# Shankar《量子力學》互動學習軟體規格設計

## 圖形使用者介面（GUI）設計

一款面向高中到大學程度學生的互動學習軟體需要直觀友好的GUI。介面應包含清晰的功能區和導航，使非程式背景的使用者也能輕鬆操作。設計要點包括：

* **章節導覽與內容選擇**：主視窗提供章節列表（對應 Shankar 教科書各章），使用者可點選進入相應章節的互動教材與模擬。每章界面包含該章的重點摘要、可播放的概念動畫，以及進入模擬與習題的按鈕。
* **模擬控制面板**：在進行量子模擬時，介面旁側顯示控制面板。提供下拉選單或按鈕選擇預設的典型系統（如「無限深方井」、「簡諧振子」、「氫原子」等），以及可讓使用者自行輸入自訂的波函數、勢能函數或算符。使用者輸入數學表達式時，系統可即時解析並檢查格式。還提供滑桿調整參數（如勢壘高度、初始波包位置和寬度等），並有「開始/暫停/重置」動畫的控制按鈕。
* **資料視圖區域**：主要區域用於顯示模擬結果的圖形和動畫。支援多視窗選項，例如同時顯示波函數演化動畫與相關量的即時圖表。圖形支持縮放、拖動與游標讀值，方便學生觀察數據。介面也應有選項切換顯示不同量（例如實部/虛部或機率密度）以及不同坐標系下的分佈。
* **互動操作**：提供可視化的操控，例如拖曳電子的初始位置、點擊按鈕應用某算符作用在波函數上等。學生可透過直觀操作來觸發模擬事件，而無需編寫程式碼。GUI 中的文字說明和按鈕標籤使用淺顯語言（搭配適當公式），降低學習門檻。
* **結果輸出與文本呈現**：在圖形區域下方或側邊，設置文字區域顯示相關計算結果（例如目前時間步長的$\langle x \rangle$、$\langle p \rangle$期望值等)，並以 LaTeX 排版數學式呈現，確保所見即所得。有需要時也可彈出對話框展示完整的符號推導過程或習題解析。

以上GUI元素應統一風格，美觀且響應迅速。建議採用成熟的跨平台GUI框架（如 **Qt** 框架的 Python 綁定）來構建介面；**PyQt/PySide** 是常用選擇，其完善的元件和跨平台能力非常適合構建複雜桌面應用[[1]](https://fullscale.io/blog/python-gui-frameworks/#:~:text=Which%20Python%20GUI%20is%20best,for%20desktop%20applications)。為提升繪圖效能與互動性，可考慮使用 **PyQtGraph** 這類基於 PyQt 的高效科學繪圖庫，它支持即時繪圖、互動操作，甚至GPU加速的2D/3D圖形[[2]](https://pyqtgraph.com/#:~:text=Blazing,PyQtGraph)[[3]](https://pyqtgraph.com/#:~:text=,PySide)。

