

ソースコードの類似性に基づいたテストコード自動推薦ツール SuiteRec

倉地 亮介[†] 崔 恩潯^{††} 飯田 元[†]

[†] 奈良先端科学技術大学院大学先端科学技術研究科情報科学領域 〒630-0192 奈良県生駒市高山町 8916-5

^{††} 京都工芸繊維大学情報工学課程 〒606-8585 京都府京都市左京区松ヶ崎橋上町

E-mail: [†]kurachi.ryosuke.kp0@is.naist.jp, ^{††}echoi@kit.ac.jp, ^{†††}iida@itc.naist.jp

あらまし テスト工程において、テスト作成コストを削減するために様々なテストコード自動生成ツールが提案されてきた。しかし、既存のツールによって生成されるテストコードはテスト対象コードの作成経緯や意図に基づいていないという性質から開発者の保守作業を困難にする課題がある。この課題の解決方法として、本研究では OSS プロジェクト上に存在する既存の品質が高いテストコードを推薦するツール SuiteRec を提案する。また、被験者実験を行い SuiteRec の有用性を確認した。

キーワード 類似コード検出, 推薦システム, ソフトウェアテスト, テストスメル, 単体テスト

SuiteRec: Automatic Test Suite Recommendation System based on Code Clone Detection

Ryosuke KURACHI[†], Eunjong CHOI^{††}, and Hajimu IIDA[†]

[†] Graduate School of Information Science, Nara Institute of Science and Technology 8916-5, Takayama, Ikoma, Nara, 630-0192, Japan

^{††} Department of Information Science Matsugasaki, Sakyo-ku, Kyoto, 606-8585 Japan

E-mail: [†]kurachi.ryosuke.kp0@is.naist.jp, ^{††}echoi@kit.ac.jp, ^{†††}iida@itc.naist.jp

Abstract Automatically generated tests tend to be less read-able and maintainable since they often do not consider the latent objective of the target code. Reusing existing tests might help address this problem. To this end, we present SuiteRec, a system that recommends reusable test suites based on code clone detection. Given a Java method, SuiteRec searches for its code clones from a code base collected from open-source projects, and then recommends test suites of the clones. It also provides the ranking of the recommended test suites computed based on the similarity between the input code and the cloned code. We evaluate SuiteRec with a human study of ten students. The results indicate that SuiteRec successfully recommends reusable test suites.

Key words clone detection, recommendation system, software testing, test smell, unit test

1. はじめに

近年、ソフトウェアに求められる要件が高度化・多様化する一方、ユーザからはソフトウェアの品質確保やコスト削減に対する要求も増加している[?]. その中でもソフトウェア開発全体のコストに占める割合が大きく、品質確保の要ともいえるソフトウェアテストを支援する技術への関心が高まっている[?]. しかし、現状ではテスト作成作業の大部分が人手で行われており、多くのテストを作成しようとするするとそれに比例してコストも増加してしまう。このような背景から、ソフトウェアの品質を確保しつつコスト削減を達成するために、様々な自動化技術が提案されている[?], [?], [?], [?], [?].

既存研究で提案されている EvoSuite[?] は、単体テスト自動生成における最先端のツールである。EvoSuite は、対象コードを静的解析しプログラムを記号値で表現する。そして、対象コードの制御パスを通るような条件を集め、条件を満たす具体値を生成する。単体テストを自動生成することで、開発者は手作業でのテスト作成時間が自動生成によって節約することができ、またコードカバレッジを大幅に向上することができる。しかし、既存ツールによって自動生成されるテストコードは対象のコードの作成経緯や意図に基づいて生成されていないという性質から後の保守作業を困難にするという課題がある[?], [?], [?]. これは、自動生成ツールの実用的な利用の価値に疑問を提示させる。テストが失敗するたびに、開発者はテスト対象のコード内

で不具合の原因を特定するまたは、テスト自体を更新する必要があるかどうかを判断する必要がある。自動生成されたテストコードは、自動生成によって得られる時間の節約よりも読みづらく、開発者の助けになるというよりかむしろ困難するという結果が報告されている[?]。

