
从电子自旋窥见量子力学

荒原

2021 年 8 月 10 日

目录

一	理论简介	3
I	引言	3
I.1	量子力学的引入	3
I.2	施特恩-格拉赫实验	4
I.2.1	银原子的 SG 实验	4
I.2.2	序列的银原子 SG 实验	4
I.2.3	序列的银原子 SG 实验与琼斯矩阵的类比	5
II	基本概念	5
II.1	右矢、左矢与算子	5
II.2	结合公理与外积	7
II.3	厄米算子的本征值和矩阵表示	7
II.3.1	厄米算子的本征值	7
II.3.2	矩阵表示	8
II.4	对狄拉克符号的评述	9
II.4.1	物理意义	9
II.4.2	态与波函数的联系	9
III	角动量理论摘要	10
III.1	角动量基本对易关系	11
III.2	升降算子与本征右矢	11
III.3	轨道角动量与球谐函数	14
IV	自旋 $1/2$ 系统	14
IV.1	构造单自旋系统的角动量与泡利矩阵	14
IV.2	多自旋系统	16

IV.2.1 多自旋系统的右矢空间	16
IV.2.2 多自旋系统的角动量	17
二 玻色子系统计算实例	19
V 引言	19
VI 简化的一维海森堡模型	20
VI.1 模型构建与分析	20
VI.2 对称性分析	21
VII 计算方案	25
VII.1 右矢的二进制存储	25
VII.2 角动量算子函数	27
VII.3 哈密顿算子函数	28
三 附录	32
A 哈密顿矩阵计算的早期版本	32
B 参考文献	37

第一部分 理论简介

I

引言

I.1 量子力学的引入

量子力学学习入门大多数起于对黑体辐射中“紫外灾难”的解释。由于卢瑟福提出的原子行星轨道模型并不能用于解释原子的稳定性、化学结构等，玻尔通过构建量子化条件提出了玻尔模型。而在 1925 年，薛定谔写出了量子力学的基础方程之一，即薛定谔方程¹

$$\hat{H}\Psi = i\frac{\partial}{\partial t}\Psi \quad (\text{I.1.1})$$

如果在式 I.1.1 中 \hat{H} 不显含时，式 I.1.1 可以分离变量得到定态薛定谔方程。此时构造 \hat{H} 来求得定态薛定谔方程的解，即可解释很多重要的物理现象。

在 1921 至 1922 年间德国物理学家奥托·施特恩和瓦尔特·格拉赫完成了著名的“施特恩-格拉赫实验”（以下简称“SG 实验”）以证实原子角动量量子化。在银原子的 SG 实验中，由于银原子只有一个价电子，忽略原子实的角动量变化，原子的角动量变化只体现在价电子的角动量变化上。银原子的 SG 实验体现了电子自旋的特殊性质，而该特殊性质并没有经典的宏观物理性质对应，是最能体现量子性质的现象之一。本文的主要参考教材 [1] 便以此为切入点，从矩阵的角度来对量子力学进行展开阐释。

¹在选取合适的单位制下可以认为 $\hbar = 1$ ，下文不再特殊说明。

I.2 施特恩-格拉赫实验

I.2.1 银原子的 SG 实验

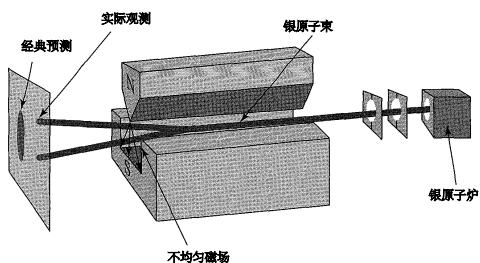


图 1: 银原子 SG 实验装置图

银原子在不均匀的磁场作用下被等分成了两束，若磁场梯度的方向沿 \hat{z} ，记磁场作用为 S_z ，两束银原子分别用 $|S_z; \pm\rangle$ 来表示²，以后约定 $|\pm\rangle = |S_z; \pm\rangle$ 。

I.2.2 序列的银原子 SG 实验

当磁场作用为 S_x 时，银原子同样可以等分为 $|S_x; \pm\rangle$ 两束 (取 \hat{y} 方向为原子束的横向运动方向，这个方向上原子束做匀速直线运动)。用屏阻挡 $|S_x, -\rangle$ ，使 $|S_x, +\rangle$ 的银原子再经过 S_z 作用，最终得到 $|\pm\rangle$ 两束银原子。这似乎给出一束银原子中“含有”四个“分量”的结论，但这个结论无法解释以下现象

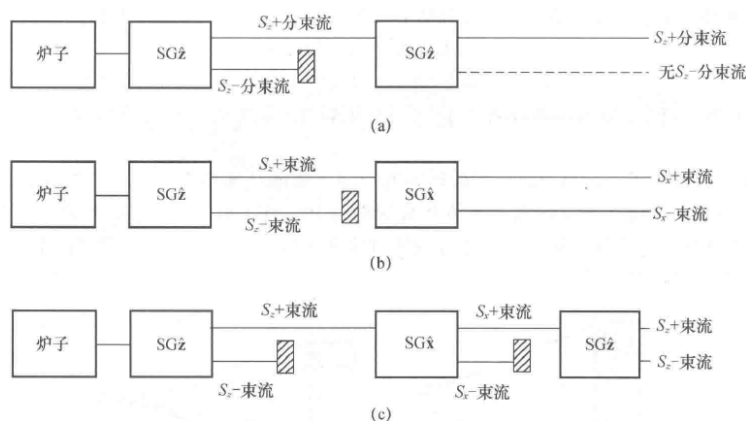


图 2: 序列的银原子 SG 实验示意图

²这里采用的符号为狄拉克符号，稍后将给出更为符合量子力学规范的定义与用法。

在图2.(c) 中我们发现在第一次 \mathcal{S}_z 作用后我们已经用屏将 $|-\rangle$ 完全阻隔，结果中却出现了 $|-\rangle$ 分量。这表明图2.(c) 中的仪器 $\text{SG}_{\hat{x}}$ 破坏了前一个仪器作用的信息。

I.2.3 序列的银原子 SG 实验与琼斯矩阵的类比

对于一束沿 \hat{y} 方向传播的线偏振光，我们可以构造其琼斯矢量

$$E = \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix}, (0 \leq \alpha \leq \frac{\pi}{2})$$

然后让该偏振光依次经过透光轴沿 \hat{x} 方向、与 \hat{x} 夹 45° 方向以及 \hat{z} 方向的偏振片，琼斯矩阵分别为

$$J_x = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, J_{xz} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, J_z = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

作用的结果为

$$E' = J_z J_{xz} J_x E = \begin{bmatrix} 0 \\ \frac{\cos \alpha}{\sqrt{2}} \end{bmatrix}$$

结论指出当 $\alpha \neq \frac{\pi}{2}$ 时， J_{xz} 会破坏 J_x 作用后的偏振信息，这与序列的银原子 SG 实验很类似。

引入复数后，琼斯矩阵与琼斯矢量也可以用来描述光的相位信息，进而描述圆偏振光，这与后文会介绍的泡利矩阵高度类似。

II

基本概念

II.1 右矢、左矢与算子

• 右矢

在量子力学里，我们对物理态进行研究，而这些物理态都是存在于一个希尔伯特空间即右矢空间里的，按保罗·狄拉克的建议，所有的物理态都被写为狄拉克符号下的右矢 (ket)，即用 $|\alpha\rangle$ 来表示矢量。

• 左矢与内积

由线性代数的知识我们定义右矢的对偶矢量**左矢 (bra)**，由左矢构成的空间为**左矢空间**， $c_\alpha |\alpha\rangle$ 的左矢表达为

$$c_\alpha |\alpha\rangle \xleftrightarrow{DC} c_\alpha^* \langle \alpha|, (c_\alpha \text{ 与 } c_\alpha^* \text{ 为共轭复数})$$