## 章節內容與互動學習模組

軟體將 Shankar《量子力學》各章內容模組化，每章提供對應的教學素材和互動畫面，使學生能循序漸進掌握該章概念。每個章節模組包含：

* **重點內容講解**：摘錄該章的重要概念和定理，以簡明文字和公式呈現，可配合插圖或簡短動畫闡明抽象概念。例如在介紹布拉-克特記號的章節，介面上可顯示線性空間投影的示意動畫。在測不準原理章節，顯示高斯波包位置變窄時動量空間變寬的示意圖等。
* **視覺化互動範例**：針對章節內容設計1～2個互動模擬。學生可透過調整參數觀察理論現象。例如：「粒子在一維勢壘穿透」（對應障礙穿透章節）模組中，學生可調變障礙高度/寬度，實時看到波包透射與反射的變化動畫；又如在角動量章節，提供「旋量的布洛赫球表示」互動圖形，學生可拖動方向向量，觀察對應自旋態在 Bloch 球上的變化。
* **符號推導工具**：每章節重要計算提供內建的符號計算說明或工具。例如在算符對易關係章節，學生可選擇兩個算符（或直接輸入算符表達式），點擊「計算對易子」，軟體將利用符號模組算出$[A,B] = AB-BA$的結果並判斷是否為零[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators)。再如在討論 Hermitian 算符的章節，提供「Hermitian 檢查」按鈕，學生輸入一算符矩陣或表達式，系統即時計算其厄密共軛是否等於自身，以回答「此算符是否 Hermitian」這類問題。
* **練習題與解析**：每章配套數道典型練習題，題型可包含選擇、填空或簡答。學生可在介面中作答並提交，系統即時判分或給出提示。對於計算類問題，學生可打開軟體內建的草稿介面利用符號工具求解。作答後可點擊「顯示解析」查看詳細解題步驟與說明，鞏固對該章內容的理解。習題解析部分也應適度圖文化，例如畫出波函數節點分布圖輔助說明答案。

透過上述多元模組，每章節內容都不僅有靜態文字，還結合了動態的視覺演示與練習互動，形成**閱讀-觀察-實作**的閉環學習體驗。這確保學生對 Shankar 教科書中每章理論都有直觀理解，並能親手驗證核心原理。

## 數值計算模擬引擎

核心的數值模擬引擎支撐著整套軟體的互動體驗。該引擎負責在背後進行**薛定諤方程求解、時間演化計算以及各種算符作用的數值模擬**，對使用者輸入給出即時反饋。主要功能與演算法設計如下：

* **定態薛定諤方程求解**：對於給定的勢能函數 $V(x)$，引擎需要解決定態薛定諤方程 $H\psi = E\psi$，求出能量本徵值和本徵態。實作上可採用有限差分離散空間網格的方法將哈密頓算符表示成矩陣，再調用 SciPy 等科學庫的**矩陣特徵值求解器**來計算本徵值問題[[5]](https://findiff.readthedocs.io/en/latest/source/examples-matrix.html#:~:text=The%20Schr%C3%B6dinger%20equation%20can%20then,scipy)。例如使用 SciPy 稀疏線性代數模組的 eigs 或 eigh 函數求出最低幾個本徵態[[5]](https://findiff.readthedocs.io/en/latest/source/examples-matrix.html#:~:text=The%20Schr%C3%B6dinger%20equation%20can%20then,scipy)。SciPy 提供了完備的優化、積分和特徵值求解功能，可滿足此類數值計算需求[[6]](https://arxiv.org/html/2412.04705v1#:~:text=SciPy%20%20%20Scientific%20Python%3A,computing%2C%20with%20automatic%20differentiation%20capabilities)。計算結果將傳回前端，用於初始化動畫或繪製幾何量圖像。