我々は、この課題の解決するために既存テストの再利用が有効であると考えます。本研究では、OSS に存在する既存の品質の高いテストコード推薦するツール SuiteRec を提案する。推薦手法の基本となるアイデアはクローンペア間でのテストコード再利用である。SuiteRec は、入力コード片に対して類似コード片を検出し、その類似コード片に対応するテストスイートを開発者に推薦する。さらに、テストコードの良くない実装を表す指標であるテストスメルを開発者に提示し、より品質の高いテストスイートを推薦できるように推薦順位がランキングされる。

評価実験では、被験者によって SuiteRec の使用した場合とそうでない場合でテストコードの作成してもらい、テスト作成をどの程度支援できるかを定量的および定性的に評価した。その結果、SuiteRec の利用は条件分岐が多く複雑なプログラムのテストコードを作成する際にコードカバレッジの向上に効果的であること、作成したテストコードに含まれるテストスメルの数が少なく品質が高いことが分かった。また、SuiteRec は開発者が参考にしたいテストコードを上位に推薦できることを確認した。実験後のアンケートによる定性的な評価では、SuiteRec を使用した場合、被験者はテストコードの作成が容易になると認識し、また自分の作成したコードに自信が持てることが分かった。

以降、2 章では、本研究に関わるコードクローン及びソフトウェアテストについての背景を説明する。3 章では、本研究で提案するテストコード自動推薦ツール SuiteRec について説明する。4 章では、被験者による SuiteRec の評価実験について説明する。5 章では、SuiteRec の有効性と妥当性への脅威について議論する。最後に、6 章でまとめと今後の課題について説明する。

2. クラスファイルの説明

2.1 テンプレートと記述方法

以下のテンプレートに従って記述してください。原稿執筆に際しては、本クラスファイルとともに配布されるテンプレート (template.tex) を利用できます。

```
\documentclass[technicalreport]{ieicej}
\jtitle{和文題目}
\jsubtitle{和文副題}
\etitle{英文題目}
\esubtitle{英文副題}
\authorlist{%
  \authorentry[densi@firstuniv.ac.jp]
    {電子 花子}{Hanako DENSHI}{Tokyo}
  \authorentry[joho@ohsakacorp.co.jp]
    {情報 太郎}{Jiro JOHO}{Osaka}
}
```

```
\affiliate[Tokyo]{第一大学工学部\hskip1zw
  〒105--0123 東京都港区山田 1--2--3}
{Faculty of Engineering,
  First University\hskip1em
  1--2--3 Yamada, Minato-ku, Tokyo,
  105--0123 Japan}
\affiliate[Osaka]{大阪株式会社開発部\hskip1zw
  〒565--0456 大阪府吹田市河田 4--5--6}
{R&D Division, Osaka Corporation\hskip1em
  4--5--6 Kawada, Suita-shi,
  565--0456 Japan}
```

```
\begin{document}
\begin{jabstract}
和文あらまし
\end{jabstract}
\begin{jkeyword}
和文キーワード
\end{jkeyword}
\begin{eabstract}
英文アブストラクト
\end{eabstract}
\begin{ekeyword}
英文キーワード
\end{ekeyword}
\maketitle
---- (略) ----
```

- 「技術研究報告」の体裁にするには、ドキュメントクラスのオプションとして `technicalreport` を指定します。

- `\jtitle` には和文題目を指定します。任意の場所で改行したいときは、`\\` で改行できます。

- `\etitle` は、英文題目を指定します。

- 和文副題および英文副題を指定することができます。それぞれ、`\jsubtitle` と `\esubtitle` に記述します。

- 和文発表者名および英文発表者名は、以下のように記述します。発表者名、所属、メールアドレスなどの出力体裁を自動的に整えます。

```
\authorlist{%
  \authorentry[メールアドレス]{和文発表者名}
  {英文発表者名}{所属ラベル}
}
```

- 第 1 引き数は、メールアドレスを指定します。これは省略可能です。

- “_” を含むメールアドレスの場合は

```
\noexpand\noexpand\noexpand\_
と記述してください。
```

発表者が複数の場合で、メールアドレスをお持ちでない方がいる場合は、かならず `[]` を記述した上で、中を空にしてください。メールアドレスは 1 人につき 1 つだけ記述します。1 人につき複数のアドレスには対応していません。

