July 1989 **Nemeth Braille Tools for BEX**

# MathematiX

**Raised Dot Computing, Incorporated**
**408 South Baldwin Street**
**Madison, Wisconsin 53703 USA**
Business: **608-257-9595**
Technical: **608-257-8833**

## Copyright

## Trademarks

## Thanks To Our Advisors

We wish to express our deep appreciation to the following individuals, who answered questions and tested the software:

Roberta Becker
Kent Cullers
Betsy Doane
Carol Gear

Phyllis Herrington
Stephen Murgaski
Lori Scharff
Diane Spence

*with especial thanks to*

Mark Dubnick
Yohei Yagi

and of course, Dr. Abraham Nemeth and his Code, who made it all possible.

While these folks kindly assisted us in developing MathematiX, they are in no way responsible for the accuracy of the software or its documentation.

## Program Design, Coding, & Documentation

| David | Jesse | Caryn | Nevin |
|---|---|---|---|
| *This is a bug-free disk!* | Can I write the Tutorial yet? | There *is* a solution! | Does everyone know what they're doing? |
| Holladay | Kaysen | Navy | Olson |

## Production Notes

Written and edited with RDC's BEX on an Apple IIgs. Braille manual transcribed with RDC's ClasX & TranscriBEX software. Inkprint manual, in Palatino, Bookman, and Courier, created with Microsoft Word® 4.0. on an Apple Macintosh SE and Apple LaserWriter Plus. (Word files created from BEX chapters using Contextual Replace and Microsoft's RTF (Interchange Format. File transfer with BEX and Hayes's Smartcom II software.)

# Corrections to the Inkprint MathematiX Manual

For each item, we provide the page number, approximate location on the page, and a brief error description. This is followed by the **corrected** material.

## 3:3 **second paragraph:** spurious "4"s in sample

The tix translation tools are extensively demonstrated in Sections 5 and 9; here's a quick sample. To tix the inkprint for "three-eighths is smaller than three-fourths," you can enter either: {`3_/8 @l is small] ?an @m 3_/4`} or {`3_/8 is smaller than 3_/4`}. To tix a fraction with a numerator of 5 and a denominator of the word "hours," you can enter either:
{`?5/_ @l h\rs_ @m #`} or {`?5/hours#`}.

## 3:4 **final paragraph:** "closing" should be "opening"

Yes, you do enter the same two cells for question mark and opening outer quotation: MathematiX is smart enough to use context to tell the difference and tix the appropriate sign. As detailed in Section 5, Part 13, MathematiX doesn't recognize the standard {`_0`} Nemeth Code empty set symbol. You use {`@_0`} instead to distinguish empty set from closing outer quotation.

## 4:3 **first paragraph:** missing "k" in sample description

space represents the less-than sign: {`-n "k n`} says "negative n is less than n." When you want the less-than sign, you must put spaces on both sides. {`-n "kn<:]`} means "negative n followed by kn overbar." If MathematiX encountered {`-n "kn `}, it returns a structure error, thinking you'd started a modified expression with dot 5 and neglected to finish it. (Section 7 lists all the structure errors and how to recover.)

## 4:11 **first paragraph of "Tix Vertical Tool Syntax"** wrong cell for "above"

Each tool is composed of three cells: Dots 2-4-6 {`[`}, followed by a direction shown with above {`<`} or below {`%`}, followed by a lowercase letter code for distance moved. (A list of the fourteen valid combinations appears on the MathematiX Reference Card.)

## 5:8 **Fourth item in chart:** mislabelled

2.b {`:"k .k"k`}

Two rarer variations on "is less than or equal to"

### 7:7 **Next-to-last paragraph:** wrong cell

That's why {**1/2**} is announced as "one slash two" instead of "one fraction line two." When MathematiX gets to the termination indicator {**]**}, it realizes that nothing has been started, so it reports "Extra termination sign." Finally, it reaches the BEX page chunk boundary, and complains about the {**,#**} which has not been preceded by {,/}.

### A:4 **Seventh item under "O":** wrong direction

**Output move down** Created from the tix vertical tool that shifts the inkprint vertical position. For example, {**[%h**} is verbalized as "output move down half character." (Examples in Section 4, Part 5 and Section 9, Parts 6 and 7.)

### C:1 **First item under "A":** wrong cells

aleph, lowercase Hebrew letter.......................................................................... {**,,a**}

### C:2 **Twenty-second item on page:** wrong cells

bet, lowercase Hebrew letter............................................................................. {**,,b**}

### ?:? **Omitted from Appendices B & C**

We neglected to include the "not equal to" symbol in the Alphabetical and Transcriber-Order Symbol Lists, but MathematiX *does* support it.

not equal to............................................................................................................ {**/.k**}

# Welcome to MathematiX!

MathematiX adds a *Math Menu* to your existing BEX program. The Math Menu lets you get verbal and regular print mathematical output from BEX chapters containing Nemeth Code braille. With MathematiX you can now use BEX to prepare documents that include text and fractions, square roots, chemistry, or other technical material for distribution to your sighted students, instructors, and colleagues.

## What's In the MathematiX Package

The MathematiX package should contain:

- **MathematiX Menu Disk:** As demonstrated in Section 2, you use this disk with your current BEX to get access to the Math Menu.
- **MathematiX Sample Data disk:** Contains chapters that create math output. The first chapter on this disk contains a brief guide to the rest.
- **MathematiX Manual***:* in regular print and in braille.
- **MathematiX Reference Card***:* regular print and braille booklets that summarize important MathematiX features.

If any item is missing, please call our Business Office at 608-257-9595 immediately!

## Symbols Used in the MathematiX Manual

For clarity, we use these techniques:

`{`**`yr brl 5try is %[n 1 ?`**`}`. In the braille edition, full cells bracket the exact cells and spaces you enter. Don't type the full cells themselves. In the inkprint edition, the braces enclose the "screen braille" characters and spaces in the Courier typefont. Don't type the braces themselves. When you read `{`**`_ @l `**`}` it means you type five characters: dots 4-5-6, space, dot 4, dots 1-2-3, space. The Section 2 Tutorial and BEX Appendix 1 provide more information on screen braille.

`The Computer Prompts:` **`Your Response`**`.` The braille edition uses the Computer Braille Code so you know the exact characters. In inkprint, you see bold Palatino type in text, and Courier type for extended computer dialogs.

<CR> **<space> <control-S>**: Represent a single character each. You make a <CR> by pressing the "Return" key. When it's crucial that you press the spacebar when dealing with the computer, we show this with <space>. You can create the single **<control-S>** character in BEX's Editor by pressing **control-C** S.

## Where To Go From Here

Before you turn on the computer and start playing with MathematiX, we urge you to read all of Section 1. You'll find out what MathematiX can and can't do, what equipment you need, and how to set up BEX to work with MathematiX. Then you'll be ready for Section 2, which takes you on a guided tour of all MathematiX's functions. To understand why and how you separate literary and mathematical braille in your MathematiX chapters, read Section 3. Section 4 explains how you and MathematiX work together to format your inkprint. When you need to know which Nemeth symbols to use to express a particular inkprint sign, check Section 5, Appendix B (Supported Symbols in Transcriber Order), or Appendix C (Supported Symbols in Alphabetical Order).

## How To Get Help If You Encounter Problems

Raised Dot Computing wants you to make best use of MathematiX. When you run into problems, check Section 8 to see if the answer's here in the MathematiX Manual, and the **KNOWN MATHEMATIX BUGS** chapter on the MathematiX Sample Data disk. If you're still boggled, please get in touch with us. To help us help you better, we ask that you:

- Make sure that the data disks containing problem chapters are on hand.
- Dig out your MathematiX Manual so you can refer to it during our conversation.
- Position the telephone near your Apple, with MathematiX up and running, if possible.
- Note down the name and model of your printer and interface card.
- Use option W - What is in the computer on BEX's Starting Menu to find out which cards are in what slots.
- Call our Technical Helpline at 608-257-8833.

# MathematiX

## The Basics of Tix-Ready Chapters 3

## Formatting Math Output 4

## Math Menu Reference 6

## Checking Your Tix-Ready Chapters with Verbalize 7

## Troubleshooting 8

## Putting MathematiX to Work 9

# Section 1: Overview

# Section 1
# Overview

MathematiX adds a new *Math Menu* to your BEX program. The Math Menu lets you get verbal and regular print mathematical output from BEX chapters containing Nemeth Code braille. MathematiX does *not* translate inkprint into Nemeth Code. We urge you to read all of this Section before you turn on the Apple. As we explain in Part 1, to use MathematiX, you must already know something about the Nemeth Code system. Read Part 2 to see where MathematiX is different from Nemeth Code. MathematiX is not a separate program, it's an additional module for BEX. Parts 3 and 4 explain the BEX skills you need for MathematiX.

## 1 WHAT DOES MATHEMATIX DO?

MathematiX's main purpose is to prepare inkprint documents with technical symbols like fractions, exponents, square roots, etc. as well as text. MathematiX can only create regular print output, through a limited group of dot-matrix printers and interface cards—the same restrictions that apply to BEX's large print. MathematiX lets you prepare documents for distribution to your sighted instructors, students, or colleagues.

MathematiX can also be used to proofread the accuracy of Nemeth Code, with either verbalization through a voice synthesizer or regular print characters on the Apple screen. The blind user can check to make sure that what's entered will be accurately interpreted by MathematiX when it comes time for print output. The sighted transcriber can use MathematiX's screen preview to check the accuracy of Nemeth Code data entry of informal material for classroom use.

### ■ "Tixing:" How MathematiX Processes Your Chapters

When you use option B - Back translate from Grade 2, BEX expands the contractions of literary braille to their inkprint equivalents. When you use option P - Print chapters, BEX makes hard copy by breaking your text into lines and pages while obeying the format indicators and commands in your chapter. When you use option M - Math output on the Math Menu, MathematiX does both tasks: it expands mathematical symbols into inkprint signs; it expands literary braille into inkprint letters; and it organizes all this information onto output pages. Throughout this manual, we use the verb *tix* to describe this combination of back-translation and formatting. When you use MathematiX, you can tix a chapter to a dot-matrix printer for regular print, or to a voice synthesizer or the screen for proofreading.

### Prerequisites: What You Need to Know to Use MathematiX Effectively

This manual won't attempt to teach you Nemeth Code. We recommend you get a copy of the *Nemeth Braille Code for Mathematics and Science Notation*, which is available from:

Print Catalog No.: 7-8743  
Braille Catalog No.: 5-8743  
American Printing House for the Blind  
1829 Frankfort Avenue  
Louisville KY 40206  
Telephone: 502-895-2405

In order to create a particular inkprint sign, you'll need to know the corresponding Nemeth Code symbol. This manual also makes no attempt to teach the subjects that are represented in Nemeth Code. In Appendix A, the Glossary provides brief definitions of the technical vocabulary we use to describe various inkprint signs. To really understand what those symbols mean and how to use them correctly, consult your math or science teacher.

## 2 WHERE MATHEMATIX DEVIATES FROM THE NEMETH STANDARDS

### MathematiX Needs Your Help to Understand Nemeth Code

*The Nemeth Braille Code for Mathematics and Science Notation* defines how to represent a wide variety of inkprint *signs* through the use of specific braille *symbols*. Some braille symbols are just one cell long: {**+**} (dots 3-4-6) shows the inkprint plus sign. Other braille symbols require several cells: {**$%33o**} (dots 1-2-4-6, 1-4-6, 2-5, 2-5, 1-3-5) represents a single down arrow sign in inkprint. But technical materials aren't all arrows and operators: the Nemeth Code uses literary (Grade 2) braille for the text. Nemeth Code was designed to be read by humans, not by computers. Humans handle ambiguity better than computers, since we can pick up clues from context. In particular, MathematiX cannot "know" the difference between math and literary braille.

That's why MathematiX requires you to explicitly distinguish between *literary* and *math* material in your tix-ready chapters. The Nemeth braille reader depends on context to decide when {**,***} (dot 6, dots 1-6) is representing the literary word *Child* and when it's the Nemeth symbol for the inkprint *therefore* sign. MathematiX, on the other hand, always assumes that {**,***} is *therefore* unless you tell it otherwise. Similarly, context lets the braille reader decide when dot 2 is representing a digit and when it's showing the literary comma. In addition to marking the boundaries between literary and math in your chapters, you also have to tell MathematiX the difference between punctuation and digits—details appear in Section 3.

### ■ Using Nemeth Code to Create Inkprint

A Nemeth Code transcription is based on the appearance of the inkprint, not its mathematical meaning. Since Nemeth Code includes symbols and indicators describing the spatial relationship of inkprint signs, MathematiX can use Nemeth to create inkprint output. However, Nemeth was not designed as a means of writing inkprint mathematics, and MathematiX needs some additional inkprint-specific information to tix correctly. To obtain intelligible inkprint, you have to write braille that does not conform to standard Nemeth Code. Section 4 discusses the issues of spacing, superimposing, and enlarging characters. MathematiX supports many but not all Nemeth symbols: Section 5 provides step-by-step guidance for how to enter the supported symbols, as well as a list of the unsupported ones.

We've attempted to make MathematiX's regular print output as clear as possible. Dr. Caryn Navy used MathematiX to produce hundreds of math handouts for her university courses. Not only did the sighted students find them easy to read, her sighted colleagues were jealous of her ability to make printed, as opposed to hand-written, materials. While MathematiX output is suitable for homework, papers, and rough drafts of theses for the technical typist, it's definitely *not* publication quality.

