(Click on the tabs above for more information on each topic. Some tabs also have tabbed subtopics.)

Welcome to the PNNL-Slides Package!

Welcome to the use of the new PNNL-Slides.cls file. We've made the process of preparing your slides as simple and handsome as possible, and hope you'll enjoy the process. For those of you who are familiar with LaTeX you'll see that we've made very few changes from standard LaTeX commands. The few new commands are shown in this documentation.

If you need help after you read this documentation, you may send email to Colleen Winters at colleen.winters@pnnl.gov or Mike Parker at mike.parker@pnnl.gov.

If possible please send a small file demonstrating the problem.

Files in this package, and what they do

pnnl-slides.cls Document class file

SlideTemplate.tex A file containing all the commands that are unique to this style, with explanations of how to use them

SlideSample.tex/.pdf Compare SlidesSample.tex with SlidesSample.pdf to see how to enter commands correctly.

SlideFigs.zip All the graphics files you may need. Please make a directory up one level from the directory in which you are working, called SlideFigs and drop the contents of SlideFigs.zip into that directory.

SlideFonts.zip Repository for fonts: You may not need these font files, since we are only using Arial for the body fonts, and Arial is commonly found on computers already; but if you do need them, put the files found in fonts.zip in the same directory as your .tex file.

SlideDocs.pdf Documentation (this document)

readme.txt List of files and their uses

Tips as you get started

You'll find many examples of commands in use in sampleslides.tex with the resulting document sampleslides.pdf. Comparing the .tex file with the resulting .pdf file is an excellent introduction to this style.

You'll also find copying the SlidesTemplate.tex and giving it your name and working in the new file will make it easier to find information about commands that are unique to this style.

Make a SlideFigs directory

To get started, make a directory or folder one level up from the directory or folder where your .tex file is found. Name this directory/folder SlideFigs, and drop the illustration files found in SlideFigs.zip into the new directory. Some commands found in the PNNL-slides.cls file will look for the ./SlideFigs directory, so we want to have it and the files contained ready to be accessed.

xelatex and lualatex

We are using the fontspec package for fonts, which is very convenient for using .ttf or .otf fonts. However, fontspec expects the user to format the .tex file using xelatex or lualatex instead of pdflatex or other varieties of LaTeX engines. If you don't use xelatex or lualatex you will get an error message, and your document will not compile, so you will be instantly reminded of this fact.

You'll find using xelatex or lualatex is no more difficult to use than pdflatex so this should not be a problem.

Packages Included

Tikz: **TikZ** with the library **calc**,
Font management: **setspace**, and **hyphenat**, **fontspec**
AMS packages: **amsmath**, **amssymb**, **amsfonts** and **bm** for AMS-LaTeX math typesetting and good-looking bold symbols
Graphics packages: **graphicx** for including figures, **subcaption** for subfigures and subcaptions, **tcolorbox** for fancy boxes

Using the SlidesTemplate file

The easiest way to start your article is to copy and rename the template file, **SlidesTemplate.tex**, and use it to start producing your slides. You'll find some examples of the commands you can enter to make the titlepage, and reminders and examples about the other commands you might use.

Using the SlideSamples files

As well as the template file, the sample file for making slides, SlideSamples.tex/.pdf will be helpful, since you can compare the code with the resulting .pdf, giving yourself guidance when making your own slides.

Document Set Up: Documentclass Options

You have these documentclass options.

[OUO] = Official Use Only, used `\documentclass[OUO]{pnnl-slides}`

[FOUO] = For Official Use Only, used `\documentclass[FOUO]{pnnl-slides}`

[SSI] = Sensitive Security Information, used `\documentclass[SSI]{pnnl-slides}`

[BusinessSensitive] = used `\documentclass[BusinessSensitive]{pnnl-slides}`

(no option) = No classification level listed. used `\documentclass{pnnl-slides}`

Choose one of the following:

```
\documentclass[OUO]{pnnl-slides}
\documentclass[FOUO]{pnnl-slides}
\documentclass[SSI]{pnnl-slides}
\documentclass[BusinessSensitive]{pnnl-slides}
\documentclass{pnnl-slides}
```

Options 1, 2, 3, and 4 will add the security level to the bottom of each page.

OUO, FOUO, and SSI will require you to fill in information for a classification box that will appear at the top of the title slide.

Click tab above for more information on classification boxes.

For title page classification box

When using the OUO, FOUO, or SSI options with the documentclass, you will need to enter the following information, after `\documentclass{pnnl-slides}` and before `\begin{document}`. This information will be used in the classification box on the title page slide.