メールアドレスの記述が原因でエラーを生じたり、出力が望

み通りの結果にならない場合は、`\MailAddress` に直接記述してください。

`\MailAddress{\daggername@xx.yy.zz.jp}`

— 第2引き数は和文発表者名を指定します。姓と名の間に必ず半角のスペースを挿入してください（スペースを挿入し忘れた場合には、ワーニングが出力されます）。

— 第3引き数は英文発表者名を指定します。ファミリーネームは大文字で記述します。

— 第4引き数は発表者の所属ラベルを指定します。このラベルは、後述する `\affiliate` の第1引き数に対応します。ラベルは大学名、企業名、地名などを表す簡潔なものにしてください。複数の所属がある場合には、カンマ “,” でラベルを区切って記述します。

発表者が一人で所属がない場合は、`none` と指定します。

発表者が複数で所属のない方がいる場合は、`none` 以外の適当なラベルを付けたうえで、`\affiliate` は記述しません。

ラベルの前後やカンマの後ろに余分なスペースを入れないでください。`{Tokyo}` と `{Tokyo_}` は所属が違うものと判断します。

- 和文発表者名および英文発表者名を任意の場所で改行する必要がある場合は、それぞれ `\alignorder`、`\breakauthorline` コマンドで制御することができます。

`\alignorder=3`

と記述すれば、和文発表者名のリストを1行に3名ずつ並べます。

また、

`\breakauthorline{3}`

と記述すれば、英文発表者名の3人目の後ろで改行します。カンマで区切って複数の数字を指定することもできます。

- 所属は `\affiliate` に指定します。

`\affiliate[ラベル]`

`{和文勤務先\hskip1zw 和文連絡先住所}`

`{英文勤務先\hskip1em 英文連絡先住所}`

第1引き数に `\authorentry` で指定したラベルを記述します。ラベルの前後に余分なスペースを挿入しないでください。第2引き数に和文の所属を、第3引き数に英文所属を指定します。勤務先と連絡先住所を `\hskip1zw` などを使うことによって少し間を空けてください。`\authorentry` に記述したラベルの出現順に記述してください。

- 和文の「あらまし」「キーワード」は、`jabstract` 環境、`jkeyword` 環境にそれぞれ記述します。また、英文の「abstract」「key words」は、`eabstract` 環境、`ekeyword` 環境にそれぞれ記述します。

2.2 論文誌の体裁に変更する場合

「論文」、「レター」などの論文誌の体裁に変更する場合は、以下の点に注意してください。

- `\jsubtitle` と `\esubtitle` は記述しても無効になります。

- `\affiliate` の和文連絡先住所を簡略化する必要があります。「和文論文誌投稿のしおり」(<http://www.ieice.or.jp/>

表1 サイズと行間の変更

Table 1 Settings of size and baselineskip.

<code>\normalsize</code>	9 pt (5.125 mm)
<code>\small</code>	8 pt (4.5 mm)
<code>\footnotesize</code>	7 pt (4 mm)
<code>\scriptsize</code>	6 pt (8 pt)
<code>\tiny</code>	5 pt (6 pt)
<code>\large</code>	10 pt (5.5 mm)
<code>\Large</code>	11 pt (6.75 mm)
<code>\LARGE</code>	12 pt (8.25 mm)
<code>\huge</code>	14 pt (25 pt)
<code>\Huge</code>	17 pt (30 pt)

`jpn/ronbun.html`), またはバックナンバーを参照してください。

- `jabstract` 環境は `abstract` 環境と見なしますが、`eabstract` 環境は、最終ページに一段組で出力されます。

- `jkeyword` 環境は `keyword` 環境と見なしますが、`ekeyword` 環境は、最終ページに一段組で出力されます。

2.3 文字サイズと行間

本文の活字の大きさを、9 pt に設定しています。したがって、`\normalsize`、`\small` などのサイズおよび行間を表1に示すように変更しています。

2.4 見出しの字どり

`\section`、`\subsection` などについては、本誌のスタイルにより、その見出しが4字以下の際、5字どりになるように設定しています(2.11の見出しを参照)。

2.5 ディスプレー数式

数式の頭は左端から1字下げのところに、また、数式番号は右端から1字入ったところに出力される設定になっています。この設定を前提に数式の折り返しを調整してください。`\documentclass` のオプションとして `fleqn` を指定する必要はありません。

