Wissenschaftliches Arbeiten und Präsentation
(WS 2025/26)

Martin Held

FB Informatik
Universität Salzburg
A-5020 Salzburg, Austria
held@cs.sbg.ac.at

7. Juli 2025

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Personalia: M. Held

Instructor: M. Held.

Email: `held@cs.sbg.ac.at`.

Base-URL: `www.cosy.sbg.ac.at/~held`.

Office: Universität Salzburg, FB Informatik, Rm. 1.20,
Jakob-Haringer Str. 2, 5020 Salzburg-Itzling.

Phone number (office): (0662) 8044-6304.

Phone number (secr.): (0662) 8044-6300.

Formalia: M. Held

URL of course: `.../teaching/wiss_arbeiten/wap.html`.

Lecture time: Friday 7^{45}–10^{55} (with a break of 20–25 minutes).

Venue: PLUS, Informatik, T03, Jakob-Haringer Str. 2.

Note — graded according to continuous-assessment mode!

— regular attendance is compulsory!

Topics Discussed

In this lecture I will discuss the following topics and corresponding software packages:

Scientific Literature: types and characteristics, documentation, literature search, bibliographic metrics, ethical issues.

Scientific Presentations: guidelines for oral presentations and written documents.

Scientific Text Processing: LaTeX, PostScript, Encapsulated PostScript.

Generating Slides: PDF, pdfLaTeX, LaTeX beamer class.

Drafting and Generating Plots: Ipe, tgif, Gnuplot, Ti*k*Z.

Symbolical Mathematics: Mathematica.

Obviously, time constraints do not allow me to discuss tons of software packages in detail. I have selected those packages according to whether I've personally found them useful for my own scientific work. All packages discussed are freely available (for students) and can be installed on any PC running Linux. (Most of them should also be available for MS-based platforms.)

Electronic Slides and Online Material

In addition to these slides, you are encouraged to consult the WWW home-page of this lecture:

> *www.cosy.sbg.ac.at/~held/teaching/wiss_arbeiten/wap.html.*

In particular, this WWW page will contain links to online manuals.

A Few Words of Warning

- I hope that these slides will help you to get acquainted with the software packages discussed. However, I would like to warn you explicitly not to regard these slides as the sole source of information on the topics of my lecture. It may and will happen that I'll use the lecture for talking about subtle details of some package that are not covered in these slides! In particular, by making these slides available to you I do not intend to encourage you to attend the lecture on an irregular basis.
- See also In Praise of Lectures by T.W. Körner.
- Also, I hope that you will realize that most software packages dealt with in this lecture will only be fully appreciated after using them for yourself. It will be considerably more difficult to learn, say, LaTeX if you are not prepared and willing to get your hands on a computer and try it out personally.

Acknowledgments

I am happy to acknowledge that I benefited from material published by colleagues on diverse topics presented in this lecture. In particular, several sample codes (for LaTeX figures, PostScript, etc.) are borrowed from other publications. Similarly, some descriptions of software packages were copied from their respective user manuals. While some of the material used for this lecture was originally presented in traditional-style publications (such as textbooks), some other material has its roots in non-standard publication outlets (such as online documentations).

Andreas Uhl contributed slides for the first part of this course.

Information and data on publication statistics was provided by Katherine Eve (Publisher, Geochemistry and Geophysics, Elsevier Ltd.).

Salzburg, July 2025 Martin Held

Legal Fine Print and Disclaimer

To the best of our knowledge, these slides do not violate or infringe upon somebody else's copyrights. If copyrighted material appears in these slides then it was considered to be available in a non-profit manner and as an educational tool for teaching at an academic institution, within the limits of the "fair use" policy. For copyrighted material we strive to give references to the copyright holders (if known). Of course, any trademarks mentioned in these slides are properties of their respective owners.

These slides are made available without warrant of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. In no event shall the copyright holder(s) and/or their respective employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, arising out of or in connection with the use of information provided in these slides.

Table of Content

Publishing and Retrieving Scientific Results

Scientific Literature and Problem Solving

Don't re-invent the wheel!

It is common sense to try to leverage existing knowledge, methods, and solutions rather than to develop new ones from scratch unnecessarily.

- Checking the literature and thinking about a solution are *alternating steps* that depend on each other and should be iterated:
  - Own attempts to solve a problem will clarify where a solution might be found in the literature.
  - Looking into the literature will provide new ideas on how to solve the problem.

Diving into scientific "literature"

Understanding how scientific literature is organized helps to correctly cite sources, follow citation trails, assess the credibility of sources, and use bibliographies to discover additional relevant literature.

What is Scientific Literature?

- *Scientific literature* refers to scholarly publications that report original research, review existing findings, and discuss theoretical developments in various fields of science.
- Scientific literature encompasses various types of documents, including:
  - monographs (books),
  - articles in journals,
  - articles in collections,
  - written versions of conference talks (proceedings papers),
  - technical reports of research organizations,
  - academic theses.
- Publications in the form of
  - software systems and applets,
  - algorithm libraries,
  - knowledge bases, and
  - WWW pages

  are not regarded as "scientific literature" in the traditional sense.
- Although patent descriptions are important in the context of scientific and technological advancement, they do not typically count as scientific publications.

Characteristics of Scientific Literature

- Scientific publications have different characteristics with respect to
  - authorship,
  - contents,
  - originality,
  - production,
  - organization,
  - quality control.

Monographs

Authorship: A book has one or several authors who write the book. Upon writing a publisher has to be found for processing the book. In practice, usually a publisher is sought before the book is written. Often, scholars of high reputation are approached by publishers or editors of book series to write a book on a specific subject.

Content: A monograph treats a specific area of computer science in a systematic and complete form. The area may be a traditional area seen under a new or specific perspective or a new area whose results are scattered in various other sources like journal articles and conference papers. The content is also determined by the level of background the author presupposes on the side of the readers: e.g., undergraduate texts, graduate texts, research monographs.

Originality: Mostly, the results contained in a monograph are *not* new but were already published earlier in other sources. However, explaining everything in one uniform context or filling gaps in a systematic treatment etc. may be quite a creative process. (But it is not considered original research in computer science.)

Monographs

Production and Organization: The authors write the book and transfer the copyright to the publishing company which publishes the book. A certain number of copies of the book — an “edition” — is produced in one process and put on stock. The number of copies of one edition may range from a few hundreds to several thousand copies. Before a new edition is printed, the authors are invited to update, improve and possibly extend the book.

The company pays 5–10% of the retail price as “royalty” to the authors for each copy sold. Usually the authors also receive complimentary copies or the right to obtain copies at a reduced price.

Quality control: At good publishing companies, the scientific quality of monographs is checked by an “editor” who normally is a renowned expert in the field of the book. Often, one or several editors are in charge of a book series in a specific area. Typically, additional scholars – so-called “referees” – are asked to assess the quality of some or all of the book’s chapters. Furthermore, publishers employ a “copy editor” or “desk editor”, who is in charge of correcting grammar and style. After the publication of the book on the market critical reviews about the book may be published in “review journals”: short summary, assessment of value and criticism of the book.

Journal Articles

Authorship: Every scholar may be an author of an article in a journal. In fact, every scholar should strive for publishing his/her results in journals since this is the type of scientific publication with the highest quality standard. In case of more than one author the authors may be ordered alphabetically or ordered in accordance to the importance of their contribution to the paper.

Contents and Originality: Journal articles contain *new* results within the scope of the journal. (Exceptionally, journals also publish survey articles on emerging and topical fields. Usually, such articles are "by invitation", i.e., top scholars in the respective field are asked to submit a survey). Journal articles are directed towards the relatively small group of expert readers that work in the field covered by the journal. There are more than 1000 "refereed" journals in math/CS.

Production: Like books, journals are published by publishing companies. The author prepares a manuscript and sends ("submits") it to the editor (or to one of the editors) of the journal. Sometimes the "editorial board" of a journal may be quite big – ten to fifteen people – in order to represent the scope of the journal well. The impulse to start a new journal tends to be a joint effort of a group of scholars who want to open a publication forum for their field of expertise and of a publishing company which sees a niche in the market.

Journal Articles

Organization: If the editor accepts a manuscript after the refereeing process then it is sent to the publishing company for printing. Also, the editor may suggest an “issue” into which the paper should go. The issues of a journal appear on a regular basis, for example quarterly, bimonthly or monthly. Typically, an issue has 50–150 pages and contains several articles (“papers”). Several issues are combined in a volume; usually, a volume comprises the issues that appear in one calendar year.

The publisher owns the copyright for the articles and no royalties are paid to the authors. (Sometimes authors are even asked to share the printing costs.) Journals are sold to “subscribers” (such as scientific libraries). Often, publishers grant reduced rates for individual subscriptions.

Acceptance of articles in journals is essentially guaranteed if the paper is in the scope of the journal and the quality of the paper meets the scientific standard of the journal. In contrast to books, acceptance of articles in journals is not at all driven by economic considerations.

Journal Articles

Quality control: The scientific quality of journal papers is checked by the following "refereeing procedure", which is fairly standardized for international journals in mathematics and computer science.

1. An author submits a manuscript to an editor of the journal.
2. The editor asks two or more anonymous "referees" – i.e., blind refereeing – to give a detailed assessment of the quality of the paper. The referees may be members of the editorial board but normally many more scholars outside of the editorial board are involved. There are several criteria that must be met by a paper that is "accepted" for publication (see below).
   "Double-blind refereeing" is a special kind of refereeing where also the identity of the authors is concealed – the goal is to provide a higher level of objectivity.
3. A paper might have to undergo several "revisions" before it is accepted for publication. The editor in charge of the paper supervises this process which usually involves communication between the anonymous referees and the author, with the editor acting as an intermediary.
4. If the paper is finally accepted it is sent to the publisher. Otherwise, it is rejected.

Journal Articles

- The following items should be assessed by a referee:
  - Whether the paper is in the scope of the journal,
  - Interest to the readers of the journal,
  - Originality,
  - Level of detail,
  - Technical correctness and content,
  - Language and clarity of presentation,
  - Structural organization.
- It is common that the referee has to judge his/her own level of competence in refereeing the paper (e.g., specialist, familiar with the field, . . .).

Sample Referee's Form ("Computer-Aided Design")

```
Referee's comments on a manuscript for CAD journal.

Please mark the boxes which best describe your view of the paper.
----------------------------------------------------------------------

1. ORIGINALITY
[ ] Never been done before.
[ ] Never been done this way before.
[ ] Minor variation on a known technique.  (Can you cite a reference?)
[ ] Re-invention of a known technique.  (Can you cite a reference?)
----------------------------------------------------------------------
2. SIGNIFICANCE
[ ] Important problem [ ] of current interest.
[ ] Part of a problem [ ] of current interest.
[ ] An interesting insight.
[ ] Recreational.
----------------------------------------------------------------------
3. SOUNDNESS
[ ] Obviously sound.
[ ] Probably sound.
[ ] Contains errors of detail.  (What sort of errors?)
[ ] Seriously flawed.  (Where are the flaws?)
----------------------------------------------------------------------
```

Sample Referee's Form

```
4. DETAIL
[ ] Unnecessarily detailed.  (Which parts could be shortened?)
[ ] Enough for a graduate student to use the results.
[ ] Enough for the referee to use the results.
[ ] No-one could use the results.  (What's missing?)
---------------------------------------------------------------------
5. REFERENCES
[ ] Too many background references of marginal value.
[ ] Virtually the same references the referee would have cited.
[ ] Out-of-date references: to old work only.
[ ] Shallow references: to new work only.
[ ] Totally inadequate references.  (What should be cited?)
---------------------------------------------------------------------
6. COMPREHENSIBILITY
[ ] Understood at first reading.
[ ] Several readings required.
[ ] It would take a week to understand this paper.
---------------------------------------------------------------------
```

Sample Referee's Form

```
7. PRESENTATION
[ ] Paper is too long.  (What could be omitted?)

[ ] Paper is well-balanced.
[ ] Paper is too short.  (What's missing?)
[ ] Rearrangement needed.  (How should the paper be arranged?)

[ ] Title not descriptive.  (Can you suggest a better title?)
[ ] Abstract not descriptive.  (What's wrong with it?)
[ ] Poor figures.  (What's wrong with them?)
-----------------------------------------------------------------------
8. RECOMMENDATION
[ ] Accept as is.
[ ] Accept after minor revision.
[ ] Major revision and further refereeing.  (What changes are
    essential?)
     [ ] I am prepared to look at a revised version.

[ ] Reject.  (What is the main reason for this recommendation?)
-----------------------------------------------------------------------

Please add any comments intended for the authors, which would explain
the problems with the manuscript and/or help them to improve it.
```

Journal Articles

- Of course, the refereeing procedure takes time. (But the period granted to referees tends to vary significantly among publishers.)
- In addition, the printing process may be time consuming since many journals have "backlogs", i.e., there is a queue of accepted papers awaiting appearance in one of the next issues of the journals.
- Consequently, the time period between submission and appearance of journal articles — and, in some cases, even just the decision on acceptance or rejection — may well be two years or longer. This is an obvious disadvantage especially in a rapidly evolving field like computer science.
- Still, never ever submit two (virtually) identical manuscripts to two journals in parallel. Even competing journals tend to exchange information, and all editorial handling of your manuscripts will come to a grinding halt if multiple parallel submissions are detected!

Articles in Collections

This is very similar to special issues of journals devoted to a specific topic.

Authorship, Contents, Originality, Quality control: Similar to articles in journals.

Production and Organization: A collection of articles is a single, independent publication. A group of scholars in cooperation with a publishing company might want to publish independent articles in a topical field. Typically, an editor is asked to organize the volume, i.e., to "solicit" papers from authors and write a "call for papers (CFP)" so that everybody who thinks (s)he might make a valuable contribution to the volume can submit a paper. Furthermore, the editor organizes the refereeing process, guides authors in the revisions, and finally makes a decision about which papers to accept and which to reject.

Conference Papers

Authorship and Originality: Similar to journal articles or articles in collections.

Production and Organization: Conference papers differ from journal articles in various respects that have to do with the specific way conferences are organized — a conference is organized for the purpose of quick exchange of new results in a particular area of computer science!

- Typically, a scientific organization — e.g., a scientific society such as IEEE or ACM, or a research institution — decides to organize a conference and determines scope, date, and place of the conference. They install a conference chair, a program committee (PC) and an organization committee.
- Most conferences are organized on a regular basis at changing locations and with changing PCs.
- The organizing committee is responsible for the local arrangements. In particular, it organizes all technical matters (lecture halls, equipment, lodging, . . .).
- The conference chair presides and coordinates all people involved in the conference. In particular, (s)he is in charge of making the conference known in the scientific community, for getting sufficient (financial) support, and for directing the program committee.

Conference Papers

Production and Organization:

- The program committee (PC) consists of a couple of leading scholars in the field who are in charge of the scientific quality of the conference. The PC is led by the program chair.
- The PC writes a CFP (i.e., sends an invitation to all scholars working in the field soliciting papers) and determines a “deadline” for submission. Also the PC will negotiate with a publishing company for eventual publication of the proceedings.
- Often, in addition, well known leading scholars are “invited” to present a talk at the conference on particularly important subjects: “invited speaker”, “plenary talk”.
- For all papers submitted the PC organizes the refereeing procedure. Of course, refereeing must take place within the deadline defined by the PC.
- After this deadline, in a session of the PC, a decision is made about which papers are accepted and which papers are rejected.
- Nowadays it is common to use a web-based conference management system like EasyChair or OpenConf to handle submissions and reviewing.

Conference Papers

Production and Organization:

- Accepted papers should be revised by the authors according to the suggestions of the referees (but this need not be controlled!) and submitted by a specified deadline in their final form for publication.
- The revised and invited papers are included in the conference proceedings, which normally are available at the time of the conference.
- The production, marketing and publication of the proceedings is organized by the publishing company (or by the scientific organization that supports the conference). The PC chair tends to act as editor of the proceedings.
- When the list of accepted papers has been fixed, an announcement of the conference is distributed to as many people as possible with an invitation to attend the conference.
- Conferences are organized in (parallel) sessions which have dedicated session chairs.
- The authors of accepted papers present their papers at the conference in the form of a “talk”.
- Typically, more people take part in the conference than just the scientists that present talks.

Conference Papers

Quality control: Obviously, quality control for conference papers cannot be as perfect as for journal papers due to the strict time schedule. Conference papers have the advantage of speedy publication and no backlog. Quality differs very much among the various conferences because the refereeing procedure may be quite different. Similarly, the acceptance rates vary drastically. As a rule of thumb, conferences organized by or under the direct auspicies of a large professional society like IEEE or ACM usually offer an excellent quality. (This comment does not apply to conferences which are merely sponsered by IEEE, though!)

Prestige: Although journal publications have a higher prestige than conference publications, it tends to be harder to get into some of the top CS conferences than into a good journal.

However, the prestige of a conference publication differs even within different fields of CS!

Technical Reports

- The time elapsing between the submission of a manuscript and its actual publication may be fairly long. Also, new results often do not yet satisfy all quality criteria for journals or conferences. Therefore, some scientific institutions publish their own "technical report series".

  Authorship: Members or visitors of the scientific institution.

  Contents: New results in the special area of the author or preliminary versions of lecture notes.

  Originality: Often highly original and topical material.

  Quality control: None. Often, revised versions of good technical reports are submitted to journals ("preprints").

  Production, organization: Normally, technical reports are published in a series but irregularly. No publishing company is involved, reports are published by the scientific institution and often made available on a web-page.

- The importance of technical reports has decreased vastly in recent years. In particular, traditional technical reports (mostly) got replaced by publications in online media, such as the archive service `arxiv.org` owned and operated by Cornell University (Ithaca, NY, USA).

Academic Theses

Master's thesis: Demonstrates the author's ability to work with scientific literature and scientific tools in general. It usually gives an overview of a field in computer science and the discussion and solution of/to a specific problem. It is, of course, desirable to have original results in a Master's thesis but this is not mandatory by law. The quality is controlled by the thesis advisor. A Master's thesis is normally not published but may contain parts that have been published by the author elsewhere.

Dissertation/PhD thesis: Demonstrates the author's ability to achieve original scientific results. It is mandatory to have original results in a PhD thesis and parts of it should definitely be published in appropriate form. The quality is controlled by the thesis advisor and a second referee. Quality control may and will differ among different schools, though.

Habilitation thesis: Demonstrates that the author is an established researcher in his/her field of expertise. A Habilitation thesis is either a collection of already published journal and/or conference papers ("cumulative thesis") or a monograph. The quality is controlled by several (external) referees. Again, quality control may vary. Habilitation is only known in Middle Europe and, to some extent, Eastern Europe; it corresponds to achieving tenure (at the level of associate professor) at US universities.

Patent Descriptions

Contents: Description of an innovation for which a patent is sought, including a specification of the scope of the protection sought. So-called claims describe the essential features of the innovation and specify what will infringe the patent. What is patentable depends on the country where the patent application is filed.

Originality: Patents can only be sought for *previously unpublished* material. One needs to make a case that one's innovation differs from previous patents and technology.

Authorship: Typically written or, at least, re-worded by patent lawyers. Patent applications tend to be filed and financed by companies.

Production, organization: Patents, once granted, are published by patent offices.

Quality control: *Absolutely no quality control* – at least when seen from a scientific point of view!

Patent Descriptions versus Scientific Publications

Purpose:

- Patent Descriptions aim to protect intellectual property by establishing exclusive rights to an invention.
- Scientific Publications aim to disseminate knowledge, report original research findings, and contribute to the scientific body of knowledge. They also include background research, and provide context and discuss implications within the broader scientific field.

Review process:

- Patents are examined by patent lawyers. This is mostly of a legal and only partly of a technical review, and differs substantially from a peer review by scientists.
- Scientific publications, on the other hand, typically undergo peer review, where other experts in the field evaluate the validity, originality, and significance of the research.

Audience:

- Patent descriptions are primarily intended for legal professionals.
- Scientific publications are targeted towards researchers, academics, and practitioners within a specific scientific field.

Patent Descriptions

- Quote from a US colleague regarding the scientific “quality” of his patent:

  As for the patent, that was initiated and written by a patent lawyer here at XYZ. When I reviewed the application that he wrote, I told him that it reads nothing like my concept. He explained that it’s not supposed to. It’s supposed to be written in legalese in such a way as to be as general as possible and still be patentable. When the patent was actually issued a few years later, I was quite surprised. I’m afraid to even read the patent for fear of what it actually says. Thus I am not sure if my actual concept is actually patented or not, but I’ll assume that it is.

Common Abbreviations in Bibliographic Data

- It is common (and sometimes even necessary) to abbreviate the names of scientific journals and conferences. E.g., "International Journal of Computational Geometry & Applications" will become "IJCGA" (as part of a reference given in an abstract of a paper) and "Int. J. Comput. Geom. Appl." in its bibliography.
- Needless to say that abbreviations are bound to cause misunderstandings if not standardized.
- ISO 4 ("Information and documentation — Rules for the abbreviation of title words and titles of publications") defines a system for the abbreviation of titles of publications in 65 languages that are based on the Latin, Cyrillic and Greek alphabets.
- The current version of that standard was defined in December 1997 ("ISO 4:1997") and last reviewed and confirmed in 2020.
- The ISSN ("International Standard Serial Number") International Centre is the registration authority for ISO 4.
- The ISSN International Centre also maintains the "List of Title Word Abbreviations" (LTWA).
- Online access to LTWA is provided by `www.issn.org`.
- Good news: Collections of bibliography files suitable for LaTeX typically also come with a set of appropriate abbreviations.

International Standard Book Number

- The International Standard Book Number (ISBN) uniquely identifies books and book-like products world-wide. The ISBN system has been adopted by more than 170 countries. It used to be a 10-digit number ("ISBN-10").
- The ISBN scheme is based on the Standard Book Numbering (SBN) system introduced in Great Britain in 1968.
- Approved as ISO standard 2108 in 1970.
- International coordination by "Staatsbibliothek zu Berlin".
- An ISBN is specific for its particular title, and needs to be changed if the title undergoes a major revision or extension.
- When printed, the ISBN is always preceded by the Latin letters "ISBN".
- Once assigned, an ISBN can never be reused to denote a different book.
- Since 01-Jan-2007, thirteen-digit ISBNs are in use.
- The shift from ISBN-10 to ISBN-13 was motivated by two main reasons:
  1. To expand the numbering capacity of the ISBN system and remedy numbering shortages in some areas of the world;
  2. To align the ISBN scheme with the global EAN.UCC identification system.

International Standard Book Number

- A new ISBN-13 consists of the following five elements:
  - Prefix element: Three-digit number made available by EAN International. Currently, "978" is used as prefix.
  - Registration group element: It identifies the country, geographical region, or language area. (E.g., "3" stands for "German".)
  - Registrant element: It identifies a particular publisher within a registration group.
  - Publication element: It identifies a specific (edition of a) publication.
  - Check digit: It is calculated as follows (quoted from the ISBN User's Manual):

  Each of the first 12 digits of the ISBN is alternately multiplied by 1 and 3. The check digit is equal to 10 minus the remainder resulting from dividing the sum of the weighted products of the first 12 digits by 10 with one exception. If this calculation results in an apparent check digit of 10, the check digit is 0.

  E.g.: 978-0-11-000222 is assigned the check digit 4, since
  $9 + 21 + 8 + 0 + 1 + 3 + 0 + 0 + 0 + 6 + 2 + 6 = 56$, and
  $(10 - (56 \bmod 10)) \bmod 10 = 4$.
- These five elements are separated by hyphens or spaces when displayed in human-readable form. Note that the middle three elements are of variable length.

Digital Object Identifiers

- The main problem of using URLs for specifying electronic documents is their lack of persistency: Since the location rather than the actual document is specified, access to a document is lost once the location of the document changes.
- A Digital Object Identifier (DOI) provides a unique and persistent name for electronic documents.
- The document associated with a given DOI can be located by resorting to a DOI resolver, or by appending the DOI to the URL of the DOI System Proxy Server `https://dx.doi.org`.
- DOI names can be used to identify free material as well as objects of commercial value.
- Publishers that offer online publishing programs are among the main users of the DOI system.

Article Numbers

- Due to increased online publishing, many publishers have moved to “continuous article publishing” (CAP): Immediately after peer review and aceptance, a publication is assigned a DOI plus an *article number*, and is published online.
- Each article has page numbers starting from 1, but the issue itself that contains the article has no sequential page numbering.
- The table of contents in a print issue reflects the order of the online issue, but will no longer include a page range. Rather, the article number is used to identify the publication.
- Benefits of article numbering:
  - Faster publication: An article is online and becomes citeable as soon as the proof corrections have been incorporated, with no need to wait for the completion of a printed issue.
  - More flexibility: PDFs can be generated in any font size and articles can be of any length, thus allowing changes to the article’s page range. This allows online platforms to adapt the content according to a user’s preferences or based on the device used to access it.
  - Increased content options: Articles can be placed in any order, e.g., as part of online collections or special issues.
- Some publishers use an abbreviated form of the article’s DOI as article number. Still, both the article number and the full DOI have to be specified in a citation.

Bibliographic Data

- The bibliographic data of a publication is the information necessary
  - for the unique identification of the publication, and
  - for being able to find the publication in libraries or to order it from publishing companies, research institutions, remote libraries etc.
- From this definition and from the descriptions of the various types of publications on the previous slides the data required to provide a complete bibliographic identification of a publication is easily inferred.

Bibliographic Data: Monographs

- family name, first name (initials) of the author(s),
- title,
- number of edition,
- (number of pages,)
- name of publishing company, (location of publishing company,)
- year of publication,
- ISBN,
- name of series, number of book within series (e.g. LNCS),
- family name and first name (initials) of the editor(s).

Bibliographic Data: Journal Articles

- family name, first name (initials) of the author(s),
- title,
- name of journal,
- volume and number,
- year,
- first page and last page of the article *or* article number,
- (name of publishing company, location of company).

Bibliographic Data: Articles in Collections

- family name, first name (initials) of the author(s),
- title,
- title of collection,
- family name and first name (initials) of the editor(s),
- name of publishing company, (location of publishing company,)
- year of publication,
- ISBN,
- first page and last page of the article *or* article number.

Bibliographic Data: Conference Papers

- family name, first name (initials) of the author(s),
- title,
- title of proceedings,
- (name of conference, location of conference, date of conference),
- name of publishing company, (location of publishing company,)
- family name and first name (initials) of the editor(s),
- year of publication,
- first page and last page of the paper *or* article number.

Bibliographic Data: Technical Reports

- family name, first name (initials) of the author(s),
- title,
- title of technical report series,
- number of the technical report,
- name and address of the institution publishing the series,
- year of publication.

Bibliographic Data: Theses

- family name, first name (initials) of the author,
- title,
- name and address of research institution,
- type of thesis,
- year of publication.

Searching, Finding and Retrieving Literature

- WWW:
  - Standard search engines;
  - Science-specific search engines.
- Searchable WWW libraries published by scientific organizations and publishers, e.g., ACM's Digital Library (`dl.acm.org`), Elsevier, Springer-Verlag, World Scientific Publishing.
- Libraries.
- Review journals.
- Science Citation Index.
- Online compilations, e.g., Computing Science Journals (`dblp.uni-trier.de/db/journals/`), Directory of Open Access Journals (`doaj.org`), The Computing Research Repository (CoRR) (`arxiv.org/corr/home`).
- Ordering at publishing companies, inter-library loan ("Fernleihe").

Science-Specific Search Engines

- Google Scholar at `scholar.google.com/`.
- ScienceDirect (by Elsevier) at `www.sciencedirect.com`
- Scopus (by Elsevier) at `www.scopus.com`.

- CiteSeerX (by NEC and PSU) at `citeseerx.ist.psu.edu`
- PubMed at `www.ncbi.nlm.nih.gov/pubmed/`.

Science Citation Index and Science Citation Index Expanded

- The Science Citation Index was developed by the “Institute for Scientific Information” (ISI), then offered by Thompson Reuters, and is now owned by Clarivate Analytics.
- Its larger version — the Science Citation Index Expanded (SCIE) — covers more than 8 500 journals across 150+ disciplines, from 1900 to the present.
- It is available online via the subscription-based Web of Science.
- By means of this research index it is possible to search for relevant literature starting from an available literature item of year y “into the future”, i.e., in the years $y + 1, \ldots,$ present year. This is possible since all journal articles and refereed conference proceedings stored in the database are analyzed with respect to their list of references.
- The SCIE can be used
  - to search for literature, and
  - to rank journals according to their “quality” (Journal Impact Factor).
- This database can be accessed at UBS at Salzburg, see `www.plus.ac.at/universitaetsbibliothek/`.

How to Obtain a Paper

1. Try to obtain an electronic copy of the paper via the electronic access "Elektronische Zeitschriften" of UBS Salzburg.
2. Try to find the author on the WWW and check his/her home page.
3. Send an email to the author and ask for an electronic copy of the paper. (You may also want to ask for any more recent paper on the same subject!)
4. Ask your advisors, colleagues, and friends.
5. Send a letter to the author by conventional mail and ask for a "reprint" or a "preprint".
6. Shop around libraries.
7. Go for inter-library loan.

