Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 293–307
https://doi.org/10.22364/bjmc.2026.14.2.02

Generalized Model and Creation of Macros for Keywords, Text Visibility, and Language Choice

Halyna M. HUBAL

Lutsk National Technical University, 75 Lvivska str., Lutsk, Ukraine

h.myk1888@gmail.com

ORCID 0000-0002-8824-4565

Abstract. The article presents the methodology and generalized model of the creation of macros in the LaTeX system and considers the problems of the creation and use of macros that provide flexibility in text formatting, facilitate working with the LaTeX system, increase productivity of it, and are motivated by the needs of preparing scientific, technical, and educational documents. In the article, we created a new macro for keywords that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by an inner macro of the macro for keywords each time the inner macro is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords, then the updated keyword is printed by the inner macro of the macro for keywords each time the inner macro is called. A new macro that provides a way to make a given box of a text invisible or visible in the output PDF file was created in the article. We created a new macro that prints a text in a selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

Keywords: LaTeX, macro, generalized model, keyword, invisible text, choice of language.

1 Introduction

The LaTeX system is often used to create scientific, technical, and educational documents (Abraham and Luke, 2022; Lamport, 1994; Lode, 2019).

In the LaTeX system, we can create macros that automate complex tasks, which allows us to optimize the document creation process (Hubal, 2023; Mittelbach et al., 2004; Van Dongen, 2012). The importance of automated processing of document texts is shown in the article (Nazaruka, 2020). The study (Zuzevičiūtė et al., 2025) shows how emerging information technologies impact professional activities in educational and research processes, which confirms the relevance of creating macros in the LaTeX system for automating the processing of educational, scientific, and technical documents.

In this article, we created the following new macros:

1. a macro for keywords that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by the inner macro of the macro for keywords each time the inner macro is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords, then the updated keyword is printed by the inner macro of the macro for keywords each time the inner macro is called;
2. a macro that provides a way to make a given box of a text invisible or visible in the output PDF file;
3. a macro that prints a text in a selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

The macro from item 1 can be used in the LaTeX system to create scientific, technical, educational documents, and their templates in which we can include keywords that mark any text elements. These keywords may contain hints, notes, short explanations, references to bibliography, and data that need to be repeated.

The macro from item 2 can be used in the LaTeX system when reviewing scientific, technical, and educational documents that may include additional comments that are not displayed in the output PDF file. This macro can also be used in the LaTeX system when preparing educational documents that include invisible proofs of theorems, lemmas, explanations, solutions of problems, auxiliary and other text elements so that the student can solve the problems himself. This macro can also be used to create tests, which is especially important in online learning (Sarac and Durakovic, 2022) and in active learning (Romansky, 2023).

The macro from item 3 can be used in the LaTeX system to create scientific, technical, and educational documents in which we can choose the language of a text for the appropriate audience. This macro is especially useful in online learning for multilingual audiences. The importance of the ability to select the language of a text is confirmed by investigations (Šostaka and Borzovs, 2023; Šostaka, Borzovs et al., 2023) that show the importance of formation (from ICT terms formed in English) of secondary ICT terms in other languages, in particular in Latvian.

These new macros expand the functionality of the LaTeX system and increase the efficiency of preparing structured documents. They automate the creation of texts that are relevant and meaningful for scientists, professors, and engineers.

2 The methodology and generalized model of the creation of macros in the LaTeX system

Let us present the methodology of the creation of macros in the LaTeX system:

- analysis of the necessity for creating new macros. At the same time, consideration is given to new functions that should be implemented to improve the presentation of various documents.
- Structural design.
  I. Macro hierarchy:
     — basic macros are macros that typically implement a single function;

— combined macros are macros that combine basic macros into blocks;
— universal macros are macros that can adapt to different contexts.

II. Modularity.
Each macro or a group of macros is placed in its own package or a separate file.

III. Dependencies.
Macros should not depend on each other, except for explicit interfaces (such as arguments).

- Implementing macros.
- Testing macros.