技術研究報告原稿は二段組みで一段の左右幅がせまいため、数式と数式番号が重なったり、数式がはみ出したりすることが頻繁に生じると思われます。`Overfull \hbox` のメッセージには特に気をつけてください。

数式記述の際のヒントについては、3.2 および 3.3 が参考になるかもしれません。

2.6 図表とキャプション

図表を置く位置、キャプションの記述、図の取り込み、表の記述などについて説明します。

2.6.1 図表を置く位置

`float` 環境は、それが初めて引用される段落の直後または直前あたりに挿入することが基本ですが、二段組みの場合は、それが初めて引用されるページより前に置くことが必要になることがあります。図表の出力位置は、図表の参照と同じページか、無理な場合は次のページに置くことが基本ですから、二段組みの図表の場合は、`float` 環境を記述する位置の試行錯誤が必要となることがあります。

図表の出力位置を指定するオプションとして、`[h]` の使用は

```

\begin{figure}[tb]
%\capwidth=60mm
%\ecapwidth=60mm
\vspace{45mm}
\caption{図キャプションの例}
\label{fig:1}
\ecaption{An example of caption.}
\end{figure}

```

図 1 図キャプションの例

Fig. 1 An example of caption.

避け、[tb]、[tbp]などを指定して、ページの天か地に置くことを基本にしてください。

2.6.2 キャプションとラベル

図表のキャプションは、和文と欧文で指定する必要があるため、`\ecaption` というマクロを用意しました。使い方は `\caption` と同じです。図 1 のように記述してください。

- キャプションの幅は、一段の場合には一段の幅に、二段ぬきの場合はテキストの幅の 3 分 2 に設定しています。
- キャプションを任意の場所で改行したい場合は、`\\` を使って改行することができます。標準の $\text{\LaTeX 2}_{\epsilon}$ でこういう使い方をすると、エラーになるので注意してください。
- また、`\capwidth` および `\ecapwidth` に長さを指定すれば、その幅で折り返すことができます。

```
\capwidth=60mm
```

これは `\caption` コマンドの前に指定します。

- `\label` を記述する場合は、必ず `\caption` の直後に置きます。上におくと `\ref` で正しい番号を参照できません。

2.6.3 図の取り込み

図の取り込みに関しては、「技術研究報告」では、「発表者が作成した原稿をそのままオフセット印刷します」ので、図はどのような形式のものでも構いません。ここでは PDF 形式の図を読み込む場合の説明を簡単にします。

例えば、パッケージとして

```
\usepackage[dvipdfmx]{graphicx}
```

を指定し、

```

\begin{figure}[tb]
\begin{center}
\includegraphics{file.pdf}
\end{center}
\caption{}
\ecaption{}
\end{figure}

```

のような使い方をします。

詳しくは TeX Wiki [23] を参照されることを勧めます。また文献では [9], [11], [13], [14], [16]~[18] などがあります。

2.6.4 表の記述

表は `\small` (8pt) で組まれるように設定しています。

例えば、以下のように記述します。

```
\begin{table}[tb]
```

```
\caption{和文キャプション}
```

```
\label{table:1}
```

```
\ecaption{英文キャプション}
```

```
\begin{center}
```

```
\begin{tabular}{|c|c|c|}
```

```
\Hline %% ←
```

```
A & B & C \\
```

```
\hline
```

```
x & y & z \\
```

```
\Hline %% ←
```

```
\end{tabular}
```

```
\end{center}
```

```
\end{table}
```

`\caption` は `tabular` 環境の上に記述します。本誌では、表の罫の一番上と一番下を太くします。このため `\Hline` というマクロを使用してください。これは

```
\def\Hline{\noalign{\hrule height 0.4mm}}
```

と定義してあります (表 1, 2 参照)。`\hline` の太さは 0.1mm です。

2.7 文献リストと文献番号の参照

\LaTeX を利用しない場合は、文献リストの記述——著者名とイニシャル、表題・書名、雑誌名・発行所および雑誌名の略語、巻、号、ページ、発行年などの体裁——は「原稿の書き方」に従ってください。