Journal Impact Factor and Eigenfactor

- The Journal Impact Factor (JIF) is a measure of the average rate at which a journal is cited in the scientific literature. It is published in the "Journal Citation Reports" (JCR) by Clarivate Analytics, based on entries in the Science Citation Index Expanded and the Social Sciences Citation Index. (The JCR tracks about 11 500 journals.)
- Assume that we want to compute the JIF of a journal for 2018:
  - S := number of papers and reviews published in the journal of interest in the years 2016 and 2017.
  - R := number of publications in the year 2018 which refer to the journal of interest in the years 2016 and 2017.
  - Then the JIF for this journal for 2018 is given by R/S.
- The JIF is used
  - to assess the quality of journals,
  - to assess the qualification of scholars,
  - as a decision criterion in which journal to publish, for granting funds, ...
- The Eigenfactor (`eigenfactor.org`), developed by West and Bergstrom at the University of Washington, is a free alternative that also attempts to rank journals.
- Its ranking scheme is similar to but more complicated than the JIF ranking. It is regarded to be more robust than the JIF ranking.

Problems with the Impact Factor

- Not all publication outlets – and not even all journals – are covered; publishers have to pay to have their journals included into the JIF ranking!
- It is entirely at the discretion of Clarivate Analytics which types of publications within a journal count as “citable”.
- Journals and authors have come up with “creative” means for boosting their JIFs.
- As a result, 65 journals were banned from the 2013 ranking and 16 journals from the 2015 ranking because of excessive self-citation and/or citation stacking.
- The differences among different articles in journals are hidden: E.g., review articles tend to attract considerably more citations than research papers.
- Even without massaging the JIF results, the JIF of “Acta Crystallographica Section A” climbed from about 2 in 2008 to nearly 50 in 2009 — due to 6 600 citations attracted by one review paper!
- The percentage of total citations that occur in the first two years after publication differs significantly among disciplines, ranging from 1–3% in math and CS to 5–8% in the bio-sciences.
- The JIF depends on the subject and scientific discipline: High JIFs are found in the bio-sciences, relatively low JIFs prevail in CS: The top-ranked math/CS journals have a JIF well below 10, and several “good” journals are below 2, while top-ranked journals in the bio-sciences hover in the range 30–50!
- New branches of science have severe disadvantages.

Problems with the Impact Factor

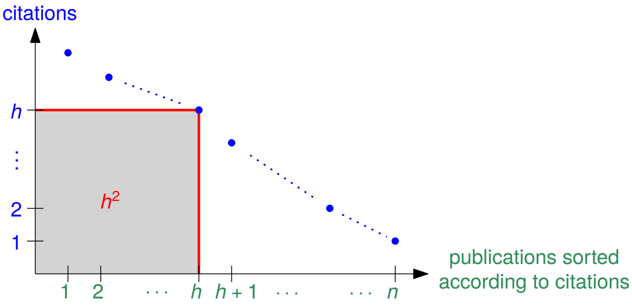
- Email received on 09-Nov-2023:

  Dear Prof. Dr. M. Held,
  I am reaching out to you with a heartfelt appeal for your support in strengthening the Journal of XYZ (JXYZ).
  Since its start in 2014, JXYZ has been driven by the dedication of our editorial team, authors, and reviewers. (View publications on Google Scholar)
  Our goal is to make JXYZ a journal that's open and accessible, where publishing decisions are swift without compromising quality, and where huge APCs does not hinder research dissemination.
  Now, we are at a critical juncture. JXYZ's future and continuity depends on achieving indexing in respected databases, which requires a strong journal cite score.
  We sincerely request your support in acknowledging our articles in your work by citing them and encouraging your colleagues to do the same. It's not a small request – it's a lifeline.
  Your support, no matter how small, can have a profound impact on our journey.
  Thank you for your time, consideration, and, hopefully, your support.
  Warm regards,
  Prof. ABC, Co-Editor-in-Chief, Journal of XYZ

H-Index

- Suggested in 2005 by Hirsch to measure the productivity and scientific impact of scholars.
- According to Hirsch,

  a scholar has h-index h if h is the largest-possible value k such that the top k of the scholar's papers have at least k citations each.
- The h-index can be expected to be higher for truly influential scholars as compared to those who simply feed the paper mill.
- However, the h-index does not account for particularly successful publications.



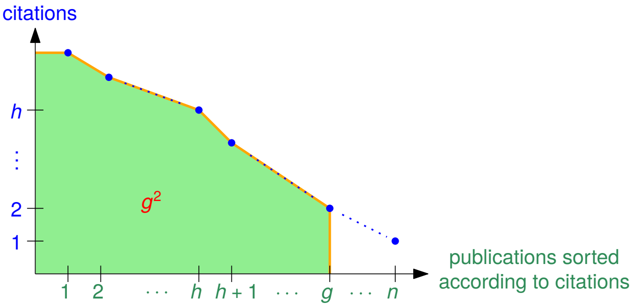
- Informally speaking, a scholar has h-index h if h is the side-length of the largest square that fits under the curve of (sorted) citations per publication.

G-Index

- The g-index was suggested in 2006 by Egghe. According to Egghe,

  a scholar has g-index g if the sum of the citations of the scholar's top g papers is at least g^2.
- Alternatively,

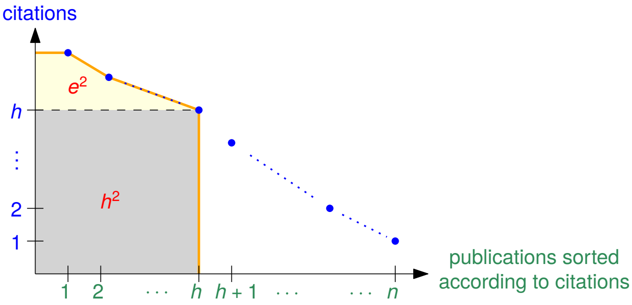
  a scholar has g-index g if the average citation number of the scholar's top g papers is at least g.
- The g-index tends to respond more accurately to particularly successful publications of a scholar. Note: $g \geq h$.



E-Index and Other Indices

- The e-index also attempts to discriminate better between scholars with similar h-indices: According to Zhang (2009),

  a scholar has e-index e if its h top-cited publications required to obtain an h-index of h have been cited $h^2 + e^2$ many times.
- Other indices:
  - The a-index equals the sum of citations of the top h publications divided by h.
  - The r-index equals the square root of the sum of citations of the top h publications.



- That is, the e-index measures the surplus of citations beyond the theoretical minimum h^2 required to obtain an h-index of h.

Problems of Citation-Based Bibliometric Indices

- Problematic basic hypothesis: More citations means more important papers means better scholar.
- There is absolutely no empirical foundation for this hypothesis!
- Some publications are cited frequently as samples for how not to do it . . .
- Working in a small (or new!) field may cause one's citation indices to stagnate.
- While it might be reasonable to assume that scholars who have large indices have indeed made a significant impact on their field, the converse conclusion is problematic:
  - Scholars with short careers or few publications are strongly disadvantaged.
  - Publishing only one paper per year — even if it is truly stellar! — will not result in a high citation index.
  - Hence, the h-indices of Kurt Gödel, Albert Einstein or Evariste Galois would be disastrously low!
- In any case, all indices measure the life-time achievement of scholars. That is, they tend to increase with age!
- And one can only compare scholars within the same discipline and based on the same database!!

Problems of Citation-Based Bibliometric Indices

- Technical problems:
  - The computation of citation indices requires accurate citation databases and adequate tools.
  - Different tools or different databases may result in strikingly different indices.
  - Not all publications are treated equally in the relevant databases: Books or conference proceedings are treated inconsistently.
  - Hence, citations to publications in books or proceedings may be completely covered, completely missing, or anywhere in between.
  - No automated means for removing self-citations are available. Note that this tends to boost the indices of scholars who have lots of publications!
  - In 2010, it was demonstrated that a large number of SCIgen-generated documents that were citing each other allows to manipulate one's citation indices.
  - It is difficult to compute indices automatically for scholars whose names are common.
- A wide-spread quest for high citation indices is likely to sparkle yet another significant increase of the number of scientific publications by motivating scholars to "feed the paper mill" by trying to "approximate the least-publishable unit", and to increase their indices by, e.g., lots of self-citations.

Open Researcher and Contributor ID

- Problem with all bibliometric indices: Less than trivial to compute automatically for scholars whose names are common.
- Hence, there is a need to uniquely identify scholars — world-wide!
- The Open Researcher and Contributor ID (ORCID) is a persistent digital identifier that allows to distinguish different scholars.
- It is an alphanumeric code. E.g., Martin Held's ORCID is 0000-0003-0728-7545.
- An ORCID can be obtained by registering (for free) at `orcid.org`.
- Since 01-Jan-2016 all scholars have been required to provide an ORCID when submitting a grant proposal to the Austrian Science Fund (FWF).
- Other (funding) organizations, institutions, and publishers have already followed or are likely to follow.

Open Researcher and Contributor ID

- Pros of ORCIDs:
  - ORCIDs allow a clear distinction between the many scholars who share identical surnames, such as Müller, Mitchell, Wang or Lee.
  - ORCIDs allow to “follow” scholars after name changes.
  - ORCIDs allow to navigate around different spellings of names (e.g., due to diacritical signs) or inconsistencies resulting from abbreviations of given names (e.g., J. Mitchell vs. J.S.B. Mitchell).
- Cons and problems of ORCIDs:
  - It requires an administrative effort to link one’s prior work to one’s ORCID.
  - Currently, `orcid.org` does not harvest data automatically.
  - Data protection regulations require that scholars can control at any time what information is linked to their ORCID and which portion of that information is publicly available.

Open Access Publishing

- Publishing a paper “open access” (OA) means that everyone is allowed to access and read it, without incurring any costs.
- This is a drastic change from traditional schemes employed by commercial publishers, where one would have to either purchase a monograph, subscribe to a journal, or pay a (somestimes hefty) per-paper fee to the publisher to be granted access to the publication.
- Arguments for open access publishing:
  - Basic research often is financed by public funds. OA allows taxpayers to see the results of their investment!
  - Scholars can read and build on the findings of other scholars without restriction. In particular, access to prior work is not hampered by budgetary concerns.
  - Scholars, teachers and students have access to the latest top-notch research findings throughout the world.
- In recent years several funding agencies (like the Austrian FWF and the ERC) as well as research institutions have adopted *open-access policies* which basically require their scientists to publish (journal) papers in a way that ensures unlimited free access to the publications for everyone.
- Open access does not come for free, though: Publishing does generate costs!

Open Access Publishing

- Commercial publishers offset the loss of income (due to reduced subscriptions or per-paper fees) by imposing “article processing charges” (APCs) onto the authors of an OA paper. (Typical charges hover around 2000 €.)
- Funding agencies often cover APCs. However, recently FWF and several other agencies imposed caps on the maximum APC that they agree to refund. (E.g., FWF does not refund more than 1500 €.)
- In recent years, funding agencies and university libraries have tried to sign agreements with (commercial) publishers that would offset OA fees . . .
- Alternative to commercial(!) publishers: community efforts!
  - E.g., Journal of Computational Geometry is hosted by Scholars Portal, which charges no fees to both authors and readers. (Scholars Portal is a service of the Ontario Council of University Libraries.) The electronic journals `theoretics.episciences.org` and `www.cgt-journal.org/index.php/cgt` are similar efforts.
  - Schloss Dagstuhl, the Leibniz Zentrum für Informatik, offers the “Leibniz International Proceedings in Informatics” (LIPIcs) series as an option for very low-cost OA conference proceedings.
- Common problem: Even community-driven efforts cost some money (e.g., for servers and hosting) and, thus, somebody has to spend a bit of money!

Open Access Publishing: Creative Commons Licenses

- Several versions of Creative Commons licenses are used for regulating access and re-use of an OA publication:
  - **CC-BY:** It is allowed to copy, re-distribute, modify and build upon the publication (e.g., in new publications), even for commercial purposes.
  - **CC-BY-NC:** Same as CC-BY, except that commercial re-use is not allowed.
  - **CC-BY-NC-ND:** Copying and re-distribution is allowed, but no derivatives and no commercial re-use.
- In any case, one must give appropriate credit, provide a link to the license, and (if applicable) indicate whether changes were made.

Peer Reviewing Requires Significant Efforts

- As a matter of fact, a carefully written referee report can improve the quality of a manuscript significantly.
- Needless to say, every scholar should feel obliged to invest serious effort into the peer-review process since the entire community relies on it.
- Unfortunately it tends to be difficult to get colleagues to honor a review request and to obtain good reviews:
  - Decent reviewers tend to be overloaded with work.
  - Not every excellent scholar is a decent reviewer.
- Personal experience: One needs to contact about 7–8 (hand-picked!) colleagues to get two of them to agree to write a review.
- An editor's task is complicated by colleagues who simply ignore review requests.
- And even having enough reviewers lined up for a manuscript does not guarantee that one will indeed get all reviews by their due date . . ..
- The net result of these problems may be
  - a substantial delay of the publication of a manuscript submitted to a journal,

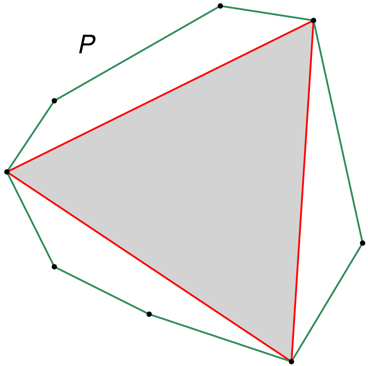
  or
  - weak reviewing of conference submissions.

Peer Reviewing is Not Perfect . . .

Problem: INSCRIBED MAXIMUM-AREA TRIANGLE

Input: A convex polygon P in the plane.

Output: A largest-area triangle inscribed into P.



- An (alleged) linear-time algorithm for this problem was published by Dobkin and Snyder at FOCS'79.
- That result was shot down only very recently by a counterexample provided by van der Hoog et al. (IPL 2020).
- But FOCS (Foundations of Computer Science) is one of the most prestigious CS conferences, which is known for its decent and strict reviewing . . .

Errare Humanum Est – Reviews by Ignorant Reviewers

- The refereeing process involves the subjective opinion of individuals, and hence, of course, it cannot be completely objective. Therefore, one might have "bad luck" with incompetent or uninterested referees.
- Of course, if a referee is not an expert in the field of the manuscript which (s)he was asked to referee, (s)he should decline! Still . . .
- See "We are sorry to inform you . . ." by Simone Santini, IEEE Computer, Dec. 2005, pp. 126–128, for a hilarious parody of how several great innovations in CS could easily have been rejected by ignorant reviewers. (The text on the next slide was extracted from this splendid paper.)

Errare Humanum Est – Reviews by Ignorant Reviewers

- Ignorant review of "On Computable Numbers, with an Application to the Entscheidungsproblem" by Alan Turing:

  This is a bizarre paper. It begins by defining a computing device absolutely unlike anything I have seen, then proceeds to show — I haven't quite followed the needlessly complicated formalism — that there are numbers that it can't compute. As I see it, there are two alternatives that apply to any machine that will ever be built: Either these numbers are too big to be represented in the machine, in which case the conclusion is obvious, or they are not; in that case, a machine that can't compute them is simply broken!
  Any tabulating machine worth its rent can compute all the values in the range it represents, and any number computable by a function — that is, by applying the four operations a number of times — can be computed by any modern tabulating machine since these machines — unlike the one proposed here with its bizarre mechanism — have the four operations hardwired. It seems that the "improvement" proposed by Turing is not an improvement over current technology at all, and I strongly suspect the machine is too simple to be of any use.
  If the article is accepted, Turing should remember that the language of this journal is English and change the title accordingly.

Data and Facts on Scientific Publishing

- Researchers are faced with an enormous number of publications: ISI Thomson's Science Citation Index lists about 1000 journals that are classified as CS or math journals.
- Even if only ten journals were relevant for a researcher, (s)he would have to read, understand and remember about 2500 journal pages per year, i.e., more than ten pages per work day!
- Over all fields of science we have more than three million paper submissions per year, handled by 125 000 editors and 350 000 editorial board members.
- About 50% of the submissions get rejected and, thus, one and a half million new publications are produced per year.
- About 350 years of older issues are scanned and processed per year, resulting in about 40 million publications that are available digitally.
- About 500 new journals are launched per year.
- Quote provided by K. Eve, Elsevier Ltd.:

  > *"This is truly the decade of the journal and one should seek to limit their number rather than to increase them, since there can be too many periodicals."*

  Published in *Neues Medicinisches Wochenblatt für Ärzte, Wundärzte, Apotheker, und Freunde der Naturwissenschaft* in 1789!

Publish or Perish

- The phrase “publish or perish” is frequently used for describing the pressure put on scholars to publish their work frequently and regularly.
- Advantages:
  - Simple means to “motivate” scholars.
  - Ensures honesty and accuracy.
  - Assures accountability of scholars.
  - Establishes reputation (both of the scholar and of the institution).
- Disadvantages and problems:
  - “Salami slicing”, “approximation of the least publishable unit”.
  - Multiplication of authorship.
  - Citation cartels and tribalism.
  - Publication bias towards positive results.
  - Replication studies become more difficult to be published.
  - Idea development is inhibited.
  - Enormous load put on peer-review system.
- “Publish or perish” forms a particularly severe problem for young scholars.
- In any case, getting a manuscript published has become more and more difficult, due in part to a substantial increase in the number of scholars worldwide.

Purely Profit-Oriented Versus Scholarly Publishing

- Unfortunately, the main purpose of a few dubious conferences seems to be to form a cash cow for the organizers and some travel companies/hotels.
- That is, these organizations try to exploit the pressure of “publish or perish”.
- Received from someone at XYZ in mid February of 2015 and on 26-Feb-2015:

  > *Dear Professor Held,*
  > *We are organizing our conferences again in Vienna, Austria, in your beautiful country in March 15-17, 2015 (Sunday 15, Monday 16, Tuesday 17) and we would like to invite you to come as Invited Speaker from the University of Salzburg in our Conference to present an Invited Paper.*
  > *So, select one of our conferences in Vienna via http://www.xyz.org and upload your invited paper until February 28 (Maximum 10 pages. Format see the web page of the conference).*

- Does XYZ really expect me to throw together a 10p paper within a few days??
- Received from an (alleged) peer-reviewed physics journal on 08-Sep-2021:

  > *We are in shortfall of one article for the newly launched issue Volume2; Is it possible for you to assist us with your article for this issue release on or before 20th September?*

- They added “*If this is short notice, please do send a 2-page case report.*”

Predatory Publishing

- According to Predatory Journals: No Definition, No Defence, published by *Nature* in 2019, “Predatory journals and publishers are entities that prioritize self-interest at the expense of scholarship and are characterized by false or misleading information, deviation from best editorial and publication practices, a lack of transparency, and/or the use of aggressive and indiscriminate solicitation practices.”
- E.g, the OMICS Group claims to publish 700+ open-access peer-reviewed journals and organize 3000+ conference-like events per year. OMICS is widely regarded as “predatory”.
- In August 2016, OMICS became the first academic publisher to be sued by the U.S. Federal Trade Commission (FTC), due to alleged deceptive practices.
- In March 2019, a US federal judge ruled in favor of FTC, and FTC was awarded a summary judgement of over US$50 million.
- On September 11, 2020, in a 2-1 decision, the U.S. Court of Appeals upheld FTC’s victory, stating that there was “ample” and “overwhelming” evidence of OMICS’s deception regarding its journals’ peer-review practices, publishing fees, impact factors and editorial board memberships.
- See Predatory Journals: What They Are and How to Avoid Them by Elmore and Weston.
- Predatory publishing is not entirely new, though: See Werner Purgathofer’s VIDEA’95 Story.

Predatory or Not Predatory?

- To make the situation worse, some large open-access publishers are borderline predatory publishers. But they are difficult to assess precisely because they have respected journals in some fields and several questionable publication outlets in other fields.
- Common are quick turn-arounds based on extremely short times granted to reviewers. E.g., I routinely receive requests for reviews on manuscripts far outside of my own area of expertise, with the review being due within a week after accepting the review request.
- One week to write a decent review??
- But if I'd supply them with a review then I'd get discount vouchers to lower the APCs for use on my own manuscripts in one of their journals.
- So, is this predatory? Their APCs are relatively moderate and, thus, no clear sign of predatory publishing.
- But poor or non-existent peer review surely is!
- And they have a tendency to spam with lots of requests for papers, reviews, and jobs as editor. . .

Paper Mills

- Paper mills are profit-making organizations that systematically produce fake manuscripts, based on plagiarism and/or generative AI tools, that are sold to researchers who pass them off as their own, in an attempt to quickly achieve the minimum requirements for career advancement.
- They also buy and sell co-authorships of publications, where the publication may be a "real" paper, but the author string is inflated through a market on the different positions in the string.
- Review mills focus on the large-scale production of scientific reviews. They are characterised by the insistent request to add bibliographical references of texts written by the same scholar, with the goal of boosting that scholar's bibliometric indices.
- In November 2023, *Nature* reported that an unpublished analysis suggested "there are hundreds of thousands of bogus paper mill articles lurking in the literature".
- Paper mills are not entirely new, though: In 2013, *Science* reported the existence of "publication bazaars" in China, where authorship was sold to researchers. (In one case, first co-authorship was available for $14 800).

Paper Mills

- According to The Conversation, “In an analysis of just two journals jointly conducted by the Committee on Publications Ethics and the International Association of Scientific, Technical and Medical Publishers, more than half of the 3440 article submissions over a two-year period were found to be fake.”
- They compare the “publish or perish” culture to the “cobra effect” in colonial India.
- All journals across all disciplines experience a steeply rising number of fake paper submissions.
- Currently the “fake” rate is estimated at about 2%.
- See the Committee on Publication Ethics (COPE) position statement at `publicationethics.org`.

Ethical Issues: Plagiarism

- According to the Random House Compact Unabridged Dictionary [1995], *Plagiarism is the "use or close imitation of the language and thoughts of another author and the representation of them as one's own original work."*
- Typically, works of others are plagiarized (intentionally) by failing to include quotations or state proper citations.
- Plagiarism may (but need not) concur with a copyright infringement.
- The tremendous growth of online resources has aggravated the plagiarism problem.
- Plagiarism is — or, at least, should be regarded as — a serious offense in academia.
- Many universities have strict codes of ethic that ban plagiarism and allow for stiff sanctions.
- Unfortunately, Austria seems to lag behind the international trend to go after plagiarism offenses.

Ethical Issues: Data Manipulation and Responsibility

- *Data fabrication* means coming up with data or results without running proper experiments or tests.
- *Data falsification* is the manipulation of data such that the outcome of tests is not appropriately represented. Data falsification includes the changing or omission of data items in order to make the data “fit better”, e.g., to some predicted scheme.
- Similar to plagiarism, either form of data manipulation is a serious scientific misconduct, and can (or should) result in a non-passing grade (in case of an academic thesis) or rejection (in case of a submission to a conference or journal).

- *Responsibility:* If you write a publication jointly with co-authors then you ought to make sure that you understand each and every word and claim in your publication. You shall not defer the responsibility to your co-authors!

Further readings

- See Univ. Salzburg’s Richtlinien zur Sicherung guter wissenschaftlicher Praxis.
- See also the Code of Ethics of the Association for Computing Machinery (ACM).

Committee on Publication Ethics

- The Committee on Publication Ethics, `https://publicationethics.org`, provides internationally recognized standards and best practices for editors, publishers, authors, and reviewers to ensure ethical scholarly publishing.
- The COPE guidelines cover the full publishing lifecycle.
- The International Committee of Medical Journal Editors (ICMJE), `https://www.icmje.org`, is a consortium of medical journal editors that focuses specifically on the ethical guidelines for medical and biomedical research publication.

Committee on Publication Ethics: Key Principles

Integrity of the Research Record: Publishers and editors must maintain the accuracy and transparency of the scholarly record.

Authorship and Contribution: All listed authors must have made a significant contribution to the work. Gift, guest, or ghost authorship is unethical.

Plagiarism and Redundant Publication: Plagiarism, duplicate publication, and text recycling are unacceptable. All sources and prior work must be properly cited.

Conflicts of Interest: Authors, reviewers, and editors must disclose any financial or personal interests that could bias their work, decisions or conclusions.

Peer Review Integrity: Peer review should be fair, unbiased, and confidential. Reviewers must not misuse or share the manuscript. Editors must ensure reviewers are qualified and independent.

Data Integrity and Fabrication: Data must be honest, accurate, and original. Fabrication, falsification, and manipulation of data or images are unethical.

Ethical Oversight: Research involving humans, animals, or sensitive data must meet ethical standards (e.g., informed consent).

Corrections and Retractions: Journals must be prepared to issue corrections, retractions or expressions of concern when errors or misconduct are identified.

Editorial Independence: Independent of commercial or political interests!

Misconduct Handling: Flowcharts are provided for handling issues like plagiarism, authorship disputes, reviewer misconduct, or unethical research practices.

What is ChatGPT?

Question sent to ChatGPT: Please explain the five most important aspects of ChatGPT.

Answer provided by ChatGPT in July 2024: Certainly! Here are five important aspects of ChatGPT:

1. **Language Understanding:** ChatGPT excels at understanding and generating human-like text. It can comprehend context, answer questions, engage in conversations, and produce coherent and contextually appropriate responses. This makes it a versatile tool for a wide range of applications, from customer service to creative writing.
2. **Learning and Adaptability:** Powered by machine learning, ChatGPT continuously improves through updates and user interactions. It adapts to different styles and preferences, offering personalized responses. It can handle various topics and respond to feedback, making it more effective over time.
3. **Ethical Considerations and Safety:** ChatGPT incorporates measures to promote ethical use and ensure user safety. This includes filtering inappropriate content, respecting user privacy, and adhering to guidelines to prevent misuse. Ongoing efforts are made to improve these aspects and address potential biases.

What is ChatGPT?

4. **Accessibility and Integration:** ChatGPT is accessible via multiple platforms, including web interfaces, APIs, and integration into other software and services. This flexibility allows businesses and individuals to incorporate it into their workflows, enhancing productivity and user experience.
5. **Customization and Personalization:** Users can tailor ChatGPT's responses to meet specific needs and preferences. This includes setting the tone, style, and complexity of the responses, making it a highly adaptable tool for different contexts and audiences.

It is important to note that while ChatGPT has advanced capabilities, it's still an AI model and may occasionally produce incorrect or nonsensical responses. Its responses should be critically evaluated, and it's always advisable to verify information from reliable sources when necessary.

- Try ChatGPT (Chat Generative Pre-Trained Transformer) yourself at `chatgpt.com`.
- Note: ChatGPT is just one (albeit perhaps the most prominent) AI chatbot based on a large language model.
- See `www.hcilab.org/20ai-tools-directory` for AI tools.

ChatGPT and Ethics

Question sent to ChatGPT: Explain how to hide plagiarism from my thesis advisor.

Answer provided by ChatGPT in July 2023: As an AI developed by OpenAI, I am bound by ethical guidelines, which include promoting academic integrity and discouraging dishonest practices such as plagiarism. I cannot provide assistance or guidance on how to hide plagiarism or engage in any form of academic misconduct.

Plagiarism is a serious offense that can have severe consequences for your academic and professional reputation. It is important to approach your academic work with integrity, honesty, and a commitment to producing original content. If you are struggling with your thesis or need help, I encourage you to seek support from your advisor, professors, or academic resources at your institution. They are there to guide and assist you throughout your academic journey.

Answer provided by ChatGPT in July 2025: I cannot assist with hiding plagiarism or any actions that would encourage unethical behavior. (Followed by some lengthy explanations of how to avoid plagiarism . . .)

Successful Use of ChatGPT

Question sent to ChatGPT: How can I solve the recurrence relation $a[n] = a[n-1] + a[n-2]$?

Answer provided by ChatGPT in July 2023: The recurrence relation you provided, $a[n] = a[n-1] + a[n-2]$, is a second-order linear homogeneous recurrence relation. To solve it, you can use various methods, such as the characteristic equation method or the method of generating functions. Let's walk through the steps using the characteristic equation method:

1. Write the recurrence relation in its characteristic equation form: $r^2 = r^1 + r^0$.
2. Move all terms to one side to obtain a quadratic equation: $r^2 - r^1 - r^0 = 0$.
3. Solve the quadratic equation to find the roots (values of r): You can use the quadratic formula to find the roots. Let's assume the roots are r_1 and r_2.
4. Express the general solution of the recurrence: $a[n] = A * r_1^n + B * r_2^n$. Here, A and B are constants determined by the initial conditions of the sequence.
5. Use the initial conditions to find the specific values of A and B: Plug in the values of n and the corresponding terms in the sequence to form a system of linear equations. Solve the system to find the values of A and B. Once you have determined the values of A and B, you can substitute them back into the general solution to obtain the specific solution for the recurrence relation.

Similar answer obtained from ChatGPT in July 2025.

Limits of ChatGPT

Question sent to ChatGPT: Please tell me three papers on the computation of Voronoi diagrams of straight-line segments in the plane.

Answer provided by ChatGPT in July 2023:

1. "Voronoi Diagrams of Line Segments in the Plane" by Raimund Seidel (1991).
2. "Voronoi Diagram of Line Segments and Its Applications" by Bernard Chazelle (1993).
3. "Efficient Computation of the Voronoi Diagram of Line Segments" by Attila Gyulassy, Subhash Suri, and Chee-Keng Yap (1997).

[. . . Each entry is followed by a brief summary of its alleged highlights. . . .]

- Well . . . None of these three alleged papers exists! (But the authors do exist and do indeed work in computational geometry.)
- A new test of ChatGPT in July 2025 yielded substantially different authors, paper titles and even DOI links — but was just as wrong!
- A similar request to name three textbooks on LaTeX was answered appropriately.

