Modern Methods of Optimization

Lecture 3: Unconstrained optimization in 1D

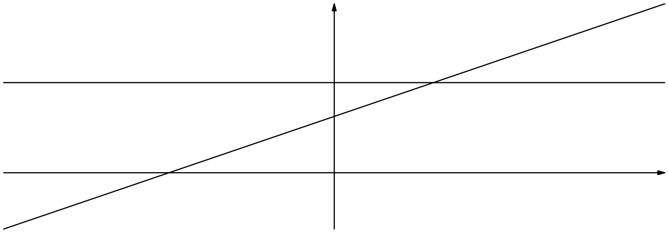
Faculty of Civil Engineering / CTU in Prague

Problem setting

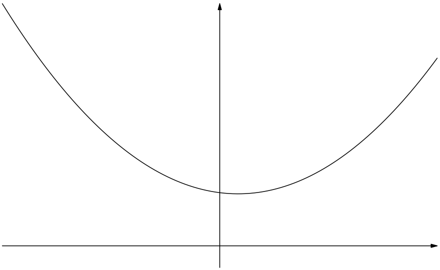
$$f(\mathbf{x}_{\text{opt}}) \overset{\mathscr{H}}{\preceq} f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathscr{O}$$

- Variable: $\mathbf{x} \equiv x$
- Set of admissible solutions $\mathscr{O} \equiv$
- Range of the objective function $\mathscr{H} \equiv$
- Partial ordering $\overset{\mathscr{H}}{\preceq} \equiv$
- Objective function $f : \quad \rightarrow$
  -
  -

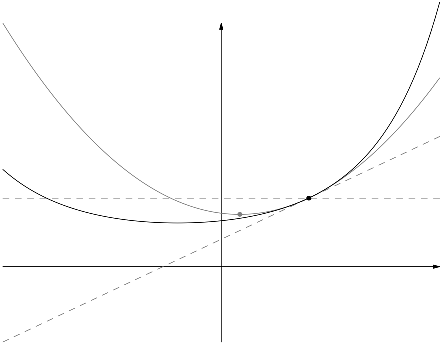
Which functions are easiest to optimize?



Which functions are easiest to optimize?

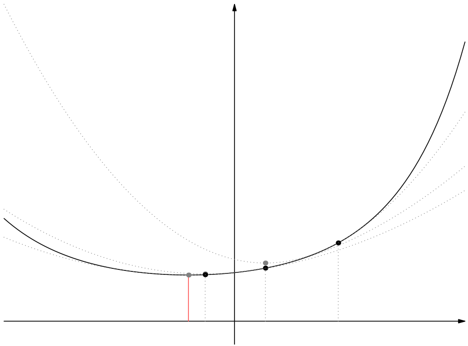


General f



- Surrogate function: Taylor expansion

Newton method



Algorithm

1. **Input**
   - $g : x \mapsto f'(x)$
   - $h : x \mapsto f''(x)$
   - initial guess x_0
   - tolerance ε
2. **Initiation:** $0 \leftarrow k$
3. $g_k \leftarrow g(x_k)$
4. if $|g_k| \leq \varepsilon$: $x_{\text{opt}} \leftarrow x_k$ and terminate
5. $h_k \leftarrow h(x_k)$
6. $d_k \leftarrow -h_k^{-1} g_k$
7. $x_{k+1} \leftarrow x_k + d_k$
8. $k \leftarrow k+1$
9. go to 3

Algorithm

- Example

$$f(x) = (x^2 - 1)^2 + x$$

$$f'(x) =$$

$$f''(x) =$$

- https://gitlab.com/jan.zeman4/132mmo > codes

Input (Line 1)

```
function newton1d
    function val = f(x)
        val = (x.^2-1).^2+x;
    end

    function val = g(x)
        val = 4*(x^3-x)+1;
    end

    function val = h(x)
        val = 4*(3*x^2-1);
    end

tolerance = 1e-6;
xk = -2;
```

Initiation (Line 2) and convergence test (lines 3 and 4)

```
k = 0;

clf;
ezplot(@(x)f(x), [-2.25,2.25]);
hold on;
```

```
gk = g(xk);

while(abs(gk) > tolerance)
   disp(['iteration = ' num2str(k) ' x = ' num2str(xk) ...
       ' gradient = ' num2str(gk) ]);
   plot(xk,f(xk),'or');
```

Newton step (lines 5–9)

```
    hk = h(xk);
    dk = -gk/hk;
    xk = xk+dk;
    gk = g(xk);
    k = k+1;
    pause
end
end
```

- Try

$$\varepsilon = 10^{-10}$$

$$x_0 = 2,\ 0.5,\ 1/\sqrt{3}$$

Newton step: Pros and cons

Pros	Cons