\LaTeX を使って、文献用データベースファイルから文献リストを作成する場合は、文献用スタイルとして `tieice.bst` が利用できます。標準でインストールされているようです。得られた文献リストを必要に応じて修正してください。 \LaTeX の使い方は、文献 [14], [16], [17] などを参考にしてください。

文献引用のコマンドは、`cite.sty` および `citesort.sty` に手を加えたものを使用しています。例えば、`\cite{latex,FG01,PEn,Fujita5,tex}` と記述すれば、“[16], [19], [20], [21], [1]” となるところを、“[1], [16], [19]~[21]” のように、番号順に並べ替え、かつ番号が続く場合は“~”でつなぎます。

2.8 定理、定義などの環境

定理、定義、命題などの定理型環境を記述するには `\newtheorem` (文献 [14], [16] 参照) が利用できますが、下の出力例のように、本誌のスタイルにあわせて、 $\text{\LaTeX 2}_{\epsilon}$ の標準と異なり、環境の上下の空きやインデントを変更し、見出しはゴシックとならず、本文の欧文もイタリックになりません。

例えば、

```
\newtheorem{teiri}{定理}
```

```
%\thmbracket{ }{ }
```

```
\begin{teiri}
```

これは“定理”の例です。

このような出力になります。

text in Roman typeface.

```
\end{teiri}
```

とすれば、

[定理 1] これは“定理”の例です。このような出力になりま

す。text in Roman typeface.

と出力されます。

また、(ステップ 1)のように、前後の括弧を変えたいときは、`\thmbracket{ }{ }` のように `\thmbracket` の 2 つの引き数に前後の括弧をそれぞれ記述します。

2.9 脚注と脚注マーク

脚注マークは “(註 1)” という形で出力されます。

2.10 verbatim 環境

`verbatim` 環境のレフトマージン、行間、サイズを変更することができます [17]。デフォルトは

```
\verbatimleftmargin=0pt
% レフトマージンは 0pt
\def\verbatimsize{\normalsize}
% 本文と同じサイズ
\verbatimbaselineskip=\baselineskip
% 本文と同じ行間
```

ですが、それぞれパラメータやサイズ指定を変更することができます。

```
\verbatimleftmargin=2zw
% --> レフトマージンを 2 字下げに
\def\verbatimsize{\small}
% --> 文字の大きさを \small に
\verbatimbaselineskip=3mm
% --> 行間を 3mm に
```

2.11 その他

2.11.1 ieicej.cls で定義しているマクロ

(1) 「証明終」を意味する記号 “□” を出力するマクロとして `\QED` を定義してあります [1]。 `\hfill\Box` では、この記号の直前の文字が行末に来る場合、記号が行頭に来てしまいますので、`\QED` を使ってください。 “□” を出力するには、パッケージとして

```
\usepackage{latexsym}
```

が必要です。

(2) `\onelineskip`, `\halflineskip` という行間スペースを定義しています。その名のとおりに、1 行空け、半行空けに使ってください。和文の組版の場合は、こうした単位の空け方が好まれます。

(3) 2 倍ダッシュの “——” は、`\ddash` というマクロを使ってください (3.1 参照)。— を 2 つ重ねると、その間に若干のスペースが入ることがあり見苦しいからです。

(4) このクラスファイルではこのほかにあらかじめ、`\RN`, `\FRAC`, `\MARU`, `\kintou`, `\ruby` というというマクロ [1], [17] を定義しています (表 2)。

2.11.2 $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ について

数式のより高度な記述のために、 $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ のパッケージ (文献 [11], [14] 参照) を使う場合には、パッケージとして

```
\usepackage[fleqn]{amsmath}
```

が必要です。この場合、オプションとして `[fleqn]` を必ず指定してください。

`amsmath` パッケージは、多くのファイルを読み込みますが、

表 2 その他のマクロ

Table 2 Miscellaneous macros.