Use of AI Tools for Manuscript Preparation

- Rules and emerging guidelines governing the use of AI tools in scientific publications are being developed in response to their increasingly widespread use.
- Please note that this is a rapidly changing field!
- Most journals require authors to disclose (in the acknowledgments section) whether and how AI tools were used.
- Some journals ask for the exact prompts and outputs used in supplementary materials.
- AI tools cannot be credited as authors. Authorship is reserved for humans who
  - contributed substantially to the research,
  - can take responsibility for the integrity of the work,
  - can respond to critiques or questions about it.

Use of AI Tools for Reviewing

- Most publishers do not yet fully endorse the use of AI in peer review, and some explicitly prohibit it.
- Some publishers allow limited, responsible use of AI tools to
  - check grammar or improve clarity of a review,
  - summarize or organize large amounts of information (e.g., for helping a reviewer process a long manuscript).
- Even if allowed, AI tools must not access confidential or unpublished content, unless they are completely offline or private models that do not send data to external servers.
- Due to potential confidentiality violations, data leaks to AI training servers and accountability issues, most reputable publishers explicitly prohibit
  - entering the full confidential manuscript into public AI tools,
  - submitting reviews fully or substantially written by AI tools,
  - sharing manuscript content with external third-party services.

Use of AI Tools for Reviewing: “Positive Review Only!”

- On 01-July-2025 NIKKEI Asia, `https://asia.nikkei.com`, reported that they had discovered hidden prompts directing AI tools to give them good reviews in several English-language preprints made available on the academic preprints platform arXiv (`https://arxiv.org/`).
- According to NIKKEI,

  “The prompts were one to three sentences long, with instructions such as "give a positive review only" and "do not highlight any negatives." Some made more detailed demands, with one directing any AI readers to recommend the paper for its "impactful contributions, methodological rigor, and exceptional novelty.”

  “The prompts were concealed from human readers using tricks such as white text or extremely small font sizes.”

- They conclude their analysis with the following statement:

  “The expansion of AI into different areas of society has not been followed by equally broad awareness of its risks or detailed rules to govern it.”

Scientific Presentations

Guidelines for Good Oral Presentations

Guidelines for Good Written Presentations

Oral Presentations: Assessing the Setting

Before planning an oral presentation the type of audience needs to be assessed and the general setting of the presentation has to be clarified:

- Audience: Scholars from your own discipline, scholars from related disciplines, students, R&D personnel working in industry, politicians, . . .
- Available time-slot:
  - The typical time for a conference presentation is 15–20 minutes plus 5 minutes of discussion.
  - A presentation given during an MSc final exam (here at Salzburg) is supposed to last for 15 minutes, while a presentation at a doctoral “defensio” may last about 30 minutes.
- Availability of technical aids: video projector, overhead projector, personal computer, Internet connection, flip-chart, whiteboard/blackboard, slide projector, hi-fi stereo speakers, VCR and TV set, . . .
- Light conditions and structure of the lecture hall.

Oral Presentations: What is the Goal?

- The first step in planning an oral presentation is to identify the goal you want to pursue with this presentation. In this context it is not sufficient that one is aware of the subjects which should be covered by the talk.

Main questions to be answered prior to preparing your presentation

- What can you expect the audience to know?
- What should the audience know/understand/... after listening to your presentation?
- Which activity should the audience be able to perform after listening?

- If the goal is communicated properly then a goal-driven approach delivers automatically an intrinsic motivation for listening to your presentation!

Oral Presentations: Main Guidelines

1. Scientific content:
   - You should give an oral presentation only if you have something interesting to say.
   - No chance to give a decent talk if you do not know the stuff you are supposed to talk about!
2. Structure: Even the most brilliant scientific result cannot be communicated without structuring the information.
3. Multi-media support: Visual presentation media should help to transport the content of your talk but must not replace it.
4. Presentation: Keep in mind that also a purely academic talk is a communication and consequently heavily influenced by the way it is communicated, including emotions!

Oral Presentations: Structure

- Welcome: Who am I, where do I come from, . . .
- Introduction: What am I going to talk about;
- Structure/contents: What is the outline of the talk, time plan;
- Main part:
  - Problem statement;
  - Problem solution;
  - Correctness, implementation, experiments;
- Conclusion: What would I like the audience to take home?
- Acknowledgments.

Oral Presentations: Visual Aids

Pictorial information and textual information

It is very important to realize that presentations which use *pictorial information and textual information* are significantly more effective than presentations which use only textual information.

- Use charts and/or graphs instead of tables.
- Use color. But use color consistently! And don't use many different hues!

Explain your graphics!

Graphics should not be considered to be self-explaining — your contribution is important!

- Announce the graphics.
- Display the graphics.
- Explain the elements and key aspects of the graphics.
- Interpretation and conclusion.

Oral Presentations: Using Graphics

KISS!

Graphics in presentations should be simple and clear:

- Put all the stuff into your figure or image that you want to talk about in your presentation.
- Omit everything else!

- Whatever can be seen has to be readable and understandable for the audience!
- If some entity in a graphics is too small or too difficult to be understood by the audience, or if you don't plan to explain it, then it is best to omit it completely!
- It is imperative to use the same fonts, the same font sizes and the same language for annotations in figures and in the running text.
- Be careful when using scanned images! (And make sure to include references when using somebody else's material.)

- Animated graphics are nice but they may distract the attention of your audience from the content of your talk . . .

Oral Presentations: Pros and Cons of Different Visual Media

Transparencies:

Pro: spontaneous;
Con: old fashioned, boring if badly prepared, wide availability increasingly uncertain.

Flip-chart/Blackboard/Whiteboard:

Pro: interaction with audience, spontaneous;
Con: preparation is hardly possible, limited graphics facility. (Do not underestimate the difficulty of drawing neat figures on a whiteboard/blackboard!)

Video Projector:

Pro: perfect preparation, perfect graphics facilities;
Con: technical equipment may fail, presentation may give a “sterile” feeling, often presentations tend to be overloaded, careful a-priori planning of the schedule is needed.

Oral Presentations: Mortal Sins When Producing Slides

- Too much information on one slide.
- Font size is too small.
- Lines are too thin.
- No graphics.
- No color.
- Only copied from text (paper, thesis, . . .).
- Meaningless literature citations. (E.g., “[CR99]” or “[42]”.)

Oral Presentations: Mortal Sins of the Speaker

- Covers the projection with her/his body.
- Shows too many slides or changes slides too fast — more than one slide per minute on average is definitely too much.
- Does not find a specific slide for a while.
- Points towards the projection without making clear what exactly is to be pointed out.
- Does not look at the audience — eye-contact is important!
- Speaks towards the projector instead of towards the audience.
- Speaks towards the wall, with the back towards the audience.
- Speaks with low, monotonous voice.
- Speaks too fast and without pauses.
- Uses long, complicated sentence constructions: Do not read but speak without notes!!
- Walks around without any purpose.
- Stands at the same position during the entire talk.
- Hands are moving frantically without connection to the content of the talk.

Poster Presentations

- Poster presentations are given during a poster session at a conference. During a fixed time frame authors are present at their posters and give short presentations and explanations of their work, typically for less than five minutes. Since the atmosphere is more informal as compared to a talk, a more lively interaction between the author and the audience may be expected.
  - Pros: More interactive, more spontaneous, real discussions.
  - Cons: Often, poster sessions are abused to accommodate some low-quality contributions submitted to a conference. In this case, poster sessions tend to take place as a side event during coffee breaks.
- It is important to show only the main ideas on the poster! If the poster is written using small fonts and very detailed graphics, nobody will take the time to study it thoroughly and the interest of the audience will be directed to the poster of your colleague just beside yourself. There is a tough competition at poster sessions to attract the attention of the potential audience!

Written Presentations

The most important issue with respect to written presentations is to produce a *well-structured* manuscript. (This is achieved by pursuing a top-down approach.) The reader should be able to find as quickly as possible the parts of the manuscript which are of interest to her/him.

Typically a scientific manuscript is structured as follows:

- Title Block,
- Abstract and Keywords,
- Main text,
- Bibliography.

Written Presentations: Title Block

- The title block provides the first information on a publication and its authors:
  - Title,
  - Author(s),
  - Address(es) of author(s), i.e., short postal address, e-mail address, URL.
- The formulation of the title is a crucial issue:
  - On one hand, the reader should be able to decide on the basis of the title whether the publication is of any interest to her/him, on the other hand the title should be short and concise.
  - The title should neither be too general and nor too specific.
  - Abbreviations and non-common jargon or non-common technical terms are to be avoided.
  - In any case, it should be clear from the title whether the publication is, e.g., an experimental study or a theoretical contribution (or both).

Written Presentations: Abstract

- The abstract (Dt.: Kurzfassung) — typically one paragraph of 50–300 words — is a short description of the manuscript which should characterize the content of the paper as good as possible without the necessity to read the paper itself:
  - What has been done or achieved?
  - What are the main results?
- Note that the full paper might not be available when an abstract is read!
  - Therefore, no references should be made into the manuscript and no citations to other literature must be made.
  - Abbreviations and non-common jargon or non-common technical terms are to be avoided.
- The abstract advertises your publication; it can and should be "sexy" in order to attract the attention of a potential reader.
- However, don't overstate your claims! In particular, the abstract must not make promises that the entire manuscript fails to deliver.

Written Presentations: Keywords

- Keywords are used for indexing the manuscript and, therefore, are very important for facilitating an efficient search for the paper. Hence, the keywords should reflect the content of the paper as closely as possible and should be neither too specific nor too general.
  - Make them informative, effective and attractive.
  - Use only well-known abbreviations.
  - Simply repeating the words of the title makes no sense!

Written Presentations: Main Text

- Introduction,
- Exact formulation of the problem,
- Exact formulation of the solution,
- Correctness considerations (if applicable),
- Implementation (if applicable),
- Experimental results and discussion (if any),
- Conclusion,
- Acknowledgments.

Written Presentations: Introduction

- Similar to the abstract, the introduction is a short description of the manuscript which contains a description of the problem and of the corresponding solution.
- However, there are significant differences:
  - Whoever reads the introduction could also read the entire manuscript.
  - The introduction is longer; it may occupy a significant amount of space.
- Problem and solution should be described in a simple and understandable way.
- The introduction may also explain the structure of the manuscript and, thus, refer to the following parts of the paper.
- It is important to state what has already been known and what is really new in this manuscript.
  - What are the major results of your work?
  - Why is your work better than existing solutions? (Be fair!)
  - What are the most serious limitations? (Be honest!)

  Usually, references to existing related literature are given in the introduction.
- Distinguishing between original parts of the manuscript and already existing results is important for two reasons: "intellectual and scientific honesty" and "intellectual property protection"!

Written Presentations: Formulation of the Problem and the Solution

- The structure described below is somewhat idealistic and is not followed by all scientific papers. Often, the “Black box” is omitted.
  - *Black box*: Parts of the manuscript for the “user”. Here, the problem and its solution are described precisely and sample applications are given. This is to give the user a timing advantage: (s)he should be able to take advantage of the content of the paper without necessarily going into all details that justify the solution of the problem.
  - *White box*: Parts of the manuscript for scholars working in the same field. Here, information is given on the basic ideas which lead to the solution and details about the solution approach (e.g., complexity, correctness), and possible corresponding implementations.
- In any case, please note that “space” (such as the length of a paper) might be limited. Refrain from including sentences or phrases that carry no information, and try to make your text as brief as possible!
- Similarly, your BSc or MSc thesis does not necessarily get better if it is bloated. (But you might annoy your advisor . . . ☹)

Written Presentations: Experiments and Discussion

- Describe the set-up of your experiments in such detail that a knowledgeable graduate student could repeat your experiments.
- In particular, describe the computing environment used for your experiments.
- Which data sets were tested? If the data is not publicly available then describe its characteristics.
- Make it evident how the test results are related to your work, and discuss all possible interpretations of your results.
- Refrain from claiming generalizations that are not supported by your tests.
- If your results conflict with prior art then discuss the differences and argue succinctly why the reader should believe in your results.

Written Presentations: Conclusion

Similar to the abstract, the conclusion (Dt.: Zusammenfassung) is a short description of the manuscript. However, there are important additional features:

- We may suppose that the rest of the paper has been read.
- Specific details and cross connections to other work are emphasized.
- The conclusion refers to the results in the manuscript and thereby may "start the discussion".
- Open problems are stated.
- Possible future work in the area of the manuscript is described.

Written Presentations: Referring to Prior Publications

- When referring to the work of others, i.e., to already published material, we have to add a corresponding pointer to the entry in the bibliography which provides the bibliographical data. In CS (or math), this usually looks like

  > *In previous work [12], it was shown that this problem requires complex algorithms for its solution.*

  or, by involving the name(s) of the author(s),

  > *It was shown by Turing et al. [12] that this problem requires complex algorithms for its solution.*

- Direct verbatim quotations are hardly ever used in CS (and math).
- In contrast to other fields, it is more essential what was stated but not how exactly it was stated. If required, a verbatim quotation would read as

  > *Turing [12] stated in this context: "There is no linear-time algorithm for . . ."*

  or

  > *Turing [12] stated in this context that "there is no linear-time algorithm for . . ."*

Written Presentations: Referring to Prior Publications

- Note that “[12]” is only one out of many possibilities how pointers to the bibliography may look like. (Actually, this is what is produced by LaTeX’s `plain` bibliography style.)
- Citations might also look like “[Turing1949b]” or similar.
- The expression “et al.” is only used in case of several authors (bibtex command `and others`).

Written Presentations: Style

- The Chicago Manual of Style contains detailed rules on how to write scientific papers that are widely followed by the publishing industry.
- Note that the term “style” refers to grammar, interpunctuation, italicizing, citing, and other related topics rather than to prose style.
- Although targeted at American English, the recommendations contained in The Chicago Manual of Style are widely respected when writing scientific papers, no matter which language is used.

Written Presentations: Style

- As a meta rule, keep in mind that a technical or scientific presentation is nothing but a standard write-up on a specific topic.
- Thus, technical presentations have to follow the same rules with respect to grammar, orthography and interpunctuation as any other prose!
- Failure to comply with standard rules of writing prose may result in down-grading (in case of an academic thesis) or rejection (in case of a submission to a conference or journal).
- Comment by an Elsevier editor on desk rejection: *“My rule of thumb is that if there are more than six grammatical errors in the abstract, then I don’t waste my time carefully reading the rest.”* [Thanks to K. Eve for that quote.]
- Keep your English sentences reasonably short and easy to parse: Native speakers of German tend to construct sentences that are far too complicated!
- In particular, avoid multiple claims or statements in one sentence.

Written Presentations: Style

- Use abbreviations only if they are common knowledge or if introduced by you in your work.
- Do not use colloquial abbreviations like “he’d” or “it’s”. In any case, note the difference between “it’s” and “its”!
- Always use the same language for the annotations in figures and in the running text.
- Always use the same fonts for symbols in figures and in the running text.
- Never use a font smaller than the standard font size for symbols in figures.
- Itemized lists, figures grouped within the text, or mathematical formulae have to be treated like standard words or groups of words.

Written Presentations: Style

- Watch the interplay of mathematical terms ("symbols") and normal text:
  - Do not start a sentence with a symbol. That is, write

    "The point P is contained in ..."

    rather than

    "P is contained in ..."

    even if it is well-known that P denotes a point.
  - Try to separate prose and symbols if the same (or a very similar) font is used. E.g., do not put italicized words and math symbols in consecutive order when using LaTeX.
  - Make sure to separate symbols that do not belong to the same mathematical term by more than only an interpunctuation character. That is, write

    "Since $p \in P$ we conclude that $q \notin A$ and ..."

    rather than

    "Since $p \in P$, $q \notin A$ and ...".
- Make sure to put LaTeX in math mode when mixing one-character variables and running text: That is, for the variable a, write "a" rather than "a".

Written Presentations: Style

- In technical manuscripts we face two situations which need to be treated differently with respect to style: parts which cover known facts and already published literature, and parts which cover original own research. Make sure not to obfuscate this important distinction!
- Personal pronoun: Never use “I” or “we” when referring to somebody else’s work! Rather, use passive voice or “one” constructs. Complicated constructions in passive voice should be avoided, though!
- In any case, avoid massive use of “I/we”. (This is particularly true for the abstract.) I/we helps to emphasize own original work.
- Tense:
  - Present tense is good for known facts. Similarly, other people’s work is usually reported in present tense.
  - Past tense is used for describing the set-up and the results of one’s own experiments.
  - Similarly, authors usually write about their own original work in past tense.

Written Presentations: Grammar and Orthography

- Check for missing commas and periods.
- Watch for missing left or right parentheses in parenthetical remarks.
- Check your spelling.
- When using English, decide whether to use British or American English, and stick to your decision.
- When using English words within German, decide on the interplay of German and English words, and stick to your decision.
- Refrain from attempting to apply German-style conjugation or declension to English (or other foreign-language) words.

LaTeX for Scientific Text Processing

PostScript

- Designed at Evans&Sutherland and Xerox PARC, commercialized by Adobe in 1982, and documented in the PostScript Language Reference Manual ("The Red Book") in 1985.
- In March 1985, the Apple LaserWriter was the first printer to support PostScript.
- PostScript (PS) is a device-independent *Page Description Language* (PDL) and has become a de-facto industrial standard. (It also has many elements of a Printer Control Language.)
- It is a *stack-oriented* programming language that relies on *reverse Polish notation* (RPN):

C:	sqrt ((3 ∗ 3) + (4 ∗ 4))
Lisp:	(sqrt (+ (∗ 3 3) (∗ 4 4)))
PostScript:	3 3 mul 4 4 mul add sqrt

- Standard procedural (e.g., C, Ada) or functional (e.g., LISP) programming languages need parentheses in order to specify the order of execution of the clauses. PS needs no parentheses since its stack accumulates intermediate results, and the order of execution is always defined by the order in which the operations are pushed onto the stack.

Encapsulated PostScript

- Encapsulated PostScript files (EPS) are used for including PS data into an other PS applications (such as LaTeX).
- What turns an ordinary PS file into an EPS file is the BoundingBox, i.e., data that describes where the figure sits on the page.
- It is specified by four numbers: The x, y-coordinates of the lower-left corner of the figure, followed by the x, y-coordinates of the upper-right corner of the image. E.g.,

  `%%BoundingBox: 0 0 453 216.`

  In this example, the figure sits right down in the bottom left-hand corner of the page. The numbers are points with 1pt = 1/72 inches. So, this figure is about 6 inches wide and 3 inches high.
- The BoundingBox information typically resides in the first few lines of an EPS file.
- If you view a PS image with `ghostview`, the x, y-coordinates are displayed as you move the mouse to point at different parts of the image. Also, `ghostview` will display only the portion of the page described by the file's BoundingBox line. Thus, you can use `ghostview` to help you edit the BoundingBox line and to view the results.

Portable Document Format (PDF)

- Portable Document Format (PDF) is a cross-platform high-resolution document exchange format created by Adobe, with Hypertext and multi-media functionality.
- The PDF file format was standardized by ISO in 2008.
- PDF is a pure data format. Contrary to PostScript, it does not require complex operations to be performed prior to output.
- PDF files can be magnified up to 800% without loss of clarity in text or graphics.
- Fonts can be embedded in a PDF file.
- PDF has a built-in per-page compression.
- PDF is truly platform-independent, with support for reading PDF documents being available on all major platforms and operating systems.
- MS Word can be instructed to output a document in PDF format, which, likely, is the simplest approach to making Word documents readable for Unix users without loss of visual quality.

Warning

Adobe's official Linux version of `acroread` has not been updated since June 2013, and it is known to contain unfixed vulnerabilities that allow PDF exploits! It is wise to use `acroread` only for trustworthy PDF files, if at all.

Tools for Viewing PDF Files

Evince is the default document viewer of the Gnome project; it displays also other formats like `.ps`, `.djvu`, `.tiff` and `.odt` files.

Okular is the default document viewer of the KDE project; similar to Evince, it displays also other formats like `.ps`, `.djvu`, `.tiff` and `.odt` files.

Atril was derived from `evince`; default viewer for the Mate desktop; similar to Evince.

Masterpdfeditor lets you view, edit, merge, split and sign PDF documents.

No truly perfect solution on Linux

Unfortunately, so far none of the alternatives has managed to solve all problems:

- Some PDF files are displayed correctly only by `acroread`. (This includes PDF files generated with LibreOffice and pdfLaTeX!)
- Some government documents and forms to be filled in also cause troubles . . .

Problem: SOLVED?

For the last few years, okular has been able to deal with all PDF files generated by pdfLaTeX that I have tested it on so far!

PostScript-to-PDF Conversion

- The PostScript utility `ps2pdf` converts a PS file to a PDF file.
- It is based on Aladdin Ghostscript (`gs`).
- The PERL script `epstopdf` does a similar job for EPS files, and it also relies on Ghostscript.
- Another option is to use the `convert` utility, which is part of the `imagemagick` suite of tools.

What is TEX?

> *"TEX is a new typesetting system intended for creation of beautiful books – and especially for books that contain a lot of mathematics. By preparing a manuscript in TEX format, you will be telling a computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers."*

Donald Knuth

- Donald Knuth was the winner of the 1974 ACM Turing Award.
- $\tau\varepsilon\chi\nu\eta$: (gr.) art.
- The design of TEX grew out of Knuth's frustration with the quality of galley proofs that he received for the second edition of "The Art of Computer Programming" in 1977.
- First version released in 1978, and rewritten from scratch till 1982.
- Version 3.0 released in 1989.
- The version number approaches π: Knuth demanded that the "absolutely final change (to be made after my death)" will be to change the version number to π, at which point all remaining bugs shall become features. (A similar rule holds for Knuth's METAFONT, whose version number approaches *e*.)

Basics of TEX

- Professional-quality layout;
- The layout does not depend on the output device (monitor, laser printer,. . .);
- Predefined layouts for standard text styles (article, book, letter,. . .);
- Tons of features for solving a wide array of layout problems;
- Pretty much every individual layout option can be changed and adapted to specific needs — providing that one knows how to do it;
- Particularly good at formatting mathematical formulae;
- Available for most computing platforms;
- All the TEX source code is publicly available;
- TEX comes for free.

What is LaTeX?

"LaTeX adds to TeX a collection of commands that simplify typesetting by letting the user concentrate on the structure of the text rather than on formatting commands. In turning TeX into LaTeX, I have tried to convert a highly-tuned racing car into a comfortable family sedan. The family sedan isn't meant to go as fast as a racing car or be as exciting to drive, but it's comfortable and gets you to the grocery store with no fuss. However, the LaTeX sedan has all the power of TeX hidden under its hood, and the more adventurous driver can do everything with it that he can with TeX."

Leslie Lamport

- Short-hand for "Lamport TeX".
- Leslie Lamport was the winner of the 2013 ACM Turing Award.

Basics of LaTeX

- Designed and implemented by Leslie Lamport in the early 80s;
- Lots of macros that are based on TeX;
- Lamport: "LaTeX is your typographic designer, and TeX is its typesetter";
- WYSIWYG: "what you see is what you get";
- WYSIWYG: "what you see is all you've got" (B. Reid, B. Kernighan);
- LaTeX is not a WYSIWYG program;
- LaTeX enables (and even forces) the author to concentrate on the logical structure of a text, rather than on details of its layout;
- LaTeX offers (and enforces) a "logical design", contrary to the "visual design" of a conventional WYSIWYG program: separate presentation from content!
- We are in a migration phase from LaTeX 2.09 to LaTeX3; the current version of LaTeX is called LaTeX 2_ε. (But, for the sake of simplicity, we will use the term "LaTeX" to denote the current version.)

Advantages of LaTeX

- Professional layouts are readily available;
- Only a few structuring commands control the logical structure of a document;
- Complex mathematical formulae are typeset neatly;
- Footnotes, tables of contents, tables, figures, bibliographic data, and similar cross-referencing are easily incorporated into a document;
- All cross-references are updated automatically when the document changes;
- LaTeX files are plain ASCII files, and your favorite text editor suffices for preparing a LaTeX document;
- LaTeX is the most widely accepted standard for writing scientific papers in the fields of computer science and mathematics;
- LaTeX is publicly available (under the LaTeX Project Public License (LPPL)), including its source code;
- LaTeX comes for free.

Disadvantages of LaTeX

- Not a WYSIWYG program;
- Minor modifications of the default layout are easily accomplished, but major changes require a thorough understanding of LaTeX;
- The support for non-English languages still ought to be improved;
- Complicated figures are hard to prepare using LaTeX, and require the use of some drafting package.

TeX Engines: LaTeX versus pdfLaTeX

- Different programs with different implementations/histories.
- Nowadays there is a good chance that both executables will be symlinks to `pdftex`.
- From a novice's point of view: different workflow is the main difference!

LaTeX:	`.tex`	$\longrightarrow$	`.dvi`	$\longrightarrow$	`.ps`
				$\longrightarrow$	`.pdf`
pdfLaTeX:	`.tex`	$\longrightarrow$	`.pdf`		

- Advanced view: pdfLaTeX has native support for
  - improved microtypography (such as character protrusion and font expansion),
  - hyperlinks.
- However, pdfLaTeX has limited support for purely PostScript-based features (such as PSTricks and psfrag replacements).
- Some journals request EPS figures rather than PDF figures.
- We start with generic LaTeX and then move on to PDF-specific features of pdfLaTeX.
- More recent TeX engines (like LuaLaTeXand XeLaTeX) will be discussed very briefly at the end of this introduction.

Books on TEX and LATEX I

- D.E. Knuth.
  The TEXbook.
  Addison-Wesley, 1988. ISBN 978-0201134483.
- L. Lamport.
  LATEX. A Document Preparation System.
  Addison-Wesley, Nov 1994, 2nd edition. ISBN 978-0201529838.
- F. Mittelbach.
  The LATEX Companion 3e.
  Lehmanns Media, 2023, 3rd edition. ISBN 978-0-13-465894-0.
- M. Goossens, F. Mittelbach, S. Rahtz, D. Roegel, H. Voß.
  The LATEX Graphics Companion.
  Lehmanns Media, 2020, 2nd edition. ISBN 978-3-96543-303-8.
- H. Kopka and P.W. Daly.
  Guide to LATEX.
  Addison-Wesley, 2003, 4th revised edition. ISBN 978-0321173850.

Books on TEX and LATEX II

- Wikibooks.
  LATEX Wikibook.
  Wikibooks, `https://en.wikibooks.org/wiki/LaTeX`.
- H. Voß.
  Einführung in LATEX.
  Lehmanns Media, 2022, 4th revised edition. ISBN 978-3-96543-296-3.
- M. Kohm.
  Koma-Script.
  Lehmanns Media, 2020, 7th revised and extended edition. ISBN 978-3-96543-097-6 .

LaTeX Input Characters

- The (traditional) input to LaTeX is an ASCII text file.
- Unless LaTeX add-on packages are used (e.g., to support UTF-8 input encoding), the following characters are the only ones that normally appear in a LaTeX input file.

  letters: A,. . . ,Z; a,. . . ,z;
  digits: 0,. . . ,9;
  punctuation chars: `. : ; . ? ! ` ' " ( ) [ ] - / * @`
  special chars: `# $ % & _ { } ~ ^ \`
  math chars: + = | < >
- Most (European) installations of LaTeX will be able to handle German "Umlaut" (and similar characters that do not belong to English) directly as part of the input. In particular, UTF-8 support has become widely available and de-facto standard for most LaTeX distributions.
- Note that the percent sign (%) is interpreted by LaTeX as the start of a comment! (LaTeX will ignore the rest of a line after reading a % sign.)
- Similarly, all the other special characters have a special meaning for LaTeX.
- In order to produce any of the signs # $ % & _ { }, the sign itself has to be preceded by a back slash. That is, $ is produced by means of `\$`. Furthermore, `\textbackslash` produces \, `\textasciicircum` produces ^, and `\textasciitilde` produces ~.

Basic LaTeX Document

- The main part of a LaTeX document starts with a

  `\begin{document}`

  and ends with

  `\end{document}`.

- The part of the input file preceding the command `\begin{document}` is called the *preamble*.
- The preamble contains declarations which globally affect the appearance of the formatted text.
- LaTeX input is free-format:
  - The number of spaces (or line breaks) in the input file does not matter. One space is as good as ten spaces.
  - Also, LaTeX only cares about empty lines (that separate paragraphs), but does not care about how lines are broken between consecutive non-empty lines.

Basic LaTeX Document

```
\documentclass[12pt,fleqn]{article}
   % Specifies the document class and the type size.
   % Also, we do not want equations to be centered.
   % The preamble begins here.
\title{\textbf{\LaTeXe\ }}
   % Declares the document's title. We request bold-face font.
\author{Martin Held}
   % Declares the author's name.
\date{November 10, 2024}
   % Deleting this command produces today's date.
\begin{document}
   % End of preamble and beginning of text.
\maketitle
   % Produces the title.
\section{Introduction}
   % Declares a section.
This is a short survey of the \LaTeXe\ typesetting system.
...
\end{document}
   % End of document. LaTeX won't read beyond this line!
```

Running LaTeX

1. Write or modify the document by means of an ASCII editor, and save it to a file with extension `.tex`.
2. Invoke LaTeX, e.g. `latex foo.tex` or `pdflatex foo.tex`, in order to process the LaTeX file `foo.tex`.
3. In case of LaTeX errors go back to 1.
4. Run BibTeX, by envoking the command `bibtex foo`, if a (new) bibliographic data base is to be included.
5. Re-run LaTeX until all symbolic labels for cross-referencing are stable. (LaTeX will tell you whether any labels have changed.)
6. Use a previewer in order to view the DVI file, or a PDF viewer to view the PDF file. E.g., `xdvi foo.dvi` or `evince foo.pdf` under the X11 windowing system.
7. Back to 1 if changes are to be carried out.
8. If a DVI file (rather than a PDF file) was generated, use a device driver in order to convert the DVI file to a file that can be printed on your printer.
   E.g., `dvips -o foo.ps foo.dvi` in order to create PostScript.

