CASE STUDY

Towards Making an Online Access Maths Course Accessible to All

Laura Maria Forero Rincon, Centre for Academic Development, University of Aberdeen, Aberdeen, Scotland. Email: laura.forerorincon@abdn.ac.uk
Brenda Harden, School of Education, University of Aberdeen, Aberdeen, Scotland. Email: brenda.harden1@abdn.ac.uk
Maija Koukkari, Centre for Academic Development, University of Aberdeen, Aberdeen, Scotland. Email: maija.koukkari@abdn.ac.uk
Maria Nikoletatou, Centre for Academic Development, University of Aberdeen, Aberdeen, Scotland. Email: maria.nikoletatou@abdn.ac.uk
Morgiane Richard, Centre for Academic Development, University of Aberdeen, Aberdeen, Scotland. Email: m.richard@abdn.ac.uk
Craig Stewart, Centre for Academic Development, University of Aberdeen, Aberdeen, Scotland. Email: craig.stewart@abdn.ac.uk

Abstract

The authors' institution developed pre-COVID a suite of three access online maths courses, at SCQF levels 5, 6 and 7 (corresponding to the last 3 years of Scottish secondary school education). These courses were designed to be 'do-it-yourselves' and are entirely online. They have grown in popularity and have attracted a wide range of applicants from around the country, and beyond, as a way into higher education. The material was written before the maths education and support community started systematically investigating the issues of making mathematics digitally accessible, and therefore, did not implement any of the good practice that experts now promote. Guidance has been produced since, however, and as the institution received two expressions of interest from prospective students with visual impairment in the past two years, it was time to make the access courses accessible. This involved work with Word and LaTeX material, diagrams, figures, as well as hand-written material. It also required thinking carefully about approaches to teaching concepts of foundation mathematics to be understandable by a visual-impaired audience. In this paper, we describe the changes we implemented and formulate advice for the design of accessible foundation maths.

Keywords: digital accessibility, online education, mathematical sciences, inclusion and diversity, access to education.

1. Introduction

1.1 The Maths Access Programme

The School of Education at the University of Aberdeen were aware of adults interested in applying for a teaching career or a change of career but who lacked the necessary minimum mathematics qualifications. As individuals could be in employment, an online course was developed to allow for flexibility of study. In 2018/2019 a team developed the first Mathematics Online Access course, at Scottish qualification level SCQF 5 (equivalent to English level RQF 2) (Scottish Credit and Qualification Framework, 2026). Once enrolled, students have access to the course for one year and can work through the material and exercises at their own pace. When requested, support is offered to groups or individuals. Following on from the success of this course, an online access course at the following level was developed (SCQF 6 or RQF 3), subsequently followed by a SCQF 7/RQF 4 course. The three online access courses are equivalent to the last three years of the Scottish school education system.

Our awareness for solutions to improve the accessibility of STEM content had increased and since we had had two queries from students with visual impairments, it was time to begin to adapt these courses to ensure their accessibility. We started working on the SCQF 6 course as the skills developed there would help updating the other courses.

The SCQF 6 Mathematics Online Access Course covers: Straight Line, Vectors, Trigonometry, Polynomials, Differentiation, and Integration.

1.2 Quick Tour of Maths Accessibility Research

Much work has been done in recent years to improve the accessibility of teaching methods and resources in Higher Education (CAST, 2024). Most of the current guidance, however, is not suitable for STEM teaching (Cooper, 2012). STEM disciplines rely heavily on mathematical texts, and figures and diagrams which bear pedagogic meaning, rather than being decorative. A specialised framework is therefore necessary. The JISC Accessible Maths Working Group (JISC A11y-Maths group, 2024) has done extensive curative work to collate current results in the domain and has built a community of experts who have developed guidance and resources.

Maths text is usually written either in the typesetting system LaTeX, or in Word, using a maths equation editor. Whilst LaTeX is recognised as best to write advanced mathematical text, it generates PDF documents which are unfortunately not fully accessible: in particular, equations are not compatible with screen reader software (University of Bath, 2026; University of York, 2024). The current advice is to generate documents in HTML format, using the JavaScript MathJax to render the maths, as this will enable compatibility of maths text with screen readers (University of York, 2024; University of Bath, 2026; Cooper, 2012). Both PDF and HTML formats should be made available to students. This can be achieved in various ways. LaTeX files can be converted to HTML using specialist software. Alternative markup languages, such as Quarto (Quarto, 2025), can otherwise be used to simultaneously generate HTML and PDF outputs.

Equation editors in Word, although less performant for writing advanced maths, generate mathematics that is accessible and compatible with screen reader software (Cliffe, 2022; University of York, 2024). However, as shown by Cliffe (2022) and our own results, editors must be used with care to ensure that the maths is indeed accessible.