Enter your own text between the angle brackets:

```
\ClassificationNameOrg{<Name/Organization>}
\ClassificationDate{<Classification Date>}
\ClassificationGuidanceUsed{<Guidance Used>}
```

Once these terms are entered, you'll find that the classification box found on the title page will have the correct information.

For instance, if the documentclass option was set to OUO, (`\documentclass[OUO]{pnnl-slides}`) and you entered

```
\ClassificationNameOrg{Name/Organization}
\ClassificationDate{Classification Date}
\ClassificationGuidanceUsed{Guidance Used}
```

This classification box would appear on the title page:

OFFICIAL USE ONLY

May be exempt from public release under the Freedom of Information Act (5 U.S.C. 552) exemption number(s) and category: (i.e., Exemption 7 Law Enforcement)

Department of Energy review required before public release.

Name/Organization
Name/Org:

Classification Date
Date:

Guidance Used
Guidance (if applicable):

Setting Background Image

Background image, choose one!

(you'll find this list in slides-template.tex)

Uncomment the name of the image you'd like to use:

```
% \BackgroundImage{ancient-glass-micro}
% \BackgroundImage{blue-glass-globes}
% \BackgroundImage{CACO3-spines}
% \BackgroundImage{cloud-formations}
% \BackgroundImage{copper-data-grid}
% \BackgroundImage{data-mapping}
% \BackgroundImage{data-visualization}
% \BackgroundImage{military-with-flag}
% \BackgroundImage{PNNL-arial-photo}
% \BackgroundImage{PNNL-black}
% \BackgroundImage{PNNL-white}
% \BackgroundImage{powerline-sunset}
% \BackgroundImage{pumpkin-slice}
% \BackgroundImage{Pu-oxide-particles}
% \BackgroundImage{root-and-microbes}
% \BackgroundImage{Sequim-Bay}
% \BackgroundImage{virus-biosample}
% \BackgroundImage{weathered-UO2}
```

Experiment until you find a design that you like and that reflects your slide set content.

Slide Title Page Commands

Before `\begin{document}` there are some commands that you need to use to make `\titlepage` work correctly.

Start with `\title{<Title for your slide set>}`.

Then use either (1) Author/Institute, or (2) Author/Department combination.
Finally, supply `\date{}` if desired. Default will be the current date.

```
%% (1) Author/Institute:
\author{<First author>\inst{1},
        <Second author>\inst{2}
}

\institute{\inst{1} <name>\\
\inst{2} <name of second institute>%
 }

%% (2) Author/Department:
\author{<Author Name>}
\department{<Department>}

%% (3) Set date if desired, current date is default.
% \date{January 3, 2020} % date of your choice
% \date{} % Turn off automatic call to current date.
```

Now the `\titlepage` command will produce the title slide with the background image. When using the OUO, FOUO, or SSI options with the documentclass, the classification box will appear on the title slide, as well as the classification term at the bottom of all slides.

The `[BusinessSensitive]` option will not use a classification box, but will cause 'Business Sensitive' to appear at the bottom of all slides.

```
\begin{frame}
\titlepage
\end{frame}
```

Frames

Every slide is produced with `\begin{frame}...\end{frame}`

Options

`\begin{frame}[noframenumbering]` – May be used for title slide to prevent a page number for that slide.
`\begin{frame}[fragile]` – This option is necessary if you want to use verbatim in your slide.

More Beamer commands

There are many other commands available in Beamer, the package that formats these slides. An example is overlay, allowing you to expose only part of the slide at one time, using the `\pause` command. See http://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf for more commands (p. 80 for overlay commands).

Frame Titles

`\frametitle{}` and `\framesubtitle{}` are positioned at the top of the slide.

```
\begin{frame}[fragile]
\frametitle{Here is a frame title}
\framesubtitle{Here is a frame subtitle}
 Text...
\end{frame}
```

Mid Frame Title:

The command `\midframetitle{<distance below top of column>}{<slide title}`

```
\midframetitle{1in}{This title is positioned 1 inch down from top of column}
Here is more text.
```

It might be visually confusing, but you can even use both kinds of title on one slide if desired.

Two columns, organizing your slide

Since slides are in landscape mode, you may want to use two columns for many of your slides. We have two methods available.

Either method is entered between `\begin{frame}...\end{frame}`.