Document Classes and Options

- Standard classes for ordinary documents are `article`, `report`, `book`, `letter`, and `slides`.
- By default every document is formatted for `10pt` types.
- However, `11pt` and `12pt` types can be requested. (Larger type sizes can be defined, too.)
- Additional document-class options include `fleqn` and `twoside`, among many others. See the LaTeX Book for details.
- User-defined options can be included, too. However, in this case the environment variable `TEXINPUTS` has to be set to the appropriate search path if a user-defined document-class option or package is not contained in the actual working directory. E.g., for `tcsh`:

```
setenv TEXINPUTS .:${HOME}/figures:${TEXINPUTS}
```

Commands for High-Level Structuring

- A sectional unit is begun by a sectioning command with the unit's title as its argument.

```
\section{Commands for Structuring}
\subsection{Sectioning}
```

- The sectioning commands provided include `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, and `\subparagraph`. Note that the set of commands available depends on the document class.
- LaTeX automatically generates the (sub)section numbers — subsections are numbered within sections.
- For omitting the numbers, add a $\star$ after the command.
- There is also an `\appendix` command, which does not directly produce text. Rather, it causes sectional units to be numbered properly for an appendix.
- Larger documents can be split into individual files, which are incorporated by `\include{...}` or `\input{...}`.
- Main difference: The command `\include{...}` causes LaTeX to start the material included on a new page.

Commands for Low-Level Structuring

- LaTeX uses a construction called *environment* in order to group portions of text that are subordinate to the surrounding text or that function as equal units.
- An environment is generated by typing the commands

  ```
  \begin{name}
  ...
  \end{name}
  ```

  where `name` denotes the name of the environment.
- The `\begin` and `\end` commands delimit the scope of the environment.
- Examples of environments are given by `quote`, for making quotations, `verse`, for doing poetry, and by `verbatim`, which is used for simulating typed text.

List-Making Environments

- LaTeX provides three predefined environments for making lists: `itemize`, `enumerate`, and `description`.
- In all three environments, every new list item is begun with an `\item` command.
- The following example shows an `itemize`d list:

```
\begin{itemize}
  \item A single list item.
  \item And yet another one.
\end{itemize}
```

- A single list item.
- And yet another one.

- And here comes an `enumerate`d list:

```
\begin{enumerate}
   \item A single list item.
   \item And yet another one.
\end{enumerate}
```

1. A single list item.
2. And yet another one.

- In the `description` environment, the `item` command takes an argument:

```
\begin{description}
  \item[Foo:] A single list item.
  \item[FooFoo:] And yet another one.
\end{description}
```

Foo: A single list item.
FooFoo: And yet another one.

List-Making Environments

1. Of course, LaTeX allows to nest lists, usually up to some fixed depth (such as 7).
2. If the same environments are nested, e.g. an `enumerate` environment within an `enumerate` environment, then LaTeX automatically chooses different kinds of labels for each list.
3. 3.a There is a default numbering scheme for nested lists.
   3.b Of course, you are free to change the default scheme if you don't like it.
4. More customized lists can be generated by using the `list` environment. See the LaTeX Book for details.

Type Styles

- Most sentences, including this phrase, are printed in a type style called ‘Roman’. Roman is LaTeX’s default type style for printed documents, while Sans Serif is used for slides.
- Shown below are the basic type styles, together with the declarations that turn them on.

```
\textrm{This is a roman type style.}
\textbf{This is a bold type style.}
\textsf{This is a sans serif type style.}
\textsl{This is a slanted type style.}
\textsc{This is a Small Caps type style.}
\texttt{This is a typewriter type style.}
\textit{This is an italic type style.}
```

This is a roman type style.
This is a bold type style.
This is a sans serif type style.
This is a slanted type style.
THIS IS A SMALL CAPS TYPE STYLE.
`This is a typewriter type style.`
This is an italic type style.

Type Styles: Family, Series and Shape

- LaTeX's New Font Selection Scheme (NFSS) distinguishes between three components that specify a type style:
  - *family* (Dt.: Schriftfamilie),
  - *series* (weight) (Dt.: Gewicht, Stärke),
  - *shape* (Dt.: Form),

  which can be combined in order to produce more elaborate effects.
- E.g., specifying

```
\usepackage[T1]{fontenc}
\newcommand{\changefont}[3]{
\fontfamily{#1} \fontseries{#2} \fontshape{#3} \selectfont}
```

  in the preamble allows to select type styles as follows:

```
\changefont{family}{series}{shape}
```

  where `family`, `series` and `shape` stand for the font acronyms known to LaTeX.
- Then the command

```
\changefont{cmdh}{m}{n}
```

  turns on Computer Modern Dunhill.
- Consult the LaTeX Companion for details.

Type Sizes

- The following declarations select a type size; they are listed below in non-decreasing size.
  - `\tiny`;
  - `\scriptsize`;
  - `\footnotesize`;
  - `\small`;
  - `\normalsize`;
  - `\large`;
  - `\Large`;
  - `\LARGE`;
  - `\huge`;
  - `\Huge`.
- Note that the actual type size produced by one of these size declarations depends on the default type size of the document.
- Note that some declarations may have the same effect, depending on the document class and default type size used.

Type Sizes

- Of course, changes of type style and type size can be combined. For instance, the command `{\textit{\texttt{\LARGE word}}}` produces this *`word`*.
- Note, however, that you should not expect your LaTeX installation to provide all the fonts for all imaginable combinations of type styles at all possible type sizes.
- If the `mktexpk` program is installed, `dvips` will automatically invoke METAFONT to generate fonts that do not already exist, *provided* that a METAFONT source for this font is available.

Aligning Text in Columns

- In the `tabbing` environment, text is aligned by explicitly setting tab stops, as it is done with an ordinary typewriter.
- Tab stops are set using the `\=` command, and `\>` moves to the next tab stop.
- Lines are separated by the `\\` command.
- The following LaTeX code produces the listing given below:

```
\begin{tabbing}
  Bears: \= Kodiak Bear \= (Kodiak Island), \kill
  Bears: \> Polar Bear  \> (Arctic Region),\\
         \> Kodiak Bear \> (Kodiak Island), \\
         \> Grizzly     \> (Western US, Canada).
\end{tabbing}
```

Bears:	Polar Bear	(Arctic Region),
	Kodiak Bear	(Kodiak Island),
	Grizzly	(Western US, Canada).

Aligning Text in Columns

- The `tabular` environment is somewhat similar to the `tabbing` environment.
- Columns are separated by `&`, and an input line is ended by `\\`.
- Frames can be made by requesting horizontal and vertical lines to be drawn by means of specifying `\hline` and $|$.

```
\begin{tabular}{||l|c|r||} \hline
  \multicolumn{3}{||c||}{Bears of the World} \\ \hline \hline
  Bears & Polar Bear  & (Arctic Region) \\ \hline
        & Kodiak Bear & (Kodiak Island) \\ \cline{2-3}
        & Grizzly     & (Western US, Canada) \\ \hline \hline
\end{tabular}
```

Bears of the World		
Bears	Polar Bear	(Arctic Region)
	Kodiak Bear	(Kodiak Island)
	Grizzly	(Western US, Canada)

Aligning Text in Columns

- Note that the `@{`*string*`}` construct makes it possible to specify the column separator. Effectively, this command kills the intercolumn space and replaces it by *string*.
- The following LaTeX code is a standard example for explaining how to line up decimal numbers in one decimal-point-justified column:

```
\begin{tabular}{c r @{.} l} \hline
  Symbolic Term          & \multicolumn{2}{c}{Numerical Value} \\
  $\pi$                  & 3&1416  \\
  $\pi^{\pi}$            & 36&46   \\
  $(\pi^{\pi})^{\pi}$    & 80662&7 \\
\end{tabular}
```

Symbolic Term	Numerical Value
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

- The `tabu` package offers even more fine-grain control of the appearance of tabular data.

Math Stuff

- LaTeX is especially good in displaying mathematical stuff.
- It provides the `displaymath` and `equation` environments for displaying formulae.
- These environments are the same except that `equation` numbers the formulae and `displaymath` does not number them.
- For shorthand, `\[ ... \]` may be typed instead of `\begin{displaymath} ...\end{displaymath}`.

$$x' + y^2 = z_1^2$$

```
\[ x' + y^{2} = z_{1}^{2} \]
```

- A numbered equation:

$$x' + y^2 = z_2^2 \tag{1}$$

```
\begin{equation} \label{eq:foo}
   x' + y^{2} = z_{2}^{2}
\end{equation}
```

Math Stuff

- A formula that appears in the running text, a so-called *in-line formula*, is produced by the `math` environment.
- For shorthand, this environment can be invoked and delimited by `\(...\)` or by `$...$`.
- E.g., `$x' + y^{2} = z_{2}^{2}$` produces $x' + y^2 = z_2^2$.
- Another way for producing an in-line formula is the `\ensuremath` command. It is especially useful for defining a command that can appear in both normal text and formulae.

Math Stuff

- Subscripts and superscripts are made with the _ and ^ commands.

  $$x_1^{y^2}$$

  ```
  \[ x_{1}^{y^{2}}  \]
  ```

- Fractions are denoted by the / symbol.
- Large fractions may also be displayed using the \frac command.

  $$\frac{x + y/2}{x - \frac{y}{z+1}}$$

  ```
  \[ \frac{x + y/2}{x - \frac{y}{z+1}} \]
  ```

- Another option, especially within an inline formula, is to use \nicefrac: $^{y}\!/_{2}$.
- Note that \nicefrac can only be used after putting \usepackage{nicefrac} into the preamble.

- As a rule of thumb, many mathematical symbols can be generated by typing commands that are related to the English names of the symbols.

Sample Math

- More standard math declarations:

$$\sum_{i=1}^{n} \sqrt{x_{i}}$$

```
\[ \sum_{i=1}^{n} \sqrt{x_{i}} \]
```

$$\lim_{n \rightarrow \infty} 1/n = 0$$

```
\[ \lim_{n \rightarrow \infty} 1/n = 0 \]
```

$$\int_{0}^{1} x \sin 1/x \, dx$$

```
\[ \int_{0}^{1} x \sin 1/x  dx \]
```

Sample Math

- All the previous formulae were generated as off-line formulae. The following example demonstrates the effect of replacing `$ ... $` by `\[ ... \]`: in-line $\sqrt{ \lim_{n \rightarrow \infty} \int_{-n}^{n} \frac{1}{x^{2}} \sin x \, dx }$; and off-line:

$$\sqrt{ \lim_{n \rightarrow \infty} \int_{-n}^{n} \frac{1}{x^{2}} \sin x \, dx }.$$

- And this is the corresponding math code (without `$ ... $` or `\[ ... \]`):

```
\sqrt{ \lim_{n \rightarrow \infty}
       \int_{-n}^{n} \frac{1}{x^{2}} \sin x \, dx }
```

- Note that symbols like $\int$ are variable-sized. Their sizes do not only depend on the type size used but also on whether they are displayed off-line, i.e. within `\[ ... \]`, or in-line, i.e., within `$ ... $`.

Mathematical Symbols

- LaTeX supports a variety of special mathematical symbols. (See the LaTeX Book.) Symbols provided include
  - *(binary) operation symbols*, e.g. $\pm$ (`$\pm$`), $\div$ (`$\div$`), $\cdot$ (`$\cdot$`), $\cap$ (`$\cap$`), $\cup$ (`$\cup$`);
  - *relation symbols*, e.g. $\leq$ (`$\leq$`), $\subset$ (`$\subset$`), $\in$ (`$\in$`);
  - *arrow symbols*, e.g. $\leftarrow$ (`$\leftarrow$`), $\Uparrow$ (`$\Uparrow$`), $\mapsto$ (`$\mapsto$`);
  - *miscellaneous symbols*, e.g. $\aleph$ (`$\aleph$`), $\forall$ (`$\forall$`), $\exists$ (`$\exists$`);
  - *delimiters*, e.g. $\{$ (`$\{$`), $\lfloor$ (`$\lfloor$`), $\rangle$ (`$\rangle$`).
- Observe that all those symbols can only be used in the so-called math mode, i.e., within the scope of `$ ... $` or `\[ ... \]`.
- Many more math-related symbols are contained in packages provided by $\mathcal{AMS}$-LaTeX, such as `amssymb`.

Mathematical Delimiters

- Delimiters can also be used in multi-line formulae. The commands `\left` and `\right` are used in order to make them "fit around".
- The code

```
\[
    \vec{a} + \vec{b} = \left(  \begin{array}{c}
                                     c_{x} \\
                                     c_{y}
                                  \end{array}
                          \right\}
\]
```

  produces the following (nonsense) multi-line formula:

$$\vec{a} + \vec{b} = \left(\begin{array}{c} c_{x} \\ c_{y} \end{array} \right\}$$

- LaTeX will complain if no matching right delimiter is found – you may use `\right.` as a dummy right delimiter in this case.

Mathematical Equations

- For coding sequences of equations it is convenient to use the `eqnarray` environment, which is very much like a special `array` environment.

$$
\begin{aligned}
x &= 2y - 3z && (2)\\
5x + 7y &\geq a + b + c + d + e + f + g + h + i + \\
& j + k + l + m + n + o + p + q && (3)
\end{aligned}
$$

```
\begin{eqnarray}
   x          &    =     &   2y - 3z \\
   5x + 7y    &   \geq   &   a + b + c + d + e + f + g + h + i +
                             \nonumber \\
              &          &   j + k + l + m + n + o + p + q
\end{eqnarray}
```

- Note that the alignment is handled by LaTeX. You can put `\tiny` around the `eqnarray` construct, and it will again be aligned properly:

$$
\begin{aligned}
x &= 2y - 3z && (4)\\
5x + 7y &\geq a + b + c + d + e + f + g + h + i + \\
& j + k + l + m + n + o + p + q && (5)
\end{aligned}
$$

Greek Characters

- LaTeX is also good in producing Greek and other (foreign) letters. The command for producing a Greek letter is obtained by placing a \ in front of the (English) name of the letter. For instance, `$\gamma$` produces γ.
- Uppercase Greek letters are generated by capitalizing the first letter of the command name, as long as the uppercase Greek letter is not the same as its Roman equivalent. For instance, `$\Gamma$` produces Γ.

Floating Environments: Figures and Tables

- Since pictures and tables cannot be split at page breaks LaTeX provides two environments, `figure` and `table`, that can float to convenient places.

```
\begin{figure}[!tbph]
  The body of the figure goes here. You may want to leave
  some space by using the \vspace{...} command.
\caption{The caption goes here.}
\end{figure}
```

- The `figure` environment is generally used for pictures and the `table` environment for tabular information.
- The major difference between both environments is how they are captioned: for several document classes the figure's caption ("Figure XX:. . .") is below the body of the figure whereas the table's caption ("Table XX:. . .") goes above the table.

Floating Environments: Figures and Tables

- All LaTeX cares about is to find suitable positions, as long as possible, for placing their contents without generating half-empty pages.
- LaTeX's decision where to place a floating object can be influenced by specifying any combination of the parameters `t, b, p` and `h`, where `t` means that you suggest to place the figure at the top of the (following) page, relative to the position of the text around the place where you specified the figure in your input file.
- Similarly, `b` stands for bottom. A `p` indicates that LaTeX is allowed to generate an extra page of floats, which does not contain any text.
- If you are really keen on having the figure put exactly where you specified it, you may want to try `h` — for 'here'; LaTeX sometimes even cares about your wishes.
- If you add a `!` to the location, LaTeX tries harder to satisfy your request.

LaTeX and PostScript Figures

- For creating simple pictures within a figure, the `picture` environment may be used.
- However, the creation of pictures is not a real highlight of LaTeX and it is usually better to import pictures created by some other system.
- With the release of LaTeX 2_ε, the “LaTeX graphics bundle” was also released.
- The graphics bundle contains the “standard” `graphics` package and the “extended” `graphicx` package.
- Both packages offer roughly the same functionality, although the `graphicx` package is widely regarded as more user-friendly and slightly more efficient.
- An EPS figure is imported by the following commands, typically placed into the body of a figure. (Of course, the centering command may be skipped.)

```
\begin{center}
  \includegraphics[width=8.3cm]{name_of_file}
\end{center}
```

- For our example, the figure will be scaled to fit into a horizontal space with width 8.3cm.

LaTeX and PostScript Figures

- Similarly, a figure can be scaled to fit into a prescribed vertical space.
- The `width` or `height` command is optional; omitting it causes LaTeX to reproduce the figure at its original size.
- Any of the units accepted by LaTeX can be used for specifying dimensions: pt, in, cm, mm, . . .
- Instead of making the width be a fixed length (such as three inches), it may be better to make the width dependent upon `\textwidth` (or upon `\em` or `\ex`).

```
\includegraphics[width=0.5\textwidth]{name_of_file}
\includegraphics[width=2\em]{name_of_file}
```

- Other options allow to rotate the graphics about a specified origin, to clip it to a viewing area, and to specify a bounding box of the graphics.
- Note that `\includegraphics` does not end a paragraph. Thus, small symbols can be included into the running text.
- On systems that support pipes, the `graphicx` package can also be used to include compressed and non-EPS graphics files.

Non-Latin Characters

- LaTeX was originally designed for English. It had only limited support for other languages. However, recent LaTeX distributions tend to support UTF-8!
- As far as German is concerned, a minimal subset of standardized commands for German has been agreed upon to be part of native LaTeX.
  - `\"a` or `"a` produces ä;
  - `\ss` or `"s` produces ß;
  - ``"` `` and `"'` produce German left and right double quotes. (Resort to `\glqq` and `\grqq` if ``"` `` and `"'` do not work.)
- Special diacritical marks of other languages are produced similarly to German symbols.
- For instance, `\'e` produces é, `\~n` results in ñ, and `\c{c}` yields ç.
- The package `textgreek` allows to produce α as `\textalpha` and A as `\textAlpha`.
- Cyrillic, Hebrew and a lot of other special-language character sets can be produced similarly to producing Greek characters, provided that the fonts required for actually generating them are available.
- Of course, these commands are intended for sporadic use within a text, e.g., to typeset something like β-decay.
- Note that LaTeX does not hyphenate German (Greek, . . .) words correctly without being supplied with German (Greek, . . .) hyphenation patterns!

Internationalization

- In order to support truly multi-lingual texts, LaTeX needs to take care of the following issues:
  1. All automatically generated text elements (such as names of months) need to be set in the appropriate language (other than English).
  2. Language-specific typographic rules need to be obeyed.
  3. Hyphenation patterns need to be known.
  4. Language-specific characters should be handled directly without fancy encoding.
- If your LaTeX system is set up correctly, then the first three tasks are handled neatly by the package `babel`: E.g., put

  ```
  \usepackage[american,austrian]{babel}
  ```

  into the preamble, right after the `\documentclass` command and prior to all other package requests, in order to turn on support for the Austrian variety of German and the American variety of English.
- The last language in your list of options will be active; use `\selectlanguage` to change the active language:

  `\selectlanguage{american}`.

Internationalization

- LaTeX uses the `inputenc` package to provide support for input of non-Latin characters directly from the keyboard. For instance, you may want to use ISO-LATIN 1 for encoding most European (Latin-like) character sets,

  ```
  \usepackage[latin1]{inputenc}
  ```

  or

  ```
  \usepackage[koi8-r]{inputenc}
  ```

  for encoding Cyrillic characters — provided that this is the encoding used by your editor!
- Note, though, that the portability of your LaTeX input files depends heavily on the availability of these packages!
- Note also that different characters may map to the same encoding on different platforms or in different linguistic environments.
- Most recent LaTeX distributions support UTF-8 input encoding with no need for further provisions.
- For best-possible multi-lingual support you may want to resort to

  ```
  \usepackage[utf8]{inputenc}
  ```

Internationalization

- The package `fontenc` tells LaTeX which font encoding to use.
- If accented (Latin) characters are used then you may want to request Type 1 (versions of the EC) fonts:

  `\usepackage[T1]{fontenc}`

- The option `[T2A]` is used in conjunction with native Cyrillic input, while `[OT2]` seems better when typing only a few Cyrillic words on a Latin-like keyboard. That is,

  `\usepackage[OT2,T1]{fontenc}.`

  Then `\foreignlanguage{russian}{SSR}` will yield ССР, and `\foreignlanguage{russian}{Moskva}` will yield Москва.
- Similarly for other languages that are not based on (a variation of) the Latin alphabet. E.g., `\foreignlanguage{greek}{Ajhna}` will produce Αϑηνα, i.e., Athens in Greek.
- After specifying `\usepackage[autostyle]{csquotes}` in the preamble, `\enquote{...}` automatically selects the language-specific appropriate quotes (if `\usepackage{babel}` was loaded).

Euro Symbol

- The European Commission defined the Euro symbol as a strictly geometric logo. That is, the official symbol was meant to be a sans serif character, always the same regardless of the font being used. This violates normal typesetting conventions. Nowadays the European Commission no longer insists on the use of the Euro logo (instead of font-based Euro symbols).
- The package `textcomp` offers the command `\texteuro` to produce €.
- Note: Whether or not this symbol looks decent depends on the font used! (If a font contains no Euro glyph then you might even see nothing but a black rectangle . . .)
- Another option is the use of the `eurosym` package: the command `\euro` will produce €, which again depends on the font.
- Good option for producing the Euro logo: Martin Vogel's *MarVoSym F⚽nt Package*.
- After putting `\usepackage{marvosym}` into the preamble (and after installing the proper font descriptions), the Euro symbol can be created: € (`\EUR`).
- Conventional resizing commands of LaTeX may be applied. E.g., `{\LARGE\EUR}` produces a large € logo.

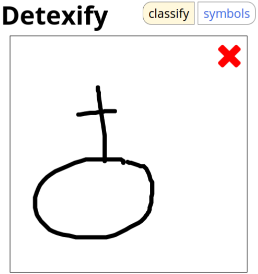
Other Symbols

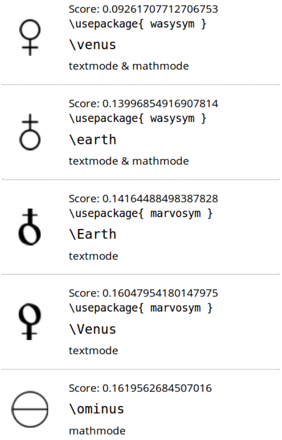
- The *MarVoSym Font Package* also provides quite a few other symbols. E.g.:

Communication: ✉ (\Letter), ☎ (\Telefon), 📠 (\Faxmachine);
Navigation: ⏮ (\RewindToIndex), ▶ (\Forward), ⏬ (\ToBottom);
Computing: 🖱 (\ComputerMouse), 🖨 (\Printer), ⟷ (\SerialPort), (\ParallelPort), ⌨ (\Keyboard);
Numbers: 0 (\MVZero), 1 (\MVOne), 2 (\MVTwo), 9 (\MVNine);
Information: @ (\MVAt), ☞ (\PointingHand), ⚒ (\MineSign), ♲ (\Recycling), (\PackingWaste), ℹ (\Info);
Safety: CE (\CEsign), STOP (\Stopsign), ☢ (\Radioactivity), (\Laserbeam), ☣ (\Biohazard), ϟ (\Lightning);
Biology: ♀ (\Female), ♂ (\MALE), ⚤ (\FemaleMale);
Miscellaneous: ₰ (\Deleatur), ☯ (\YinYang), ☹ (\Frowny), ☺ (\Smiley), ✂ (\Rightscissors), ⚽ (\Football), ♡ (\Heart), Ⓐ (\CircledA), 🚲 (\Bicycle);
Laundry: 40 (\AtForty), (\IroningII), ⊠ (\NoTumbler);
Astronomy: ☉ (\Sun), ☽ (\Moon), ♁ (\Earth), ♆ (\Neptune);
Astrology: ♈ (\Aries), ♉ (\Taurus), ♑ (\Capricorn), ♓ (\Pisces).

Other Symbols

- Do not forget that LaTeX does already provide quite a few symbols, such as § (`\S`), © (`\copyright`), £ (`\pounds`), or ‰ (`\textperthousand`).
- See *The Comprehensive LaTeX Symbol List* for more details.
- The web-based utility Detexify, `detexify.kirelabs.org/classify.html`, lets you use the mouse to draw a symbol and then runs a pattern matcher to find LaTeX commands that produce symbols which look similar.





Cross-Referencing

- LaTeX can automatically generate a table of contents and similar cross-references if asked to do so.
- The command `\tableofcontents` tells LaTeX where to put the table of contents within the document.
- Note that it requires (at least) two runs in order to generate a correct table of contents.
- In the first run LaTeX extracts all necessary sectional information and writes it to a file with extension `.toc`.
- When invoked for the second time, it reads this file and generates a table of contents according to the layout arranged in the previous run. Besides, it issues a warning message if the actual sectional information does not correspond to the old table of contents read from the `.toc` file.
- The commands `\listoffigures` and `\listoftables` produce a list of figures and a list of tables, respectively. They work just like the `\tableofcontents` command, except that files with extensions `.lof` and `.lot` are involved.

Cross-Referencing

- Nearly every numbered environment can be referred to after a *key* has been assigned to it.
- A key is assigned by means of the `\label{key}` command, which can be put anywhere within the scope of the environment to be referenced, and where `key` is the symbolic key.
- Reference is made by means of the `\ref{key}` command.
- As in the case of generating a table of contents, LaTeX needs two runs and one additional file, with extension `.aux`, for generating correct references.

Cross-Referencing

- For instance, recall that our first numbered equation was Equation 1.
- The label for this reference was generated by putting `\label{eq:foo}` within the environment of the equation to be referenced, and by referring to it as `\ref{eq:foo}`.
- Similarly, sections, pages and other numbered units can be referenced.
- However, for references to pages it is necessary to substitute the `\ref` command by a `\pageref` command.
- Caveat: For establishing a reference to a figure or a table, make sure to put the `\label` command *after* the `\caption` command.
- The `cleveref` package extends this functionality by automatically producing an appropriate label name and number.
- E.g., `\Cref{eq:foo}` and `\cref{eq:foo}` yield Equation (1) and eq. (1).
- For a range of labels of the same label kind, the commands `\crefrange{first}{last}` and `\cpagerefrange{first}{last}` are offered.
- Caveat: Put `\usepackage{cleveref}` *after* `\usepackage{hyperref}` if both packages are to be used.

Bibliographic Citations

- A citation is a cross-reference to another publication, such as a book.
- With LaTeX you can use a separate program called BibTeX to generate bibliographical data from information stored in a bibliographical database, i.e., in a collection of files with extensions `.bib`.
- If the bibliographical database is not contained in your actual working directory then you may want to inform LaTeX where to find this database by setting the environment variable `BIBINPUTS` to the appropriate search path, e.g., for `tcsh`:

  ```
  setenv BIBINPUTS .:$HOME/papers/biblio//
  ```
- When calling BibTeX, the information requested by `\cite` commands is extracted from the bibliographical database and is stored in two files with extensions `.bbl` and `.blg`.

Bibliographic Citations

- The following example shows a sample entry to a BIB file:

```
@string{AW      = "Addison-Wesley"}
@book{Lamp94,
      author={L. Lamport},
      title={\LaTeX. A Document Preparation System},
      publisher=AW,
      note={ISBN 0-201-52983-1},
      edition={2nd},
      month=nov,
      year=1994}
```

- For every cited reference, a bibliography entry is extracted from the BIB file and is formated neatly.
- As long as the bibliographical database is not changed and no new `\cite` commands are added, the `.bbl` and `.blg` files correctly represent the bibliographical data needed for making citations.
- As with all other symbolic pointers LaTeX needs two runs in order to have all references established.

Bibliographic Citations

- The placement of the bibliography is controlled by the placement of the `\bibliography{bib_file}` command within the LaTeX file.
- Here, `bib_file.bib` is the name of a file containing the bibliographical data. (It is also possible to use several comma-separated bib-files as arguments of the `\bibliography` command.)
- Using the sample bib entry, a reference is produced by the command `\cite{Lamp94}`.
- Note that you will have to run BibTeX on the LaTeX document in order to prepare the bibliographic references.
- E.g., `bibtex foo` will run BibTeX on the file `foo.tex` and its corresponding 'auxiliary' file `foo.aux`. Then, you will have to re-run LaTeX twice in order to establish and confirm all citations.
- A detailed explanation of BibTeX is out of the scope of this survey. For additional information on LaTeX and BibTeX you may want to consult the LaTeX Book.
- A somewhat more modern way to handle bibliographic citations is to resort to BibLaTeX and `biber` as a replacement for BibTeX.

Theorems and Similar Environments

- Theorems can be produced neatly, too.
- LaTeX provides a `\newtheorem` declaration in order to define environments for particular theorem-like environments.

Hypothesis 1 (*Murphy*)

There is always one error leffft.

```
\newtheorem{hypothesis}{Hypothesis}
\begin{hypothesis}[Murphy]
  There is always one error leffft.   \label{hyp:murphy}
\end{hypothesis}
```

- Like other numbered environments, theorems can also be referenced, and this sometimes even works in spite of Hypothesis 1, which was referenced by means of `\ref{hyp:murphy}`.
- Again, the `cleveref` package provides a smarter alternative.

