

Accessible Content Creation in Mathematics

Chris Hughes Scot Leavitt

April 3, 2013

Abstract

Mathematics presents many accessibility challenges that other disciplines may not have to deal with. Portland Community College (PCC) is dedicated to making education accessible to all and funded two full-time mathematics faculty members to investigate these challenges. This report summarizes the process and the findings, including best practices. The main conclusion is that there is no ‘one-size-fits-all’ answer. Every student’s needs must be addressed individually; however, there are some common practices that can help to support a wide range of students. This is not surprising considering the diverse student population that PCC encompasses.

Contents

1 Acknowledgments	1
2 Background of the project	2
3 Project process and progress	3
4 General best practices	7
5 Specific best practices	9
6 Recommendations	12
References	13
A Converting content	14
B Anticipated questions	17

1 Acknowledgments

We would like to thank the administrators that provided financial support for this project: Sue Quast, Loraine Schmitt, Andy Freed, Irene Giustini, and Dan Findley. We would also like to thank the director of Disability Services, Kaela Parks, for her support and the many enlightening discussions that she lead. Thank you to Supada Amornchat and Melany Budiman for their technical support.

We gained invaluable insight from our time with Maurice Mines– thank you for the many experiments that you helped us with and all of the discussions that we had. We would also like to thank Winslow Parker– we learned a great deal from

you and have a far better understanding of many of the aspects of life that a blind person deals with on a day-to-day basis.

Finally, we would like to thank Karen Sorensen for her continued support, encouragement, and positive attitude. Karen kept us on track, kept our spirits up when we hit roadblocks, and made the project fun and interesting from the start.

2 Background of the project

THE BEGINNING

At the start of Fall Term 2011, the Mathematics Subject Area Committee (Math SAC) was made aware of the college's initiative of accessibility in its online courses. Mathematics faculty members quickly realized that our subject matter presented unique complications not faced by other disciplines. While properly formatting documents following the college's guidelines [15] would help in the accessibility of the documents faculty create, it would not assist mathematics faculty members with the symbolic and graphical nature of our discipline.

FALL 2011

At the first SAC meeting of the 2011-12 academic year, the Math SAC formed the Mathematics Accessibility Committee. Focused on distance learning (DL) classes, this committee created several sample documents that contained various examples of mathematical content and investigated how each would be converted to a webpage and accessed using JAWS for Windows, a screen reading program for the PC. Based on those experiments and the beginning research into other areas the Mathematics Accessibility Committee would need to consider – such as Braille and Nemeth Braille Code, DAISY readers, Desire2Learn (D2L – our current learning management system), \LaTeX , Microsoft Word, and publisher-created online content – the committee quickly realized that the scope of this project would require more time than any faculty member could dedicate while also maintaining a full teaching load.

MAY 2012

This committee's opinion that release time would be necessary to truly investigate this topic was further confirmed at a presentation by Haris Gunadi (representing OSU at the time, now at PCC). The presentation focused on the different tools used to create mathematical content and the accessibility of their output.

RELEASE TIME

The Mathematics Accessibility Committee determined that it would require release time from classroom obligations for two mathematics faculty members to investigate the topic of accessible mathematical content in detail. The committee requested the release time for two faculty members given the scope of the project and the various ways members of the Math SAC produce classroom materials. By balancing the backgrounds of the two faculty members, the committee would be able to bring a greater depth of experience and knowledge to the project.

AUGUST 2012

In mid-August 2012, the Mathematics Accessibility Committee was informed that release time had been granted for one faculty member to have release from one class for two terms. While this was a positive step, the committee felt that the project would proceed better having two faculty members having a one-class release for one term. Again, the committee felt that the benefit of two people with complementary backgrounds working together was a necessary component to investigate this topic as fully as possible. Happily, the recommended revision to the project (a one-class release for two people for one term) was approved by the various deans and directors providing the financial support to the work.

FALL 2012

A few weeks prior to the start of Fall Term 2012, Chris Hughes, Scot Leavitt,

Andy Freed, and Karen Sorensen met to plan the goals and general direction of the project.

3 Project process and progress

For the following section, “we” will refer to Chris Hughes and Scot Leavitt, as this section is a narrative of our work from Fall Term 2012.

We met every Tuesday and Thursday, often accompanied by Karen and Andy. As the narrative that follows will show, we balanced our time between researching options, contacting the various companies that produce products that we use, and testing potential options for our SAC member to use.

INPUT FROM SAC

We realized that before diving into what does and doesn’t work from the perspective of accessibility, we needed to find out what SAC members were using to create and deliver the content used in their courses. Before the start of the term, we conducted a SurveyMonkey survey of the SAC asking:

- if an instructors classes are face-to-face, online, or both;
- what programs are used to create content for mathematics classes;
- how the content is delivered;
- what non-text items are used;
- how non-text items are created;
- what type of computer this material is created on;
- what textbook publisher content is used;
- how online tests/quizzes are taken;
- how homework for online classes is submitted.