## 3 HARDWARE REQUIREMENTS

### *128K Memory and Two Disk Drives*

MathematiX requires an Apple II with at least 128K memory. An Apple IIgs or an Apple IIc always have at least 128K memory; you must install an *extended 80-column card* in an Apple IIe to get the memory up to 128K. (Details in BEX Interface Guide 1:2). An Apple II Plus can only have 64K memory, so you can't run MathematiX on an Apple II Plus. If you're not sure how much memory your Apple has, use option W on BEX's Starting Menu to find out.

You need at least two disk drives to run MathematiX; one of these drives must be a 5.25-inch floppy disk drive. At the Master Level, you can establish "extended disk systems" that include 3.5-inch disk drives, RAM drives, and/or the Sider hard disk system. As long as your configuration has more than one disk drive (5.25 inch, 3.5 inch, RAM memory, or Sider hard disk) you can use MathematiX. Using RAM drives and 3.5-inch disk drives requires BEX 3.0; you *can* load the MathematiX software on RAM drive.

### *"Large Print" Printer Requirements*

BEX uses a particular set of programming techniques to make large print output. When MathematiX tixes to a printer, it uses those same programming techniques to make *regular print* that includes math. Although MathematiX output is *not* large print, MathematiX requires that your dot-matrix printer and interface be compatible with BEX large print. This means you need:

- Apple ImageWriter
- Apple ImageWriter II
- Epson FX-80
- a truly ImageWriter or Epson FX-80 compatible clone

connected to your Apple through:

- Apple Parallel Card
- Apple Super Serial Card
- Apple IIc Serial Port
- Apple IIgs Serial Port
- SlotBuster Serial *or* Parallel Port
- Grappler Plus Parallel Card
- ProGrappler Parallel Card (version 2.0 or later)

See the BEX Interface Guide Section 4 on Printers for details.

## 4 GETTING READY TO USE MATHEMATIX WITH BEX

MathematiX is an add-on to BEX. Before you can use MathematiX, you have to know how to use the base program. This manual assumes that you've read through at least the BEX User Level: you know what BEX chapters are, and how to select, edit, and print them. We won't explain how to do all these things here—the MathematiX Manual is long enough. Instead, we just point your way to relevant sections of the BEX Dox for crucial facts.

### *MathematiX works with BEX 2.2 or BEX 3.0*

Your BEX version must be 2.2 or later to work with MathematiX—use option U - Update date on the Starting Menu to check your BEX version. If you own BEX 2.1, contact us for update or upgrade information. Beginning with BEX 3.0, BEX works well on the Apple IIgs (taking advantage of expansion memory and doing large print through a built-in port.) Throughout this manual, we show samples of MathematiX and BEX screen output. The BEX samples are based on BEX version 3.0; if you're using an earlier version, the wording of the prompts will vary slightly.

### *Transferring Tix-Ready Data from an External Braille Device*

While you can enter braille in BEX's Editor, a stand-alone braille computer is the best environment for writing braille materials. When you write your material on an external device like the VersaBraille or Braille 'n Speak, you must know how to transfer the data into BEX chapters. The BEX Interface Guide gives the nitty gritty details of cabling and switch settings, and the BEX User Level has procedural instructions. User Level Section 11 describes BEX and the VersaBrailles; User Level Section 12 describes Input through Slot, which lets you create a BEX chapter by sending characters in to the Apple through a serial port.

The *RDC Newsletter* April/May 1988 issue has Robert Carter's article detailing the Braille 'n Speak interface.

## Configuring BEX for MathematiX

As always, your BEX configuration describes your equipment preferences. At a minimum, your MathematiX configuration must:

- Be at the User or Master Level
- Include one large print printer

MathematiX won't run at the Learner Level of BEX. While MathematiX output is regular print, it uses the same programming techniques as BEX large print. When you ask MathematiX to tix in inkprint, MathematiX must find a large print printer in your configuration in order to access your printer's graphics capabilities. If you can make large print, you can make MathematiX output. If you're having trouble making large print, check Section 4 of the Interface Guide. If you're still boggled, please call Raised Dot Computing's technical helpline at 608-257-8833 so we can get you in business.

### *Configuring a Large Print Printer for MathematiX*

Once you include a large print printer in your configuration, MathematiX can make math output. All MathematiX cares about is that you've configured one large print printer; when it's tixing inkprint MathematiX completely ignores the font size, line spacing, carriage width, and form length values you supply. However, you can also get large print output from this device by using option P - Print on the Math or Main Menus. When you're printing, your configuration values do affect output. When you're tixing, they don't—more on this topic in Section 4.

### *Sample MathematiX-compatible Configurations*

Here are two possible configurations that work with MathematiX, as they'd appear when you use option V - View a configuration on the Starting Menu.

```
User level configuration: ADA
Apple 2e computer
Using 40 column screen
Using Echo speech
      using fast speech
      using a pitch code of 24
      most punctuation spoken
      using a volume code of 8
Model D tape-based VersaBraille in slot 2
Printer one using slot 1
      Large print printer
      carriage width 40; form length 25
      auto linefeed on
      Sequence: <Esc>T16
Printer two using slot 1
      Generic printer
      carriage width 72; form length 56
      Sequence: <Esc>c<Esc>L008
Printer three using slot 3
```

```
        Review class printer
        carriage width 72; form length 56
  Printer four using slot 2
        Brailler; T S I VersaPoint
        carriage width 41; form length 25
  There are 2 disk drives:
        1 is slot 6; drive 1
        2 is slot 6; drive 2
```

This ADA user has a switch box attached to slot 2, so she can connect both her VersaBraille and her VersaPoint to a single Super Serial Card. She prepares tix-ready chapters on her VersaBraille, and transfers them to the Apple through slot 2; when she wants hardcopy braille, she prints it to the VersaPoint in that slot. Her ImageWriter in slot 1 plays three roles. When she prints, she can specify it as printer 1 for large print or printer 2 for regular print output. (The printer 2 set-up sequence starts with a software reset command **<Esc>c** that clears out lingering graphics spacing, followed by an eight-character left margin.) When she uses option M - Math output on the Math Menu, MathematiX tixes to the same ImageWriter in regular print.

```
  Master level configuration: JOHN
  Apple 2gs computer
  Using 80 column screen
  Using Echo speech
        using fast speech
        using a pitch code of 18
        using a volume code of 10
  Remote serial device in slot 2
  Printer one using slot 1
        Specific printer; ImageWriter 1
        carriage width 72; form length 56
        Sequence: <Esc>c<Esc>L008
  Printer two using slot 1
        Large print printer
        carriage width 40; form length 25
        auto linefeed on
        Sequence: <Esc>T16
  Printer three using slot 2
        Paperless brailler
        carriage width 40; form length 0
  Printer four using slot 3
        Review class printer
        carriage width 72; form length 56
  There are 6 disk drives:
        1 is slot 3; drive 1
        2 is slot 6; drive 1
        3 is slot 5; drive 1
        4 is slot 5; drive 3
        5 is slot 3; drive 3
        6 is slot 3; drive 2
```

The ImageWriter is interfaced through the Apple IIgs's built-in serial port, and is referenced twice in the configuration: printer 1 is a regular print specific printer, while printer 2 is BEX large print. MathematiX can tix inkprint thanks to the printer 2 definition. Master JOHN has a Braille 'n Speak to prepare his tix-ready chapters. He installed a Super Serial Card in the IIgs's slot 2 (and set the Control Panel to "your card") so he can transfer files between the Braille 'n Speak and

BEX. He uses Input through slot to get material from the Braille 'n Speak to the Apple, and prints to the "Paperless brailler" (printer 3) to send Apple text to the Braille 'n Speak. The Apple IIgs has one 5.25-inch disk drive (virtual drive 2), one 3.5-inch disk drive (virtual drives 3 and 4), and 512K memory. John loads both the Main side of BEX and the MathematiX software on RAM drive—more details on why he numbered the RAM drives this way in Section 10.

# Section 2: A Guided Tour Though MathematiX

# Section 2
# A Guided Tour through MathematiX

With MathematiX, you can prepare inkprint materials that include text and math, chemistry, and scientific notation. The starting point for MathematiX is BEX chapters containing a modified form of Nemeth Code braille. You can create these chapters with BEX's Editor. Writing and editing braille is even easier on a braille-oriented device like the VersaBraille; you can prepare the data on another device and then bring it in to BEX with From VersaBraille or Input through Slot. If you're in doubt about how you transfer information from another device into BEX chapters, please see BEX User Level 11 for how to proceed.

Once you have BEX chapters, there are two steps to the MathematiX process. You proofread the chapters to check for structural errors, with feedback in spoken words or as screen graphics. You can fix any errors you find in the BEX Editor. When you know the chapters are correct, you output this data to a dot-matrix printer. This tutorial Section demonstrates exactly how you use MathematiX to accomplish these tasks.

## ■ Getting Ready to Go

The best way to learn about MathematiX is to actually follow along with every step in this Section. Before you begin, you need to establish a BEX configuration that's compatible with MathematiX. As detailed in Section 1, Part 4 this means you configure at the User or Master Level, *and* you define at least one large print printer. When you ask MathematiX to produce regular-print inkprint, MathematiX must find a large print printer in your configuration in order to access your printer's graphics capabilities. Two sample configurations that work with MathematiX are shown in Section 1, Part 4. We suggest you interface and test the large print output first, so you can focus on learning about MathematiX in this tutorial.

Once you have established and tested this configuration, here's what you need for the tour:

- around ten sheets of paper for your dot-matrix printer.
- two high-quality blank disks.
- around two hours of time.

## 1 MAKING WORKING COPIES OF YOUR MATHEMATIX DISKS

Boot BEX, specify your MathematiX-compatible configuration by name, and get to the Starting Menu. To ensure that you always have access to MathematiX, we urge you to make a working copy of the two disks in the MathematiX package: the MathematiX Menu Disk and the MathematiX Sample Data disk. Floppy disks can be easily destroyed. Once you make working copies, store the original disks in a different location. If you didn't make working copies and misfortune rendered your original disks useless, RDC would charge you $25 to replace them.

The MathematiX disks are not copy-protected, but they are copyrighted by Raised Dot Computing, Inc. RDC's copyright for MathematiX is identical to the BEX copyright. You may use MathematiX with BEX on any one computer at one time. You may not make multiple copies of BEX or MathematiX to use on more than one computer at a time. When you want to use MathematiX on more than one computer simultaneously, contact RDC for special "pack" discounts.

Employing option C - Copy disks on BEX's Starting Menu, make working copies of both MathematiX disks on to the two high-quality blank disks. (Don't copy the two master disks on to both sides of one flippy disk—you'll see why this wouldn't work in Part 3.) Insert the MathematiX Menu Disk in drive 1, one high-quality blank disk in drive 2, and press **C** at the Starting Menu. BEX describes the copying process and prompts you to press Return—do so and then sit back for around two minutes. Repeat this process with the MathematiX Sample Data disk. Label your working copies, then remove your original MathematiX disks to a safe place. That way, if a working copy is damaged in the future, you can make a new one from the original.

## 2 INTRODUCING THE MATH MENU

Now that you have working copies of the MathematiX disks, you can explore MathematiX. The MathematiX Menu Disk is *not* a bootable program disk. Instead, it adds a new menu, the Math Menu, to your existing BEX program. You must be at BEX's Main Menu to move to the Math Menu. Insert your BEX Main Disk in drive 1 and press the spacebar to move from the Starting Menu to the Main Menu.

### *Main to Math With Spacebar*

At the Main Menu, remove the Main Disk from drive 1 and replace it with the MathematiX Menu Disk. Press the spacebar to move from the Main Menu to the Math Menu—there's a short pause while BEX reads software from the disk and then you're presented with the **Math Menu:** prompt. (When you've configured at the Master Level, the prompt is shortened to simply **Math:**.)

If your configuration wouldn't work with MathematiX, then you get an error message describing what's lacking. Read BEX User Level Section 3 on how to establish a new configuration that includes a large print printer, and please do so before proceeding with the rest of the tutorial.

### *From Math to Other Menus*

Like all BEX menus, you can list your choices by pressing <CR> at the **Math Menu** prompt. Press <CR> now and you'll see many familiar options. The heart of MathematiX is option M - Math Output, which we demonstrate in detail real soon. The other options work more or less identically to their cousins on the Main and Second Menus—full details in Section 6.

The Second and Page Menus aren't directly accessible from the Math Menu—you have to go to the Main Menu first. To move from the Math Menu to the Main Menu, you insert the BEX Main Disk in drive 1 and press J. To go from the Math Menu to the Starting Menu, you insert the BEX Boot Disk in drive 1 and press spacebar. To get from the Starting Menu back to the Math Menu, you must first move to the Main Menu.

Try it out! Place the BEX Main Disk in drive 1 and press **J**. There's a brief disk access and BEX presents the Main Menu prompt. Now put the MathematiX Menu Disk in drive 1 and press spacebar, and you're back at the Math Menu. Press spacebar, and MathematiX asks you to confirm that you want to move to the Starting Menu. Place the BEX Boot Disk in drive 1 and press <CR>. Now put the MathematiX Menu Disk back in drive 1 and press spacebar again—BEX reminds you that the Math Menu is only accessible from the Main Menu. Swap disks to get to the Main Menu, then swap again to get to the Math Menu. These disk swaps will become second nature after a little experience—you can refer to the chart on the MathematiX Reference Card for guidance.

### ■ Now You Know ...

You have learned the basics of navigating the MathematiX Menu Disk. Keep in mind:

- MathematiX requires a compatible BEX configuration
- The Math Menu is only accessible from BEX's Main Menu

You're ready to actually use the Math Menu.

## 3 TRYING OUT OPTION M - MATH OUTPUT

Get back to the Math Menu, swapping disks as needed. Insert the MathematiX Sample Data disk in drive 2. Like all BEX menus, the Math Menu has option D - Disk catalog. Press **D** and then <CR> to see the chapters on the disk. For your convenience, we supply several chapters to use in the tutorial: Some were written on a VersaBraille, and some were created in the BEX Editor.