Figures and diagrams in STEM are often used to illustrate a concept and are seldom simply decorative. The University of Bath has published guidance on producing accessible figures and diagrams, which includes advice on the use of fonts, colours, and legends (University of Bath, 2025). The online dynamic graphing tool, Desmos (Desmos Studio PBC, 2026), creates accessible graphs and geometric figures. GeoGebra (GeoGebra GmbH, 2026) creates less accessible figures but is a more advanced tool. Both tools will generate a permanent URL link to the figure as well as allow for exporting the figure as a PNG file. The recommendation is to use one of these tools to create figures whenever possible, and include the URL, as users can modify the formatting as they need (Cliffe, 2023). Including an alternative description, or a long descriptive caption is good practice: guidance on alternative text for maths graphs is available from DIAGRAM Center (DIAGRAM Center, 2015).

In this paper, we describe the changes we implemented to the SCQF 6 Mathematics Online access course: format and layout of documents (Word, LaTeX and handwritten), figures and diagrams, and amendments to some of the pedagogic approaches. In the following section, we first provide details of the screen reader we have used and explain how we used it to test our new material.

2. Using NVDA Screen Reader

SCQF level 6 access maths course consist of 10 units. The learning materials in each unit included videos, written notes and examples, exercises and solutions. Most of the content was originally written in Microsoft Word using Wiris MathType 7 equation editor and exported as PDFs before uploading to the course area in the Virtual Learning Environment (VLE) Blackboard. Some documents had been written in LaTeX and the exported PDFs uploaded to the VLE. One of our aims was to make the course material accessible to screen reader users. We used NVDA, which is a free, open-source screen reader for the blind and vision impaired available for Windows (NV Access, 2025), for our tests. NVDA is the second most widely used screen reader after the software JAWS developed by the company Freedom Scientific (WebAIM, 2024).

We first ran tests on the original Word documents, which allowed us to identify the NVDA settings required for reading maths written with MathType 7. First, an NVDA add-on is required. Two add-ons are available: MathCAT and Access8Math. We tested both using their default settings and found that the two add-ons can generate different audio outputs of the same maths, as shown in section 3. We also noted that although users can download both add-ons, only one can be activated at a time in NVDA Settings – Math Providers. Second, in NVDA Settings – Advanced, we found that the option “Use UI Automation to access Microsoft Word document controls” must be set to “Only when necessary” for NVDA to read maths written with MathType 7. Third, NVDA users must open one of the equations in the MathType 7 equation editor the first time they read a Word document with NVDA. After this initial step, NDVA will recognise and read maths without requiring the editor to be opened again. Finally, users need an active licence for MathType from Wiris for NVDA to read the maths, as well as to write it.

To overcome the issue of the necessity for a licence to read maths written in MathType 7, we considered using the MathType option to publish Word documents as MathPages. This option allows for Word documents to be exported as HTML, using MathJax to render maths. However, we discovered that not all equations would render with MathJax if the original Word document contained images. Wiris Technical Support confirmed that this was a bug and advised us to remove images from documents. Since many of our documents have images, we decided not to pursue this option further. We also considered using the Wiris MathType Add-in for Microsoft 365. The MathType Add-in converts all maths equations to images and generates alternative text, so screen readers will read the alternative text instead of recognising the equations as maths content. The alternative text generated, however, contained mathml code, which was read verbatim by the screen reader and other text-to-speech tools (Figure). It meant as well that the maths text lost other accessibility features, such as enlargement. Therefore, we did not retain MathType Add-in as a viable accessible option and continued to work with MathType 7. Since our institution does not have an institutional Wiris licence, we checked that funding was available to cover the MathType licence cost for any students who enrolled on the course and required the licence to read the maths.

LaTeX documents were converted to HTML, with MathJax to render maths, using the software LaTeXML (NIST, 2024), and both the PDF and the HTML files were uploaded on the VLE. We also assessed the HTML document with NVDA, which worked well provided one of the maths add-ons was installed.

Since we maintained the original formats of documents, our work focused on improving the accessibility of these formats and devising accessible layout and content. We discuss these in the next section.

Alternative text generated by MathType add-in

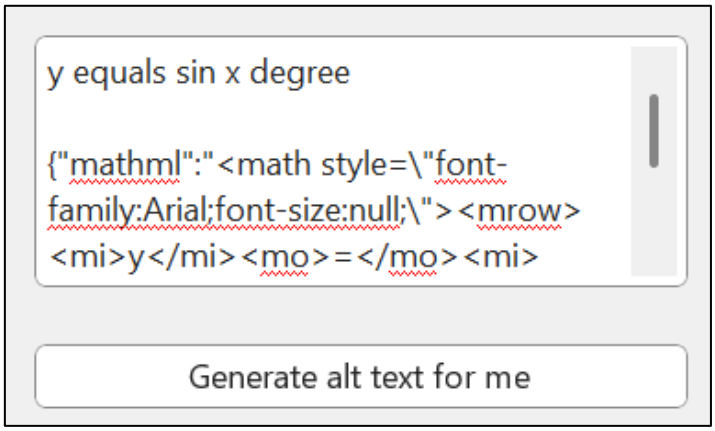


Figure 1: Alternative text pane from a Word document generated by MathType add-in for the maths equation $y = \sin x^{\circ}$.