This information gave us a starting point for the computer applications and online systems that we would need to investigate from the perspective of accessibility.

JAWS

Based on the initial DL-focused work done in the prior academic year, we started the project from the perspective that JAWS [6] was the ideal format for delivering accessible content for our classes. Our understanding at the time was: “If we can get the document into a webpage that uses MathML for all mathematical content, then we’re set.” Based on this understanding, we started by focusing our attention on making sure the documents and equations we create could be converted to webpages containing MathML.

MATHML

MathML stands for ‘Math Markup Language’ (which parallels html standing for ‘Hypertext Markup Language’) and was first developed by the W3C (World Wide Web Consortium) in 1998. Typically, a user does not create MathML from source, but instead uses a conversion process to get MathML. The conversion process varies depending on the software used to create the mathematical content. MathML has huge advantages from the accessibility perspective, as it can be read ‘mathematically’ by JAWS as long as the page is rendered in IE 5.5 through 8 with the MathPlayer plugin from Design Science [10] (at the time of this report).

SURVEY

We started by looking at the most common ways math faculty members create content for their courses (Microsoft Word + MathType, Microsoft Word + Microsoft’s native equation editor, L^AT_EX, and LibreOffice/OpenOffice) and how we

could export those documents to webpages containing MathML. Depending on the method of creation, the particular content, and the computer-type used, the process varied from ‘perfect’ to impossible.

MS WORD When converting a Microsoft Word document to a webpage that uses MathML, we must have Design Science’s MathType [11] installed on our computer. It is actually MathType, not Word, that does all the work. Whether we create a Word document and use MathType or Microsoft’s native equation editor, we must still rely upon MathType to create the MathML webpage. Unfortunately, not all versions of Microsoft Word and MathType can perform this conversion. We elaborate on this more in our Specific Guidelines section and in the Appendix.

L^AT_EX When converting a L^AT_EX document to a webpage that uses MathML, certain packages need to be installed. When we have these packages loaded, the conversion process generally works smoothly. Equations are cleanly converted into MathML and the webpages are read fairly well by JAWS. This conversion process is explained in the appendix of this article.

LIBREOFFICE Using LibreOffice’s built-in equation editing capabilities, LibreOffice¹ word-processed documents convert well to webpages that contain MathML. LibreOffice can do this conversion without any additional plugins or software.

POWERPOINT After finding how Word, L^AT_EX, and LibreOffice fared, we turned our attention to Microsoft PowerPoint. While we can embed MathType equations in PowerPoint slides, there is no way to convert the entire slide or slide show into a webpage with MathML. Any attempt resulted with each slide being converted into an image and all content (mathematical or not) simply became part of the image. There were some suggestions of a way to convert from PowerPoint to Word, but further investigation found that there is no useful way to do so at this time. The Word document we obtain from the process has an editable PowerPoint slide on each page, but once we convert the Word document to a webpage, the entire slide is again converted to an image.

SCIENTIFIC NOTEBOOK Scientific Notebook, an application that no one in the Math SAC currently uses, was recommended to us. Scientific Notebook is a PC application that allows a user to create documents containing text, mathematical content, and graphics. Specifically, Kaela Parks (the director of Disability Services at PCC [16]) has had tremendous success using this software as a note-taking device in the classroom. Scientific Notebook saves the source as a .tex file, which can easily be converted into several different electronic accessible formats.

TESTGEN Some faculty use Pearson’s TestGen to create their tests. TestGen is a test-generating program that, when combined with the test bank for a particular Pearson textbook, can be used to create assessments in both print and an online environment. The output file from TestGen saves all mathematical content (fractions, expressions, equations) as images without alt text. Since the mathematical content is saved as images, JAWS cannot interpret them. We contacted Pearson about this, inquiring if they were working on a way to make the output from TestGen into something that uses MathML. The response was:

There is not a way to export a TestGen Test to a screen-reader-accessible format and the MyMathLab website will only take TestGen .tst files

¹Some may confuse LibreOffice with OpenOffice; see the appendix for some details of the history

MATHJAX Related to our investigation of MathML, we learned about MathJax. Quoting from [9]

MathJax is an open source JavaScript display engine for mathematics that works in all modern browsers.

In other words, MathJax makes mathematical content appear correctly in any modern web browser on any computer, whether the content is stored on the webpage as MathML or \LaTeX .

Before the advent of MathJax, there was vast inconsistency in a browser's ability to render MathML correctly (if at all). MathJax provides a clean way to provide uniform rendering across all modern browsers [9]: 'No more setup for readers. No more browser plugins. No more font installation... It just works.'

TWO DIRECTIONS At this point, our project began to split into two directions. One direction was focusing on online courses and the various online systems that mathematics faculty members at PCC use. The other direction came about when we met Maurice Mines.

WEB-BASED SYSTEMS While PCC's current learning management system is D2L, many faculty members make substantial use of Pearson's MyMathLab (MML), while others use the MAA's WeBWorK.

