Newton's method

Assume there is a minimum in the interval $[x_L, x_U]$, and $x_0 \in [x_L, x_U]$.

To seek the root of $f'(x) = 0$, the Newton's fixed-point iteration becomes,

$$x_{i+1} = x_i - \frac{f'(x)}{f''(x)}$$

Iteration stops when $|x_{i+1} - x_i|$ or $|f(x_{i+1}) - f(x_i)|$ is very small.

Example

Find the minimum of $f(x) = \frac{x^2}{10} - 2\sin x$ over the interval of $[0, 4]$.

Use the "distance" based stopped criterion. For example, $|x_3 - x_1| < 10^{-6}$ for quadratic interpolation.

Solution

$$f'(x) = \frac{x}{5} - 2\cos(x)$$
$$f''(x) = \frac{1}{5} + 2\sin(x)$$

Golden-section

# of iterations	x_1^* or x_2^*	$\|x_2 - x_1\|$
30	1.42755134	$8.21214(10^{-7})$

*whichever gives lower function value.

Quadratic interpolation with $x_1 = 1$

# of iterations	x_3	$\|x_3 - x_1\|$
11	1.42755207	$2.96747(10^{-7})$

Newton's method with $x_0 = 1$

# of iterations	x_{i+1}	$\|x_{i+1} - x_i\|$
4	1.42755178	$4.78198(10^{-10})$

The question remains how to determine the interval $[x_L, x_U]$.

The following bracketing scheme may be suggested, which is part of the Davies-Swann-Campey algorithm.

Step 1: Select an x_1 that is close to the x^*being sought. Also assign a small value close to Δ.

Step 2: Let $x_0 = x_1 - \Delta$ and $x_2 = x_1 + \Delta$. Evaluate $f_0 = f(x_0),\ f_1 = f(x_1),\ f_2 = f(x_2)$.

There are three cases.

2a. If $f_0 \geq f_1$ and $f_1 \leq f_2$, then $[x_0, x_2]$ is the interval. Together with x_1, the quadratic interpolation can be started. For golden-section search, $[x_0, x_2]$ is the $[x_L, x_U]$;

2b. If $f_0 > f_1$ and $f_1 > f_2$, the following is determined:

$$x_3 = x_2 + 2\Delta,\ f_3 = f(x_3)$$
$$x_4 = x_3 + 4\Delta,\ f_4 = f(x_4)$$
$$x_5 = x_4 + 8\Delta,\ f_5 = f(x_5)$$
$$\cdots$$

Until the current f_i is greater than the previous f_{i-1}. Then $[x_0, x_i]$ is the interbal, and x_{i-1} is x_1, if needed.

2c. If $f_0 < f_1$ and $f_1 < f_2$, $x_2 = x_0 - \Delta$, $f_2 = f(x_2)$. The following is determined:

$$x_3 = x_2 - 2\Delta, f_3 = f(x_3)$$
$$x_4 = x_3 - 4\Delta, f_4 = f(x_4)$$
$$x_5 = x_4 - 8\Delta, f_5 = f(x_5)$$
$$\cdots$$

Until f_i is greater than f_{i-1}. Then $[x_i, x_0]$ is the interbal, and $x_{i-1}$is x_1 is needed.

Example

Find the minimum of $f(x) = \frac{x^2}{10} - 2\sin x$ over the interval of $[0, 4]$.

Use the "distance" based stopped criterion. For example, $|x_3 - x_1| < 10^{-6}$ for quadratic interpolation.

Golden-section

# of iterations	x_1^* or x_2^*	$\lvert x_2 - x_1\rvert$
30	1.42755134	$8.21214(10^{-7})$

* whichever gives lower function value

# of iterations	Interval	x_1^* or x_2^*	$\lvert x_2 - x_1\rvert$
29	$[0, 2.8]$	1.42755300	$9.30125(10^{-7})$

* whichever gives lower function value

Quadratic interpolation with $x_1 = 1$

# of iterations	x_3	$\lvert x_3 - x_1\rvert$
11	1.42755207	$2.96747(10^{-7})$

# of iterations	Interval	x_1	x_3	$\lvert x_3 - x_1\rvert$
6	$[0, 2.8]$	1.2	1.42755196	$2.09784(10^{-7})$

Newton's method with $x_0 = 1$

# of iterations	x_{i+1}	$\lvert x_{i+1} - x_i\rvert$
4	1.42755178	$4.78198(10^{-10})$

With $x_0 = 1.2$

# of iterations	x_{i+1}	$\lvert x_{i+1} - x_i\rvert$
4	1.42755178	$7.36522(10^{-13})$

Summary of one-dimensional optimization:

Golden-search, or quadratic interpolation, together within the David-Swann-Campey bracketing method, are within the category of "search method" as no derivative is required.