3. Accessible Digital Maths

3.1 Layout of math content

We audited the accessibility of the original course using the SCULPT framework (Wilson, 2020), Anthology Ally and the built-in accessibility checker of Microsoft, and created a new template for the course. The new template included a new heading structure for navigation within a file using Microsoft Word styles and a new file naming convention for navigation within the course on the VLE.

The template also devised a new heading structure and numbering system for exercises and their subsections, to replace the original lists-based numbering, to ease navigation in exercise and solution sheets. This way, the template clearly differentiated between the various heading levels, including exercises and their answers.

Finally, the template included a new layout for exercises and examples. The original design presented some maths examples and solutions with a combination of horizontal and vertical displays. This affected the reading of maths by the screen reader, as content was recognised as tabular content, and was read row by row. For example, two solutions of a mathematical equation would be commonly displayed in two adjacent columns. The screen reader, however, reads content line by line (Figure). In the new material, we took care to write mathematical steps either exclusively horizontally or vertically. In addition, multistep examples and solutions that included text explanations were embedded in a single MathType object. This improved the flow of reading, compared to using multiple Mathtype objects interspaced with Word text.

3.2 Maths within MathType editor

We aimed to write the maths so that it would be read in a consistent manner throughout documents. We were particularly concerned with this issue, after we discovered that NVDA read maths differently for content written with different maths editors (i.e. Microsoft equation editor vs MathType), if formulas had been added as a figure, and even depending on the maths add-on used with NVDA (MathCAT vs Access8Math). Some of these challenges could be overcome by devising a custom stylesheet in MathType which we applied throughout the course content. This ensured automatic use of formatting styles and improved consistency of maths writing and reading.

However, the use of the stylesheet did not address all issues. We noticed that symbols within MathType must be chosen with care, as they represent specific mathematics operators, and may be read differently in NVDA, even if they may look the same. For instance, at the time of testing in summer 2025, x°, written using the symbol for composition, was read as such, while $x°$ uses the correct degree symbol. Other examples include the decimal point and dot product symbols. All maths had to be scrutinised and manually edited, as this issue had not been identified at the time of creating the course.

Writing multi-steps mathematical work

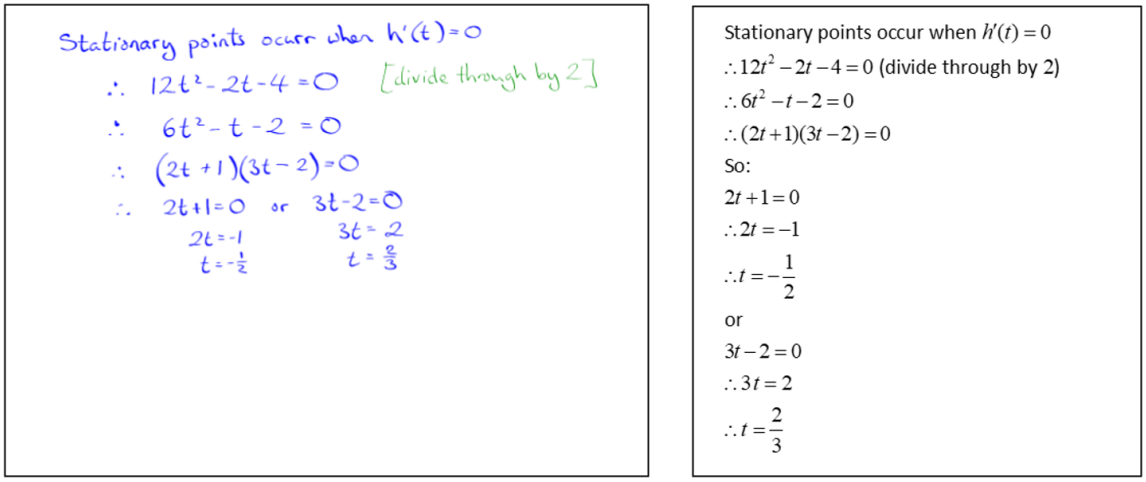


Figure 2: (left) Maths steps are often written as a mixture of horizontal and vertical text, such as the two solutions of an equation. However, the screen reader reads left to write, regardless of reasoning logic (right). Maths steps were re-written systematically vertically. The entire solution is written as a single Mathtype object, not three different objects, for the flow of reading.

Furthermore, some letters can also have different meanings, and may be read differently, depending on MathType style used, as well as the NVDA maths add-on installed. For example, the single, small letter vector notation in STEM could be **u**, $\underline{u}$ or $\vec{u}$. The course team preferred using the notation $\underline{u}$. Unfortunately, $\underline{u}$ was read as 'u below' by NVDA, with both maths add-ons, using their default settings at the time of testing, in summer 2025. The notation **u** was read as 'u' with Access8Math, but as 'atomic mass unit' with MathCAT, and the notation $\vec{u}$ was read as 'vector u' by both add-ons. Given that software and add-on versions and settings can affect the audio output, screen reader users may require support to determine the best settings.