This methodology of the creation of macros in the LaTeX system is the generalized model of the creation of the macros.

Three key components can be identified in this model:

1. packages;
2. the macros that implement specific tasks;
3. the arguments of macros.

Arguments provide adaptability to macros: a single macro can implement different actions depending on the arguments without changing its source code.

The generalized model systematizes the process of the creation of macros in the LaTeX system, ensures the flexibility, the reuse, and the scalability of macros in various documents.

In a LaTeX macro, the result is determined by the set of its arguments and the body of the macro, whereas in a Microsoft Word macro, the result also depends on the current state of a document and the execution environment. The tasks implemented by a single macro in the LaTeX system require Microsoft Word to write procedural subroutines with the direct control of the Document Object Model, which significantly complicates the source code.

3 Results and discussion

3.1 Creating the macro for keywords

Let us solve the following problem in the LaTeX system. We print a keyword in several places of Section 1 of the output PDF file (see Fig. 3). To do this, in a LaTeX document, we mark Section 1 with the keyword that is not printed to the output PDF file by calling the new macro `\keywdef` (see Fig. 1 and Fig. 2). Then we print this keyword in several places of Section 1 of the output PDF file by calling the inner macro `\keyw` of the macro `\keywdef` (see Fig. 2). We perform the same operations with another keyword for Section 2 by calling the same macros `\keywdef` and `\keyw`.

In order to solve this problem, in the LaTeX document, we create the new macro `\keywdef` with one argument `#1` that represents the keyword. In this macro, we define the empty macro `\keyw` and redefine the macro `\keyw` by using `\renewcommand` so that this macro takes one argument `#1`. The macro `\keyw` prints the passed argument

each time the macro is called. The source code of the created macro for keywords is given in Fig. 1.

```
\newcommand{\keywdef}[1]{%
  \def\keyw{}
  \renewcommand{\keyw}{#1}
}
```

Fig. 1. The created macro for keywords (the source code is available here)

In the body of the LaTeX document, we mark Section 1 with the keyword 'Q4-2024' that is not printed to the output PDF file by calling the created macro `\keywdef` and passing the argument 'Q4-2024' to the inner macro `\keyw` of the macro `\keywdef`. Then we print the argument 'Q4-2024' of the macro `\keyw` to the output PDF file each time the macro `\keyw` is called. We mark Section 2 with the other keyword 'Q1-2025' that is not printed to the output PDF file by calling the macro `\keywdef` again and passing the argument 'Q1-2025' to the inner macro `\keyw` of the macro `\keywdef`. Then we print print the argument 'Q1-2025' of the macro `\keyw` to the output PDF file each time the macro `\keyw` is called (Fig. 2).

```
\documentclass{article}

\begin{document}

\section{Quarterly Financial Report for Q4-2024}
\keywdef{Q4-2024}
Note that \keyw\ is the last quarter of the fiscal year
covered in this report. This report includes detailed analyses
of revenue, expenses, and profit margins for \keyw.
Additionally, the market trends observed during \keyw\ are
summarized to provide a comprehensive overview of the
quarter's performance \ldots

\section{Quarterly Financial Report for Q1-2025}
\keywdef{Q1-2025}
Note that \keyw\ is the first quarter of the fiscal year
covered in this report. This report includes detailed analyses
of revenue, expenses, and profit margins for \keyw.
Additionally, the market trends observed during \keyw\ are
summarized to provide a comprehensive overview of the
quarter's performance \ldots

\end{document}
```

Fig. 2. Calling the macros `\keywdef` and `\keyw` in the body of the LaTeX document

The source code in Fig. 1 and Fig. 2 prints the keywords to the output PDF file each time the macro `\keyw` is called (Fig. 3).