On the other hand, the Newton's method belongs in the category of gradient method.

They form the basis of solving multi-dimensional unconstrained optimization problems.

Part 4: Optimization (III)

Multi-dimensional unconstrained optimization means, in mathematical terms,

$$\min_{x} f(x) \quad ; \quad for\ x \in R^n$$

Where $f(\boldsymbol{x})$ is a continuous real-values function.

Some math first.

1. Local minimum and local maximum

If $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ for all $\boldsymbol{x}$ near $\boldsymbol{x}^*$**,** $\boldsymbol{x}^*$ is the local minimum.

If $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ for all $\boldsymbol{x}$ near $\boldsymbol{x}^*$**,** $\boldsymbol{x}^*$ is the local maximum.

2. The gradient of $f(\boldsymbol{x})$ is:

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_n}\right)^T$$

3. Critical or stationary point:

If the gradient vector is zero at $\boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is a critical or stationary point.

4. First derivative test:

A local minimum or maximum must be a critical point of $f(\boldsymbol{x})$.
In other words, if $f(\boldsymbol{x})$ has a local minimum or maximum at $\boldsymbol{x}^*$, the the first order derivatives of $f(\boldsymbol{x})$ exist at $\boldsymbol{x}^*$, then:

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}^*} = 0 \quad ; \quad i = 1, 2, 3, \dots$$

5. The Hessian (matrix) of $f(\boldsymbol{x})$ is:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Or the Hessian is the Jacobian matrix of the gradient.

- If $\frac{\partial^2 f}{\delta x_i \partial x_j}$ is continuous, then

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$$

- The Hessian determinant, $|H|$, means the determinant of the Hessian matrix H. It is sometimes called the discriminant of H.

6. Second derivative test

If $\boldsymbol{x}^*$ is a critical point of $f(\boldsymbol{x})$, and all the second order partial derivatives of $f(\boldsymbol{x})$ are continuous, then:

- $\boldsymbol{x}^*$ is a local minimum if H (evaluated at $\boldsymbol{x}^*$) is positive definite (that is, all eigenvalues of H are positive)
- $\boldsymbol{x}^*$ is a local maximum if H is negative definite (all eigenvalues of H are negative)
- $\boldsymbol{x}^*$ is a saddle point if H has both positive and negative eigenvalues.
- However, the test is inconclusive in cases not listed above.

For two-dimensional problems:

- $\boldsymbol{x}^*$ is a local minimum if $|H| > 0$ and $\left.\frac{\partial^2 f(x)}{\partial x_1^2}\right|_{\boldsymbol{x}^*} > 0$;
- $\boldsymbol{x}^*$ is a local maximum if $|H| > 0$ and $\left.\frac{\partial^2 f(x)}{\partial x_1^2}\right|_{\boldsymbol{x}^*} < 0$;
- $\boldsymbol{x}^*$ is a saddle point if $|H| < 0$.
- However, it is inconclusive is $|H| = 0$.

7. The Taylor expansion of $f(\boldsymbol{x})$, at $\boldsymbol{x}^*$ and up to the second order, is,

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + (\nabla f)^T(\boldsymbol{x} - \boldsymbol{x}^*) + \left(\frac{1}{2}\right)(\boldsymbol{x} - \boldsymbol{x}^*)^T H(\boldsymbol{x} - \boldsymbol{x}^*) + \cdots$$

Where the gradient ∇f and Hessian H are evaluated at $\boldsymbol{x}^*$.

Examples:

Note: in the following, $\boldsymbol{x} = (x, y)^T$.