3.3 Maths in LaTeX Documents

Three sections contained documents written in LaTeX: Straight Line, Vectors and Polynomials.

Prior work at the authors' institution had identified LaTeXML (NIST, 2024) as the conversion software that produced most satisfactory outputs (Price et al., 2021). Institutional guidance on using LaTeXML to convert tex documents had been produced and was available.

These documents were short (19 and 11 pages) and given that the maths content was school-level maths, we used LaTeXML with success to convert the tex files to HTML. LaTeXML converts tex files to xml, and converts the xml file to html. We used the following LaTeXML commands:

latexml latex-source-file.tex --dest=intermediate-xml-file.xml

latexmlpost intermediate-xml-file.xml --dest= accesiblehtml-file.html --split --splitat=section --navigationtoc=context --javascript="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.9/MathJax.js?config=TeX-MML-AM_CHTML" --urlstyle=file

The options used in the latexmlpost command generates an html document per section (--split --splitat=section) and a navigable table of contents (--navigationtoc=context). Note that MathJax was loaded using the cdnjs source, as cdn.mathjax.org (used previously) has been retired.

The figures in the original files had all been created using the tikZ LaTeX package. This was the only element that was not converted well by LaTeXML (also noted by Price et al., 2021). All figures, however, were recreated in Desmos, with the PNG export added as a figure to the .tex file, together with a link to the Desmos file in the caption.

3.4 Handwritten Maths

In some cases, maths content was presented as scanned handwritten notes, added as images to Word documents. This was poor from an accessibility standpoint as screen readers could not read the content of the pages. Additionally, the use of handwriting could sometimes be ambiguous, potentially causing confusion between, for example, a multiplication symbol and an x used to denote a variable. The handwritten material was initially manually re-typed in Word with MathType to improve its accessibility. This process was slow and meant that errors could appear due to the ambiguity of some written symbols, particularly when those working on the conversion were not subject experts.

To convert the handwritten content more efficiently while reducing the chance of errors, we experimented with using AI assistants. The authors' institution has a Microsoft licence, so Copilot was initially used. The image file containing the scanned handwritten equations was uploaded and Copilot was asked to generate the content of the images as an equation compatible with MathType. Copilot was unable to carry this out. The prompt was expanded to request that the equations be converted to LaTeX and MathML, but at the time of testing in summer 2025, Copilot was not able to produce a satisfactory output. On discussion with the rest of the team and establishing that there were no risks in using other AI assistants, we used Google's Gemini AI assistant instead. Gemini was able to convert the images quickly and accurately into LaTeX which could then be copied and pasted into the MathType editor (Figure).

While this saved a lot of time, some editing and checks were still necessary. Gemini returned the scanned documents as a continuous string of LaTex code. This was copied and pasted into MathType and then edited for appropriate display. This also allowed for an additional level of sense-checking of the AI assistant's output.

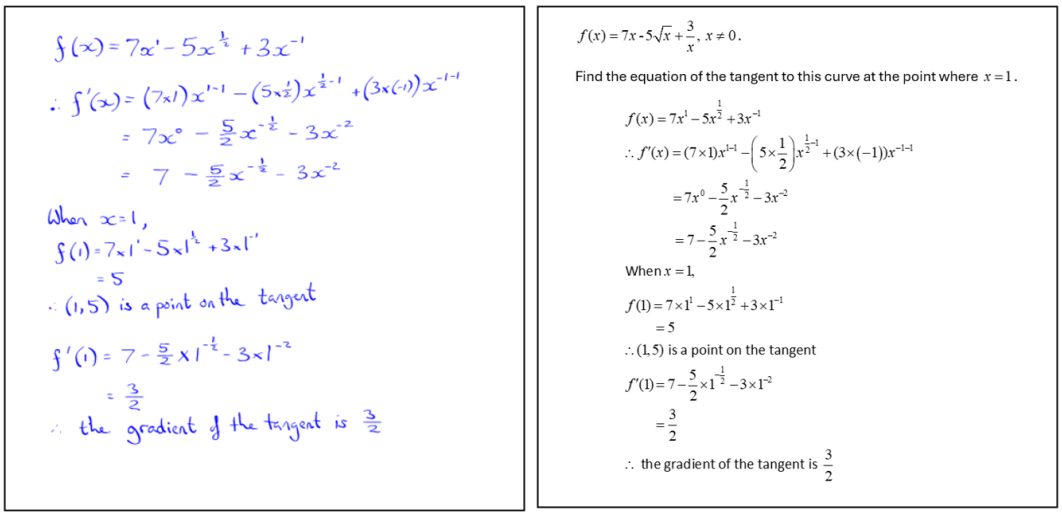


Figure 3: Example of conversion of (left) a handwritten solution to (right) a typed solution, using the Google Gemini AI tool. The image of the handwritten maths was uploaded to Google Gemini, which translated it in LaTeX code. The LaTeX code was copied in the MathType editor in a Word document.