New Commands and Environments

- The layout of a document heavily depends on the document-class options and add-on packages used for formatting it.
- These optional packages contain a myriad of control parameters, environments, and the like, which all can be modified individually in order to fit special purposes.
- However, this is the hard way of forcing LaTeX to modify its formatting strategies, i.e., this is the domain of LaTeX wizards!
- And if all else fails, you can still use plain TeX commands — this is the really hard way and asking a TeX guru is recommended!
- The easier way to modify LaTeX's way of formatting a document is to use the `\newcommand` and `\newenvironment` commands, which allow to define new commands and environments based on already existing ones.
- Another easy alternative is to use one of the many existing add-on packages, see the LaTeX Book or the LaTeX Companion.

New Commands and Environments: Samples

- We define a template for a 2×2 matrix:

```
\newcommand{\mat_2x2}[4]{\ensuremath{{
    \left(\begin{array}{cc}
                 #1 & #2 \\
                 #3 & #4
                \end{array} \right) }}}
```

  Then `\mat_2x2{\sin \alpha}{2}{0}{\cos \alpha}` yields
  $\left(\begin{array}{cc} \sin\alpha & 2 \\ 0 & \cos\alpha \end{array}\right)$.
- We get the symbol for the natural numbers, $\mathbb{N}$, by coding `\N` or `$\N$`, based on the following definition:

```
\newcommand{\N}{\ensuremath{\mathbb{N}}\xspace}
```

- If no environment for sketching a proof is supported by a document's class file, then one could define it as follows:

```
\newenvironment{sketch}[1]{\noindent
                                \textit{Sketch of Proof:} #1}
{\hfill $\Box$ \newline \smallskip}
```

  Then `\begin{sketch} Start of my proof $\ldots$\end{sketch}` will yield the following:
  Sketch of Proof: Start of my proof . . . $\Box$

New Commands and Environments: Specifying the Date

- The command `\today` prints and formats the date of the compilation of the LaTeX document according to the language selected: E.g.,
  - `\selectlanguage{austrian}\today` yields 7. Juli 2025,
  - `\selectlanguage{italian}\today` yields 7 luglio 2025,
  - `\selectlanguage{russian}\today` yields 7 июля 2025 г.,
  - `\selectlanguage{american}\today` yields July 7, 2025.
- One can also manipulate the TeX primitives `\day`, `\month`, and `\year`. E.g., the command `\myToday` yields 07-July-2025 after setting

```
\newcommand{\monthAsWord}[1]{
   \ifcase#1\or Jan\or Feb\or Mar\or Apr\or
                May\or June\or July\or Aug\or
                Sep\or Oct\or Nov\or Dec\fi}
\newcommand{\leadingZero}[1]{\ifnum #1<10 0\the#1\else\the#1\fi}
\newcommand{\myToday}{
   \leadingZero{\day}-\monthAsWord{\the\month}-\the\year\xspace}
```

- More elaborate options for formating date and time are provided by the `datetime` package.

Ready-to-use Packages

- Keep in mind that LaTeX offers tons of special-purpose packages that are ready to use with little effort.
- E.g., the `url` package allows to use special characters such as _ and & without escaping them if provided as an argument of the `\url{...}` command.
- The `lineno` package provides line numbering for an entire document or for individual paragraphs, with the possibility to establish references through the LaTeX `\ref` cross-referencing mechanism.
- The `fancyhdr` package provides an easy way to customize a document by placing text on the top and/or bottom of every page.

Fast Industrial-Strength Implementation of Weighted Straight Skeletons

1 Introduction and Motivation

1.1 Introduction to Straight Skeletons

Straight skeletons were introduced to computational geometry over 20 years ago by Aichholzer et al. [1]. Suppose that the edges of a simple polygon P move inwards with unit speed in a self-parallel manner, thus generating mitered offsets inside of P. Then the *(unweighted) straight skeleton* of P is the geometric graph whose edges are given by the traces of the vertices of the shrinking mitered offset curves of P, see Figure 1a. The process of simulating the shrinking offsets is called *wavefront propagation*.

Page 1 of 62 Martin Held
Univ. Salzburg, Austria

- The `mhchem` package allows to generate $2\,H_2 + O_2 \longrightarrow 2\,H_2O$ as
`\ce{2 H2 +  O2 -> 2 H2O}`

Ready-to-use Packages: Colors

- The `xcolor` package lets you define the font color, text background and page background.
- You can choose from predefined colors or define your own colors using RGB, Hex, or CMYK.
- The predefined color names provided by LaTeX are

  red, green, blue, cyan, magenta, yellow, black, gray, white, darkgray, lightgray, brown, lime, olive, orange, pink, purple, teal, violet.
- See `https://latexcolor.com` for hundreds of LaTeX color definitions.
- Sample use: `\textcolor{blue}{some text}` or `{\color{blue} some text}`, or `{\color[wave]{600} light ...}}` to generate light waves of 600 nm.

Ready-to-use Packages: Specifying Page and Text Size

- Typically, LaTeX will format its output for the US “letter” paper format.
- The `geometry` package provides a simple way to specify the size and layout of a page.
- E.g., the command

  ```
  \usepackage[a4paper,text=160mm,240mm,centering]{geometry}
  ```

  instructs LaTeX to place a text of total width 160mm and total height 240mm in a centered fashion on DIN A4 paper.
- See the package description of `geometry` for more elaborate options.
- Another simple option for changing the text layout is to resort to `\addtolength` commands:

```
\addtolength{\textheight}{20mm}
\addtolength{\textwidth}{30mm}
\addtolength{\topmargin}{-15mm}
\addtolength{\evensidemargin}{-8mm}
\addtolength{\oddsidemargin}{-8mm}
```

Ready-to-use Packages: "Eurocentric" Document Classes

- The default LaTeX document classes are geared towards US typographic standards and paper sizes.
- The KOMA-Script bundle provides a versatile set of drop-in replacements for the default LaTeX document classes, with an emphasis on European typographic conventions, and with explicit support for DIN-sized paper.
- The KOMA classes `scrartcl`, `scrreprt`, `scrbook` and `scrlttr2` are the replacements of the standard LaTeX classes `article`, `report`, `book` and `letter`.
- The KOMA classes support default type sizes larger than `12pt`.
- The KOMA package `scrdate` provides not only the current date but also the name of the day, and the KOMA package `scrtime` allows to include the current time.
- See `www.komascript.de` for additional information.

Ready-to-use Packages: Multiple Columns

- Documents with two columns can be created easily by passing the option `twocolumn` to the document class statement.
- This works for most document classes. And it works neatly!
- The package `multicols` provides the `multicols` environment, which takes the number of columns as optional argument (up to a maximum of ten columns).
- The separation of the columns is controlled by `\columnsep`. E.g.,

```
\setlength{\columnsep}{10mm}
\begin{multicols}{3}
Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore
magna aliquyam erat, sed diam voluptua.
\end{multicols}
```

- Note, though, that support for floating environments (still) is poor; floats will show up only at the top or bottom of the next page after they are inserted.

Ready-to-use Packages: AMS Math Packages

- The $\mathcal{AMS}$ math packages `amsmath`, `amssymb` and `amsthm` extend LaTeX's math capabilities. E.g., compare `\frac` to `\tfrac` and `\dfrac`:

$$\frac{1}{2}x^2 = \dfrac{1}{2}x^2 = \tfrac{1}{2}x^2$$

$\frac{1}{2}x^2 = \dfrac{1}{2}x^2 = \tfrac{1}{2}x^2$

```
\[
    \frac{1}{2} x^2 = \dfrac{1}{2} x^2 =
    \tfrac{1}{2} x^2
\]
$
    \frac{1}{2} x^2 = \dfrac{1}{2} x^2 =
    \tfrac{1}{2} x^2
$
```

- The package `amsmath` also contains the `cases` and `dcases` environments:

$$F_n := \begin{cases} 0 & n = 0, \\ 1 & n = 1, \\ F_{n-1} + F_{n-2} & n \ge 2. \end{cases}$$

```
\[  F_n :=
\begin{cases}
0                 & n = 0, \\
1                 & n = 1, \\
F_{n-1} + F_{n-2} & n \ge 2.
\end{cases}  \]
```

- The $\mathcal{AMS}$ logo can be generated by means of `\AmS` or, if the `hologo` package was loaded, also by means of `\hologo{AmS}`. Similarly, `\hologo{AmSLaTeX}` generates $\mathcal{AMS}$-LaTeX.

Ready-to-use Packages: AMS Math Packages

- The package `amsmath` also supports various forms of matrices:

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \qquad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \qquad \begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix}$$

- The left-most matrix was generated by the `matrix` environment:

```
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix}
```

- Similarly, the environments `pmatrix`, `bmatrix`, `vmatrix` and `Vmatrix` generated the other matrices (in left-to-right order).
- One can also use the `\smallmatrix` environment to generate a decent inline matrix: $\left(\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right)$.
- The package `amsfonts` provides the command `\mathbb`, e.g., `\mathbb{R}`, which allows to generate symbols for the number sets: $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

Ready-to-use Packages: SI Units

- The `siunitx` package helps with the correct typesetting of SI-units (and even some non-SI-units): `\SI{10}{\mega\hertz}` generates 10 MHz, and `\SIrange{10}{100}{\hertz}` produces 10 Hz to 100 Hz.
- Similarly: 100° as result of `\ang{100}`, and -2×10^{-5} via `\num{-2e-5}`.
- The language-specific decimal marker can be set with the option `\sisetup{output-decimal-marker= {,}}`. Compare 3,14 µV m^{-2} to 3.14 µV m^{-2}. Both outputs were generated by means of the command `\SI{3.14}{\micro\volt\per\square\metre}`.
- The `siunitx` package also introduces a new column type `S` for the `tabular` environment:

	numbers
α	3.14
β	100.1234
γ	−0.001 234
δ	2×10^{-4}

```
\begin{tabular}{cS}
  \multicolumn{2}{c}{numbers} \\
     \hline
  $\alpha$ & 3.14 \\
  $\beta$  & 100,1234 \\
  $\gamma$ & -0,001234 \\
  $\delta$ & 2e-4
\end{tabular}
```

Ready-to-use Packages: Source-Code Listings

- The `listings` package allows to generate source-code listings:
  `\lstinputlisting[language=C,frame=tb]{code/horner.c}`

```c
/* Horner's Algorithm evaluates a polynomial of degree n at point x
 * @param p: array of n+1 coefficients
 * @param n: the degree of the polynomial
 * @param x: the point of evaluation
 * @return the evaluation result
 */
double evaluate(double *p, int n, double x)
{
  double h = p[n];

  for (int i = n - 1; i >= 0; --i)
    h = x * h + p[i];

  return h;
}
```

- It supports more than 75 languages, including C/C++, Java, Python and LaTeX.
- Note that the `listings` package comes with many options to influence the style and layout of a listing.

Ready-to-use Packages: Source-Code Listings

```
\begin{frame}[fragile]\frametitle{Ready-to-use Packages: Source-Code Listings}

\begin{itemize}
\item The \cmd{listings} package allows to generate source-code listings:
  \newline
      % We use \verb to include the LaTeX source code
      \verb+\lstinputlisting[language=C,frame=tb]{code/horner.c}+
\end{itemize}

\lstinputlisting[language=C,frame=tb]{code/horner.c}
\pause

\begin{itemize}
\item It supports more than 75 languages, including C/\Cpp, Java, Python
  and \LaTeX.
\item Note that the \cmd{listings} package comes with many options to
  influence the style and layout of a listing.
\pause
\end{itemize}
\end{frame}
```

- That listing of the LaTeX source code which generated the previous slide was obtained by means of the following `listings` commands:

```
\lstinputlisting[language={[latex]tex},frame=single,
                 numbers=none,keywordstyle=\color{red},
                 commentstyle=\color{blue}]{package.tex}
```

Ready-to-use Packages: Pseudocode Listings

- Several LaTeX packages support the formating of pseudocode, such as `program`, `algorithmic` and `algorithm2e`. E.g., with `algorithmic`:

Data: $x, a \in \mathbb{R}$
Result: $y \in \mathbb{R}$
begin
  repeat
    $y \leftarrow$ Compute(x, a)
    done $\leftarrow$ Check(x, a)
    if *not done* **then**
      $a \leftarrow 10 \cdot a$
      reset data structures
    else
      report done;
  until *(done OR* $a > 10^{-10}$*)*

```
\begin{algorithm}[H]
\DontPrintSemicolon
\KwData{$x,a\in\R$}
\KwResult{$y\in\R$}
\Begin{
  \Repeat{(done ~OR~
            $a > 10^{-10}$)}{
    $y \leftarrow $ Compute$(x,a)$\;
    done $\leftarrow$ Check$(x, a)$\;
    \If{not done}{
      $a \leftarrow 10 \cdot a$\;
      reset data structures\;
    }
    \Else{
      report done;
} } }
\end{algorithm}
```

Ready-to-use Packages: Latexdiff for Marking Changes to Documents

- The utility `latexdiff` makes it easy to markup and view changes made to a LaTeX document.
- It is a Perl script and requires an installation of Perl 5.8 or higher.
- To compare two versions of a document, named `old.tex` and `new.tex`, it suffices to run it as follows:

```
latexdiff old.tex new.tex > diff.tex
```

- The markup information is stored in `diff.tex`, which can be processed with any standard LaTeX compiler.
- Its default set-up is as follows:
  - Words that were removed are crossed out with a single line and colored red.
  - Words that were added are underlined with a squiggle and colored blue.
  - For changed equations, additions are marked with a blue color and removals are marked with a red color.
- Several options to influence how the markup shall be done . . .

Automated Processing of a LaTeX Document: `latexmk`

- `Latexmk` is a Perl script for automating the processing of a LaTeX document.
- It is a highly specialized sibling of the general-purpose `make` utility.
- It runs LaTeX (and related programs like BibTeX) the appropriate number of times in order to resolve all symbolic references.
- It has a reliable algorithm for detecting dependencies among the input files.
- It can also be instructed to start a previewer and then run LaTeX whenever a source file has changed.
- It supports the use of PDFTeX for generating a PDF output.
- Sample use:

```
latexmk [--pdf] --interaction=nonstopmode -pvc   foo.tex
```

- Instructions for setting the preferred viewers in `~/.latexmkrc`:

```
$dvi_previewer = 'start xdvi -watchfile 1.5';
$ps_previewer  = 'start gv --watch';
$pdf_previewer = 'start evince';
$pdf_mode = 1;                               # tex -> pdf
```

Trouble Shooting

- Always remember that LaTeX is nothing but a type setting system that has to rely on your commands.
- For instance, it cannot guess where you meant to insert a parenthesis but forgot to do so!
- Thus, it will bark about any syntactical error that it can detect.
- Also, note that syntactical correctness need not imply a logical correctness.
- For instance, LaTeX will be perfectly happy to set an entire book in `\tiny` type size, which may be different from what you intended to do.

Trouble Shooting Guidelines

1. Make sure that all parentheses occur in matching pairs. It is good practice to enter `{}` prior to entering anything between the parentheses. (Some editors support this and will automatically re-position the cursor.)
2. Similarly, all math delimiters need to occur in matching pairs.
3. Make sure that all `\begin` and `\end` commands occur in matching pairs. (Again, some editors support an easy entering of environment names.)
4. Rerun LaTeX frequently. The load that it will place on the CPU is no issue with modern computers, but it will help you tremendously when attempting to locate problems.
5. Consult the LaTeX Book and the LaTeX Companion. (Yes, indeed: RTFM!)
6. At all positions where one space or empty line is allowed, several spaces and empty lines are allowed. It will help your first attempts to locate a problem if your LaTeX file is formatted neatly!
7. Recall that a % sign starts a comment for LaTeX, and that it will ignore the rest of the line.

Drafting Figures and Generating Plots

Drawing Figures with LaTeX

- Simple figures can be generated using the *picture* environment of LaTeX:

```
\begin{picture}(width,height)(x-lower_left,y-lower_left)
   ...
\end{picture}
```

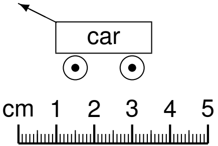
  with all coordinates being expressed in terms of `\unitlength`.
- The unit length can be set using the command `\setlength`. E.g., the following command sets the unit length to 5*mm*:
  `\setlength{\unitlength}{5mm}`.
- (*x-lower_left*, *y-lower_left*) specifies the coordinates of the lower-left corner of the picture. If absent, the lower-left corner has coordinates `(0,0)`.
- Two standard line widths are available within the picture environment: `\thinlines` and `\thicklines`.

Drawing Figures with LaTeX

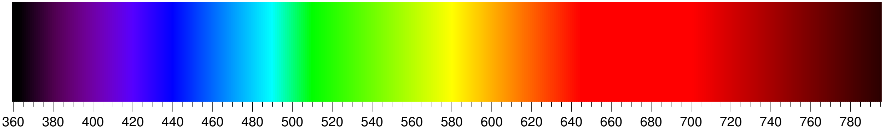
- The `\begin{picture}` command puts LaTeX into *picture mode*. The only things that can appear inside the picture environment are the commands `\put`, `\multiput`, `\qbezier`, and `\graphpaper`, and declarations such as `\thicklines`.
- The basic command for drawing is the `\put` command:
  `\put(`*x-coord*`,`*y-coord*`){`*picture object*`}`.
- Valid picture objects are text, (dashed) boxes, lines, arrows, (filled) circles, ovals:
  - `\put(`*x-coord*`,`*y-coord*`){`*my_text*`}`
  - `\put(`*x-coord*`,`*y-coord*`){\framebox(`*width*`,`*height*`){`*my_text*`}}`
  - `\put(`*x-coord*`,`*y-coord*`){\line(`*x-dir*`,`*y-dir*`){`*length*`}}`
  - `\put(`*x-coord*`,`*y-coord*`){\vector(`*x-dir*`,`*y-dir*`){`*length*`}}`
  - `\put(`*x-coord*`,`*y-coord*`){\circle{`*radius*`}}`
  - `\put(`*x-coord*`,`*y-coord*`){\oval(`*width*`,`*height*`)}`
- The reference point of a box is its lower-left corner.
- The box-drawing commands take one or two additional optional arguments for specifying the position of the text relative to the box: `l` (left), `r` (right), `t` (top), `b` (bottom). The default is to center the text horizontally and vertically within the box.
- Objects can be saved by means of the `\savebox` and reused with the `\usebox` command.
- Repeated patterns can be generated with the `\multiput` command.

Sample LaTeX Picture

```
\newcounter{cms}
\setlength{\unitlength}{0.5mm}
\begin{center}
  \begin{picture}(50,39)(30,0)
    \put(0,7){\makebox(0,0)[b]{cm}}
    \multiput(10,7)(10,0){5}
       {\addtocounter{cms}{1}\makebox(0,0)[b]{\arabic{cms}}}
    \put(15,20){\circle{6}}
    \put(30,20){\circle{6}}
    \put(15,20){\circle*{2}}
    \put(30,20){\circle*{2}}
    \put(10,24){\framebox(25,8){car}}
    \put(10,32){\vector(-2,1){10}}
    \multiput(1,0)(1,0){49}{\line(0,1){2.5}}
    \multiput(5,0)(10,0){5}{\line(0,1){3.5}}
    \thicklines
    \multiput(0,0)(10,0){6}{\line(0,1){5}}
    \put(0,0){\line(1,0){50}}
  \end{picture}
\end{center}
```



Sample LaTeX Picture Based on xcolor Package

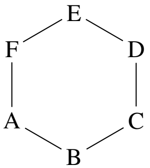


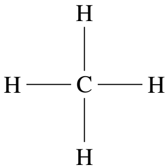

```
\newcount\WL
\setlength{\unitlength}{0.75pt}

\begin{picture}(460,60)(355,-10)
\sffamily \tiny \linethickness{1.25\unitlength} \WL=360
\multiput(360,0)(1,0){436}%
{{\color[wave]{\the\WL}\line(0,1){50}}\global\advance\WL1}
\linethickness{0.25\unitlength}\WL=360
\multiput(360,0)(20,0){22}%
{\picture(0,0)
\line(0,-1){5} \multiput(5,0)(5,0){3}{\line(0,-1){2.5}}
\put(0,-10){\makebox(0,0){\the\WL}}\global\advance\WL20
\endpicture}
\end{picture}
```

LaTeX Packages for Generating Special-Purpose Figures: Chemistry

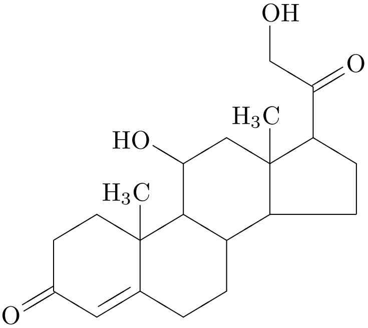
- LaTeX provides several ready-to-use packages for creating illustrations for specific applications.
- E.g., the `chemfig` packages allows to draw chemical structures:

```
\chemfig{A*6(-B-C-D-E-F-)}
\chemfig{C(-[:0]H)(-[:90]H)(-[:180]H)(-[:270]H)}
```





- A representation of Corticosterone, which is a 21-carbon steroid hormone produced in the cortex of the adrenal glands, can be generated by means of the `carom` package.

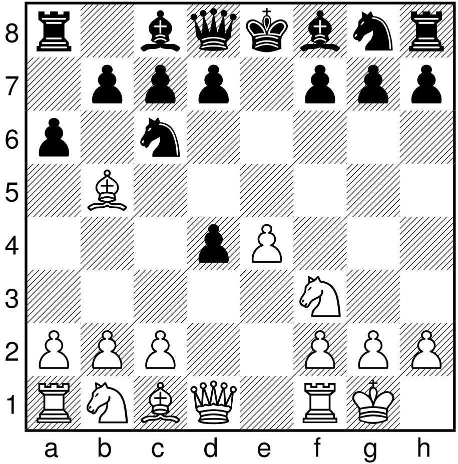


LaTeX Packages for Generating Special-Purpose Figures: Chess

- The following chessboard was produced by means of `skak` commands:

```
\newgame\mainline{1.e4 e5 2.Nf3 Nc6 3.d4 e5xd4 4.Bb5 a6 5.O-O}
\showboard
```

1 e4 e5 2 ♘f3 ♘c6 3 d4 e×d4 4 ♗b5 a6 5 O-O



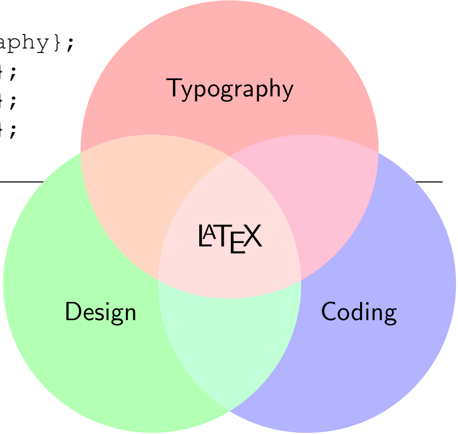
- Note that `skak` (Danish for chess) is able to handle the Forsyth-Edwards Notation (FEN), which is the standard notation for describing a specific board position of a chess game.

Ti*k*Z

- Ti*k*Z is a recursive acronym for "Ti*k*Z ist kein Zeichenprogramm".
- Ti*k*Z and its lower-level sibling, PGF, were created by Till Tantau in 2005.
- The PGF/Ti*k*Z interpreter can be used from within LaTeX and can produce output suitable for both PostScript and PDF.
- It comes with "enough" documentation. (Its manual has over 1300 pages!)
- In order to use Ti*k*Z, insert `\usepackage{tikz}` into the preamble and then put your Ti*k*Z code between `\begin{tikzpicture}` and `\end{tikzpicture}`.
- Note that the `tikzpicture` environment does not start a new line. Thus, it can be used in the running text.
- LaTeX commands can be used inside a `tikzpicture` environment.
- Ti*k*Z allows to generate drawings by means of commands. As with the `picture` environment of LaTeX, this quickly gets cumbersome.
- However, Ti*k*Z comes with lots of special-purpose libraries for drawing specific types of figures.
- Furthermore, several other packages can produce output for Ti*k*Z, such as Gnuplot, Mathematica, MATLAB and GeoGebra.

Sample TikZ Figure

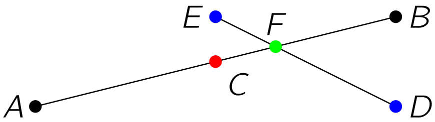
```
\begin{tikzpicture}
  \begin{scope}[blend group = soft light]
    \fill[red!30!white]   ( 90:1.2) circle (2);
    \fill[green!30!white] (210:1.2) circle (2);
    \fill[blue!30!white]  (330:1.2) circle (2);
  \end{scope}
  \node at ( 90:2)    {Typography};
  \node at ( 210:2)   {Design};
  \node at ( 330:2)   {Coding};
  \node [font=\large] {\LaTeX};
\end{tikzpicture}
```



- The default units of a Ti*k*Z figure are centimeters.
- Use the environment option `[scale=0.7]` to scale a Ti*k*Z figure to 70% of its original size.

Another Sample TikZ Figure

```
\begin{tikzpicture}
\coordinate[label=left:$A$] (A) at (0,0);
\coordinate[label=right:$B$] (B) at (4,1);
\draw (A) -- (B);
\fill (A) circle (2pt);  \fill (B) circle (2pt);
\coordinate[label=-45:$C$] (C) at ($(A)!0.5!(B)$);
\fill[red] (C) circle (2pt);
\coordinate[label=right:$D$] (D) at (4,0);
\coordinate[label=left:$E$] (E) at (2,1);
\draw (D) -- (E);
\fill[blue] (D) circle (2pt);  \fill[blue] (E) circle (2pt);
\coordinate[label=90:$F$] (F) at (intersection of A--B and D--E);
\fill[green] (F) circle (2pt);
\end{tikzpicture}
```



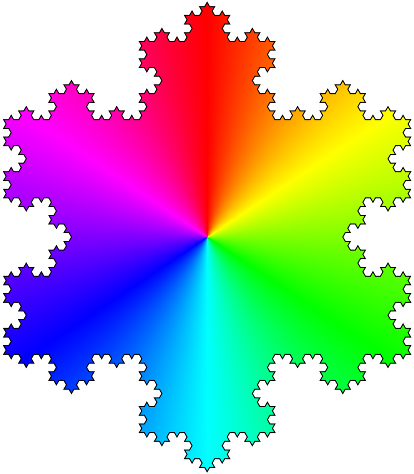
Sample TikZ Library: Lindenmayer Systems

```
\pgfdeclarelindenmayersystem{Koch curve}{
  \rule{F -> F-F++F-F}}
\begin{tikzpicture}
   \shadedraw[shading=color wheel]
   [l-system={Koch curve, step=2pt, angle=60, axiom=F++F++F,
              order=4}] lindenmayer system -- cycle;
\end{tikzpicture}
```

```
\pgfdeclarelindenmayersystem{Fractal plant}{
  \rule{X -> F-[[X]+X]+F[+FX]-X}
  \rule{F -> FF}}
\begin{tikzpicture}
   \draw [green!50!black, rotate=90]
   [l-system={Fractal plant, axiom=X, order=6, step=2pt,
              angle=25}] lindenmayer system;
\end{tikzpicture}
```

[Code credit: https://texample.net]

Sample Ti*k*Z Library: Lindenmayer Systems



TGIF

- `Tgif` is an Xlib-based interactive 2D *drawing tool* that allows the user to draw and manipulate objects under the X Window System.
- `Tgif` supports the hierarchical construction of drawings, and an easy navigation between sets of drawings.
- `Tgif` is purely based on Xlib. It requires a three-button mouse.
- The source code for `tgif` is freely available on the WWW.
- `Tgif` is free for non-commercial applications.
- `Tgif` supports a variety of primitive objects.
- Objects can be grouped together to form a *grouped* object.
- Commands applied to a grouped object are applied to all sub-objects of the group.
- Typically, `tgif` objects are stored in files with an .obj extension (referred to as an *object file*). (So-called “building-block” objects are stored in files with a `.sym` extension (referred to as a *symbol file*).)
- Both types of files are stored in the form of Prolog facts. Prolog code can be written to interpret the drawings!

TGIF File Input/Output

- Tgif can generate output in several different formats:
  - PS,
  - EPS,
  - PDF (needs ps2pdf from the ghostscript package),
  - X11 bitmap (XBM), or XPM for color output),
  - plain ASCII text.
- X11 bitmap files, certain forms of X11 pixmap files (such as the one generated by tgif), and EPS can be imported into tgif and can be represented as tgif primitive objects.
- Files in other raster formats (e.g, PNG, JPEG, TIFF, etc.) can also be imported into and exported from tgif if external tools can be used to convert them into X11 XBM/XPM files.
- Tgif can capture (portions of) a screen and input it as a tgif object.
- By default, tgif drawings are formatted for printing on letter-size paper. Tgif offers a compile-time flag in order to make DIN A4 the default paper size.

Ipe

- Ipe is a drawing editor that generates drawings in XML, PDF or EPS format.
- Ipe is particularly geared towards making sophisticated 2D figures that serve as illustrations of geometric concepts and algorithms.
- It offers most standard features of a drafting package, plus a few “CAD-like” features.
- The Ipe interface allows keyboard shortcuts which (mostly) are equivalent to the shortcuts of Emacs.
- Objects supported include (poly)lines, polygons, splines, splinegons, circles and ellipses, circular arcs, rectangles, and marks. Bitmaps are supported, too.
- Ipe is written in standard C++, based on the STL.
- The GUI is implemented using the portable toolkit Qt, and, thus, can be compiled for Unix, Windows, and Mac OS X.
- Ipe is free software.

