

第一部分

小时物理笔记

第一章

小时物理笔记

电磁场角动量分解

电磁场的动量为

$$\mathbf{p} = \epsilon_0 \int dV \mathbf{E} \times \mathbf{B} \quad (1)$$

角动量为

$$\mathbf{J} = \mathbf{r} \times \mathbf{p} = \epsilon_0 \int dV \mathbf{r} \times \mathbf{E} \times (\nabla \times \mathbf{A}) \quad (2)$$

现在假设电磁场只在一定范围内不为零，且体积分的边界处场强为零。假设该范围内没有净电荷与电流，则

$$\begin{aligned} \mathbf{r} \times \mathbf{E} \times (\nabla \times \mathbf{A}) &= \mathbf{r} \times [\nabla(\mathbf{E} \cdot \mathbf{A}) - \mathbf{A}(\mathbf{E} \cdot \nabla)]_{\partial A} \\ &= [(\mathbf{r} \times \nabla)(\mathbf{E} \cdot \mathbf{A}) - (\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial A} \end{aligned} \quad (3)$$

其中转微分算符 $[\]_{\partial A}$ 的作用是把方括号内的 ∇ 作为普通矢量进行计算，再把展开结果中每一项的偏微分作用在 A 的分量上。上式第一项为 $\sum_i E_i (\mathbf{r} \times \nabla) A_i$ ，第二项为

$$- [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial A} = -(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A}) + [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial r} \quad (4)$$

其中第二项为 $[(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial r} = [(\mathbf{E} \cdot \nabla)\mathbf{r}] \times \mathbf{A} = \mathbf{E} \times \mathbf{A}$ ，第一项中

$$\begin{aligned} (\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A}) &= [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E r A} - [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E} \\ &= [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E r A} \end{aligned} \quad (5)$$

这是因为 $[(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E} = (\nabla \cdot \mathbf{E})(\mathbf{r} \times \mathbf{A}) = 0$ 。综上，

$$\mathbf{J} = \epsilon_0 \int dV \sum_i E_i (\mathbf{r} \times \nabla) A_i + \epsilon_0 \int dV \mathbf{E} \times \mathbf{A} + \epsilon_0 \int dV [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E r A} \quad (6)$$

现在证明最后一项为 0。以 x 分量为例，

$$\begin{aligned} \hat{x} \int dV [(\mathbf{E} \cdot \nabla)(\mathbf{r} \times \mathbf{A})]_{\partial E r A} &= \int dV \nabla \cdot [\mathbf{E}(\hat{x} \cdot \mathbf{r} \times \mathbf{A})] \\ &= \oint d\mathbf{s} \mathbf{E}(\hat{x} \cdot \mathbf{r} \times \mathbf{A}) = 0 \end{aligned} \quad (7)$$

最后一步是因为边界处场强为零。现在我们可以看出角动量由两部分组成

$$\mathbf{J} = \mathbf{L} + \mathbf{S} \quad \mathbf{L} = \epsilon_0 \int dV \sum_i E_i (\mathbf{r} \times \nabla) A_i \quad \mathbf{S} = \epsilon_0 \int dV \mathbf{E} \times \mathbf{A} \quad (8)$$

其中 \mathbf{L} 是轨道角动量， \mathbf{S} 是自旋角动量。

晶体衍射

米勒指数

这里只讨论长方体晶格, 假设三条边分别为 a, b, c . 晶格面是通过许多格点的平面. 取一个格点为原点, 平面过三条坐标轴的截距分别定义为 $a/h, b/k, c/l$. 其中 h, k, l 必须是整数. 这样, 平面方程为

$$\frac{x}{a/h} + \frac{y}{b/k} + \frac{z}{c/l} = 1 \quad (1)$$

法向量为 $(h/a, k/b, l/c)/\sqrt{(h/a)^2 + (k/b)^2 + (l/c)^2}$. 现在来看相邻的两平面相距多少. 截点有仍然需要落在格点上, 所以只能是所有截距变为两倍. 两平面的距离为法向量点乘任何一个截距的增量矢量

$$d = \frac{1}{\sqrt{(h/a)^2 + (k/b)^2 + (l/c)^2}} \quad \frac{1}{d^2} = \frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2} \quad (2)$$

布拉格衍射公式

两个平面, 不管格点在上面如何分布, 若入射光和出射光和平面夹角都为 θ , 那么光程差为 $\delta = 2d \sin \theta$, 干涉条件为

$$2d \sin \theta = n\lambda \quad (3)$$

Crystalline Scattering Factor

晶格中各个原子的位置用 (x, y, z) 表示, 坐标为 (ax, by, cz) . 在进行布拉格衍射时, 同一个 cell 里面的不同格点会产生不同平面组, 即不同相位. 例如两点 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) 所在的两个平面

$$\begin{aligned} d_{12} &= \frac{(h/a, k/b, l/c)}{\sqrt{(h/a)^2 + (k/b)^2 + (l/c)^2}} \cdot [a(x_2 - x_1), b(y_2 - y_1), c(z_2 - z_1)] \\ &= \frac{(h\Delta x + k\Delta y + l\Delta z)}{\sqrt{(h/a)^2 + (k/b)^2 + (l/c)^2}} = d(h\Delta x + k\Delta y + l\Delta z) \end{aligned} \quad (4)$$

然而发生衍射时， d 对应的相位差为 2π ，所以 d_{12} 对应的相位差为

$$\delta = 2\pi(h\Delta x + k\Delta y + l\Delta z) \quad (5)$$

如果一个晶格有多个原子，每个原子的散射振幅为 f_i ，那么总振幅为

$$F = \sum_i f_i e^{2\pi i(hx_i + ky_i + lz_i)} \quad (6)$$

Hartree-Fork 方法

Hartree 方法的精髓是假设多粒子波函数 $\Psi(\mathbf{r}_1, \mathbf{r}_2 \dots \mathbf{r}_N)$ 等于每个单粒子波函数（包括自旋）的乘积 $u_1(q_1)u_2(q_2) \dots u_N(q_N)$ （Hartree 函数），其中不同的单粒子波函数要求正交归一。然后找到最优的单粒子波函数 $u_1, u_2, \dots u_N$ 使总哈密顿的平均值最小。所以该方法属于变分法。得到的能级大于精确能级。

注意 Hartree 函数并不满足全同粒子的对易关系，既不是对称也不是反对称。对于全同费米子，方法是令多粒子函数为单粒子函数（包括自旋）的 Slater 行列式。

$$\Psi(q_1, q_2 \dots) = \frac{1}{\sqrt{N!}} \begin{vmatrix} u_1(q_1) & u_2(q_1) & \dots \\ u_1(q_2) & u_2(q_2) & \dots \\ \vdots & \vdots & \ddots \end{vmatrix} \quad (1)$$

变分法的拉格朗日乘数函数为

$$L = \langle \Psi | \hat{H} | \Psi \rangle - \sum_{ij} \varepsilon_{ij} \langle u_i | u_j \rangle \quad (2)$$

为了化简第二项，令原基底为另一组基底的么正变换

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \end{pmatrix} = \hat{U} \begin{pmatrix} u'_1 \\ u'_2 \\ \vdots \end{pmatrix} \quad (3)$$

可以证明第一项不变. 因为

$$\begin{aligned}\Psi(q_1, q_2 \dots) &= \frac{1}{\sqrt{N!}} \begin{vmatrix} u_1(q_1) & u_2(q_1) & \cdots \\ u_1(q_2) & u_2(q_2) & \cdots \\ \vdots & \vdots & \ddots \end{vmatrix} = \frac{1}{\sqrt{N!}} \left| \mathbf{U} \begin{pmatrix} u'_1(q_1) & u'_1(q_2) & \cdots \\ u'_2(q_1) & u'_2(q_2) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \right| \\ &= \frac{|\mathbf{U}|}{\sqrt{N!}} \begin{vmatrix} u'_1(q_1) & u'_2(q_1) & \cdots \\ u'_1(q_2) & u'_2(q_2) & \cdots \\ \vdots & \vdots & \ddots \end{vmatrix} = \frac{e^{i\theta}}{\sqrt{N!}} \begin{vmatrix} u'_1(q_1) & u'_2(q_1) & \cdots \\ u'_1(q_2) & u'_2(q_2) & \cdots \\ \vdots & \vdots & \ddots \end{vmatrix}\end{aligned}\quad (4)$$

注意这里用到了 $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$ 和 $|\mathbf{U}| = e^{i\theta}$. 这是说, 平均能量关于 u 的公式不受么正变换的影响. 对第二项,

$$\sum_{ij} \varepsilon_{ij} \langle u_i | u_j \rangle = \int (u'_1, u'_2 \dots) \mathbf{U}^\dagger \varepsilon \mathbf{U} \begin{pmatrix} u'_1 \\ u'_2 \\ \vdots \end{pmatrix} \mathbf{d}q \quad (5)$$

若我们选择么正变换, 使得 $\mathbf{U}^\dagger \varepsilon \mathbf{U} = E_i \delta_{ij}$, 即把矩阵 ε 对角化 (下文可知 ε 为对称矩阵), 得

$$\sum_{ij} \varepsilon_{ij} \langle u_i | u_j \rangle = \sum_i E_i \langle u_i | u_i \rangle \quad (6)$$

现在把所有的撇号省略, 拉格朗日函数为

$$L = \langle \Psi | \hat{H} | \Psi \rangle - \sum_i E_i \langle u_i | u_i \rangle \quad (7)$$

即约束条件只需要归一化, 正交会自动完成. 现在来化简第一项. 首先把总波函数中的行列式记为求和的形式

$$\Psi = \frac{1}{\sqrt{N!}} \sum_{N!} (-1)^p \hat{P} \Psi_H \equiv \sqrt{N!} \hat{A} \Psi_H \quad (8)$$