<code>\RN{2}</code>	II
<code>\RN{117}</code>	CXVII
<code>\FRAC{\${\pi}\$}{2}</code>	$\pi/2$
<code>\FRAC{1}{4}</code>	$1/4$
<code>\MARU{1}</code>	①
<code>\MARU{a}</code>	Ⓐ
<code>\kintou{4zw}{記号例}</code>	記号例
<code>\ruby{砒}{ひ}\ruby{素}{そ}</code>	砒素

ボールドイタリックだけを使いたい場合は、

```
\usepackage{amsbsy}
```

で済みます。

また、記号類だけを使いたい場合は、

```
\usepackage{amssymb}
```

で済みます。

なお、 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X} 2_{\epsilon}$ では `\mbox{\boldmath x}` に代えて、`\boldsymbol{x}` を使うことを勧めます。これならば、数式の上付き・下付きで使うと文字が小さくなります。

3. タイピングの注意事項

3.1 美しい組版のために

(1) 和文の句読点は “,” “.” (全角記号) を使用してください。和文中では、欧文用のピリオドとカンマ “,” “.” (“”) (半角) は使わないでください。

(2) 括弧類は、和文中で欧文を括弧でくくる場合は全角の括弧を使用してください。欧文中ではすべて半角ものを使用してください。

例：スタイル (Style) ファイル / some (Style) files

上の例のように括弧のベースラインが異なります。

(3) ハイフン (—), 二分ダッシュ (—), 全角ダッシュ (—), 二倍ダッシュ (`\ddash`) の区別をしてください。

ハイフンは well-known など一般的な欧単語の連結に、二分ダッシュは pp.298–301 のように範囲を示すときに、全角ダッシュは欧文用連結の em-dash (—) として、二倍ダッシュは和文用連結として使用してください。

(4) アラインメント以外の場所で、空行を広くとるため、`\` による強制改行を乱用するのはよくありません。

空行の直前に `\` を入れたり、`\` を 2 つ重ねれば、確かに縦方向のスペースが広がりますが、`\Underfull \hbox` のメッセージがたくさん出力されて、重要なメッセージを見落としがちになります [2]。

(5) (`_word_`) のように “()” 内や “()” 内の単語の前後にスペースを入れないでください。

(6) プログラムリストなど、インデントが重要なものは、`\hspace*{??mm}` の使用や `\` などによる強制改行で整形するのではなく、`list` 環境や `tabbing` 環境などを使って赤字が入っても修正がしやすいように記述してください。

3.2 数式の記述

(1) 数式モードの中でのハイフン、二分ダッシュ、マイナスの区別をしてください。

例えば、

```
$A^{\mathrm{b}}\mbox{\scriptsize -}
\mathrm{c}$
```

$A^{b-c} \Rightarrow$ ハイフン

```
$A^{\mathrm{b}}\mbox{\scriptsize --}
\mathrm{c}$
```

$A^{b-c} \Rightarrow$ 二分ダッシュ

```
$A^{\mathrm{b-c}}$
```

$A^{b-c} \Rightarrow$ マイナス

となります。それぞれの違いを確認してください。

(2) 数式の中で、 $<$, $>$ を括弧のように使用することがよくみられますが、数式中ではこの記号は不等号記号として扱われ、その前後にスペースが入ります。このような形の記号を括弧として使いたいときは、`\langle` (`()`), `\rangle` (`()`) を使うようにしてください。

(3) 複数行の数式でアラインメントをするときに数式が + または - で始まる場合、+ や - は単項演算子とみなされます (つまり、「 $+x$ 」と「 $x+y$ 」の + の前後のスペースは変わります)。したがって、複数行の数式で + や - が先頭にくる場合は、それらが 2 項演算子であることを示す必要があります [16]。

```
\begin{eqnarray}
y &=& a + b + c + \dots + e \\
&& \& \& \mbox{} + f + \dots
\end{eqnarray}
```

```
\end{eqnarray}
```

(4) $\mathrm{T}_\mathrm{E}X$ は、段落中の数式の中 ($\$...\$$) では改行をうまくやってくれないことがあるので、その場合には `\allowbreak` を使用することを勧めます [12]。

3.3 長い数式の処理

数式と数式番号が重なったり数式がはみ出したりする場合の対処策を、いくつか挙げます。

例 1 `\!` で縮める。

$$y = a + b + c + d + e + f + g + h + i + j + k + l + m(1)$$

のように数式と数式番号が重なるか、かなり接近する場合は、まず、2 項演算記号や関係記号の前後を、`\!` ではさんで縮める方法があります。

```
\begin{equation}
y\!=\!a\!+\!b\!+\!c\!+\! \dots \!+\!m
\end{equation}
```

$$y = a + b + c + d + e + f + g + h + i + j + k + l + m \quad (2)$$

例 2 `eqnarray` 環境を使う。

上のようにして縮めても、なお重なったりはみ出してしまう場合は、`eqnarray` 環境を使って

```
\begin{eqnarray}
y &=& a+b+c+d+e+f+g+h\nonumber\\
&& \& \& \mbox{} + i+j+k+l+m
\end{eqnarray}
```

```
& \& \mbox{} + i+j+k+l+m
\end{eqnarray}
```

