

C++11 code 2.13.6: Sub-problem (2-13.c): Implementation of `Eval()` member function of `FESourceElemVecProvider` for quadrilaterals → [GitLab](#)

```
2 // I. OBTAIN COORDINATES OF THE MIDPOINTS
3 // I.i Hard-code the midpoints of the edges on the reference quadrilateral
4 Eigen::Matrix<double, 2, 4> midpoints_ref(2, 4);
5 midpoints_ref << 0.5, 1, 0.5, 0, 0, 0.5, 1, 0.5;
6 // I.ii Obtain the midpoints of the parametrized quadrilateral
7 auto midpoints_param = cell_geometry->Global(midpoints_ref);
8
9 // II. COMPUTE LOCAL INTEGRATION DATA
10 // II.i Compute the integration element
11 //      integration_element(x) = sqrt(det(J^T*J))
12 // where J is the Jacobian of the parametrization map
13 const Eigen::VectorXd integration_element(
14     cell_geometry->IntegrationElement(midpoints_ref));
15 // II.ii Evaluate w(x) at the midpoints
16 Eigen::VectorXd w(4);
17 for (int i = 0; i < 4; i++) {
18     w(i) = coeff_expansion_(cell_global_idx[i]);
19 }
20
21 // III. PERFORM NUMERICAL QUADRATURE
22 element_vector = Eigen::VectorXd::Zero(4);
23 for (int i = 0; i < 4; i++) { // for each local degree of freedom
24     element_vector(i) +=
25         integration_element(i) *
26         (0.5 / (1. + std::pow(0.5 * w(i) + 0.5 * w((i + 1) % 4), 2)));
27     element_vector((i + 1) % 4) +=
28         integration_element(i) *
29         (0.5 / (1. + std::pow(0.5 * w(i) + 0.5 * w((i + 1) % 4), 2)));
30 }
31 element_vector *= 1. / 4.;
32 break;
```