When you use Math Output, MathematiX expands mathematical symbols into inkprint signs and literary braille into inkprint letters. Math Output also organizes all this information onto output pages. We use the verb *tix* to describe this combination of back-translation and formatting. The chapters that end in the # number sign character are *tix-ready chapters,* containing the modified Nemeth Code that MathematiX requires for correct output. Math Output provides three tix destinations: hardcopy regular-size inkprint on your printer; screen preview for sighted users; and verbal preview of the content of the chapter.

### *Verbalize to Proofread Content*

Let's see what Verbalize does. You may wish to switch to expanded speech at this point by issuing the command **control-E E**. Now with your MathematiX Menu Disk in drive 1 and the MathematiX Sample Data disk in drive 2, do this:

```
Math Menu: M
Math output
Drive or chapter: HELLO# <CR>
Drive or chapter: <CR>
Tix where: ? <CR>
S - screen preview
V - verbalize
```

```
I - inkprint hard copy
Tix where: V <CR>
```

Entering question mark gives you an on-line reminder of the three tix destinations. Once you answer **V <CR>** for the Verbalize destination, MathematiX reads briefly from both disks, and then starts the *tix scratch* to tell you it's working. Whenever you use Math Output, MathematiX must read software from the MathematiX Menu Disk. (This is why MathematiX requires two disk drives, and why we advised you not to copy the MathematiX disks onto a flippy disk.) When the tix scratch and disk access is finished, MathematiX starts verbalizing.

For Verbalize, Math Output sends the **HELLO#** chapter to your voice synthesizer and the 40-column screen. Literary material is spoken as words, while mathematical material is spelled out sign for sign and letter by letter. BEX format commands are also spelled out, not executed. Verbalize provides a way to proofread the structure and content of a tix-ready chapter; it does not preview the format.

When the screen is full, output pauses and you hear a low boop. You can use screen review to examine what's been said. Press the spacebar when you're ready for more output. You quickly get to the end of the sample information in the HELLO# chapter; press the spacebar once more to return to the Math Menu prompt.

### *Screen Preview in Graphics for Sighted Users*

If you can see the 40-column screen, you can check both the content and the format of the HELLO# chapter. Follow this sample:

```
Math Menu: M
Math output
Drive or chapter: HELLO# <CR>
Drive or chapter: <CR>
Tix where: S <CR>
```

The word **tixing ...** on the screen accompanies the tix scratch. The format commands in the **HELLO#** chapter are executed. Because the chapter begins with **$$np**, you see **Page 1** at the bottom of the first screen. When the screen is full, you hear a boop; press spacebar for the next screen.

## ■ Now You Know ...

This quick sample has shown you the basic pattern when using Math Output:

- Math Output can only handle tix-ready chapters
- Name tix-ready chapters with a final number-sign character to identify them at a glance
- Once you specify the chapters, you tell MathematiX where to tix
- The Verbalize destination allows you to know exactly how MathematiX interprets the braille material in your chapter

It's time to find out what makes a chapter "tix-ready."

# 4 EXAMINING A TIX-READY CHAPTER

The **THREE Q#** chapter on the MathematiX Sample Data disk contains three review questions and solutions for the dreaded Imaginary MathematiX Aptitude Test. The Math Menu duplicates the Editor option from your Main Menu. You can use all and any of the Editor commands explained in BEX User Section 5. Edit the **THREE Q#** chapter, and then issue the command **control-S L** to lock out changes. Now any Editor command that would add or delete text just beeps, so you can review the contents of the chapter without changing anything.

### *Understanding Screen Braille*

When you read a braille chapter in the Editor, the screen and your voice synthesizer show the information in the *screen braille* system. As detailed in BEX Appendix 1, each braille cell has a unique ASCII inkprint equivalent. For example, the @ at-sign is the same as dot 4. Many important cells are shown with punctuation, so be sure to request "most" punctuation from your voice device. The MathematiX Reference Card includes a chart of the screen braille characters. To make it easier to examine this chapter, we provide a hard-copy version below. Sighted users can request braille dot patterns on the screen by entering the command **control-S S B**.

### *Hard-copy version of* THREE Q# *page 1*

The next paragraph shows all the characters in the first BEX page of the **THREE Q#** chapter, exactly as they appear in the Editor (or on a VersaBraille display).

```
 @l $$np $$h ,ma? ,review = ,,imat $p $$c ,sample #a3 ,a4+
,rates $$i0$$ml0 $p ,frank takes @m ?2/3# @l h\rs 6pa9t a
po/1 :ile ,nancy c pa9t ! po/ 9 @m ?3/4# @l h\rs4 ,if !y "w
tgr 6pa9t "o po/1 h[ _m m9utes w x take !m 6f9i% ! job8 $$ml6
$p ,,solu;n3 ,! orig9al data is giv5 9 @m ?hours/post#_1 @l b
x is easi] 6"w ) po/s p] h\r4 ,frank c pa9t @m ?3/2# @l po/s
p] h\rl & ,nancy c pa9t @m ?4/3# @l po/s p] h\r4 ,if !y "w
tgr = an h\rl !y c pa9t @m ?3/2#+?4/3# .k ?9/6#+?8/6# .k
?9+8/6# .k ?17/6# @l po/s p] h\r4 ,? -b9$ rate ( @m ?17/6# @l
po/s p] h\r c al 2 express$ z @m ?6/17# @l h\rs p] po/1 or @m
#0.353 @l ( an h\r4 ,ea* m9ute is @m 1_/60 @l ( an h\r4 ,use
@m #60 @* 0.353 @l 6f9d ! total m9utes =! job--approximately
#ba4 $$ml0 $f $$c @l ,sample #b3 ,"r ,triangles $p ,a
triangle ) "o @m #90^.* @l angle is call$ a $$ub ."r $$uf
triangle4 ,! side t is opposite ! "r angle is call$ !
hypot5use4 ,! ,py?agorln !orem tells u t ! squ>e (! l5g? (!
hypot5use is ! sum (! squ>es (! o!r two sides4 ,if y h a "r
triangle ) "o side ( @m #6''_1 @l & ano!r side ( @m #8''_1 @l
:at is ! l5g? (! hypot5use8 $$ml6 $p ,,solu;n3 ,9 ma!matical
not,nl ! ,py?agorln !orem is /at$ z3 @m x^2"+y^2 .k z^2_4 @l
,s9ce y "k ! l5g? ( two sidesl & nei is /at$ 6be ! hypot5usel
all y d is solve = ;z3 @m z^2 .k #6^2"+8^2 .k 36+64 .k 100_4
,since #10^2 .k #100_1 @l ! hypot5use is @m #10''_4 @l
,alt]natively1 y c "w ! pro#m ? way3 @m $1 z .k >x^2"+y^2"]
.k >6^2"+8^2"] .k >36+64] .k >100] .k #10 $$ml0
```

## *Marking Math and Literary Braille with Tix Translation Tools*

Unlike standard Nemeth Code, MathematiX requires you to explicitly distinguish between *literary* and *math* material in your tix-ready chapters. The human reader uses context to decide when {,*} (dot 6, dots 1-6) means *Child* and when it's the Nemeth symbol for the inkprint *therefore* sign. But unless you tell it otherwise, MathematiX assumes that {,*} is always *therefore*. In other words, MathematiX always begins in math mode.

Press **control-G** to go ahead one word: the first four characters in **THREE Q#** are space, at-sign, lowercase *l*, space. This is the tix translation tool that turns on MathematiX's literary mode. Let's jump ahead to some math material, by locating for other tix translation tools.

Press **control-L** to begin the locate command, then type **<space> @**, then press **control-A** to advance the cursor. You land at character 92—you can hear the math mode tix translation tool by pressing **control-G**. You must alert MathematiX to math material with the four characters space, dot 4, lowercase *m*, space. Press **control-G** to hear the math material: it's the simple fraction two-thirds, spoken as question mark, two, slash, three, number sign. The next "word" is the literary mode tool, followed by the grade 2 words *hours to paint a post*. When you use option M - Math output, the four-character tix translation tools become one space. In Section 3, Part 2 we explain two other tix translation tools, which don't create a space when output.

## *Using BEX Format Commands and Indicators*

You've seen that a tix-ready chapter contains literary and math braille, with tix translation tools marking their boundaries. In addition, you can use familiar BEX format tools to control the inkprint output. Just like BEX, you show a new paragraph with space, dollar sign, lowercase *p*, space. Prove it to yourself: Issue the command **control-A control-P** to advance to the next paragraph indicator. Now press **control-R** to hear the previous word. It's the BEX format command **$$ml6** which establishes a left margin of six characters. The problem statements use the full width of the page, while the solutions have a six-character left indent. Two **control-G** commands move over the paragraph indicator, and another **control-G** gets you the grade 2 word *SOLUTION* followed by a colon. You can use many, but not all, of the BEX format commands to control inkprint output with MathematiX—details in Section 4.

## *Labelling Punctuation in Math Mode*

Punctuation symbols are another area where MathematiX requires more information than standard Nemeth. MathematiX can't use context to know when dot 2 is representing the digit 1 and when it's showing the literary comma. Repeat the **control-L control-A** command to place your cursor at character 278, another math mode tix translation tool. Use **control-G** to read the next two "words." After the space, at-sign, m, space, there's a simple fraction followed by a comma. This is spoken as question mark, hours, slash, post, number sign, underline, 1. The underline is the Nemeth Code Punctuation Indicator, dots 4-5-6. As detailed in Section 3, Part 2, MathematiX requires you to use the punctuation indicator in math mode with eleven symbols.

There's one more interesting thing about this fraction: its numerator and denominator are words, not digits. Because this fraction is in math mode, no contractions are taken in the numerator *hours* and the denominator *post*. You can't write *any* literary contractions in MathematiX's math mode. Although you can tell by context when dots 3-4 means "st" and when it means "fraction line," MathematiX is not that clever.

### *Italics and Underlining*

To see what else a tix-ready chapter can contain, let's check out the second problem. We begin each problem with { **$f** } to force a page break. Locate for these four characters to advance to character 736, where Sample 2 begins. The format command **$$ml0** appears right before the new-page indicator, cancelling the six-character left margin used for the solution to Sample 1. Advance to the next paragraph with **control-A control-P**, and move through the first sentence with **control-G**. The first four words are in literary braille. The math mode tix translation tool comes next, followed by the Nemeth for *ninety degrees*. Then there's the literary mode tool, followed by four more words. Now you find the BEX underline begin command **$$ub**, the literary word *right*, and the underline finish command **$$uf**. The underlined word is italicized in braille—it begins with dots 4-6, the period in screen braille. MathematiX ignores the presence of italics signs—when you want your literary material underlined, use the BEX format commands. MathematiX won't underline any material in math mode.

Now you have a general idea of what's in the **THREE Q#** chapter. Issue the Editor command **control-S L** to toggle off the lock out changes mode, and then enter **control-Q** to Quit the chapter.

### ■ Now You Know ...

The important things to remember about tix-ready chapters are:

- MathematiX won't recognize truly standard Nemeth Code
- In tix-ready chapters, literary and math braille are separated by tix translation tools
- You must use the punctuation indicator before punctuation in math mode
- You can use BEX's format indicators and commands to control output format

Let's see how Verbalize handles the material you've just reviewed.

## 5 VERBALIZING THE THREE Q# CHAPTER

With your MathematiX Menu Disk in drive 1 and the MathematiX Sample Data disk in drive 2, proceed like this:

```
Math Menu: M
Math output
Drive or chapter: THREE Q# <CR>
Drive or chapter: <CR>
Tix where: V <CR>
```

MathematiX always begins in math mode. The first characters in the chapter are { **@1** }, which turn on literary mode and create one space when tixed.

Therefore, the first thing you hear is **space**. Since the first three paragraphs are in literary mode, MathematiX verbalizes the material in words. The first math mode tix translation tool appears after the second "dollar sign, p" and the words "Frank takes." Because you're still in literary mode, you don't hear the output "space" that results when this { `@m` } is tixed. The problem statement switches between literary and math modes several times.

When MathematiX verbalizes the solution paragraph for sample 1, you hear more of the "picky" style used for math mode. The fraction that's composed of words is spelled out letter for letter: you hear **start fraction h o u r s fraction line p o s t end fraction comma**. When the screen is full and the synthesizer's finished talking, you hear a low boop. Press the spacebar for more data.

Sample 2 begins most of the way down the second screen of data. After you press spacebar a second time and move to the third screen, you hear one of the many instances where MathematiX can't express the exact mathematical meaning of the inkprint signs. Instead, MathematiX uses the most general term, leaving the interpretation up to you.

### *Understanding Verbalize's Generic Vocabulary*

Where the **THREE Q#** chapter has { `@m #6''_1 @l` }, the verbalized version is **6 double prime comma space**. The inkprint "double prime" sign is used in several contexts. In this case, double prime means "inches." When you're writing about degrees, minutes, and seconds, you use the same "double prime" symbol for the seconds. And when you're writing about variations on a variable name, you use the same symbol to literally convey the concept "double prime." Once you start using MathematiX, you will be verbalizing material that you have written. This will make it easier to interpret the meaning when MathematiX offers generic terms like "double prime."

There's another example of generic vocabulary on this third screen. You can find it with Screen Review. After the low boop that means the screen is full, press **control-L** to begin Screen Review, then press the letter **K** to read the 11th screen line. You hear: **stated as: x squared plus y squared**. Press the down arrow to hear the next line, which is **equals z superscript 2 baseline period space**. In the **THREE Q#** chapter, this material appears as {`/at$ z3 @m x^2"+y^2 .k z^2_4 @l` }. MathematiX attempts to say "squared" when it sees the digit 2 as an exponent. However, MathematiX can't always meet this goal, so the right-hand side of this equation is verbalized in more generic terms as "z followed by an exponent of 2."