由左矢可以定义**内积 (inner product)** ³ $\langle \beta|\alpha\rangle$

对于内积我们还有两条假设

1. $\langle \beta|\alpha\rangle = \langle \alpha|\beta\rangle^*$
2. $\langle \alpha|\alpha\rangle \geq 0$ ，当且仅当 $|\alpha\rangle$ 为零右矢的时候等号成立。

依此可以定义 $|\alpha\rangle$ 的模为 $\sqrt{\langle \alpha|\alpha\rangle}$ ，且可以得到非零右矢的归一化形式

$$|\tilde{\alpha}\rangle = \frac{|\alpha\rangle}{\sqrt{\langle \alpha|\alpha\rangle}}$$

当 $|\beta\rangle \neq |\alpha\rangle$ 且 $\langle \beta|\alpha\rangle = \langle \alpha|\beta\rangle^* = 0$ ，则定义 $|\beta\rangle$ 与 $|\alpha\rangle$ **正交**。

• 算子

位置、动量以及自旋等被称为**可观测量**，可用所涉矢量空间的**算子**来表示⁴。算子从左侧作用到右矢上，作用结果仍是一个右矢，如算子 \mathcal{A} 作用于右矢 $|\alpha\rangle$ 上表达为

$$\mathcal{A}(|\alpha\rangle) = \mathcal{A}|\alpha\rangle = |\tilde{\alpha}\rangle$$

本文中所有算子都是线性算子，可以定义相等，对加法满足交换律、结合律，对乘法满足结合律但不满足交换律。

算子从右侧作用到左矢上

$$(|\beta\rangle)\mathcal{B} = \langle \beta|\mathcal{B} = \langle \tilde{\beta}|$$

从左矢与右矢的对偶关系可以得到

$$\mathcal{X}|\alpha\rangle \xleftrightarrow{DC} \langle \alpha|\mathcal{X}^\dagger$$

³这个符号也被称为 bracket，即 bra(左矢) c(column 代表中间的 |) ket(右矢)。

⁴部分算子由经典力学中的物理量如位置、动量“升格”得到，而自旋算子的特殊性在于其并没有经典物理量对应。

其中 \mathcal{X}^\dagger 为 \mathcal{X} 的伴算子，满足性质 $(\mathcal{X}\mathcal{Y})^\dagger = \mathcal{Y}^\dagger \mathcal{X}^\dagger$ 。

对于满足

$$\mathcal{X} = \mathcal{X}^\dagger$$

的算子被称为自伴算子，有穷维内积空间上的自伴算子也被称为厄米算子，本文暂不涉及无穷维内积空间，以下讨论均使用厄米算子。

II.2 结合公理与外积

狄拉克认为只要合理规定左矢、右矢与算子之间“乘法”的结合性，这种结合性就可以普遍适用，这被称为结合公理。

• 外积

右矢可以“乘以”左矢，即可以存在

$$\mathcal{Z} = |\alpha\rangle\langle\beta| \quad (\text{II.2.1a})$$

$$\mathcal{Z}^\dagger = |\beta\rangle\langle\alpha| \quad (\text{II.2.1b})$$

可以验证式II.2.1a与式II.2.1b满足算子的运算律，因此这种假定是合理的，式II.2.1a即 $|\alpha\rangle$ 与 $\langle\beta|$ 的外积 (outer product)。

• 简化书写

结合公理给出

$$\langle\beta|(\mathcal{X}|\alpha\rangle) = (\langle\beta|\mathcal{X})|\alpha\rangle = \langle\beta|\mathcal{X}|\alpha\rangle$$

以及

$$\langle\alpha|\mathcal{X}^\dagger|\beta\rangle = \langle\beta|\mathcal{X}|\alpha\rangle^*$$

II.3 厄米算子的本征值和矩阵表示

II.3.1 厄米算子的本征值

厄米算子是量子力学中的最重要的算子，由线性代数的结论，厄米算子的本征值一定是实数，且属于不同本征值的本征右矢是正交的，即满足

$$\langle \alpha^i | \alpha^j \rangle = \delta_{ij} \quad (\text{II.3.1})$$

在厄米算子的本征值不出现简并时，不同本征值的本征右矢可以组成完备归一的基右矢即 $\{ | \alpha^i \rangle \}$ 。任意右矢可以表示为基右矢的线性组合

$$| \alpha \rangle = \sum_i c_i | \alpha^i \rangle \quad (\text{II.3.2})$$

由式II.3.1可以得到系数的表达式为

$$c_i = \langle \alpha^i | \alpha \rangle \quad (\text{II.3.3})$$

若按照量子力学的要求认为态 $| \alpha \rangle$ 是归一化的，即满足

$$\langle \alpha | \alpha \rangle = \sum_i | \langle \alpha^i | \alpha \rangle |^2 = \sum_i |c_i|^2 = 1 \quad (\text{II.3.4})$$

将式II.3.3和式II.3.4带入式II.3.2可以得到投影算符

$$\Lambda_i = | \alpha^i \rangle \langle \alpha^i | \quad (\text{II.3.5})$$

以及单位算符

$$\mathcal{I} = \sum_i \Lambda_i = \sum_i | \alpha^i \rangle \langle \alpha^i | \quad (\text{II.3.6})$$

II.3.2 矩阵表示

对于任意厄米算子 \mathcal{A} ，总可以进行如下计算

$$\begin{aligned} \mathcal{A} &= \mathcal{I} \cdot \mathcal{A} \cdot \mathcal{I} \\ &= \sum_i \sum_j | \alpha^i \rangle \langle \alpha^i | \mathcal{A} | \alpha^j \rangle \langle \alpha^j | \\ &= \begin{bmatrix} | \alpha^1 \rangle & \cdots & | \alpha^i \rangle & \cdots \end{bmatrix} \begin{bmatrix} \langle \alpha^1 | \mathcal{A} | \alpha^1 \rangle & \cdots & \langle \alpha^1 | \mathcal{A} | \alpha^i \rangle & \cdots \\ \vdots & \ddots & \vdots & \\ \langle \alpha^i | \mathcal{A} | \alpha^1 \rangle & \cdots & \langle \alpha^i | \mathcal{A} | \alpha^i \rangle & \\ \vdots & & & \ddots \end{bmatrix} \begin{bmatrix} \langle \alpha^1 | \\ \vdots \\ \langle \alpha^i | \\ \vdots \end{bmatrix} \end{aligned} \quad (\text{II.3.7})$$

由式II.3.7可以得到算子 \mathcal{A} 的矩阵表示 A

$$A = \begin{bmatrix} \langle \alpha^1 | \mathcal{A} | \alpha^1 \rangle & \cdots & \langle \alpha^1 | \mathcal{A} | \alpha^i \rangle & \cdots \\ \vdots & \ddots & \vdots & \\ \langle \alpha^i | \mathcal{A} | \alpha^1 \rangle & \cdots & \langle \alpha^i | \mathcal{A} | \alpha^i \rangle & \\ \vdots & & & \ddots \end{bmatrix} \quad (\text{II.3.8})$$

很容易证明厄米算子的矩阵表示一定是厄米矩阵。

这样，我们建立了厄米算子和厄米矩阵之间的同构关系，而右矢也可以建立与列向量之间的对应关系。而线性代数给出的结论是，厄米矩阵一定可以通过酉矩阵进行相似对角化。在计算中，求算子的本征值被转化为了求矩阵的本征值问题。

II.4 对狄拉克符号的评述

II.4.1 物理意义

从线性代数的角度出发，用狄拉克符号表示出的右矢 $|\psi\rangle$ 与欧氏空间内的空间向量 \vec{r} 本质上是一致的。线性算子 \mathcal{A} 对右矢的作用本质上也是向量的线性运算。