* **時間依賴薛定諤方程模擬**：為實現波函數隨時間的演化，數值引擎需實作穩定高效的時間積分算法。一種可靠方案是採用 **Crank–Nicolson 隱式差分**方法對薛定諤方程進行時間離散[[7]](https://blog.c0nrad.io/posts/sim-10-crank-that-nicolson/#:~:text=Crank%20Nicolson)。Crank–Nicolson 屬於無條件穩定的算法，可在每個時間步解一線性方程組來更新波函數[[7]](https://blog.c0nrad.io/posts/sim-10-crank-that-nicolson/#:~:text=Crank%20Nicolson)。由於薛定諤方程的演化算符是酉矩陣，也可選用直接計算時間演化算符$U=\exp(-iHt/\hbar)$的方法：對時間獨立哈密頓量，先對角化$H$取得$H=WDW^{-1}$，再快速計算$\exp(-iHt) = W \exp(-iDt) W^{-1}$；或對較小矩陣使用 SciPy 的矩陣指數函數 expm。另一種作法是利用現有的量子模擬庫（如 **QuTiP**）提供的時間演化求解器，它封裝了各種 ODE 積分器，使使用者可方便模擬量子態在任意哈密頓作用下的演進[[8]](https://arxiv.org/html/2412.04705v1#:~:text=At%20its%20core%2C%20QuTiP%20is,in%20QuTiP%E2%80%99s%20concept%20of%20a)。
* **波包傳播與干涉**：引擎應允許使用者自訂初始波函數，例如高斯波包、疊加態等。對自由粒子或指定勢場中的波包傳播，可結合傅立葉變換技巧計算演化（自由粒子可在動量空間相乘相位因子後逆變換回位置空間）。對一般勢能則按上述時間步進積分計算。為展示干涉效應，可允許設定雙波包初始條件，演化過程中計算兩子波函數的總機率密度，觀察干涉條紋形成。
* **算符作用模擬**：實現動量$\hat{p}=-i\hbar\frac{\partial}{\partial x}$、角動量$\hat{L}$、升降算符$\hat{a}, \hat{a}^\dagger$等對態的作用。對於位置表象中的波函數，動量算符作用可透過傅立葉變換到動量空間乘以$p$再轉回得到結果（或直接在位置空間用差分近似微分）。角動量算符在球坐標可作用於球諧函數，升降算符作用可在諧振子本徵態基底用解析公式計算（例如$\hat{a}|n\rangle=\sqrt{n}|n-1\rangle$）並轉換回$x$表象繪圖。引擎中的算符定義資料結構應包含其矩陣表示或作用規則，使得一旦使用者選擇對當前態施加某算符，數值模組能據內建的矩陣或算法即時算出新態函數。
* **常見物理系統內建**：引擎預載多種典型系統的解析或數值解。例如無限深方井、有限方勢阱、簡諧振子等的一維系統已有解析本徵解，可直接套用公式產生本徵態函數（如正弦函數、Hermite 多項式等）。氫原子等三維系統可內建其已知的能階和波函數（球諧函數和徑向解），供學生調用查看。然而為增強自由度，軟體仍允許這些系統通過數值方法重新求解，以便學生比較數值解與解析解的差異、驗證正確性。