DESIRE2LEARN D2L has received the Gold Level NFB-NVA Certification from the National Federation of the Blind (NFB)[5]. Two of the optional interfaces (MathML and \LaTeX) of the equation editor tool in D2L (version 10) are JAWS friendly. The third (the graphical editor) is not. But no matter which editor a student or instructor uses to create an equation in D2L, the code obtained from the equation editor is MathML. This means that JAWS can read the discussion postings, e-mails, etc. that have equations embedded in them.

PEARSON The Math SAC currently subscribes to Pearson for the majority of its courses: MTH 60, 65, 70, 95, 111, 112, 243, and 244. All MTH courses that are taught online currently use a Pearson textbook, except for MTH 20. As such, when considering online publisher-based content we decided to focus exclusively on Pearson's MyMathLab.

MYMATHLAB We explored MML by creating a sample course and experimented with the accessible nature of optional content for the course. We asked for, and eventually obtained, a Voluntary Product Accessibility Template (VPAT) for MML. While we have obtained the VPAT for MML, it is not included in this article as we were told by Pearson that

this (VPAT) is provided solely for their evaluation of Pearson's products and not to be distributed outside the organization.

The VPAT confirmed what we had been able to learn on our own: Pearson is trying to make more MML content accessible, but there are some definite limitations. As of this writing, Pearson clearly states [12],

MyMathLab supports access to selected single and multiple answer multiple-choice and free response questions with the JAWS 12 (only) screen reader.

On 13 December 2012 we, along with several other invested staff members at PCC, took part in a phone conference with a senior product manager and an

accessibility representative from Pearson. The conversation proved to be very encouraging. Besides getting the history of how Pearson ended up with so much Flash content, they explained what Pearson is doing to move away from a Flash-based delivery system in favor of a MathML with MathJax implementation. This move would have big implications in terms of the accessibility of their content. They do have a time-line to roll out a new version of MML that will make use of the non-flash-based content, but the soonest we expect to have it commercially available is 12 months from now.

However, the optimism created by the meeting is only applicable to newly-developed books at Pearson. The online content and question libraries for established textbooks, which they refer to as ‘legacy’ textbooks, will be retrofitted when possible.

For more specific recommendations about MML, please see the Specific Guidelines and Recommendation sections below.

VPAT A VPAT is a company’s own assessment of how their product conforms to U.S. Federal Accessibility Standards (defined by Section 508 [17, 18]).

If a school or department is looking into using any online resources, be sure to ask for the VPAT for the online resource. A VPAT gives a school or department a starting point from which to assess the accessibility of those resources. We also encourage the school or department to experiment with the product, as the VPAT tends to be an overall assessment of the product and might not list all the particular pieces of the product that the school or department will be using.

WEBWORK WeBWorK is an open-source homework system supported by the MAA and the NSF. Several PCC mathematics faculty members use WeBWorK for both their online and on-campus courses. Because WeBWorK is open-source and can be installed locally, it gives several advantages to other online systems:

- Each college can either use problems from the National Public Library or create its own libraries to fit the particular college’s needs.
- Using the most current version allows for the implementation of MathJax, which in turns makes all symbolic mathematical content accessible for JAWS.

MAURICE MINES About four weeks into our project, we were introduced to Maurice Mines and this was a definite turning point in the project. Maurice was referred to us by Kaela Parks because he has a strong educational and technical background, is blind, and Kaela knew the benefit our project would receive from including the experience of an end user. During the five sessions we had with Maurice, we were able to ‘test-drive’ all of the ideas and sample files that we believed to be accessible. Additionally, Maurice was able to give us both a personal and a historical perspective on our project.

It became so obvious to us (Chris and Scot) once we started working with Maurice that JAWS was not the solution to our entire project. Instead, the rule of four (discussed in detail on page 7) becomes much more important, especially in an accessibility context. Depending on the individual student, the accommodation needed might involve JAWS, printed Braille, a .brf file for an refreshable Braille device, printed tactile graphs, or something else entirely. Maurice was also able to explain the difference between Grade 1, Grade 2 [3], and Nemeth Braille [13] and then demonstrate the difference between them.

WINSLOW PARKER We were introduced to Winslow Parker, retired lead technical specialist for the

Oregon Commission for the Blind (OCB)[14], via Sonya Redmond, a mathematics faculty member at PCC. We had two incredibly enlightening discussions with Winslow (one phone, and one face-to-face). He told us that the OCB is a state agency that receives 90% of its funding from the federal government and 10% from the state. Its main focus is to help people who are blind receive job training, and to get them into the workforce.

VISUALLY IMPAIRED

There are clearly many different types of accommodations that we, as faculty, may be required to make. Some accommodations will certainly be easier to implement than others. Our research and meetings with Maurice and Winslow lead us to focus most of our efforts on trying to accommodate students who are blind or visually impaired; in what follows we will therefore mostly discuss accommodations from this perspective.