在物理上，右矢表示抽象的物理态，算子表示对物理态的测量。量子力学中最重要的厄米算子之一就是哈密顿算子 \mathcal{H} ，将哈密顿算子作用于物理态上可以得到哈密顿算子的本征值与本征态。不含时哈密顿算子的本征值即为对应物理态的能量。

II.4.2 态与波函数的联系

算子的本征值可以是连续的，如规定坐标算子 \mathcal{X} 的本征值与本征态如下

$$\mathcal{X} |x\rangle = x |x\rangle$$

正交关系可以推广为

$$\langle x | x' \rangle = \delta(x - x')$$

那么对态 $|\alpha\rangle$ 展开得到

$$|\alpha\rangle = \int dx |x\rangle \langle x | \alpha \rangle$$

量子力学要求 $\langle \alpha | \alpha \rangle = 1$

可以认为 $\psi_\alpha(x) = \langle x | \alpha \rangle$ ，量子力学的归一化条件可以表述为

$$\int |\psi_\alpha|^2 = \int \psi_\alpha^*(x) \psi_\alpha(x) dx = 1 \quad (\text{II.4.1})$$

这个结论可以推广到高维，在不同的空间坐标系下也适用，即

$$\psi_\alpha(\mathbf{r}) = \langle \mathbf{r} | \alpha \rangle \quad (\text{II.4.2})$$

对于定态，哈密顿算子作用于本征态上可以写为

$$\mathcal{H} | \alpha \rangle = E_\alpha | \alpha \rangle$$

两侧同时左乘 $\langle \mathbf{r} |$ 得到

$$\langle \mathbf{r} | \mathcal{H} | \alpha \rangle = E_\alpha \langle \mathbf{r} | \alpha \rangle$$

即式I.1.1

$$\hat{H} \psi_\alpha = E_\alpha \psi_\alpha$$

III

角动量理论摘要

在量子力学中，角动量是一个更为抽象的概念。从宏观经典的视角切入，三维欧氏空间中的旋转有三个自由度，即绕 x, y, z 三根轴的旋转。有两个容易证明的结论，即

1. 绕三根坐标轴的旋转是对称的，不存在优先的旋转方向。
2. 旋转的先后顺序是不能随意交换的。

我们引入对易子

$$[\mathcal{A}, \mathcal{B}] = \mathcal{A}\mathcal{B} - \mathcal{B}\mathcal{A}$$

来描述两个算子之间的对易关系。很明显 $[\mathcal{A}, \mathcal{B}] = 0$ 时， \mathcal{A}, \mathcal{B} 两算子对易，由线性代数的结论可知，两算子对易意味着它们有相同的本征向量组，可以同时对角化。

III.1 角动量基本对易关系

经典力学下角动量的定义为

$$\mathbf{L} = \mathbf{r} \times \mathbf{p}$$

\mathbf{L} 有三个分量即

$$\mathbf{L} = [L_x \ L_y \ L_z]$$

仿照经典力学的定义可以定义量子力学下的轨道角动量

$$\mathcal{L} = \mathcal{R} \times \mathcal{P} \quad (\text{III.1.1})$$

其中 $\mathcal{R} = [\mathcal{X} \ \mathcal{Y} \ \mathcal{Z}]$, $\mathcal{P} = [-i\nabla_x \ -i\nabla_y \ -i\nabla_z]$

利用列维-奇维塔符号

$$\varepsilon_{ijk} = \begin{cases} 1, & ijk \text{ 为偶排列} \\ -1, & ijk \text{ 为奇排列} \\ 0, & ijk \text{ 中存在相等} \end{cases}$$

可以写出

$$[\mathcal{L}_i, \mathcal{L}_j] = i\varepsilon_{ijk} \mathcal{L}_k \quad (\text{III.1.2})$$

其实在量子力学中，三个满足如上关系的厄米算子都被称为角动量算子，这种关系被称为**角动量基本对易关系**。

角动量算子还满足雅可比恒等式，即

$$[\mathcal{L}_x, [\mathcal{L}_y, \mathcal{L}_z]] + [\mathcal{L}_y, [\mathcal{L}_z, \mathcal{L}_x]] + [\mathcal{L}_z, [\mathcal{L}_x, \mathcal{L}_y]] = 0$$

III.2 升降算子与本征右矢

对于满足基本对易关系的角动量组 $\mathcal{J} = [\mathcal{J}_x \ \mathcal{J}_y \ \mathcal{J}_z]$ ，其具有一些普适的、与角动量具体形式无关的重要结论。

虽然角动量算子的三个分量是对称的，在进行坐标变换的时候，我们会认为 ϕ 角的旋转轴与 z 轴重合，并且在 SG 实验中，我们也认为以磁场梯度方向为 \hat{z} 方向为基础。因此我们对 \mathcal{J}_z 予以特殊的关注。而对 $\mathcal{J}_x, \mathcal{J}_y$ 进行如下变换，定义升降算子

$$\begin{cases} \mathcal{J}_+ = \mathcal{J}_x + i\mathcal{J}_y \\ \mathcal{J}_- = \mathcal{J}_x - i\mathcal{J}_y \end{cases} \quad (\text{III.2.1})$$

$\mathcal{J}_+, \mathcal{J}_-$ 二者都不是厄米算子，但是两者互为伴算子，且 $\mathcal{J}_+\mathcal{J}_-, \mathcal{J}_-\mathcal{J}_+$ 均为厄米算子。且满足关系

$$[\mathcal{J}_+, \mathcal{J}_-] = \mathcal{J}_+\mathcal{J}_- - \mathcal{J}_-\mathcal{J}_+ = 2\mathcal{J}_z \quad (\text{III.2.2})$$

定义角动量平方算子

$$\mathcal{J}^2 = \frac{1}{2}(\mathcal{J}_+\mathcal{J}_- + \mathcal{J}_-\mathcal{J}_+) + \mathcal{J}_z^2 = \mathcal{J}_x^2 + \mathcal{J}_y^2 + \mathcal{J}_z^2 \quad (\text{III.2.3})$$

可以证明

$$[\mathcal{J}^2, \mathcal{J}_z] = 0$$

这意味着二者有共同的本征右矢组，由式III.2.3可知， \mathcal{J}^2 的本征值一定为非负数，所以可以设 \mathcal{J}^2 与 \mathcal{J}_z 的共同本征右矢为 $|j, m\rangle, j \geq 0$ ，有

$$\mathcal{J}^2 |j, m\rangle = j(j+1) |j, m\rangle \quad (\text{III.2.4a})$$

$$\mathcal{J}_z |j, m\rangle = m |j, m\rangle \quad (\text{III.2.4b})$$

升降算子的“升降”二字体现在如下关系中

$$[\mathcal{J}_z, \mathcal{J}_\pm] = \pm \mathcal{J}_\pm \quad (\text{III.2.5})$$

$$\mathcal{J}_z \mathcal{J}_\pm = \mathcal{J}_\pm (\mathcal{J}_z \pm \mathcal{I}) \quad (\text{III.2.6})$$

在式III.2.6两边同时右乘 $|j, m\rangle$ 得到

$$\begin{aligned}\mathcal{J}_z \mathcal{J}_\pm |j, m\rangle &= \mathcal{J}_\pm (\mathcal{J}_z \pm \mathcal{I}) |j, m\rangle \\ &= (m \pm 1) \mathcal{J}_\pm |j, m\rangle \\ &= (m \pm 1) |j, m \pm 1\rangle\end{aligned}$$