4. Accessible Digital Graphs and Diagrams

We used the Graphing and Geometry tools of Desmos Studio (Desmos Studio PBC, 2026) to create graphs and diagrams, as Desmos generates graphs that are screen reader accessible (Desmos Studio PBC, 2026; University of Bath, 2025). Figures were exported as PNG and a link to the permanent URL was included in the figure captions, in both Word and LaTeX documents (Cliffe, 2023).

4.1 Desmos Geometry Tool

We used the Desmos Geometry tool for figures in the Straight Line and Vectors units and part of the Trigonometry units.

We used a range of objects available in the Geometry library to create the figures. However, since it is not possible to add a label to all types of objects, it was sometimes necessary to use several library functions to create a single object in the way the pedagogic team wished it to appear. For instance, a vector cannot currently be labelled in the Desmos Geometry tool, therefore, to add a label, we either superposed a segment line to a vector object, or added an invisible point object to a vector, as segments and points can be labelled (Figure).

Because of this, the left-hand side expression list sometimes contained several objects that were irrelevant for students and could become a distraction. To avoid this, we used the folder functionality to hide these objects from immediate view. We also used the folder in exercise sheets when the expression was the answer to the exercise to hide the answers from immediate view.

Generating labels in Desmos Geometry

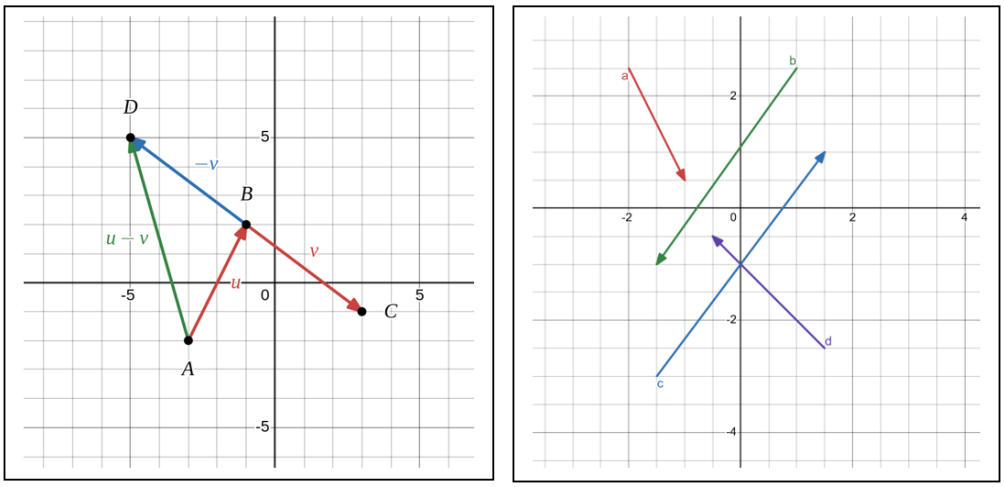


Figure 4: Labelling vectors in Desmos Geometry. To label vectors, either (left) segment lines were superposed to vectors, or (right) invisible points were added to vectors.
See the figures in Desmos: (figure 4 left) and (figure 4 right)

The Geometry tool has a screen reader label option for all objects. We used this option to name and/or describe objects in the figure. For instance, we indicated in the screen reader label when a segment line was a median or an altitude, or when a polygon was a particular figure such as a triangle or rectangle (Figure).

Using screen reader labels in Desmos

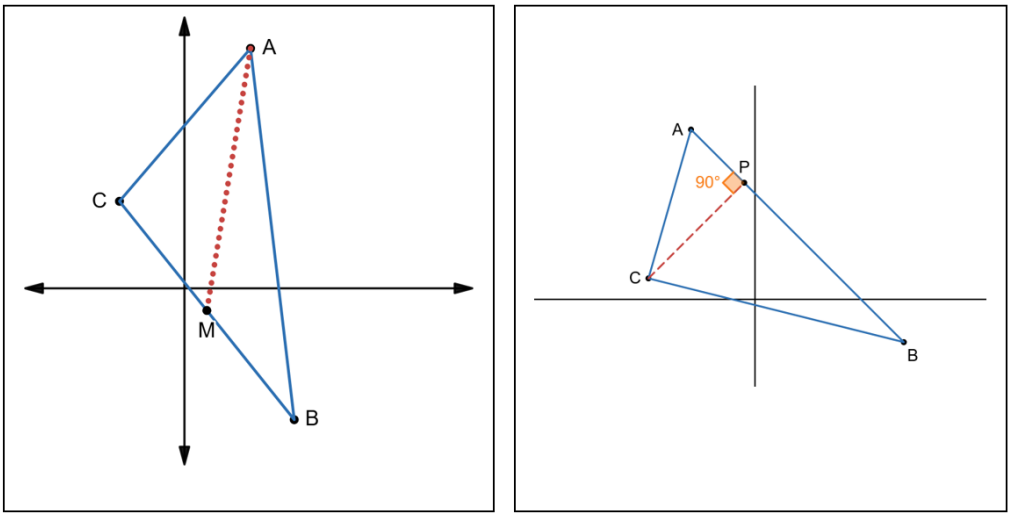


Figure 5: Use of screen reader labels in Desmos Geometry. The screen reader label was used to indicate that the polygon ABC was a triangle (both left and right) and that (left) the segment AM was a median and (right) the segment CP was an altitude.
See the figures in Desmos (figure 5 left) and (figure 5 right)

4.2 Desmos Graphing Tool

The Desmos Graphing tool was used for function graphs in the Trigonometry, Differentiation and Integration units.