### *Format Information from Verbalize*

Verbalize allows you to check the contents and meaning of your tix-ready chapters, but it doesn't provide a preview of the format. You can use BEX $$ commands and format indicators to control final inkprint format. When you Verbalize, the commands themselves are parrotted back to you. For easier listening, MathematiX uses a special vocabulary for the basic format indicators in math mode: "new paragraph" means { `$p` }, "new line" means { `$l` } or a hard <CR>, and "new page" means { `$f` }. Check out lines V and W on the current screen to see an example. The source material is {`way3 @m $l z .k`}, and it's verbalized as **way: new line z equals**. When you're in literary mode, MathematiX doesn't use this special vocabulary.

### *Cancelling Verbalize with Escape*

That's enough close work for now. If you're still using Screen Review, press the <Esc> key to exit. You can continue to Verbalize the rest of the chapter on your own, or you can stop verbalizing early. All you do to cancel Verbalize is press the <Esc> key. If Verbalize has paused for a full screen, you may hear the next word before you return to the Math Menu prompt.

### ■ Now You Know ...

Verbalizing a chapter lets you know how MathematiX interprets your braille material:

- Literary material is spoken as words.
- Math material is spelled out sign for sign and space for space.
- Format indicators and commands are spoken but not interpreted.

You're ready to make an inkprint version of the **THREE Q#** chapter.

## 6 MAKING INKPRINT HARD-COPY

Tixing inkprint is very similar to verbalizing: you use option M - Math Output for both. The only difference is you specify the **I** destination instead of the **V** destination. Before you begin, load the paper into your dot-matrix printer and set the top of form. Make sure that the printer is "on-line," ready to receive data from the Apple. With the MathematiX Menu Disk in drive 1 and the MathematiX Sample Data disk in drive 2, follow these prompts:

```
Math Menu: M
Math output
Drive or chapter: HELLO# <CR>
Drive or chapter: <CR>
Tix where: I <CR>
```

If your BEX configuration doesn't include a large print printer definition, MathematiX will complain with an error message at this point. Please read Section 1, Part 4 to find out how to make a MathematiX-compatible configuration.

The printer won't start immediately—MathematiX has to go through the same tixing process as when you verbalize. MathematiX reads software from disk, makes the tix scratch, reads more software from disk, and then starts sending graphics to your printer. Once MathematiX has tixed all the data in BEX page 1, it goes through the same process for BEX page 2. The final result is three inkprint pages long, because we began samples 2 and 3 with the { **$f** } form-feed indicator.

## 7 USING VERBALIZE TO CATCH ERRORS

You have just seen how the MathematiX process goes when everything's perfect. One of the big advantages of using a computer is that you don't have to be perfect the first time: you can find and correct your errors. A tix-ready chapter must have a particular structure or MathematiX won't create the results you want.

The **WRONG#** chapter on the MathematiX Sample Data disk is an incorrect version of Sample 1 from the **THREE Q#** chapter. We've introduced six small errors that result in some interesting mayhem. See what happens when you do this:

```
Math Menu: M
Math output
Drive or chapter: WRONG# <CR>
Drive or chapter: <CR>
Tix where: I <CR>
```

Some of the errors we introduced in **WRONG#** are so blatant that MathematiX can't create legible inkprint. That's why you hear the **Please use Verbalize** error message after the tix scratch. Verbalize provides greater detail about the *structure errors* that triggered this error message. Verbalize also helps you confirm that MathematiX can correctly interpret what you have written.

### ■ Step-by-Step through WRONG#

As we analyze what verbalize says, you will learn how to recognize the most common kinds of errors. Then we will step you through finding and fixing them. Put on your thinking cap, and follow along:

```
Math Menu: M
Math output
Drive or chapter: WRONG# <CR>
Drive or chapter: <CR>
Tix where: V <CR>
```

The first few format commands sound reasonable enough, but things quickly degenerate. Instead of hearing **Math Review for IMAT** as words, the information is spelled out letter for letter and space for space. That's a sure sign that you are in math mode. As we have stressed, MathematiX can't cope with literary material when it's in math mode. The first error in this **WRONG#** chapter is that we deleted the initial `{ @1 }` that turns on literary mode.

### *Misplaced Modes Can Cause Structure Errors*

Very quickly, you hear **ERROR: fraction not finished in chapter WRONG# page 1 paragraph 0**, all spoken in an emphatic high pitch. This is one of MathematiX's *structure error* messages. MathematiX knows that fractions follow a pattern: they start with {**?**}, have a fraction line {**/**} in the middle, and end with {**#**}. Because MathematiX is in math mode, it reads the literary "th" contraction from the word *Math* as "start fraction." When MathematiX reaches the end of the paragraph, it checks to see if the fraction has been completed. Since the heading doesn't happen to use the "st" or "ble" contractions, MathematiX can't find all the elements in a well-structured fraction, and emits the error description **Fraction not finished**.

Verbalize provides location information to help you track down errors. MathematiX uses paragraph { `$p` } indicators to check for structural problems in chunks and it reports errors with paragraph numbers. This problem showed up before the first paragraph indicator in the BEX page, so it's located as **Chapter WRONG# page 1 paragraph 0**.

After the first paragraph indicator, that missing literary translation tool continues to influence Verbalize's vocabulary. The sample number followed by a literary colon gets mangled to **a subscript 3 baseline space**. When MathematiX finally encounters another literary translation tool (it's after the two-thirds fraction halfway down this screen) then literary material is pronounced appropriately.

### *Structure Errors are Reported in Chunks*

The next problem pops up right at the end of the first screen. You hear the same error description, "fraction not finished," but this one refers to paragraph 2. MathematiX complains because we removed one of the {`#`} end fraction indicators in the problem statement paragraph. MathematiX uses natural divisions to break your material into "chunks," and checks for structural integrity at the *end* of the chunk. In addition to paragraph { `$p` } indicators, MathematiX recognizes hard <CR>s, new-line { `$l` } indicators, new-page { `$f` } indicators, and the end of each BEX page as "chunk" boundaries. The upshot is that some structure errors are only reported after the next "chunk."

### *Each Character Contributes to Nemeth Meaning*

The next problem shows up on the second screen. You hear that fraction composed of words as **start fraction h o u r s fraction line p o s t end fraction 1 space**. What's that digit 1? It should be a comma. This is the third error in **WRONG#**: we deleted the dots 4-5-6 punctuation indicator that MathematiX requires to recognize dot 2 as a comma. As you verbalize material, keep your ears peeled for digits that pop up where punctuation should be. Section 3, Part 2 and the MathematiX Reference Card list the eleven cases where you must use the punctuation indicator.

The bottom half of the second screen is devoted to a reduction of four fractions with equals signs between them. Lines P and Q show the result of {`?9/6#+?8/6#.k`}. You get the Greek letter **kappa** instead of **equals**. This is the fourth error we introduced in this chapter: omitting the space on both sides of this { `.k` } equals sign. Without the initial and final space MathematiX sees dot 4-6, dots 1-3 as lowercase Greek kappa. MathematiX also requires spaces before and after the greater-than and less-than signs.

On to the third and final screen of the **WRONG#** chapter! The first few lines are fine, but then you get a "new fraction finished, but ..." error. When MathematiX sees the start of a fraction, it immediately checks to make sure that you have finished the previous one. This error popped up because we omitted an end fraction indicator in the middle of this paragraph.

Finally, things get strange again after the words **cross space 0 point 3 5 3 space**. When you verbalized the correct version in Part 5, there sure weren't any subscripts or integrals in Sample 1. **6 f subscript 9 base line d space integral** is MathematiX trying to interpret the cells {**6f9d  !**} as math braille. Thanks to your human intelligence, it's obvious that those cells are representing the literary braille words *to find the*. Remember, MathematiX is only a software program. As you've probably guessed by now, the sixth error is a missing literary translation tool after the {**#60 @* 0.353**}.

### Now You Know ...

You've seen how Verbalize gives you a MathematiX perspective on exactly what's in a tix-ready chapter.

- The difference between "conversational" and "picky" Verbalize styles can identify misplaced tix translation tools.
- Structure errors can be caused by misplaced tix translation tools as well as truly incorrect mathematical structures.
- Some data entry errors can result in legitimate signs: a missing space turns an equals into a kappa.

Let's clean up the problems in this chapter.

## 8 FINDING AND CORRECTING THE ERRORS

In this Part, we demonstrate using BEX's Editor to correct the six errors we found in Part 7:

- At the very start of the **WRONG#** chapter, there's a literary translation tool missing.
- After the second paragraph indicator, there's a fraction without an end fraction indicator.
- After the third paragraph indicator, there's a comma without a punctuation indicator following the "wordy" fraction.
- Again in this third paragraph, there's an equal sign that needs a space before or after it.
- In that same paragraph, there's a fraction that doesn't have its end fraction indicator.
- Finally, near the very end of the chapter there's another literary translation tool missing before the words *to find the*.

Use option C - Copy chapters on the Math Menu to copy the **WRONG#** chapter to the name **RIGHT#** on the MathematiX Sample Data disk. (This preserves the errors in **WRONG#** in case you or someone else wants to repeat this tutorial.) Edit the **RIGHT#** chapter. If you're comfortable with screen braille, you can make your changes by typing on the full Apple keyboard.

### *Setting Braille Keyboard in the Editor*

When you're still learning the screen braille equivalents, you may find it easier to use BEX's braille keyboard mode. (If you have an Apple IIgs with a detached keyboard, then you can't use braille keyboard mode.) You turn on braille keyboard by clicking down the Caps Lock, then issuing the command **control-S K B**. Now use the S-D-F and J-K-L keys as a standard six-key braille entry system. In addition to the Control key method, you can *chord* your Editor commands by depressing the space bar at the same time as the command letters. For example, to go ahead one word, press space and dots 1-2-4-5 simultaneously.

### *Tracking Down the Six Errors*

To fix the first problem, you need to insert the literary mode translation tool right at the start. Begin the insert with **control-I**, type space, dot 4, dots 1-2-3, space, then finish the insert with **control-N**.

In the second paragraph, you need to find a fraction that's missing its final {**#**}. Enter the command **control-A 2 control-P** to advance the cursor to the second paragraph indicator. With braille keyboard, you enter this as dot 1-space, dots 2-3, dots 1-2-3-4-space. Unfortunately, you can't ask BEX to locate for something that's not there, but you know that this faulty fraction *does* have the {**?**} at the start. In other words, you can find the start of a fraction by locating for space, dots 1-4-5-6.

Begin the search by entering **control-L <space> ? control-A**. The braille version is **dots 1-2-3-space space dots 1-4-5-6 dot 1-space**. You land at character 94: read the fraction with **control-G**. The result is **question mark, two, slash, three, number sign**. All signs accounted for here! Enter **control-L control-A** to repeat the locate: you arrive at character 151. When you read *this* word, you only hear **question mark, three, slash, four** and you have found the problem. Press **control-I** to insert, then type one number sign (dots 3-4-5-6), and finish the correction with **control-N**.

The remainder of the errors showed up in the third paragraph. Enter **control-A control-P** to get to its start. The missing punctuation indicator showed up after the fraction with the word *hours* in the numerator. Move to character 281 by entering **control-L ?hou control-A**. You braille the locate command with dots 1-2-3-space, 1-4-5-6, h, o, u, dot 1-space. Now press **control-G** to hear the word: it's **?hours/post#1**. The digit 1 needs dots 4-5-6 before it: press left arrow (or **control-H**) until you hear **one**. Type **control-I _ control-N** to insert the punctuation indicator.

On to the questionable equals sign. Use the **control-L .k control-A** command to locate for the meat of the symbol. This is brailled as **dots 1-2-3-space, dots 4-6, dots 1-3, dpt 1-space**. You land at position 460. Press **control-R** to go back and hear the entire word: you'll hear **point k**. Since a BEX "word" by definition has a space on both sides, you know that this equal sign is correct. Type **control-L control-A** to repeat the locate, then **control-R** again: this time, you hear **question mark 9 slash 6 number sign plus** and so forth finishing up with **point k**. Since the initial space is missing, you hear the entire fraction plus the vagrant kappa. Use right arrow (or **control-U**) to place your cursor on the period, and type **control-I <space> control-N** to insert the space.

You can use the technique described above to find the missing end fraction indicator in this paragraph: combine locating for **<space> ?** with **control-G** to check the fraction's completeness. You will find the problem fraction begins at character 529—insert dots 3-4-5-6 after the digit 6.

The last error is a missing literary translation tool, which revealed itself in Verbalize as **6 f subscript 9 base line d**. Locate for dots 2-3-5, 1-2-4, 3-5, 1-4-5 and you land at position 679. Press left arrow to move to the space before, then insert the `{ @l }` literary mode tool. You're done. Before you quit the Editor, it's a good habit to restore the keyboard mode to normal. Enter **control-S control-K control-N** then unclick the Caps Lock key. Now you can press **control-Q** to save your changes. Congratulations!

## Confirming Your Corrections

In addition to verbalizing every symbol in your tix-ready chapter, you can also ask MathematiX to verbalize selected portions of the data. The **HUSH#** chapter on the MathematiX Menu Disk contains one of the *tix verbalization tools* that control what's spoken. With the MathematiX Menu Disk in drive 1 and the MathematiX Sample Data disk in drive 2, try this out:

```
Math Menu: M
Math output
Drive or chapter: 1HUSH# <CR>
Drive or chapter: RIGHT# <CR>
Drive or chapter: <CR>
Tix where: V <CR>
```

When you faithfully followed along and made all the corrections, you'll hear the tix scratch and disk access and then return back to the Math Menu prompt. The **HUSH#** chapter contains the eight characters `{@notalk }`, which tell MathematiX to suppress speech output during Verbalize unless it encounters a structure error. Find out more about the other verbalization tools and structure error messages in Section 7.

