

```

template <int dim>
unsigned int PolynomialSpace<dim>::
compute_n_pols (const unsigned int n) {
    unsigned int n_pols = n;
    for (unsigned int i=1; i<dim; ++i) {
        n_pols *= (n+i);
        n_pols /= (i+1);
    }
    return n_pols;
}

```

- - - - - After template specialization and applying optimizations: - - - - -

```

unsigned int PolynomialSpace<1>::
compute_n_pols(const unsigned int n) {
    return n;
}

```

```

template <int dim> inline
unsigned int TensorProductPolynomials<dim>::
x_to_the_dim (const unsigned int x) {
    unsigned int y = 1;
    for (unsigned int d=0; d<dim; ++d) {
        y *= x;
    }
    return y;
}

```

```

unsigned int TensorProductPolynomials<1>::
x_to_the_dim(const unsigned int x) {
    return x;
}

```