In the original course material, figure annotations were included in the Word documents as text boxes overlaying the figure file. In the updated Word documents, we included all figure annotations within the Desmos figures. To annotate graphs, we either created labels by creating an invisible point, following Desmos guidance (Desmos Studio PBC, 2025), or by using the note option on the Graphing tool (Figure). We ensured that graphs could be identified through line type as well as colour (University of Bath, 2025).

Annotating graphs in Desmos

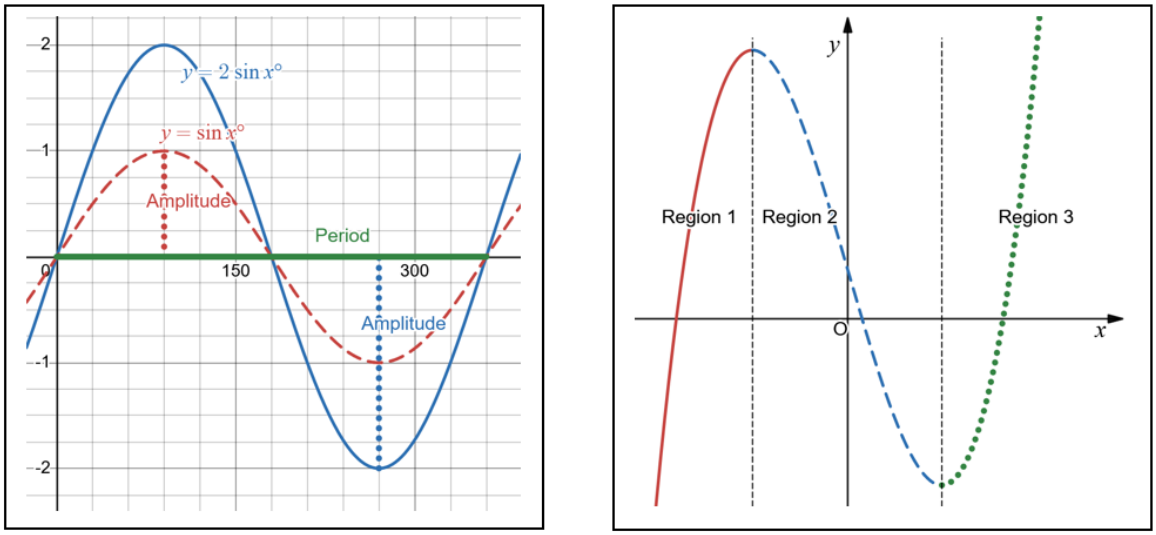


Figure 6: Annotating graphs. We annotated graphs in Desmos, either (left) by using labels of invisible points or (right) by using the note option in Desmos when the annotation was longer. Here the note was used to describe Regions 1, 2 and 3. The note text was also included in the figure caption to ensure that readers who did not access the Desmos link could also get the description.
See the figures in Desmos (figure 6 left) and (figure 6 right).

5. Changing Pedagogy

In some cases, improving the accessibility of the content meant that we had to consider alternative pedagogical strategies, as the updates made using tools such as the Word Template, MathType or LaTeX were not enough to make content accessible. In this section, we describe the three topics that required us to rethink how they were taught.

5.1 Trigonometry: CAST Diagram

The CAST diagram (see Figure) is a common method used in solving trigonometric equations. While this can be highly effective for visual learners, it is not useful for learners who are visually impaired.

CAST diagram

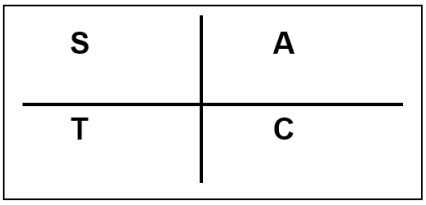


Figure 7: CAST diagram for solving trigonometric equations.

We continued to use the CAST diagram for learners for whom this would be useful, but we added the following detailed alternative description whenever we used it:

In quadrant 1 $\left(0^\circ \text{ to } 90^\circ\right)$, $y = \sin x^\circ$, $y = \cos x^\circ$, and $y = \tan x^\circ$ are all positive.

In quadrant 2 $\left(90^\circ \text{ to } 180^\circ\right)$, $y = \sin x^\circ$ is positive, and $y = \cos x^\circ$ and $y = \tan x^\circ$ are negative.

In quadrant 3 $\left(180^\circ \text{ to } 270^\circ\right)$, $y = \tan x^\circ$ is positive, and $y = \sin x^\circ$ and $y = \cos x^\circ$ are negative.