其中 $\Psi_H = u_1(q_1)u_2(q_2)\dots$ 是 Hartree 函数, \hat{P} 是置换算符, 相当于做 p 次双粒子置换 (p 是逆序数), 行列式展开后共有 $N!$ 项. \hat{A} 为反对称化算符, 由于 \hat{H} 和 \hat{A} 存在一组共同本征矢, $[\hat{H}, \hat{A}] = 0$, 另外可以证明, $\hat{A}^2 = \hat{A}$ (意义是反对称化只需要一次) (可先证明 $N = 2, 3$, 高阶行列式同理).

$$\begin{aligned}\langle \Psi | \hat{H} | \Psi \rangle &= N! \langle \Psi_H | \hat{A} \hat{H} \hat{A} | \Psi_H \rangle = N! \langle \Psi_H | \hat{H} \hat{A}^2 | \Psi_H \rangle \\ &= N! \langle \Psi_H | \hat{H} \hat{A} | \Psi_H \rangle = \sum_{N!} (-1)^p \langle \Psi_H | \hat{H} \hat{P} | \Psi_H \rangle\end{aligned}\quad (9)$$

我们现在考虑多电子原子（离子）问题

$$\hat{H} = \sum_i \hat{h}_i + \sum_{i<j} \frac{1}{r_{ij}} \quad \hat{h}_i = -\frac{1}{2}\nabla^2 + \frac{Z}{r_i} \quad (10)$$

$$\begin{aligned} \langle \Psi | \sum_i \hat{h}_i | \Psi \rangle &= \sum_i \sum_{N!} (-1)^p \langle \Psi_H | \hat{h}_i \hat{P} | \Psi_H \rangle = \sum_i \langle \Psi_H | \hat{h}_i | \Psi_H \rangle \\ &= \sum_i \langle u_i | \hat{h}_i | u_i \rangle \end{aligned} \quad (11)$$

这是因为只有当 \hat{P} 为 1 时（行列式的对角项，逆序数 $p=0$ ）积分才不为零。同理， \hat{H} 剩下的部分为

$$\langle \Psi | \sum_{i<j} \frac{1}{r_{ij}} | \Psi \rangle = \sum_{i<j} \sum_{N!} (-1)^p \langle \Psi_H | \frac{1}{r_{ij}} \hat{P} | \Psi_H \rangle \quad (12)$$

现在 $P=1$ 或者 $P=P_{ij}$ ($p=1$) 时积分都可能不为零。所以

$$\begin{aligned} \langle \Psi | \sum_{i<j} \frac{1}{r_{ij}} | \Psi \rangle &= \sum_{i<j} \langle \Psi_H | \frac{1}{r_{ij}} (1 - P_{ij}) | \Psi_H \rangle \\ &= \sum_{i<j} \langle u_i u_j | \frac{1}{r_{ij}} | u_i u_j \rangle - \sum_{i<j} \langle u_i u_j | \frac{1}{r_{ij}} | u_j u_i \rangle \\ &\equiv \sum_{i<j} J_{ij} - \sum_{i<j} K_{ij} \end{aligned} \quad (13)$$

注意这里 $|u_i u_j\rangle^\dagger$ 记为 $\langle u_i u_j|$ 而不是 $\langle u_j u_i|$ 。另外易证 $J_{ij} = J_{ji}$, $K_{ij} = K_{ji}$ （交换积分变量即可）所以

$$L = \sum_i \langle u_i | \hat{h}_i | u_i \rangle + \sum_{i<j} \langle u_i u_j | \frac{1}{r_{ij}} \langle u_i u_j | - \sum_{i<j} \langle u_i u_j | \frac{1}{r_{ij}} | u_j u_i \rangle - \sum_i E_i \langle u_i | u_j \rangle \quad (14)$$

现在，类似于变分法中的过程，把任意一个 $\langle u_k |$ 变为 $\langle u_k + \delta u_k |$ ，减去上式，令为 0，得

$$\langle \delta u_k | h_k | u_k \rangle + \sum_j^{(j \neq k)} \langle \delta u_k u_j | \frac{1}{r_{jk}} | u_k u_j \rangle - \sum_j^{(j \neq k)} \langle \delta u_k u_j | \frac{1}{r_{ij}} | u_j u_k \rangle - E_k \langle \delta u_k | u_k \rangle = 0 \quad (15)$$

即

$$\langle \delta u_k | \left[h_k | u_k \rangle + \sum_j^{(j \neq k)} \langle u_j | \frac{1}{r_{jk}} | u_j \rangle | u_k \rangle - \sum_j^{(j \neq k)} \langle u_j | \frac{1}{r_{ij}} | u_k \rangle | u_j \rangle - E_k | u_k \rangle \right] = 0 \quad (16)$$

最后, 由于 $\langle \delta u_k |$ 可以取任意微小函数, 与之相乘的 ket 必须为零

$$h_k |u_k\rangle + \left[\sum_j^{(j \neq k)} \langle u_j | \frac{1}{r_{jk}} |u_j\rangle \right] |u_k\rangle - \sum_j^{(j \neq k)} \left[\langle u_j | \frac{1}{r_{ij}} |u_k\rangle \right] |u_j\rangle = E_k |u_k\rangle \quad (17)$$

这是所谓的, 非线性耦合微分积分本征方程组.

注意虽然现在 trial 波函数满足全同费米子的反对称, 一般却不是总自旋角动量的本征函数 (其实也不是总轨道角动量的本征函数, 除非把无穷个不同的 Ψ_H 求和). 为了实现这点, 可以先指定总自旋角动量 $|S, M\rangle$, 然后通过行列式线性组合构建自旋部分为 $|S, M\rangle$ 的 trial 波函数.

例 1 氦原子

对于 He 原子的 singlet 自旋态 $(\uparrow\downarrow - \downarrow\uparrow)/\sqrt{2}$, 可以用一个行列式构建 trial 波函数. 由于自旋为反对称, 轨道波函数必须为对称, 对于基态, 这意味着两个轨道波函数相同

$$\Psi = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi(\mathbf{r}_1) \uparrow & \phi(\mathbf{r}_1) \downarrow \\ \phi(\mathbf{r}_2) \uparrow & \phi(\mathbf{r}_2) \downarrow \end{vmatrix} = \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)(\uparrow\downarrow - \downarrow\uparrow)/\sqrt{2} \quad (18)$$

把 $u_1 = \phi \uparrow$, $u_2 = \phi \downarrow$ 代入本征方程组得单个轨道波函数的本征方程

$$\hat{h} |\phi\rangle + \langle \phi | \frac{1}{r_{12}} | \phi \rangle | \phi \rangle = E |\phi\rangle \quad (19)$$

当然也可以直接把 $\Psi = \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)(\uparrow\downarrow - \downarrow\uparrow)/\sqrt{2}$ 代入能量平均值公式, 用分母 $\langle \phi | \phi \rangle$ 归一化或者用拉格朗日乘数法做, 得到的方程与上式一样

$$\begin{aligned} L &= 2 \langle \phi \phi | \hat{h} | \phi \phi \rangle + \langle \phi \phi | \frac{1}{r_{12}} | \phi \phi \rangle - \lambda [\langle \phi | \phi \rangle - 1] \\ &= 2 \langle \phi | \hat{h} | \phi \rangle + \langle \phi \phi | \frac{1}{r_{12}} | \phi \phi \rangle - \lambda [\langle \phi | \phi \rangle - 1] \end{aligned} \quad (20)$$

注意拉格朗日乘数法中的优化函数可以使用限制条件化简. 令增量为 0

$$2 \langle \delta \phi | h | \phi \rangle + 2 \langle \delta \phi \phi | \frac{1}{r_{12}} | \phi \phi \rangle - \lambda \langle \delta \phi | \phi \rangle = 0 \quad (21)$$

其中使用了 $\delta \langle \phi \phi | = \langle \delta \phi \phi | + \langle \phi \delta \phi |$.

$$\hat{h} |\phi\rangle + \langle \phi | \frac{1}{r_{12}} | \phi \rangle | \phi \rangle = \frac{\lambda}{2} |\phi\rangle \quad (22)$$

对于其他的自旋态，往往需要行列式的线性组合才能构造总自旋本征态，这个比较复杂，先来看另一种方法。我们可以直接指定总自旋态，如果要求轨道波函数反对称，可把 $u_i(q_i) = \phi_i(\mathbf{r}_i)$ 直接代入本征方程组即可，而当要求轨道波函数对称时，可以把以上的反对称化算符 \hat{A} 中的 $(-1)^p$ 去掉改成对称化算符 \hat{B}

$$\hat{B} |\Psi_H\rangle = \frac{1}{N!} \sum_{N!} \hat{P} |\Psi_H\rangle \quad (23)$$

\hat{B} 同样满足 $[\hat{H}, \hat{B}] = 0$, $\hat{B}^2 = \hat{B}$. 以上推导全部有效（所有负号改成正号即可），新的本征方程组变为

$$h_k |u_k\rangle + \left[\sum_j^{(j \neq k)} \langle u_j | \frac{1}{r_{jk}} |u_j\rangle \right] |u_k\rangle + \sum_j^{(j \neq k)} \left[\langle u_j | \frac{1}{r_{ij}} |u_k\rangle \right] |u_j\rangle = E_k |u_k\rangle \quad (24)$$