(1) `\dopage\lside ...\rside ...\endpage`, or

(2)
```
\begin{columns}
    \begin{column}{<column width>...\end{column}
    \begin{column}{<column width>...\end{column}
\end{columns}
```

Either method allows `\midframetitle{<distance down from top of column>}{<title>}` in the left column.

Using dopage...endpage

This method is easy to use but doesn't allow changing the width of the two columns.

```
\dopage
\lside
  <Left side column text>

\rside
  <Right side column text>
\endpage
```

Using the Columns commands

You can choose the width of each column. The `[T]` option makes the top of the columns align.

```
\begin{columns}[T]
   \begin{column}{0.6\textwidth}
     <Left side column text>
   \end{column}

   \begin{column}{0.25\textwidth}
     <Right side column text>
   \end{column}
\end{columns}
```

Font sizes

- Default font for slides is Arial 28pt.
- We can size fonts up or down, depending on how much information we need to put on a slide.
- In case you want to fit more on a page, you can choose a smaller font; or if you want to emphasize a phrase, you can choose a larger font:
- In general, we don't want authors to make the fonts **too** small, but it is always a matter of opinion and requires flexibility.

List of Font Sizes

This is only an approximation of the actual size of the fonts, but it gives you the general idea of the fonts that are available.

Example taken from SlideSamples.pdf:

This is the default font size, `\normalsize`, 28 point.

This is `\fiftypt`

This is `\thirtysixpt`

This is `\thirtypt`

This is `\twentyeightpt`

This is `\twentysixpt`

This is `\twentyfourpt`

This is `\twentytwopt`

This is `\twentypt`

This is `\eighteenpt`

This is `\sixteenpt`

To contain font change, precede it with `{` and follow changed text with `}`.

This code:

```
Here is {\sixteenpt sixteenpt
text} and here is {\thirtysixpt
thirtysixpt text}.
```

Produces this result:

Here is sixteenpt text and here is thirtysixpt text.

Blocks

- Block takes an argument: `\begin{block}...\end{block}{<title>}`
- You can use the argument to make : `theorem`, `example`, `definition` or any other block title.

```
\begin{block}{An Equation}
  Here's a block containing an equation:
  \begin{equation}
    \partial_t p(X, t) + \nabla \cdot \mathbf{v}(X, t) p(X, t) =
      \nabla \cdot  \left [ \mathbf{D}(X, t) \nabla p(X, t) \right ]
  \end{equation}
\end{block}
```

An Equation

Here's a block containing an equation:

$$\partial_t p(X,t) + \nabla \cdot \mathbf{v}(X,t) p(X,t) = \nabla \cdot [\mathbf{D}(X,t) \nabla p(X,t)] \quad (1)$$

tcolorbox

Easy too, but notice the syntax of the title: `\begin{tcolorbox}[title={<Title>}]`

```
\begin{tcolorbox}[title={Using tcolorbox}]
  This one is a \verb+tcolorbox+ box, with an unnumbered equation:
  \begin{equation*}
    \partial_t p(X, t) + \nabla \cdot \mathbf{v}(X, t) p(X, t) =
        \nabla \cdot \left [ \mathbf{D}(X, t) \nabla p(X, t) \right ]
  \end{equation*}
\end{tcolorbox}
```

Using tcolorbox

This one is a `tcolorbox` box, with an unnumbered equation:

$$\partial_t p(X,t) + \nabla \cdot \mathbf{v}(X,t) p(X,t) = \nabla \cdot [\mathbf{D}(X,t) \nabla p(X,t)]$$

Listing

Our itemize and enumerate environments are the same as used in LaTeX generally:

```
\begin{itemize}
\item text...
\end{itemize}

\begin{enumerate}
\item text...
\end{enumerate}
```

The size of the font automatically gets smaller with each new level of either itemize or enumerate.

One anamoly: for the itemize fourth level in, we need to use the command `\tinyitem{text...}`, instead of `\item`.

