WeBWorK: Open Source On-line Homework for Mathematics

P. Gavin LaRose

Department of Mathematics
University of Michigan
glarose@umich.edu

21 September 2018

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Outline

- A brief and incomplete history of web homework systems, 1992–present.
- Some similarities and differences between current systems.
- Capabilities and characteristics of WeBWorK and (some) other systems.
- WeBWorK demonstration, including student and instructor views.
- Some details of WeBWorK
  - Features,
  - Philosophy, and
  - Neat and Technical Stuff.
- Conclusions, such as they are.

```
# normal_prob(a, b, mean=>0, deviation=>1);
$n = random(2,4,1);
$p = random(0.2, 0.5, 0.1);
$pans1 = Compute( $p );
$ex = $n*$p;
$pans1rule = $pans1->ans_rule();

# first set up some probabilities that work
$pans2 = Compute( 0 );
$pans2rule = $pans1->ans_rule();
@pvals = ();
push( @pvals, Compute( "1 - 2*$p" ) );
for ( my $i=1; $i<$n; $i++ ) {
  push( @pvals, Compute( "2*$p/($n-1)" ) );
}
push( @pvals );

# then we can set up the multianswer for the probabilities
$mp = MultiAnswer( @pvals )->with(
    singleResult=>0,
    checker=>sub {
        my ( $c, $s, $ansHash ) = @_;
        my @c = @$c;
        my @s = @$s;

        my $tprob = 0;
        my $eval  = 0;
        for ( my $i=0; $i<@s; $i++ ) {
            $tprob += $s[$i];
            $eval  += $i*$s[$i];
        }
```

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

Early Web Homework: a Brief and Incomplete History

- 1992 : *CAPA (Computer-Assisted Personalized Approach)* piloted in a small physics class of 92 students, at Michigan State University. This is now the homework/testing engine behind LON-CAPA.
- 1993 : *First graphical web browser*, Mosaic, released by NCSA. (Replaced by Netscape Navigator between 1995–97.)
- 1995 : *WeBWorK* developed by Mike Gage and Arnie Pizer at the University of Rochester.
- 1997 : *WebAssign* developed by John Risley at North Carolina State University.
- c.1999 : *eGrade* developed by John Orr at the Univeristy of Nebraska (as a testing platform, initially). This was licensed by Wiley as Wiley eGrade (c.2001), and was forked to be the homework/testing system underlying Maple T.A. in 2003.

The Past Decade

- 2009 : The AMS undertook a survey of US mathematics departments to assess use of and experiences with on-line homework (contacted 1230 departments; responses from 467, including 57 statistics departments).
  - About half of doctoral mathematics departments were using on-line homework.
  - MyMathLab had the largest overall user base (110 departments, 230,000 students annually).
  - WebAssign (80 users, 100,000 students) and WeBWorK (55 users, 100,000 students) were the next most popular.

n.b.: the survey predated Cengage's acquisition of WebAssign.

The Past Decade

- 2009 : The AMS undertook a survey of US mathematics departments to assess use of and experiences with on-line homework (contacted 1230 departments; responses from 467, including 57 statistics departments).
  - About half of doctoral mathematics departments were using on-line homework.
  - MyMathLab had the largest overall user base (110 departments, 230,000 students annually).
  - WebAssign (80 users, 100,000 students) and WeBWorK (55 users, 100,000 students) were the next most popular.

n.b.: the survey predated Cengage's acquisition of WebAssign.

The Past Decade

- 2009 : The AMS undertook a survey of US mathematics departments to assess use of and experiences with on-line homework (contacted 1230 departments; responses from 467, including 57 statistics departments).
  - About half of doctoral mathematics departments were using on-line homework.
  - MyMathLab had the largest overall user base (110 departments, 230,000 students annually).
  - WebAssign (80 users, 100,000 students) and WeBWorK (55 users, 100,000 students) were the next most popular.

n.b.: the survey predated Cengage's acquisition of WebAssign.

The Past Decade

- 2009 : The AMS undertook a survey of US mathematics departments to assess use of and experiences with on-line homework (contacted 1230 departments; responses from 467, including 57 statistics departments).
  - About half of doctoral mathematics departments were using on-line homework.
  - MyMathLab had the largest overall user base (110 departments, 230,000 students annually).
  - WebAssign (80 users, 100,000 students) and WeBWorK (55 users, 100,000 students) were the next most popular.

n.b.: the survey predated Cengage's acquisition of WebAssign.

The Present

- Publisher systems dominate standalone web homework:
  - MyMathLab (Pearson) boasts *over 11 million students* annually.
  - WebAssign (Cengage) claims *over 1 million, & over 2,600 inst*.
  - WileyPlus (Wiley); ALEKS (McGraw Hill). . .
- Independent systems have a smaller user base
  - WeBWorK *(over 1,300 institutions)*
  - Maple T.A., rebranded Möbius (spun off to DigitalEd)
  - . . . and IMathAS ("Internet Mathematics Assessment System"), which powers MyOpenMath and LumenOHM, Ximera (Ohio State—an "interactive textbook platform"), local systems (Kansas State,. . .), . . .

- *currently developing:*
  - EdFinity (claims *over 500,000 students, & over 3,500 institutions*).

The Present

- Publisher systems dominate standalone web homework:
  - MyMathLab (Pearson) boasts *over 11 million students* annually.
  - WebAssign (Cengage) claims *over 1 million, & over 2,600 inst*.
  - WileyPlus (Wiley); ALEKS (McGraw Hill). . .
- Independent systems have a smaller user base
  - WeBWorK *(over 1,300 institutions)*
  - Maple T.A., rebranded Möbius (spun off to DigitalEd)
  - . . . and IMathAS ("Internet Mathematics Assessment System"), which powers MyOpenMath and LumenOHM, Ximera (Ohio State—an "interactive textbook platform"), local systems (Kansas State,. . .), . . .

- *currently developing:*
  - EdFinity (claims *over 500,000 students, & over 3,500 institutions*).