例 2 氦原子

考虑 He 的 $1s2s$, 1S 态，即自旋为 singlet $|0,0\rangle = (\uparrow\downarrow - \downarrow\uparrow)/\sqrt{2}$. 两个空间波函数不相同。直接把 $u_1 = \phi_{1s}$, $u_2 = \phi_{2s}$ 代入上面的对称本征方程组，得

$$\hat{h}_1 |\phi_{1s}\rangle + \langle \phi_{2s} | \frac{1}{r_{12}} | \phi_{2s} \rangle |\phi_{1s}\rangle + \langle \phi_{2s} | \frac{1}{r_{12}} | \phi_{1s} \rangle |\phi_{2s}\rangle = E_1 |\phi_{1s}\rangle \quad (25)$$

$$\hat{h}_2 |\phi_{2s}\rangle + \langle \phi_{1s} | \frac{1}{r_{12}} | \phi_{1s} \rangle |\phi_{2s}\rangle + \langle \phi_{1s} | \frac{1}{r_{12}} | \phi_{2s} \rangle |\phi_{1s}\rangle = E_2 |\phi_{2s}\rangle \quad (26)$$

如果直接用 $|\Psi\rangle = (\phi_1\phi_2 + \phi_2\phi_1)/\sqrt{2}$, 结果相同。

FEDVR 算法

预备知识 高斯积分 (Gauss-Lobatto)

FEDVR 基底

现在以一维 FEDVR 为例，把整个区间划分成 N_e 个有限元，第 j 个有限元的区间为 $[x_j, x_{j+1}]$. 每个有限元内进一步加入格点，令 x_n^j 为第 j 区间的 N 阶 Gauss-Lobatto 数值积分的 N 个采样点。这样， $x_1^j = x_j$, $x_N^j = x_{j+1}$. 把每个

有限元的区间作为定义域建立 N 个多项式作为函数基底，第 j 个有限元内的第 n 个基底记为 u_n^j ，要求满足

$$u_n^j(x_{n'}^j) = \delta_{n,n'} \quad (1)$$

这样，根据 N 阶 Gauss-Lobatto 数值积分公式

$$\langle u_n^j | u_{n'}^j \rangle = \int_{x_j}^{x_{j+1}} u_n^j(x) u_{n'}^j(x) dx = \frac{x_{j+1} - x_j}{2} \sum_{i=1}^N w_i u_n^j(x_i) u_{n'}^j(x_i) \quad (2)$$

如果 u_n^j 的阶数都小于等于 $N - 1$ ，则上式精确成立。把式 1 代入，可得正交关系

$$\langle u_n^j | u_{n'}^j \rangle = \delta_{n,n'} \frac{x_{j+1} - x_j}{2} w_n \quad (3)$$

在边界 x_j 处，我们要求函数连续但不要求导数连续，于是可以把定义域为 $[x_j, x_{j+1}]$ 的 u_N^j 与定义域为 $[x_{j+1}, x_{j+2}]$ 的 u_1^{j+1} 合并为一个函数，重新记为 u_N^j ，放弃符号 u_1^j 。另外，对于两个端点 x_1 和 x_{N_e+1} ，令边界条件为函数值等于 0，所以我们放弃 $u_1^1(x)$ 和 $u_N^{N_e}(x)$ 两个基底。这样，我们最终共有 $N_e N - (N_e - 1) - 2$ 个基底。最广义的正交关系为

$$\langle u_n^j | u_{n'}^{j'} \rangle = \delta_{j,j'} \delta_{n,n'} \frac{x_{j+1} - x_j}{2} w_n \quad (4)$$

这个积分范围可以是 $[x_1, x_{N_e+1}]$ 。以下为了方便描述，把每个有限元 j 的区间 $x \in [x_j, x_{j+1}]$ 均匀映射到 $y \in [-1, 1]$ 上

$$y = \frac{x_{j+1} - x_j}{2} x + \frac{x_j + x_{j+1}}{2} \quad (5)$$

且定义 y_i 由 $x = x_i^j$ 代入该式得到。我们在所有的有限元中都使用同样的基底，换言之，对于每个 j ，每个 $n = 2, 3 \dots N - 1$ ，在 $x \in [x_j, x_{j+1}]$ ，令

$$u_n^j(x) = u_n(y) \quad y \in [-1, 1] \quad (6)$$

把剩下的基底 $u_N^j(x)$ 定义为

$$u_N^j(x) = \begin{cases} u_N(y) & x \in [x_j, x_{j+1}] \\ u_1(y) & x \in [x_{j+1}, x_{j+2}] \end{cases} \quad (7)$$

现在只要找到 N 个多项式 $u_n(y)$ ，就可以找到所有 $u_n^j(x)$ 基底。给 $u_n(y)$ 加上式 1 的条件得

$$u_n(y_i) = \delta_{n,i} \quad (8)$$

满足该式的 $N-1$ 阶多项式可以选用拉格朗日插值公式，即

$$\begin{aligned} u_n(y) &= \prod_{i=1}^{n-1} \frac{y - y_i}{y_n - y_i} \prod_{i=n+1}^N \frac{y - y_i}{y_n - y_i} \\ &= \frac{y - y_1}{y_n - y_1} \times \dots \times \frac{y - y_{n-1}}{y_n - y_{n-1}} \frac{y - y_{n+1}}{y_n - y_{n+1}} \dots \frac{y - y_N}{y_n - y_N} \end{aligned} \quad (9)$$

另一种等效的表示方法是利用 N 阶 Gauss-Lobatto 数值积分对应的多项式， $N-1$ 阶勒让德多项式的导数， $P'_{N-1}(y)$ ，来构建满足条件的多项式。根据定义，其 $N-2$ 个零点分别为 $y_2, y_3 \dots y_{N-1}$ ，为了加入 $y_1 = -1$ 与 $y_n = 1$ 这两个零点，将其变为 N 阶多项式

$$(1 - y^2)P'_{N-1}(y) \quad (10)$$

然而，式 8 要求 $u_n(y_n) = 1$ ，所以我们将式 10 除以它自己在 y_n 处的切线，在 $y = y_n$ 处形成极限类型 $0/0 = 1$ 即可得到阶多项式 $u_n(y)$ 。

$$u_n(y) = \frac{(1 - y^2)P'_{N-1}(y)}{[(1 - y^2)P'_{N-1}(y)]'_y} = y_n(y - y_n) \quad (11)$$

该式与式 9 事实上是完全相同的多项式，因为所有具有 $N-1$ 个零点的 $N-1$ 阶多项式都可以因式分解成式 9 的形式乘以一个待定常数。用式 11 便于快速地展开多项式（因为勒让德多项式的系数可以直接通过公式计算）。

现在知道了所有基底，任意时刻的波函数便可近似展开为

$$\Psi(x, t) = \sum_{j=1}^{N_e-1} \sum_{n=2}^N c_n^j(t) u_n^j(x) + \sum_{n=2}^{N-1} c_n^{N_e}(t) u_n^{N_e}(x) \quad (12)$$

可以用一个 $N_e N - N_e = 1$ 维的列矢量描述波函数

$$\psi = (c_2^1, \dots, c_N^1, \dots, c_2^2, \dots, c_N^2, \dots, c_{N-1}^{N_e})^\dagger \quad (13)$$

建立全局下标 α, β ，例如 $\psi_\alpha = c_n^j$ 中 $\alpha = (j-1)(N-1) + (n-1)$ 。

哈密顿算符

现在讨论在该基底中如何把哈密顿算符转换成矩阵.

$$\begin{aligned} H_{\alpha\beta} &= K_{\alpha\beta} + V_{\alpha\beta} \\ &= -\frac{\hbar^2}{2m} \int_{x_1}^{x_{Ne+1}} u_n^j(x) \cdot u_{n'}^{j'}(x)'' \, dx + \int_{x_1}^{x_{Ne+1}} u_n^j(x) u_{n'}^{j'}(x) V(x) \, dx \end{aligned} \quad (14)$$

其中

$$V_{\alpha\beta} = \int_{x_1}^{x_{Ne+1}} u_n^j(x) u_{n'}^{j'}(x) V(x) \, dx = \quad (15)$$

Lanczos 算法

背景介绍

在计算含时薛定谔方程时, 若已知某时刻 t 的波函数 $\Psi(\mathbf{r}, t)$, 要求 $\Psi(\mathbf{r}, t + \Delta t)$, 通常使用传播算符

$$U(\Delta t) = \exp\left(-\frac{i}{\hbar} \int_t^{t+\Delta t} dt \cdot H(t)\right) \approx \exp(-iH(t + \Delta t/2)\Delta t) \quad (1)$$

其中 H 为哈密顿算符. 若 H 不含时, 约等号变为等号

$$\Psi(\mathbf{r}, t + \Delta t) = \exp(-iH\Delta t/\hbar)\Psi(\mathbf{r}, t) \quad (2)$$

通常情况下, 用有限个正交归一基底 $\Psi_0, \dots, \Psi_{K-1}$ 近似展开 $\Psi(\mathbf{r}, t + \Delta t)$, 这时, H 可以表示成矩阵的形式.

$$H_{ij} = \langle \Psi_i | H | \Psi_j \rangle \quad (3)$$

这样, 么正算符 $\exp(-iH\Delta t/\hbar)$ 也可以表示成矩阵. 根据定义

$$\exp(-iH\Delta t/\hbar) = 1 + (-iH\Delta t/\hbar) + \frac{1}{2!}(-iH\Delta t/\hbar)^2 \dots \quad (4)$$

此时若把 H 对角化 (求解本征方程), 得到本征矢列矩阵 \mathbf{P} (单位正交阵) 以及本征值矩阵 $\mathbf{\Lambda}$ (对角矩阵), 则可进行基底变换变到 H 矩阵的本征空间求

解上式

$$\begin{aligned}\exp(-iH\Delta t/\hbar) &= \mathbf{P} \times \text{diag}(\mathbf{e}^{-iE_1 t/\hbar}, \mathbf{e}^{-iE_2 t/\hbar} \dots \mathbf{e}^{-iE_N t/\hbar}) \times \mathbf{P}^{-1} \\ &= \sum_{j=0}^{K-1} P_{ij} \mathbf{e}^{-iE_j t/\hbar} P_{jk}^{-1} = \sum_{j=0}^{K-1} P_{ij} P_{kj} \mathbf{e}^{-iE_j t/\hbar}\end{aligned}\quad (5)$$