と記述すれば、

$$y = a + b + c + d + e + f + g + h + i + j + k + l + m \quad (3)$$

となります。

例 3 `\mathindent` を変更する。

数式を途中で切りたくない場合は

```
\mathindent=0zw % <-- <1>
```

```
\begin{equation}
```

```
y=a+b+c+d+e+f+g+h+i+j+k+l+m
```

```
\end{equation}
```

```
\mathindent=1zw % <-- <2> デフォルト
```

と記述すれば (`<1>`)、

$$y = a + b + c + d + e + f + g + h + i + j + k + l + m \quad (4)$$

となって、数式の頭が左端にきます。この場合、その数式のあとで `\mathindent` を元に戻すことを忘れないでください (`<2>`)。

例 4 `\lefteqn` を使う。

$$\iint_S \left(\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} \right) dx dy = \oint_C \left(U \frac{dx}{ds} + V \frac{dy}{ds} \right) ds \quad (5)$$

上のように、= までが長くて、数式がはみ出したり、数式と数式番号がくっつく場合には、`\lefteqn` を使って

```
\begin{eqnarray}
\lefteqn{
\int\!\!\!\int_S
\left(\frac{\partial V}{\partial x}-\frac{\partial U}{\partial y}\right)
dx dy
}\quad\mathrm{nonumber}\\
&=& \oint_C \left( U \frac{dx}{ds} + V \frac{dy}{ds} \right) ds
\end{eqnarray}
```

と記述すれば、

$$\iint_S \left(\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} \right) dx dy = \oint_C \left(U \frac{dx}{ds} + V \frac{dy}{ds} \right) ds \quad (6)$$

のような形にできます。

例 5 `\arraycolsep` を変える。

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (7)$$

上の行列は `array` 環境を使って記述しましたが、`array` 環

境を使って数式がはみ出す場合は、数式全体のフォントサイズを変える前に、

```
\begin{equation}
\arraycolsep=3pt % <-- <1>
A = \left(
\begin{array}
{@{\hskip2pt}cccc@{\hskip2pt}}% <-- <2>
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{array}
\end{equation}
```

<1> のように、`\arraycolsep` の値（デフォルトは 5pt）を小さくしてみるか、<2> のように @ 表現を使うことができます。

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (8)$$

式 (7) と式 (8) を比べてください。

例 6 `\quad` の定義を変える。

行列を記述する場合に使用する `\matrix`、`\pmatrix` はコラムの間に `\quad` が挿入されているので、間隔を縮めるには、ディスプレイ数式環境の中で、`\def\quad` の定義を変えてみてください。例えば

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (9)$$

のような `\pmatrix` で記述した行列式で、`\quad` の定義を変更すると

```
\begin{equation}
\def\quad{\hskip.5em\relax}
% デフォルトは \hskip1em
A = \pmatrix{
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
}
\end{equation}
```

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (10)$$

となります。

`amsmath` パッケージを利用するときは、`\matrix`、`\pmatrix` はそれぞれ、`\begin`、`\end` 型の `matrix`、`pmatrix` 環境に変わるので注意してください。この場合は、例 5 が参考になります。

以上挙げたような処理でもなお数式がはみ出す場合は、あまり勧められませんが、以下のような方法があります。

- `small`、`footnotesize` で数式全体を囲む。
- 分数が横に長い場合は、分子・分母を `array` 環境で 2 階建てにする。
- `\scalebox` を使って、数式の一部もしくは全体をスケールリングする。
- 二段抜きの `table*` もしくは `figure*` 環境に入れる。この場合、数式番号に注意する必要があります。

