We’ll assume all the constants in the Schrödinger equation are one, so the differential equation is

\[ i \frac{\partial \Psi(x,t)}{\partial t} = -\Delta \Psi(x,t) + V(x)\Psi(x,t). \]

(See http://vergil.chemistry.gatech.edu/notes/quantrev/node9.html.) We write \( \Psi(x,t) = \psi(x,t) + i\phi(x,t) \); \( V(x) \) is real, with \( \Delta \) the Laplacian operator.

Let’s think of \( \Psi(x,t) = (\psi(x,t) \ \phi(x,t)) \) as a vector. Then multiplying by \( i \) in the original formulation is the same as multiplying by the matrix

\[ \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \]

in vector form, so we write

\[ \frac{\partial}{\partial t} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \Psi(x,t) = \begin{pmatrix} -\Delta + V(x) & 0 \\ 0 & -\Delta + V(x) \end{pmatrix} \Psi(x,t). \]

Let’s consider the Crank–Nicolson method for time-stepping. Here we have

\[ \left( \begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array} \right) \frac{\Delta t}{2} \frac{[\Psi(x,t^k) - \Psi(x,t^{k-1})]}{\Delta t} = \begin{pmatrix} -\Delta + V(x) & 0 \\ 0 & -\Delta + V(x) \end{pmatrix} \frac{[\Psi(x,t^k) + \Psi(x,t^{k-1})]}{2}. \]

Multiplying by \( \Delta t \) and collecting all the \( \Psi(x,t^k) \) terms together gives

\[ \frac{\Delta t}{2} \left( \Delta - V(x) \right) \left[ \Psi(x,t^k) - \Psi(x,t^{k-1}) \right] = \begin{pmatrix} -\frac{\Delta t}{2} (\Delta - V(x)) & -1 \\ 1 & -\frac{\Delta t}{2} (\Delta - V(x)) \end{pmatrix} \Psi(x,t^{k-1}). \]

If we use finite differences for the spatial discretization, then \( \frac{\Delta t}{2} (\Delta - V(x)) \) will be replaced by a finite-difference operator \( A_h \) and 1 will be replaced by the identity matrix \( I \) and you need to invert the \( 2 \times 2 \) block matrix

\[ \left( \begin{array}{cc} A_h & -I \\ I & A_h \end{array} \right). \]

Because \( I \) commutes with \( A_h \), simple matrix multiplication shows that the inverse of this matrix is

\( (I + A_h^2)^{-1} \left( \begin{array}{cc} A_h & I \\ -I & A_h \end{array} \right). \)

Then the iteration is

\[ \Psi(x,t^k) = (I + A_h^2)^{-1} \left( \begin{array}{cc} A_h & I \\ -I & A_h \end{array} \right) \left( \begin{array}{cc} -A_h & -I \\ +I & -A_h \end{array} \right) \Psi(x,t^{k-1}). \]

\[ = (I + A_h^2)^{-1} \left( \begin{array}{cc} I - A_h^2 & -2A_h \\ 2A_h & I - A_h^2 \end{array} \right) \Psi(x,t^{k-1}). \]

\[ = (I + A_h^2)^{-1} \left( \begin{array}{cc} I + A_h^2 - 2A_h^2 & -2A_h \\ 2A_h & I + A_h^2 - 2A_h^2 \end{array} \right) \Psi(x,t^{k-1}). \]

\[ = \Psi(x,t^{k-1}) + 2(I + A_h^2)^{-1} \left( \begin{array}{cc} -A_h^2 & -A_h \\ A_h & -A_h^2 \end{array} \right) \Psi(x,t^{k-1}). \]
The software in this class uses the finite element method with piecewise-linear elements. For a totally discrete problem we replace $\Psi$ with its finite element approximation and $v$ and $\frac{\nabla^2}{2}(\Delta - V(x))$ with the matrices $B$ and $A$ with