要说明的是，这种算法的误差（除数值计算误差）来源于两个方面，第一是用 $\exp(iH\Delta t/\hbar)$ 代替 $\exp\left(-\frac{i}{\hbar} \int_t^{t+\Delta t} dt \cdot H(t)\right)$ ，但如果不含时就没有该误差，第二是有限数量的基底不具有完备性，这个误差可以随着基底数量增加而减小。

Lanczos 算法

在以上的算法中，选取施密特正交归一化的 Krilov 基底作为基底。令 $\Psi_0 \equiv \Psi(\mathbf{r}, t)$ ， K 阶的 Krilov 基底为

$$\{|\Psi_0\rangle, H|\Psi_0\rangle, H^2|\Psi_0\rangle \dots H^{K-1}|\Psi_0\rangle\} \quad (6)$$

我们把正交归一化后的基底记为

$$\{|\Psi_0\rangle, |\Psi_1\rangle, |\Psi_2\rangle \dots |\Psi_{K-1}\rangle\} \quad (7)$$

这样，用其展开哈密顿算符

$$H_{ij} = \langle \Psi_i | H | \Psi_j \rangle \quad (8)$$

在算法上，本来我们可以按部就班地按照以上步骤做，然而由于 Krilov 基底的性质，可以通过一些定理（见词条最后）大大减少计算量。正交归一化步骤如下

1. 如果 $\Psi(\mathbf{r}, t)$ 没有归一化，将其进行归一化 $|\Psi_0\rangle = \Psi(\mathbf{r}, t) / \langle \Psi(\mathbf{r}, t) | \Psi(\mathbf{r}, t) \rangle$.
2. 把 Ψ_1 进行施密特正交化，令

$$|\tilde{\Psi}_1\rangle = (1 - |\Psi_0\rangle \langle \Psi_0|) |\Psi_1\rangle \quad |\Psi_1\rangle = |\tilde{\Psi}_1\rangle / \langle \tilde{\Psi}_1 | \tilde{\Psi}_1 \rangle \quad (9)$$

3. 现在起我们令 $\beta_j = \langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle$.

定理 1

对以上定义的基底

$$(1 - |\Psi_j\rangle\langle\Psi_j| - |\Psi_{j-1}\rangle\langle\Psi_{j-1}| - \dots - |\Psi_0\rangle\langle\Psi_0|)H^{j+1}|\Psi_0\rangle \quad (10)$$

与

$$(1 - |\Psi_j\rangle\langle\Psi_j| - |\Psi_{j-1}\rangle\langle\Psi_{j-1}| - \dots - |\Psi_0\rangle\langle\Psi_0|)H|\Psi_j\rangle \quad (11)$$

共线。故现在起采用后者进行正交化。

定理 2

$$\langle\Psi_{j-1}|H|\Psi_j\rangle = \sqrt{\langle\tilde{\Psi}_j|\tilde{\Psi}_j\rangle} = \beta_j \quad (12)$$

4. 把 $H^2|\Psi_0\rangle$ 进行施密特正交化，根据两个定理，正交化结果也可以写成

$$\begin{aligned} |\tilde{\Psi}_2\rangle &= (1 - |\Psi_1\rangle\langle\Psi_1| - |\Psi_0\rangle\langle\Psi_0|)H|\Psi_1\rangle \\ &= H|\Psi_1\rangle - \alpha_1|\Psi_1\rangle - \beta_1|\Psi_0\rangle \end{aligned} \quad (13)$$

5. 再次归一化 $|\Psi_2\rangle = |\tilde{\Psi}_2\rangle / \langle\tilde{\Psi}_2|\tilde{\Psi}_2\rangle = |\tilde{\Psi}_2\rangle / \beta_2$

定理 3

$$\begin{aligned} |\tilde{\Psi}_{j+1}\rangle &= (1 - |\Psi_j\rangle\langle\Psi_j| - \dots - |\Psi_0\rangle\langle\Psi_0|)H|\Psi_j\rangle \\ &= (1 - |\Psi_j\rangle\langle\Psi_j| - |\Psi_{j-1}\rangle\langle\Psi_{j-1}|)H|\Psi_j\rangle \\ &= H|\Psi_j\rangle - \alpha_j|\Psi_j\rangle - \beta_j|\Psi_{j-1}\rangle \end{aligned} \quad (14)$$

这就是最简洁的正交化公式。比起最原始的正交化

$$|\tilde{\Psi}_{j+1}\rangle = (1 - |\Psi_j\rangle\langle\Psi_j| - \dots - |\Psi_0\rangle\langle\Psi_0|)H^{j+1}|\Psi_0\rangle \quad (15)$$

定理 3 只用到了一个矩阵-矢量乘法，两个矢量数乘和两个矢量减法。

接下来只要对 $j = 2, 4, 5 \dots K-1$ 不断重复定理 3 中的正交化和归一化，就可以得到所有正交归一基底 $|\Psi_j\rangle$ 。

现在我们来求该基底下的 \mathbf{H} 矩阵。根据 α_i 的定义以及定理 2 可以发现它们分别是矩阵的主对角元和副对角元。

定理 4

在正交归一基底 $|\Psi_j\rangle$ 中，矩阵 \mathbf{H} 是对称三对角矩阵。所以我们已经顺便求出了所有的矩阵元！

$$\mathbf{H} = \begin{pmatrix} \alpha_0 & \beta_1 & & & \\ \beta_1 & \alpha_1 & \beta_2 & & \\ & \beta_2 & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \beta_{K-1} \\ & & & \beta_{K-1} & \alpha_{K-1} \end{pmatrix} \quad (16)$$

该矩阵具有维数小，易求本征问题的优势。

定理 1 证明

根据施密特正交化的性质，Krilov 基底的前 j 项的与 $|\Psi_0\rangle \dots |\Psi_j\rangle$ 展开同一空间。所以 $H^j |\Psi_0\rangle = c_j |\Psi_j\rangle + \dots + c_0 |\Psi_0\rangle$ ，所以（以下所有系数只是用来表示线性组合，具体值不重要）

$$\begin{aligned} H^{j+1} |\Psi_0\rangle &= c_j H |\Psi_j\rangle + H(c_{j-1} |\Psi_{j-1}\rangle \dots + c_0 |\Psi_0\rangle) \\ &= c_j H |\Psi_j\rangle + H(c'_{j-1} H^{j-1} |\Psi_0\rangle + c'_{j-2} H^{j-2} |\Psi_0\rangle \dots + c'_0 |\Psi_0\rangle) \\ &= c_j H |\Psi_j\rangle + (c'_{j-1} H^j |\Psi_0\rangle + c'_{j-2} H^{j-1} |\Psi_0\rangle \dots + c'_0 |\Psi_0\rangle) \\ &= c_j H |\Psi_j\rangle + (c''_{j-1} |\Psi_j\rangle + c''_{j-2} |\Psi_{j-1}\rangle + \dots + c''_0 |\Psi_0\rangle) \end{aligned} \quad (17)$$

把 $H^{j+1} |\Psi_0\rangle$ 施密特正交归一化

$$\begin{aligned} &(1 - |\Psi_j\rangle \langle \Psi_j| - \dots - |\Psi_0\rangle \langle \Psi_0|) H^{j+1} |\Psi_0\rangle \\ &= c_j (1 - |\Psi_j\rangle \langle \Psi_j| - \dots - |\Psi_0\rangle \langle \Psi_0|) H |\Psi_j\rangle \\ &\quad + (1 - |\Psi_j\rangle \langle \Psi_j| - \dots - |\Psi_0\rangle \langle \Psi_0|) (c''_{j-1} |\Psi_j\rangle + c''_{j-2} |\Psi_{j-1}\rangle + \dots + c''_0 |\Psi_0\rangle) \\ &= c_j (1 - |\Psi_j\rangle \langle \Psi_j| - \dots - |\Psi_0\rangle \langle \Psi_0|) H |\Psi_j\rangle \end{aligned} \quad (18)$$

定理 2 证明

要证 $\langle \Psi_{j-1} | H | \Psi_j \rangle = \sqrt{\langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle}$ ，即证 $\langle \Psi_{j-1} | H | \tilde{\Psi}_j \rangle = \langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle$ ，即证 $\langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle = \langle \tilde{\Psi}_j | H | \Psi_{j-1} \rangle$ 而

$$\langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle = \langle \tilde{\Psi}_j | (1 - |\Psi_{j-1}\rangle \langle \Psi_{j-1}| - \dots - |\Psi_0\rangle \langle \Psi_0|) H | \Psi_{j-1} \rangle \quad (19)$$

由于上式中 $\langle \tilde{\Psi}_j | (|\Psi_{j-1}\rangle \langle \Psi_{j-1}| - \dots - |\Psi_0\rangle \langle \Psi_0|) = 0$

$$\langle \tilde{\Psi}_j | \tilde{\Psi}_j \rangle = \langle \tilde{\Psi}_j | H | \Psi_{j-1} \rangle \quad (20)$$

定理 3 证明

要证明定理 3，即证，对 $n \leq j-2$ ，有 $\langle \Psi_n | H | \Psi_j \rangle = 0$ ， H 为厄米算符时即证 $\langle \Psi_j | H | \Psi_n \rangle = 0$ ，即证对 $m \geq j+2$ ， $\langle \Psi_m | H | \Psi_j \rangle = 0$ 。在定理 1 的证明类似，我们知道

$$\begin{aligned} H | \Psi_j \rangle &= H (c_j H^j | \Psi_0 \rangle + \dots + c_0 | \Psi_0 \rangle) = c_j H^{j+1} | \Psi_0 \rangle + \dots + c_0 H | \Psi_0 \rangle \\ &= c'_j | \Psi_{j+1} \rangle + \dots + c'_0 | \Psi_0 \rangle \end{aligned} \quad (21)$$