The Present

- Publisher systems dominate standalone web homework:
  - MyMathLab (Pearson) boasts *over 11 million students* annually.
  - WebAssign (Cengage) claims *over 1 million, & over 2,600 inst*.
  - WileyPlus (Wiley); ALEKS (McGraw Hill). . .
- Independent systems have a smaller user base
  - WeBWorK *(over 1,300 institutions)*
  - Maple T.A., rebranded Möbius (spun off to DigitalEd)
  - . . . and IMathAS ("Internet Mathematics Assessment System"), which powers MyOpenMath and LumenOHM, Ximera (Ohio State—an "interactive textbook platform"), local systems (Kansas State,. . .), . . .

- *currently developing:*
  - EdFinity (claims *over 500,000 students, & over 3,500 institutions*).

The Present

- Publisher systems dominate standalone web homework:
  - MyMathLab (Pearson) boasts *over 11 million students* annually.
  - WebAssign (Cengage) claims *over 1 million, & over 2,600 inst*.
  - WileyPlus (Wiley); ALEKS (McGraw Hill). . .
- Independent systems have a smaller user base
  - WeBWorK *(over 1,300 institutions)*
  - Maple T.A., rebranded Möbius (spun off to DigitalEd)
  - . . . and IMathAS ("Internet Mathematics Assessment System"), which powers MyOpenMath and LumenOHM, Ximera (Ohio State—an "interactive textbook platform"), local systems (Kansas State,. . .), . . .

- *currently developing:*
  - EdFinity (claims *over 500,000 students, & over 3,500 institutions*).

System Comparisons

- Pricing Model

Stu. license	Inst. license	Inst. support	Ext. support
MyMathLab, WebAssign,. . . , Möbius ($29/s) LumenOHM ($25/s)	*Möbius, LumenOHM WebAssign?*	*WeBWorK, IMathAS*	*Ximera, MyOpenMath, "WeBWorK"*

- Hosting Model

Cloud	Local
MyMathLab, WebAssign,. . . , Möbius, LumenOHM, Ximera, MyOpenMath, WeBWorK	*WeBWorK, IMathAS*

- LMS Interoperability

Advanced	Basic	None
Commercial, WeBWorK,. . .	*IMathAS, LumenOHM?*	*MyOpenMath?, Ximera?*

- Authoring and Content Ownership

System Comparisons

- Pricing Model

Stu. license	Inst. license	Inst. support	Ext. support
MyMathLab, WebAssign,. . . , Möbius ($29/s) LumenOHM ($25/s)	*Möbius, LumenOHM WebAssign?*	*WeBWorK, IMathAS*	*Ximera, MyOpenMath, "WeBWorK"*

- Hosting Model

Cloud	Local
MyMathLab, WebAssign,. . . , Möbius, LumenOHM, Ximera, MyOpenMath, WeBWorK	*WeBWorK, IMathAS*

- LMS Interoperability

Advanced	Basic	None
Commercial, WeBWorK,. . .	*IMathAS, LumenOHM?*	*MyOpenMath?, Ximera?*

- Authoring and Content Ownership

System Comparisons

- Pricing Model

Stu. license	Inst. license	Inst. support	Ext. support
MyMathLab, WebAssign,. . . , Möbius ($29/s) LumenOHM ($25/s)	*Möbius, LumenOHM WebAssign?*	*WeBWorK, IMathAS*	*Ximera, MyOpenMath, "WeBWorK"*

- Hosting Model

Cloud	Local
MyMathLab, WebAssign,. . . , Möbius, LumenOHM, Ximera, MyOpenMath, WeBWorK	*WeBWorK, IMathAS*

- LMS Interoperability

Advanced	Basic	None
Commercial, WeBWorK,. . .	*IMathAS, LumenOHM?*	*MyOpenMath?, Ximera?*

- Authoring and Content Ownership

System Comparisons

- Pricing Model

Stu. license	Inst. license	Inst. support	Ext. support
MyMathLab, WebAssign,. . . , Möbius ($29/s) LumenOHM ($25/s)	*Möbius, LumenOHM WebAssign?*	*WeBWorK, IMathAS*	*Ximera, MyOpenMath, "WeBWorK"*

- Hosting Model

Cloud	Local
MyMathLab, WebAssign,. . . , Möbius, LumenOHM, Ximera, MyOpenMath, WeBWorK	*WeBWorK, IMathAS*

- LMS Interoperability

Advanced	Basic	None
Commercial, WeBWorK,. . .	*IMathAS, LumenOHM?*	*MyOpenMath?, Ximera?*

- Authoring and Content Ownership

System Comparisons

- Pricing Model

Stu. license	Inst. license	Inst. support	Ext. support
MyMathLab, WebAssign,. . . , Möbius ($29/s) LumenOHM ($25/s)	*Möbius, LumenOHM WebAssign?*	*WeBWorK, IMathAS*	*Ximera, MyOpenMath, "WeBWorK"*

- Hosting Model

Cloud	Local
MyMathLab, WebAssign,. . . , Möbius, LumenOHM, Ximera, MyOpenMath, WeBWorK	*WeBWorK, IMathAS*

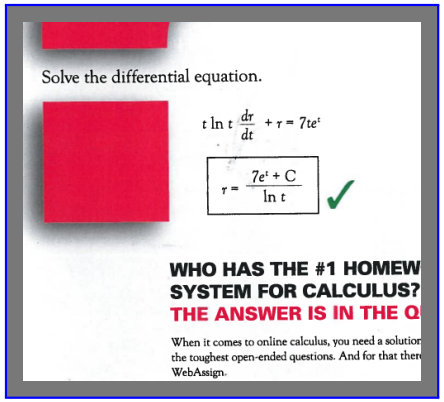
- LMS Interoperability

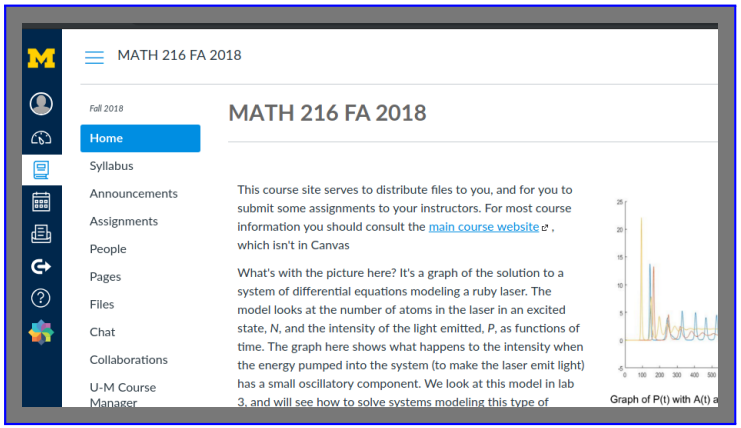
Advanced	Basic	None
Commercial, WeBWorK,. . .	*IMathAS, LumenOHM?*	*MyOpenMath?, Ximera?*