$$B_{ij} = \int_{\Omega} \Phi_j \Phi_i \, dx \text{ and } A_{ij} = -\frac{\Delta t}{2} \int_{\Omega} \nabla \Phi_j \cdot \nabla \Phi_i + V \Phi_j \Phi_i \, dx,$$ respectively,

where $\{\Phi_j\}$ is a basis for the finite element space. Then we want to invert the $2 \times 2$ block matrix

$$(1) \quad \begin{pmatrix} A & -B \\ B & A \end{pmatrix}.$$

But now $A$ and $B$ don’t commute (something that may not be obvious, but which can easily be checked computationally), but we can compute

$$(2) \quad \begin{pmatrix} A & -B \\ B & A \end{pmatrix}^{-1} = \begin{pmatrix} AB^{-1} & -I \\ I & AB^{-1} \end{pmatrix}^{-1} = B^{-1} \begin{pmatrix} AB^{-1} & -I \\ I & AB^{-1} \end{pmatrix}^{-1} = B^{-1}(I + AB^{-1}A^{-1})^{-1} \begin{pmatrix} AB^{-1} & -I \\ I & AB^{-1} \end{pmatrix} = [(I + AB^{-1}A^{-1})B]^{-1} \begin{pmatrix} AB^{-1} & -I \\ I & AB^{-1} \end{pmatrix} = (B + AB^{-1}A)^{-1} \begin{pmatrix} AB^{-1} & -I \\ I & AB^{-1} \end{pmatrix}.$$

So to invert (1) we need to compute $Az$, $B^{-1}z$, and $(B + AB^{-1}A)^{-1}z$ for any $z$.

The complete iteration, after some simplification, is

$$\Psi(x, t^k) = \Psi(x, t^{k-1}) + 2(B + AB^{-1}A)^{-1} \begin{pmatrix} -AB^{-1}A \\ A \end{pmatrix} \Psi(x, t^{k-1}).$$

In the course we covered the (preconditioned) conjugate-gradient (CG) method and the multigrid (MG) method for solving linear systems $Ay = z$, or equivalently, to calculate $y = A^{-1}z$ for an operator $A$. The important thing about both CG and MG is that one only multiplies by $A$; that’s it. (The Richardson smoother for $A$ multiplies by $A$ and then applies a few vector operations to finish.)

So let’s consider how to compute (2) using CG. We need to be able to compute $y = B^{-1}z$ for any finite-element vector $z$, or equivalently to solve $By = z$. Now, $\kappa(B)$, the condition number of $B$, satisfies $\kappa(B) = O(1)$, i.e., it doesn’t depend on $h$ at all, so one can solve $By = z$ with un-preconditioned CG in a small number of steps that doesn’t depend on $h$. Each step requires one multiplication by $B$ and a few vector operations. As $B$ is sparse, it has a bounded number of nonzero elements in each row, so each multiplication by $B$ takes $O(N)$ operations, so CG takes $O(N)$ operations to compute $B^{-1}z$ for any $z$ to within machine accuracy.

Then we need to compute $(B + AB^{-1}A)^{-1}z$ for any $z$, or equivalently, solve $(B + AB^{-1}A)y = z$. Again, using CG, we just need to multiply by $(B + AB^{-1}A)$. Again, $A$ and $B$ are sparse, so applying either $A$ or $B$ takes $O(N)$ operations; the previous paragraph shows that applying $B^{-1}$ takes $O(N)$ operations; so multiplying by $(B + AB^{-1}A)$ takes $O(N)$ operations.
The error bound for CG applied to $Ay = z$ states that $y_k$, the approximate solution after $k$ steps of CG, satisfies
\[ \|y - y_k\|_A \leq \left( \frac{\kappa(A)^{1/2} - 1}{\kappa(A)^{1/2} + 1} \right)^k \|y - y_0\|_A, \]
so in our case it's important to get a reasonable bound for $\kappa(B + AB^{-1}A)$.