DISCLAIMER

The findings described in this report are based on many experiments, hours of reading, and meetings with experts on accommodations. At time of writing, we believe the results to be accurate; as software programs evolve over time the *specific* findings may become obsolete, but the *general* principles remain valid. Any inaccuracies are our own.

4 General best practices

RULE OF FOUR

The rule of four is one of the most useful guiding principles both in teaching, and from the perspective of accessibility. Explicitly, when a concept or idea is introduced and discussed, we try to do so in four different ways:

- algebraically
- numerically
- verbally
- graphically

Depending on the student that we are working with, and the particular accommodations that the student has, one or more of these different descriptions may be harder for the student to access than the others.

For example, if we are accommodating a student that is hearing impaired then the *verbal* description will need to be accommodated. This can be achieved in a number of different ways which include: using a sign language interpreter; captioning videos and other audio content. A student who is visually impaired may have more difficulty accessing each of the different descriptions except the *verbal*.

ONE SIZE?

Accommodations will vary from student to student, and there is not a ‘one-size-fits-all’ solution. This parallels the reality of working with a diverse student population, even those that do not need any accommodations; everyone is unique and needs to be treated as an individual. This is the main accommodation that we can make.

EQUALLY EFFECTIVE

The rule of four is a particularly useful reference because, as instructors, we always hope that we are able to use each one with an equivalent level of effectiveness. One of the most important principles of accessibility is to provide

an equally effective learning experience.

In particular, when constructing a lecture or an activity, we must be sure to create

equally effective activities that provide a similar level of learning outcomes.

Note that this applies to every single class, whether it contains a student that has an accommodation or not. In the current era of technology we are often tempted to have our lectures and activities explode with animations, slide-show presentations, interactive applets, etc. If the purpose behind the media has not been well-thought out, then it certainly risks being lost on the students.

One of the many advantages to creating content that is equally effective, is that we naturally reflect and think about how valuable the activity actually is: do we need this interactive applet? does this animation benefit the student, or is it more distracting? what do I want the student to get from this activity- is it just fun, or is there a learning objective behind it?

UNIVERSAL DESIGN

The Rule of Four is closely aligned with Universal Design [21] which is

a way to describe the concept of designing all products and the built environment to be aesthetic and usable to the greatest extent possible by everyone, regardless of their age, ability, or status in life.

ACCESSIBLE FORMATS

Throughout this report we describe ‘accessible’ formats. There are many different types of accessible formats; for example:

- for the JAWS screen reader, the most accessible format is a webpage that contains MathML, opened in IE 5.5-8 with Design Science’s MathPlayer plugin;
- for a refreshable Braille device, the most accessible format is a .brf file;
- for some students, a paper copy that simply has enlarged font may suffice.

The common link between these and all of the other accessible file format is that if Disability Services is to create one of these accessible formats, *the source file must be available to Disability Services if they are to be able to create these accessible formats for the student.* This means that we must always keep our .docx, .tex, .odt, etc files readily available and D2L users should have a single folder in which the sources files are saved in their course shell.

It should be noted that conforming to document formatting standards (such as using style-based headings, for example) will assist in producing accessible formats, but there may be unforeseeable issues that arise. Maintaining up-to-date versions of source files is still a vital best practice.

INACCESSIBLE FORMATS

Based on our experiments with the JAWS screen reader and on our conversations with Disability Services, we have generally found the following file types to be inaccessible.

- pdf files with Mathematical content;
- Flash applets;
- Java applets;

By saying these files types are inaccessible, we mean one or more of the following:

- JAWS cannot access any content of the file,
- Disability Services cannot make use of the any part of the the file,

- the mathematical content in the file is inaccessible to JAWS, or
- the mathematical content in the file is in a format that is unusable to Disability Services and it would need to be recreated manually in order for Disability Services to make an accommodation for a student.

IMPORTANT PCC ensures equal access through the complementary approaches of proactive alignment with best practices and good facilitation of the accommodation process.

We should design courses ‘from the ground up’ to be as universally accessible as possible. More explicitly, almost all of the content should be screen-reader accessible, keyboard navigable, and not rely exclusively on images and other media that are not accessible to all. We will then be well prepared to accommodate students without much additional effort.

Even if we have a class that does not have a student who requires accommodating, we should still not use inaccessible tools unless an equally effective and accessible tool is also available.

5 Specific best practices

CREATING CONTENT There are many different ways that we, as Math faculty, create our content. The way that we accommodate each student is, of course, going to vary from student to student. Some students may prefer to use .brf files on their refreshable Braille devices, others may prefer Braille-embossed paper, others will prefer a webpage that JAWS can read, and some may prefer any combination of all of the above.

One of the most important facts to remember is that we, as faculty, are not expected to perform the conversion to these (and other) outputs ourselves.

Disability Services will be happy to work with us in the production of such alternative media, but there are certain steps that we can take to assist them with the process.