- Authoring and Content Ownership

Common Capabilities/Features

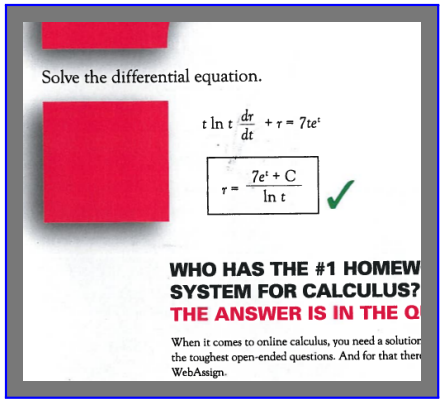
- Similarities between WeBWorK and Publisher Systems
  - Robust and varied answer checking.
    - *CAS integration vs. programmatic and MathObjects answer checking, Sage. . .*.
  - Significant problem libraries.
    (WeBWorK's OPL currently has 35,000 problems)
    - *Some direct textbook problem tagging*.
  - LTI Support (that is, connections to any modern LMS).

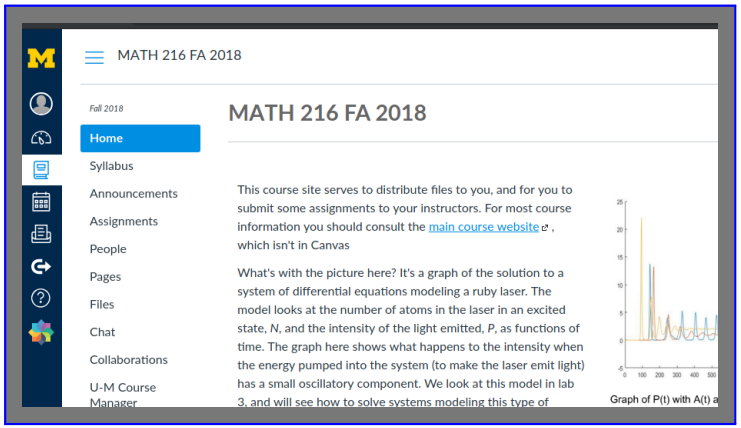




Common Capabilities/Features

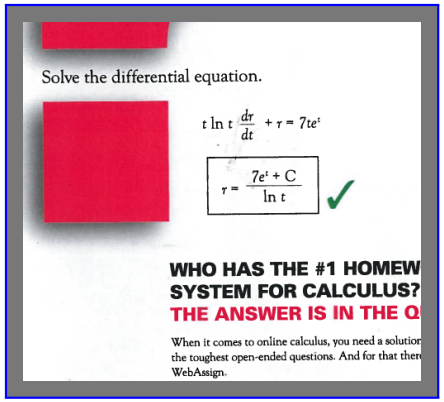
- Similarities between WeBWorK and Publisher Systems
  - Robust and varied answer checking.
    - *CAS integration vs. programmatic and MathObjects answer checking, Sage. . .*.
  - Significant problem libraries.
    (WeBWorK's OPL currently has 35,000 problems)
    - *Some direct textbook problem tagging*.
  - LTI Support (that is, connections to any modern LMS).

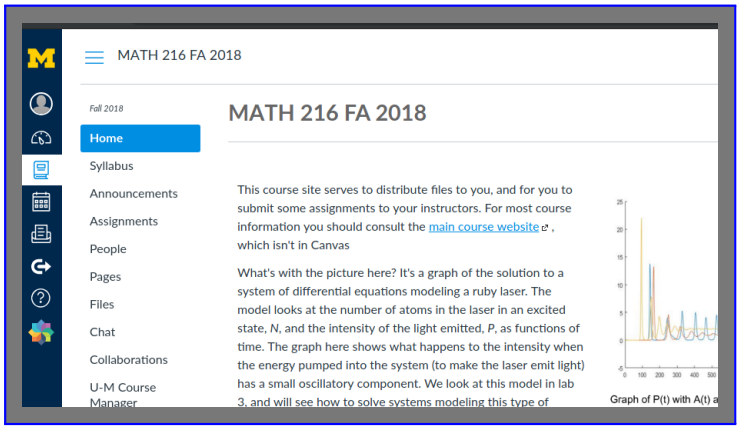




Common Capabilities/Features

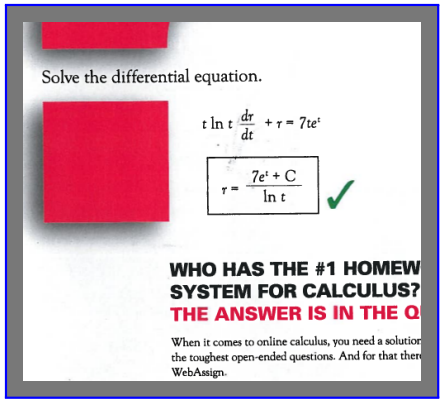
- Similarities between WeBWorK and Publisher Systems
  - Robust and varied answer checking.
    - *CAS integration vs. programmatic and MathObjects answer checking, Sage. . .*.
  - Significant problem libraries.
    (WeBWorK's OPL currently has 35,000 problems)
    - *Some direct textbook problem tagging*.
  - LTI Support (that is, connections to any modern LMS).