In quadrant 4 $\left(270^\circ \text{ to } 360^\circ\right)$, $y = \cos x^\circ$ is positive, and $y = \sin x^\circ$ and $y = \tan x^\circ$ are negative.

This provided students with both pictorial and descriptive explanations of the CAST diagram.

5.2 Polynomials: Synthetic Division

When solving polynomials, synthetic division is often used (see Figure).

Polynomial division

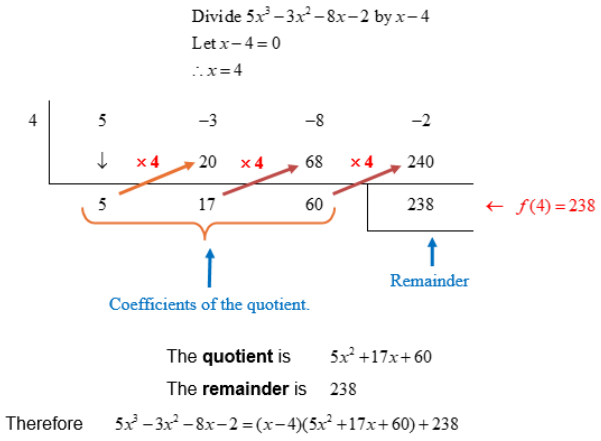


Figure 8: Example of polynomial division, as taught in the online maths access course. The information flow is both vertical and horizontal, and text boxes overlay the table.

This presents information that flows both vertically and horizontally. It also uses elements such as arrows and curly brackets in text boxes overlaying the table and therefore cannot be picked up by a screen reader. In discussion with Sight Scotland and the Royal Blind School (Sight Scotland, 2025), we devised an alternative way of teaching synthetic division in table form.

Polynomial division: accessible alternative

Accessible version of remainder theorem

Divide $5x^3-3x^2-8x-2$ by $x-4$ $\therefore$ Let $x-4=0$ $\therefore$ $x=4$

term from highest to lowest powers	coefficient	when $x=4$	answer	Quotient term
$5x^3$	5	copy coefficient of highest term	$=5$	$5x^2$
$-3x^2$	-3	$5\times4+(-3)$	$=17$	$17x$
$-8x$	-8	$17\times4+(-8)$	$=60$	60
-2	-2	$60\times4+(-2)$	$=238$	Remainder 238

The **quotient** is $5x^2+17x+60$

The **remainder** is 238

Therefore $5x^3-3x^2-8x-2=(x-4)(5x^2+17x+60)+238$

Figure 9: Alternative accessible presentation of the polynomial division. No overlaying text box is used, and the information flows in rows.

The information is presented in a table and is organised in agreement with the screen reader's direction of reading (by rows).

5.3 Calculus: Curve Sketching and Nature Table

In calculus the nature of stationary points can be determined with a nature table (see Figure).

Using the nature table to determine stationary points

The curve $y=\frac{x^3}{3}+\frac{5x^2}{2}-6x+5$ has a stationary point when $x=-6$. Determine the nature of this stationary point.

x	$\rightarrow$	-6	$\rightarrow$
$\frac{dy}{dx}$	positive	0	negative
Shape	/	—	\

Therefore, there is a **maximum stationary point** at $(-6, 59)$

Figure 10: Example using the nature table.

The nature table is very pictural and can be particularly useful for visual learners, but inaccessible to visually impaired learners. The order in which the information would be read in such a table is also, again, problematic. Therefore, for every traditional nature table, we added an accessible version of the table, which reversed the way in which the information was laid out, and did not make use of pictural symbols (see Table 1).

Table 1: Accessible nature table

x	$\frac{dy}{dx}$	Shape
$x < -6$	positive	increasing
$x = -6$	0	stationary
$x > -6$	negative	decreasing

6. Conclusion

In this paper, we describe the work we did to improve the accessibility of an online access maths course. The maths content is at foundation level, and many of the topics taught are covered in university foundation maths courses. Therefore, our results are also relevant to HE practitioners. The course originally consisted of PDF documents generated from Word or LaTeX. Word PDFs have all been replaced by accessible Word documents, and LaTeX PDFs have all been uploaded together with alternative HTML documents.

Extensive work has been done in recent years on digital solutions for mathematical and STEM content. Our own work greatly benefited from this research: we used current guidance on how to ensure maths is readable by screen readers and how to design figures and diagrams. We were also able to provide an accessible alternative to LaTeX documents quickly and efficiently.

We have also uncovered some issues that STEM teachers must pay attention to. Using an equation editor in Word documents does not necessarily make equations accessible. Symbols must be used with care for the maths to be read accurately. Additionally, thought must be given to the layout of exercises, and maths within exercises, to ensure that documents, either Word or LaTeX, can be navigated easily, and maths read in a sensible manner.

All tests were carried out on Windows operating systems, as our institution has a Microsoft licence. We noted that neither NVDA nor JAWS, another popular screen reader software, are available for Mac. To our knowledge, Mac VoiceOver does not have add-ons for handling maths content. We suggest, therefore, that more investigation is done for Mac operating systems.