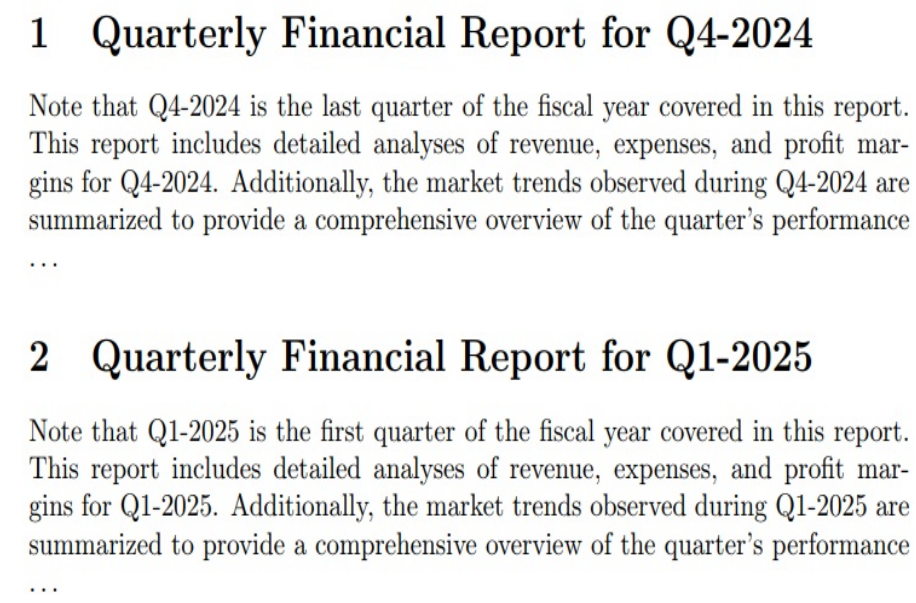

Fig. 3. The result of calling the macros `\keywdef` and `\keyw` in the output PDF file

The created macro `\keywdef` can be used in other macros that change text formatting, create structures unique to the text element marked by a keyword.

The created macro `\keywdef` can be used to create scientific, technical, educational documents, and their templates in which we can also include keywords that mark any other text elements. These keywords may contain hints, notes, short explanations, references to bibliography, and data that need to be repeated. If we change a keyword that marks any text element by calling the macro for keywords `\keywdef`, then the updated keyword is printed by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro is called. If there is no need to use standard bibliography tools, then by calling the created macro `\keywdef`, we can mark each section by a keyword and print this keyword by calling the macro `\keyw`. This keyword contains bibliographic references related to this section.

3.2 Creating the macro to control text visibility

Let us solve the following problem in the LaTeX system. In a LaTeX document, we make a given box of a text invisible in the output PDF file. At the same time, let us reserve the option to make this box of the text visible. We reserve the space occupied by this box

of the text so that we can handwrite or type the text in the blank space. This box of the text can contain the proof of a theorem, a lemma, a problem solution, an explanation, a hint, a note, and any other data.

In order to do this, in the LaTeX document, we use the 'amsmath' package for advanced capabilities for working with mathematical expressions and the 'amsthm' package for creating environments for theorems, lemmas, and proofs. We use the 'etoolbox' package for conditional logic programming.

We also use the 'tcolorbox' package to create colored boxes with customizable box styling options, and we load the 'skins' library to style different types of content and the 'breakable' library to automatically break the box from one page to another if the content of the box is too long. We define the theorem environment with the 'theorem' name and the 'Theorem' caption.

We create a new macro `\textcontrol` that provides a way to control the visibility of a given box of a text, i.e., provides a way to make it invisible or visible in the output PDF file. At the same time, the space occupied by this box of the text is reserved.

The macro `\textcontrol` takes two arguments:

- `#1` is the first argument to control the visibility of the given box of the text;
- `#2` is the second argument to input the content of the given box of the text.

In the macro `\textcontrol`, we create a local group for commands that must work within this group. In this group, creating the command `\control{visible}`, we set the text to be visible by default. Then using the command `\ifstrempty` from the 'etoolbox' package, we check whether a value is passed to the argument `#1`. If the value is not passed to the argument `#1`, then the text remains visible in the box. If the value is passed to the argument `#1`, then the given text becomes invisible when we use the command `\control{invisible}`.

