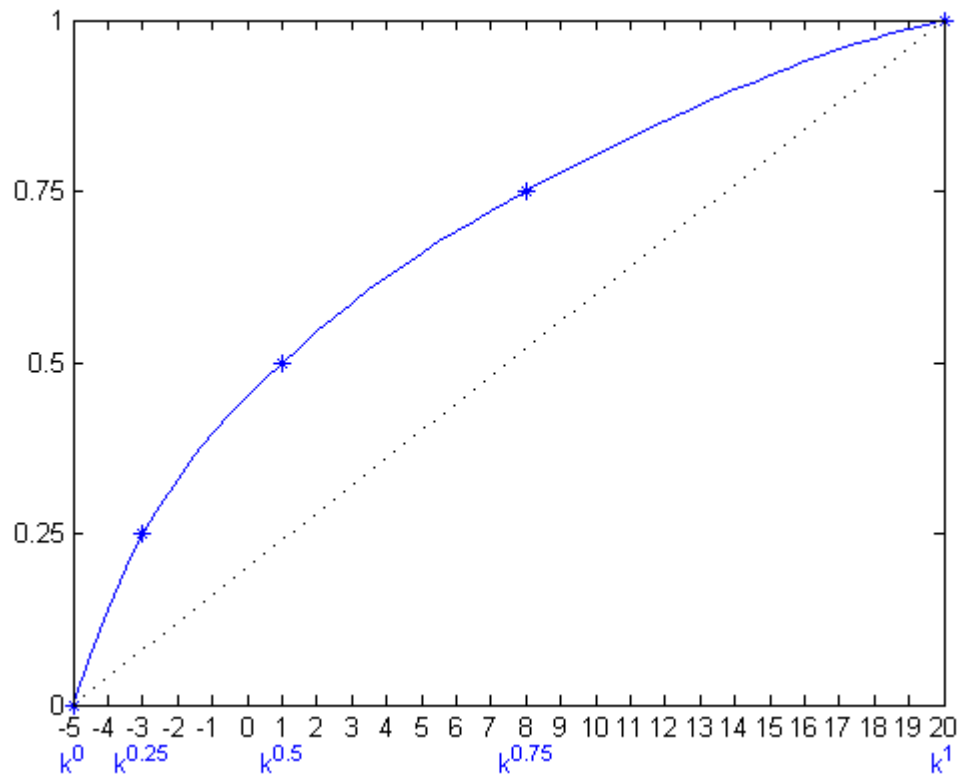


TeX pro začátečníky

Pavel Holeček

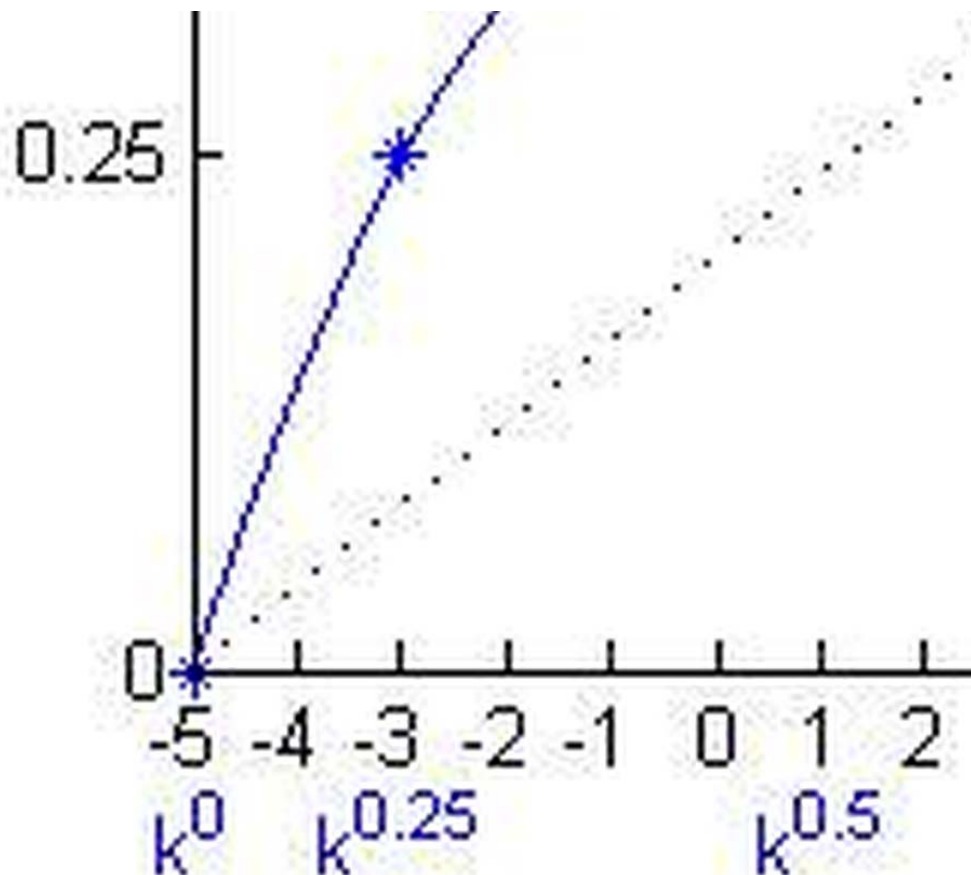
TeX – formáty obrázků

- ▶ Máte možnost vkládat obrázky ve formátu JPG, PNG, PDF a EPS.
- ▶ Rozdíl si ukážeme na následujícím grafu vytvořeném v Matlabu:



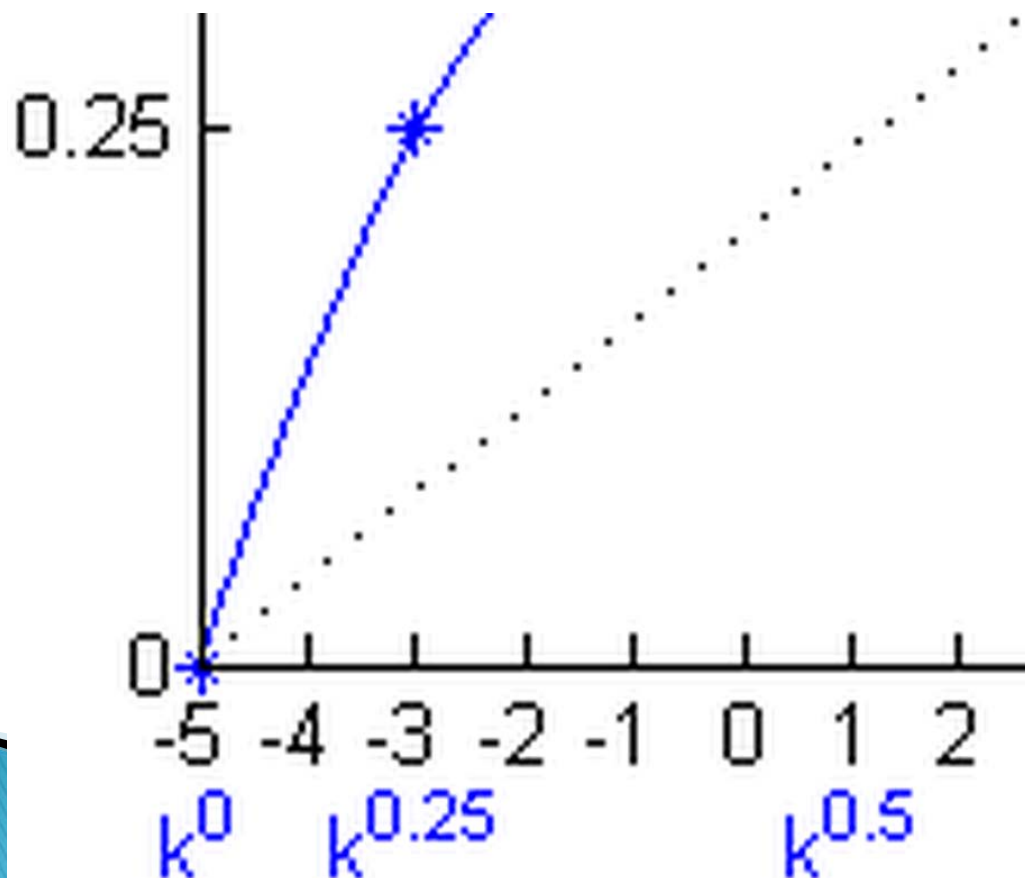
TeX – formáty obrázků

- ▶ **JPG:** formát je velice vhodný pro fotografie. Používá ale ztrátovou kompresi, takže je nevhodný na různé diagramy a grafy (vzniká zkreslení, tzv. artefakty).