上式表明升降算子可以将 \mathcal{J}_z 本征值为 m 的本征右矢变换为本征值为 $m + 1$ 的本征右矢。

由式III.2.2和式III.2.3可以得到如下关系

$$\begin{aligned}\mathcal{J}_+ \mathcal{J}_- &= \mathcal{J}^2 - \mathcal{J}_z^2 + \mathcal{J}_z \\ \mathcal{J}_- \mathcal{J}_+ &= \mathcal{J}^2 - \mathcal{J}_z^2 - \mathcal{J}_z\end{aligned}$$

由本征右矢不为 0 可以得到

$$\begin{aligned}\langle j, m-1 | j, m-1 \rangle &= \langle j, m | \mathcal{J}_+ \mathcal{J}_- | j, m \rangle = j(j+1) - m^2 + m \geq 0 \\ \langle j, m+1 | j, m+1 \rangle &= \langle j, m | \mathcal{J}_- \mathcal{J}_+ | j, m \rangle = j(j+1) - m^2 - m \geq 0\end{aligned}$$

解得

$$-j \leq m \leq j \quad (\text{III.2.9})$$

m 的最大值必然取 j ，否则有 $j-1 < m_{\max} < j$ 使得 $\langle j, m+1 | j, m+1 \rangle < 0$ 。同理， m 的最小值必然取 $-j$ 。此时有

$$\mathcal{J}_\pm |j, \pm j\rangle = 0$$

因此我们得到 $m_{\max} - m_{\min} = 2j = k, k \in \mathbb{N}$ ，即 j 一定为非负整数或者正半奇数， m 仅且一定取遍 $-j, -j+1, \dots, j$ 。

上述结论不涉及角动量算子的具体形式，为量子力学中角动量的普适结论，这也是量子力学中态是不连续的直接体现。

III.3 轨道角动量与球谐函数

量子力学的结论进一步指出, 当 j 为非负整数时角动量是有经典力学的物理量对应的, 式III.1.1给出的轨道角动量 \mathcal{L} 即对应三维坐标空间中的旋转。在式III.2.4a与式III.2.4b两边同时乘以球坐标下的本征左矢可以得到球谐函数

$$\begin{cases} \langle \mathbf{r} | \mathcal{L}^2 | l, m \rangle = l(l+1) \langle \mathbf{r} | l, m \rangle \\ \langle \mathbf{r} | \mathcal{L}_z | l, m \rangle = m \langle \mathbf{r} | l, m \rangle \end{cases}$$

$$\Downarrow$$

$$\begin{cases} - \left[\frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} + \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \right) \right] Y_l^m(\theta, \varphi) = l(l+1) Y_l^m(\theta, \varphi) \\ -i \frac{\partial}{\partial \varphi} Y_l^m(\theta, \varphi) = m Y_l^m(\theta, \varphi) \end{cases}$$

当 j 取正半奇数时, \mathcal{J} 为自旋角动量, 没有经典力学的物理量对应, 下一章会详细讨论。

IV

自旋 1/2 系统

IV.1 构造单自旋系统的角动量与泡利矩阵

我们在寻找算子的矩阵表示时发现式II.3.8中的

$$\langle \alpha^i | \mathcal{A} | \alpha^j \rangle = \alpha^j \delta_{ij}$$

那么算子可以表示为

$$\mathcal{A} = \sum_i \alpha_i | \alpha^i \rangle \langle \alpha^i | = \sum_i \alpha_i \Lambda_i \quad (\text{IV.1.1})$$

单自旋 $\frac{1}{2}$ 系统的本征向量可以被表示为已归一化的 $| \pm \rangle$, 可以构造单位算子

$$\mathcal{I} = | + \rangle \langle + | + | - \rangle \langle - |$$

由于需要满足IV.1.1要求, 梯度沿 \hat{z} 方向的磁场作用可以构造为

$$\mathcal{S}_z = \frac{1}{2} (| + \rangle \langle + | - | - \rangle \langle - |) \quad (\text{IV.1.2})$$

同时构造升降算子

$$\mathcal{S}_+ = |+\rangle\langle -| \quad (\text{IV.1.3a})$$

$$\mathcal{S}_- = |- \rangle\langle +| \quad (\text{IV.1.3b})$$

由图2.(b) 可知

$$\begin{aligned} |\langle +|\mathcal{S}_x; + \rangle|^2 + |\langle -|\mathcal{S}_x; + \rangle|^2 &= 1 \\ |\langle +|\mathcal{S}_x; + \rangle|^2 &= |\langle -|\mathcal{S}_x; + \rangle|^2 \end{aligned} \quad (\text{IV.1.4})$$

再由正交归一化条件构造

$$\begin{aligned} |\mathcal{S}_x; + \rangle &= \frac{1}{\sqrt{2}} (|+\rangle + e^{i\delta_1} |- \rangle) \\ |\mathcal{S}_x; - \rangle &= \frac{1}{\sqrt{2}} (|+\rangle - e^{i\delta_1} |- \rangle) \end{aligned}$$

那么梯度沿 \hat{x} 方向的磁场作用可以写为

$$\mathcal{S}_x = \frac{1}{2} (e^{-i\delta_1} |+\rangle\langle -| + e^{i\delta_1} |- \rangle\langle +|) \quad (\text{IV.1.5})$$

同理可以构造

$$\begin{aligned} |\mathcal{S}_y; + \rangle &= \frac{1}{\sqrt{2}} (|+\rangle + e^{i\delta_2} |- \rangle) \\ |\mathcal{S}_y; - \rangle &= \frac{1}{\sqrt{2}} (|+\rangle - e^{i\delta_2} |- \rangle) \\ \mathcal{S}_y &= \frac{1}{2} (e^{-i\delta_2} |+\rangle\langle -| + e^{i\delta_2} |- \rangle\langle +|) \end{aligned} \quad (\text{IV.1.6})$$

与式IV.1.4对比可以写出

$$\begin{aligned} |\langle \mathcal{S}_y; +|\mathcal{S}_x; + \rangle|^2 + |\langle \mathcal{S}_y; -|\mathcal{S}_x; + \rangle|^2 &= 1 \\ |\langle \mathcal{S}_y; +|\mathcal{S}_x; + \rangle|^2 &= |\langle \mathcal{S}_y; -|\mathcal{S}_x; + \rangle|^2 \end{aligned} \quad (\text{IV.1.7})$$

若取 $\delta_1 = 0$, 有 $e^{i\delta_2} = \pm i$, 在取定右手坐标系, 同时要求式IV.1.3a IV.1.3b满足式III.2.1定义时, 取定 $e^{i\delta_2} = i$ 。

最终我们得到 $\mathcal{S} = [\mathcal{S}_x \ \mathcal{S}_y \ \mathcal{S}_z]$

$$\begin{aligned}\mathcal{S}_x &= \frac{1}{2} \left(|+\rangle\langle-| + |- \rangle\langle+| \right) \\ \mathcal{S}_y &= \frac{i}{2} \left(-|+\rangle\langle-| + |- \rangle\langle+| \right) \\ \mathcal{S}_z &= \frac{1}{2} \left(|+\rangle\langle+| - |- \rangle\langle-| \right)\end{aligned}\tag{IV.1.8}$$

以及升降算子

$$\begin{aligned}\mathcal{S}_+ &= \mathcal{S}_x + i\mathcal{S}_y = |+\rangle\langle-| \\ \mathcal{S}_- &= \mathcal{S}_x - i\mathcal{S}_y = |- \rangle\langle+|\end{aligned}\tag{IV.1.9}$$