We customize the style of the box in the 'tcolorbox' environment provided by the 'tcolorbox' package. We set white background, white frame, all rules of the frame to 0pt, sharp corners, enhanced styling option, automatic box breaking from one page to another, the left and right spaces between all text parts and frame to negative numbers (to keep all text parts closer to the frame), the vertical space before the box to 0ex and the vertical space after the box to 1ex. Inside the box, we place the text (for example, the proof of the theorem from a higher mathematics course) in the 'proof' environment with the argument `#2`. After that, we complete the 'tcolorbox' environment and the local group of commands. The used 'amsmath', 'amsthm', 'etoolbox', 'tcolorbox' packages, the used 'skins', 'breakable' libraries, and the created new macro `\textcontrol` are shown in Fig. 4.

```
\newcommand{\textcontrol}[2][]{%
  \begingroup
  \def\control{visible}
  \ifstrempty{#1}{}{\def\control{invisible}}%
  \tcolorbox[
  \control,
    colback=white,
    colframe=white,
    boxrule=0pt,
    sharp corners,
    enhanced,
    breakable,
    left=-0.5ex,
    right=-0.5ex,
    before skip=0ex,
    after skip=1ex
  ]
  \begin{proof}
   #2
  \end{proof}
  \endtcolorbox
  \endgroup
```

Fig. 4. The created macro `\textcontrol` that enables to control the visibility of the given box of the text (the source code is available here)

In the body of the LaTeX document, we call the created macro `\textcontrol` to which we pass the value 'off ' for the first argument and input the text (for example, the proof of the theorem from a higher mathematics course placed in the 'proof' environment) for the second argument (Fig. 5).

The source code in Fig. 4 and Fig. 5 makes the proof of the theorem from a higher mathematics course invisible in the output PDF file (Fig. 6).

If we pass any value other than 'off' to the first argument of the macro `\textcontrol` in the body of the LaTeX document, then the given box of the text will also be invisible. If we pass an empty word to the first argument of the macro `\textcontrol` (we leave the first argument empty), then the given box of the text will be visible (Fig. 7).

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{amsthm}
\usepackage{etoolbox}
\usepackage{tcolorbox}
\tcbuselibrary{skins,breakable}
\newtheorem{theorem}{Theorem}

\begin{document}

Let us discuss a property of solutions of a homogeneous linear
equation
\begin{equation}\label{eq1}
y^{\prime\prime}+a_1(x)y^{\prime}+a_2(x)y=0.
\end{equation}

It is supposed that the functions \(a_1(x)\) and \(a_2(x)\) are
continuous in an interval \((a,b)\) which can be finite or
infinite.

\begin{theorem}
If \(y_1(x)\) and \(y_2(x)\) are two solutions of homogeneous
linear equation \eqref{eq1}, then the function \(y=C_1y_1(x)
+C_2y_2(x)\) is also a solution of the equation for any
constants \(C_1\) and \(C_2\).
\end{theorem}

For brevity, we write these solutions as \(y_1\) and \(y_2\).

\textcontrol[off]{%
  Differentiate the function \(y=C_1y_1+C_2y_2\) twice:
\begin{equation*}
y^{\prime}=C_1y_1^{\prime}+C_2y_2^{\prime},\quad
y^{\prime\prime}=C_1y_1^{\prime\prime}+C_2y_2^{\prime\prime}.
\end{equation*}
Substituting \(y,y^\prime,y^{\prime\prime}\) into the left-hand
side of equation \eqref{eq1}, we obtain
\begin{multline*}
 C_1y_1^{\prime\prime}+C_2y_2^{\prime\prime}+a_1(x)
(C_1y_1^{\prime}+C_2y_2^{\prime})+a_2(x)(C_1y_1+C_2y_2)=
 \\
 =C_1(y_1^{\prime\prime}+a_1(x)y_1^{\prime}+a_2(x)y_1)
+C_2(y_2^{\prime\prime}+a_1(x)y_2^{\prime}+a_2(x)y_2).
\end{multline*}