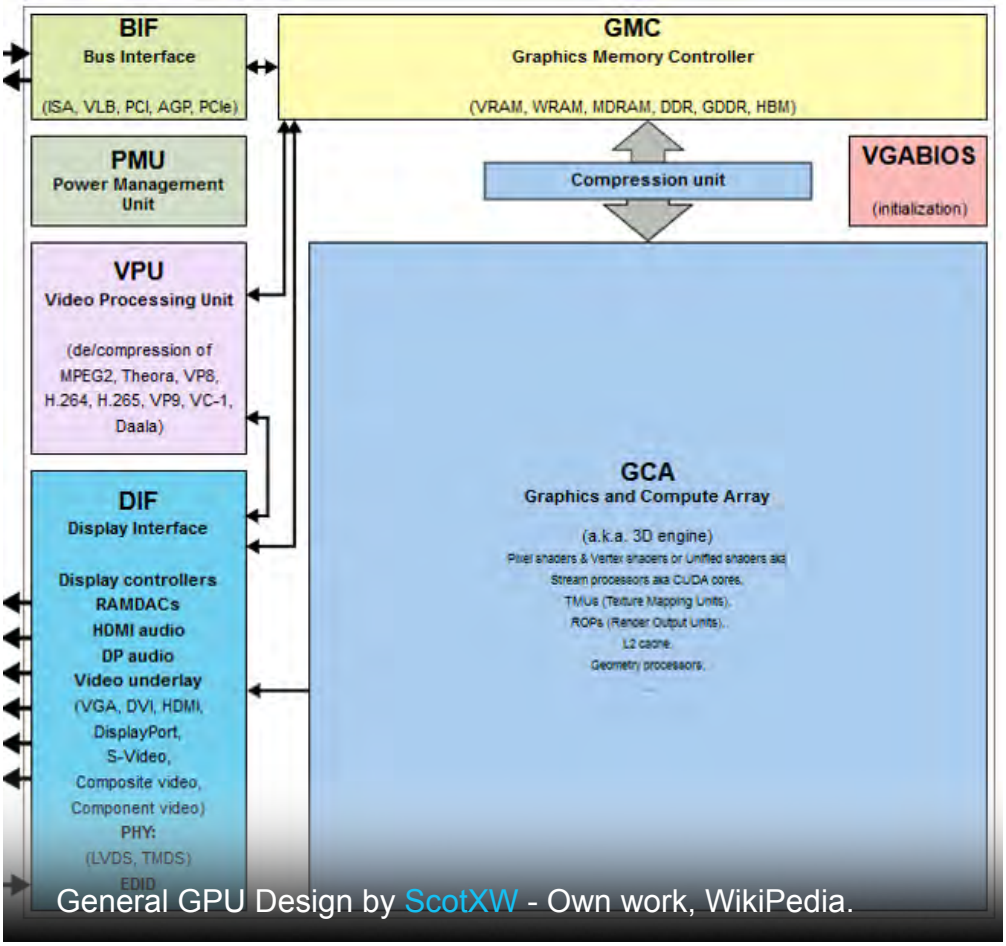
```
\begin{itemize}
\item Font sizes reduce as we create the levels in a list, starting with Arial 28

\begin{itemize}
\item Then Arial 24

\begin{itemize}
\item And then Arial 20

\tinyitem{And Arial 18 -- I think that's pretty small for a slide but
   someone may need it. }

\tinyitem{Turing architecture discussion from Wikipedia}

\end{itemize}   \end{itemize}   \end{itemize}
```

- Font sizes reduce as we create the levels in a list, starting with Arial 28
  - Then Arial 24
    - ✓ And then Arial 20
      - And Arial 18 – I think that's pretty small for a slide but someone may need it.
      - Turing architecture discussion from Wikipedia

Making Figures and Captions

Very easy! Two commands: `\fig{<width>}{<file name for graphics file>}` and `\figcaption{<Caption>}`. (Caption may have more than one line.) An example, taken from SlideSamples.tex/.pdf. It is highly recommended that you compare the SlideSamples.tex file with the .pdf results, particularly for figures.

```
\fig{\hsize}{./slideFigs/picture1.jpg}
\figcaption{General GPU Design by
\href{https://commons.wikimedia.org/wiki/User:ScotXW}{ScotXW} - Own
 work, WikiPedia.}
```



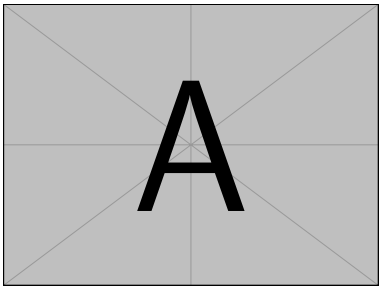
General GPU Design by ScotXW - Own work, WikiPedia.

