
# Boltzman Machine

## Review and Prospects

Hooman Zolfaghari
Ramtin Moslem
Borna (lastname)
Houman (lastname)

# Energy Based Models

# Botzman Machine

# Research Prospect

# Template Things

# Single Column

For 4-by-3 aspect ratio slides, specify `standard` as an option to the document class. Write your presentation like a normal LaTeX file with a `\maketitle` command and `\chapter` and `\section` headings. The `\maketitle` contents are defined by the following macros:

```
\pretitle    \author
\title       \subtitle
```

The `\chapter` heading creates a slide with just the chapter name, and the `\section` heading sets the title of a new slide. However, if no text follows the section, no slide will be created. Text which does not fit on one slide will flow onto the next slide automatically.

# Double Column

Use the `\twocolumn` and `\onecolumn` commands right after the section heading to control the number of columns. Text will flow from the left column to the right.

- Point one
- Point two
- Point three
- Point four
- Point five
- Point six
- Point seven
- Point eight
- Point nine
- Point ten
- Point eleven
- Point twelve

You can use `\pagebreak` to force text onto the next column.

# Table of Stuff

You can create any variety of subdivisions on your slide by using the `tabular` environment.

| Primary | Secondary | Tertiary |
|---|---|---|
| First | Second | Third |
| One | Two | Three |

| Alpha | Beta | Gamma |
|---|---|---|
| Green | Blue | Red |
| Cyan | Yellow | Magenta |

The `\cellcolor` command sets the background color of a table cell.

# Centering

Use the `Center` environment
to center horizontally *and* vertically.

# Explicit Code

# Python

Use the `python` environment for Python code.

```
def write_list(fid, x, level):
    ind = '  '*level
    xs = '0' if abs(x[0]) < 1e-3 else "%.3f"
    txt = '\n%svalues=\"%s' % (ind, xs)
    for n in range(1, len(x)):
        xs = '0' if abs(x[n]) < 1e-3 else "%.3f"
        if len(txt) + 3 + len(xs) >= 80:
            fid.write(txt + ';\n')
            txt = ind + '  ' + xs
        else:
            txt += '; ' + xs
    fid.write(txt + '\"')
```

# Python

You can use the `\HL` command to highlight a line of code.

```
1  def write_list(fid, x, level):
2      ind = '  '*level
3      xs = '0' if abs(x[0]) < 1e-3 else "%.3f"
4      txt = '\n%svalues=\"%s' % (ind, xs)
5      for n in range(1, len(x)):
6          xs = '0' if abs(x[n]) < 1e-3 else "%.3f"
7          if len(txt) + 3 + len(xs) >= 80:
8              fid.write(txt + ';\n')
9              txt = ind + '  ' + xs
10         else:
11             txt += '; ' + xs
12     fid.write(txt + '\"')
```

# MATLAB

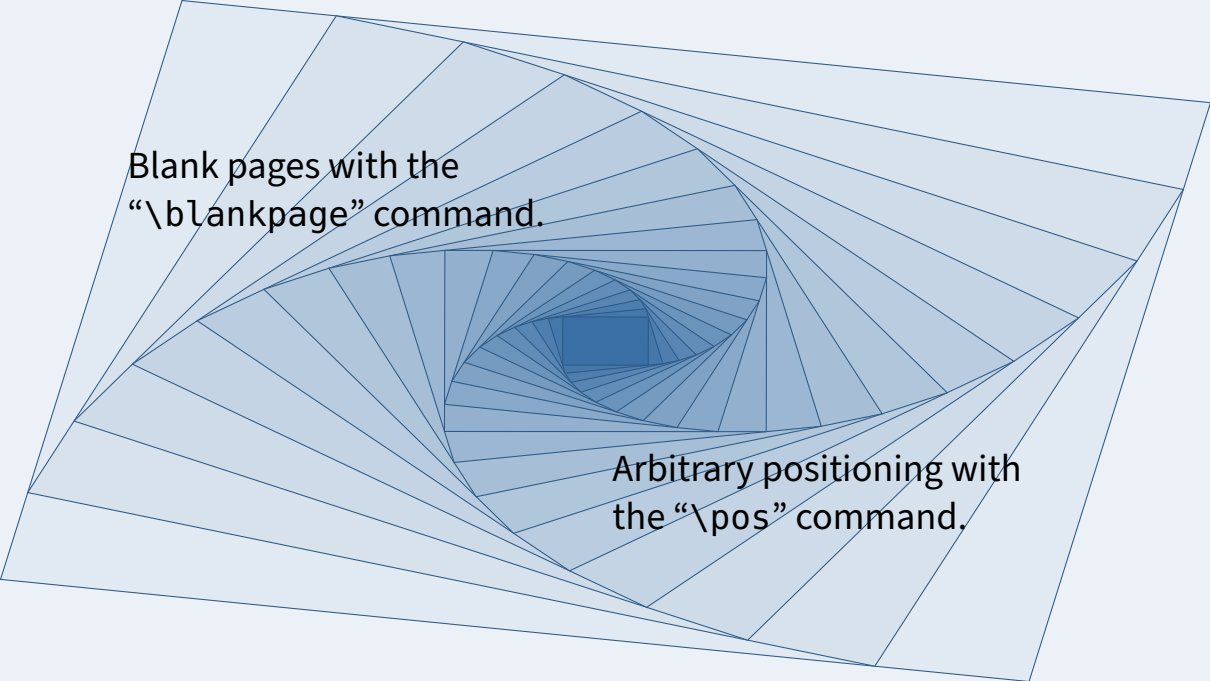
Use the `matlab` environment for MATLAB code.

```matlab
function savepdf(name, width, height)
    % name is the file name including ".pdf".
    % Both width and height are in (cm).
    set(gcf, 'units', 'centimeters', ...
        'position', [0, 0, width, height])
    set(gca, 'FontSize', 9);
    set(gca, 'FontName', 'Times New Roman');
    exportgraphics(gcf, name, ...
        'ContentType', 'vector');
end
```

# R Language

Use the `rlang` environment for R code.

```r
factorial <- function(n) {
    if (n == 0 || n == 1) {
        return(1)
    } else {
        return(n * factorial(n - 1))
    }
}
```


Blank pages with the
“\blankpage” command.
Arbitrary positioning with
the “\pos” command.