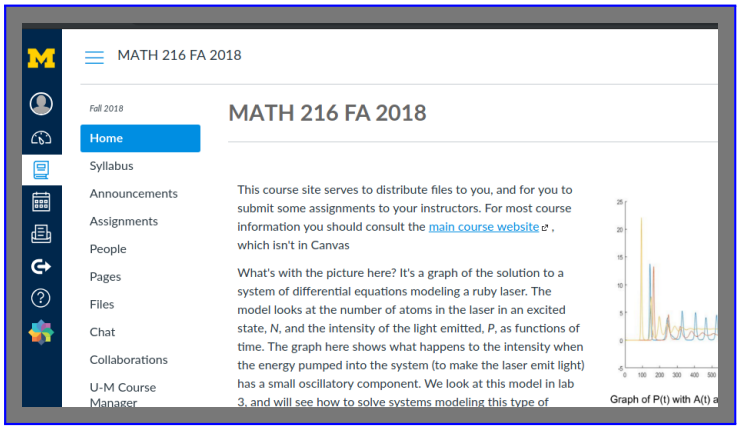




Common Capabilities/Features

- Similarities between WeBWorK and Publisher Systems
  - Robust and varied answer checking.
    - *CAS integration vs. programmatic and MathObjects answer checking, Sage. . .*.
  - Significant problem libraries.
    (WeBWorK's OPL currently has 35,000 problems)
    - *Some direct textbook problem tagging*.
  - LTI Support (that is, connections to any modern LMS).





Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . . .*

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No “plug and play.”
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . . .*

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . . .*

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . . .*

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . . .*

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . .*.

Differences in Capabilities/Features

- Differences between WeBWorK and Publisher Systems
  - Incomplete text problems
    - *But: textbook problems tend to be similar. . .*
  - No links back to textbooks
  - No "plug and play."
    - *But: LMS Interoperability*
- Other Thoughts
  - SSO support exists in WeBWorK, and maybe other systems.
  - Features generally exceed *used* features.
  - Paying usually buys something
    - *Problem-text matching, data management, server maintenance, cosmetics. . . low student cost, server control, ownership of material. . .*.

WeBWorK Demonstration

- Student view
  - Setup
  - Homework assignments
  - (Gateway) Tests
  - Settings & Grades
- Administrator view
  - Web interface—Add/edit courses, manage locations.
- Instructor view
  - Setup
  - Homework assignments—added information, views, answer checking, visibility.
  - Classlist and homework editors—import, export, edit.
  - Student Progress & Grades
  - Library Browser and OPL structure
  - Gateway test set-up

WeBWorK Demonstration

- Student view
  - Setup
  - Homework assignments
  - (Gateway) Tests
  - Settings & Grades
- Administrator view
  - Web interface—Add/edit courses, manage locations.

- Instructor view
  - Setup
  - Homework assignments—added information, views, answer checking, visibility.
  - Classlist and homework editors—import, export, edit.
  - Student Progress & Grades
  - Library Browser and OPL structure
  - Gateway test set-up

WeBWorK Demonstration

- Student view
  - Setup
  - Homework assignments
  - (Gateway) Tests
  - Settings & Grades
- Administrator view
  - Web interface—Add/edit courses, manage locations.

- Instructor view
  - Setup
  - Homework assignments—added information, views, answer checking, visibility.
  - Classlist and homework editors—import, export, edit.
  - Student Progress & Grades
  - Library Browser and OPL structure
  - Gateway test set-up

WeBWorK Demonstration

- Student view
  - Setup
  - Homework assignments
  - (Gateway) Tests
  - Settings & Grades
- Administrator view
  - Web interface—Add/edit courses, manage locations.

- Instructor view
  - Setup
  - Homework assignments—added information, views, answer checking, visibility.
  - Classlist and homework editors—import, export, edit.
  - Student Progress & Grades
  - Library Browser and OPL structure
  - Gateway test set-up

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students.

  . . . and everything just works

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS.

  . . . and everything just works.

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students.

  . . . and everything just works

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS.

  . . . and everything just works.

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students. *. . . and everything just works*

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS. *. . . and everything just works.*

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students.

  . . . and everything just works

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS.

  . . . and everything just works.

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students. *. . . and everything just works*

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS. *. . . and everything just works.*

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students.

  . . . and everything just works

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS.

  . . . and everything just works.

WeBWorK Workflow

- Traditional workflow
  1. WeBWorK Administrator creates course(s).
  2. Instructor creates assignments.
  3. Instructor or administrator adds students.

  . . . and everything just works

- LTI workflow
  1. WeBWorK Administrator creates course(s), and configures it (them) to use LTI.
  2. Instructor creates assignments, and configures the LMS to use WeBWorK as an LTI Tool for assignments.
  3. Students are automatically added as they log in through the LMS.

  . . . and grades may be passed back to the LMS.

  . . . and everything just works.

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

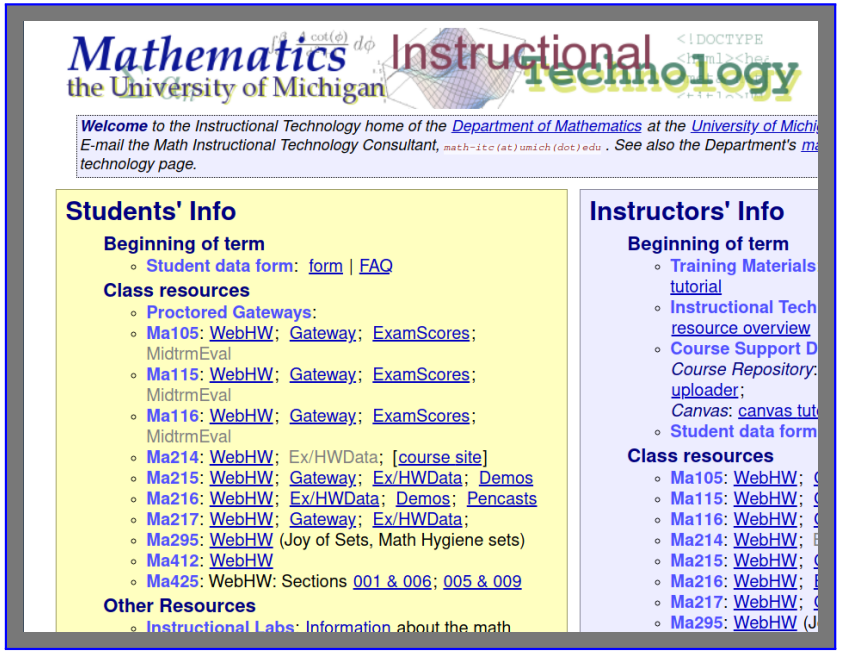
Underpinnings and Further Use Cases

- What is WeBWorK?
  - A *Perl Apache module* that is loaded into an apache server process,
  - with most data in a *MySQL database*.
  - Additional data (problem source code, macro files, classlist import files, set definition import files. . .) are stored in *small text files* on the webserver.
- Where does it live for a given University?
  - Locally, or
  - Cloud-based (e.g., AWS), or
  - Alternate provider (e.g., MAA hosting, university consortium. . .)