所以对 $m \geq j+2$ 有

$$\langle \Psi_m | H | \Psi_j \rangle = \langle \Psi_m | (c'_j | \Psi_{j+1} \rangle + \dots + c'_0 | \Psi_0 \rangle) = 0 \quad (22)$$

定理 4 证明

首先证明 \mathbf{H} 是三对角矩阵。即证 $m \geq j+2$ 或 $m \leq j-2$ 时 $\langle \Psi_m | H | \Psi_j \rangle = 0$ 。 $m \geq j+2$ 的情况在证明 3 中已经证明，只需证明另一种情况。对厄米矩阵， $\langle H \Psi_m | \Psi_j \rangle = \langle \Psi_m | H | \Psi_j \rangle = 0$ ，取复共轭，即 $\langle \Psi_j | H | \Psi_m \rangle = 0$ 。可见左边的角标的确小于等于右边的角标减 2，证毕。

然后证明 \mathbf{H} 是实矩阵，考虑到厄米矩阵 $H_{mn} = H_{nm}^*$ 的性质，只需要证明 $H_{j,j+1}$ 是实数。对任意 $j = 2 \dots K-1$

$$\begin{aligned} H_{j,j+1} &= \langle \Psi_j | H | \Psi_{j+1} \rangle = \langle \Psi_j | H (H | \Psi_j \rangle - \alpha_j | \Psi_j \rangle - \beta_j | \Psi_{j-1} \rangle) \\ &= \langle \Psi_j | H^2 | \Psi_j \rangle - \alpha_j \langle \Psi_j | H | \Psi_j \rangle - \beta_j \langle \Psi_j | H | \Psi_{j-1} \rangle \end{aligned} \quad (23)$$

易证 $\langle \Psi_j | H^2 | \Psi_j \rangle$ ， α_j 及 β_j 是实数，只需要证明 $\langle \Psi_j | H | \Psi_{j-1} \rangle$ 是实数即可，即证 $\langle \Psi_{j-1} | H | \Psi_j \rangle$ 是实数，重复上述论证，即证 $\langle \Psi_{j-2} | H | \Psi_{j-1} \rangle$ 是实数...，即证 $\langle \Psi_1 | H | \Psi_0 \rangle$ 是实数，即证 $\langle \Psi_0 | H | \Psi_1 \rangle$ 是实数，即证 $\langle \Psi_0 | H | \tilde{\Psi}_1 \rangle$ 是实数。而

$$\langle \Psi_0 | H | \tilde{\Psi}_1 \rangle = \langle \Psi_0 | H (H | \Psi_0 \rangle - \alpha_0 | \Psi_0 \rangle) = \langle \Psi_0 | H^2 | \Psi_0 \rangle - \alpha_0 \langle \Psi_0 | H | \Psi_0 \rangle \quad (24)$$

显然是实数。以上的思路是数学归纳法的逆过程。

Crank-Nicolson TDSE 笔记

氢原子的数值误差

解三维氢原子的时候，为了防止势能为无穷，中心点要选在八个格点中间。

首先用的 $60 \times 60 \times 60$ a.u. 的范围， $401 \times 401 \times 401$ 的格点，0.032 a.u. 的时间步长。

TDSE 传播时，如果把 log 概率灰度图的范围取得很大 (10^{-25} 左右)，即使没有电场，也会出现一个方形的花纹很快地扩散出来，然后波函数还会向外温柔地扩散。

比较了用解析的基态波函数和 Imaginary Time 得到的基态波函数，虽然两者都有类似情况，但是后者的程度明显要轻。

爱华说 XUV 的强度并不太重要，频率越大，氢原子的截面就越小，所以反而需要更大的强度才能电离出同样的概率。但只要不超过 1×10^{16} W/cm² (0.5338 a.u.) 就好。

Virtual Detector 的两个问题

一个是如果电场还没有完全消失，那么电离出来的波函数仍然会随着电场振动，那么检测到的动量分布就不可靠了。

第二个是如果用 histogram 的办法来获得动量分布，那么有可能数据点不够，bin 太窄又波动大，bin 太宽又分辨率不够。如果像 Uwe 的论文那样通过 interpolate 的方法来获取动量分布，那么要求 Virtual Detector 检测到的“概率-时间”曲线和“动量-时间”曲线都非常平滑，用高斯波（无势能）测试的时候还足够平滑，一旦用氢原子电离的波函数就不行了。而且电离出的波函数检测到的动量会上下波动（有可能是因为电场），这样就要把“动量-时间”函数分成单调的分段，而且在导数为零的时候，还有可能会 blowup。

所有考虑了一下还是想做傅里叶变换。一是即使 Virtual Detector 做了种种修改后出结果了，也可以用 FFT 进行验证。二是如果 FFT 比较好，就直接用呗。FFT 的一个限制就是要等到波函数电离完成后，最远的波函数也不能超过 box 的边界，还有就是电离出的波函数最好远离原子核，少受库仑力的影响，这样

才能保证接下来的动量谱不随时间变化.

从高斯波包的实验来看, 即使在理想情况下 3D 的 local Virtual Detector 和 FFT 也是不同的, 因为高斯波包会随时间变大, 概率波也会更稀, 所以 Virtual Detector 检测到的动量分布并不是对称的, 而是前面胖后面瘦.

另外我觉得即使使用 FFT 也需要一圈 absorber 用于吸收开始是的方形花纹噪音.

Streaking

SFA 公式中当 XUV 为零时, IR 是不会对能量谱造成任何影响的. 那么 TDSE 中会吗? 理想的情况是, IR 不会产生任何电离, 只会使基态波函数发生微弱的形变. 一旦 XUV 电场结束后, IR 电场同样不会产生任何电离, 只会让已经电离出来的波函数加速和减速.

那接下来 IR 对电离出的波函数的影响应该完全模拟吗? 如果是的话, 那么将需要一个很大的 box, Ivanove paper 中的 TDSE 用了 5000 个径向格点 (不等间距), 2000 个原子单位的长度, 我的 GPU 代码是绝对实现不了这个的.

有一种想法是在电离波函数离开 box 之前将它们 FFT 分解成平面波, 然后在用 Volkov State 的解析办法分析接下来每个平面波在剩下的 IR 场中的变化情况, 这样或许是可行的.

Streaking

这里的目标是用 TDSE 模拟出氢原子的 Streaking 谱, XUV 和 IR 参数就用 Ivanov 的论文 (Attosecond recorder of ...) 中的.

XUV 是 250 as, 41 eV (动能 1.007 a.u.), 还有 varying ellipticity, 最大强度约 5×10^{-4} a.u. (不重要), IR 是 800 nm, 7.2 fs (297.5 a.u.), 强度 3.5×10^{12} W/cm² (0.01 a.u.).

用经典电子来估计 TDSE 所需 box 的大小, 速度为 $\sqrt{2}$ 左右, 若最大传播时间为 IR 长度的两倍, 可以走 841 a.u., 比我测试 XUV 用的 60 a.u. 盒子大了一个数量级. 再加上我的直角坐标网格不能改变长度, 结果就非常地惨烈了.

唯一有可能挽救我的程序的方法就是只模拟到 XUV 消失, 这时只要保证波函数还在盒内就可以做一个 FFT 然后用 Volkov 波函数来解析传播. 但是 XUV 消失的时候电离波包也就离原子核 40 a.u. 左右, 所以如果这时开始忽略

库仑力，那么动能谱将会有 2.5% 的误差，这应该是不能接受的。即使波函数继续传播到 100 a.u. 开外，动能谱也会有 1% 的误差。所以师兄还是老的辣，第一反应就是不行。

所以最后的解决方法只能是用参考文献 29 的超级复杂的球坐标 TDSE 代码了 (Fortran)。

SCID-TDSE

看来 TDSE Streaking 真的远远没有我想的那么简单。试了一下 State of Art 的 SCID-TDSE 代码，单核 CPU 就完爆我的 GPU 代码了！而且 Memory 使用更是少得不能再少。这个算法是在球坐标中把波函数在球谐函数展开，具体见论文就好，总之这个算法超级复杂。

好不容易用 Matlab 写好了处理运行结果的程序，用傅里叶变换分析最后的波函数谱。但出来以后动量谱并不是一个完美的波包。然而师兄一跑出来就是！师兄真的好猛啊，想起了当年的坦兄。

傅里叶变换的一个问题是，如果波函数不做任何处理，由于电离出的总概率很小，那么基态波函数的动量谱将会把电离出的波包的动量谱覆盖。我的第一反应是挖掉中间的波函数，而师兄的做法是挖掉 $l = 0$ 的分量，理由是根据微扰理论/选择定理，电离出来的 $l = 1$ 。但我试过以后发现如果电离波包和基态完全分离两种做法并无差别，但如果有重叠的，还是师兄的做法好。另外用我的参数 $2 \times 10^{14} \text{ W/cm}^2$ 的光强，比师兄的 1×10^{10} 到 1×10^{11} 强很多，我的电离总概率是 1/1000 左右，而师兄的是 7.6×10^{-7} ，差了三个数量级。结果他的动量谱是高斯的，而我的是一个不对称的包；他的电离波包是和基态完全分离的，我的不是。根据微扰理论，电离概率应该适合电场平方成正比的。