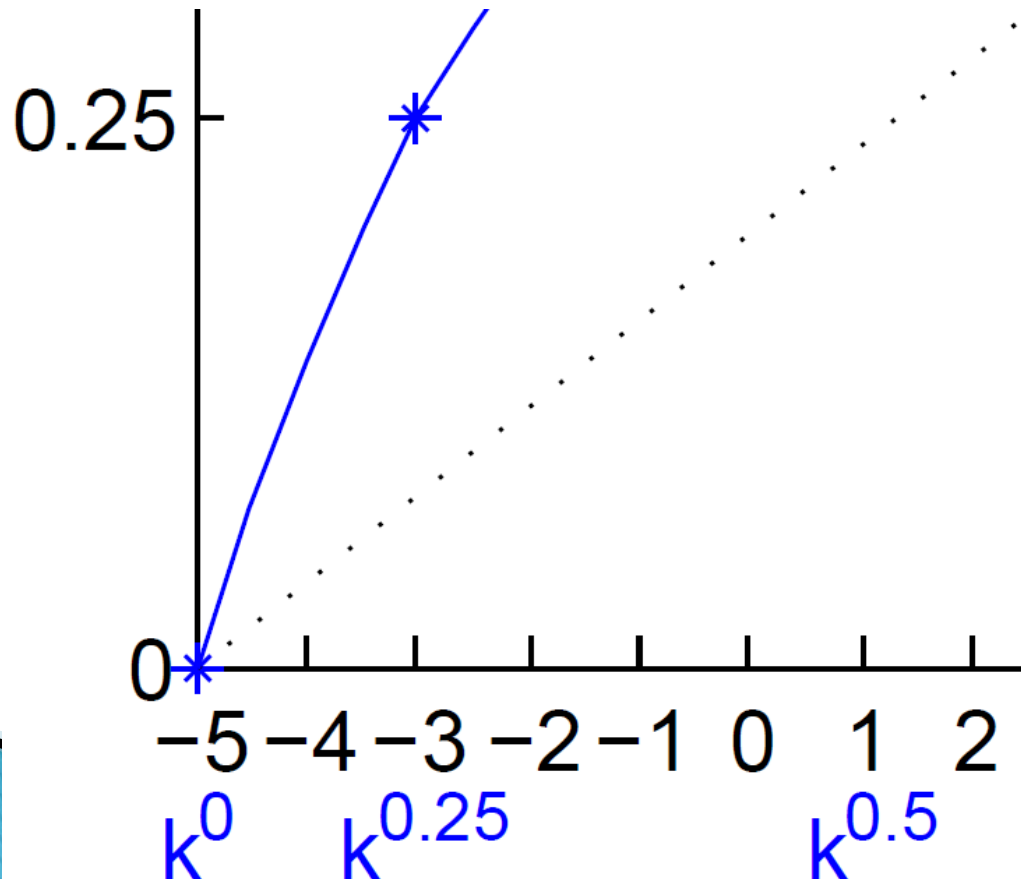
TeX – formáty obrázků

- ▶ **PNG:** formát je naopak velice vhodný pro diagramy a grafy. Stejně jako JPG jde ale o bitmapu, takže při velkém zvětšení jsou vidět jednotlivé obrázkové body jako čtverečky.