數值引擎需注意**效能和穩定性**。藉助高效的Python科學計算庫（如 NumPy, SciPy），搭配向量化運算以及必要的稀疏矩陣技巧，可在保證精度的同時加快計算。若遇到大尺度計算需求，可評估引入 JAX、Numba 等加速器。整體而言，此模擬引擎將提供可靠的數值基礎，使前端互動教學內容基於嚴謹的計算結果[[6]](https://arxiv.org/html/2412.04705v1#:~:text=SciPy%20%20%20Scientific%20Python%3A,computing%2C%20with%20automatic%20differentiation%20capabilities)。

## 符號計算模組

為滿足教學中推導計算的需求，軟體將內建一個符號計算模組，透過調用 Python 的符號運算庫（如 **Sympy**）來處理各種代數計算[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators)。該模組提供如下功能：

* **對易子與運算符代數**：使用者可在介面中輸入兩個算符$A, B$（預置常見算符或由使用者以符號矩陣/表達式定義），點擊計算後，模組會利用 Sympy 計算$[A,B]=AB-BA$並簡化結果[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators)。對於較複雜的算符代數，結果將以交換關係的簡化形式呈現（例如給出$[x,p]=i\hbar$）。模組也能識別算符是否對易（結果是否為0）並提示學生物理意義。
* **Hermitian/單位算符檢測**：符號模組可接受一個算符表達式或矩陣，計算其達格算符（厄密共軛） $A^\dagger$，然後比較$A^\dagger$與$A$是否相等，以判斷 Hermitian 性[[9]](https://docs.sympy.org/latest/modules/physics/quantum/operator.html#:~:text=class%20sympy.physics.quantum.operator.HermitianOperator%28)[[10]](https://docs.sympy.org/latest/modules/physics/quantum/operator.html#:~:text=Examples)。若相等則輸出「此算符為 Hermitian 算符」，否則給出反例（例如導出$A^\dagger = -A$暗示反Hermitian）。對於單位算符判斷則計算$A^\dagger A$與單位矩陣的差是否為零矩陣。
* **狄拉克記號展開**：提供符號工具將 kets 和 bras 表達式展開至坐標表象。例如輸入態$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$，符號模組可利用預知的基底展開規則輸出對應坐標波函數$\psi(x)$的形式，或將疊加態表示為矩陣列向量。Sympy 的量子模組支持 Hilbert 空間、態矢、內積/外積等操作[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators)，可用於實現這類符號轉換。另外，模組允許計算內積 $\langle\phi|\psi\rangle$ 的符號值，以及對應的歐幾里得范數檢查態是否歸一。
* **符號微分與積分**：針對教材中的解析推導，如勢能函數的梯度、動量算符作用等，符號模組可以對給定的波函數進行微分運算，或對機率密度進行積分以計算規範化常數等。這讓學生可以驗證手動推導的過程，例如親自輸入$\psi(x)$，點擊「計算$\int |\psi|^2 dx$」，系統給出積分結果及判斷是否等於1（是否歸一化）。
* **線性代數操作**：許多量子力學推導涉及矩陣與向量。符號模組允許生成符號矩陣表示算符（例如角動量$L\_z$在|l,m>基底的矩陣），並執行矩陣運算（乘法、對易、求逆等）。Sympy 可處理符號矩陣的特徵值問題，因而甚至可以符號地求解某些簡單系統的本徵值。雖然對高維系統符號計算可能較慢，但對低維示範（如2x2自旋-$\frac{1}{2}$系統）非常有用，學生可看到特徵方程的解析解形式。

符號計算模組的輸出將與GUI整合，以易讀形式展現。如算式推導過程可在GUI的文本區逐步顯示，重點結果用顏色標示。透過Sympy強大的物理量子模組支援（涵蓋了態、算符、對易子、厄密共軛等概念）[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators), 軟體能夠**象徵性地演示教科書中的推導**，幫助學生確認自己推算的正確性，亦可加深對公式背後依賴的代數關係的理解。