FROG

如果我们要测量线偏振的超快激光脉冲的波形（即电场关于时间的函数），我们不能直接用仪器测量，因为电子元件的时间分辨率远远不够。但我们可以使用 Frequency Resolved Optical Gating (FROG)。FROG 有许多不同的实现方法，这里只讨论比较常见的 SHG FROG (Second Harmonic Generation FROG) 和 PCGPA。

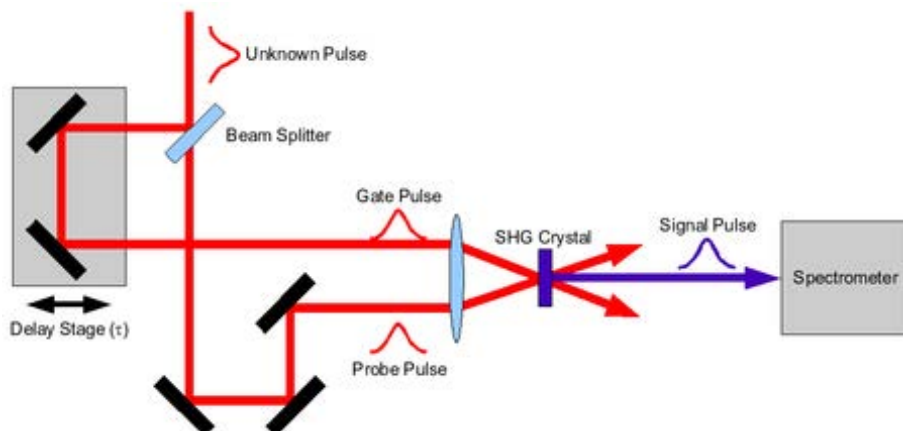


图 1: SHG FROG 的光路 (图片来自 Wikipedia)

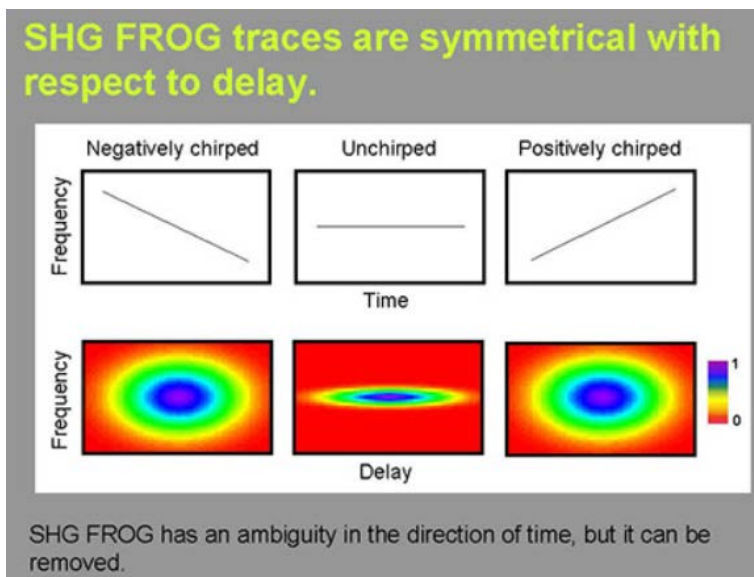


图 2: SHG FROG trace (图片来自网络)

Frog 的大概原理就是把被测量的脉冲 (prob) $f(t)$ 和另一个脉冲 (gate) $g(t)$ 叠加 (注意 $f(t)$ 和 $g(t)$ 都是实函数), 然后通过一个二阶非线性介质产生一个 $f(t)g(t)$ 脉冲并分离出来, 再测量光谱 (即傅里叶变换). 接下来我们可以控制两个脉冲的相对延迟 (time delay) τ , 就得到不同延迟下 $f(t-\tau)g(t)$ 的光谱, 也就是一个二维函数, 叫做 **FROG trace** (图 2). 很多情况下, gate 就是 prob 通过一个 beam splitter 分出来的, 实验光路如图 1 所示.

$$a(\omega, \tau) = \left| \int_{-\infty}^{\infty} f(t - \tau)g(t)e^{-i\omega t} dt \right|^2 \quad (1)$$

只要用特定的算法，就可以从式 1 中解出 $f(t)$ 和 $g(t)$ 。

PCGPA 算法

PCGPA (Principal Component General Projection Algorithm) 就是从式 1 中解出 $f(t)$ 和 $g(t)$ 的一种常见算法。在得到我们假设 $f(t)$ 和 $g(t)$ 都是等时间间隔的离散点 f_1, f_2, \dots, f_N 和 g_1, g_2, \dots, g_N ，可以看成两个列矢量。我们先来做两矢量的外积，即

$$\mathbf{f}\mathbf{g}^T = \begin{pmatrix} f_1g_1 & \dots & f_1g_N \\ \vdots & \ddots & \vdots \\ f_Ng_1 & \dots & f_Ng_N \end{pmatrix} \quad (2)$$

然后我们把第 i 行向左移动 $i - 1$ 个元（左边多出的矩阵元补到右边），得到

$$\begin{pmatrix} f_1g_1 & f_1g_2 & \dots & f_1g_N \\ f_2g_2 & f_2g_3 & \dots & f_2g_1 \\ \vdots & \vdots & \vdots & \vdots \\ f_Ng_N & f_Ng_1 & \dots & f_Ng_{N-1} \end{pmatrix} \quad (3)$$

现在可以发现每一列恰好是 $f(t - \tau)g(t)$ 的离散形式¹，第 i 列 $\tau = (i - 1)\Delta t$ 。然而当 $i > N/2$ 的时候，更自然的理解是 $\tau = (i - N)\Delta t$ 。例如与其认为第 N 列是 g 向上移动了 $N - 1$ 个元，倒不如认为是向下移动了 1 个元。根据这种思想，我们可以把矩阵左半和右半调换得到

$$\begin{pmatrix} f_1g_{N/2+1} & f_1g_{N/2+2} & \dots & f_1g_{N/2} \\ f_2g_{N/2+2} & f_2g_{N/2+3} & \dots & f_2g_{N/2+1} \\ \vdots & \vdots & \vdots & \vdots \\ f_Ng_{N/2} & f_Ng_{N/2+1} & \dots & f_Ng_{N/2-1} \end{pmatrix} \quad (4)$$

这样，从左到右的每列分别对应从 $\tau = -(N/2)\Delta t, \dots, (N/2 - 1)\Delta t$ 。

¹唯一的区别是从最上面移出的 g_i 跑到了最下面，而在实验中最下面应该由 0 来填补。但如果 \mathbf{f} 和 \mathbf{g} 矢量的首尾都有足够多的 0，这个问题就自动解决了。

现在我们对每一列做离散傅里叶变换（注意前后都要 `fftshift`），就得到了含相位的 **Frog trace** 矩阵，其中的每一列都是式 1 中的

$$\int_{-\infty}^{\infty} f(t - \tau)g(t)e^{-i\omega t} dt \quad (5)$$

Frog-Crab

Frog-Crab 的论文是 Y. Mairesse 和 F. Quere 2005 年的 *Frequency-resolved optical gating for complete reconstruction of attosecond bursts*. 以下的公式全部使用原子单位.

Strong Field Approximation

主要参考文献是 Lewenstein 1994 的 HHG 论文（见 Carlos Research），但这里另外做了修改.

令 \mathbf{E}_{XUV} 为 XUV 的电场，而 IR 电场用矢势来表示 $\mathbf{E}_{IR} = -\partial\mathbf{A}/\partial t$. 令末态波函数为

$$|\Psi(t, \tau)\rangle = e^{iI_p t} \left(|0\rangle + \int a(\mathbf{v}, \tau) |\mathbf{v}\rangle d^3\mathbf{v} \right) \quad (1)$$

强场近似下， $a(\mathbf{v}, \tau)$ 的解为²

$$a(\mathbf{v}, \tau) = -i \int_{-\infty}^{\infty} dt e^{i\phi(\mathbf{v}, t)} \mathbf{d}_{\mathbf{v}(t)} \mathbf{E}_{XUV}(t - \tau) e^{i(W + I_p)t} \quad (2)$$

$$\phi(\mathbf{v}, t) = - \int_t^{+\infty} dt' [\mathbf{v} \cdot \mathbf{A}(t') + \mathbf{A}^2(t')/2] \quad (3)$$

其中 $\mathbf{d}_{\mathbf{v}} = \langle \mathbf{v} | \mathbf{r} | 0 \rangle$ ， $|\mathbf{v}\rangle$ 为速度为 \mathbf{v} 的平面波，由于经典力学中 $\mathbf{p} = \mathbf{v} - \mathbf{A}$ 是一个守恒量³,

$$\mathbf{v}(t) = \mathbf{v}(\infty) - \mathbf{A}(\infty) + \mathbf{A}(t) = \mathbf{v} + \mathbf{A}(t) \quad (4)$$

下面来算 $\mathbf{d}_{\mathbf{k}}$ ，已知氢原子基态为 $|0\rangle = e^{-r}/\sqrt{\pi}$ ， $|\mathbf{k}\rangle = \exp(i\mathbf{k} \cdot \mathbf{x})/(2\pi)^{3/2}$. 由于氢原子基态球对称，不失一般性，可以将 \mathbf{k} 的方向设为极轴的方向（ z ），同样由对称性可得 $\mathbf{d}_{\mathbf{k}}$ 只有 z 分量不为零.

²我故意将论文中的 $\mathbf{d}_{\mathbf{p}(t)}$ 写成 $\mathbf{d}_{\mathbf{v}(t)}$ ，只是符号不同，计算是一样的.

