

C++ code 3.4.7: Computing end encoding a Givens rotation

```
2  std::tuple<double, double, double> compGivensRotation(Eigen::Vector2d a) {
3      double rho, gamma, sigma;
4      // TODO (3-4.b): Compute the parameters of a rotation that when
5      // applied to a results in the vector  $(\|a\|, 0)^T$  or  $(-\|a\|, 0)^T$ .
6      // Hint: You can use the function sign() above when determining rho
7      // from gamma and sigma.
8      // START
9      if (a(1) != 0.0) {
10         double t;
11         // Separate cases to avoid cancellation/overflow
12         if (std::abs(a(1)) > std::abs(a(0))) {
13             t = -a(0) / a(1);
14             sigma = 1.0 / std::sqrt(1.0 + t * t);
15             gamma = sigma * t;
16         } else { // Then abs(a(0)) >= abs(a(1)) > 0
17             t = -a(1) / a(0); // a(0) != 0
18             gamma = 1.0 / std::sqrt(1.0 + t * t);
19             sigma = gamma * t;
20         }
21     } else {
22         // No rotation.
23         gamma = 1.0;
24         sigma = 0.0;
25     }
26     // Determine rho
27     if (gamma == 0.0)
28         rho = 1.0;
29     else if (std::abs(gamma) > std::abs(sigma))
30         rho = 0.5 * sign(gamma) * sigma;
31     else
32         rho = 2.0 * sign(sigma) / gamma;
33     // END
34     return std::make_tuple(rho, gamma, sigma);
35 }
```

Get it on  GitLab (tridiagqr.hpp).