User:Chassisplans

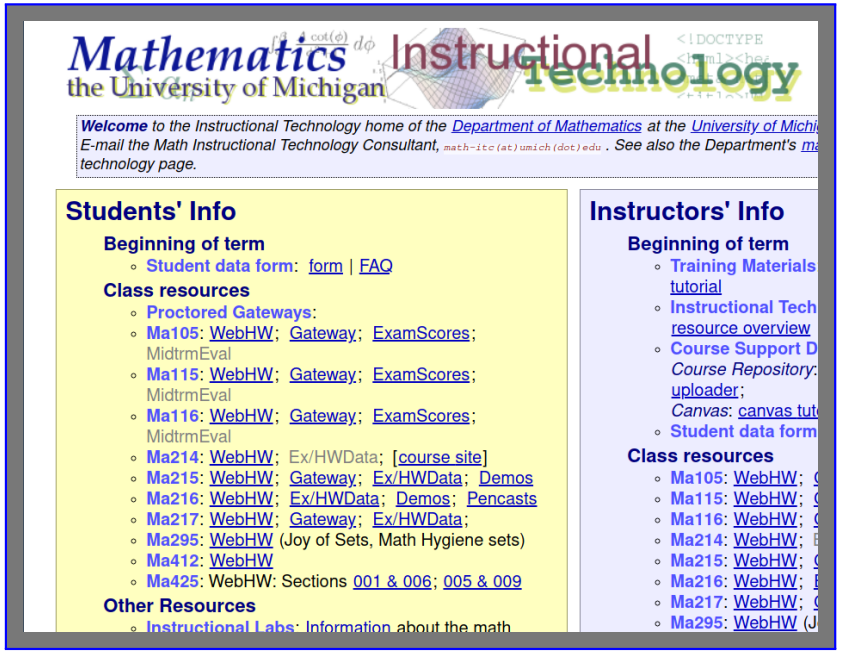
WeBWorK at Michigan

- Locally hosted, large installation.
  - ≈230 WeBWorK courses for fall 2018.
  - ≈50 hosted for UM Dearborn, using LTI to integrate with Canvas.
  - UM AA courses run through SSO.



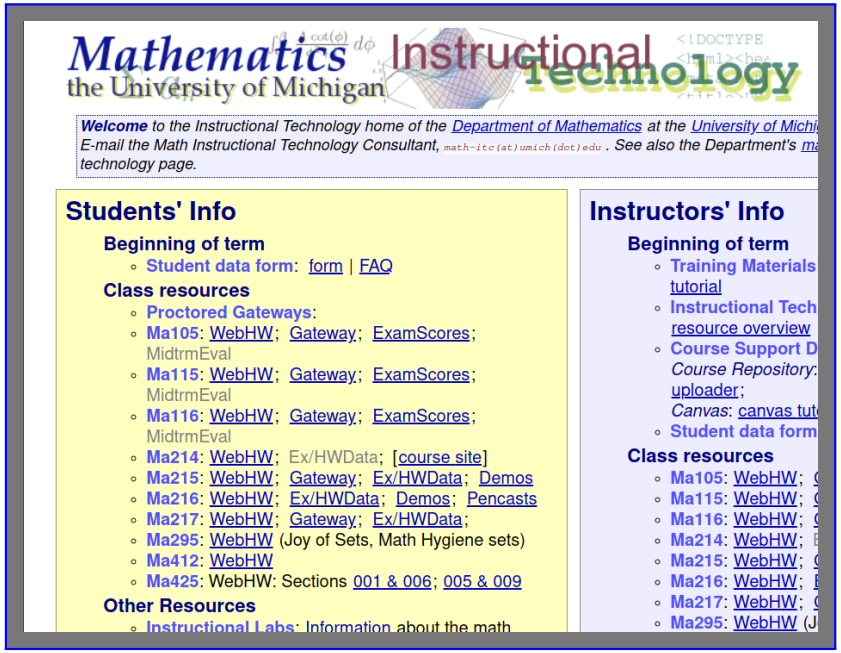
WeBWorK at Michigan

- Locally hosted, large installation.
  - ≈230 WeBWorK courses for fall 2018.
  - ≈50 hosted for UM Dearborn, using LTI to integrate with Canvas.
  - UM AA courses run through SSO.



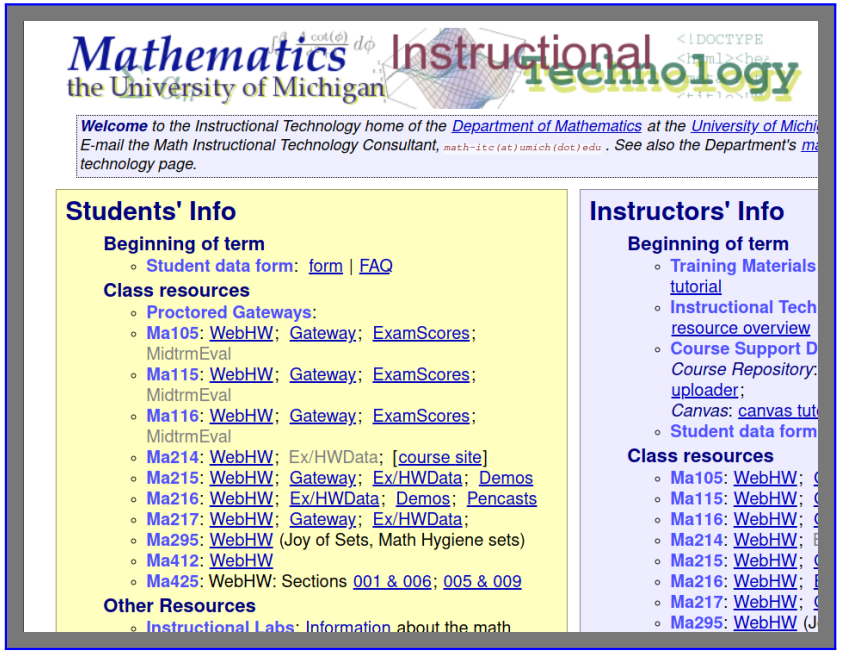
WeBWorK at Michigan

- Locally hosted, large installation.
  - ≈230 WeBWorK courses for fall 2018.
  - ≈50 hosted for UM Dearborn, using LTI to integrate with Canvas.
  - UM AA courses run through SSO.