The expressions in the parentheses are the results of the
substitution of the functions \(y_1\) and \(y_2\) in
\eqref{eq1}; since these functions are solutions of \eqref{eq1},
then both expressions are identically equal to zero, and hence
the function \(y=C_1y_1+C_2y_2\) satisfies equation \eqref{eq1}.
}

We can discuss other properties of solutions of a homogeneous
linear equation \eqref{eq1}.
\end{document}
```

Fig. 5. Calling the created macro `\textcontrol` in the body of the LaTeX document so that the proof of the theorem from a higher mathematics course is invisible

Let us discuss a property of solutions of a homogeneous linear equation

$$y'' + a_1(x)y' + a_2(x)y = 0. \tag{1}$$

It is supposed that the functions $a_1(x)$ and $a_2(x)$ are continuous in an interval (a, b) which can be finite or infinite.

Theorem 1. *If $y_1(x)$ and $y_2(x)$ are two solutions of homogeneous linear equation (1), then the function $y = C_1y_1(x) + C_2y_2(x)$ is also a solution of the equation for any constants C_1 and C_2.*

For brevity, we write these solutions as y_1 and y_2.

We can discuss other properties of solutions of a homogeneous linear equation (1).

Fig. 6. The result of calling the created macro `\textcontrol` is the proof of the theorem from a higher mathematics course made invisible in the PDF file

Let us discuss a property of solutions of a homogeneous linear equation

$$y'' + a_1(x)y' + a_2(x)y = 0. \tag{1}$$

It is supposed that the functions $a_1(x)$ and $a_2(x)$ are continuous in an interval (a, b) which can be finite or infinite.

Theorem 1. *If $y_1(x)$ and $y_2(x)$ are two solutions of homogeneous linear equation (1), then the function $y = C_1y_1(x) + C_2y_2(x)$ is also a solution of the equation for any constants C_1 and C_2.*

For brevity, we write these solutions as y_1 and y_2.

Proof. Differentiate the function $y = C_1y_1 + C_2y_2$ twice:

$$y' = C_1y_1' + C_2y_2', \quad y'' = C_1y_1'' + C_2y_2''.$$

Substituting y, y', y'' into the left-hand side of equation (1), we obtain

$$\begin{aligned} C_1y_1'' + C_2y_2'' + a_1(x)(C_1y_1' + C_2y_2') + a_2(x)(C_1y_1 + C_2y_2) = \\ = C_1(y_1'' + a_1(x)y_1' + a_2(x)y_1) + C_2(y_2'' + a_1(x)y_2' + a_2(x)y_2). \end{aligned}$$

The expressions in the parentheses are the results of the substitution of the functions y_1 and y_2 in (1); since these functions are solutions of (1), then both expressions are identically equal to zero, and hence the function $y = C_1y_1 + C_2y_2$ satisfies equation (1). □

We can discuss other properties of solutions of a homogeneous linear equation (1).

Fig. 7. The result of calling the created macro `\textcontrol` is the proof of the theorem from a higher mathematics course made visible in the PDF file

In order to make parts of a text invisible in the LaTeX system, we can choose the white text color that prints on white paper. However, the text made invisible by using this approach can be copied and pasted to the corresponding page in a PDF viewer, which is a disadvantage of this approach when testing students' knowledge. Therefore, it is advisable to use the created macro `\textcontrol`, which provides a way to control the visibility of the given box of the text, i.e., provides a way to make it invisible or visible in the output PDF file determined by the value passed to the first argument of the created macro `\textcontrol`. At the same time, this macro uses the tools of the 'tcolorbox' package, and there is no way to copy and paste an invisible text.

3.3 Creating the macro for printing text in a selected language

Let us solve the following problem in the LaTeX system. Let a LaTeX document have the text written in English, German, and French. It is necessary that the text be written, for example, in English in the output PDF file after compiling the LaTeX document. Solving such a problem is necessary to adapt the text to an audience that speaks a particular language.

For example, the LaTeX document can contain the text in several languages. Changing the language in the preamble of the LaTeX document and compiling this document, we obtain the output PDF file with the text in the selected language, and the text in other languages will not be printed. This is convenient so that the text written in different languages is in one LaTeX document.

In order to do this, let us use the 'fontenc' and 'inputenc' packages to determine the character encoding in the LaTeX document. We connect the 'babel' package to support writing in different languages and customize typographic and hyphenation rules. We pass German, French, and English in the list of language options of the 'babel' package. At the same time, the last language name passed as an option is English, and it becomes the document's current language. Then we connect the 'etoolbox' package for conditional logic programming.

The variable `\languagename` provided by the 'babel' package displays the current language name in the document, i.e., English that is the last in the list of language options of the 'babel' package. The last language first is a principle predetermined by the 'babel' package.

We define a variable `\langnm` in which we store `\languagename`. If we do not store `\languagename` in the variable `\langnm`, then the source code will not work correctly because `\languagename` changes dynamically.

We create a new macro `\langchange` that tests the current language of the LaTeX document using the command `\ifdefstring` from the 'etoolbox' package and prints the text (for example, on higher mathematics) in the current language stored in the variable `\langnm` (i.e., in English) to the output PDF file (Fig. 8).