As we discuss the following ideas and outlines, remember that every student has different needs. The following serves as a guide to best practices, but should not be thought of as golden rules that will always work.

STAND ALONE We will use the following legend to help us in discussing specific software, but it is important to note that we will not rely on this legend exclusively as that would not be universally accessible. Rather, we will use it as an equally effective visual aid:

- green check marks (✓) represent content formats that will stand alone from an accessibility perspective and are recommended;
- orange circles (●) represent content formats that *might* stand alone from an accessibility perspective and caution should be exercised;
- red crosses (✗) represent content formats that will *not* stand alone from an accessibility perspective and are *not* recommended.

A recommended item means that we may consider using the given format in a stand-alone, accessible fashion. An item that is not recommended means that we may use the given format, but there must be an equally effective tool or activity to accompany it- in this sense, the given format does *not* stand alone.

For example, during a discussion or demonstration, it is not appropriate for us to use a GeoGebra (Java-based) applet *exclusively*. However, it is acceptable to direct students to a GeoGebra applet provided that there are *equally effective alternative* tools that students may choose to use instead. In this sense, GeoGebra is not recommended as a stand-alone tool.

MSWORD • Recommended if used in conjunction with MathType. When using Microsoft Word to produce documents, the established protocols for structuring documents must be followed, including using headings, formatted lists, etc. Please see [15] for additional information on properly structuring documents. To include mathematical content, use Design Science’s MathType and not the native equation editor which uses Microsoft’s Office Math Mark-up Language (OMML).

MATHTYPE ✓ Recommended. MathType [11] is an equation editor created by Design Science that is compatible with many word processing and desktop publishing programs.

Disability Services is able to convert a Word document containing MathType mathematical content to an accessible electronic format. PCC has purchased campus licenses for each campus, so all staff members and students can install a copy of MathType on their home and work computers. This way there is no financial burden on any staff or student to use this option.

MICROSOFT OMML × Not Recommended. Microsoft has a native equation editor in the following versions of Word: Word 2007(PC), Word 2010(PC), and Word 2011(Mac).

Disability Services is not able to convert a Word document containing mathematical content created with Microsoft’s native equation editor to an accessible electronic format. Mathematical content created using Microsoft’s native equation editor will be converted into images without alt tags when DS tries to convert a document into an accessible electronic format. Since PCC has licenses for MathType, we recommend the use MathType instead of Microsoft’s native equation editor.

Sometimes MathType can convert equations made with Microsoft’s native equation editor into MathType equation, but this does not work perfectly all of the time. Using Microsoft’s native equation editor runs the risk of having to retype the mathematical content using MathType so that DS can use the file.

L^AT_EX ✓ Recommended. L^AT_EX is a mark-up language; converting L^AT_EX documents into an accessible format is usually straight forward as the accessible formats are also mark-up languages.

However, not all packages will work nicely with the conversion process. Given that there are over three thousand constantly-evolving packages on ctan (Comprehensive T_EX Archive Network), it would be very hard to give a comprehensive list of the packages that do and do not work. We have detailed some results in the appendix, but be prepared to make adjustments to documents as and when the need arises.

LIBREOFFICE ✓ Recommended. LibreOffice (with its native equation editor) converts easily to an accessible format. DS has less experience of working with LibreOffice and MathType. Additionally, LibreOffice and MathType are not currently compatible on the Mac OS.

SCIENTIFIC NOTEBOOK ✓ While we highly recommend Scientific Notebook from the perspective of creating accessible documents, we do not recommend that people change their currently

content creation process to use Scientific Notebook. A simple assessment is that Scientific Notebook is essentially a GUI-interface for a basic version of \TeX with a built-in computer algebra engine. While Scientific Notebook is useful from the accessibility standpoint, it is more limited than other available options from the creation standpoint as it cannot make use of many of the modern packages developed for \LaTeX .

POWERPOINT ✗ Not Recommended. While text-based Powerpoint files can sometimes be converted into an accessible electronic format, Powerpoint files containing mathematical content cannot. The mathematical content is converted into images without alt text. Currently there is no way to convert a Powerpoint file into a Word file that maintains the MathType objects in a usable format.

Powerpoint users must be prepared to somehow extract all the content (narrative and mathematical) and place it in a file format that DS can use.

GRAPHS ✓ Recommended. There are many ways to create graphs, including Winplot, Excel, Graph, pgfplots, PSTricks, etc. Any graph or graphic, regardless of how it was created, will always be read as an image by a screen reader. As such, appropriate alt text must be included for webpages.

Printing a tactile graph on embossed paper is a very time-intensive process. When working with a visually-impaired student it is possible that Disability Services may ask the instructor to choose which are the most important images, as not all images may be printed.

PREPARED NOTES ✓ Recommended. Many faculty use prepared, fill-in-the-blank style lecture notes that are completed during class, using Starboards and document cameras for example. As long as the faculty member can get the source file to Disability Services ahead of time, then Disability Services will be able to provide the student with a suitable accessible output format.