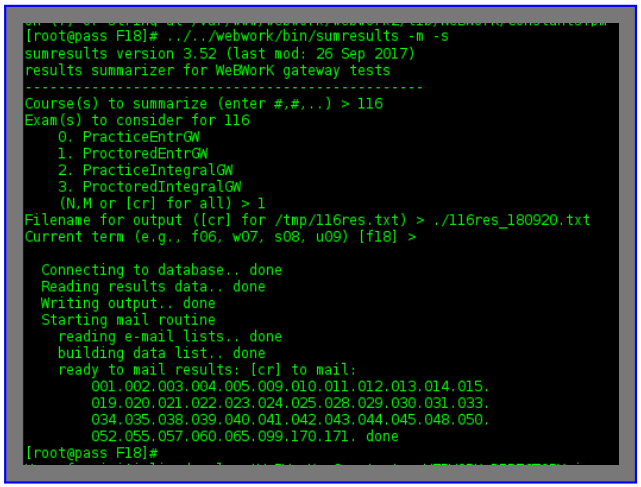
WeBWorK at Michigan

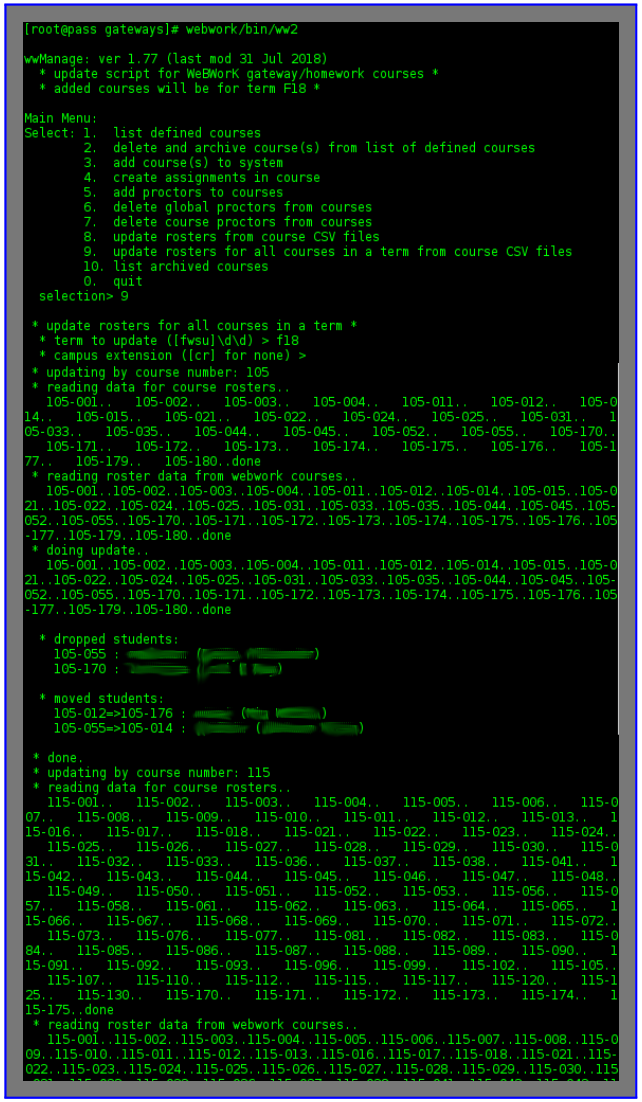
- Locally hosted, large installation.
  - ≈230 WeBWorK courses for fall 2018.
  - ≈50 hosted for UM Dearborn, using LTI to integrate with Canvas.
  - UM AA courses run through SSO.



Underpinnings and Implications

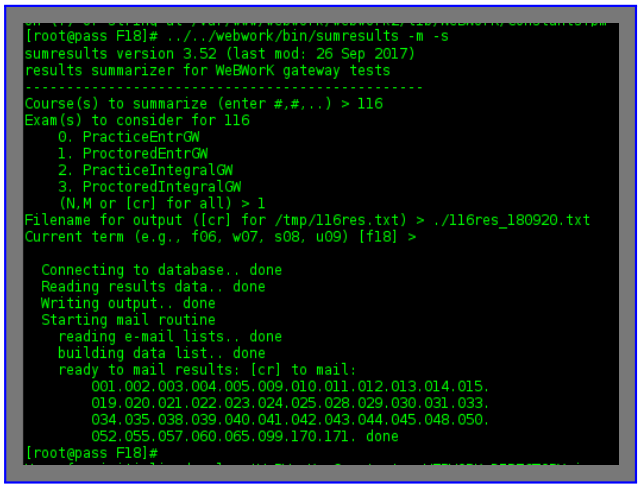
- Because data are in a database, or files:
  - It is possible to automate administrative tasks.
  - It is possible to provide additional service or student/instructor interaction.