```
\let\langnm\languagename
\newcommand{\langchange}{%
  \ifdefstring{\langnm}{english}{%
    The linear space \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) of summable functions \(f_s(x_1,\ldots,x_s)\) defined on
the phase space \(\rm{R}^{\nu s}\times \rm{R}^{\nu s}\) and
invariant under permutations of the arguments \((x_1,
\ldots,x_s)\) has the norm
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
  }{%
    \ifdefstring{\langnm}{german}{%
      Der lineare Raum \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) der summierbaren Funktionen \(f_s(x_1,\ldots,x_s)\), die
im Phasenraum \(\rm{R}^{\nu s}\times \rm{R}^{\nu s}\)
definiert sind und invariant unter Permutationen der Argumente
\((x_1,\ldots,x_s)\) sind, hat die Norm
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
    }{%
      L'espace linéaire \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) des fonctions sommables \(f_s(x_1,\ldots,x_s)\),
définies sur l'espace des phases \(\rm{R}^{\nu s}\times \rm{R}
^{\nu s}\) et invariantes par permutations d'arguments \((x_1,
\ldots,x_s)\), a pour norme
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
    }%
  }%
}
```

Fig. 8. The created macro `\langchange` that tests the current language of the LaTeX document and prints the text (for example, on higher mathematics) in the current language (English) to the output PDF file (the source code is available here)

In the body of the LaTeX document, we call the created macro `\langchange` (Fig. 9).

```
\documentclass{article}

\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[german,french,english]{babel}
\usepackage{etoolbox}

\begin{document}

\langchange

\end{document}
```

Fig. 9. Calling the created macro `\langchange` in the body of the LaTeX document

The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in English (English was stored in the variable `\langnm`) to the output PDF file (Fig. 10).

The linear space $L^1(\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s})$ of summable functions $f_s(x_1, \ldots, x_s)$ defined on the phase space $\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s}$ and invariant under permutations of the arguments $(x_1, \ldots, x_s)$ has the norm

$$\|f_s\| = \int\limits_{\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s}} dx_1 \ldots dx_s |f_s(x_1, \ldots, x_s)|.$$

Fig. 10. The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in the current language (English) to the output PDF file

If the last language name passed in the list of language options of the 'babel' package is German (Fig. 11), then German becomes the document's current language.