DESIRE2LEARN ✓ Recommended. Given the awards and recognition D2L has won in the area of accessibility, we recommend that as much of the online class as possible be conducted within D2L. All output from D2L's equation editor is stored as MathML, so it is screen reader-accessible. Students can also cut and paste from MathType into D2L's equation editor using either the \LaTeX or MathML interface by changing the Cut and Copy Preferences in MathType.

Quizzes, tests, or homework that will be assessed in D2L must have the mathematical content stored as MathML. One way to ensure this is to use D2L's equation editor to produce all mathematical content in these assessment tools.

MYMATHLAB ● Proceed with caution when dealing with MML and other publisher-based content. Our research has shown that MML offers some screen-reader accessible questions within each homework and quiz set. Typically this represents a small percentage of the total number of questions available.

There are other aspects of MML that are not screen-reader accessible; these include, but are not limited to, the 'Content' area where students can read an electronic version of the book.

Some instructors may assume that a student can simply be directed to the problems in the text book. However, this does not provide an equally effective learning experience as the student does not receive immediate feedback from algorithmically-generated problems.

WEBWORK ✓ Recommended. WeBWorK has a variety of screen-output options, including MathJax which can be configured (easily) to use MathML. We found that JAWS was able to read and navigate even the most complicated problems. In addition to this, WeBWorK is very close to being completely Section-508-accessible [22], and allows students to print problems to a pdf file so that time extensions can be easily accommodated.

WeBWorK is free, open source, and, assuming that we at PCC can maintain our own version of it, easily customizable.

VIDEOS ✓ Recommended. Of course, this is provided that the content of the video provides both audio *and* visual stimulus that convey the same content. If it does not provide both, then it does not stand alone. It should go without saying that videos need to be captioned.

TESTGEN ✗ Not recommended. TestGen does not export to an accessible format; unless instructors wish to re-generate the mathematical content (using MathType or \LaTeX), TestGen should be avoided.

OTHER MEDIA ✗ Not recommended. Instructors often use GeoGebra, Flash applets, animations, and other media to enhance the learning experience.

Here are some specific barriers:

- GeoGebra is Java-based, and (along with a lot of other Java applets) can not be read well by JAWS;
- Flash (in general) can not be read by JAWS; there are exceptions;
- slider-based activities are not accessible to those students that have mobility impairments, unless they are able to be controlled with a keyboard in addition to the mouse.

We are not proposing that an instructor remove these tools completely, but are instead highlighting that these types of media are not universally accessible and equally effective alternatives must also be available.

6 Recommendations

The following recommendations are intended as guidelines; bear in mind that there is no ‘one-size-fits-all’ solution, and that these recommendations may change as we all learn more.

(R_1) Instructors need to keep source files (.docx, .tex, .odt, etc) readily available.

Disability Services can convert most source files into the necessary format, be that .brf, .xht with MathML, enlarged print, etc. Without the source files, converting content to an accessible format is significantly harder and much more time intensive. For online classes, the source files should be saved to a single, consistent folder within every D2L course.

(R_2) Instructors can create graphs and images however they prefer, but all images need alt text.

Disability Services can print graphs and images on embossed or heat-sensitive (swell) paper.

- (R_3) Instructors should work ahead of schedule and try to be prepared at least a week in advance.

Having a week's lead time will allow Disability Services to convert content into an alternative form in time for the student to use the material in class.

- (R_4) Instructors should consider splitting longer tests into smaller parts for students that are allowed a time extension (such as a double time extension).

If a student has a time extension which would make taking the test unreasonable to complete in one sitting (imagine taking a six-hour exam in one session!), the instructor should split the test into multiple parts so that the student can take the different parts on different days.

- (R_5) Instructors relying upon online homework systems should consider migrating to WeBWorK.

WeBWorK is significantly more accessible than any publisher's *current* online homework system.

- (R_6) New course developments and takeover courses relying upon MyMathLab and MyStatLab will be put on hold.

This freeze will remain in effect until Pearson releases the updated, more accessible, non-Flash-based replacement and an appropriate book with accessible online, publisher-created content is adopted. (See the next recommendation.)

- (R_7) New textbook selections must consider accessible online content a top priority.

In order for SAC members to depend upon the online publisher-created content that accompanies an adopted textbook, nearly all of the online content must be accessible.

References

- [1] *Accessible Technology Initiative*. URL: <http://www.calstate.edu/accessibility/tutorials/math.shtml>.
- [2] *Active Perl*. URL: <http://www.activestate.com/activeperl>.
- [3] *Braille*. URL: <http://www.acb.org/tennessee/braille.html>.
- [4] *Design Science home page*. URL: <http://www.dessci.com/en/>.
- [5] *Desire2Learn*. URL: <http://www.desire2learn.com/news/2010/Desire2Learn-Attains-National-Federation-of-the-Blind-Nonvisual-Accessibility-Gold-Level-Web-Certification/>.
- [6] *JAWS screen reader*. URL: <http://www.freedomscientific.com/products/fs/jaws-product-page.asp>.
- [7] *Math Talk*. URL: <http://metroplexvoice.com/>.
- [8] *Math Trax*. URL: <http://prime.jsc.nasa.gov/mathtrax/homepage.htm#>.
- [9] *MathJax*. URL: <http://www.mathjax.org/>.
- [10] *MathPlayer*. URL: <http://www.dessci.com/en/products/mathplayer/>.