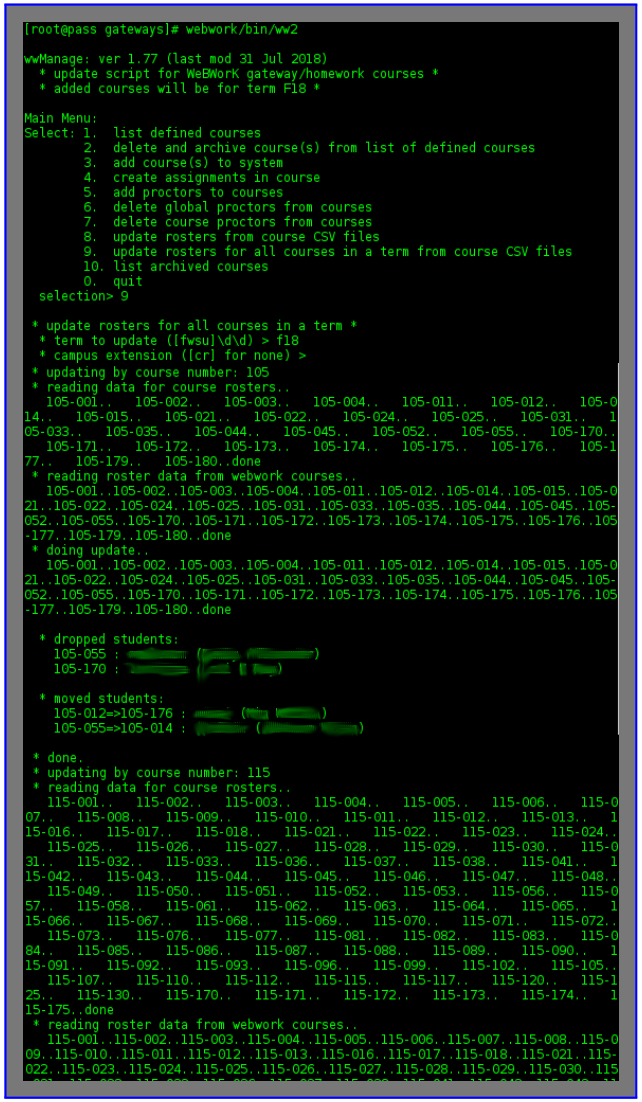




Underpinnings and Implications

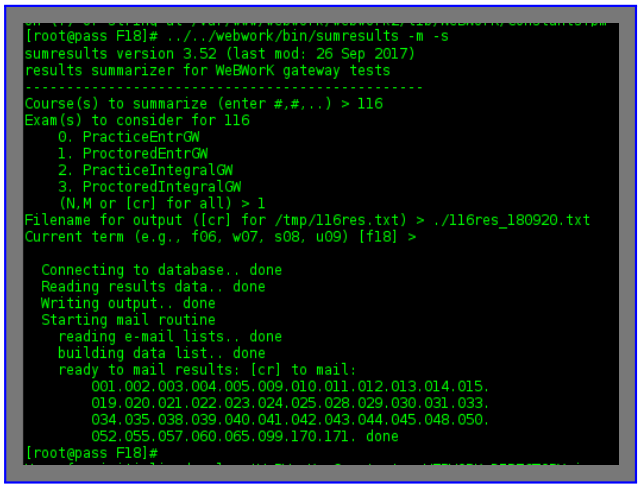
- Because data are in a database, or files:
  - It is possible to automate administrative tasks.
  - It is possible to provide additional service or student/instructor interaction.

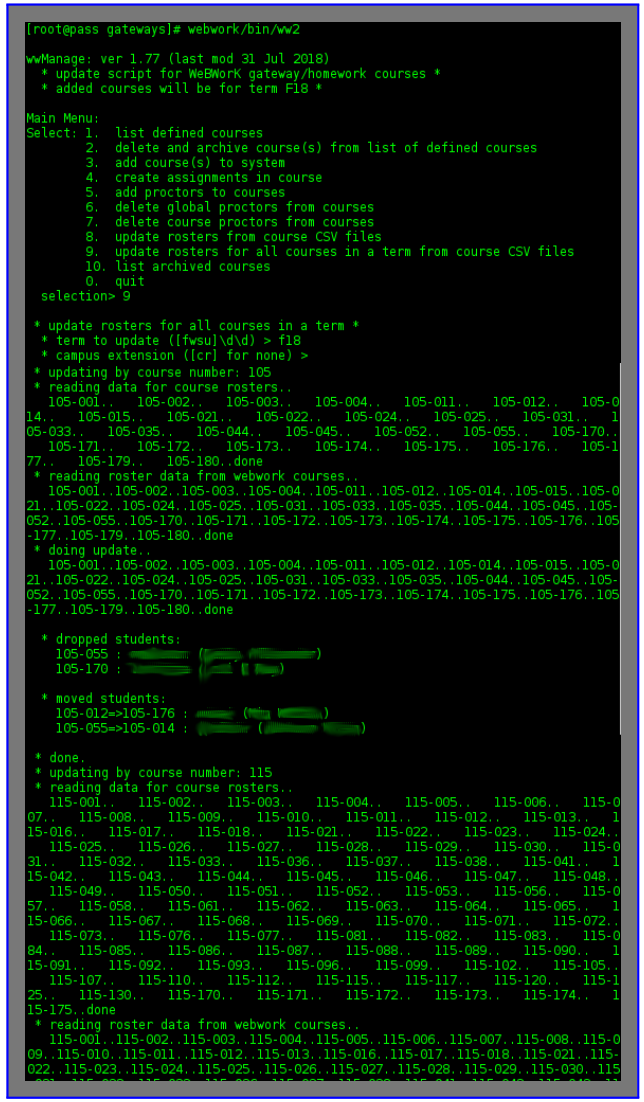




Underpinnings and Implications

- Because data are in a database, or files:
  - It is possible to automate administrative tasks.
  - It is possible to provide additional service or student/instructor interaction.





The Philosophy of WeBWorK

- Why WeBWorK?
  think: LaTeX
- What should it do?
  - *Let you do what you want*.
    Implication: Not necessarily to make it easy to code problems.

    But: one thing we want to do is write problems in LaTeX.

```
## EditionText1('7')
## AuthorText1('Hughes-Hallett')
## Section1('8.2')
## Problem1('1')

DOCUMENT();
loadMacros("PGstandard.pl", "MathObjects.pl", "PGML.pl",
           "parserPopUp.pl");

Context("Numeric");
$m  = random(2,4,1); $x0 = random(1,3,1); $b  = $m*$x0;

TEXT(beginproblem());
BEGIN_PGML
The region bounded by the line [` y = [$b] - [$m] x `] between [`x = 0`]
and [`x = [$x0]`] is revolved around the [`x`]-axis.  To find the volume
of this solid of revolution, we would evaluate the integral
    [`\int_0^{[$x0]} `] [__]{"pi($b - $m*x)^2"}{20} [`dx`]
END_PGML

# give a solution (available after the due date)
# BEGIN_PGML_SOLUTION
#_
# END_PGML_SOLUTION

ENDDOCUMENT();
```

The Philosophy of WeBWorK

- Why WeBWorK?
  think: LaTeX
- What should it do?
  - *Let you do what you want*.
    Implication: Not necessarily to make it easy to code problems.