```
\usepackage[english,french,german]{babel}
```

Fig. 11. Connection of the 'babel' package assigning German as the current language of the LaTeX document

Then calling the created macro `\langchange` in the body of the LaTeX document (Fig. 9), we obtain the text (for example, on higher mathematics) printed in German to the output PDF file (Fig. 12).

Der lineare Raum $L^1(\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s})$ der summierbaren Funktionen $f_s(x_1,\ldots,x_s)$, die im Phasenraum $\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s}$ definiert sind und invariant unter Permutationen der Argumente $(x_1,\ldots,x_s)$ sind, hat die Norm

$$\|f_s\| = \int\limits_{\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s}} dx_1\ldots dx_s|f_s(x_1,\ldots,x_s)|.$$

Fig. 12. The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in the current language (German) to the output PDF file

If it is necessary to print the text in a language other than the three languages listed above, then this language must be the last in the list of language options of the 'babel' package, and the test `\ifdefstring` for this language must be added in the created macro `\langchange`. The created macro `\langchange` is especially useful in online learning for multilingual audiences.

4 Conclusions

In this article, we presented the methodology and generalized model of the creation of macros in the LaTeX system, and we created the following new macros:

1. the macro for keywords `\keywdef` that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro `\keyw` is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords `\keywdef`, then the updated keyword is printed by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro `\keyw` is called;
2. the macro `\textcontrol` that provides a way to make a given box of a text invisible or visible in the output PDF file;
3. the macro `\langchange` that prints a text in the selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

The source code of the created macros is presented. The created macros can be used in LaTeX documents of different classes. These macros provide flexibility in text formatting when creating scientific, technical, and educational documents in the LaTeX system. They automate working and increase productivity in this system.

References

Abraham, N. P., Luke, D. M. (2022). Preparing structured document using LaTeX. *APT Tunes*, **1**, 1–5.

Hubal, H. M. (2023). Improvement of references and footnotes in mathematical and other texts by creating macros in the LaTeX programming language. *International Journal on Information Technologies & Security*, **15(3)**, 15–22.

Lamport, L. (1994). LaTeX: *A Document Preparation System: User's Guide and Reference Manual*, 2nd ed., Reading, Massachusetts: Addison-Wesley Professional.

Lode, C. (2019). *Better books with LaTeX the agile way*, Düsseldorf, Germany: Clements Lode Verlag E.K.

Mittelbach, F., Goossens, M., Braams, J., Carlisle, D., Rowley, C. (2004). *The LaTeX companion*, 2nd ed., Boston: Addison-Wesley Professional.

Nazaruka, E. (2020). Processing use case scenarios and text in a formal style as inputs for TFM-based transformations. *Baltic Journal of Modern Computing*, **8(1)**, 48–68.

Romansky, R. (2023). Empirical evaluation of the transfer of information resources in active learning. *International Journal on Information Technologies and Security*, **15(1)**, 39–48.

Sarac, S., Durakovic, B. (2022). Analysis of student performances in online and face-to-face learning: A case study from a Bosnian public university. *Heritage and Sustainable Development*, **4(2)**, 87–94.

Šostaka, D., Borzovs, J. (2023). Towards computer-assisted Latvian ICT terminology development: from theoretical guidelines to case studies. *Baltic Journal of Modern Computing*, **11(1)**, 161–180.

Šostaka, D., Borzovs, J., Cauna, E., Keiša, I., Pinnis, M., Vasiļjevs, A. (2023). The semi-algorithmic approach to formation of Latvian Information and Communication Technology terms. *Baltic Journal of Modern Computing*, **11(1)**, 67–89.

Van Dongen, M.R.C. (2012). LaTeX *and friends*, Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

Zuzevičiūtė, V., Jatautaite, D., Butrime, E. (2025). Contemporary higher education teacher's challenges: The perspective on the AI in studies. *Baltic Journal of Modern Computing*, **13(1)**, 304–314.

Received January 21, 2026 , revised February 23, 2026, accepted April 12, 2026