As you're developing your MathematiX skills, it's a good idea to verbalize a corrected chapter again to make sure you have caught your errors. As you've seen, MathematiX does not understand the material in your chapter. While some problems result in explicit structure errors, other defects are verbalized without incident. When you tix to inkprint, any data that would cause a Verbalize structure error halts output entirely. But there are many problems that MathematiX won't recognize as such. For example, when you've omitted a punctuation indicator, MathematiX blithely tixes the digit instead of the punctuation. When your data contains `{?hours/post#1}` then the inkprint shows the digit 1 instead of a comma.

## Bon Voyage

Good work! You have reached the end of the MathematiX Guided Tour. With this basic understanding of MathematiX, you're ready to use the program for your own materials. Sections 3 through 8 in the MathematiX Manual provide in-depth reference material on the topics we have briefly explored here. Once you have used MathematiX for a short time, you'll want to browse through Section 9, where we have collected many hints and tips for making the most of MathematiX. Here's an overview of where to look for more details:

| **When You Want to Know** | **Check Here** |
|---|---|
| How to get out of a jam | Section 8: Troubleshooting |
| The meaning of a term used in Verbalize | Appendix A: Verbalize Vocabulary |
| The cells to use for a specific inkprint sign | Appendix C: Symbols in Alphabetical Order |
| Whether MathematiX knows how to interpret a specific Nemeth symbol | Appendix B: Symbols in Transcriber Order |
| How to move around between BEX and the Math Menu | Section 6 |
| What Screen Braille Means | BEX Appendix 1 and MathematiX Reference Card |
| How to switch between literary and math translation | Section 3, Part 2 |
| What BEX format commands to use | Section 4 for reference and Section 9 for many samples |

***We hope that you find MathematiX a useful and enjoyable tool.***

# Section 3: The Basics of Tix-Ready Chapters

# Section 3
# The Basics of Tix-Ready Chapters

In order for MathematiX to create the math output you desire, your chapters must be *tix-ready*. This Section provides an overview of what a tix-ready chapter contains. Section 5 details how to enter specific math symbols for predictable inkprint output. As MathematiX tixes, it creates the spatial math format based on time-honored rules. You can also use **$$** commands and tix format tools to control the output format—more information on this appears in Section 4.

## 1 CREATING TIX-READY CHAPTERS

There are several ways you can write your Nemeth chapters. The easiest tool for braille readers is a device with both braille input and braille display—like the tape-based or disk-based VersaBraille. The next best thing is a device that at least allows braille input, such as the Braille 'n Speak, Eureka, or the braille keyboard mode of BEX's Editor. The drawback to BEX's Editor is that the voice feedback uses *screen braille* characters: the Nemeth Code for the square root of x over y sounds like "question mark, greater-than, x, ready, slash, y, number sign." As detailed in Section 7, you can Verbalize your tix-ready chapter to proofread the content as words, not graphics. If you have a braille embosser, you can make paper braille copies periodically to help you proofread your input in a natural way.

### ■ Prevent Overflow with 2500-Character or Less BEX Pages

Because MathematiX expands your data *significantly* while it's tixing, you must limit each BEX page to no more than 2500 characters. (When you're doing very graphic-intensive material—more than three levels of hypercomplex fractions or calculus—keep each BEX page to 2000 characters or less.) In the Editor, use **control-W C** to check the character count of the current page. You can use option F - File list on the Page Menu to check the size of every BEX page in a chapter, and option A - Adjust Pages on the Second Menu to make a copy of one or more BEX chapters with different page boundaries (Learner Level Section 11). If you didn't obey these character limits, you would get a spirited audio reminder. When MathematiX doesn't have enough room to tix a BEX page, it emits the overflow shriek. Press <Esc> to cancel Math output, and then use Adjust Pages or **control-C control-P** in the Editor to ensure each BEX page has 2500 or less characters.

When you prepare your tix-ready chapters on an external device, you must get the data into BEX chapters before you can tix it. When you're working with a tape-based VersaBraille, you use option F - From VB to bring your data into BEX chapters. For all other devices, you use option I - Input through Slot. For your convenience, both of these options are available on the Math Menu. The Math Menu versions are slightly different than the Main and Second Menu counterparts. One important change in the Math Menu options is From VB and

Input through Slot create smaller BEX pages to help you avoid overflow problems. Other minor changes are described in Section 6.

### The Number Sign Naming Convention

MathematiX allows you to specify any chapter name when you choose the Math Output option. Just as BEX allows you to back-translate an inkprint chapter, resulting in incomprehensible data, you could specify an inkprint or literary braille chapter to tix, resulting in "garbage." As introduced in Section 2, we suggest that you name tix-ready chapters with the number sign # character at the end. This lets you immediately identify the chapters as tix-ready. You can enter /# <CR> at the chapter prompt to selectively scan a disk, and BEX presents a numbered list restricted to the chapters that end with the number sign character. See BEX User Level Section 4 for a full explanation of selective scanning with the slash.

## 2 HELPING MATHEMATIX UNDERSTAND NEMETH

In standard Nemeth Code, you can (within limits) intermix mathematical expressions and literary contractions. For example, the four cells { **`f>m]`** } (dots 1-2-4, 3-4-5, 1-3-4, 1-2-4-5-6) can mean either the literary word *farmer* or the mathematical expression "f times the square root of m." However, MathematiX can't discriminate between literary and mathematical Nemeth on its own. MathematiX always starts out in *math mode,* where { **`f>m]`** } gets tixed as "f times the square root of m." MathematiX assumes all material is math unless you inform it otherwise. Your tix-ready chapters must contain explicit information for MathematiX to understand the data it's processing.

### Changing Translation Modes

The *tix translation tools* are plain text characters that tell MathematiX how it should interpret the braille characters in your chapters:

{ **`@l`** } (space, dot 4, l, space) turns on literary mode; becomes one space in your final output

{_ **`@l`** } (dots 4-5-6, space, dot 4, l, space) turns on literary mode without creating space in final output

{ **`@m`** } (space, dot 4, m, space) turns on math mode; becomes one space in your final output

{_ **`@m`** } (dots 4-5-6, space, dot 4, m, space) turns on math mode without creating space in final output

These tix translation tools are slightly different from the Translator Controls explained in BEX User Level Section 9. With BEX's Grade 2 translation and Back from Grade 2, the dot 4 (at-sign) TCs are residual: they remain in the translated chapter. With Math Output, the four-character tix translation tools become one space in the final output while the five-character tools disappear entirely.

### Maintain Pure Translation Modes

It's your responsibility to keep the modes "pure": after you turn on literary mode, MathematiX won't interpret any characters as math until you turn math mode on again by entering `{ @m }` or `{_ @m }`. Each time you choose Math Output, it's as if your chapter starts with `{_ @m }`. Be sure you have a space on both sides: only then will MathematiX switch translation. In math mode, `{@l }` or `{ @l}` gets tixed as "pounds sterling;" in literary mode, it's tixed as simply "at-sign, l." For both math and literary modes, `{@m }` or `{ @m}` get tixed as "at-sign, m."

The tix translation tools are extensively demonstrated in Sections 5 and 9; here's a quick sample. To tix the inkprint for "three-eighths is smaller than three-fourths," you can enter either: `{#4 3_/8 @l is small] ?an @m #4 3_/4}` or `{#4 3_/8 is smaller than #4 3_/4}`. To tix a fraction with a numerator of 5 and a denominator of the word "hours," you can enter either: `{?5/_ @l h\rs_ @m #}` or `{?5/hours#}`.

These tools function completely differently than their Grade 2 and Back from Grade 2 counterparts. For the literary braille translator, space, dot 4, dots 3-6 space turns off translation entirely. You cannot turn off translation when tixing. You must use either the literary or math braille codes to write all your material.

### Alternate Math Mode Tools

To facilitate creating tix-ready chapters with the Grade 2 translator, MathematiX provides an alternative way to signal math translation:

`{ @- }` (space, dot 4, dots 3-6, space) turns on math mode; becomes one space in your final output

`{_ @- }` (dots 4-5-6, space, dot 4, dots 3-6, space) turns on math mode without creating space in final output

Section 9 discusses preparing tix-ready chapters with the assistance of the Grade 2 translator.

## ■ Distinguishing Between Numbers, Punctuation, and Letters

Because standard Nemeth Code allows mixing of mathematical and literary material, three braille indicators help the reader when context alone doesn't make things clear. In addition to requiring you to explicitly distinguish math and literary material, the use of the English Letter, Numeric, and Punctuation indicators in MathematiX is not standard Nemeth.

### Letter Sign: Optional in Math and Mandatory in Literary

The English letter sign plays a similar role in grade 2 and Nemeth: it identifies isolated letters that are not contractions. In MathematiX's math mode, contractions are never allowed: dots 1-3-4-5-6 is always the letter *y*, never the word *you*. Generally, the letter sign is optional for math mode; your output will be fine whether it's there or not. (See Section 5, Part 10 for the single exception, forming boldface characters.)

However, in *literary* mode, MathematiX requires the use of the English letter sign in some contexts where good grade 2 does not. (This limitation is shared with BEX's option B - Back from grade 2.) *Whenever* you want an isolated letter in your inkprint, precede it with the dots 5-6 letter sign—it won't ever cause an "extra" letter sign to be tixed. When you want to abbreviate the compass direction in the short sentence, *Go west, young man!* you must write { `@l ,g ;,w41 "y man6`}. If you used { `@l ,g ,w41 "y man6`}, the result would be **Go Will., young man!**.

### *Numeric Indicator:*
### *Optional in Math and Mandatory in Literary*

In standard Nemeth Code, all numbers are shown with lowered letters, except the page numbers on the title page and on each braille page. Rule 2 in the Nemeth Braille Code for Mathematics and Science Notation details the requirements for the Numeric Indicator, dots 3-4-5-6.

In math mode, always use Nemeth digits (dropped letters). MathematiX lets you be sloppy when it comes to the Numeric Indicator: you can follow the code book to the dot or you can leave it out almost entirely. (See Section 5, Part 10 for the exception, boldface digits.) Don't use Nemeth digits in literary mode. Instead, follow standard grade 2 practice: use the Numeric Indicator and the letters *a* through *j*. The three phrases: {`#4 tigers 4 tigers  @l #d tig]s @m `} are tixed as **4 tigers 4 tigers 4 tigers**.

### *Punctuation Indicator: Mandatory in Math Only*

In standard Nemeth Code, the same braille cells are used for common punctuation or digits. As detailed in Rule 6, sometimes context is enough to determine which is which; the Punctuation Indicator, dots 4-5-6, resolves ambiguity. You never use the Punctuation Indicator in literary mode for MathematiX. You must *always* use it for math material. Eleven symbols require use of the Punctuation Indicator:

Apostrophe: {`_'`}

Closing inner quotation: {`_0'`}

Closing outer quotation: {`_0`}

Colon: {`_3`}

Exclamation point: {`_6`}

Literary comma: {`_1`}

Opening inner quotation: {`_,8`}

Opening outer quotation: {`_8`}

Period: {`_4`}

Question Mark: {`_8`}

Semicolon: {`_2`}

Yes, you do enter the same two cells for question mark and closing outer quotation: MathematiX is smart enough to use context to tell the difference and tix the appropriate sign. As detailed in Section 5, Part 13, MathematiX doesn't recognize the standard {`_0`} Nemeth Code empty set symbol. You use {`@_0`} instead to distinguish empty set from closing outer quotation.

# Section 4: Formatting Math Output

# Section 4
# Formatting Math Output

Math Output is a unique hybrid of translation and format. MathematiX is largely in control of how many characters appear on each output page—details in Part 1. As stressed in Part 2, where you put spaces in your tix-ready chapter dramatically influences whether MathematiX can tix at all. Like BEX, you can control tix format through **$$** commands and four-character format indicators like `{ $p }`—see Part 3.

In many situations, the inkprint format is controlled through your choice of standard Nemeth symbols. When you enter `{#100_/900}` Math Output creates a linear fraction, while `{?100/900#}` results in a spatial fraction, with an appropriate length fraction bar centered vertically between the numerator and denominator. For samples of how you enter particular symbols in your tix-ready chapter, consult Section 5 or the alphabetical list of symbols in Appendix C. In this Section, Part 4 discusses the few special cases where you create correct inkprint format with non-standard Nemeth. Part 5 explains the *tix vertical tools* that let you determine inkprint vertical spacing, if you really want to. With those exceptions, MathematiX makes most format decisions unilaterally: you can't change the size and relative spacing of the characters MathematiX creates.

## 1 THE SIZE OF TIXED INKPRINT PAGES

Regular inkprint output is a cooperative process between BEX and your printer. It's an entirely different story when MathematiX tixes inkprint. A little background on just what happens will help you understand how many characters can fit on one output page.

For regular inkprint, you define the basic page size with carriage width and form length—in your configuration, or with **$$w#** and **$$f#** commands in your chapters. Within this framework, BEX's formatter breaks your text into lines and pages, sending the printer a stream of characters plus spaces, <CR>s, and form feeds. What happens then depends largely on the printer. Important details like the depth of a single "line," the size of the characters, and the spacing between them are the printer's department. You can generally control these features through "escape sequences"—control characters, letters and numbers that instruct the printer to do something special, like double-wide characters.