³我实在不明白为什么许多论文上都写 $\mathbf{p} = \mathbf{v} + \mathbf{A}$!

$$\mathbf{d}_{\mathbf{k}} = \langle \mathbf{k} | \mathbf{r} | 0 \rangle = \frac{\hat{\mathbf{k}}}{\sqrt{2\pi}} \int_0^{+\infty} \int_0^\pi e^{-r} e^{-ikr \cos \theta} r \cos \theta \cdot r^2 \sin \theta d\theta dr \quad (5)$$

换元, 令 $u = \cos \theta$, 得

$$\begin{aligned} \mathbf{d}_{\mathbf{k}} &= \frac{\hat{\mathbf{k}}}{\sqrt{2\pi}} \int_0^{+\infty} r^3 e^{-r} \int_{-1}^1 e^{-ikru} u du \cdot d\mathbf{r} \\ &= i \frac{\sqrt{2}\hat{\mathbf{k}}}{\pi k} \int_0^{+\infty} r^2 e^{-r} \left[\cos(kr) - \frac{1}{kr} \sin(kr) \right] dr \\ &= -i \frac{8\sqrt{2}}{\pi} \frac{\mathbf{k}}{(k^2 + 1)^3} \end{aligned} \quad (6)$$

这与 Lewenstein 1994 的论文中给出的多了一个负号（不过我应该是对的）。

用 Matlab 根据上面的公式计算了一下 Frog-Crab trace, 但我的是对称的, 而论文上不对称, 师兄说这是论文上的 XUV pulse 有 chirp, 一看果然是. 想了一下 Phase Gate 是什么原理, 发现 $e^{i\phi(t)}$ 的确随 v 变化较小（见 “phi change.png”），但还有一项 $\mathbf{d}_{\mathbf{p}}$ 似乎随时间变化还是比较明显的, 魏晖的意思是也当成常数就行了, 不过他还给了我另一篇文章专门讨论 dipole 变化的, 存在 ipad 相册里面. 至于最后一项 $e^{i(W+I_p)t}$, 说明这是傅里叶变换到能量表象. 然而既然到能量表象了, 就不能直接使用 $|a(v, t)|^2$ 了, 而是需要做一个换元, 导致 $a(v)$ 需要除以 \sqrt{v} .

至于 dipole 随 v 的变化, 观察了一下发现基本上可以认为

$$d(v, t) \approx -i \frac{8\sqrt{2}}{\pi} \frac{v}{(v^2 + 1)^3} g(t); \quad (7)$$

其中 $g(t)$ 可看做一个不随 v 变化的时间函数, 其值约等于 $C_0 + \gamma A(t)$, γ 是一个较小的常数（见 “g(t) change.png”）。这样一来, 我们只需要将能量谱（先除以 \sqrt{v} ）除以上式中的 v 因子, 再用 PCGPA 就可以得到 pulse 和 gate, 而得到的 gate 将会是 $g(t) \exp[\phi(t)]$.

到此为止, Frog-Crab 基本上已经被我吃透了, 写出程序只是时间问题.

新的困难: 实验中产生的 trace 的时候只有 $E + I_p > 0$ 的部分, 然而完整的傅里叶变换 $\exp[i(E + I_p)t]$ 显示 $-(E + I_p)$ 处也有一条类似的 trace, 而且并不是完全对称的（因为被积函数既不是实函数也不是偶函数）。在论文上这点完全没有被提及. 试了一种方法就是在 PCGPA 保留负 efrog 不变, 而只把正

efrog 的模长换成正 trace 的模长，但是几乎看不出有 converge。从傅里叶变换的性质来看，pulse 是函数，gate 是复函数，相乘就是复函数，所以傅里叶变换以后正频谱和负频谱理论上并没有直接的关系。如果被傅里叶变换的函数是实函数，那么我们只需要正频谱就可以做反傅里叶变换，方法是用正频谱做反傅里叶变换乘以 2 然后取实部。但是不管了，反正是近似，就姑且这么做试试，结果效果还非常不错！这可能是因为 gate 的频率远小于 pulse 的频率吧！注意在每个循环中 SVD/power 后一定要把 pulse 的虚部归零。

另一个困难：由于 XUV+IR 至少需要 10000 个格点，Matlab 跑起来已经非常吃力了，现在一是要在 C++ 中实现，二是要进行优化。Numerical Recipes 中已经有 FFT 和 SVD 的算法了，然而 NR3 的 SVD 却不支持复数！然而没关系，我弄明白了 SVD 的 Principal Component 为什么可以用矩阵乘法代替，就是所谓的 power method，见 PCGPA 论文。

至于优化，首先算 SVD/power 的时候输入矩阵只有中间的一横条不为零，完全可以只对这部分做 SVD/power。另外 FFT 虽然比 DFT 要快，但如果用 DFT，就可以只对不为零的部分积分，且只算不为零部分的能谱，这样说不定会更快，另外还可以 openMP 并行。这些现在已经实现了，如果还要进一步优化，就是让 tau 的格点数（即 trace 的横向格点数）少于 gate 的格点数（实际试验中也肯定是这样的），但这样 outer product 矩阵会有很多零（见 sparse.png），就不知道 power method 还能不能用了，有待探索。不过感觉目前的速度已经够快了，弄两百个 iteration 不成问题。

现在来做实验，首先我根据 Frog-Crab 论文上的数据生成了 pulse 和 gate（并加入了 chirp），然后直接用 Frog 的方法生成 trace（而不是套用 SFA 公式）。结果非常理想，最多 100 多个循环就可以 converge 得非常完美，但这只是普通的 pcgpa，并没有近似（除了负频谱缺失）。

接下来可以试试用 SFA 公式了。

原创 Exact Frog-Crab 算法

这是我原创的一个算法，如果已知 $E_{IR}(t)$ 和 Frog-Crab trace，就可以精确解出

$$a(\mathbf{v}, \tau) = -i \int_{-\infty}^{\infty} dt e^{i\phi(\mathbf{v}, t)} \mathbf{d}_{\mathbf{v}(t)} \mathbf{E}_{XUV}(t - \tau) e^{i(W + I_p)t} \quad (1)$$

$$\phi(\mathbf{v}, t) = - \int_t^{+\infty} dt' [\mathbf{v} \cdot \mathbf{A}(t') + \mathbf{A}^2(t')/2] \quad (2)$$

而不需要任何近似.

首先我们把问题写为更简单的数学形式

$$S_{i,j} = \int_{-L_t/2}^{L_t/2} dt f(t) g_i(t + \tau_j) e^{i\omega_i t} \quad (3)$$

其中 i 取 N_E 个值, L_t 必须要比 $f(t)$ 的范围大, L_t 决定 **trace** 中必要的 ΔE , 即⁴ $\Delta\omega$. 如果已知 $E_{IR}(t)$, 就知道了 $g_i(t)$. 所以上式就是一个类似傅里叶变换的变换, 同样是丢失了变换后的 **phase** 信息.

如果能找到反变换, 那我们的思路就和 **PCGPA** 差不多, 甚至更简单. 只需要用一个 $f(t)$ 的 **guess**, 代入公式得到含相位的 a , 然后把模长替换成正确的模长, 反变换得到 $f(t)$, 然后再变换得到 a , 如此循环即可.

现在假设 $g_i(t + \tau_j)$ 在 $[-L_t/2, L_t/2]$ 上可以近似展开为傅里叶级数

$$g_i(t + \tau_j) = \sum_{k=-\infty}^{\infty} c_{ijk} e^{i\Delta\omega k t} \quad (4)$$

这样式 3 就变为

$$S_{i,j} = \sum_k c_{ijk} \int_{-L_t/2}^{L_t/2} dt f(t) e^{i\omega_{i+k} t} = \sum_k c_{ijk} F_{i+k} = \sum_l c_{i,j,l-i} F_l \quad (5)$$

其中 F 是 f 的反傅里叶变换, F_i 对应 ω_i , 所以式 4 中的 k 也最多只需要 $2N_E$ 个值. 所以注意 **trace** 的带宽只需比 $f(t)$ 的大即可.

式 5 是 F 的 $N_E N_\tau$ 条超定方程, 解超定方程用最小二乘法.

$$\mathbf{S} = \mathbf{C}\mathbf{F} \quad (6)$$

但事实上我们只知道矢量 \mathbf{S} 的每个矩阵元的模长, 我们同样可以用所谓的模长替换法迭代得到正确的 \mathbf{F} , 然后做一个傅里叶变换再取实部得 $f(t)$.

Phase Retrieval

与真正的 **Frog-Crab** 一样, 这种幼稚的方法收敛得非常慢. 这类只知道模长的线性方程组已经有相当多成熟的算法. [GitHub](#) 上找到一个包叫做 **PhasePack**

⁴但 $\omega_i = E_i + I_p$ 却不一定是 $\Delta\omega$ 的整数倍.