E1: Show that $f(x, y) = x^2 - y^2$ has a saddle point at $(0, 0)^T$,

$$H = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \quad ; \quad |H| = -4$$

E2: Find the local optimum of:

$$f(x, y) = x^2 + 2y^2 - 2xy - 2x$$

$$H = \begin{bmatrix} -2 & 2 \\ 2 & -4 \end{bmatrix} \quad ; \quad |H| = 4$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial x_1^2} = -2 < 0$$

Categories of methods include,

- Line search methods
- Trust-region methods

Trust-region methods:

- The trust region is the neighborhood near $\boldsymbol{x}^*$
- $f(\boldsymbol{x})$ is represented by a high-dimensional parabolic "surface"
- $\boldsymbol{x}^*$ is the $\boldsymbol{x}$ that minimizes the high-dimensional parabolic "surface"

Line search methods:

A multi-dimensional problem is transformed into a sequence of one-dimensional problems.

- Univariate searches; and
- Steepest-descent methods

Part 4: Optimization (IV)

Line Search Methods

The key is to transform a multi-dimensional problem into a sequence of one-dimensional problems.

For one dimensional unconstrained optimization, we perform bracketing, then golden-search section or quadratic interpolation or Newton's method.

But all is done along one single search ***direction*** or the x −axis.

Line search is about searching along a direction (i.e., a line) that is hopefully effective.

<u>Univariate searches</u>
The search directions are, x_1, then x_2, …, and finally x_n

The main steps are:

<u>Step 1:</u> Initial guess x_0 and Δ

<u>Step 2:</u> Perform the following logical loop:

for $k = 1{:}n$

$1D$ *unconstrained optimization along* x_k

end

This step ends with an $\boldsymbol{x}^*$

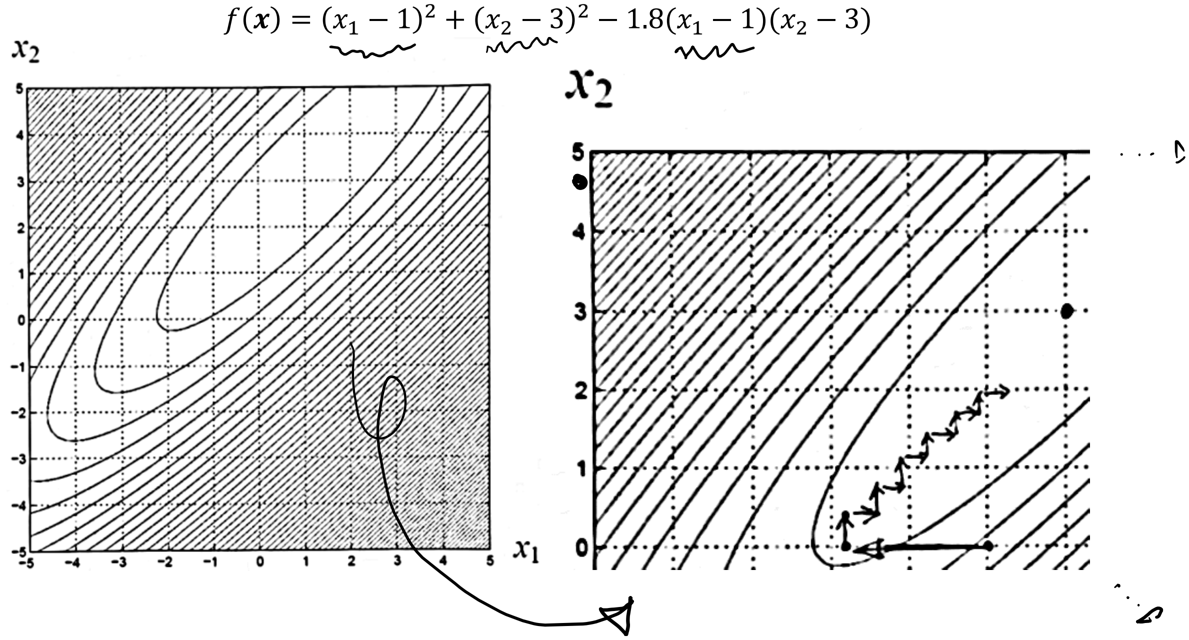
<u>Step 3:</u> Check if $||\boldsymbol{x}^* - \boldsymbol{x_0}||$ meets the stopping criterion.

If yes, $\boldsymbol{x}^*$ and $f(\boldsymbol{x}^*)$ are the solution sought.

Otherwise, $\boldsymbol{x_0} \leftarrow \boldsymbol{x}^*$, and go back to Step 2.