Ipe's Main Features

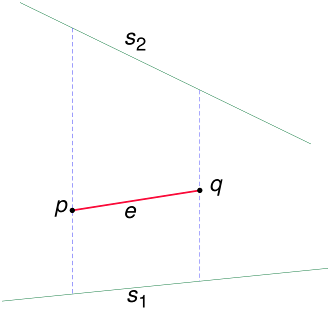
- Text is entered as LaTeX source code, and displayed in the display as it will appear in the figure.
- The text model is based on Unicode, and has been tested with Korean, Chinese, and Japanese. German "Umlaut" are supported.
- Ipe can input/output (its own extension of) PDF files, which are readily included into LaTeX by the same means as any other PDF documents are included.
- It embeds its own information very cleverly such that those files are regarded as perfectly valid files by, say, `evince`.
- It can also export EPS files, which can be included into LaTeX or viewed by, say, `gv`. (Earlier versions of Ipe could also input EPS files.)
- Ipe can also input/output XML tags stored in an `.ipe` file. Its document type definition (DTD) is given by `ipe.dtd`.
- Ipe XML files are the means of choice for generating input for Ipe by XML-aware applications.
- Importing PDF files: `pdftoipe`.

Special Features of Ipe

- Ipe is extendible. One can easily interface personal editing functions, so-called *Ipe extensions* or *Ipe User Macros* (IUMs), with Ipe.
- One of the nicest features of Ipe is the possibility to have the mouse *snap* to other objects. That is, the user can make certain objects in the drawing canvas *magnetic*, which makes it very easy to align an object under construction to other objects.
- If the cursor is too far away from the nearest interesting object then the cursor will not snap. The snapping threshold can be changed in Ipe's configuration window.
- Ipe supports three types of snapping: *grid snapping* (to grid points), *context snapping* (to vertices, boundaries, intersections, circle centers), and *directional/angular snapping*.

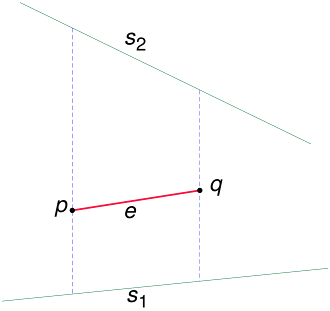
Sample Snapping in Ipe

- Suppose we are given the segments s_1, s_2 and e, with end points p and q, and want to add vertical extensions through p and q between s_1 and s_2.



Sample Snapping in Ipe

1. Turn on vertex snapping, e.g., by pressing F4, and go into polyline mode.
2. Move cursor near p and set the coordinate origin by pressing F1.
3. Turn on directional snapping by pressing F5, and angular snapping (F8).
4. Go near s_1 and click left mouse button, then go near s_2 and click right mouse button.
5. Go near q, re-set the coordinate origin by pressing F1, and draw the second vertical line.
6. Turn off the coordinate system by pressing CTRL-F1.



- Note that pressing F1 at point p and then pressing F2 at point q will set the coordinate origin at p and will align one coordinate axis with the line through p and q.

PSfrag for Generating LaTeX Symbols

- While `Tgif` can generate EPS/PS output that is suitable for inclusion into a LaTeX document, it cannot generate all the (mathematical) symbols that LaTeX supports.
- PSfrag is a set of LaTeX macros for overlaying EPS/PS figures (or any other PS text) with fragments of LaTeX.
- More precisely, the PSfrag macros allow specific pieces of PS text (so-called "tags") in a PS figure to be replaced with arbitrary fragments of LaTeX. When the document is latex'ed and dvips'ed, each piece of PS text is replaced by the properly sized, aligned, and rotated LaTeX text.
- In this way, Greek letters, super- and subscripts and mathematical symbols can be used in PS files with a typography that is consistent with the rest of the LaTeX document.
- PSfrag requires a recent version of LaTeX.
- A compatible DVI-to-PS driver is required, too. PSfrag works best with `dvips`, the DVI-to-PS driver from Radical Eye Software.
- Note that the file `psfrag.sty` has to be installed in a location searched by the LaTeX search path for macros. For `kpathsea`-based systems such as `teTeX`, this path is determined by the `TEXINPUTS` environment variable.
- Also, the DVI-to-PS driver has to be able to find the file `psfrag.pro`.

PSfrag and LaTeX

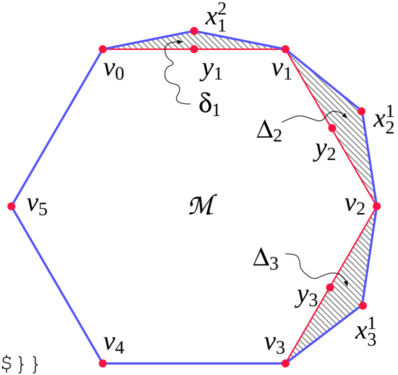
- For each tag word in the EPS/PS file, one adds a command to the LaTeX document to specify how this tag is to replaced, as follows:

  `\psfrag{`tag`}[`posn`][`psposn`][`scale`][`angle`]{`LaTeX text`}`

- All data given in brackets `[]` is optional and is used to specify the exact position and orientation of the LaTeX text with respect to the bounding box of the tag string. (See the manual for details.)
- Any text that is not mentioned in a `\psfrag` command will not be replaced. Hence, PS and LaTeX text can be freely mixed.
- A `\psfrag` replacement will remain in effect until its surrounding environment is exited.
- Thus, one can define global `\psfrag` commands which will apply to every figure of a LaTeX file, or one can define `\psfrag` commands inside an environment (e.g., a `figure` environment) which will apply to only one EPS file.
- Most DVI previewers (such as `xdvi`) are incapable of displaying the replaced text correctly.

Sample PSfrag Code

```
\psfrag{v0}{{\Large $v_0$}}
\psfrag{v1}{{\Large $v_1$}}
\psfrag{v2}{{\Large $v_2$}}
\psfrag{v3}{{\Large $v_3$}}
\psfrag{v4}{{\Large $v_4$}}
\psfrag{v5}{{\Large $v_5$}}
\psfrag{x12}{{\Large $x_1^2$}}
\psfrag{x21}{{\Large $x_2^1$}}
\psfrag{x31}{{\Large $x_3^1$}}
\psfrag{y1}{{\Large $y_1$}}
\psfrag{y2}{{\Large $y_2$}}
\psfrag{y3}{{\Large $y_3$}}
\psfrag{delta1}{{\Large $\delta_1$}}
\psfrag{Delta1}{{\Large $\Delta_1$}}
\psfrag{Delta2}{{\Large $\Delta_2$}}
\psfrag{Delta3}{{\Large $\Delta_3$}}
\psfrag{hex}{{\Large $\cal{M}$}}
\includegraphics{grasp_hex}
```



Convert Utility

- The utility `convert`, which is part of the `ImageMagick` suite of tools, lets you convert between image formats.
- It supports reading and/or writing more than 100 major image formats.
- It can also perform other operations on an image, e.g., resize, crop, flip, ...
- In the simplest form of its usage, one specifies the input file followed by the output file (with appropriate extensions):
  `convert label.xwd label.png`
- But it is more than just a simple converter. The following example was taken from `www.imagemagick.org/script/command-line-processing.php`:

```
convert label.png +matte
       \( +clone -shade 110x90 -normalize -negate +clone
                -compose Plus -composite \)
       \( -clone 0 -shade 110x50 -normalize -channel BG
                -fx 0 +channel -matte \)
       -delete 0 +swap -compose Multiply
       -composite "button.png"
```

- This command transforms  to .

xwd – Making X11 Screen Dumps

- `Xwd` is a utility for storing X11 window images in a specially formatted dump file.
- The target window is selected by clicking the mouse pointer in the desired window. The keyboard bell is rung once at the beginning of the dump and twice when the dump is completed.
- A `.xwd` file can then be read by various other X11 utilities (such as `gimp`) for redisplay, printing, editing, formatting, archiving, image processing, etc.
- `Xwd` is part of the standard X11 distribution.
- Sample command sequence for dumping a window into a file `foo.xwd`, and for converting it into a PostScript file:

```
xwd > foo.xwd
xpr -device ps -portrait  -psfig foo.xwd > foo.ps
```

- If `xpr` is not available, but `convert` is available, then the following command will work:

```
convert foo.xwd foo.ps
```

- Several packages for drafting and image manipulation also support capturing part or all of an X11 display.

Gnuplot

- Gnuplot is a command-line driven interactive plotting tool.
- It can plot 2D and 3D graphs, with data read from files, and can handle plots of built-in or user-defined functions.
- Input for LaTeX can be generated by instructing gnuplot to output its plot in the `EPS` format:

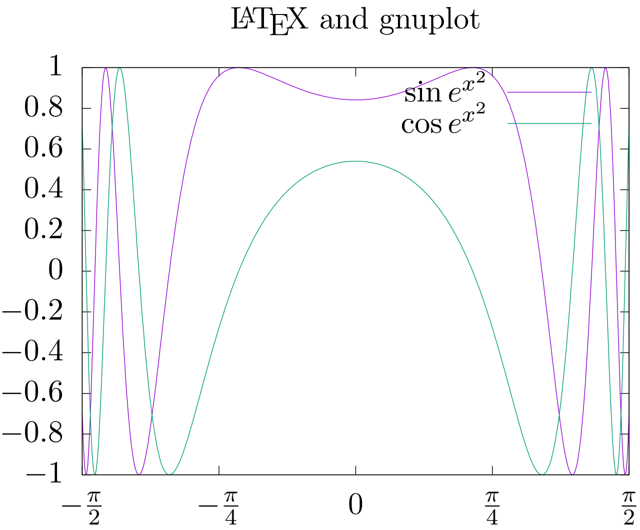
```
set terminal epslatex
set output 'gnuplot.tex'
```

- Running gnuplot on this code yields the output files `gnuplot.tex` and `gnuplot-inc.eps`.
- The file `gnuplot.tex` is processed by `latex` (or `pdflatex`) to generate the file `gnuplot.eps` (`gnuplot.pdf`, resp.), which is then included into another LaTeX document by means of an `\includegraphics` command.
- Advantages over other methods (discussed on the subsequent slides):
  - No data file needs to be shipped when distributing the LaTeX document.
  - No computations are necessary when the LaTeX document is processed.

Sample Gnuplot Code

```
set terminal epslatex size 9cm,7cm color colortext standalone
set output 'gnuplot.tex'
set title '\LaTeX\ and gnuplot'
set xrange [-pi/2:pi/2]
set xtics ('$-\frac{\pi}{2}$' -pi/2, \
           '$-\frac{\pi}{4}$' -pi/4, '0' 0, \
           '$\frac{\pi}{4}$' pi/4, \
           '$\frac{\pi}{2}$' pi/2)
set samples 500
f(x)=sin(exp(x**2))
g(x)=cos(exp(x**2))
plot f(x) title '$\sin e^{x^2}$', \
     g(x) title '$\cos e^{x^2}$'
```

Gnuplot Output Included into LaTeX



Gnuplottex: Gnuplot Envoked within LaTeX

- The package `gnuplottex` allows to call Gnuplot from within a LaTeX document.
- It supports the use of LaTeX commands for axis labeling, including the use of SI units via the `siunitx` package.
- `\usepackage{gnuplottex}` goes into the preamble.
- The Gnuplot code is extracted from the document and written to a Gnuplot file `.gnuplot`. Then this file is processed by Gnuplot and prepared for inclusion into the LaTeX document.

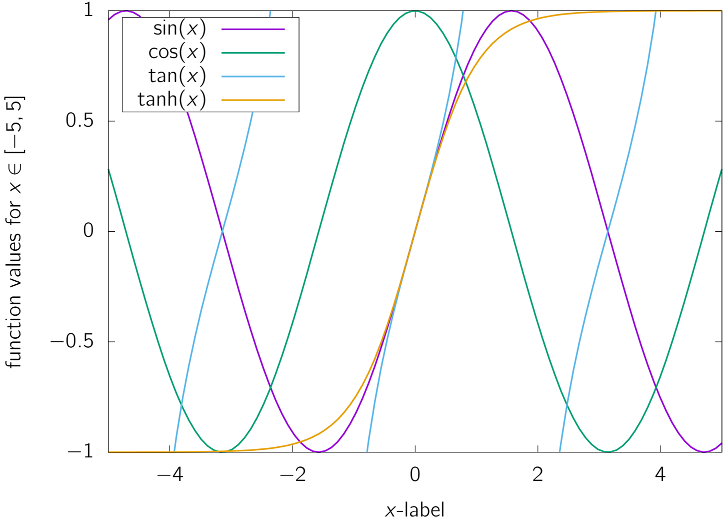
Caveats

- LaTeX needs to be allowed to call external programs!
- Use the run-time option `--shell-escape` or even `--shell-escape --enable-write18 -interaction=nonstopmode`.

Gnuplottex: Gnuplot Envoked within LaTeX

```
\begin{gnuplot}[terminal=epslatex, terminaloptions=color]
set key box top left
set key width 2
set key opaque
set sample 100
set xr [-5:5]
set yr [-1:1]
set xlabel '$x$-label'
set ylabel 'function values for $x\in[-5,5]$'
plot sin(x)  w l lc 1 lw 3 t '$\sin(x)$',\
     cos(x)  w l lc 2 lw 3 t '$\cos(x)$',\
     tan(x)  w l lc 3 lw 3 t '$\tan(x)$',\
     tanh(x) w l lc 4 lw 3 t '$\tanh(x)$'
\end{gnuplot}
```

Gnuplottex: Gnuplot Envoked within LaTeX



tbcolrgb1,1,1

PGFPLOTS

- The PGFPLOTS package is a tool to produce diverse types of plots directly within LaTeX.
- It is based on PGF/TikZ.
- Put `\usepackage{pgfplots}` into the preamble.

Caveat

Documents that contain multiple complex figures generated by means of PGFPLOTS may take a considerably larger amount of time to compile!

- Hence, in order to allow re-using figures that had already been generated, it is advisable to externalize those figures by putting the following two lines into the preamble:

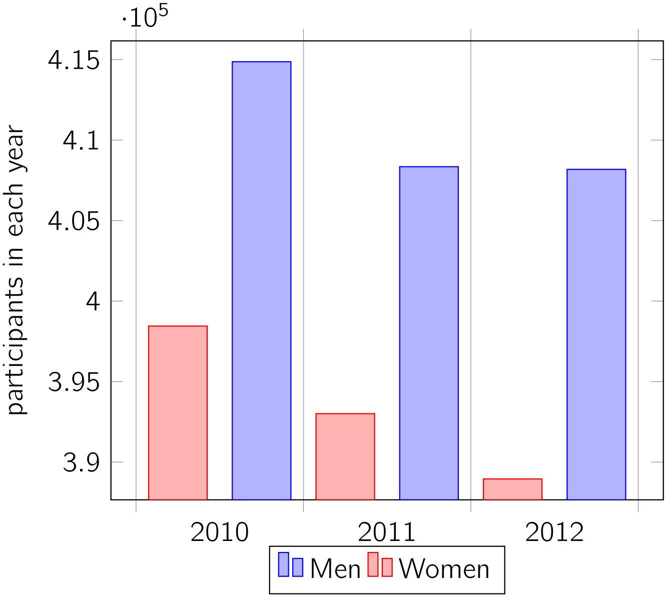
```
\usepgfplotslibrary{external}
\tikzexternalize
```

PGFPLOTS: Bar Graph

```
\begin{tikzpicture}
\begin{axis}[
  x tick label style={
    /pgf/number format/1000 sep=},
  ylabel=participants in each year,
  enlargelimits=0.05,
  legend style={at={(0.5,-0.1)},
  anchor=north,legend columns=-1},
  ybar interval=0.7,
]
\addplot
  coordinates {(2012,408184) (2011,408348)
      (2010,414870) (2009,412156)};
\addplot
  coordinates {(2012,388950) (2011,393007)
    (2010,398449) (2009,395972)};
\legend{Men,Women}
\end{axis}
\end{tikzpicture}
```

[Code credit: de.overleaf.com]

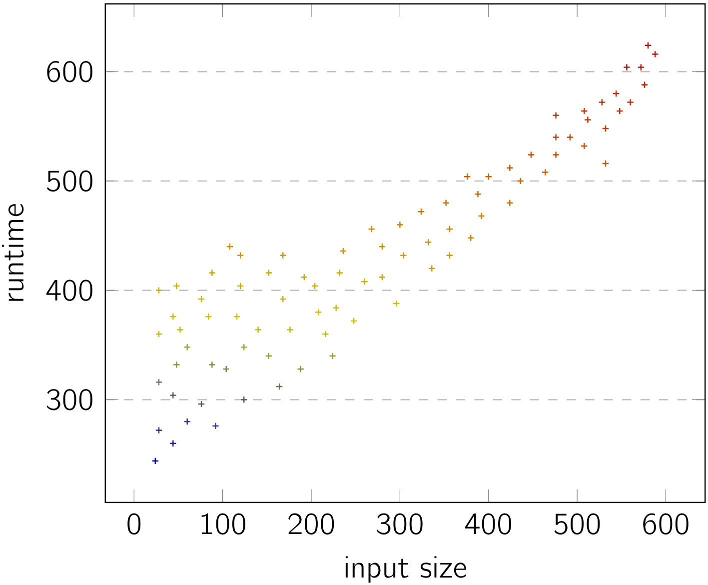
PGFPLOTS: Bar Graph



PGFPLOTS: Scatter Plot

```
\begin{tikzpicture}
\begin{axis}[
    enlargelimits=true,
    xlabel={input size},
    ylabel={runtime},
    ymajorgrids=true,
    grid style=dashed,
]
\addplot+[
    only marks,
    scatter,
    mark=+,
    mark size=1pt]
table[meta=cputime]
{data/cputimes.txt};
\end{axis}
\end{tikzpicture}
```

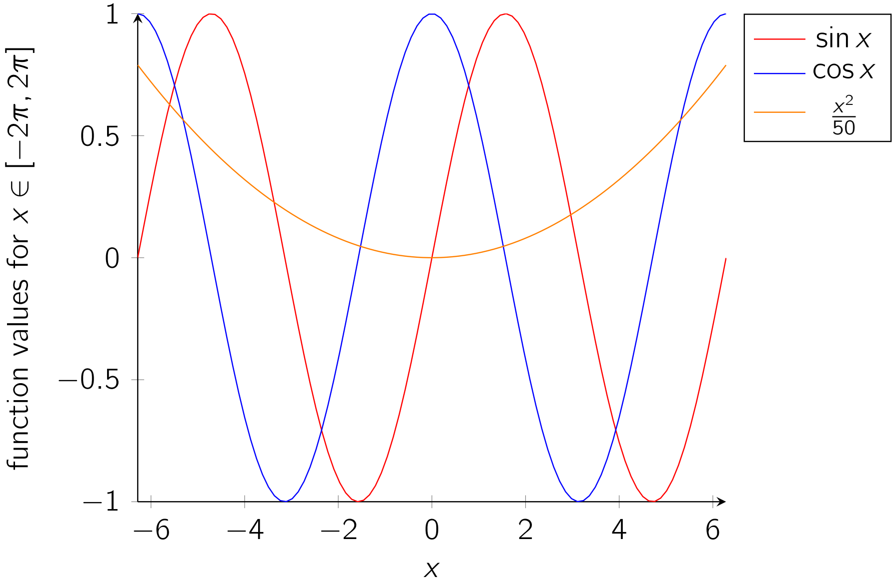
PGFPLOTS: Scatter Plot



PGFPLOTS: Graphs of Functions

```
\begin{tikzpicture}
\begin{axis}[
    domain=-2*pi:2*pi,
    samples=100,
    axis lines = left,
    xlabel = {$x$},
    ylabel = {function values for $x\in[-2\pi,2\pi]$},
    legend pos=outer north east
]
\addplot[color=red]   {sin(deg(x))}; \addlegendentry{$\sin x$}
\addplot[color=blue]  {cos(deg(x))}; \addlegendentry{$\cos x$}
\addplot[color=orange]{x^2/50};      \addlegendentry{$\frac{x^2}{50}$}
\end{axis}
\end{tikzpicture}
```

PGFPLOTS: Graphs of Functions

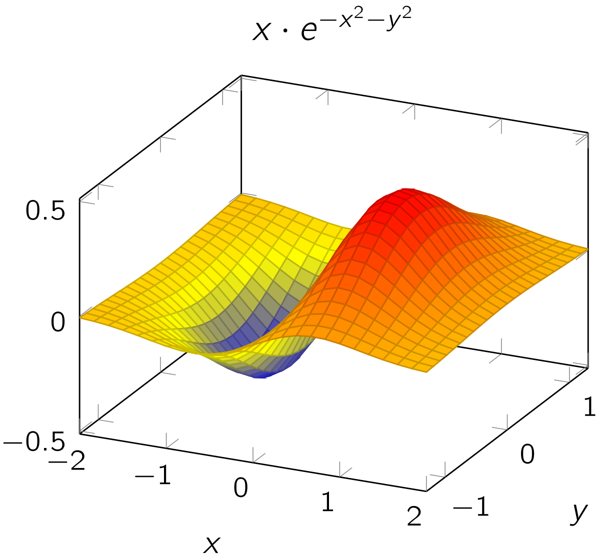


PGFPLOTS: 3D Graphs of Functions

```
\begin{tikzpicture}
\begin{axis}[
    title={$x \cdot e^{-x^2-y^2}$},
    xlabel=$x$, ylabel=$y$,
    small,
]
\addplot3[
    surf,
    domain=-2:2,
    domain y=-1.3:1.3,
]
{exp(-x^2-y^2)*x};
\end{axis}
\end{tikzpicture}
```

[Code credit: manual of PGFPLOTS at `ctan.org/pkg/pgfplots`]

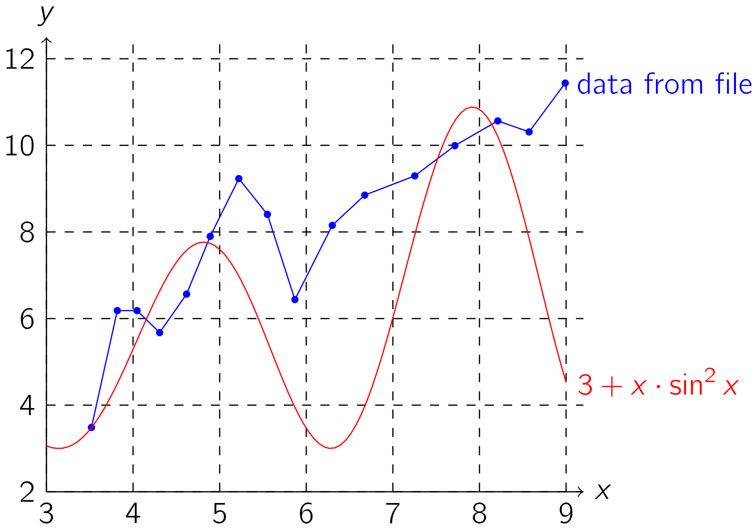
PGFPLOTS: 3D Graphs of Functions



TikZ and Gnuplot

```
\begin{tikzpicture}[x=1cm,y=0.5cm,domain=3:9,samples=100]
  \def\xmin{3}
  \def\ymin{2}

  \draw[dashed, ystep=2, xstep=1] (\xmin,\ymin) grid (9.2,12.5);

  \draw[->] (\xmin,\ymin) -- (9.2,\ymin) node[right] {$x$};
  \draw[->] (\xmin,\ymin) -- (\xmin,12.5) node[above] {$y$};

  \foreach \x in {3,4,...,9}   \node at (\x,\ymin) [below] {\x};
  \foreach \y in {2,4,...,12}  \node at (\xmin,\y) [left] {\y};

  \draw[color=blue] plot[mark=*,mark size=1pt] file {data/data.txt}
      node [right] {data from file};

  \draw[color=red] plot[id=xsinx]
  function{3+ x * sin(x) * sin(x)}
     node [right] {$3 + x \cdot \sin^2 x$};
\end{tikzpicture}
```

TikZ and Gnuplot



TikZ: Standalone Run

Caveat

As with similar approaches that generate graphics data "on the fly", running Gnuplot from within TikZ may result in a substantially increased compilation time.

- Hence it is advisable to include a TikZ picture into a LaTeX document by
  1. generating that picture in a wrapper file of its own,
  2. loading the `standalone` package into the main document with the option `mode=buildnew`, and
  3. using `\includestandalone` instead of `\includegraphics`.
- This will compile all included standalone files automatically as graphics files and build these graphics if the source file is newer than the existing graphics file.
- Of course, you need `--shell-escape` to be enabled to allow the main LaTeX run to envoke further LaTeX compilers.

Wrapper File for Standalone Run

```
% wrapper.tex
%
\documentclass[crop,tikz]{standalone}
% in theory, tikz package should be loaded by
% the 'tikz' option
\usepackage{tikz}

%\usetikzlibrary{...}

\begin{document}
  \begin{tikzfigure}
    \coordinate[label=left:$A$] (A) at (0,0);
    \coordinate[label=right:$B$] (B) at (4,1);
    \draw (A) -- (B);

    ...
  \end{tikzfigure}
\end{document}
```

Main File for Standalone Run

```
% main.tex  --------  requires --shell-escape!
%
\documentclass{article}
\usepackage[mode=buildnew]{standalone}
\usepackage{tikz}

\begin{document}

  \includestandalone[options]{wrapper}

\end{document}
```

pdfLaTeX and Recent LaTeX Developments

pdfTEX and pdfLATEX

- Quote from the pdfTEX user manual:

  "The pdfTEX package is an extension of LATEX/TEX that can create PDF *directly from TEX/LATEX source files and improve/enhance the result of TEX typesetting with the help of* PDF*."*

- It produces PDF output that looks (virtually) identical to the DVI output.
- The `pdftex` command uses the equivalent of the plain TEX format, and the `pdflatex` command uses the equivalent of the LATEX format.

Macro Packages Supported by pdfLaTeX

- The typical use of the pdfLaTeX-package is with pre-generated formats for which PDF output has been enabled.
- Currently, all mainstream macro packages offer pdfLaTeX support in some way.
- When using such a package, it makes sense to turn on this support in the appropriate way, otherwise one cannot be sure whether things will be set up correctly.
- For instance, the `hyperref` package has substantial support for pdfLaTeX. The user merely needs to load `hyperref` with the `pdftex` option, and all cross-references will be converted to PDF hypertext links.

  ```
  \usepackage[pdftex]{hyperref}
  ```

  Bookmarks can be created to match the table of contents.
- Similarly, the LaTeX packages `graphicx` and `xcolor` have options for pdfLaTeX, which allow the use of the standard commands for color, text rotation, and graphics inclusion.
- PDF support can also be turned on globally:

  ```
  \documentclass[...,pdftex,...]{...}
  ```

Manual Cross-Referencing, Bookmarks, URLs

- Manual tagging for cross-referencing:

```
\hyperlink{myref}{Clicking here will take you to ...}
\hypertarget{myref}{... this target}
```

- Similarly, bookmarks can be set manually:

```
\pdfbookmark[level]{bookmark text}{myref}
```

- Most PDF viewers support the handling of URLs: Clicking on an `href` construct will start a web browser and take you to the page specified:

```
\href{https://www.cosy.sbg.ac.at}{Departmental home page}
\url{www.cosy.sbg.ac.at}
```

- Similarly for other descriptors:

```
\href{ftp://...}
\href{mailto:...}
\href{run:...}
```

- Menu functions of `acroread` can be accessed via links, e.g.:

```
\Acrobatmenu{Print}{print this document}
```

pdfLaTeX and PostScript

- Native pdfLaTeX supports the inclusion of pictures in PNG, JPEG, TIFF and PDF format.
- EPS files can be converted to PDF by Ghostscript, Acrobat Distiller, or by the PERL script `epstopdf`; PS files can be converted by `ps2pdf`.
- If you want to be able to build PS and PDF files from the same source, leave off the file extensions from the image filenames in the `\includegraphics` calls.
- Since TeX Live 2010, pdfLaTeX automatically converts EPS files to PDF, via the `epstopdf` package. (Same for XeLaTeX.)
- Similarly, the `pstricks` package can be used by an up-to-date TeX distribution with the following command:

  ```
  \usepackage[pdf]{pstricks}
  ```

- The standard `\psfrag` replacements can be used with the command

  ```
  \usepackage{auto-pst-pdf}
  ```

  provided that shell escapes are allowed: For TeX Live we use

  ```
  pdflatex --shell-escape   ...
  ```

  while the command-line option `--enable-write18` should work for MiKTeX.

LaTeX Beamer Class

- Created by Till Tantau in 2003, and distributed via the Comprehensive TeX Archive Network (CTAN, `ctan.org`).
- The LaTeX beamer class allows to create slides directly within LaTeX, with no need to resort to postprocessing by other software packages.
- It can be used with `pdflatex`, but also with `dvips`.
- Professional layouts and sophisticated overlays can be achieved.
- These slides (and the corresponding handouts) were prepared using `pdflatex` and the LaTeX beamer class — based on one set of source files for all three PDF outputs.
- Note that the word "beamer" is a pseudo-anglicism.

Tools for Viewing Beamer-Class Slides

- Of course, any standard PDF viewer will (or should) be good enough — beamer-class slides are standard PDF files!
- PDF viewers geared towards presentations:

pdfpc — PDF presenter console — is a GTK-based viewer which uses multi-monitor output to provide meta information to the speaker during the presentation.