### *MathematiX sends dots, not characters*

On the other hand, the Math Output option controls just about everything. Instead of sending your printer the letter *a*, for example, MathematiX tells the printer to make 13 dots that together draw an *a*. Based on the graphic capabilities of your printer, MathematiX itself determines the maximum number of characters that fit on one line horizontally: 76 for the ImageWriter and 58 for the Epson. You can narrow the text and increase the white space on the left and right with the BEX margin commands **$$ml#** and **$$mr#**, but you *can't* change the carriage width. MathematiX overrides any value you supply in your large print printer definition, and ignores any **$$w#** command in your tix-ready chapters.

MathematiX is also completely in control of form length. Like regular print, the default vertical spacing is one <CR> for hard and soft <CR>s and two <CR>s for paragraphs. However, the height of the material on each line determines the exact vertical distance between one line and the next. An output line that only contains text and linear math requires much less vertical space than an output line containing a hypercomplex fraction. MathematiX makes sure that your output's readable by adding a little extra to the tallest sign on each line. MathematiX ignores the form length value in your configuration and any **$$f#** commands because MathematiX isn't counting *lines*. The constant factor for tixed inkprint is that around 700 dots fit on each page vertically. The default single line spacing is deeper than the six lines per inch common to most regular printers. A hypothetical tix-ready chapter containing only literary material would output with 43 lines per page on both ImageWriter and Epson.

## 2 THE IMPORTANCE OF THE SPACE CHARACTER

In a regular print or literary braille chapter, the space character plays several roles. It defines a "word" so readers can understand what's meant: **therapist** and **the rapist** are two different concepts! BEX's translator creates correct grade 2 based in part on where spaces appear. When it comes time to output the text, BEX's formatter requires periodic spaces to divide the document into output lines. Finally, you must space your format commands, format indicators, and translator controls correctly so the translator and formatter can recognize them.

Spaces similarly play several roles in your tix-ready chapters. For math mode, where spaces appear is crucial for meaning. Various tix spacing tools let you modify the space as a word boundary, provide good places to break the output into lines, and differentiate between characters-as-commands and characters-as-data. Except for the special cases covered in this Part, the spaces you enter in your tix-ready chapters result in spaces on the tixed page. MathematiX does not afford you total control over the spacing of each sign on the page; remember, the goal of MathematiX is readable, *draft* mathematics.

### *Spaces Affect Nemeth Meaning*

MathematiX is unforgiving of Nemeth shortcuts: for predictable output, use spaces exactly as the Nemeth Code dictates. For example, a space following a superscript or subscript means a return to the baseline. {**x^2 + y**} means "x squared plus y." When tixed in inkprint, the "x," "plus," and "y" share a common baseline while the "2" exponent floats higher. When you don't have a space on both sides of the plus sign, {**x^2+y**} means "x raised to the 2 plus y power." The inkprint floats an unspaced "2 plus y" above the "x" at the baseline. See "Using the Sticky Space," below, for how to include an output space in a super- or subscript.

MathematiX and Nemeth depend on spaces to differentiate between several meanings of the same braille symbol. Here's just one example: Space, dot 5, k, space represents the less-than sign: {`-n "k n`} says "negative n is less than n." When you want the less-than sign, you must put spaces on both sides. {`-n "kn<:]`} means "negative n followed by n overbar." If MathematiX encountered {`-n "kn `}, it returns a structure error, thinking you'd started a modified expression with dot 5 and neglected to finish it. (Section 7 lists all the structure errors and how to recover.)

## Three Tix Space Tools

These tools let you precisely control whether a space appears in your output.

{`_ `} dot 4-5-6, space is the *disappearing space:* Treated as space for word-by-word editing on VersaBraille or in BEX. Valid as "space" for any math sequence or tix translation tool. No output space is produced when tixed; not a legal place to break the line.

{`@ `} dot 4, space is the *discretionary line break:* like a space, gives MathematiX permission to move to a new output line at that spot if needed. No output space is produced when tixed.

**<control-S>** *sticky space:* Not seen as a space in a math sequence. Always produces output space when tixed; not a legal place to break the line.

### *Using the Disappearing Space*

The {`_ `} dot 4-5-6, space facilitates creating tix-ready chapters and better inkprint output. Editing very long math expressions on the 20-cell VersaBraille display can be awkward: use the disappearing space to break them up into more manageable "words." When MathematiX encounters {`_ `} in your tix-ready chapter, it recognizes that a space is present for the purpose of translating the math. Yet when it tixes a disappearing space, no space appears in the output.

The disappearing space allows you to suppress inkprint spaces to better follow math conventions. Suppose you wish to show three variables inside parentheses, separated with commas but no spaces. You can't use {`(a,b,c)`}, because the dot 6s are seen as capitalization indicators: the tixed result would be **(aBC)**. Entering {`(a,_ b,_ c)`} does the trick. (As an alternative, you *could* use *literary* commas: {`(a_1b_1c)`} also results in **(a,b,c)**).

The dissappearing space is also useful when writing signs of comparison in upper or lower limits. When you write {`x .k 1`} then MathematiX tixes an output space on either side of the equal sign. When you write {`x_ .k_ 1`} then the "x", equals sign, and "1" are unspaced. More on this topic in Section 9, Part 5. The dissappearing space lets you tix a Hebrew aleph or bet that's immediately followed by an English letter. {`,,ax`} would result in "uppercase A, uppercase X," while {`,,a_ x`} results in "aleph, lowercase x."

The five-character tix translation tools {`_ @l `} and {`_ @m `} are a special use of the disappearing space: they signal the change between literary and math modes while producing *no* spaces in the output.

### *Using the Discretionary Line Break*

As MathematiX tixes, it uses spaces to divide your material into output lines. The program knows that some spaces are not good places to break the line: it won't move to a new output line if a space is inside a fraction, square root, or modified expression. If you entered very long expressions without any spaces, then MathematiX wouldn't know where to divide the output lines. In the grip of this dilemma, MathematiX can respond in two ways. You may get overprinted and illegible output, or the program may crash. In the latter case, you should press control-Reset to get the BASIC prompt. Then type **RUN START <CR>** (in all caps) and fix up the chapter.

To ensure pleasant sailing, you must provide MathematiX with places to break the line. Either type a space every 50 or so inkprint signs, or use the discretionary line break, {`@ `} dot 4, space. This gives MathematiX permission to break a line at that spot if needed. When MathematiX doesn't need to break the line, then the dot 4 space is suppressed and no space appears in the output.

BEX supports a different discretionary line break system: after you enter the format command **$$sd** to enable it, BEX recognizes the <ASCII 30> control character as an acceptable place to break a line. MathematiX does *not* recognize <ASCII 30> as a discretionary line break.

### *Using the Sticky Space*

As introduced in BEX, the sticky space is a "hard" or non-breaking space. In terms of translation, MathematiX does *not* consider **<control-S>** a "space." The sticky space allows you to include output spaces in subscripts or superscripts. While a real space means "return to baseline," the **<control-S>** doesn't. When MathematiX encounters a sticky space in your tix-ready chapter, it always creates an output space but won't break the line at that spot.

These samples demonstrate the difference between a sticky space and a regular space:

| MathematiX entry | Inkprint result |
|---|---|
| {`x^2 + y`} | x squared plus y |
| {`x^2+y`} | x raised to the 2 plus y power, no spaces in inkprint "2 plus y" |
| {`x^2<control-S>+<control-S>y`} | x raised to the 2 plus y power, inkprint appears as 2, space, plus, space, y. |

However, those are nonsensical samples. A more realistic application of the sticky space is for function names in subscripts or superscripts. The sticky space allows you to write "y to the sin x power:" {`y^sin<control-s>x`}. (Since MathematiX *won't* allow you to transfer control characters from a tape-based VersaBraille, you can also repeat the superscript indicator, entering {`y^sin ^x`} for the same result.)

### The Null Tool Resolves Ambiguity

There are four Nemeth symbols that could be confused with format indicators or tix translation tools. When you want the inkprint sign tixed with spaces on both sides, you must use the tix null tool, { `"]` } dot 5, dots 1-2-4-5-6 to interrupt the command sequence:

{ `$l"]` } parallel to (instead of a new-line indicator)

{ `$p"]` } perpendicular to (instead of a paragraph indicator)

{ `@l"]` } pounds Sterling (instead of turn on literary mode)

{ `.k"]` } kappa with spaces on both sides (instead of equals sign)

When you write the perpendicular to symbol without spaces on either side, then MathematiX knows you mean perpendicular to, and tixes the correct sign. But if you want spaces on both sides of the inkprint sign, you must use the tix null tool to tell MathematiX that you don't want a new paragraph. You *can't* put the null tool between the dots 1-2-4-5-6 and the *p*, because MathematiX wouldn't see the "perpendicular to" symbol at all.

## 3 FORMAT COMMANDS SUPPORTED BY MATH OUTPUT

When you tix to the screen or inkprint, option M - Math Output can interpret some $$ format commands in your chapters. Because of the complexity of inkprint mathematical output, it was not possible to implement every one of the $$ commands supported by option P - Print chapters on the Main or Math Menus. This Part examines four groups of commands:

- New $$ commands that *only* work for Math Output.
- Modified $$ commands that work a little differently for Math Output than for Print.
- Identical $$ commands that function the same with both Math Output and Print.
- BEX-only $$ commands that won't work with Math Output at all.

### New MathematiX $$ Commands

These commands *only* work when you tix with Math Output. If you use option P - Print on the Math or Main Menus with a tix-ready chapter containing these commands, the command letters appear in your output.

$$o# overprint # times for each line of output. The more overprinting you specify, the darker your output—and the more time required for final printing. The default is **$$o1** (one printing pass). Beyond two or three passes, you begin to punch holes in the paper.

**$$kb** and **$$kf** keep together, begin and keep together, finish. These commands work as a pair to prevent MathematiX from breaking a long math expression into two output lines. When the material between the **$$kb** and **$$kf** commands won't fit in the current output line, MathematiX moves to a new line and tixes it there. If the material between the **$$kb** and **$$kf** commands is too long to fit on *any* single output line, then MathematiX crashes. Press control-Reset to get the BASIC prompt. Then type **RUN START** <CR> (in all caps) and fix up the chapter, inserting spaces or discretionary line breaks to enable MathematiX to tix correctly.

## Modified $$ Commands

For each command, we state how it works in MathematiX, and then the differences from Print.

**$$c** Center one line of text, from previous <CR>, `{ $l }`, or `{ $p }` to next soft <CR>. Default carriage width is 76 for ImageWriter and 58 for Epson. For Print, you have multi-line, "smart" centering: your text continues to center until the next hard <CR>, `{ $l }`, or `{ $p }`.

**$$h** In literary mode, center and underline one line of text from previous <CR>, `{ $l }`, or `{ $p }` to next soft <CR>. In math mode, the line of text is not underlined, only centered. For Print, you have multi-line headings that are both centered and underlined.

**$$ub** begin underlining if you're in literary mode. When you're in math mode, this command is ignored for inkprint although it's verbalized. For Print, this always underlines.

**$$uf** finish underlining if you're in literary mode. When you're in math mode, this command is ignored for inkprint although it's verbalized. For Print, this always finishes underlining.

**<CR>$$vh1 [running header text] $p** For each output page, repeat the "running header text" on line 1 and skip line 2. Use other **$$** commands to position the header horizontally; the "running header text" *can* contain math material. With Print, you skip line 2 with the **$$vs2** command, and it doesn't matter whether you end the header with <CR>, `{ $l }`, or `{ $p }`.

**<CR>$$vh1 [running header text]<CR>** For each output page, repeat the "running header text" on line 1 but don't skip line 2.

**<control-S>** Sticky space token: always make a space here in the output and never break the line at this spot. With Print, you must also enter **$$ss** in your chapter to enable the use of <control-S>; sticky spaces are *always* enabled with MathematiX.

## Identical $$ Commands

Consult BEX User Level Section 7 for details on how these commands work.

`{ $p }` begin new paragraph (space, dots 1-2-4-6, p, space). Number of spaces to indent set by combination of **$$i#** and **$$ml#**; number of <CR>s set by **$$s#**; default is **$$ml0 $$i5 $$s2**.

**$$s#** establish number of <CR>s for paragraph { **$p** } indicators

**$$i#, $$i-#, $$i+#** establish or adjust indent for paragraph { **$p** } indicators, relative to left margin.

{ **$l** } or **<CR>** begin new output line (space, dots 1-2-4-6, l, space) at current left margin. Default is single spacing, **$$l1**

**$$l#** establish line spacing for new-line { **$l** } indicators and hard or soft <CR>s.

**$$ml#, $$ml+#, $$ml-#** establish left margin numerically; default **$$ml0**

**$$ml*** establish left margin at current position on the output line

**$$mr#, $$mr+#, $$mr-#** establish or adjust right margin numerically; default **$$mr0**

{ **$f** } begin new output page (space, dots 1-2-4-6, f, space).

**$$np** number pages with word *Page* centered at bottom

**$$n#** set the next output page number to #

**<Del>** (delete character); page number token in running header

**$$p#** place the following text at # position horizontally on current output line

**$$p+#** or **$$p-#** place subsequent text # characters to left or right of current position on output line

**$$tc** clear all tabs

**$$t#** establish tab position numerically

**$$t*, $$t-#, $$t+#** establish tab based on current position on output line

**$$d** reset Math Output to default: no page numbering, no header, **$$ml0 $$mr0 $$l1 $$s2 $$i5**

## ■ BEX-Only $$ Commands That Don't Work in MathematiX

If you include these commands in your tix-ready chapter, they won't do a thing. They are verbalized as "not supported." When you tix to inkprint or the screen, these commands are completely suppressed.