Graphically, consider a 2D problem:

$$f(\boldsymbol{x}) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$



Example:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Initial guess $\boldsymbol{x_0} = [0.75, -1.25]^T$ and $\Delta = 0.1$

<u>Golden-section search for 1D</u>

Along x_1,

$$\boldsymbol{x_L} = [-5.45, -1.25]^T,$$
$$\boldsymbol{x_U} = [0.75, -1.25]^T$$

After 30 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_L} = [-2.825, -1.25]^T,$$
$$\boldsymbol{x_U} = [-2.825, 0.15]^T$$

After 26 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 61 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999983, 2.999982]^T$$
$$f(\boldsymbol{x}^*) = 5.752007^{-11}$$

<u>Quadratic interpolation for 1D</u>

Along x_1,

$$\boldsymbol{x_L} = [-5.45, -1.25]^T,$$
$$\boldsymbol{x_U} = [0.75, -1.25]^T$$
$$\boldsymbol{x_1} = [-2.25, -1.25]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_L} = [-2.825, -1.25]^T,$$
$$\boldsymbol{x_U} = [-2.825, 0.15]^T$$
$$\boldsymbol{x_1} = [-2.825, -0.65]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 67 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999996, 2.999997]^T$$
$$f(\boldsymbol{x}^*) = 2.862871^{-12}$$

<u>Newton's method for 1D</u>

Along x_1,

$$\boldsymbol{x_1} = [-2.25, -1.25]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_1} = [-2.825, -0.65]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 66 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999996, 2.999997]^T$$
$$f(\boldsymbol{x}^*) = 3.524268^{-12}$$

<u>Comparison of elapsed CPU times:</u>
Golden-section search: 0.140625 sec.
Quadratic interpolation: 0.125000 sec.
Newton's method: 0.109375 sec.

Other search direction? "Good" directions especially?

There are a few options here. Conjugate direction is one; The steepest-descent is another.

<u>Steepest-descent Methods</u>
What is the steepest direction? The concept of directional derivative is the starting point.

If ∇f is the gradient of $f(\boldsymbol{x})$ at any $\boldsymbol{x}$, the direction is $\boldsymbol{n}$, a unit vector $\left(for\ example, n = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0\right)^T\right)$, then the directional derivative along $\boldsymbol{n}$ is,

$$D_n(\boldsymbol{x}) = (\boldsymbol{\nabla} f)^T \boldsymbol{n}$$

Directional derivative is a scalar function.

Treating $\boldsymbol{n}$ as the independent variables, seeking the optimum $D_n(\boldsymbol{x})$ will result in the steepest direction. It has been proven that the steepest direction is the gradient itself. In other words, the optimum of $D_n(\boldsymbol{x})$ is obtained when:

$$n = \nabla f$$

The three main steps of the steepest-descent method are,

<u>Step 1</u>: Initial guess $\mathbf{x_o}$ and Δ

<u>Step 2:</u> evaluate ∇f at x_0;

1D unconstrained optimization along ∇f;
obtain a $\boldsymbol{x}^*$

Step 3: check if $||\boldsymbol{x}^* - \boldsymbol{x_0}||$ meets the stopping criterion

If yes, x^* and $f(x^*)$ are the solution sought.

Otherwise, $x_0 \leftarrow x^*$, and go back to Step 2.

Some programming notes:

Bracketing:

- Is done along ∇f

Applying Golden-section search along ∇f:

- ℓ_0 means the second norm;
- The gradient should be normalized to a unit vector;
- The scalar $x's$ are now vectors.

Applying quadratic interpolation along ∇f:

- The gradient should be normalized to a unit vector;
- For one dimensional problems,

$$x_3 = \frac{1}{2} \frac{f_0(x_1^2 - x_2^2) + f_1(x_2^2 - x_0^2) + f_2(x_0^2 - x_1^2)}{f_0(x_1 - x_2) + f_1(x_2 - x_0) + f_2(x_0 - x_1)}$$

Where $f_i = f(x_i)$

Now, $f_i = f(\boldsymbol{x_i})$, x_j^2 is replaced by the dot product of $\mathbf{x_j}$, or $(\mathbf{x_j})^{\boldsymbol{T}}\mathbf{x_j}$ and $x_i - x_j$ is replaced by the second norm of $\mathbf{x_i} - \mathbf{x_j}$.