## 視覺化與動畫結果呈現

直觀的視覺化是本軟體教學效果的關鍵。為了做到**所見即所得**，軟體需要將抽象的量子態和運算結果轉化為圖形和動畫呈現給使用者。主要的視覺化內容及其實現構想包括：

* **波函數實時動畫**：在一維系統中，以動畫形式顯示波函數 $\Psi(x,t)$ 的演化。繪圖時可同時顯示實部（實線）與虛部（虛線），以及機率密度 $|\Psi|^2$（陰影區域）。透過逐幀動畫，學生可以觀察波包在勢阱中來回反射、干涉條紋形成或隨時間散射的過程。圖表附有坐標軸和刻度，並在標題處標示當前時間$t$值。可選擇加速或減慢播放速度，以便詳細觀察特定時刻的行為。
* **幾何量圖形化**：將量子態隨時間的**期望值**、**機率分佈**等以圖表呈現。例如繪製$\langle x \rangle(t)$ 和 $\langle p \rangle(t)$的隨時間變化曲線，或將不確定度$\sigma\_x(t), \sigma\_p(t)$疊加顯示，用於直觀驗證何時達到最小不確定關係等。對於定態解，則顯示不同能級的能量譜圖，以及對應的波函數形狀（可疊加在勢能曲線上展示節點位置）。在展示氫原子徑向分佈時，可繪製$4\pi r^2 |R\_{nl}(r)|^2$對$r$的曲線，標示出幾個概率最大值處。所有這些圖形應支持互動：滑鼠懸停顯示數值、點擊某條曲線可高亮或隱藏。
* **算符矩陣與基底變換**：為幫助理解不同基底表示，軟體提供矩陣表示的可視化。當學生選擇特定算符和基底（如動量算符在位置表象），系統計算出矩陣後，以表格影像形式展示矩陣元素（可以色深或大小表示元素複數大小）。例如顯示升降算符在諧振子本徵基${|0\rangle,|1\rangle...}$下的矩陣是偏移的次對角矩陣。另提供**基底變換**動畫，例如從位置基底變到動量基底，矩陣隨之做相似轉換，圖形上反映為矩陣元素的重新排列。
* **模態分解與正交展開**：對於給定的非本徵態，軟體能計算其在某組本徵態基底下的展開係數，並以長條圖方式顯示各模態的佔比。例如將一個波包狀態展開在無限深方井的能量本徵態上，用柱狀圖顯示不同量子數 $n$ 分量的概率 $|c\_n|^2$。這直觀展示態是如何由許多本徵模態組成以及隨時間的係數變化（對自由態係數恆定，對非對易觀察量隨時間變化）。正交基底的展開也可用於驗證基底完整性之概念——當選取正交歸一的一組基底時，展開係數機率之和應為1，軟體可在圖上標示此和的值供學生比對。
* **三維與其他高維視覺化**：儘管大部分互動側重於一維情形，為涵蓋如氫原子等三維案例，軟體可提供簡化的3D視覺化工具。例如顯示氫原子軌域的三維等值面圖或者在二維平面上以顏色表示截面機率密度。自旋-$\frac{1}{2}$等雙態系統則可利用 **Bloch 球**展示其態向量的位置，並動畫演示疊加態在 Bloch 球上的旋進。由於3D顯示較複雜，可選用預渲染模型或引入輕量3D繪圖庫，確保性能流暢。

為實現上述圖形效果，可採用 Matplotlib 繪圖並將其嵌入 GUI，或使用 PyQtGraph 獲得更流暢的動畫效果[[2]](https://pyqtgraph.com/#:~:text=Blazing,PyQtGraph)。這些圖形皆屬**所見即所得**：學生對某參數的操作將幾乎同步地反映在圖像更新上，強化實驗操作與理論結果的因果連結。所有圖例和數據標注也將清晰可辨，使學生在視覺體驗中獲取精確資訊而非模糊印象。

## AI 助理問答整合

為提升互動體驗，軟體可考慮加入**AI 助理**模組，允許學生以自然語言提問並獲得即時解答。這部分可透過調用大語言模型（LLM）介面（如 OpenAI GPT 系列）或本地部署的模型來實現。設計構想如下：

* **概念疑問解答**：學生在使用軟體過程中若有理論概念不理解，可直接在介面中的聊天對話框輸入問題，如「為什麼動量算符是 Hermitian？」或「測不準原理的數學表達是什麼？」。AI 助理將基於內置的教科書知識或上下文進行解釋，引導學生理解概念。如果涉及計算步驟，助理也可列出主要推導步驟或引用相關章節內容。
* **計算檢查與輔助**：學生進行某符號或數值計算時，可以詢問AI助理來驗證。例如學生手算得到一結果，可問：「這個推導結果對嗎？」AI 可拿到目前軟體的計算上下文（如當前算符矩陣或計算值）進行比對，給出反饋。如果學生問「這算符是 Hermitian 嗎？」，AI 可以調用軟體後端符號模組檢查算符 Hermitian 性，再用自然語言回答是或否並給出理由。
* **操作建議**：AI 助理還可根據學生當前學習進度給出下一步建議。例如學生長時間停留在某模擬，可以提示「你可以嘗試改變一下初始波包的寬度，看看有何不同」。或者在學生完成一章練習後，AI 助理可提供進一步的延伸閱讀或額外習題來源。
* **技術實現**：AI 助理需要與軟體內部資料溝通。可透過定義一套API，允許AI讀取當前模擬參數、計算結果等作為上下文，然後經由語言模型生成回答。初期可簡單地將使用者問題和相關數據發送至雲端LLM服務獲取答覆；若要求本地離線，則可整合輕量的本地模型（取決於2025年可用的開源模型性能）。介面上以對話形式呈現問答，並支持點選過往回答中高亮的專有名詞以獲得定義（超連結到章節內容或維基百科等）。