We have $\kappa(A) = \|A\| \|A^{-1}\|$ for whichever matrix norm $\| \cdot \|$ we'd like to choose, so
\[ \kappa(AB) = \|AB\| \|\kappa(AB)^{-1}\| = \|AB\| \|B^{-1}A^{-1}\| \leq \|A\| \|B\| \|B^{-1}\| \|A^{-1}\| = \kappa(A)\kappa(B). \]
Because $(B + AB^{-1}A) = B^{1/2}(I + B^{-1/2}AB^{-1/2}B^{-1/2}AB^{-1/2})B^{1/2}$ and $\kappa(B^{1/2}) = O(1)$, the previous inequality shows that we just need to bound $\kappa(C)$, where $C = (I + B^{-1/2}AB^{-1/2}B^{-1/2}AB^{-1/2})$.

The matrices $A$, $B$, $C$, and $I$ are all symmetric, so we'll use the matrix 2-norm. Every vector is an eigenvector of the identity, so the eigenvectors of $C$ are the eigenvectors of $B^{-1/2}AB^{-1/2}$, and the eigenvalues of $C$ are $1 + \lambda_i^2$, where $\lambda_i$ ranges over the eigenvalues of $B^{-1/2}AB^{-1/2}$.

Thus the smallest eigenvalue of $C$ is $O(1)$, while the absolute value of the largest eigenvalue of $B^{-1/2}AB^{-1/2}$ is bounded by
\[ \sup_x \frac{|x^T B^{-1/2} AB^{-1/2} x|}{x^T x} = \sup_y \frac{|y^T Ay|}{y^T By} = O(\Delta t h^{-2}). \]

Tracing things back, we get $\kappa(B + AB^{-1}A) = O(\Delta t^2 h^{-4})$.

For Crank-Nicolson we'd like to take $\Delta t = h$ (since the total error is likely to be $O(\Delta t^2 + h^2)$), in which case $\kappa(B + AB^{-1}A) = O(h^{-2})$. (On a uniform $65 \times 65$ triangulation with one set of diagonals on $[0,1]^2$, a simple power iteration estimates $\kappa(B) = 14.65$ and $\kappa(B + AB^{-1}A) = 204689$ with $V = 0$. On a $33 \times 33$ grid the corresponding condition numbers were 14.62 and 52686, with 204689/52686 $\approx 3.89$.)

The error bound for CG applied to $Ay = (B + AB^{-1}A)y = z$ states that $y_k$, the approximate solution after $k$ steps of CG, satisfies
\[ \|y - y_k\|_A \leq \left( \frac{1 - 1/\kappa(A)^{1/2}}{1 + 1/\kappa(A)^{1/2}} \right)^k \|y - y_0\|_A \approx \left( \frac{1 - Ch}{1 + Ch} \right)^k \|y - y_0\|_A \]
for some $C$.

For each time-step we'd like the error in solving the linear system to be $O(\Delta t^2) = O(h^3)$ (so after $T/\Delta t$ time steps the error adds up to less than $O(\Delta t^2)$, assuming everything is stable), so we'd like
\[ k \log \left( \frac{1 - Ch}{1 + Ch} \right) \leq 3 \log h, \]
or, using $\log(1 - Ch) \approx -Ch$ for $h$ small enough,
\[ k \geq - \frac{C}{h} \log h. \]

Since $N$, the number of unknowns, is $O(h^{-2})$ in two dimensions, we'll need $k \geq C \sqrt{N} \log N$ iterations of CG to solve $(B + AB^{-1}A)y = z$, for a total operation count per time step of $O(N^{3/2} \log N)$.

Since there will be $O(\Delta t^{-1}) = O(h^{-1})$ time steps, the total operation count will be $O(N^2 \log N)$. On a $K \times K$ grid, $N \approx K^2$, so the total operation count will be $O(K^4 \log K)$. Ignoring the logarithmic term, this will be on the order of $10^8$ operations when $K = 100$ and $10^{12}$ operations when $K = 1,000$.

Later we'll think about how to apply MG to this problem.