntersectionPoints3\*0.2+numIntersectionPoints2\*0.8+numIntersectionPoints1\*0.5;

if sum>0

% 平面方程

% 假设平面方程为 Ax + By + Cz + D = 0

A = 0;

B = 0;

C = 1;

D = 2.5;

syms m

point = point1 + m \* direction;

eqn = A \* point(1) + B \* point(2) + C \* point(3) + D == 0;

t\_sol = solve(eqn, m);

% 计算交点坐标

intersectionPoints = point1 + double(t\_sol) \* direction;

if sum>12

scatter3([intersectionPoints(1)], [intersectionPoints(2)],[intersectionPoints(3)],'filled', 'MarkerFaceColor','k','MarkerFaceAlpha', 0.8);

num=num+1;

elseif sum>6

scatter3([intersectionPoints(1)], [intersectionPoints(2)],[intersectionPoints(3)],'filled', 'MarkerFaceColor','k','MarkerFaceAlpha', 0.5);

num=num+1;

elseif sum>0

scatter3([intersectionPoints(1)], [intersectionPoints(2)],[intersectionPoints(3)],'filled', 'MarkerFaceColor','k','MarkerFaceAlpha', 0.2);

num=num+1;

end

%disp(['阴影程度为：',num2str(sum)])

hold on;

end

end

end

disp(['阴影点个数为：',num2str(num)])

end

axis equal;

附件5 single\_apple\_tree\_model\_dark\_area.m（单株苹果树树冠阴影面积拟合模型，MATLAB代码）

y1 = [1346 790 580 492 452 446 474 542 692 1040] \* 0.02;

y2 = [746 548 446 402 368 364 384 426 500 644] \* 0.02;

y3 = [1378 800 588 500 458 448 484 546 696 1052] \* 0.02;

y4 = [1944 1092 840 734 718 778 952 1458] \* 0.02;

x = [8 9 10 11 12 13 14 15 16 17];

x2=[9 10 11 12 13 14 15 16];

% 设置多项式阶数

n = 4;

% 进行四条曲线的曲线拟合

p1 = polyfit(x, y1, n);

p2 = polyfit(x, y2, n);

p3 = polyfit(x, y3, n);

p4 = polyfit(x2, y4, n);

% 生成一系列 x 值

x\_fit = linspace(min(x), max(x), 100);

% 计算四条拟合曲线对应的 y 值

y\_fit1 = polyval(p1, x\_fit);

y\_fit2 = polyval(p2, x\_fit);

y\_fit3 = polyval(p3, x\_fit);

y\_fit4 = polyval(p4, x\_fit);

% 绘制原始数据和拟合曲线

figure;

hold on;

scatter(x, y1, 'o','filled','MarkerFaceColor','r');

scatter(x, y2, 'o', 'filled','MarkerFaceColor',[0,0.5,0.5]);

scatter(x, y3, 'o', 'filled','MarkerFaceColor','b');

scatter(x2, y4, 'o', 'filled','MarkerFaceColor',[0.5,0,0.5]);

plot(x\_fit, y\_fit1,'color','r','LineWidth',1.5);

plot(x\_fit, y\_fit2,'color',[0,0.5,0.5],'LineWidth',1.5);

plot(x\_fit, y\_fit3,'color','b','LineWidth',1.5);

plot(x\_fit, y\_fit4,'color',[0.5,0,0.5],'LineWidth',1.5);

hold off;

xlabel('时间（单位：小时）','FontSize',12,'FontName','宋体');

ylabel('树冠阴影面积（单位：平方米）','FontSize',12,'FontName','宋体');

title('苹果树树冠阴影面积随季节、时间的变化情况','FontSize',16,'FontName','宋体');

legend('','','','', '春分', '夏至', '秋分', '冬至','FontSize',12,'FontName','宋体');