AI 助理的定位是**學習的輔助工具**而非替代品，回答會控制在提示和方向引導上，鼓勵學生自行探索答案。同時AI的引入能讓軟體更具互動性和智能化，使學生彷彿有個24小時在線的家教，可以隨時發問解惑。

## 框架與開發實現建議

綜合以上功能需求，開發此互動學習軟體可採用**模組化、以 Python 為主的技術棧**實現，具體建議如下：

* **編程語言與總體架構**：選擇 Python 作為開發語言。Python 擁有豐富的科學計算和GUI庫，可以快速實現原型並具備可讀性。採用**MVC（模型-視圖-控制）架構**將邏輯與界面分離：模型層負責量子計算（數值模型、符號模型），視圖層是Qt GUI，控制器負責協調用戶操作與模型計算。如此便於團隊並行開發與日後維護擴展。
* **GUI 框架**：使用 **PyQt5/PySide6** 等 Qt 綁定構建桌面應用。Qt 元件豐富且跨平台表現佳，非常適合構建複雜科學應用[[1]](https://fullscale.io/blog/python-gui-frameworks/#:~:text=Which%20Python%20GUI%20is%20best,for%20desktop%20applications)。介面設計可借助 Qt Designer 等工具加速。繪圖方面，建議嵌入 **Matplotlib** 圖表到Qt介面以利用其豐富的科學繪圖功能，或直接採用 **PyQtGraph** 提升實時繪圖效能[[2]](https://pyqtgraph.com/#:~:text=Blazing,PyQtGraph)。PyQtGraph 與 Qt 無縫整合，且專為科學數據視覺化優化[[3]](https://pyqtgraph.com/#:~:text=,PySide)。
* **數值計算庫**：核心數值模擬使用 **NumPy/SciPy**。NumPy提供高效矩陣運算，SciPy則有數值積分、FFT、線性代數特徵值求解等眾多功能[[6]](https://arxiv.org/html/2412.04705v1#:~:text=SciPy%20%20%20Scientific%20Python%3A,computing%2C%20with%20automatic%20differentiation%20capabilities)。可以利用 SciPy 的 solve\_ivp 進行時間演化的ODE數值積分，或使用其線性代數模組進行本徵值問題求解[[5]](https://findiff.readthedocs.io/en/latest/source/examples-matrix.html#:~:text=The%20Schr%C3%B6dinger%20equation%20can%20then,scipy)。如需提升性能，可考慮 **Numba** 將熱點演算法JIT編譯，或 **JAX** 利用GPU加速（視開發資源決定）。
* **專用量子軟體庫** (選項)：考慮集成開源的 **QuTiP (Quantum Toolbox in Python)** 庫。QuTiP 提供了高階的量子物理計算物件和solver，可方便地構造態矢量、算符並模擬其時間演化[[8]](https://arxiv.org/html/2412.04705v1#:~:text=At%20its%20core%2C%20QuTiP%20is,in%20QuTiP%E2%80%99s%20concept%20of%20a)。例如使用 QuTiP，可在幾行代碼內建立哈密頓量並呼叫內建的 mesolve 進行薛定諤方程演化，同時獲取觀察量期望值結果[[11]](https://arxiv.org/html/2412.04705v1#:~:text=3)。這將大幅減少自行實現複雜演算法的工作量。不過，引入外部庫需留意其學習曲線並確保與本軟體框架的整合順暢。
* **符號計算庫**：使用 **Sympy** 作為符號計算後端。Sympy 的 physics 子模組已支持量子態、算符、內積、對易子等概念[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators)，能滿足推導驗證所需。將Sympy計算封裝在背景執行，結果轉為LaTeX在前端呈現。注意控制符號計算規模以防止卡頓（必要時對大型表達式採用數值近似輔助）。
* **開發流程**：建議先構建**原型**實現關鍵的模擬功能和一兩個章節示範，以驗證技術可行性。之後採用**迭代開發**方式逐章增添內容模組。開發團隊可分工：一組專注於數值算法和後端架構（確保計算正確高效），另一組專注於前端界面與用戶體驗（確保操作流暢直觀）。定期用真實學生測試原型，收集反饋優化UI和教學設計。
* **擴充與維護**：系統架構應具備良好擴充性。例如章節內容採用插件式載入，使未來可新增更多主題（甚至涵蓋其他教材）。代碼採用單元測試驗證關鍵計算模組的正確性，保障推廣使用時的可靠。文檔和註解齊全，以方便日後接手維護或開源社群共建。