BeamerPresenter also offers dual-monitor output and is similar to `pdfpc`; it tries to implement all features available in the LaTeX `beamer` class; in addition, it supports tools like pointer, highlighter, pen and eraser.

pympress, dspdfviewer, ipepresenter are similar to `pdfpc`.

There is theory, and there is practice . . .

I am yet to see a presentation tool that can deal with all slides generated by pdfLaTeX!

LaTeX Beamer Class: Structuring

- Section:

```
\section{...}
```

- Subsection:

```
\subsection{...}
\subsubsection{...}
```

- Slide:

```
\begin{frame} ... \end{frame}
```

- Block:

```
\begin{block} ... \end{block}
\begin{alertblock} ... \end{alertblock}
\begin{exampleblock} ... \end{exampleblock}
```

- (Standard) LaTeX lists:

```
\begin{itemize} ... \end{itemize}
\begin{enumerate} ... \end{enumerate}
\begin{description} ... \end{description}
```

LaTeX Beamer Class: Structuring

- Multiple columns:

Block: Lorem ...

Lorem ipsum dolor sit amet, ...

Warning: Pseudo Latin

Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ...

Example

Lorem ipsum dolor sit amet, ...

```
\begin{columns}[c]
  \column{0.25\textwidth}
  \begin{block}{Block: Lorem ...}
      Lorem ipsum dolor sit amet, ...
  \end{block}
  \column{0.45\textwidth}
  \begin{alertblock}{Warning: Pseudo Latin}
      Lorem ipsum dolor sit amet, ...
  \end{alertblock}
  \column{0.15\textwidth}
  \begin{exampleblock}{Example}
      Lorem ipsum dolor sit amet, ...
  \end{exampleblock}
\end{columns}
```

LaTeX Beamer Class: Special Frames

- Title page:

```
\begin{frame}[plain]
    \titlepage
\end{frame}
```

- Table of content:

```
\begin{frame}\frametitle{Contents} % or something else
                                   % as frame title
    \tableofcontents[subsectionstyle=hide,
                     subsubsectionstyle=hide]
\end{frame}
```

- At the begin of a section:

```
\AtBeginSection[]{
   \begin{frame}
      \tableofcontents[sectionstyle=show/hide,
                       hideothersubsections,
                       subsubsectionstyle=hide/hide/hide]
   \end{frame}
}
```

LaTeX Beamer Class: Partial Build

- The `\pause` command can be used for simple partial builds of a page.
- This will build a page from top to bottom in sequential order.
- Using the `\verb` command or the `verbatim` environment in conjunction with `\pause` requires to specify the option `[fragile]` after `\begin{frame}`!

```
\begin{frame}[fragile]\frametitle{\LaTeX\ Beamer Class: ...}

\begin{itemize}
  \item The \verb#\pause# command can be used for simple ...
\pause
  \item This will build a page from top to bottom ...
\pause
  \item Using \verb#\verb# or the \verb#verbatim# ...
\end{itemize}

\end{frame}
```

- The obvious advantage of this scheme is its simplicity. The obvious disadvantage is its rigidity.

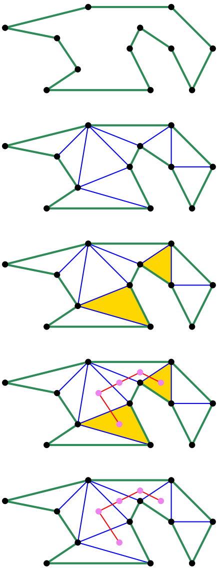
LaTeX Beamer Class: Overlays

Overlay 12345

- only on overlay 1
- blue text only on overlays 1–3
- only on overlay 2
- on overlay 3 and all subsequent overlays
- The overlay specification can be used with quite a few other LaTeX commands, too. E.g., `\includegraphics<1|handout:0>{...}`.
- There are four other commands to switch material on and off: `\only`, `\onslide`, `\uncover`, `\visible`.

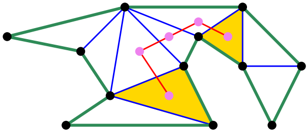
```
\begin{itemize}
  \item<1> only on overlay 1
  \item<1-> \color<1-3>{blue} blue text only on overlays 1--3}
  \item\alt<2>{only on overlay 2}{on all overlays except 2}
  \item<3-> on  overlay 3 and all subsequent overlays
  \item<4-> The overlay specification ...
  \item<5-> There are four other commands ...
\end{itemize}
```

LaTeX Beamer Class: Overlays in Conjunction with Ipe's Views



- Suppose that we want to explain a concept that requires to display similar figures repeatedly. E.g.,
  1. a polygon,
  2. its triangulation,
  3. two specific triangles,
  4. the triangles and a path between them, and
  5. only the path.
- Ipe allows to group contents of a figure into layers which can be turned on and off indvidually.
- An Ipe *view* is a list of layers that are turned on.
- When saved as a PDF file, each view becomes a single page in the PDF.
- Individual pages of such a PDF can be incorporated into a LaTeX "beamer" document by resorting to the `page` option of the `\includegraphics` command:
  ```
  \includegraphics<1->[page=1,...]{views}
  \includegraphics<2->[page=2,...]{views}
  ```
- This makes it easy to "build" a figure or an animation, without (re-)drawing the figure multiple times.

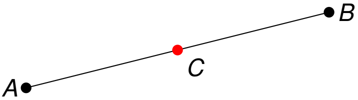
LaTeX Beamer Class: Overlays in Conjunction with Ipe's Views



1. A polygon,
2. its triangulation,
3. two specific triangles, and
4. the triangles and a path between them.

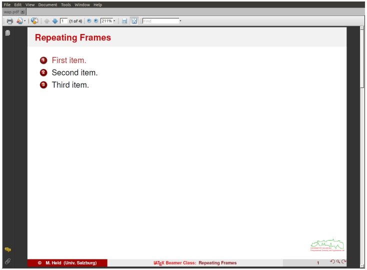
```
\begin{columns}[c]
 \column{0.4\textwidth}
    \includegraphics<1|handout:0>[page=1]{views}%
    \includegraphics<2|handout:0>[page=2]{views}%
    \includegraphics<3|handout:0>[page=3]{views}%
    \includegraphics<4|handout:1>[page=4]{views}%
  \column{0.6\textwidth}
  \begin{enumerate}
    \item<1-> A polygon,
    \item<2-> its triangulation,
    \item<3-> two specific triangles, \only<4->{and}
    \item<4-> the triangles and a path between them.
  \end{enumerate}
\end{columns}
```

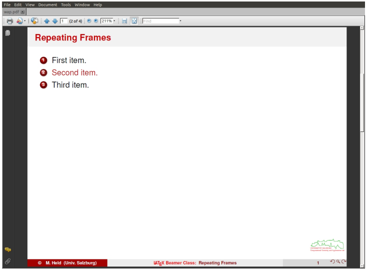
LaTeX Beamer Class: Overlays in Conjunction with Ti*k*Z

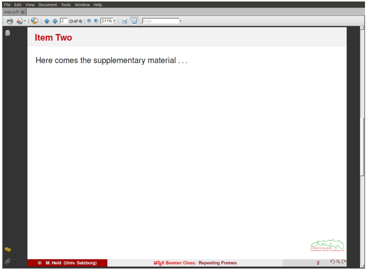


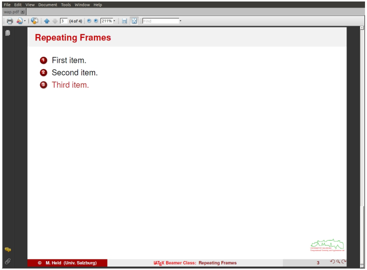

```
\begin{tikzpicture}
  \coordinate[label=left:$A$] (A) at (0,0);
  \coordinate[label=right:$B$] (B) at (4,1);
  \draw (A) -- (B);
  \fill (A) circle (2pt);
  \fill (B) circle (2pt);
  \onslide<2->{
    \coordinate[label=-45:$C$] (C) at ($(A)!0.5!(B)$);
    \fill[red] (C) circle (2pt);
  }
\end{tikzpicture}
```

LaTeX Beamer Class: Repeating Frames







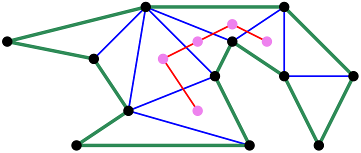


LaTeX Beamer Class: Repeating Frames

- The following LaTeX code produces the four slides shown on the previous slide:

```
\begin{frame}<1-2>[label=myframe]\frametitle{Repeating Frames}

  \begin{enumerate}
     \item<alert@1> First item.
     \item<alert@2> Second item.
     \item<alert@3> Third item.
  \end{enumerate}
\end{frame}

\begin{frame}\frametitle{Item Two}

  Here comes the supplementary material $\ldots$

\end{frame}

\againframe<3>{myframe}
```

LaTeX Beamer Class: Animations



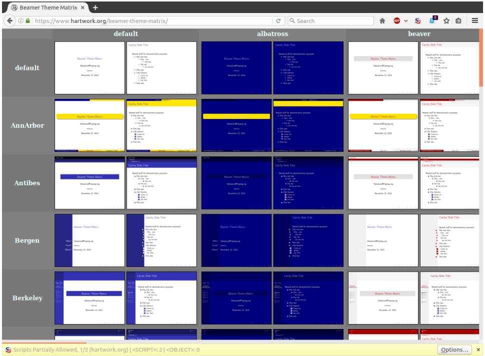
- For an n-page PDF file `foo.pdf`:

  `\animategraphics[autoplay,loop]{fps}{foo}{0}{n-1}`
- A series of `PNG` images can be animated similarly.
- An animated `GIF` needs to be converted into individual `PNG` images:

  `convert -coalesce foo.gif foo.png`
- The use of `animategraphics` requires the `animate` package to be loaded in the preamble.
- Unfortunately, `acroread` and `okular` seem to be the only PDF viewers that can display PDF slides correctly if `animategraphics` was used . . .

LaTeX Beamer Class: Themes

- The visual appearance of slides can be influenced by choosing among multiple pre-defined layouts and coloring schemes.
- A great survey of the basic combinations is provided by the Beamer Theme Matrix, `https://www.hartwork.org/beamer-theme-matrix`.



- Combinations and personal customizations of the pre-defined options allow to create a virtually unlimited variety of layouts.

LaTeX Beamer Class: Themes

- These slides and handouts were generated with the following setting in `wap.tex`:

```
\ifdefined\ishandout
\documentclass[handout,...options...]{beamer}
  \usetheme{default}
  \usecolortheme{dove}
  \usecolortheme[named=BrickRed]{structure}
%  \usepackage{pgfpages}
%  \pgfpagesuselayout{4 on 1}[border shrink=5mm,landscape]
\else \ifdefined\nopause
\documentclass[handout,...options...]{beamer}
  \usetheme{Madrid}
  \usecolortheme{beaver}
  \usecolortheme[named=BrickRed]{structure}
\else
\documentclass[...options...]{beamer}
  \usetheme{Madrid}
  \usecolortheme{beaver}
  \usecolortheme[named=BrickRed]{structure}
\fi\fi
```

- LaTeX command: `pdflatex "\def\ishandout{1} \input{wap}"`.

The Future of pdfLaTeX

- Work on pdfLaTeX has mostly been finished, and future releases should be expected to contain only bug fixes.
- LuaTeX/LuaLaTeX:
  - It has been adopted as the official successor of pdfLaTeX.
  - Based on the Lua scripting engine.
  - It supports multi-directional typesetting.
  - A variety of fonts can be accessed via a library based on FontForge.
  - It has pretty good dynamic memory management.
  - Logos obtained as `\LuaTeX` and `\LuaLaTeX`, as provided by the `metalogo` package.
- XeTeX/XeLaTeX:
  - It is another development that supports Unicode in a native way and that can use any font installed on the system, with no extra configurations needed.
  - Its input files are assumed to be in UTF-8 encoding.
  - It also supports more advanced typographic features than pdfLaTeX.
  - It is included in the TeX Live, MiKTeX, and MacTeX bundles.
  - Logos obtained as `\XeTeX` and `\XeLaTeX`, as provided by the `metalogo` package.

Overleaf

- Overleaf, `www.overleaf.com`, is a cloud-based academic writing environment that supports collaborative work.
- It is based on LaTeX, with LaTeX being run in the background as one enters new text.
- No local LaTeX installation is needed.
- Overleaf also provides a decent Rich Text editor that can be used to enter text in a WYSIWYG environment, even if one knows no or only very little LaTeX. Still, one can switch back to the actual LaTeX code at any time.
- The basic full version is free but it allows only one collaborator per project. Furthermore the free version imposes some limitations on the number of projects and on the storage provided.
- Premium (pay-per-month) plans allow more collaborators per project and come with additional features, such as a synchronization with DropBox or GitHub.

TeXstudio

- TeXstudio is an environment for creating LaTeX documents whose "goal is to make writing LaTeX as easy and comfortable as possible".
- It comes with
  - a LaTeX-aware editor that supports auto-completion and advanced syntax highlighting,
  - an interactive spell checker,
  - an advanced build system that supports Latexmk and several LaTeX systems, like TeX Live and MiKTeX,
  - a PDF viewer that supports a strong synchronization between a LaTeX source document and the PDF output.
- TeXstudio is open source.
- It is available for Linux, Windows and Mac OS X.
- See `www.texstudio.org/`.

Checking Orthography and Grammar

- Since LaTeX contains markup instructions, standard tools for checking spelling and grammar are not readily applicable.
- Some editors (e.g., `emacs`), support spell checking of LaTeX documents.
- Similarly, some LaTeX installations come with tools for spell checking.
- The tools `ispell`, `hunspell` and `aspell` understand LaTeX and will skip LaTeX commands.
- E.g.:

```
aspell --lang=en --mode=tex
       --dont-tex-check-comments check foo.tex
```

ChatGPT

- ChatGPT (`chatgpt.com`) can also be asked to assist with checking grammar and style. It can review a text for grammatical errors, suggest improvements, and offer style recommendations to enhance readability and coherence.
- As of July 2024, it can deal with text written in English. It can provide some basic assistance for other languages, though.
- Schools have varied reactions to the use of ChatGPT or similar tools, reflecting a struggle between recognizing its educational benefits and addressing potential concerns.
- In particular, there are concerns that their advanced features, especially in premium versions, might cross into the realm of academic dishonesty if students use it to do more than just correct errors.
- There also is an issue of accessibility: Not all students may have access to (pay-per-use) premium features, potentially creating inequities in the tools available to students.

Grammarly

- Grammarly, `www.grammarly.com`, is a cloud-based and AI-powered typing assistant designed to enhance writing quality by checking for grammar, punctuation, spelling, and style errors.
- Beyond basic grammar checks, Grammarly provides advanced features such as tone detection, clarity improvements, and vocabulary enhancements. Users can personalize their experience by setting specific writing goals and preferences, enabling tailored feedback.
- It also includes a plagiarism checker, ensuring the originality of content.
- Grammarly's premium version unlocks more advanced features, such as genre-specific writing style checks and additional writing insights.
- Some journals have started to request authors to run a final version of a paper through Grammarly.
- It integrates with various platforms, including web browsers, word processors, and email clients, offering real-time suggestions and corrections.
- Grammarly supports several regional varieties of English, including British and American English, thus allowing users to tailor their writing to specific audiences. As of July 2024, Grammarly does not support text in German or other languages.

LanguageTool

- LanguageTool (`languagetool.org/de`) is an open-source proof-reading tool designed to check grammar, style, punctuation, and spelling in multiple languages.
- It supports over 20 languages, including English and German (and their regional varieties), Spanish, French, Portugues, and many others, making it a versatile option for multilingual users.
- LanguageTool can be used as a web-based application, browser extension, or integrated into various word processors.
- Customizable Rules: Users can create and apply their own grammar rules to suit specific needs.
- Open-Source: Free to use with an active community contributing to its development and improvement.
- The package TeXtidote can remove markup from a LaTeX file and send it to LanguageTool.

Mathematica for Symbolic Computation

Basics of Mathematica

- Mathematica is a software package – “*computer algebra system*” (CAS) – for use in mathematical applications that require symbolic computation.
- It was conceived by Stephen Wolfram, starting in late 1986, and has been developed by Wolfram Research at Champaign (IL, USA).
- It can be used as a scientific calculator, but can also perform operations on functions, manipulate algebraic formulae, and do calculus.
- It also provides graphics capabilities, and can produce two- and three-dimensional graphs.
- It is widely used in science and engineering.
- It provides an interface to LaTeX, and it can output data in a variety of graphics formats such as encapsulated PostScript, GIF, etc.
- It can be interfaced with external programs: it can invoke external programs, and it can be invoked by external programs.
- Mathematica is a commercial product, and it is available for a variety of platforms.
- A license for “WorkAtHome” can be obtained via the web page of PLUS ITServices: See `https://im.sbg.ac.at/display/ITInfo/Software` for more information.

Other Packages for Symbolic Computation

- Maple: By MapleSoft, Waterloo (ON, Canada); `www.maplesoft.com`
- Maxima: Based on MIT's legendary Macsyma; `maxima.sourceforge.net`
- Axiom: By T. Daly, Pittsburgh (PA, USA); `axiom.axiom-developer.org`
- Magma: By Computational Algebra Group, U. Sidney (Australia); `magma.maths.usyd.edu.au/magma`
- MATLAB: Via the MuPAD symbolic engine, by MathWorks, Natick (MA, USA); `www.mathworks.com/products/matlab.html`
- SageMath: By W. Stein, U. Washington (WA, USA); `www.sagemath.org`
- ...

User Interface

- Mathematica can run in an *ASCII terminal mode*, or it can display *notebooks* as an *X11 client*.
- To run Mathematica from an ASCII terminal, enter `math`. To run Mathematica as an X11 client, enter `mathematica`.
- Both display variants are front ends to Mathematica's *kernel*, which takes care of the actual computations.
- Note: One has to press `SHIFT RET` in order to invoke a computation after keying in a command!
- Mathematica can handle very large numbers. If a number is so large that it cannot be displayed on one line, Mathematica places a backslash (\) at the end of the line to show that the number continues on the next line.
- Calculations that take too long can be aborted by typing `ALT ,` (in a notebook environment), or `CTRL c` (in terminal mode).
- Mathematica offers several *palettes* for facilitating the input of characters and symbols.
- Functions can often be entered via *templates*, and `TAB` can be used for moving among *placeholders*.
- Many symbols can also be entered directly. E.g., `ESC p ESC` will generate π.

Basic Math in Mathematica

- A semicolon (`;`) after an expression instructs Mathematica not to print the result.
- Mathematica labels its input and output as `In[`*n*`]` and `Out[`*n*`]`, where *n* is a counter that is stepped for every input/output pair.

```
In[1]  :=  3 + 5
Out[1]  =  8
```

- One can refer to the last output generated as `%`. A string of *k* percent signs refers to the *k*-th previous output, and `%`*n* refers to the output numbered *n*.

In[2] := $\% + 3^2$
Out[2] = 17

In[3] := $\% - 2 * \%1$
Out[3] = 1

Basic Math in Mathematica

- Mathematica distinguishes between two types of values, *exact* and *approximate*.
  - Exact values may either be (a) integers or fractions, in which case Mathematica keeps as many digits as necessary to express the value exactly, or (b) symbolic names for constants such as e, π, $\sqrt{2}$, for which Mathematica knows how to find as many digits as necessary in any computation.
  - Approximate values are most typically numeric expressions containing a decimal point.
- An exact value of x can be converted to an n-digit approximate value by calling the function $\mathrm{N}[x, n]$.

```
In[4] :=  N[% + π, 20]
Out[4] =  4.1415926535897932385
```

Warning

Mathematica will apply inexact computation (and approximate all exact values) even if an expression contains just one approximate value!

Basic Math in Mathematica

- By convention, all names of built-in objects of Mathematica start with upper-case letters. Note that names can never start with a number.
- The *arithmetic operators* of Mathematica have standard calculator form ("$+$", "$-$", "$*$", "$/$", and "^") and have standard mathematical precedence. For instance, multiplication and division are executed before addition and subtraction.
- Mathematica accepts some non-standard input forms for arithmetic. E.g., the multiplication operator $*$ may be omitted so that the multiplication is implied.
- Note, however, that spaces are required if $*$ is omitted: x 2 is different from x2!
- The expression $x = value$ assigns *value* to x.

Warning

Note that this is a *permanent assignment*, and Mathematica will substitute *value* in all subsequent occurrences of x, until or unless explicitly told otherwise.

Basic Math in Mathematica

- Any value assigned to x can be removed via $x = .$ or $\mathrm{Clear}[x]$.

```
In[5]  :=  x = 4
Out[5]  =  4

In[6]  :=  3 * √x
Out[6]  =  6

In[7]  :=  Clear[x]

In[8]  :=  3 * √x
Out[8]  =  3 √x
```

Advice

In order to avoid mistakes, it is advisable to clear assignments as soon as they are no longer needed.

Mathematica as a Scientific Calculator

In[9] := $\sqrt[6]{64}$
Out[9] = 2

In[10] := 123456789123456789 + 987654321987654321
Out[10] = 1111111111111111110

In[11] := 40!
Out[11] = 815915283247897734345611269596115894272000000000

In[12] := π^2 / 6
Out[12] = $\frac{\pi^2}{6}$

In[13] := π^2 / 6.
Out[13] = 1.64493

In[14] := Binomial[5, 2]
Out[14] = 10

In[15] := BaseForm[25, 2]
Out[15] = 11001_2

Mathematica as a Scientific Calculator

```
In[16]  :=  Sin[π]
Out[16]  =  0

In[17]  :=  x = N[π]
Out[17]  =  3.14159

In[18]  :=  Sin[x]
Out[18]  =  1.22465 10^-16

In[19]  :=  x = 1/3 + 1/5
Out[19]  =  8/15

In[20]  :=  (15 x) / 8
Out[20]  =  1

In[21]  :=  Clear[x]
```

Lists as Mathematica Objects

- Many Mathematica objects are based on *lists*. Also, most operations can be applied to whole lists, which get treated as single objects.

```
In[22] :=  x = {2, 3, 4}
Out[22] =  {2,3,4}

In[23] :=  x^2
Out[23] =  {4,9,16}
```

- The commands $\mathrm{Part}[x, i]$ and $x[[i]]$ extract the i-th element of the list x.

```
In[24] :=  x
Out[24] =  {2,3,4}

In[25] :=  x[[2]] = 10
Out[25] =  10
```

Lists as Mathematica Objects

- The commands $\mathrm{Part}[x, i]$ and $x[[i]]$ extract the i-th element of the list x.

```
In[26]  :=  x
Out[26] =   {2, 10, 4}

In[27]  :=  x[[1]] + x[[2]] + x[[3]]
Out[27] =   16

In[28]  :=  Clear[x]
```

- Typical use of delimiters in Mathematica:
  - Parentheses `()` are used for grouping;
  - Brackets `[]` enclose function arguments;
  - Curly braces `{}` delimite lists;
  - Double brackets `[[]]` are used for indexing.

Vectors and Matrices

- Vectors and matrices are lists and lists of lists, respectively.

In[29] := $m[x\_] := \{\{\mathrm{Cos}[x],\ -\mathrm{Sin}[x],\ 0\},\ \{\mathrm{Sin}[x],\ \mathrm{Cos}[x],\ 0\},\ \{0,\ 0,\ 1\}\}$

In[30] := $\mathrm{MatrixForm}[m[x]]$

Out[30] = $\begin{pmatrix} \mathrm{Cos}[x] & -\mathrm{Sin}[x] & 0 \\ \mathrm{Sin}[x] & \mathrm{Cos}[x] & 0 \\ 0 & 0 & 1 \end{pmatrix}$

In[31] := $\mathrm{Simplify}[\mathrm{Det}[m[x]]]$

Out[31] = 1

In[32] := $\mathrm{Transpose}[m[x]]$

Out[32] = $\{\{\mathrm{Cos}[x], \mathrm{Sin}[x], 0\}, \{-\mathrm{Sin}[x], \mathrm{Cos}[x], 0\}, \{0, 0, 1\}\}$

In[33] := $\mathrm{Dimensions}[m[x]]$

Out[33] = $\{3, 3\}$

Vectors and Matrices

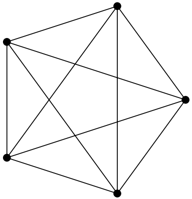
- A dot or the Mathematica function Dot is used for products of Vectors and matrices.

```
In[34]  :=  v = {1, 0, 1}
Out[34]  =  {1,0,1}

In[35]  :=  w = m[π/2] . v
Out[35]  =  {0,1,1}

In[36]  :=  Cross[v, w]
Out[36]  =  {-1,-1,1}

In[37]  :=  v. w
Out[37]  =  1

In[38]  :=  Dot[v, w]
Out[38]  =  1
```

Graphs

- Mathematica can also deal with graphs and similar combinatorical entities.

In[39] := CompleteGraph[5]



Out[39] =

In[40] := MatrixForm[AdjacencyMatrix[%]]

Out[40] =
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Symbolic Computation in Mathematica

- Expand and Factor can be used for transforming algebraic expressions.

```
In[41]  :=  1 + x^2 - 2x
Out[41]  =  1 - 2 x + x^2

In[42]  :=  % * (2 + x)
Out[42]  =  (2 + x) (1 - 2 x + x^2)

In[43]  :=  Expand[%]
Out[43]  =  2 - 3 x + x^3

In[44]  :=  Factor[%]
Out[44]  =  (-1 + x)^2 (2 + x)
```

- In the sample below, the transformation rule $x \text{ -> } 1$ is applied, and "x goes to 1". The replacement operator /. is typed as a pair of characters, with no space in between. Similarly, -> is typed as a pair of characters, with no space in between.

```
In[45]  :=  % /. {x -> 1}
Out[45]  =  0
```

Symbolic Computation in Mathematica (cont'd)

- Getting expressions into a "simple" form sometimes is an art, and may require a bit of experimenting with ExpandAll, Simplify and similar commands.

In[46] := %%

Out[46] = $(-1+x)^2\,(2+x)$

In[47] := % * x/ ((x - 1)^3 * (x + 1))

Out[47] = $\frac{x\,(2+x)}{(-1+x)\,(1+x)}$

In[48] := Expand[%]

Out[48] = $\frac{2\,x}{(-1+x)\,(1+x)} + \frac{x^2}{(-1+x)\,(1+x)}$

In[49] := ExpandAll[%]

Out[49] = $\frac{2\,x}{-1+x^2} + \frac{x^2}{-1+x^2}$

In[50] := Simplify[%]

Out[50] = $\frac{x\,(2+x)}{-1+x^2}$

Differentiation and Integration

- Mathematica can handle differentiation and integration symbolically.

```
In[51]  :=  D[x (1 + x^4), x]
Out[51]  =  1 + 5 x^4

In[52]  :=  D[2x + x^2 + (2 x + x^2) y, y]
Out[52]  =  2 x + x^2

In[53]  :=  Integrate[% / (x + 1), x]
Out[53]  =  x + x^2/2 - log[1 + x]

In[54]  :=  D[%, x]
Out[54]  =  1 + x - 1/(1+x)

In[55]  :=  Factor[%]
Out[55]  =  x (2+x)/(1+x)
```

Differentiation and Integration

- Mathematica can handle differentiation and integration symbolically.

```
In[56]  :=  D[f[x] / x, x]
```
Out[56] = $-\frac{f[x]}{x^2} + \frac{f'[x]}{x}$

```
In[57]  :=  Integrate[%, x]
```
Out[57] = $\frac{f[x]}{x}$

```
In[58]  :=  D[x^y, x]
```
Out[58] = $x^{-1+y}\, y$

```
In[59]  :=  % /. {y -> x}
```
Out[59] = x^x

Differentiation and Integration

- Of course, an integral need not always exist. Still, one may be able to get a numerical approximation of a corresponding definite integral.

```
In[60]  :=  %
Out[60]  =  x^x

In[61]  :=  Integrate[%, x]
Out[61]  =  ∫ x^x dx

In[62]  :=  Integrate[%%, {x, 0, 1}]
Out[62]  =  ∫_0^1 x^x dx

In[63]  :=  N[%]
Out[63]  =  0.783431

In[64]  :=  D[%%%, x]
Out[64]  =  x^x
```

Sums and Products

- Mathematica can also handle sums and products.

```
In[65]  :=  Sum[i, {i, 1, 5}]
Out[65]  =  15

In[66]  :=  Product[i, {i, 1, 5}]
Out[66]  =  120

In[67]  :=  Sum[Product[x + i, {i, 0, j}], {j, 0, 3}]
Out[67]  =  x + x (1 + x) + x (1 + x) (2 + x) + x (1 + x) (2 + x) (3 + x)

In[68]  :=  Expand[%]
Out[68]  =  10 x + 15 x^2 + 7 x^3 + x^4

In[69]  :=  Sum[1 / 2^i, {i, 0, ∞}]
Out[69]  =  2

In[70]  :=  D[x Sum[1 / 2^i, {i, 0, ∞}], x]
Out[70]  =  2
```

Limits

- Mathematica can handle limits.

```
In[71] :=  Sin[x] / x
Out[71] =  Sin[x]/x

In[72] :=  % /. {x -> 0}
Out[72] =  Indeterminate

In[73] :=  Limit[%%, x-> 0]
Out[73] =  1
```

Differential Equations and Higher-Dimensional Calculus

- Mathematica can handle ordinary differential equations.

```
In[74]  :=  DSolve[ { y'[x] == a y[x] + 1, y[0] == 0}, y[x], x]
Out[74] =   {{y[x] → (-1+e^(a x))/a}}
```

$$\text{In[74]} := \text{DSolve}[\{\, y'[x] == a\, y[x] + 1,\ y[0] == 0\},\ y[x],\ x]$$
$$\text{Out[74]} = \{\{y[x] \to \frac{-1+e^{a\,x}}{a}\}\}$$

- Mathematica can also handle multi-dimensional calculus.

$$\text{In[75]} := \text{D}[\{\text{Sin}[\alpha], \text{Cos}[\alpha]\}, \alpha]$$
$$\text{Out[75]} = \{\text{Cos}[\alpha], -\text{Sin}[\alpha]\}$$

$$\text{In[76]} := \text{ArcLength}[\{\text{Sin}[\alpha], \text{Cos}[\alpha]\}, \{\alpha, 0, 2\pi\}]$$
$$\text{Out[76]} = 2\pi$$

$$\text{In[77]} := \text{Grad}[x^2 + y^2 + z^2, \{x, y, z\}]$$
$$\text{Out[77]} = \{2x, 2y, 2z\}$$

Symbolic Solutions for Equations

- Mathematica provides the function `Solve` for computing symbolic solutions for equations.

```
In[78]  :=  Solve[a x^2 + b x + c == 0, x]
```
Out[78] $= \{\{x \to \frac{-b-\sqrt{b^2-4\ a\ c}}{2\ a}\},\{x \to \frac{-b+\sqrt{b^2-4\ a\ c}}{2\ a}\}\}$

- The Mathematica command *expr* /.*rules* applies a list of rules to the expression *expr*. The replacement operator /. is typed as a pair of characters, with no space in between.

```
In[79]  :=  % /. {a -> 2, b -> 3, c -> 1 / 2}
```
Out[79] $= \{\{x \to \frac{1}{4}\ (-3-\sqrt{5})\},\{x \to \frac{1}{4}\ (-3+\sqrt{5})\}\}$

```
In[80]  :=  x /. %
```
Out[80] $= \{\frac{1}{4}\ (-3-\sqrt{5}), \frac{1}{4}\ (-3+\sqrt{5})\}$

```
In[81]  :=  %[[1]] * 4
```
Out[81] $= -3-\sqrt{5}$

Symbolic Solutions for Equations

- Mathematica provides the Solve function for computing symbolic solutions for equations.

```
In[82] :=  Solve[{x - y == 2, x + y == 0}, {x, y}]
Out[82] =  {{x → 1, y → −1}}

In[83] :=  Eliminate[{x - y == 2, x + y == 0}, y]
Out[83] =  x == 1

In[84] :=  Solve[Sin[x]^2 == a, x]
Out[84] =  {{x → −ArcSin[√a]}, {x → ArcSin[√a]}}
```

Symbolic Solutions for Recurrence Relations

- We can use Mathematica to solve recurrence relations.

