```
Clear[jacobiMatrix]
jacobiMatrix[function_] :=
 With[
  {var = Variables[function]},
  Table[D[func, #] & /@ var, {func, function}]
 ]

Clear[hesseMatrix]
hesseMatrix[function_] :=
 With[
  {var = Variables[function]},
  Table[D[func, firstVar, #] & /@ var,
   {firstVar, var}, {func, function}]
 ]
```

```
Clear[halley, jacobiN, hesseN, fN, t, r]
halley[function_, initial_, precision_] :=
 Module[
  {
   roots = initial,
   jacobiN, hesseN, fN,
   var = Variables[function],
   replaceRule,
   jacobi = jacobiMatrix[function],
   hesse = hesseMatrix[function],
   t, r
  },
  replaceRule = MapThread[#1 → #2 &, {var, roots}];
  While[
   Or @@ (Abs[# /. replaceRule] > precision & /@
       function),
   jacobiN = jacobi /. replaceRule;
   hesseN = hesse /. replaceRule;
   fN = function /. replaceRule;
   t = Last /@ Last[Solve[jacobiN.var == -fN, var]];
   r =
    Last /@ Last[Solve[jacobiN.var == hesseN.t.t,
       var]];
   roots = roots + t^2/(t + 0.5 r);
   replaceRule = MapThread[#1 → #2 &, {var, roots}];
  ];
  roots
 ]
```

```
f = {x1^2 - x2^2 - 1, x1 x2^3 - x2 - 1}
```

$\{-1 + x1^2 - x2^2,\ -1 - x2 + x1\ x2^3\}$

```
halley[f, {0.5, 0.5}, 0.001]
```

{-1.19723, -0.658403}

```
NSolve[x1^2 - x2^2 - 1 == 0 && x1 x2^3 - x2 - 1 == 0, {x1, x2}][[
  2]]
```

{x1 → -1.19726, x2 → -0.658357}