總而言之，此軟體的開發應結合**成熟框架**與**專業算法庫**來降低實現難度，又要注意軟體工程方法確保產品品質。依照上述規格進行實作，可打造出一套功能完整、高效穩定的量子力學互動學習平台，真正將 Shankar 教科書的深奧理論轉化為學生可親身探索的直觀體驗。[[1]](https://fullscale.io/blog/python-gui-frameworks/#:~:text=Which%20Python%20GUI%20is%20best,for%20desktop%20applications)[[6]](https://arxiv.org/html/2412.04705v1#:~:text=SciPy%20%20%20Scientific%20Python%3A,computing%2C%20with%20automatic%20differentiation%20capabilities)

[[1]](https://fullscale.io/blog/python-gui-frameworks/#:~:text=Which%20Python%20GUI%20is%20best,for%20desktop%20applications) 10 Powerful Python GUI Frameworks for 2024: Simplify Your Desktop App Development

<https://fullscale.io/blog/python-gui-frameworks/>

[[2]](https://pyqtgraph.com/#:~:text=Blazing,PyQtGraph) [[3]](https://pyqtgraph.com/#:~:text=,PySide) PyQtGraph - Fast 2D/3D Scientific Plotting in Python

<https://pyqtgraph.com/>

[[4]](https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp#:~:text=,inner%2Fouter%20products%2C%20operators%2C%20commutators) [PDF] Symbolic Quantum Circuit Simplification in SymPy

<https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1039&context=physsp>

[[5]](https://findiff.readthedocs.io/en/latest/source/examples-matrix.html#:~:text=The%20Schr%C3%B6dinger%20equation%20can%20then,scipy) Schrödinger Equation — findiff 0.11.1 documentation

<https://findiff.readthedocs.io/en/latest/source/examples-matrix.html>

[[6]](https://arxiv.org/html/2412.04705v1#:~:text=SciPy%20%20%20Scientific%20Python%3A,computing%2C%20with%20automatic%20differentiation%20capabilities) [[8]](https://arxiv.org/html/2412.04705v1#:~:text=At%20its%20core%2C%20QuTiP%20is,in%20QuTiP%E2%80%99s%20concept%20of%20a) [[11]](https://arxiv.org/html/2412.04705v1#:~:text=3) QuTiP 5: The Quantum Toolbox in Python

<https://arxiv.org/html/2412.04705v1>

[[7]](https://blog.c0nrad.io/posts/sim-10-crank-that-nicolson/#:~:text=Crank%20Nicolson) Sim 10: Numerically solving Schrodinger via Crank-Nicolson in C++

<https://blog.c0nrad.io/posts/sim-10-crank-that-nicolson/>

[[9]](https://docs.sympy.org/latest/modules/physics/quantum/operator.html#:~:text=class%20sympy.physics.quantum.operator.HermitianOperator%28) [[10]](https://docs.sympy.org/latest/modules/physics/quantum/operator.html#:~:text=Examples) Operator - SymPy 1.14.0 documentation

<https://docs.sympy.org/latest/modules/physics/quantum/operator.html>