- [11] *MathType*. URL: <http://www.dessci.com/mathtype>.
- [12] *MyMathLab*. URL: <http://www.mymathlab.com/accessibility>.
- [13] *Nemeth Braille*. URL: http://en.wikipedia.org/wiki/Nemeth_Braille.
- [14] *Oregon Commission for the blind*. URL: <http://www.oregon.gov/Blind/Pages/index.aspx>.
- [15] *PCC accessibility webpage*. URL: <http://www.pcc.edu/access>.
- [16] *PCC Disability Services*. URL: <http://www.pcc.edu/resources/disability/>.
- [17] *Section 508 (primary)*. URL: <http://www.section508.gov/index.cfm?fuseAction=Laws>.
- [18] *Section 508 (secondary)*. URL: <http://www.access-board.gov/sec508/standards.htm>.
- [19] *The document foundation*. URL: <http://www.documentfoundation.org/>.
- [20] *The World of the Blind*. URL: <http://www.ams.org/notices/200210/comm-morin.pdf>.
- [21] *Universal Design*. URL: <http://enact.sonoma.edu/content.php?pid=218878&sid=1850089&preview=2992569949056263234954a15a575930>.
- [22] *WeBWorK*. URL: <http://webwork.maa.org/wiki/Accessibility#.UMT29Pkv8hY>.

A Converting content

We, as faculty, are not expected to convert content into alternative forms. However, we may find it useful to experiment before sending source files to Disability Services.

This appendix gives some details of the conversion process for the most common content types.

JAWS In order for JAWS to read mathematical content correctly (even something simple such as x^2 being read as ‘x squared’), we need to work with IE5.5 through 8, and the MathPlayer plugin from Design Science [10]. The faculty labs are at the following locations:

- SY library rm 208;
- RC library rm 206;
- CA library rm 121;
- SE MT Tabor rm 122.

Each lab has computers that have the demonstration version of JAWS installed. The SY lab has a dedicated computer that has IE8 installed- most of the other computers at PCC use IE9 or higher, and rolling the version back is problematic for TSS.

Regardless of the source file (.tex, .docx, etc), JAWS works best when dealing with .xht files that contain MathML.

MSWORD When converting from Microsoft Word to a webpage that uses MathML, you must first have MathType installed. In general, having the most current version of MathType will be helpful.

To convert from a Word document to a webpage that uses MathML, you'll need to use the "Publish to MathPage" (PMP) option from the MathType menu in Word. You can now use the following steps:

1. This option will open a new dialog box asking your for some information about the MathPage you want to create.
2. You can define a title for the webpage you're creating and set the file save location in the top of the dialog box.
3. In the "Equations" section, you want to make sure that the radio button next to "MathML" is selected.
4. You do *not* want to use images for the equations as they are not universally accessible. Under the MathML options, make sure you have it set to "XHTML+MathML".
5. You want to leave the Target Browser set to "All browsers".

Table 1 shows which versions of Microsoft Word and MathType can correctly convert from a Word document to a webpage that uses MathML.

Table 1: MSWord and MathType versions

MS Word Version	MathType v6.7a	MathType 6.7e	MathType 6.8
2007 (PC)	works		works
2008 (Mac)		does <i>NOT</i> work	
2010 (PC)	does <i>NOT</i> work		works
2011 (Mac)		works	

Here is an equally effective verbal description of Table 1:

- PMP in Word 2007 works with MathType v6.7a and MathType v6.8.
- PMP in Word 2008 does not work. This is due to the fact that there is no MathType menu in Word 2008.
- PMP in Word 2010 only works with MathType v6.8.
- PMP in Word 2011 only works with MathType v6.7e. v6.7e is a free update to v6.7d. MathType v6.7e fixes the PMP bugs that existed in v6.7d.

TEX4HT tex4ht stands for 'T_EX for hyper text' and is a bundle that provides a robust set of tools for converting .tex files into .html, .xht, or other hyper-text related files.

Windows users will need to install a Perl distribution, for example Active Perl [2]. From the DOS command prompt you need to run the following two commands

```
mk4ht mzl原因 myfile.tex 'html, mathplayer'
tex4ht mzl原因 myfile.tex
```

Most MAC and Linux operating systems have Perl installed by default, and simply need to run

```
mk4ht mzl原因 ./myfile "html,mathplayer"
```

from the command line.