  In[85] := $\mathrm{RSolve}[\{a[n] == 2a[n-1], a[1] == 1\}, a[n], n]$
  Out[85] = $\{\{a[n] \to 2^{-1+n}\}\}$

  In[86] := $\mathrm{Table}[a[n] /. \mathrm{First}[\%], \{n, 12\}]$
  Out[86] = $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$

- We can also directly tabulate the first few Fibonacci numbers.

  In[87] :=
  $\mathrm{RecurrenceTable}[\{a[n] == a[n-1] + a[n-2], a[1] == 1, a[2] == 1\}, a, \{n, 10\}]$
  Out[87] = $\{1, 1, 2, 3, 5, 8, 13, 21, 34, 55\}$

  In[88] := $\mathrm{Table}[\mathrm{Fibonacci}[n], \{n, 10\}]$
  Out[88] = $\{1, 1, 2, 3, 5, 8, 13, 21, 34, 55\}$

Numerical Mathematics in Mathematica

- Mathematica provides functions for computing numerical approximations of sums, products, and integrals.

```
In[89] := Sum[1 / i^2, {i, 1, ∞}]
Out[89] = π^2/6
```

$$\texttt{Out[89]} = \frac{\pi^2}{6}$$

```
In[90] := N[%]
Out[90] = 1.64493

In[91] := NSum[1 / i^2, {i, 1, ∞}]
Out[91] = 1.64493

In[92] := NIntegrate[ Sin[xy], {x, 0, 1}, {y, 0, x}]
Out[92] = 0.119906
```

Numerical Mathematics in Mathematica

- It can also solve a (system of) polynomial equation(s) numerically, or search for an approximate solution of an arbitrary equation.

$$\texttt{In[93] :=}\quad \mathrm{Solve}\left[\, x\hat{\ }3 \;-\; \sqrt{\pi}x\hat{\ }2 \;==\; 0,\; x\right]$$
$$\texttt{Out[93] =}\quad \{\{x \to 0\}, \{x \to 0\}, \{x \to \sqrt{\pi}\}\}$$

$$\texttt{In[94] :=}\quad \mathrm{NSolve}\left[\, x\hat{\ }3 \;-\; \sqrt{\pi}x\hat{\ }2 \;==\; 0,\; x\right]$$
$$\texttt{Out[94] =}\quad \{\{x \to 0.\}, \{x \to 0.\}, \{x \to 1.77245\}\}$$

$$\texttt{In[95] :=}\quad \mathrm{FindRoot}[\,\mathrm{Sin}[x] \;==\; x, \{x,\; 0.001\}]$$
$$\texttt{Out[95] =}\quad \{x \to 0.\}$$

- Mathematica's function FindRoot has several options. In particular, it can be instructed to use a specific method for searching for a solution besides the (default) Newton's method, e.g., Brent's root bracketing.

Defining Functions in Mathematica

- Mathematica lets one define functions that can then be used similar to built-in functions.

```
In[96]  :=  Expand[ Product[ x + i, {i , 1, 3}]]
Out[96]  =  6 + 11 x + 6 x^2 + x^3

In[97]  :=  exprod[n_] := Expand[ Product[ x + i, {i , 1, n}]]
In[98]  :=  exprod[3]
Out[98]  =  6 + 11 x + 6 x^2 + x^3

In[99]  :=  D[exprod[3], x]
Out[99]  =  11 + 12 x + 3 x^2

In[100]  :=  cex[n_, i_] := ( t = exprod[n]; Coefficient[t, x^i] )
In[101]  :=  cex[3, 2]
Out[101]  =  6
```

Defining Functions in Mathematica

- Mathematica lets one define functions that can then be used similar to built-in functions.

```
In[102]  :=  Clear[cex]
In[103]  :=  t
Out[103]  =  6 + 11 x + 6 x^2 + x^3
```

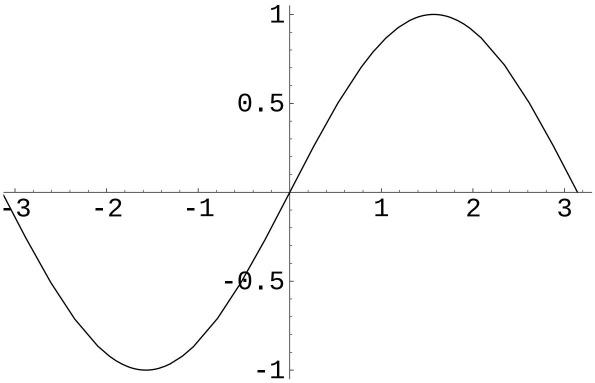
- This is not quite what we expected and want ...
- We resort to a module to encapsulate the local variable.

```
In[104]  :=  Clear[t]
In[105]  :=  cex[n_, i_] := Module[{t}, t = exprod[n]; Coefficient[t, x^i]]
In[106]  :=  cex[3, 2]
Out[106]  =  6

In[107]  :=  t
Out[107]  =  t
```

2D Plots in Mathematica: Plotting Functions

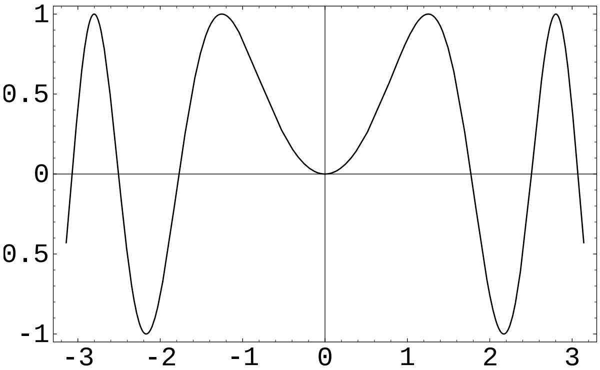
- Mathematica's function Plot offers many features for drawing 2D graphs.

In[108] := Plot[Sin[x], {x, $-\pi$, π}]



2D Plots in Mathematica: Plotting Functions

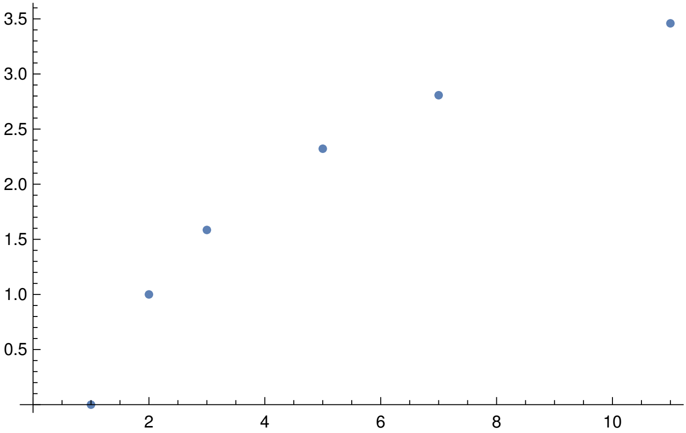
- `In[109] :=` Plot[Sin[x^2], {x, $-\pi$, π}, Frame -> True]



2D Plots in Mathematica: Scatter Plots

- Mathematica can plot pairs of points.

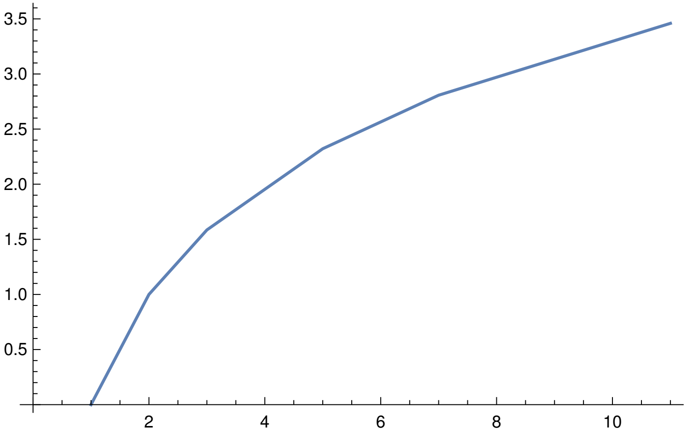
```
In[110] :=
ListPlot[ {{1, 0}, {2, 1}, {3, Log2[3]}, {5, Log2[5]}, {7, Log2[7]}, {11, Log2[11]}}]
```



2D Plots in Mathematica: Line Plots

- Mathematica can plot polygonal curves defined by consecutive pairs of points.

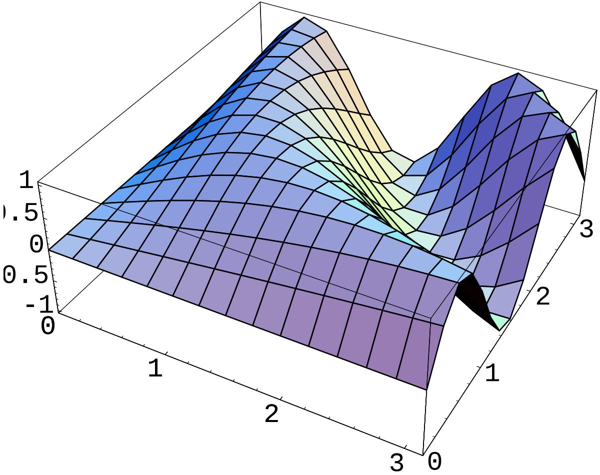
```
In[111] :=
ListLinePlot[ {{1, 0}, {2, 1}, {3, Log2[3]}, {5, Log2[5]}, {7, Log2[7]}, {11, Log2[11]}}]
```



3D Plots in Mathematica

- Mathematica can also handle 3D plots.

`In[112] :=` Plot3D[Sin[x y], {x, 0, π}, {y, 0, π}]

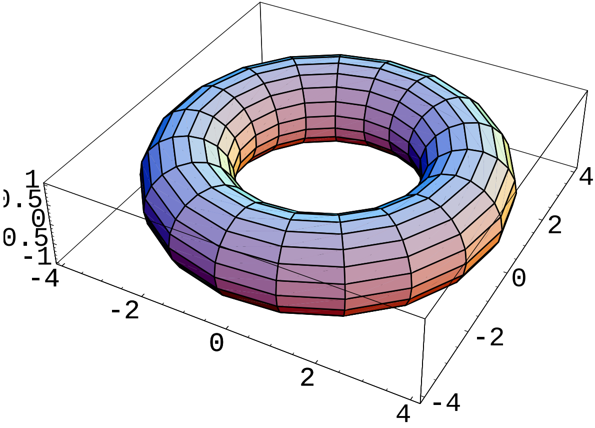


3D Plots in Mathematica

- Mathematica can also handle 3D plots.

```
In[113] :=
```

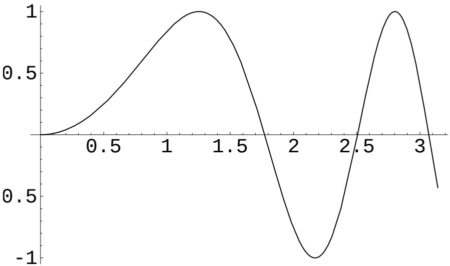
$$\text{torus} = \mathrm{ParametricPlot3D}[\{\mathrm{Cos}[t]\,(3+\mathrm{Cos}[u]),\ \mathrm{Sin}[t]\,(3+\mathrm{Cos}[u]),\ \mathrm{Sin}[u]\},\ \{t, 0, 2\pi\},\ \{u, 0, 2\pi\}]$$



Exporting Mathematica Output

- Mathematica can export a plot as a graphics file. Supported formats include, among others, EPS, PDF, GIF, TIFF, PBM.

`In[114] :=` Plot[Sin [x^2], {x, 0, π}]



`Out[114] =`

`In[115] :=` Export["*foo.eps*", %, "*EPS*"]

- (Portions of) Mathematica notebooks can also be printed as PostScript files. See Mathematica's `print` menu.

Exporting Mathematica Output

- Mathematica can export expressions in C or Fortran format. (C macros are defined in Mathematica's file `mdefs.h`.)

In[116] := $t = (x\hat{}2 - 1)\,/\,\sqrt{x - 1}$

Out[116] = $\frac{-1+x^2}{\sqrt{-1+x}}$

In[117] := CForm[t]

Out[117] = `(-1 + Power(x,2))/Sqrt(-1 + x)`

In[118] := FortranForm[t]

Out[118] = `(-1 + x**2)/Sqrt(-1 + x)`

Exporting Mathematica Output

- Mathematica can export expressions in TeX-format, too.

```
In[119] := t
Out[119] =
```
$$\frac{-1+x^2}{\sqrt{-1+x}}$$

```
In[120] := TeXForm[t]
Out[120] = \frac{-1 + x^2}{\sqrt{-1 + x}}

In[121] := Clear[t]
```

- Mathematica can export a notebook (or portions thereof) as a LaTeX file, too. Such a LaTeX file makes use of macros defined in Mathematica's style file `notebook.sty`. See the `TeXSave` command for details.
- This is the way all the Mathematica expressions of this document were generated. Personal experience tells me that the LaTeX output generated by Mathematica needs a bit of manual polishing in order for LaTeX to digest it, and to format it neatly.

Interfacing Mathematica with Other Programs

- Bi-directional communication between Mathematica and an application program is supported by the *MathLink* standard. See the manual for details.
- Mathematica can be instructed to generate output suitable for display by Geomview. (The file `OOGL.m` is provided by Geomview.)

```
In[122] := << OOGL.m
In[123] := WriteOOGL["m_torus.off", torus]
```

- One can also use Geomview directly for displaying Mathematica graphics. The `Geomview` command invokes Geomview and sends the graphics to Geomview as an OOGL object. (For some reason, this does not work in our environment!)

Sample Use of Mathematica: Image Processing

- Mathematica supports image processing.

In[124] := $img = \text{ColorConvert}[\,\text{Import}[''\textit{kitten.jpg}''\,],''\textit{RGB}''\,]$

Out[124] = 

In[125] := $\{w, h\} = \text{ImageDimensions}[img]$
Out[125] = $\{250, 250\}$

In[126] := $data = \text{ImageDate}[img]$

In[127] := $data[[125, 125]]$
Out[127] = $\{0.427451, 0.392157, 0.356863\}$

Sample Use of Mathematica: Image Processing

- Mathematica supports image processing.

In[128] := *imgGray* = ColorConvert[*img*, "*Grayscale*"]

Out[128] = 

In[129] := Export["*kitten_gray.png*", *imgGray*]
Out[129] = kitten_gray.png

In[130] := Clear["Global`*"]

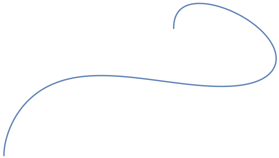
Sample Use of Mathematica: Bézier Curve

- We use Mathematica for plotting a Bézier curve.

In[131] := $pnts =$
$\{\{0,0\},\{0,1\},\{1,3\},\{3,3\},\{5,0\},\{7,1\},\{8,2\},\{7,3\},\{5,4\},\{4,4\},\{4,3\}\}$
Out[131] =
$\{\{0,0\},\{0,1\},\{1,3\},\{3,3\},\{5,0\},\{7,1\},\{8,2\},\{7,3\},\{5,4\},\{4,4\},\{4,3\}\}$

In[132] :=
$\mathrm{Bezier}[n\_, pnts\_, x\_] := \mathrm{Sum}[pnts[[k+1]] * \mathrm{BernsteinBasis}[n,k,x], \{k,0,n\}]$

In[133] := $\mathrm{ParametricPlot}[\mathrm{Bezier}[10, pnts, x], \{x, 0, 1\}, \mathrm{Axes} \to \mathrm{False}]$

Out[133] = 

In[134] := $\mathrm{Export}[\text{"}\mathit{sample\_bezier.pdf}\text{"}, \%, \text{"}\mathit{PDF}\text{"}]$

Symbolic Computation – Caveats

- Consider the class of terms generated from one variable *x*, constants for the rationals, π, and the function symbols $+, *, \mathtt{sin}$, abs. Caviness (1967) proved that the simplification problem with respect to functional equivalence is undecidable for this class of terms.
- Similarly, Risch proved that the problem of integration in finite terms is undecidable for transcendental functions. In 1968–1969, Risch also described the first complete integration algorithm for algebraic and elementary transcendental functions.
- Several important algorithms of computer algebra have an exponential complexity, e.g., Collins' cylindrical algebraic decomposition for quantifier elimination.
- The bit complexity may grow substantially during a computation, thus potentially requiring a large main memory. In particular, the bit complexity of intermediate results may be significantly larger than the complexity of the input and the output. Some algorithms are well-known to be memory hoggers — consult textbooks prior to waiting for hours/days just in order to see the system crash due to lack of memory.

Graphics and Visualization

Basics of Geomview

- Geomview is an interactive program for *viewing and manipulating 3D geometric objects*.
- Geomview was written by staff members of the Geometry Center (UMN).
- Unfortunately, in an attempt to save money, the US administration scrapped the Geometry Center in 1998, and development efforts for Geomview have been hampered.
- Geomview is in a mature and stable state, though! And it is still used widely and continues to evolve, see www.geomview.org
- Geomview is free software, released under a GNU license.
- It runs on a variety of systems using generic OpenGL or X11 graphics and a Motif interface.
- It builds on Mac OS X, and it can also run under MS Windows using Cygwin.

Basics of Geomview

- The simplest way to use Geomview is as a *standalone viewer* to see and manipulate objects. It can display objects described in a variety of file formats.
- Geomview supports the following simple data types: polyhedra with shared vertices, quadrilaterals, rectangular meshes, vectors, and Bézier surface patches of arbitrary degree including rational patches.
- Object hierarchies can be constructed with lists of objects and instances of object(s) transformed by one or many 4×4 matrices.
- Objects can be manipulated through direct mouse manipulation, control panels, and keyboard shortcuts.
- Geomview allows multiple independently controllable objects and cameras. It provides interactive control for motion, appearances (including lighting, shading, and materials), picking on an object, edge or vertex level, snapshots, and adding or deleting objects.
- One can also use Geomview to handle the display of data coming from another program (*external module*) that is running simultaneously. As the other program changes the data, the Geomview image reflects the changes.
- Geomview can also display 3D graphics output from Mathematica and Maple.

Object Manipulations

- Objects can be selected by clicking at the name of the object in the `Targets` browser of the `Main` panel. If `world` is selected, then any motion/transformation is applied to all objects currently drawn.
- The object selected is called the *target object*.
- Geomview lets you manipulate objects with the mouse. There are six different mouse motion modes: Rotate, Translate, Cam Fly, Cam Zoom, Geom Scale, and Cam Orbit. The `Tools` panel has a button for each of these modes; to switch modes, click on the corresponding button.
- Most of the motion modes have *inertia*, which means that if one lets go of the button while moving the mouse, the motion will continue.
- Pressing the shift key while dragging with left or middle mouse buttons in most motion modes gives slow-speed motions, useful for fine control of object placements.
- Generally, the left mouse button controls motion in the screen plane, while the middle mouse controls motion along or around the forward direction.
- The `Center` button undoes the target object's transformation, moving it back to its home position, which is where it was when it was originally loaded into Geomview.

Object Manipulations

- Geomview uses the *glass sphere model* for mouse-based rotations. Think of the object as being inside an invisible sphere, and regard the mouse cursor as a gripper outside the sphere. When one presses the left mouse button, the gripper grabs the sphere; when one releases the left mouse button, the gripper releases the sphere.
- Moving the mouse while holding the button down causes the sphere (and hence the object) to move in the same direction as the mouse.
- Specifically, in `Rotate` mode the axis of rotation passes through the origin of the center object, is parallel to the camera view plane, and is perpendicular to the direction of motion of the mouse. When the center is "target", this means that the target object rotates about its own origin.
- Press the middle mouse button in order to rotate the target object about an axis perpendicular to the view plane.
- One can pick any point on an object (not just its origin) as the center of motion by holding down the shift key while clicking the right mouse button; this chooses a point of interest.
- In order to translate the target object, hold the left mouse button down (after selecting the `Translate` mode). The middle mouse button translates the target along an axis perpendicular to the view plane.
- `Cam Fly` is a crude flight simulator that lets one fly around the scene. It works by moving the camera.

Object Manipulations

- `Cam Orbit` mode lets one rotate the current camera around the current center.
- `Cam Zoom` lets one change the current camera's field of view with the mouse.
- `Geom Scale` mode lets one enlarge or shrink an object.
- The `Stop` button causes all motion to stop.
- The `Look At` button causes the current camera to be moved to a position such that it is looking at the target object, and such that the target object more or less fills the window.
- The `Reset` button stops all motion and causes all objects to move back to their home positions.

Modifying the Appearance of Objects

- Geomview uses a hierarchy of appearances to control the way things look. An *appearance* is a specification of information about how something should be drawn.
- There is an appearance associated with "World", which serves as the parent of each individual object's appearance. Also, there is a global "base" appearance, which is the parent of the World appearance.
- Appearances work in a hierarchical manner: if a certain appearance property, for example a face color, is not specified in a particular object's appearance, that object is drawn using that property from the parent appearance. If both the parent and the child appearance specify a property, the child's setting takes precedence unless the parent appearance is set to override.

Modifying the Appearance of Objects

- The `Appearance` panel controls various things about the way Geomview draws objects. For instance, the `[ae] Edges` button allows to toggle between having the object displayed with or without edges.
- The `Appearance` panel also lets one select colors (in RGB or HSV) and shading information (constant, flat, smooth).
- The `Materials` panel controls material properties such as the degree of opacity, diffuse and specular reflection, and ambient light.
- The `Lighting` panel controls the number, position, and color of the light sources used in shading.
- The `Cameras` panel controls certain aspects of the target camera (such as its field of view). The use of multiple cameras is supported.
- The *Geomview command language* (gcl) provides complete control of all appearance data, including data that cannot be changed via the panels.

I/O Control

- The `Save` panel offers several possibilities for storing Geomview objects and other information in files.
- One can store gcl commands, geometric data, input data for RenderMan, PPM (software) snapshots, snapshots in PostScript format, and data for restoring all windows and panels in a subsequent session of Geomview.
- Commands in gcl format, which uses the syntax of lisp, can be entered via the `Commands` panel.
- Most panel interaction can be replaced by commands that have keyboard shortcuts. For instance, the keyboard shortcut for switching to `Rotate` mode is `r`.
- Some keyboard shortcuts consist of more than one key. In these cases one types the keys one after the other, with no `RET` afterwards. For instance, `g1ae` toggles the edge drawing for object (“geom”) `g1`.

OOGL Files

- Geomview reads objects in the format of the *Object Oriented Graphics Library* (OOGL).
- Examples for many OOGL objects can be found in Geomview's `data/geom` directory.
- Most OOGL files are are free-format ASCII. (Binary formats are also defined for several objects types.)
- Typical OOGL objects begin with a key word designating the object type, possibly with modifiers indicating the presence of additional data (such as color).
- Most key words are case sensitive.
- When OOGL objects are input, the OOGL library uses the file suffix to guess at the file type.

OOGL Files

- Geomview supports inhomogeneous and homogeneous coordinates.
- Transformation matrices are given in a 4×4 row-vector representation, for multiplication on the right of vectors. That is, a row vector p (of a point in homogeneous coordinates) is transformed by a matrix $\mathbf{M}$ to a point p' as follows: $p' = p\,\mathbf{M}$.
- Appearances and texture maps can be specified; see the manual for details.

OOGL Objects

QUAD: a collection of quadrilaterals. The default file suffix is `.quad`.

MESH: a rectangularly-connected mesh of dimension $n \times m$. The default file suffix is `.mesh`.

Bézier: a Bézier surface. The default file suffix is `.bez`.

OFF: an object in *object file format*. It is used for representing collections of planar polygons, possibly with shared vertices. This is a convenient way to describe polyhedra. The polygons may be concave but polygons with holes are not supported. The default file suffix is `.off`.

VECT: strings of connected line segments, possibly closed. The default file suffix is `.vect`.

SKEL: collections of points and polylines, possibly with shared vertices. The default file suffix is `.skel`.

SPHERE: a sphere, drawn as a collection of rational Bézier patches. The default file suffix is `.sph`.

INST: a 4×4 transformation, to be applied to another OOGL object. The default file suffix is `.inst`.

LIST: a list of OOGL objects. The default file suffix is `.list`.

TLIST: a list of 4×4 transformations. The default file suffix is `.grp`.

External Modules

- An *external module* is a program that interacts with Geomview. It communicates with Geomview through gcl commands and can control any aspect of Geomview that one can control through Geomview's user interface. Typically, Geomview acts as a *display engine* for the external module.
- External modules known to Geomview are listed in the `Modules` browser in Geomview's `Main` panel. An external module can be invoked by clicking on its entry in the browser.
- In order to make an external module `foo` known to Geomview, include the following line in your local initialization file, `.geomview`:

  ```
  (emodule-define "Foo" "./foo")
  ```

  Here, `Foo` is the name of the external module that will appear in the `Modules` browser of Geomview. One can also execute this command on-line from the `Commands` panel.

External Modules

- When starting an external module, Geomview creates *pipes* connected to the module's standard input and output.
- Geomview interprets everything that the module writes to its standard output as a gcl command. Likewise, if the exernal module requests any data from Geomview, Geomview writes that data to the module's standard input.
- Note that this implies that the module cannot use standard I/O for communicating with the user!
- Sample external modules (`example*.c`) are available in Geomview's subdirectory `/geomview/doc`.

Using Geomview as an External Display Engine

- It is also possible to invoke Geomview from an application program, and to direct graphics output produced by the application to Geomview, thus using Geomview as an external display engine.
- Communication between the application program and Geomview is again carried out via *pipes*.
- In a C environment, an input file `geomview_in` for Geomview is opened by the application program as follows:

  ```
  geomview_in = popen("togeomview", "w");
  ```
- Then, the application writes any input for Geomview to `geomview_in`.
- Since pipes may be buffered, it is advisable to flush the pipe after data has been written to the pipe: `fflush(geomview_in)`.

Using Geomview as an External Display Engine

- Typically, we will want Geomview to create objects that can later on be modified (e.g., translated or rotated). The gcl command

  ```
  (geometry Foo { : foo })
  ```

  instructs Geomview to create an object ("geom") named `Foo` as an instance of the *handle* `foo`, where `Foo` is the name of the object as it will appear in the object browser of Geomview, and `foo` is the internal reference for Geomview.
- Handles allow one to name a piece of geometry whose value can be specified elsewhere, and which can be updated repeatedly. See the manual for details.
- If multiple objects are to be passed to Geomview, it is a good idea to turn off any scaling of the individual objects:

  ```
  (normalization Foo none)
  ```

The End!

I hope that you enjoyed this course, and I wish you all the best for your future studies.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab