

C++ code 2.18.7: EIGEN-based implementation of GaussSeidel_iteration()

```
2 bool GaussSeidel_iteration(const CRSMatrix &A, const Eigen::VectorXd &b,
3                             Eigen::VectorXd &x, double atol = 1.0E-8,
4                             double rtol = 1.0E-6, unsigned int maxit = 100) {
5     assert(A.n == A.m && "Matrix must be square");
6     assert(A.n == b.size() && "Vector b length mismatch");
7     assert(A.n == x.size() && "Vector x length mismatch");
8
9     // Main loop for Gauss-Seidel iteration
10    for (unsigned int k = 0; k < maxit; ++k) {
11        // TODO: (2-18.d) Implement the Gauss-Seidel iteration with a
12        // correction-based termination criterion.
13        // START
14        double x_normsq = 0.0;
15        double delta_x_normsq = 0.0;
16        for (unsigned int i = 0; i < A.n; ++i) {
17            double Aii = 0.0;
18            double s = b[i];
19            for (unsigned int l = A.row_ptr[i]; l < A.row_ptr[i + 1]; ++l) {
20                const unsigned int j = A.col_ind[l];
21                if (j != i) {
22                    s -= A.val[l] * x[j];
23                } else {
24                    // Fetch diagonal entry of A
25                    Aii = A.val[l];
26                }
27            }
28            if (Aii != 0.0) {
29                // Update iterate and compute Eucidean norm of current iterate
30                // and of correction
31                const double tmp = x[i];
32                x[i] = s / Aii;
33                delta_x_normsq += (x[i] - tmp) * (x[i] - tmp);
34                x_normsq += x[i] * x[i];
35            } else
36                return false;
37        }
38        // Correction based termination cirterion
39        if ((delta_x_normsq < atol * atol) ||
40            (delta_x_normsq < rtol * rtol * x_normsq)) {
41            return true;
42        }
43        // END
44    }
45    // No termination
46    return false;
47 }
```

Get it on 🐱 GitLab ([gaussseidelcrs.hpp](#)).