$\mathcal{S} = [\mathcal{S}_x \ \mathcal{S}_y \ \mathcal{S}_z]$ 是厄米的，且满足角动量基本对易关系式III.1.2，即

$$[\mathcal{S}_i, \mathcal{S}_j] = i\varepsilon_{ijk}\mathcal{S}_k$$

因此， \mathcal{S} 即单自旋 $\frac{1}{2}$ 系统的**自旋角动量**。

在 \mathcal{S} 的分量两边同时乘以单位算子，可以得到得到**泡利矩阵**

$$\sigma_x = \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_y = \frac{1}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \sigma_z = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\tag{IV.1.10}$$

IV.2 多自旋系统

符号约定

在讨论多自旋系统前先进行一个符号约定，在不至于引起混淆的情况下，下标 i 表示第 i 个角动量、态等，上标表示相应角动量、态等的分量。如 \mathcal{S}_2^z 表示作用于态 $|\psi_2\rangle$ 上的自旋角动量的 z 分量。

IV.2.1 多自旋系统的右矢空间

在引入多个自旋相互用之后，可以认为每一个角动量算子都只作用在对应的态上，而不作用于其他的态，角动量的基本对易关系也可以表达为

$$[\mathcal{S}_m^i, \mathcal{S}_n^j] = i\varepsilon_{ijk}\delta_{mn}\mathcal{S}_m^k\tag{IV.2.1}$$

上式表明了下标不同的 \mathcal{S}_i^z 之间是相容的，这表明了它们有共同的本征右矢。即自旋角动量可以进行线性叠加，而叠加后的角动量作用的希尔伯特空间可以写成每个自旋态的直积^[2]，即

$$|\Psi\rangle = |\Psi_1\rangle \otimes |\Psi_2\rangle \otimes \cdots \otimes |\Psi_L\rangle$$

对于自旋和自旋角动量来说，本征向量组可以写为如下形式

$$\text{共 } 2^L \text{ 个} \left\{ \begin{array}{l} \overbrace{|\ +\ +\ \cdots\ +\ \rangle}^{\text{共 } L \text{ 个}} \\ | \ +\ +\ \cdots\ -\ \rangle \\ \vdots \\ | \ -\ -\ \cdots\ -\ \rangle \end{array} \right. \quad (\text{IV.2.2})$$

式IV.2.2表明了随着自旋态的增多，用于描述自旋的本征向量是成指数规律增加的，这为与自旋相关的计算带来了不小的困难，这意味着我们需要寻求更多的物理关系来去除冗余的部分使得计算变得简化。

IV.2.2 多自旋系统的角动量

多自旋系统的角动量可以表示为所有的自旋角动量的和，即

$$\begin{aligned} \mathcal{S}^i &= \mathcal{S}_1^i \otimes \mathcal{I}_2 \otimes \cdots \otimes \mathcal{I}_L \\ &\quad + \mathcal{I}_1 \otimes \mathcal{S}_2^i \otimes \cdots \otimes \mathcal{I}_L \\ &\quad \vdots \\ &\quad + \mathcal{I}_1 \otimes \mathcal{I}_2 \otimes \cdots \otimes \mathcal{S}_L^i \end{aligned}$$

可以简化为

$$\mathcal{S}^i = \sum_m \mathcal{S}_m^i \quad (\text{IV.2.3})$$

这种方式定义的角动量，结合IV.2.1，是能够满足角动量基本对易关系，即式III.1.2的。升降算子的定义与式IV.1.9一致，角动量平方算子的定义也与前文保持一致，即

$$\mathcal{S}^2 = \frac{1}{2}(\mathcal{S}^+ \mathcal{S}^- + \mathcal{S}^- \mathcal{S}^+) + (\mathcal{S}^z)^2 \quad (\text{IV.2.4})$$

也有

$$[\mathcal{S}^2, \mathcal{S}^z] = 0 \quad (\text{IV.2.5})$$

我们发现 $\mathcal{S}^z |++\cdots+\rangle = \frac{L}{2} |++\cdots+\rangle$, 表明 $j = m_{\max} = \frac{L}{2}$

很明显 L 的奇偶性决定了系统的对称性, 当 L 为偶数的时候, 存在 $m = 0$ 的能量最低态, 该系统为**玻色子系统**, 后文将对一个玻色子系统进行简单的分析。

当 L 为奇数的时候, 系统能量最低态是二重简并的, 且能量高于玻色子系统, 此系统为**费米子系统**。

第二部分 玻色子系统计算实例

V

引言

前文构建多自旋系统时，我们发现当需要考虑的自旋个数增多时，右矢空间的维度会与自旋个数成指数关系增加。而实际上在考虑两个自旋相互作用时，角动量算子对应的矩阵就已经是 4×4 的矩阵了，为此我们不得求助于计算机程序来解算或近似解算我们所需的数据。

本部分将构建简化的一维海森堡模型，给出并简要分析由对称性所决定的简化方法。还将给出一个多自旋系统的数据存储方案，并给出一些数值计算结果。本部分的主要参考文献为 [2]，下文均为玻色子系统。

笔者在测试时使用的编程语言为 C++，测试环境为 Ubuntu 20.04LTS 下集成开发环境 CLion 2021.1.3x64，编译器为 GCC 9.3.0，科学计算模块为 GSL-2.7。

VI

简化的一维海森堡模型

VI.1 模型构建与分析

简化的一维海森堡模型只考虑相邻自旋的相互作用，且作用系数为 1，开放边界条件。所以对于一个维自旋 $\frac{1}{2}$ 链，哈密顿算子写为如下形式

$$\begin{aligned}
 \mathcal{H} &= \sum_{i=1}^{L-1} \mathcal{S}_i \cdot \mathcal{S}_{i+1} \\
 &= \sum_{i=1}^{L-1} \left(\mathcal{S}_i^x \mathcal{S}_{i+1}^x + \mathcal{S}_i^y \mathcal{S}_{i+1}^y + \mathcal{S}_i^z \mathcal{S}_{i+1}^z \right) \\
 &= \sum_{i=1}^{L-1} \left[\frac{1}{2} (\mathcal{S}_i^+ \mathcal{S}_{i+1}^- + \mathcal{S}_i^- \mathcal{S}_{i+1}^+) + \mathcal{S}_i^z \mathcal{S}_{i+1}^z \right]
 \end{aligned} \tag{VI.1.1}$$

式VI.1.1中使用了升降算子，即式IV.1.9来进行构造，这样做的好处是，哈密顿算子中的所有算子都是对 \mathcal{S}^z 的本征右矢定义的，而且不会引入虚数。由于构成 \mathcal{H} 的算子均是厄米算子， \mathcal{H} 也是厄米算子。

计算的任务是寻找哈密顿算子的本征值和本征态，本征态是系统可以稳定存在的态，而本征值系统处在该态下的能量。

我们可以通过式II.3.8哈密顿矩阵 H ，此处给出 4 个自旋相互作用的矩阵表达，为 16×16 的矩阵

$$\begin{bmatrix}
 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0.25 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0.5 & -0.25 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0.25 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0.5 & 0 & -0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0.5 & 0 & -0.75 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.5 & -0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & -0.75 & 0 & 0.5 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0 & 0.5 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.25 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & -0.25 & 0.5 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.25 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.75
 \end{bmatrix} \tag{VI.1.2}$$

利用 GSL 的本征值计算器，可以得到这个矩阵的本征值如下

$$\begin{array}{cccccccc} 0.75 & -0.95711 & -0.25 & & 0.75 & 0.457107 & -1.61603 & 0.116025 & -0.95711 \\ -0.95711 & & 0.75 & 0.75 & 0.457107 & 0.457107 & & -0.25 & -0.25 & 0.75 \end{array}$$

可以看出，有很多本征值是重复的，说明哈密顿算子存在能量相同的简并态。

VI.2 对称性分析

如果对于 \mathcal{H} 可以找到 $[\mathcal{H}, \mathcal{A}] = 0$ ，意味着 \mathcal{A} 和 \mathcal{H} 有共同的本征右矢，如果 \mathcal{A} 的本征值与本征右矢是容易确定的，那么可以通过 \mathcal{A} 的本征值将 \mathcal{H} 化为分块对角矩阵。

