C++11 code 3.3.2: Sub-problem (3-3.d): Implementation of `GlobalInverseTria()`
➙ GitLab

```
Eigen::Vector2d GlobalInverseTria(Eigen::Matrix<double, 2, 3> mycorners,
                                  Eigen::Vector2d x) {
  Eigen::Vector2d x_hat;

  // The unique affine mapping of the unit triangle to a general
     triangle
  // is given by the formula x = A_K * x_hat + s,
  // Use the vertex coordinates to calculate matrix A and translation
     vector s
  Eigen::Matrix2d A(2, 2);
  A.col(0) = mycorners.col(1) - mycorners.col(0);
  A.col(1) = mycorners.col(2) - mycorners.col(0);
  // The inverse mapping:  x_hat = A_K^{-1} * (x - a_0)
  x_hat = A.partialPivLu().solve(x - mycorners.col(0));
  return x_hat;
}
```