文 献

- [1] D.E. クヌース, 改訂新版 \TeX ブック, アスキー出版局, 東京, 1992.
- [2] 磯崎秀樹, \LaTeX 自由自在, サイエンス社, 東京, 1992.
- [3] S. von Bechtolsheim, \TeX in Practice, Springer-Verlag, New York, 1993.
- [4] 藤田眞作, 化学者・生化学者のための \LaTeX —パソコンによる論文作成の手引, 東京化学同人, 東京, 1993.
- [5] 阿瀬はる美, てくてく \TeX , アスキー出版局, 東京, 1994.
- [6] N. Walsh, Making \TeX Work, O'Reilly & Associates, Sebastopol, 1994.
- [7] D. Salomon, The Advanced \TeX book, Springer-Verlag, New York, 1995.
- [8] 藤田眞作, \LaTeX マクロの八衢, アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 東京, 1995.
- [9] 中野賢, 日本語 $\LaTeX 2_{\epsilon}$ ブック, アスキー出版局, 東京, 1996.
- [10] 藤田眞作, $\LaTeX 2_{\epsilon}$ 階梯, アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 東京, 1996.
- [11] 乙部蔵己, 江口庄英, $\pLaTeX 2_{\epsilon}$ for Windows Another Manual, ソフトバンク パブリッシング, 東京, 1996–1997.
- [12] ポール W. エイブラハム, 明快 \TeX , アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 東京, 1997.
- [13] 江口庄英, Ghostscript Another Manual, ソフトバンク パブリッシング, 東京, 1997.
- [14] マイケル・グーセンス, フランク・ミッテルバッハ, アレキサンダー・サマリン, \LaTeX コンパニオン, アスキー出版局, 東京, 1998.
- [15] ビクター・エイコー, \TeX by Topic— \TeX をよく深く知るための 39 章, アスキー出版局, 東京, 1999.
- [16] レスリー・ランポート, 文書処理システム $\LaTeX 2_{\epsilon}$, ピアソンエデュケーション, 東京, 1999.
- [17] 奥村晴彦, [改訂版] $\LaTeX 2_{\epsilon}$ 美文書作成入門, 技術評論社, 東京, 2000.
- [18] マイケル・グーセンス, セバスチャン・ラッツ, フランク・ミッテルバッハ, \LaTeX グラフィックスコンパニオン, アスキー出版局, 東京, 2000.
- [19] マイケル・グーセンス, セバスチャン・ラッツ, \LaTeX Web コンパニオン— \TeX と HTML/XML の統合, アスキー出版局, 東京, 2001.
- [20] ページ・エンタープライゼス(株), $\LaTeX 2_{\epsilon}$ マクロ & クラスプログラミング基礎解説, 技術評論社, 東京, 2002.
- [21] 藤田眞作, $\LaTeX 2_{\epsilon}$ コマンドブック, ソフトバンク パブリッシング, 東京, 2003.
- [22] 吉永徹美, $\LaTeX 2_{\epsilon}$ マクロ & クラスプログラミング実践解説, 技術評論社, 東京, 2003.
- [23] <https://oku.edu.mie-u.ac.jp/~okumura/texwiki/>

付 録

1. PDF の作成方法と A4 用紙への出力

- PDF に書き出すには二通りの方法があります。

(1) dvipdfmx を使って PDF に変換する。

```
dvipdfmx -p a4 -x 1in -y 1in -o file.pdf file.dvi
```

オプションの `-p a4 -x 1in -y 1in` は省略できます。

(2) まず, dvips を使用して, ps に書き出します (以下では段幅の関係で折り返します)。

```
dvips -Pprinter -t a4 -0 0in,0in
```

```
-o file.ps file.dvi
```

`printer` には, 使用するプリンタ名を記述します。オプションの `-t a4 -0 0in,0in` は省略できます。

次に Acrobat Distiller で PDF に変換します。

- dvips を使用して A4 用紙に出力する場合のパラメータはおおよそ以下のような設定になります。

```
dvips -Pprinter -t a4 -0 0in,0in file.dvi
```

`printer` には使用するプリンタ名を記述します。オプションの `-t a4 -0 0in,0in` は省略できます。

2. 削除したコマンド

本誌の体裁に必要なないコマンドは削除しています。削除したコマンドは, `\part`, `\theindex`, `\tableofcontents`, `\titlepage`, ページスタイルを変更するオプション (`headings`, `myheadings`) などです。