在上述模型中代入III.2.5

$$\begin{aligned} [\mathcal{H}, \mathcal{S}^z] &= \sum_i \sum_j \left[\frac{1}{2} (\mathcal{S}_i^+ \mathcal{S}_{i+1}^- + \mathcal{S}_i^- \mathcal{S}_{i+1}^+) + \mathcal{S}_i^z \mathcal{S}_{i+1}^z, \mathcal{S}_j^z \right] \\ &= \sum_i \left[\frac{1}{2} (\mathcal{S}_i^+ \mathcal{S}_{i+1}^- + \mathcal{S}_i^- \mathcal{S}_{i+1}^+), \mathcal{S}_i^z + \mathcal{S}_{i+1}^z \right] \\ &= \frac{1}{2} \sum_i \left(\mathcal{S}_i^+ \mathcal{S}_{i+1}^- - \mathcal{S}_i^- \mathcal{S}_{i+1}^+ - \mathcal{S}_i^+ \mathcal{S}_{i+1}^- + \mathcal{S}_i^- \mathcal{S}_{i+1}^+ \right) \\ &= 0 \end{aligned} \tag{VI.2.1}$$

由对称性可知 $[\mathcal{H}, \mathcal{S}^x] = [\mathcal{H}, \mathcal{S}^y] = 0$ ，那么

$$[\mathcal{H}, \mathcal{S}^+] = 0 \tag{VI.2.2a}$$

$$[\mathcal{H}, \mathcal{S}^-] = 0 \tag{VI.2.2b}$$

由式IV.2.4也很容易证明

$$[\mathcal{H}, \mathcal{S}^2] = 0 \tag{VI.2.3}$$

对于 S^z 的本征值为 m 的第 i 个本征右矢 $|\varphi_m^i\rangle$ 有

$$\begin{aligned} S^z |\varphi_m^i\rangle &= m |\varphi_m^i\rangle \\ S^z \mathcal{H} |\varphi_m^i\rangle &= \mathcal{H} S^z |\varphi_m^i\rangle \\ &= m \mathcal{H} |\varphi_m^i\rangle \end{aligned} \quad (\text{VI.2.4})$$

表明 $\mathcal{H} |\varphi_m^i\rangle$ 也是 S^z 的本征右矢。我们可以把哈密顿矩阵写为 S^z 的本征右矢表象下的形式，即

$$H_{mn}^{ij} = \langle \varphi_m^i | \mathcal{H} | \varphi_n^j \rangle$$

注意到

$$\langle \varphi_m^i | \varphi_n^j \rangle = \delta_{ij} \delta_{mn}$$

表明 $m \neq n$ 时，哈密顿矩阵元一定为 0。而对于 S^z ，其本征值和本征右矢均已知，即式IV.2.2。本征右矢可以按照如下的规则取定^{6 7}

$$\begin{aligned} &\text{有且只有 } 0 \text{ 个自旋方向为 } - \quad |++\cdots+\rangle \\ &\text{有且只有 } 1 \text{ 个自旋方向为 } - \quad \left\{ \begin{array}{l} | - + \cdots + \rangle \\ | + - \cdots + \rangle \\ \vdots \\ | + + \cdots - \rangle \end{array} \right. , \text{ 共 } L \text{ 个} \\ &\vdots \\ &\text{有且只有 } l, (0 \leq l \leq L) \text{ 个自旋方向为 } -, \text{ 共 } \binom{L}{l} \text{ 个} \\ &\vdots \\ &\text{有且只有 } L \text{ 个自旋方向为 } - \quad | - - \cdots - \rangle \end{aligned}$$

哈密顿矩阵将成分块对角的形式，空白均为 0 元

⁶玻色子系统，偶数个自旋， L 为偶数。

⁷具体取定规则会在下一节给出。

m	$\frac{L}{2}$	$\frac{L}{2} - 1$	\dots	0	\dots	$-\frac{L}{2}$
$\frac{L}{2}$	$M_{\frac{L}{2}}$					
$\frac{L}{2} - 1$		$M_{\frac{L}{2}-1}$				
\vdots			\ddots			
0				M_0		
\vdots					\ddots	
$-\frac{L}{2}$						$M_{-\frac{L}{2}}$

(VI.2.5)

依然以 4 个自旋相互作用为例，式VI.1.2将变换为

0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.25	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.5	-0.25	0.5	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.5	-0.25	0.5	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.5	0.25	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.25	0.5	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.5	-0.75	0.5	0.5	0	0	0	0	0	0	0
0	0	0	0	0	0	0.5	-0.25	0	0.5	0	0	0	0	0	0
0	0	0	0	0	0	0.5	0	-0.25	0.5	0	0	0	0	0	0
0	0	0	0	0	0	0	0.5	0.5	-0.75	0.5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.5	0.25	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.25	0.5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.5	-0.25	0.5	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.5	-0.25	0.5	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	-0.25	0.5
0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.25	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75

(VI.2.6)

在式VI.2.5中我们已经把哈密顿算子的本征态分到了不同的子空间，即算子 $\mathcal{M}_{\frac{L}{2}}, \mathcal{M}_{\frac{L}{2}-1}, \dots, \mathcal{M}_{-\frac{L}{2}}$ 的本征空间中。现在我们来研究不同子空间之间的本征态的能量关系。用 $|\psi_m^i\rangle$ 来表示处于 \mathcal{M}_m 本征空间中、哈密顿算子的第 i 个本征态。由式VI.2.2a

$$\begin{aligned}
 \mathcal{H}\mathcal{S}^+|\psi_m^i\rangle &= \mathcal{S}^+\mathcal{H}|\psi_m^i\rangle \\
 &= E_m^i\mathcal{S}^+|\psi_m^i\rangle \\
 &= E_m^i|\psi_{m+1}^i\rangle \\
 &= E_{m+1}^i|\psi_{m+1}^i\rangle
 \end{aligned}$$

令升算符从 $\mathcal{M}_{-\frac{L}{2}}$ 的本征空间开始作用，我们发现哈密顿算子在 $\mathcal{M}_{-\frac{L}{2}+1}$ 的本征空间的本征值一定包含在 $\mathcal{M}_{-\frac{L}{2}}$ 的本征空间中的本征值。于是我们发现哈密顿算子在 \mathcal{M}_0 的本

征空间 (式VI.2.6中间最大的块) 下的本征值包含了 \mathcal{H} 所有的本征值, 这个空间是 $(\frac{L}{2})$ 维的, 运算得以被简化。

在 \mathcal{M}_0 的本征空间里我们发现 \mathcal{S}^z 算子的本征值为 0, 那么可以认为在 \mathcal{M}_0 的本征空间内 $\mathcal{S}^z = \mathcal{O}$ 。那么有

$$\begin{aligned} [\mathcal{S}^+, \mathcal{S}^-] &= 0 \\ \mathcal{S}^2 &= \frac{1}{2}(\mathcal{S}^+\mathcal{S}^- + \mathcal{S}^-\mathcal{S}^+) + (\mathcal{S}^z)^2 \\ &= \mathcal{S}^+\mathcal{S}^- \end{aligned} \quad (\text{VI.2.7})$$

\mathcal{S}^2 的本征值为 $s(s+1)$, $0 \leq s \leq \frac{L}{2}$, 那么可以利用 $\mathcal{S}^+\mathcal{S}^-$ 的矩阵将基 $|\varphi_0^i\rangle$ 变换到⁸ $|j, s, 0\rangle$ 。这样在基 $|j, s, 0\rangle$ 的表征下, 矩阵 M_0 也可以类似式VI.2.5被表示为分块对角的形式

s	0	1	\dots	$\frac{L}{2}$
0	S_0			
1		S_1		
\vdots			\ddots	
$\frac{L}{2}$				$S_{\frac{L}{2}}$

