

C++ code 2.18.3: EIGEN-based implementation of GaussSeidelstep_crs()

```
2 bool GaussSeidelstep_crs(const CRSMatrix &A, const Eigen::VectorXd &b,
3                          Eigen::VectorXd &x) {
4     assert(A.n == A.m && "Matrix must be square");
5     assert(A.n == b.size() && "Vector b length mismatch");
6     assert(A.n == x.size() && "Vector x length mismatch");
7
8     // TODO: (2-18.b) Implement a single step of the Gauss-Seidel
9     iteration with
10    // the system matrix in CRS format.
11    // START
12
13    // Outer loop over rows of the matrix
14    for (unsigned int i = 0; i < A.n; ++i) {
15        double Aii = 0.0;
16        double s = b[i];
17        // Inner summation loop over non-zero entries of i-th row.
18        // Skip diagonal entry in the summation and store it separately.
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            x[i] = s / Aii;
30        else
31            return false;
32    }
33    // END
34    return true;
}
```

Get it on  GitLab (gaussseidelcrs.hpp).