### *Unsupported commands controlling page format*

**$$f#** set form length to # lines per page.

**$$w#** set carriage width to # characters per line.

**$$mt#** set top margin to # lines.

**$$vf [text]<CR>** running footer of "text" on bottom line of each output page.

**$$vh2, $$vh3,** etc running head on line 2, 3, etc. You can only have a running head on line 1 with MathematiX.

**$$vs#** skip line # on each output page.

**$$vo#** restore line # to text, clearing a header or skip command.

### *Unsupported commands controlling line and page breaks*

**<control-T>** "touching token" to replace initial or final space in BEX **$$** commands.

**<ASCII 30>** Discretionary line break: if needed, place a soft <CR> here. Use `{@ }` dot 4, space instead.

**<ASCII 31>** Discretionary hyphen: if needed, print a hyphen and place a soft <CR> here.

**$$sd** Enable use of <ASCII 30> and <ASCII 31> as discretionary characters.

**$$a#** advance to line # on the output page.

**$$vn** go to a new page unless that would make a blank page.

**$$vl#** move to a new page when fewer than # lines remain on current output page.

**$$vi#** finish current line, then move to a new page when fewer than # lines remain on current output page.

### *Miscellaneous unsupported commands*

**$$r** Flush right text from previous `{ $p }`, `{ $l }`, or <CR> to next `{ $p }`, `{ $l }`, or <CR>.

**$$b#** or **$$b** stop and beep # times

**$$eX** "escape code" commands for specific printer features, like boldface, superscript, etc.

**$$sp** suppress underlining of trailing punctuation.

**$$su** force uppercase only output.

**$$vg** switch to Roman numeral page numbering.

**$$vk** cram a BEX word against the right margin.

**$$vrX** fill the current output line by repeating any character X.

**$$z** zap formatter; output all **$$** commands as they appear in the Editor.

## 4 FORMAT CREATED THROUGH NON-STANDARD NEMETH SYMBOLS

In some situations, MathematiX requires non-standard Nemeth entry in order to create correct inkprint output.

### ■ Superimposing symbols

While Nemeth provides various ways to show two inkprint signs are on top of each other, MathematiX takes a different approach. You use the general pattern of multipurpose indicator, first symbol, termination indicator, second symbol. For example, to get a plus sign superimposed on a circle, enter `{"$c]+}`. Once you understand how MathematiX interprets this pattern, you can use it to superimpose *any* symbols.

The multipurpose indicator `{"}` tells MathematiX to save its current output location. The termination indicator tells MathematiX to restore that location. When MathematiX tixes the sequence `{"$c]+}`, it goes through this reasoning: "OK, I'll save my position, print a circle, restore the remembered position, print a plus, and then get ready to print the next sign." The result is a plus printed on top of a circle.

## Enlarged Symbols of Enclosure

MathematiX lets you specify "regular-sized" and "enlarged" symbols of enclosure, such as parentheses, brackets, braces, and vertical bars. When the expression you're enclosing includes a spatial fraction or other material with more than two vertical levels, your inkprint will be easier to read if you use the non-standard enlarged versions.

1.a `{,,\,??24/8#+2,/?64/8#+2,#,,\ .k ,\?3+2/8+2#,\}`

The absolute value of the complex fraction having the quantity twenty-four eighths plus two in the numerator and the quantity sixty-four eighths plus two in the denominator is equal to the absolute value of the fraction with a numerator of three plus two and a denominator of eight plus two.

1.b `{,\,??24/8#+2,/?64/8#+2,#,\ .k ,\?3+2/8+2#,\}`

Same as 1.a but standard Nemeth.

2.a `{.,(?1/2#, ?1/3#, '''.,)}`

The set containing one-half, one-third, and so forth.

2.b `{.(?1/2#, ?1/3#, '''.)}`

Same as 2.a but standard Nemeth

3.a `{?1/3#,(?>x+y]/2#,)}`

One-third times the quantity the square root of x plus y over 2.

3.b `{?1/3#(?>x+y]/2#)}`

Same as 3.a but standard Nemeth

These enlarged symbols are twice as high as a standard MathematiX character; they straddle the reference baseline. Half of their "extra" height is placed below the baseline, and half goes higher than the top-of-caps line. More details in Section 9, Part 6.

## Adjusting Droopy Numerators in Fractions

While MathematiX does a good job of making draft quality output, it ocassionally makes some poor choices. In Section 9, Part 6, we detail exactly how MathematiX places characters on the page. Here we just discuss a "cookbook" solution to the most common problem. When tixing spatial fractions, MathematiX automatically "drops down" the denominator to ensure legibility—when a denominator includes a radical, for example, there's adequate space between the horizontal fraction bar and the top of the radical sign. MathematiX doesn't handle the placement of numerators as elegantly. For example, when {**?x2/7#**} is tixed, the bottom stroke of the subscript digit 2 is superimposed on the horizontal fraction line.

The simple way to solve this problem is to make your fraction more complex! When tixing inkprint, MathematiX puts more vertical space in a complex fraction than a simple fraction. When you have something tall in a numerator of a simple fraction, you'll get better results if you write it as a complex fraction. The subscript in {**,?x2,/7,#**} is perfectly legible. When your numerator with a subscript is already in a hypercomplex fraction, then you can control the vertical spacing manually with a tix vertical tool—details below.

### ■ When Inkprint Uses Enlarged Operators by Convention

Larger inkprint versions of two Greek letters have conventional meanings in higher math. While a regular-sized sigma is just a sigma, the sigma that shows summation is twice as high. Similarly, a larger pi represents a product. In order for MathematiX to make the proper-size inkprint Greek letters, use {**.,,s**} for the summation and {**.,,p**} for the product. If you neglect to request these enlarged operators, the print reader won't recognize the sigma and pi as operators.

#### *Positioning Limits Above and Below*

While Nemeth Code simply says, "this stuff is above, this stuff is below," the traditional inkprint presentation *centers* the limits below and above the central operator—an enlarged sigma or pi, or a standard intersection or union. MathematiX can't center the limits automatically. If both your lower and upper limits are one or two characters (inkprint signs) long, then you can't control centering anyway. But when either limit is three signs or longer, you must add spaces *inside* the modified expression.

MathematiX outputs these spaces, centering the elements by brute force. Each space after the above symbol, dots 1-2-6, nudges the upper limit to the right, while each space after the below symbol, dots 1-4-6, nudges the lower limit to the right. Each space you put after the initial dot 5 nudges the central operator one character to the right. Several examples of these techniques appear in Section 9, Part 5.

## 5 PRECISE CONTROL OF VERTICAL SPACING

The *tix vertical tools* let you tell MathematiX exactly how to space your inkprint output vertically. Each MathematiX inkprint "character" is built up from individual dots on the screen or your printer. (These dots are called "pixels" when refering to the screen—we'll use this term to avoid confusion with the dots in the braille cell.) The basic size of a MathematiX character is 7 pixels wide by 8 pixels high. You can't control horizontal spacing at the pixel level: use BEX **$$** commands to move horizontally in character-chunk (7 pixel) increments. (As detailed in Section 9, Part 6, some "tall" inkprint signs are composed of several "characters;" the enlarged sigma { **.,,s** } for example, is two characters stacked vertically.)

### *Tix Vertical Tool Syntax*

Each tool is composed of three cells: Dots 2-4-6 { **[** }, followed by a direction shown with above {**>**} or below {**%**}, followed by a lowercase letter code for distance moved. (A list of the fourteen valid combinations appears on the MathematiX Reference Card.) The code letters are:

| Code letter | Mnemonic | Pixels moved |
|---|---|---|
| {**q**} | quarter character | two |
| {**h**} | half character | four |
| {**o**} | one character | eight |
| {**x**} | (one-and-a-half) | twelve |
| {**d**} | double character | sixteen |
| {**t**} | three characters | twenty-four |
| {**f**} | four characters | thirty-two |

For example, { **[%o**} means "move down one character (eight pixels)" while { **[<q**} means "move up a quarter of a character (two pixels)." You can get intermediate values by entering several tools in a row: { **[<q[<h**} moves up three-quarters of a character or six pixels.

## Using the Tix Vertical Tools

Where you use the tix vertical tools determines the duration of the change in vertical position. You can enter a tix vertical tool immediately after one of the five Nemeth indicators that imply vertical movement: opening simple fraction indicator {**?**}, opening complex fraction indicator {**,?**}, opening hypercomplex fraction indicator {**,,?**}, directly-over indicator {**<**}, or directly-under indicator {**%**}. At this point, MathematiX is about to make its own vertical spacing decision; it sees the tix vertical tool as modifying the vertical value. When MathematiX encounters a tix vertical tool after one of these five indicators, the change in vertical position only affects the following element.

Alternatively, you can take complete responsibility for the duration of the tix vertical tool's effects. Recalling the discussion of superimposition in Part 4, MathematiX interprets a dot 5 { **"** } as a command to remember the current position on the page, and { **]** } dots 1-2-4-5-6 as a command to restore the current position. (Although the same symbols are used for modified expressions, MathematiX *doesn't* alter the spacing in that case.) Combining the tix vertical tools with superimposition, you can save your current position, move up or down a specified amount, print something, and then get back to your starting location.

Before we get down to samples, there are several important cautions. Remember that MathematiX is a tool for draft mathematics and *not* a typesetting program. In all cases, MathematiX attempts to space the inkprint appropriately. When you use the tix vertical tools, you are in effect overriding MathematiX's automatic spacing decisions, which requires you to a) know what inkprint math *should* look like and b) think like a software program. As with **$$** commands, Verbalize doesn't provide feedback on the format created by the tix vertical tools: a sharp-eyed sighted assistant is a necessary peripheral. A brailler graphics program like RDC's pixCELLS can produce tactile versions of MathematiX's screen display to help interpret the tix vertical tool's effects—see Section 9, Part 4 for how to do this.

### *Adjusting Droopy Numerators with Tix Vertical Tools*

As mentioned earlier, when a numerator in a spatial fraction drops below the normal baseline, the inkprint result can be illegible. Don't attempt to use tix vertical tools to adjust a denominator, because MathematiX will override your adjustments.

You can intervene with a tix vertical tool when the numerator of a fraction goes further than a quarter of a character under the baseline. A subscript moves down half a character. Moving the numerator up by four pixels yields a more legible result: use {`?[<hx2/7#`} instead of {`?x2/7#`}. If the numerator of a fraction contains a summation sign with material above and below it, you need to raise the numerator by one and a half characters:
{`?[<x"  .,,s%i_  .k_  1<  ,=]  i^-5"/4#`}.

The prohibitior against denominator-fiddling doesn't rule out numerators that are in the denominator of a complex fraction. With
{`,?[<h?1/x2#,/?[<hy2/567#,#`}, for example, the subscript digit 2s are much more legible. Notice that the tix vertical tools here are *not* "superimposed" within a dot 5, dots 1-2-4-5-6 pair. That's because the tool immediately follows the opening fraction indicator. When MathematiX encounters the tool, it only changes the vertical position of the numerator.

### *Creating New Symbols with Tix Vertical Tools*

By superimposing shifted signs, you can generate an almost unlimited number of shapes and scribbles. The superimposition syntax is "save position, base symbol, restore position, overprinted symbol." To shift *and* superimpose, this becomes: "save position, tix vertical tool, base symbol, restore position, overprinted symbol." In chess notation, for example, the bishop is represented with a stylized mitre. A recognizable version of the bishop sign can be created with
{`"[<q$p]$c`}. MathematiX tixes this as: save position, move up 2 pixels, draw a perpendicular to sign, restore position, draw a circle shape. The dot 5 save and dots 1-2-4-5-6 restore bracketing the tix vertical tool are crucial: if you omitted them, then you'd shift the baseline of all subsequent text. Other examples of the tix vertical tools appear in Section 9, Parts 6 and 7.

# Section 5:Entering Your Math Symbols in Tix-Ready Chapters

# Section 5
# Entering Your Math Symbols in Tix-Ready Chapters

This reference Section gives general guidelines and samples for how you enter the math part of your document, beginning with symbols used in elementary school and working up to calculus, chemistry, and post-secondary topics. While we've tried to illustrate all the common symbols, these samples are not exhaustive. When you need to find a particular symbol, check the transcriber order list in Appendix B or the alphabetical order list in Appendix C.

## 1 GENERAL PRINCIPLES

We've tried to make MathematiX understand Nemeth Code, but Nemeth wasn't designed as an inkprint typesetting system. Most importantly, MathematiX requires you to use `{ @l }` and `{ @m }` to explicitly show the boundaries between *literary* and *math* braille. MathematiX requirements for the Numeric, English Letter, and Punctuation Indicators are neither exactly standard Nemeth nor standard grade 2. These deviations from standard Nemeth are constant no matter what math symbols you're entering in your tix-ready chapters—see Section 3, Part 2 for a complete explanation of what's required.

In this Section, we draw your attention to the *other* instances where you must enter non-standard Nemeth to get correct inkprint. The exact placement of spaces in your tix-ready chapter is very important: so crucial, in fact, that we devote an entire Part to it—see Section 4, Part 2. In the many samples that follow, take careful note of where spaces do and don't appear.

With those exceptions, in general, the more perfect your Nemeth Code, the better the MathematiX result. Many Nemeth users develop their own shorthand for personal materials, but this will only create structure errors in MathematiX. As you're learning MathematiX, take frequent advantage of the Verbalize feature (Section 7) to discover where MathematiX would choke on shorthand.