(VI.2.8)

哈密顿矩阵的本征值与本征右矢的计算可以得到进一步的简化。

⁸指第 j 个本征值为 $s(s+1)$ 的本征矢量。

VII

计算方案

VII.1 右矢的二进制存储

从式IV.2.2中得到启发，考虑 L 个自旋相互作用时，右矢空间的维度为 2^L 维。每个自旋有两个态，为 S_i^z 的两个本征右矢，这提示二进制的使用。如果直接用 0 替换式IV.2.2中的 +，用 1 替换 -，便可以得到一种用无符号二进制整数来表示自旋态的方法。

下文均以 4 个自旋相互作用为例

$$| - - - - \rangle \doteq 0000_{(2)} = 0_{(10)}$$

$$| - - - + \rangle \doteq 0001_{(2)} = 1_{(10)}$$

$$| - - + - \rangle \doteq 0010_{(2)} = 2_{(10)}$$

$$| - - + + \rangle \doteq 0011_{(2)} = 3_{(10)}$$

在 C++ 中可以构造 Ket 对象来完成对右矢的存储

```

1 #include <gsl/gsl_vector.h>
2 class Ket {
3 public:
4     unsigned long n; //自旋的个数，对应存在  $2^n$  个基右矢
5     gsl_vector *v; //用于存储系数的向量，向量下标的二进制表示即为本征右矢
6     Ket(unsigned long n){
7         this->n = n;
8         this->v = gsl_vector_calloc(1<<n); //左移运算符用于快速运算  $2^n$ 
9     }
10    Ket(Ket &k){ //深度复制
11        this->n = k.n;
12        this->v = gsl_vector_calloc(1<<n);
13        gsl_vector_memcpy(v,k.v);
14    }
15    Ket(unsigned long n, gsl_vector* v){
16        this->n = n;

```

```

17     this->v = gsl_vector_calloc(1<<n);
18     gsl_vector_memcpy(this->v,v);
19 }
20 ~Ket(){//在 GSL 科学计算包使用过程中请注意析构
21     gsl_vector_free(v);
22 }
23 unsigned long getLength(){
24     return 1 << n;
25 }
26 double get(unsigned long i){
27     return gsl_vector_get(this->v,i);
28 }
29 void set(unsigned long i, double value){
30     gsl_vector_set(this->v,i,value);
31 }
32 //一些基本的右矢运算
33 void add(const Ket *k){
34     gsl_vector_add(this->v,k->v);
35 }
36 void sub(const Ket *k){
37     gsl_vector_sub(this->v,k->v);
38 }
39 void multiple(double k){
40     gsl_vector_scale(this->v,k);
41 }
42 };

```

现在可以给出一个寻找式VI.2.4中本征右矢的算法。在有 L 个自旋的时候，有 l 个自旋为 $+$ ，此时 $m = -\frac{L}{2} + l$ 。以 $L = 6, l = 2$ 为例

```

1 unsigned long head = 1 << l; //第一个符合要求的右矢是 000011(2)，检索从 000100(2) 开始
2 unsigned long tail = (head-1) << (L-l); //最后一个符合要求的右矢是 110000(2)
3 for (; head <= tail; head++) {
4     unsigned long c = 0;
5     for(int j = 0; j<L; j++){//用于计算自旋为 + 的个数
6         if((1<<j)&head)
7             c++;

```

```

8     }
9     if(c==1); //筛选出了所需要的右矢
10 }

```

VII.2 角动量算子函数

角动量算子 S^z 以及升降算子 S^+, S^- 的函数如下

```

1 Ket* spinZ(Ket &k){ //角动量算子  $S^z$ 
2     gsl_vector *temp = gsl_vector_calloc(k.getLength());
3     for (unsigned long i = 0; i < k.getLength(); i++){
4         double c = 0;
5         for (unsigned long j = 0; j < k.n; j++){
6             if ((1 << j) & i){
7                 c+=0.5;
8             }else{
9                 c-=0.5;
10            }
11        }
12        gsl_vector_set(temp,i,gsl_vector_get(k.v,i)*c);
13    }
14    Ket *r = new Ket(k.n, temp);
15    gsl_vector_free(temp);
16    return r;
17 }
18 Ket* spinRaise(Ket &k){ //升算子  $S^+$ 
19     gsl_vector *temp = gsl_vector_calloc(k.getLength());
20     for (unsigned long i = 0; i < k.getLength(); i++){
21         for (unsigned long j = 0; j < k.n; j++){
22             unsigned long t = 1 << j;
23             if (!(t & i)){
24                 gsl_vector_set(temp,i+t,gsl_vector_get(temp,i+t)+
25                     gsl_vector_get(k.v,i));
26             }
27         }
28     }
29 }

```

```

28     Ket *r = new Ket(k.n, temp);
29     gsl_vector_free(temp);
30     return r;
31 }
32 Ket* spinLower(Ket &k){//降算子  $S^-$ 
33     gsl_vector *temp = gsl_vector_calloc(k.getLength());
34     for (unsigned long i = 0; i < k.getLength(); i++){
35         for (unsigned long j = 0; j < k.n; j++){
36             unsigned long t = 1 << j;
37             if (t & i){
38                 gsl_vector_set(temp,i-t,gsl_vector_get(temp,i-t)+
39                     gsl_vector_get(k.v,i));
40             }
41         }
42     }
43     Ket *r = new Ket(k.n, temp);
44     gsl_vector_free(temp);
45     return r;
46 }

```

VII.3 哈密顿算子函数

构造计算哈密顿算子的函数如下

```

1 Ket* hamiltonian(Ket &k){
2     gsl_vector *temp = gsl_vector_calloc(k.getLength());
3     for (unsigned long i = 0; i < k.getLength(); i++){
4         double t = 0;
5         for (unsigned long j = 0; j < k.n-1; j++){
6             if (((1<<j)&i)&&((1<<(j+1)&i)==0)){
7                 gsl_vector_set(temp,i-(1<<j)+(1<<(j+1)), gsl_vector_get(temp,i-
8                     (1<<j)+(1<<(j+1)))+0.5*k.get(i));
9             }else if((((1<<j)&i)==0)&&(1<<(j+1)&i)){
10                gsl_vector_set(temp,i+(1<<j)-(1<<(j+1)), gsl_vector_get(temp,i+
11                    (1<<j)-(1<<(j+1)))+0.5*k.get(i));
12            }else;
13        }
14    }
15    Ket *r = new Ket(k.n, temp);
16    gsl_vector_free(temp);
17    return r;
18 }

```

```

11         if(((1<=j)&i)&&((1<=(j+1))&i)){
12             t += 0.25;
13         }else if (!((1<=j)&i)&&!((1<=(j+1))&i))){
14             t += 0.25;
15         }else{
16             t += -0.25;
17         }
18     }
19     gsl_vector_set(temp,i, gsl_vector_get(temp,i)+t*k.get(i));
20 }
21 Ket *r = new Ket(k.n, temp);
22 gsl_vector_free(temp);
23 return r;
24 }

```

用于获取式VI.2.5中矩阵 M_0 的算法为

```

1 #include <gsl/gsl_matrix.h>
2 #include <gsl/gsl_blas.h>
3 gsl_matrix* getSimplifiedHamiltonian(unsigned long n){
4     size_t getCombination(size_t n, size_t k);
5     double innerProduct(Ket &k1,Ket &k2);
6     unsigned long l = (n+1)/2;
7     unsigned long head = 1 << l;
8     unsigned long tail = (head-1) << (n-l);
9     unsigned long length = getCombination(n,l);
10    static gsl_matrix *r = gsl_matrix_calloc(length,length);
11    /* 用于构造简化矩阵的右矢组 */
12    gsl_vector_ulong *v = gsl_vector_ulong_alloc(length);
13    unsigned long p = 1;
14    gsl_vector_ulong_set(v,0,head-1);
15    for (; head < tail; head++) {
16        unsigned long c = 0;
17        for(int j = 0; j<n; j++){
18            if((1<=j)&head)
19                c++;
20        }