Using Subcaption

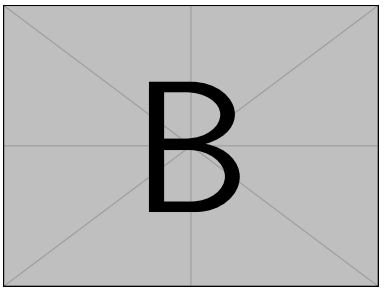
- The **subcaption** package provides the **subfigure** environment
- The **subfigure** environment can be used to align and caption subfigures inside a **figure** float
- The **subcaption** package **v3.3-111** and newer is compatible with Beamer

The subfigures will be within the **\begin{figure}...\end{figure}** environment. You may want to use the **\centering** command.
You should use the standard **\includegraphics** command to insert your illustrations.

```
\begin{figure}
\centering
  \begin{subfigure}{<width>}
   \includegraphics
    [width=<width of illustration>]{<name of graphics file>}
   \caption{<subfigure label>}
  \end{subfigure}
  (repeat subfigures as many times as desired)

        \caption{<Caption for complete figure>}
     \end{figure}
```



Subfigure label



Another subfigure label

Fig. 1. Figure caption.

Image Grids

We have grid commands, which are flexible enough to work in a single column or in the full width of the slide.

For the top line of the grid, use **\topgrid{<number of images>}**;
for the bottom line, use **\bottomgrid{<number of images>}** The number of images on each line can be between 1 and 6. Thus you can make a grid of 12 images if you want. However, keep in mind that the more images per line, the less width available for each image.

Now, you will make images and captions as many times as the number you've given as the argument to **\topgrid** or **\bottomgrid**, surrounding each with curly brackets.

We can choose either **\dofig{<graphic file location>**
\docaption{<caption>}
Or, **\shortdofig{<graphic file location>}**
\shortdocaption{<caption>}

```
\topgrid2
{\dofig{./slideFigs/penguin3}
\docaption{First penguin with a very
                        long caption.}}
%%
{\dofig{./slideFigs/penguin2}
\docaption{Second penguin.}}

\bottomgrid2{\dofig{./slideFigs/penguin5}
\docaption{Puff Penguin.}
}
%%
{\shortdofig{./slideFigs/tux-back.jpg}
\shortdocaption{TuX the Penguin Back View.}}
```


First penguin with a very long caption.


Second penguin.


Puff Penguin.


TuX the Penguin Back View.

Full Slide Width Grids

Amazingly enough, these same commands will work for full slide width as well as for single column width. There is one additional command for full slide grids: **\TallImages**. The default is to assume that the slide maker may want some explanatory text above the images. If this is not the case we can take advantage of the extra vertical space available. **\TallImages**, entered before the grid will accomplish this.

Copper Top Table

- The command `\coppertoptable{<number>}` uses the argument to determine the number of columns, between 1 to 5 columns.
- The available horizontal space will be divided by the given number, so that the same command can be used in a single column, or for the full slide width.
- The cells have alternating light and dark bands.
- Almost all formatting is done by macros in the PNNL-slides package, except that you must add the command `coppertop` to the top of each column.
- The horizontal lines are made with `\bigwhitehline` for the line underneath the column headers; and `\whitehline` for the following lines.
- Optionally you can use `\rr` for a ragged right that works within tables.

Here is a code example with results. Easy? You can see more examples in the SlideSamples.tex/.pdf files, but we've just covered all the necessary commands.

```
\coppertoptable{2}
\coppertop
Two Column Table&\coppertop Col Two\\
\bigwhitehline
\rr Cell Content goes here and wraps to fit.&\rr Second Column\\
\whitehline
Single Line&Single Line\\
\whitehline
1234&5678
\end{tabular}
```

Two Column Table	Col Two
Cell Content goes here and wraps to fit.	Second Column
Single Line	Single Line
1234	5678

Copper Top Table with Variable Column Width

With the command **\setcolwidth{<column number>}{<column width>}** we can set the width of columns in **\coppertoptable{}** explicitly, instead of having each column be the same width.

We use these settings before the start of the **\coppertoptable{}**:
\setcolwidth{<column number>}{<column width>}.

These commands will work in either single column or full width slide. Widths can be set using percent of **\hsize** or a particular point width.

For example

```
\setcolwidth{1}{.3\hsize}
\setcolwidth{2}{.2\hsize}
\setcolwidth{3}{.1\hsize}
\setcolwidth{4}{.1\hsize}
\setcolwidth{5}{.2\hsize}

\coppertoptable{5}
\coppertop
Typical Table&
...
\end{tabular}
```

Now we try setting the width of each column explicitly with the `\setcolwidth{}{}` command, used: `\setcolwidth{<column number>}{<column width>}`.