    But: one thing we want to do is write problems in LaTeX.

```
## EditionText1('7')
## AuthorText1('Hughes-Hallett')
## Section1('8.2')
## Problem1('1')

DOCUMENT();
loadMacros("PGstandard.pl", "MathObjects.pl", "PGML.pl",
           "parserPopUp.pl");

Context("Numeric");
$m  = random(2,4,1); $x0 = random(1,3,1); $b  = $m*$x0;

TEXT(beginproblem());
BEGIN_PGML
The region bounded by the line [` y = [$b] - [$m] x `] between [`x = 0`]
and [`x = [$x0]`] is revolved around the [`x`]-axis.  To find the volume
of this solid of revolution, we would evaluate the integral
    [`\int_0^{[$x0]} `] [__]{"pi($b - $m*x)^2"}{20} [`dx`]
END_PGML

# give a solution (available after the due date)
# BEGIN_PGML_SOLUTION
#_
# END_PGML_SOLUTION

ENDDOCUMENT();
```

The Philosophy of WeBWorK

- Why WeBWorK?
  think: LaTeX
- What should it do?
  - *Let you do what you want*.
    Implication: Not necessarily to make it easy to code problems.

    But: one thing we want to do is write problems in LaTeX.

```
## EditionText1('7')
## AuthorText1('Hughes-Hallett')
## Section1('8.2')
## Problem1('1')

DOCUMENT();
loadMacros("PGstandard.pl", "MathObjects.pl", "PGML.pl",
           "parserPopUp.pl");

Context("Numeric");
$m  = random(2,4,1); $x0 = random(1,3,1); $b  = $m*$x0;

TEXT(beginproblem());
BEGIN_PGML
The region bounded by the line [` y = [$b] - [$m] x `] between [`x = 0`]
and [`x = [$x0]`] is revolved around the [`x`]-axis.  To find the volume
of this solid of revolution, we would evaluate the integral
    [`\int_0^{[$x0]} `] [__]{"pi($b - $m*x)^2"}{20} [`dx`]
END_PGML

# give a solution (available after the due date)
# BEGIN_PGML_SOLUTION
#_
# END_PGML_SOLUTION

ENDDOCUMENT();
```

Neat Tricks

- Note: WeBWorK problems are small Perl programs.
  - Thus: it's possible to check any answer for which one can write an algorithmic test.
  - But *it's better than that!* They are Perl programs with added macros that allow mathematical operations:
    - ```
      $f = Formula('sin(x^2 + 3)');
      $f_deriv = $f->D();
      $tanline = $f->eval(x=>$x0) +
          $f_deriv->eval(x=>$x0)*(x - $x0);
      ```
    - ```
      $v2 = Compute("<-3,1>");
      ANS( $v2->cmp( checker=>sub {
          my ($correct, $student, $ansHash) = @_;
          return $correct->isParallel($student);
          }, showCoordinateHints => 0 ) );
      ```
  - . . . and more: determine relationships between student answers, disallow use of functions or operations. . .

Neat Tricks

- Note: WeBWorK problems are small Perl programs.
  - Thus: it's possible to check any answer for which one can write an algorithmic test.
  - But *it's better than that!* They are Perl programs with added macros that allow mathematical operations:
    - ```
      $f = Formula('sin(x^2 + 3)');
      $f_deriv = $f->D();
      $tanline = $f->eval(x=>$x0) +
          $f_deriv->eval(x=>$x0)*(x - $x0);
      ```
    - ```
      $v2 = Compute("<-3,1>");
      ANS( $v2->cmp( checker=>sub {
          my ($correct, $student, $ansHash) = @_;
          return $correct->isParallel($student);
          }, showCoordinateHints => 0 ) );
      ```
  - . . . and more: determine relationships between student answers, disallow use of functions or operations. . .

Neat Tricks

- Note: WeBWorK problems are small Perl programs.
  - Thus: it's possible to check any answer for which one can write an algorithmic test.
  - But *it's better than that!* They are Perl programs with added macros that allow mathematical operations:
    - ```
      $f = Formula('sin(x^2 + 3)');
      $f_deriv = $f->D();
      $tanline = $f->eval(x=>$x0) +
          $f_deriv->eval(x=>$x0)*(x - $x0);
      ```
    - ```
      $v2 = Compute("<-3,1>");
      ANS( $v2->cmp( checker=>sub {
          my ($correct, $student, $ansHash) = @_;
          return $correct->isParallel($student);
          }, showCoordinateHints => 0 ) );
      ```
  - . . . and more: determine relationships between student answers, disallow use of functions or operations. . .

Neat Tricks

- Note: WeBWorK problems are small Perl programs.
  - Thus: it's possible to check any answer for which one can write an algorithmic test.
  - But *it's better than that!* They are Perl programs with added macros that allow mathematical operations:
    - ```
      $f = Formula('sin(x^2 + 3)');
      $f_deriv = $f->D();
      $tanline = $f->eval(x=>$x0) +
          $f_deriv->eval(x=>$x0)*(x - $x0);
      ```
    - ```
      $v2 = Compute("<-3,1>");
      ANS( $v2->cmp( checker=>sub {
          my ($correct, $student, $ansHash) = @_;
          return $correct->isParallel($student);
          }, showCoordinateHints => 0 ) );
      ```
  - . . . and more: determine relationships between student answers, disallow use of functions or operations. . .

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's **`httpd.conf`** file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's **`httpd.conf`** file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's `httpd.conf` file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's `httpd.conf` file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's **`httpd.conf`** file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's `httpd.conf` file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's `httpd.conf` file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

and Technical Stuff

- . . . if you want to write problems.
- What about installation?
  - Needed: a server; WeBWorK installs with git, from github, and requires LaTeX, Perl, apache, MySQL (or MariaDB);
  - configuration is in apache's `httpd.conf` file, and text configuration files for WeBWorK.
  - Which usually works perfectly fine. . .
    once all the required Perl modules are installed.
  - Documentation is pretty good, though it needs freshening.
- And this works fine on cloud services (*e.g.,* AWS).
- Personal observation: WeBWorK is very stable.

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*

Conclusions

- WeBWorK vs. Publisher Systems
  - Both can provide good homework sets
  - and can be (relatively) easy to use.
  - Both have costs to use.
- We pick a system based on our situation
  - University policy.
  - Available resources (monetary, staff).
  - Philosophy and interest.
- WeBWorK is open source, robust, powerful, and has an active developer community.
  - *it's been around for 20-odd years, and there's a concerted effort to ensure it will continue.*