里面有很多算法而且统一接口。目前测试了 10 来种算法，其中最好用的是 Amplitude Flow，Wirt Flow 也不错，但是貌似初始 guess 不如用上面的模长替代法循环 100 次左右，用自带的初始 guess generator 反而不好。收敛的时间从半分钟到几分钟不等，有时候甚至会停止收敛。

原创 LSFC 算法

LSFC 的主要思想就是 GPA（基本就是 Gerchberg-Saxton algorithm）在一些情况下，这个收敛得很慢，远远没有直接优化快。所以我想尝试直接用最小二乘法单向优化来解 SFA 公式。

VTGPA 算法计算论文中的例子需要 1.5 小时，而我自己发明的 LSFC (Least Square Frog Crab) 只需要短短几分钟即可达到同样的效果。

参数选取

LSFC 的主要的思路是用一组参数确定 XUV，一组参数确定 IR。XUV 波包必须是完整的波包，即时域的采样范围要包含任何非零点。IR 的波包不必是完整的，只需要满足 XUV 的长度和 time delay 的要求即可。

目前 XUV 的参数选取其实数波函数的 DFT 非负频谱（第一个频率为零的格点必定是实数，其他格点为复数）。如果带宽比正半频谱宽度要窄，那就排除带宽以外的点。要从参数还原任何密度的 XUV，只需在参数两边适当添零再做反 DFT 即可。一般来说，只需要在左边添零就能满足条件。

由于 IR 在时域上并不需要是完整的波包（如果需要，就用与 XUV 相同的参数），用 XUV 同样的参数效率并不高或者可能有边缘效应。我尝试用 VTGPA 论文中的方法以振幅，初相位，频率和各阶 chirp 作为参数也并不理想，因为优化时很有可能出现很离谱的 local minimum 解。

现在的方法是，假设 IR 只需要知道一两个周期，倒不如直接用 $E_{IR}(t)$ （或者直接 $A(t)$ ）的散点作为参数，用 cubic spline 插值还原 $E_{IR}(t)$ 。使用越直接的参数，优化时 converge 越快。

IR 其实还有一个重要的参数就是 $t = +\infty$ 处矢势 A 的值。 $A(t)$ 是 $E_{IR}(t)$ 的反原函数，但却包含一个任意常数（对 trace 的影响不可忽略）。如果 E_{IR} 是完整的波包，我们或许可以令 $A(\pm\infty)$ 的其中一个为零，然而若我们只知道一

小段 E_{IR} , 任意常数就无从得知, 所以我们将 $A(+\infty)$ 放到参数中. 虽然同理对 $A(t)$ 积分后得到的 $\phi(t)$ 也有一个任意常数, 但在对 trace 取绝对值时, global phase 并没有任何影响.

优化算法

最简单的算法就是把所有参数放在一起优化, 但这么做的确很慢. 对 Frog-Crab trace 的公式稍加观察就会发现, 如果 IR 固定, trace 和 XUV 波函数离散点的关系可以用 generalized Phase-Retrieval 来描述

$$\left| \sum_j A_{ij} x_j \right| = b_i \quad (1)$$

其中矩阵 \mathbf{A} 只与 IR 有关. 如果事先计算好 \mathbf{A} , 使用这个公式比从头计算 trace 要快得多. 所以我们可以交替优化 XUV 和 IR 的参数, 也可以偶尔优化所有参数.

目前的算法是, 规定三种优化的最小循环次数. 先衡量三种优化的效率 (单位时间的 trace 误差减小), 然后选取效率最高的来优化一轮, 重新计算它的优化效率, 再次比较三者, 再选取效率最高的来优化第一轮, 重新计算它的优化效率, 再比较三者……

目前三个优化都是使用 Matlab 的 lsqnonlin() 函数中的 Levenberg-Marquadt 算法, 用差分计算梯度.

之前有考虑过用专门的 Phase-Retrieval 算法 (见 github 的 phase-pack-Matlab, 如 Wirt Flow) 来优化, 但 Phase-Retrieval 一般假设 x_j 是复数, 而离散的 XUV 波函数是实数, 所以测试得到了十分荒谬的结果. 如果有 x_j 是实数的 Phase-Retrieval 算法, 相信效率会比 LS 还要高许多. 但总时间应该不会有太大改进, 因为测试中, 单独优化 XUV 的时间并不占大比例.

也有考虑过用之前的 Exact-FrogCrab 算法来优化 XUV, 但是计算 \mathbf{A} 的时间就会更长 (因为还要做许多 DFT), 也是没太大必要.

Crank-Nicolson 算法（一维）

薛定谔方程为

$$-\frac{1}{2}\frac{\partial^2\psi}{\partial x^2} + V\psi = i\frac{\partial\psi}{\partial t} \quad (1)$$

用 Crank-Nicolson 或 Caley scheme⁵ 得到的结果是

$$\left(1 + \frac{i}{2}\mathbf{H}\Delta t\right)\psi^{n+1} = \left(1 - \frac{i}{2}\mathbf{H}\Delta t\right)\psi^n \quad (2)$$

其中二阶导数用三点差分计算，得

$$\begin{aligned} \psi_i^{n+1} - \psi_i^n &= \frac{i\Delta t}{4\Delta x^2}(\psi_{i-1}^n - 2\psi_i^n + \psi_{i+1}^n + \psi_{i-1}^{n+1} - 2\psi_i^{n+1} + \psi_{i+1}^{n+1}) \\ &\quad - \frac{i\Delta t}{2}(V_i^n\psi_i^n + V_i^{n+1}\psi_i^{n+1}) \end{aligned} \quad (3)$$

令 $\alpha = i\Delta t/(4\Delta x^2)$, $\beta = i\Delta t/2$ ，整理可得

$$-\alpha\psi_{i-1}^{n+1} + (1+2\alpha+\beta V_i^{n+1})\psi_i^{n+1} - \alpha\psi_{i+1}^{n+1} = \alpha\psi_{i-1}^n + (1-2\alpha-\beta V_i^n)\psi_i^n + \alpha\psi_{i+1}^n \quad (4)$$

其中 $\psi_i^n = \psi(x_i, t_n)$, $V_i^n = V(x_i, t_n)$.

我们把一个区间划分成 $N_x - 1$ 段等长的区间，并令 N_x 个格点为 $x_1 \dots x_{N_x}$. 最简单的边界条件是取 $\psi(x_1) = \psi(x_{N_x}) = 0$. 这样式 4 中的 i 可以取 $i = 2 \dots N_x - 1$ ，得到 $N_x - 2$ 条式子，其中只有 $\psi_2^{n+1} \dots \psi_{N_x-1}^{n+1}$ 这 $N_x - 2$ 个未知量，每条式子最多包含连续 3 个未知量. 将线性方程用矩阵表示，就可以得到一个三对角矩阵（第一行和最后一行只有两个系数）.

但事实上，还可以继续减少计算量. 将式 4 整理后得

$$\left(\frac{1}{2} + \frac{i}{4}\mathbf{H}\Delta t\right)(\psi^{n+1} + \psi^n) = \psi^n \quad (5)$$

解这个方程，再减去 ψ^n 即可.

⁵二者是一回事，见 Numerical Recipes 19.2

虚时间

使用虚时间后，式 4 和式 5 分别变为

$$\left(1 + \frac{1}{2}\mathbf{H}\Delta t\right)\psi^{n+1} = \left(1 - \frac{1}{2}\mathbf{H}\Delta t\right)\psi^n \quad (6)$$

$$\left(\frac{1}{2} + \frac{1}{4}\mathbf{H}\Delta t\right)(\psi^{n+1} + \psi^n) = \psi^n \quad (7)$$

虚时间法求基态波函数

这里介绍一种求解基态波函数的数值方法。如果薛定谔方程中势能不含时间，用分离变量法解薛定谔方程的结果是

$$\Psi(\mathbf{x}, t) = \sum_i \psi_i(\mathbf{x}) e^{-iE_i t} \quad (1)$$

其中 $\psi_i(\mathbf{x})$ 是能量为 E_i 的能量本征态。

现在若要求基态，我们可以用虚数时间，即 $t' = -it$ ，使得含时波函数变为

$$\Psi(\mathbf{x}, t) = \sum_i \psi_i(\mathbf{x}) e^{-E_i t} \quad (2)$$

这样，激发态衰减的速度就都比基态要快，当 $t \rightarrow +\infty$ 的时候，就只剩下基态波函数了。

假设我们有一个求解 TDSE 的数值方法，那么我们只需要用其求解方程

$$\mathbf{H}\Psi = -\frac{\partial \Psi}{\partial t} \quad (3)$$

然后每个循环对波函数进行归一化即可。因为该方程的分离变量解就是式 2。

函数求值

函数求值归根结底，就是找到一个解析公式，用加减乘除表示一个函数。如果算法没有用到近似，有限精度和任意精度都是一样的，只是每个数占的内存不同罢了。任意精度的加减乘除和开根号参考 Numerical Recipes 最后一节。

最常见的展开有三种，分别是泰勒展开，渐进展开，以及连续分数。这些展开在 functions.wolfram.com 都可以找到。

接下来是如何计算级数或连续分数。求级数的方法一般是用

$$f_N(x) = \sum_{n=0}^N c_n x^n = (\dots((c_n x + c_{n-1})x + c_{n-2} \dots)x + c_0 \quad (1)$$

这么做误差和直接对多项式求和一样，但计算量却少了很多。将上式中的 x 换成 $1/x$ 就是渐进展开的形式。注意我们可以在计算开始前估计误差，根据精度要求得到我们需要的项数。

显然， $|x|$ 越小时泰勒展开收敛得越快，而 $|x|$ 越大时渐进展开收敛得越快。再来看连续分数

$$f_N(x) = b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \dots \frac{a_N}{b_N}}} \quad (2)$$

也可以表示为

$$f_N(x) = b_0 + \frac{a_1}{b_1 +} \frac{a_2}{b_2 +} \frac{a_3}{b_3 +} \dots \frac{a_N}{b_N} \quad (3)$$

其中 a_n 和 b_n 都可以是 x 的函数。

乍看之下连续分数只能从右往左求，其实不然。由递归法可以证明

$$f_n = \frac{A_n}{B_n} \quad (4)$$

其中

$$A_n = b_n A_{n-1} + a_n A_{n-2} \quad B_n = b_n B_{n-1} + a_n B_{n-2} \quad (5)$$

$$A_{-1} = 1 \quad B_{-1} = 0 \quad A_0 = b_0 \quad B_0 = 1 \quad (6)$$

还有一种 Steed's 方法，详见 Numerical Recipes。正向求和的好处是可以判断什么时候开始收敛。据说特定情况下连续分数收敛较快，但具体什么时候用还有待考察。

另外两种不明觉厉的算法分别是切比雪夫多项式 (Chebyshev Polynomial) 和算数几何平均 (Algorithmic-Geometric Mean, AGM)，后者也通常被用于计算高精度的 π ，且有二次收敛。