There are many different document classes available for \LaTeX including the basic article, report, and book, together with the slightly more sophisticated beamer, exam, standalone, memoir, Koma bundle, and more. Each document class has its own idiosyncrasies, and may make its own (re)definitions that perturb the conversion process. For example, to convert a beamer document to a .xht file put the document in ‘handout’ mode

```
\documentclass[handout]{beamer}
```

and then run mzl原因 as described above.

There are other commands provided by the tex4ht bundle, but they are less useful from an accessibility perspective.

LIBRE VS OPEN LibreOffice and OpenOffice are both open-source office suites that use .odt file extensions. A very brief history follows.

In the beginning (late 1990s) there was StarOffice. In 1999, Sun Microsystems took over and renamed it OpenOffice. In 2010 Oracle bought Sun Microsystems, and put OpenOffice into the ‘Apache Incubator’ in 2011. In 2010 LibreOffice was founded by The Document Foundation [19] and has been continually developed by the community since.

MATHJAX A sample .html file that uses MathJax (line 6 is the most important).

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http
  ://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="
  en">
3
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf
  -8"/>
6 <script type="text/javascript" src="http://cdn.mathjax.org/
  mathjax/latest/MathJax.js?config=TeX-AMS-MMLHTMLorMML"></
  script>
7 </head>
8
9 <body>
10 When  $(a \neq 0)$ , there are two solutions to  $(ax^2 + bx + c =$ 
11  $0)$ 
12 and they are  $[x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}]$ 
13
14 </body>
</html>
```

MATHML Some sample MathML code; note that MathML is not intended to be written from source, but rather as a final output via conversion from PMP (MathType) or tex4ht (\LaTeX) for example. Some typical output is shown below.

```
1 <body>
2 The quadratic formula uses <math xmlns="http://www.w3.org/1998/
  Math/MathML">
```



```

3 <mi>x</mi>
4 <mo>=</mo>
5 <mfrac>
6 <mrow>
7 <mo>-</mo>
8 <mi>b</mi>
9 <mo>#177;</mo>
10 <msqrt>
11 <msup>
12 <mi>b</mi>
13 <mi>2</mi>
14 </msup>
15 <mo>×</mo>
16 <mi>4</mi>
17 <mi>a</mi>
18 <mi>c</mi>
19 </msqrt>
20 </mrow>
21 <mrow>
22 <mi>2</mi>
23 <mi>a</mi>
24 </mrow>
25 </mfrac>
26 </math>
27 </body>

```

GRADE 1 AND 2 BRAILLE

There is a significant difference between Grade 1 and Grade 2 Braille, which we learnt while working with Maurice. Grade 1 Braille spells out each letter in each word; for example, ‘the’ would have the three letters ‘t’, ‘h’, and ‘e’. Grade 2 Braille has short hand notation for the most common types of letter combinations; there are special symbol for ‘the’, ‘for’, and many others. Maurice is fluent in Grade 2 Braille, so found it significantly slower to read Grade 1 Braille.

B Anticipated questions

Each of these questions assumes that you have a student who requires an accommodation in your class.

(Q₁) Can I require the use of MML if I have an alternative for students for whom the publisher’s material is inaccessible?

Provided that the alternative is equally effective, then yes. In this instance, the only equally effective tool would be another (accessible) online homework management system (such as WeBWorK).

Note that it is *not* equally effective to simply direct the student to homework problems from the text book.

(Q₂) Can I still direct students to MML if they desire to use it, as long as I don’t require it?

Yes, but you must also provide one or more equally effective resources for students who will not be able to use MML.

(Q₃) That last question actually raises another question: Do only the required components of the class have to be accessible?

Yes. This is the ‘Stand alone’ concept (see page 9). If a component is required, then it must stand alone as universally accessible.

- (Q₄) If I find a resource that is awesome, enlightening, but inaccessible to some, can I still say, “Go and look at this if you want?” even if there is no equivalent accessible alternative?

Yes, provided you don’t *require* every student to view/use/interact/whatever it, and provided that it is not the *only* way that you are communicating this topic/concept/idea.

- (Q₅) Can I still post pdf files?

Yes, provided that you have the source files so that, if necessary, Disability Services would be able to convert the source file into the appropriate accessible format for a student.

- (Q₆) Should I stop using GeoGebra, animations, and other formats that are not universally accessible?

No, but it would be best not to use them *exclusively*. It is perfectly reasonable to direct students to these resources as optional material to enhance the learning process, but *requiring* the use of such materials that can not be accessed by all is not acceptable.

Equally effective alternatives should be supplied wherever possible; in the case of an animation (for example), a verbal description may not seem adequate, but if described in detail with appropriately detailed follow-up questions and comments, it may be as effective.

Students who require accommodations certainly do not wish to ‘punish’ other students by not allowing them access to tools.

- (Q₇) Do I need to enter in alt text for all of my MathType equations in MS Word? I ran the accessibility checker in Word and it said I needed the alt text.

No. The issue here is that Word’s accessibility check only sees the MathType equations as images. Word does not realize the equations are more than just images and that Disability Services can convert MathType equations into an accessible format.