- $\mathbf{x_3}$ is x_3 times the normalized gradient

Applying Newton's method along ∇f:

- The iteration scheme for one-dimensional problems is,

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

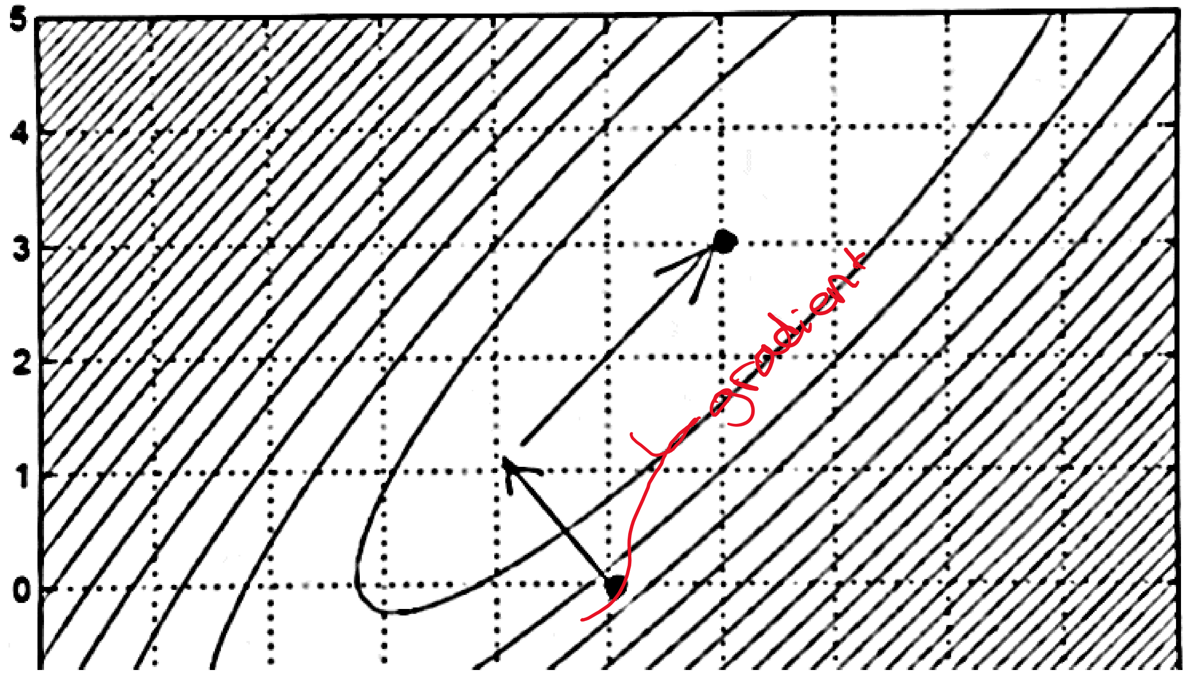
- Extending it to multi-dimension,

$$\mathbf{x_{i+1}} = \mathbf{x_i} - H^{-1}\nabla f$$

Where H and ∇f are evaluated at x_i.

Graphically, consider a 2D problem.

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$



Initial guess $\mathbf{x_0} = [0.75. -1.25]^T$ and $\Delta = 0.1$.

<u>Golden-section search</u>

17 rounds of gradient computation, the converged solution is:

$$\boldsymbol{x}^* = [0.999992, 2.999992]^T$$
$$f(\boldsymbol{x}^*) = 5.752007^{-11}$$
$$\text{cputime} = 0.046875\ sec.$$

<u>Newton's method</u>

1 round of gradient computation, the converged solution is:

$$\boldsymbol{x}^* = [1, 3]^T$$
$$f(\boldsymbol{x}^*) = 0$$
$$\text{cputime} = 0.031250\ sec.$$

Example: the Rosenbrock function (a.k.a. the banana function) is a "standard" test problem on the performance of any unconstrained optimization solver.

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$$

m is an integer. The dimension of the problem is $m + 1$.

Set $m = 4$, initial guess of $x_0 = [0, 0, 0, 0, 0]^T$, and $\Delta = 0.1$.

Golden-section search:

4214 rounds of gradient computation, cputime = 0.516525 s

$$\boldsymbol{x}^* = \begin{pmatrix} 0.999665 \\ 0.999331 \\ 0.998657 \\ 0.997312 \\ 0.994615 \end{pmatrix}, \qquad f(\boldsymbol{x}^*) = 9.466281^{10^{-6}}$$

Newton's method:

2 rounds of gradient computation, cputime = 0.3125 s

$$\boldsymbol{x}^* = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad f(\boldsymbol{x}') = 0$$