Making STEM content accessible is not straightforward and requires expert knowledge and skills. Nonetheless, it is essential if we are to ensure that STEM courses are accessible to all. We hope that, with this work, we have contributed to this young, dynamic and important area of research.

7. Appendix: Project Workflow

The original course consisted of 174 PDF files, generated either from Word documents or from LaTeX documents, 15 of which were scans of hand-written material. Documents were short: the Word-PDFs were less than 10 pages, and the LaTeX-PDFs less than 20 pages.

The reviewed course consists of:

- 123 accessible Word documents using the new template, devised as described in section 3, and MathType 7 for writing all maths.
- 12 documents in both LaTeX-PDF and HTML with MathJax to render maths text.

The course also contained 104 images which were all re-designed in Desmos, with all annotations included in the Desmos file, as described in section 4.

The project team consisted of 2 eLearning advisors, 3 students employed as eLearning support assistants, 1 eLearning lead advisor, 1 University maths support advisor, and the course coordinator. The eLearning advisors and eLearning support assistants worked on Word documents, Desmos figures and hand-written documents. The eLearning lead advisor managed the timeline of the project, general accessibility issues, and worked on the NVDA settings. The maths support advisor worked on LaTeX documents and managed mathematics and STEM specific accessibility issues. The course coordinator worked on pedagogy and approved the final course documents.

8. References

CAST, 2024. *CAST Universal Design for Learning Guidelines version 3.0.* Available at: https://udlguidelines.cast.org/ [Accessed 8 January 2026].

Cliffe, E., 2023. *Getting started with accessible maths.* Available at: https://stem-enable.github.io/Getting-started-with-accessible-maths-March-2023/index.html [Accessed 8 January 2026].

Cliffe, E., 2022. *Using Word to Write an Accessible Mathematical Document*. Paper presented at the CETL-MSOR Conference, Abertay University, 2022. Available at: https://stem-enable.github.io/WordWorkshop/ [Accessed 23 April 2026].

Cooper, M., 2012. Making online maths accessible to disabled students – issues and lessons from the Open University's experience. In: E. Cliffe and P. Rowlett, eds. *Good Practice on Inclusive Curricula in the Mathematical Sciences.* Maths, Stats & OR Network. pp. 39–45.

Desmos Studio PBC, 2026. *Desmos Studio.* Available at: https://www.desmos.com/ [Accessed 12 January 2026].

Desmos Studio PBC, 2025. *Labels.* Available at: https://help.desmos.com/hc/en-us/articles/4405487300877-Labels [Accessed 12 December 2025].

DIAGRAM Center, 2015. *Image Description Guidelines.* Available at: http://diagramcenter.org/table-of-contents-2.html [Accessed 9 January 2026].

GeoGebra GMBH, 2026. *GeoGebra.* Available at: https://www.geogebra.org/ [Accessed 08 January 2026].

JISC A11Y-Maths Group, 2024. *JISC Accessible Maths Working Group.* Available at: https://sites.google.com/view/accessible-maths/home?authuser=0 [Accessed 8 January 2026].

NIST, 2024. *LaTeXML A LaTeX to XML/HTML/MathML Converter*. Available at https://math.nist.gov/~BMiller/LaTeXML/ [Accessed 14 January 2026].

NV Access, 2025. *NVDA.* Available at: https://www.nvaccess.org/about-nvda/ [Accessed 13 January 2026].

Price, I., Baptista, M., Collinson, M., Martin, B., Oren N., Richard M., 2021. *Accessible Online Maths: Investigating Solutions to Convert LaTeX Teaching Material to HTML*. Paper presented at the CETL-MSOR Conference, Coventry University, 2021.

Quarto, 2025. *Welcome to Quarto*. Available at: https://quarto.org/ [Accessed 14 January 2026].

Scottish Credit and Qualification Framework Partnership, 2026. *Interactive Framework*. Available at: https://scqf.org.uk/the-framework/interactive-framework/ [Accessed 23 April 2026].

Sight Scotland, 2025. *Supporting people in Scotland with sight loss.* Available at: https://sightscotland.org.uk/ [Accessed 13 January 2026].

University of Bath, 2026. *Introduction to Accessible STEM Content.* Available at: https://teachinghub.bath.ac.uk/guide/writing-accessible-technical-content-2/ [Accessed 8 January 2026].

University of Bath, 2025. *How to create accessible figures and diagrams.* Available at: https://teachinghub.bath.ac.uk/guide/accessible-maths-diagrams-and-figures-guidance/ [Accessed 12 December 2025].

University of York, 2024. *Accessible Maths Primer.* Available at: https://xerte.york.ac.uk/play.php?template_id=2036#primer [Accessed 8 January 2026].

WebAim, 2024. *Screen Reader User Survey #10 Results.* Available at: https://webaim.org/projects/screenreadersurvey10/ [Accessed 9 January 2026].

Wilson, H., 2020. *Everyone can SCULPT for Accessibility.* Available at: https://abilitynet.org.uk/resources/digital-accessibility/everyone-can-sculpt-accessibility [Accessed 9 January 2026].