Section 8.2: Gauss-Seidel. There is a second method based on the same expression for $x_i$ in terms of the other $x_j$s, that is

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{i \neq j} a_{ij} x_j \right).$$

This time, instead of using all the previous values $x_j^{(k)}$ to compute $x_i^{(k)}$, we will always use the latest computed value for $x_j$.

That means the recurrence this time takes the form

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right).$$

Indeed, as we advance $i$ from 1 through $n$, $x_j^{(k+1)}$ becomes the newest approximate value for $x_j$.

It makes sense then to write $A = D + L + U$, where $L$ and $U$ stand for the strictly upper, respectively, strictly lower triangular parts of $A$. With this, we can rewrite the recurrence as

$$x^{(k+1)} = D^{-1} \left( b - L x^{(k+1)} - U x^{(k)} \right),$$

and we can rewrite this as

$$D x^{(k+1)} + L x^{(k+1)} = b - U x^{(k)},$$

or equivalently

$$(A - U) x^{(k+1)} = b - U x^{(k)}.$$

We can rewrite this as

$$x^{(k+1)} = (A - U)^{-1} \left( b - U x^{(k)} \right),$$

and since $b - U x^{(k)} = b - A x^{(k)} + A x^{(k)} - U x^{(k)}$, it follows that

$$x^{(k+1)} = (A - U)^{-1} \left( r^{(k)} + (A - U) x^{(k)} \right),$$

which is to say,

$$x^{(k+1)} = x^{(k)} + (A - U)^{-1} r^{(k)}.$$

This should remind you of the recurrence for Jacobi (see above), except now we have $(A - U)$ instead of $D$. But this is a simple recurrence, and it is simple to implement.

Important: $(A - U)$ is the lower triangular part of $A$.

How do we know if we converge? We will do the same kind of analysis as we did before, for Jacobi: look at the error vector. Let $\tilde{x}$ be the true solution of $A x = b$, and let $e^{(k)} = \tilde{x} - x^{(k)}$. As
before, we will get that

\[
e^{(k+1)} = \tilde{x} - x^{(k+1)} \\
= \tilde{x} - x^{(k)} - (A - U)^{-1} r^{(k)} \\
= e^{(k)} - (A - U)^{-1} (b - A v^{(k)}) \\
= e^{(k)} - (A - U)^{-1} (A \tilde{x} - A v^{(k)}) \\
= e^{(k)} - (A - U)^{-1} A e^{(k)} \\
= (I - (A - U)^{-1} A) e^{(k)}.
\]

So, just like before, this becomes a power method recurrence for the matrix \((I - (A - U)^{-1} A)\), and once again, since the vectors are unnormalized, \(e^{(k)}\) will converge to 0 if and only if the largest modulus eigenvalue of \(I - (A - U)^{-1} A\) is smaller than 1 in absolute value.

**Flop count.** There are many other methods, and the book presents some others; all are based on some maximum eigenvalue modulus condition like the ones for Jacobi and Gauss-Seidel. Some are easy to parallelize, some are less so, but all are \(O(n^2)\). The reason is that the single update (or way to compute \(x^{(k+1)}\) from \(x^{(k)}\)) is always \(O(n^2)\).

This can be seen as follows: on the one hand \(y = (A - U)^{-1} r^{(k)}\) can be obtained as the solution to the system \((A - U)y = r^{(k)}\) through forward substitution, since \(A - U\) is lower triangular. That flop count is \(O(n^2)\). On the other hand, \(r^{(k)} = b - Ax^{(k)}\) is obtained through a matrix-vector multiplication, followed by vector subtraction. That is also \(O(n^2)\). So the total per iteration is \(O(n^2)\).

The error will always go down like \(|\lambda_1|^k\), where \(|\lambda_1|\) is the largest eigenvalue in modulus of either \(I - D^{-1} A\) in the case of Jacobi, and \(I - (A - U)^{-1} A\) in the case of Gauss-Seidel. Hence, to get the error smaller than \(\epsilon\), we will need to take roughly

\[
\log_{\lambda_1} \epsilon = \ln \epsilon / \ln \lambda_1 = \ln \epsilon^{-1} / \ln \lambda_1^{-1}
\]

steps. As \(\lambda_1\) gets closer to 0, \(1/\lambda_1\) increases, so the number of steps to get to some preset tolerance \(\epsilon\) decreases.

**Total flop count:** \(O(n^2 \ln \epsilon / \ln \lambda_1)\).

Note that the smaller the largest eigenvalue of the matrix in question (which is \((I - D^{-1} A)\) for Jacobi, and \((I - (A - U)^{-1} A)\) for Gauss-Seidel), the faster the convergence of the method on \(A\).