TeX – formáty obrázků

- ▶ **PDF, EPS:** Podporují vektorovou grafiku. Při zvětšení je čára rovná a neskládá se ze čtverců. Pokud to program umožňuje (například Matlab ano), použijte tyto formáty.



TeX – přímý vstup

- ▶ V prostředí přímého vstupu se ignorují veškerá typografická pravidla a pro sazbu se použije strojopisné písmo.
- ▶ Používají se v ukázkách kódu programovacího jazyka. Například, pokud budete ve vaší práci pracovat s programy jako je
 - Matlab
 - R
 - TeX
 - atd.

TeX – přímý vstup

- ▶ Pro kratší texty v jednom řádku slouží příkaz `verb`:
`\verb+\begin{document}+`
- ▶ Místo znaku `+` můžeme použít i jiný. Slouží pro ohraničení našeho textu. V tomto textu můžeme používat i speciální znaky `\`, `{` a `}`.

TeX – přímý vstup

- ▶ Pro delší texty slouží prostředí `verbatim`.
- ▶ Příklad vložení kódu v Matlabu (nemusíme používat `\\` pro přechod na nový řádek, můžeme používat speciální znaky přímo).

```
\begin{verbatim}
```

```
function R = NovakFuzzyRelaceProPravidlo( A, B )
```

```
    R = zeros(length(A), length(B));
```

```
    for i=1:length(A)
```

```
        for j=1:length(B)
```

```
            R(i,j) = min(1, 1 - A(i) + B(j));
```

```
        end
```

```
    end
```

```
end
```

```
\end{verbatim}
```


TeX – přímý vstup

- ▶ Pro delší texty můžeme použít prostředí `lstlisting` z balíčku `listings`.
- ▶ Bude se i zvýrazňovat syntaxe.

```
\usepackage{listings}
```

```
\begin{document}
```

```
\lstset{language=Matlab}
```

```
\begin{lstlisting}
```

... zde bude zdrojový kód

```
\end{lstlisting}
```

```
\end{document}
```

Musíme nastavit, jaký programovací jazyk používáme.

TeX – přímý vstup

- ▶ Pro delší texty můžeme použít prostředí `lstlisting` z balíčku `listings`.
- ▶ Bude se i zvýrazňovat syntaxe.

```
\usepackage{listings}
```

```
\begin{document}
```

```
\lstset{language=Matlab}
```

```
\begin{lstlisting}
```

```
... zde bude zdrojový kód
```

```
\end{lstlisting}
```

```
\end{document}
```

Musíme nastavit, jaký programovací jazyk používáme.

TeX – přímý vstup

- ▶ Pro delší texty můžeme použít prostředí [lstlisting](#) z balíčku [listings](#).
- ▶ Výsledek:

```
% Pokusna funkce
function R = NovakFuzzyRelaceProPravidlo( A, B )
    R = zeros(length(A), length(B));
    for i=1:length(A)
        for j=1:length(B)
            R(i,j) = min(1, 1 - A(i) + B(j));
        end
    end
end
```

TeX – přímý vstup

- ▶ Příkazem `lstset` můžeme nastavit nejen jazyk, ale i barvy (musíme zároveň použít balíček `color`):

- ▶ `\usepackage{color}`

- ▶ `\usepackage{listings}`

- ▶ `\lstset{`

- `language=matlab,`

- `commentstyle=\color{red},`

- `keywordstyle=\color{blue}`

Programovací jazyk,

Barva komentářů,

Barva klíčových slov

- ▶ `}`

Další nastavení najdete v dokumentaci k balíčku `listings`.

TeX – přímý vstup

- ▶ Výsledek po nastavení barev:

```
% Pokusna funkce
function R = NovakFuzzyRelaceProPravidlo( A, B )
    R = zeros( length(A), length(B) );
    for i=1:length(A)
        for j=1:length(B)
            R(i,j) = min(1, 1 - A(i) + B(j));
        end
    end
end
```

- ▶ **Nevýhoda balíčku listings:** nemůžete používat znaky s diakritikou (většinou to ale ani není potřeba).