```

```

21     if(c==l){
22         gsl_vector_ulong_set(v,p,head);
23         p++;
24     }
25 }
26 gsl_vector_ulong_set(v,p,tail);
27 /*****
28  /* 根据 $\langle \varphi_0^i | \mathcal{H} | \varphi_0^j \rangle$  构造矩阵 */
29 for (unsigned long i = 0; i < length; i++) {
30     for (unsigned long j = 0; j < length; j++) {
31         Ket *ki = new Ket(n);
32         Ket *kj = new Ket(n);
33         ki->set(gsl_vector_ulong_get(v,i),1);
34         kj->set(gsl_vector_ulong_get(v,j),1);
35         Ket *k = hamiltonian(*ki);
36         gsl_matrix_set(r,j,i, innerProduct(*kj,*k));
37         delete ki;
38         delete kj;
39         delete k;
40     }
41 }
42 /*****
43 gsl_vector_ulong_free(v);
44 return r;
45 }
46 size_t getCombination(size_t n, size_t k){//用于计算组合数
47     if(k > n|k < 0)
48         return 0;
49     size_t l = k<(n-k)?k:(n-k);
50     size_t up = 1, lo = 1;
51     for (int i = 0 ; i < l ; i++){
52         up = up * (n-i);
53         lo = lo * (1+i);
54     }
55     return up/lo;
56 }

```

```
57 double innerProduct(Ket &k1,Ket &k2){//用于计算右矢的内积
58     double r = 0;
59     gsl_blas_ddot(k1.v,k2.v, &r);
60     return r;
61 }
```

注:

以上程序由面向对象的编程思想写成, 右矢被构造为 *Ket* 对象, 每一个 *Ket* 对象中包含 S^z 的所有本征右矢, 并由一个 *gsl_vector_ulong* 来进行存储。这样虽然能完备地描述右矢空间, 并且准确地描述哈密顿算子和哈密顿矩阵的构造过程, 却带来了额外的空间和时间消耗。在附录A中给出了一个用于本文早期测试的 C 语言程序, 该程序计算矩阵 H 和 M_0 , 由于不采用面向对象编程的方式, 计算速度要更快, 但在编写上没有完全符合物理规则。

附录

A

哈密顿矩阵计算的早期版本

```

1 #include <gsl/gsl_matrix.h>
2 #include <gsl/gsl_vector.h>
3 /*Struct ket
4 *k indicates the coefficient.
5 *dimension indicates the dimensions of the bases.
6 *base indicates the base vector by using binary method.
7 *example:
8 *3(10)=11(2), with dimension of 2 and k of 0.25, this ket is like:
9 *1/4|1,1>
10 */
11 struct ket {
12     double k = 1;
13     unsigned long base = 0;
14     unsigned long dimension = 1;
15 };
16 /*
17 *Construct a ket.
18 */
19 void getKet(double k, unsigned long base, unsigned long n, ket *r) {
20     r->k = k;
21     r->base = base;
22     r->dimension = n;
23 }
24 /*
25 *Construct a copy from a known ket.
26 */
27 void copy(ket *s1, ket *s2) {

```

```

28     s2->k = s1->k;
29     s2->base = s1->base;
30     s2->dimension = s1->dimension;
31 }
32 /*
33 *X component of Hamiltonian, indicates Sx1Sx2.
34 */
35 void spinx(ket *s, unsigned long n1, unsigned long n2) {
36     s->k = (s->k) / 4;
37     unsigned long n = s->base;
38     n = (1 << n2) ^ ((1 << n1) ^ n);
39     s->base = n;
40 }
41 /*
42 *The coefficient after Sy twice will still be a real number.
43 */
44 void spiny(ket *s, unsigned long n1, unsigned long n2) {
45     s->k = -(s->k) / 4;
46     unsigned long n = s->base;
47     if (((1 << n1) & n) != 0) {
48         s->k = -s->k;
49     }
50     if (((1 << n2) & n) != 0) {
51         s->k = -s->k;
52     }
53     n = (1 << n2) ^ ((1 << n1) ^ n);
54     s->base = n;
55 }
56
57 void spinz(ket *s, unsigned long n1, unsigned long n2) {
58     s->k = (s->k) / 4;
59     unsigned long n = s->base;
60     if (((1 << n1) & n) != 0) {
61         s->k = -s->k;
62     }
63     if (((1 << n2) & n) != 0) {

```

```
64     s->k = -s->k;
65 }
66 s->base = n;
67 }
68
69 double innerProduct(ket *s1, ket *s2) {
70     if (s1->base != s2->base) {
71         return 0;
72     }
73     return s1->k * s2->k;
74 }
75
76 double hamiltonian(ket *s1, ket *s2) {
77     unsigned long n = s1->dimension;
78     double result = 0;
79     for (int i = 0; i < n - 1; i++) {
80         int j = i + 1;
81         ket x;
82         ket y;
83         ket z;
84         copy(s2, &x);
85         copy(s2, &y);
86         copy(s2, &z);
87         spinx(&x, i, j);
88         spiny(&y, i, j);
89         spinz(&z, i, j);
90         result = result + innerProduct(s1, &x);
91         result = result + innerProduct(s1, &y);
92         result = result + innerProduct(s1, &z);
93     }
94     return result;
95 }
96
97 size_t getCombination(size_t n, size_t k){
98     if(k > n|k < 0)
99         return 0;
```

```

100     size_t l = k<(n-k)?k:(n-k);
101     size_t up = 1, lo = 1;
102     for (int i = 0 ; i < l ; i++){
103         up = up * (n-i);
104         lo = lo * (1+i);
105     }
106     return up/lo;
107 }
108
109 size_t getLocation(size_t n, size_t k){
110     size_t c = 0;
111     for (unsigned long i = 0; i < n; i++){
112         unsigned long r = 1 << i;
113         if ((k&r)>0)
114             c++;
115     }
116     return c;
117 }
118 /*
119 *Construct the M0 matrix.
120 */
121 gsl_matrix* getSimplifiedHamiltonian(long n){
122     gsl_vector_ulong *v = gsl_vector_ulong_alloc(1 << (n-1));
123     unsigned long c = 0;
124     unsigned long i = (1 << ((n+1)/2))-1;
125
126     while(c < getCombination(n,(n+1)/2)){
127         if (getLocation(n, i)==(n+1)/2){
128             gsl_vector_ulong_set(v,c,i);
129             c++;
130         }
131         i++;
132     }
133     static gsl_matrix *r = gsl_matrix_calloc(c,c);
134     for (i = 0; i < c; i++) {
135         for (unsigned long j = 0; j < c; j++) {

```

```
136     ket s1;
137     ket s2;
138     getKet(1, gsl_vector_ulong_get(v,i), n, &s1);
139     getKet(1, gsl_vector_ulong_get(v,j), n, &s2);
140     double a = hamiltonian(&s1, &s2);
141     gsl_matrix_set(r,i,j,a);
142 }
143 }
144 gsl_vector_ulong_free(v);
145 return r;
146 }
```

B

参考文献

- [1] SAKURAI J J, COMMINS E D. *Modern quantum mechanics, revised edition*. 1995.
- [2] 蔡旭锋. 玻色性杂质模型的数值研究[D]. 华中科技大学, 2021.