Typical Table	With Headings that wrap	Col 3	Col 4	Col 5
Cell Content goes here and wraps to fit	This is Arial 18	I'd go with whole numbers for font sizes.	Another	Again
Single line	Single line	Single line	Single line	Single line
1234	567	89	10 11	12 13

Making Tables in PNNL Style

The PNNL tables are very similar to the standard LaTeX tables. Yet with a minimum of extra commands, we are able to produce the stylized and elegant PNNL table. Here are the additional commands that are used in this style:

- Before `\pnnltable{` you can change the dimension `\tabcolsep`; ie, `\tabcolsep=12pt`. This will determine the width between columns.
- `\pnnltable{...}` for correct formatting and gray bars on alternating lines.
- Within pnnltable:

  `\topcopperhline`, at top of table
  `\midcopperhline` below headers
  `\bottomcopperhline` at bottom of table

  `\tabnote{}` for raised letter, to be referred to in tablenotes.

  An oddity: NOTE; the caption goes ***within*** the tabular environment.

Table notes are used AFTER end of pnnltable, but before `\end{table}`:

```
\begin{tablenotes}
\tablenote{<letter>}{<text>}
(repeat as often as necessary}
\end{tablenotes}
```

Syntax for \pnnltable

```
 \begin{table}

 \pnnltable{
 \begin{tabular}{<table preamble>}
 \tabcaption{Caption here} / or \numberedtabcaption{Caption}
  <table contents>
\end{tabular}
            } %% <<== end pnnltable{}
 \begin{tablenotes}
 \tablenote{(a)}{text}
 \tablenote{(b)}{text} etc.
 \end{tablenotes}
\end{table}
```

Sample \pnnltable Code

```
\tabcolsep=12pt
\pnnltable{\begin{tabular}{lccccccc}
\tabcaption{Table with Optional Caption.\tabnote{(a)}}
\topcopperhline
&$F$&$\rho_b$&&$V_w$&$v$&$t_o$&&
$K_d$ \\
\multicolumn1c{Experiment\tabnote{(b)}}&
$(cm^3/hr)$&
$(g/cm^3)$&
$\theta$&
$(mL)$&
$(cm/hr)$&
$V_w$&
$R$&
$(mL/g)$\\
\midcopperhline
Sodium orthophosphate
&
30.37& 1.478& 0.386& 20.89& 16.01& 11.22& 5.54& 1.19\\
Sodium pyrophosphate& 41.93&1.44\phantom{0}& 0.385& 20.33& 22.18& 15.90& 7.61&
1.76\\
Sodium tripolyphosphate& 40.80& 1.460& 0.392& 21.27& 21.22& 14.70&
5.17& 1.12\\
Calcium& 31.41& 1.478& 0.386& 20.89& 16.57& 11.95& \llap{1}4.14& 3.44\\
\bottomcopperhline
\end{tabular}}
\begin{tablenotes}
\tablenote{(a)}{$F$ = flow rate; $\rho_b$ = bulk density; $\omega =
$average volumetric water content (standard deviation); $V_w = $average pore
volume; $v = $average pore water velocity; $t_o = $step input; $R =
$retardation factor; $K_d = $sediment water distribution
coefficient based on $R$.}

\tablenote{(b)}{Columns appeared saturated and had reached a stable
water content. }
\end{tablenotes}
```

Resulting PNNL Table

Table with Optional Caption.[a]

Experiment[b]	F (cm^3/hr)	ρ_b (g/cm^3)	θ	V_w (mL)	v (cm/hr)	t_o V_w	R	K_d (mL/g)
Sodium orthophosphate	30.37	1.478	0.386	20.89	16.01	11.22	5.54	1.19
Sodium pyrophosphate	41.93	1.44	0.385	20.33	22.18	15.90	7.61	1.76
Sodium tripolyphosphate	40.80	1.460	0.392	21.27	21.22	14.70	5.17	1.12
Calcium	31.41	1.478	0.386	20.89	16.57	11.95	14.14	3.44

(a) F = flow rate; ρ_b = bulk density; ω =average volumetric water content (standard deviation); V_w =average pore volume; v =average pore water velocity; t_o =step input; R =retardation factor; K_d =sediment water distribution coefficient based on R.

(b) Columns appeared saturated and had reached a stable water content.

Final Slide

The final slide of your slide set is formatted slightly differently from the other slides, with more of the page being taken by the background graphic.

To get this effect, the user should type either **\thankyou**, to get the page format and the word 'Thank You' on the slide, or

\finalslide which will format the last slide correctly and leave space for final contact information, or perhaps acknowledgments.