## 2 BASIC MATHEMATICAL SYMBOLS

### Conforms to standard Nemeth Code

1. `{#1 "1 #2 ;2 #3 "1 #6}`

   1 is to 2 as 3 is to 6

2. `{ (10+2)./3 .k #4}`

   The quantity 10 plus 2 divided by 3 equals 4

3. `{#7@0 of @s3.00 is #21@c}`

   Seven percent of 3 dollars is 21 cents

4. `{#3@*(3_/4) and #3*(3./4) @l >e two ways 6write ! same expres.n4}`

   Three cross the quantity 3 over 4 and 3 dot the quantity 3 divided by 4 are two ways to write the same expression

5. `{#5 - #2 "k #4_1 but #5 + #2 .1 #4_4}`

   Five minus 2 is less than 4, but 5 plus 2 is greater than 4

Since sample 3 is all in math mode, the *of* contraction isn't used. Sample 4 doesn't use the *and* contraction; literary text only appears after the `{ @l }` translation tool. This also shows uses of `{_/}` to write the slash in an *in-line* fraction; Part 3 explains writing spatial fractions. Notice the use of the `{_}` Punctuation Indicator for the comma and period in sample 5; MathematiX always requires its use before ten common punctuation characters in math mode.

Nemeth and MathematiX both require spaces on either side of the equals, greater than, and less than signs of comparison. In some inkprint situations, you may wish these three signs to appear unspaced: use the disappearing space `{_ }` to suppress the output space—details in Section 4, Part 2. Unlike standard Nemeth Code, MathematiX recognizes any other sign of comparison even if it's not preceded and followed by a space.

## 3 SIMPLE SPATIAL FRACTIONS

### Non-standard Nemeth for In-Line Mixed Numbers

1. `{ @l ,: is grt]3 @m ?3/4# or ?5/6#_8}`

   Which is greater: three over four or five over six?

2.a `{ @l ,a c>'s spe$ is f.d ) @m ?miles/hours#_4}`

   A car's speed is found with miles over hours

2.b `{ @l ,a c>'s spe$ is f.d ) @m ?miles/_ @l h\rs_ @m #_4}`

   A car's speed is found with miles over hours

3. `{#4_?3/8_# @l is small] ?an @m #4_?3/4_#}`

   Four and three-eighths is smaller than four and three-fourths

4. `{#4 3_/8 is smaller than #4 3_/4}`

   Four and three-eighths is smaller than four and three-fourths

You don't use the Numeric Indicator within a spatial fraction, since `{#}` is the closing simple-fraction indicator. Notice the Punctuation Indicator before the final question mark in sample 1. Sample 2 shows the two ways MathematiX can cope with literary braille appearing within a math expression. 2.a keeps to "pure" math mode: the contraction is not taken in the denominator. 2.b encloses the contracted word *hours* within the five-character disappearing tix translation tools.

Sample 3 uses the standard Nemeth Code for an in-line mixed fraction but MathematiX tixes this spatially, with the three-eighths and three-fourths written vertically. To get in-line mixed fraction output, you must use the non-standard Nemeth shown in sample 4.

## 4 SUPERSCRIPTS AND SUBSCRIPTS

### Non-Standard for Fractions and Functions not at Baseline

1. `{x^2 + y^2 .k z^2}`

   x squared plus y squared equals z squared

2. `{x1+x2 .k y"1-y"2}`

   x subscript 1 plus x subscript 2 equals y1 minus y2

3. `{e^x^^2 "k e^x^;2}`

   e raised to the x squared power is less than e raised to the x subscript 2 power

4. `{q^e^^5+n^^;1}`

   q raised to the e to the quantity 5 plus n sub 1 power

5. `{x;n-1, x;n[1}`

   x sub n minus 1, x sub n comma 1

6. `{?x^2"/y^2"#}`

   x squared over y squared as a spatial fraction

7. `{?x^1_/2"/2#}`

   A spatial fraction with a numerator of x raised to the one-half power and a denominator of 2

8.a `{y^sin ^x}`

y raised to the sin x power

8.b `{y^sin<control-s>x}`

y raised to the sin x power

As seen in sample 2, Nemeth and MathematiX interpret a number immediately following a letter as a subscript. To write a number that immediately follows a letter at the baseline, use the dot 5 multipurpose indicator, { " }, as shown. Nemeth Code and MathematiX support 14 superscript and subscript levels. Sample 4 demonstrates that you can combine up to three superscript { ^ } and subscript { ; } symbols. Sample 5 shows the { [ } used for a comma within a subscript or superscript.

Samples 7 and 8 show the non-standard Nemeth that MathematiX requires for some superscripts and subscripts. In a spatial fraction, any fractional superscript or subscript must be linear (or decimal). If you used `{?x^?1/2#"/2#}` instead of sample 7, MathematiX generates a structure error.

In standard Nemeth, the space following a function name in a superscript or subscript doesn't return you to the baseline. MathematiX *always* sees a space or dot 5 `{"}` as the return to the baseline from a superscript or subscript. You have two ways to correctly write "y to the sin x power." You can repeat the superscript symbol after the space, as shown in sample 8.a. Alternatively, follow sample 8.b and use the <control-S> sticky space between the "sin" and the "x". <Control-S> always creates a space in the output, but it's not interpreted as a space by MathematiX as it tixes—more on this topic in Section 4, Part 2.

## 5 COMPLEX AND HYPERCOMPLEX FRACTIONS

### Conforms to standard Nemeth Code

1. `{,??24/8#+2,/?64/8#+2,# .k ?3+2/8+2# .k ?5/10# .k ?1/2#}`

   Four fractions demonstrating reduction: the first, complex fraction has a numerator of the simple fraction 24 over 8, plus 2; the denominator is the simple fraction 64 over 8, plus 2. This is equal to the second, simple fraction with a numerator of 3 plus 2 and a denominator of 8 plus 2. This is equal to the third, simple fraction with a numerator of 5 and a denominator of 10, which is finally equal to the simple fraction one-half

2. `{,,?1,,/,?1,/?1/7#,#,,# .k ,?1,/?7/1#,# .k ?1/7#}`

   Three fractions showing reduction: The first, hypercomplex fraction has numerator of 1, and a denominator of the complex fraction with a numerator of 1 and a denominator of the simple fraction 1 over 7. This is equal to the second, complex fraction with a numerator of 1 and a denominator of the simple fraction 7 over 1. This is equal to the third, simple fraction 1 over 7

3. `{?x^3_/2"/7#}`

   A simple fraction with a numerator of x raised to the three-halves power and a denominator of 7.

A *complex* fraction contains a *spatial* fraction in its numerator or denominator. A *hypercomplex* fraction contains a complex fraction in its numerator or denominator. In-line fractions don't count when determining complexity or hypercomplexity. MathematiX doesn't permit spatial fractional exponents (Part 4). Since Sample 3 uses an in-line fractional exponent, it's "simple," not complex.

## 6 SQUARE ROOTS AND RADICALS

### Conforms to standard Nemeth Code

1. `{>2]}`

   The square root of 2

2. `{<3>8] .k #2}`

   The cube root of 8 is equal to 2

3. `{>20+.>25.]] .k >20+5] .k >25] .k #5}`

   The square root of the quantity 20 plus the square root of 25 is equal to the square root of the quantity 20 plus 5 is equal to the square root of 25 is equal to 5

4. `{>x^2"+y^2"]}`

   The square root of the quantity x squared plus y squared

5. `{>?x+y/2#]}`

   The square root of the entire fraction x plus y divided by 2

6. `{?>x+y]/2#}`

   A simple fraction with a numerator of the square root of the quantity x plus y and a denominator of 2.

## 7 SHAPES AND GEOMETRY

### Limited Number of Shapes and Arrows

1. `{$t ,a,b,c has $[ ,a,c .k #90^.*}`

   triangle ABC has angle AC equal to 90 degrees

2. `{$r "1 $4 ;2 $e "1 $c}`

   rectangle is to square as ellipse is to circle

3. `{>area $4 ,x] .k #5}`

   The square root of the area of square X is equal to 5

4. `{$a $s $'}`

   concave upwards arc followed by star followed by concave downward arc

5. `{$[3333}`

   A long left arrow

6. {`$<33o$1$%33o`}

   up arrow is parallel to down arrow

7. {`$<33o $p"] $o`}

   up arrow is perpendicular to right arrow

8. {`$[o$p$<[o`}

   bidirectional horizontal arrow is perpendicular to bidirectional vertical arrow

9. {`$*33o`}

   A right arrow with a solid dot on the left side

10. {`$[33.*`}

    A left arrow with a hollow dot on the right side

11. {`#0$.*33*1`}

    A line segment from zero to one, open at zero and closed at one

Nemeth Rule 16 describes a wide variety of shapes: MathematiX only supports the shapes shown here. MathematiX can't draw pentagons, rhombuses, trapezoids, specialized angles, specialized triangles, or modified shapes. For example, in standard Nemeth code, {`$q`} means quadrilateral and {`$t.i]`} means isosceles triangle: MathematiX would output a plain *q* for the former and a structure error for the latter. When you need to express these concepts, use words instead of Nemeth symbols.

Sample 6 shows an *unspaced* parallel to symbol, which MathematiX interprets correctly. If you used standard Nemeth by writing spaces on both sides of this sign of comparison, you would need to use the tix null tool to prevent MathematiX from seeing {` $1 `} as a new-line indicator. That's why the {`"]`} appears where it does in Sample 7. BEX interprets the four characters space, dots 1-2-4-6, lowercase *p*, space as a paragraph indicator. Only when you interrupt the sequence with the null dot 5, dots 1-2-4-5-6 can MathematiX "see" a perpendicular to symbol—this also applies to a spaced pounds sterling symbol and a spaced Greek kappa.

The *Nemeth Braille Code for Mathematics and Science Notation* discusses arrows in Rules 14 and 21. MathematiX only supports the Rule 14 arrows, as illustrated in Samples 5 through 11. Here's the MathematiX perspective on the six steps in arrow construction outlined in Rule 21:

Step 1: Begin with the shape indicator {`$`}.

Step 2: Nothing. Since only horizontal is supported, no direction indicator allowed.

Step 3: Nothing. Since only standard "barbed" arrowheads are supported, no arrow type allowed.

Step 4: Choose between none, left arrow {**[**}, solid dot {*****}, or hollow dot {**.***}.

Step 5: Choose between none or some number of {**33**} shaft indicators. Each {**33**} you add gives another "character chunk" of shaft. For example, {**$[33333333333333333333**} yields an arrow as long as 11 letters.

Step 6: Choose between none, right arrow {**o**}, solid dot {*****}, or hollow dot {**.***}.

MathematiX recognizes the contracted as well as the uncontracted arrow symbols. For vertical arrows, the inkprint result is identical. For contracted right arrow, {**$o**} makes a shorter inkprint arrow than {**$33o**}.

## 8 MODIFIED EXPRESSIONS

### Limited to first order

1. {**#3.54:**}

3.54 with a bar above the 4

2. {**x:+y:**}

x with overbar plus y with overbar

3. {**"x:+y:<:]**}

a bar above the expression x with overbar plus y with overbar

5. {**""x .k y%:]%:]**}

The expression x equals y with two parallel underbars

6. {**""x+y%:]<:]**}

The expression x plus y with simultaneous underbar and overbar.

MathematiX doesn't support modified expressions of the second or higher order: {**"x+y<:<<a .k #3]**} just gets you a structure error when tixed. As seen in Sample 5, repeating the basic underbar or overbar structure yields multiple parallel lines. Sample 6 shows the non-standard MathematiX requirements for simultaneous overbar and underbar: if you entered the correct {**"x+y%:<:]**} the tixed result would only show an overbar.

## 9 MODIFIED & COMPOUNDED SIGNS OF COMPARISON

### Non-standard for Compounded Arrows

1.a `{.1: .1.k}`

Two standard variations on "is greater than or equal to"

1.b `{:.1 .k.1}`

Two rarer variations on "is greater than or equal to"

2.a `{"k: "k.k}`

Two standard variations on "is less than or equal to"

2.a `{:"k .k"k}`

Two rarer variations on "is less than or equal to"

3. `{,a @:@: ,b_1 ,a .k@: ,b_1 ,a @:.k ,b}`

A is approximately equal to B, A is approximately equal to B, A tilde with equals below B

4. `{:@+.k .k_.1 .+:}`

Bar over and equal sign under logical sum; equal sign over reverse inclusion; bar under union

5. `{"$[33*<$.*33o]}`

Two arrows stacked vertically. The top arrow begins with a hollow dot and points right. The bottom points left, with a solid dot at the right.

6. `{"$3333o<$[3333]}`

Two arrows stacked vertically. A long left arrow above a long right arrow

7. `{$33o $[33}`

Two arrows next to each other horizontally: a right arrow immediately followed by a left arrow

Nemeth and MathematiX allow scores of combinations—see Rule 20 for a list. In Samples 1 and 2, we recommend using the "a" versions rather than the "b" versions, when possible. This is especially important when the sign appears above or below the baseline, because MathematiX produces a single character for the "a" versions. For the "b" versions, MathematiX builds up several characters, which can throw off the spacing of other elements when included in a subscript, superscript, fraction, radical, or modified expression.

In standard Nemeth Code, when one arrow follows another, you assume they are stacked vertically. If you really want one arrow followed horizontally by another arrow, you do something special. As shown in Samples 6 and 7, MathematiX reverses these assumptions. To stack two arrows vertically, you need to use the directly over or directly under modified expression. Two arrows next to each other create one arrow followed horizontally by another. The standard Nemeth for Sample 6 would be `{$[333$333o}`, while the standard Nemeth for Sample 7 would be `{$o"$[}`.

## 10 SPECIAL TYPEFACES

### Limited to Script Letters, Bold Letters and Bold Digits

1. `{x+_;,b}`

   x plus capital boldface B

2. `{@;,r^2}`

   Script capital R squared

3. `{_#1_#2 .k _;,l_;,m_;,n}`

   Boldface 12 equals boldface LMN

4. `{?_#1_#2/x#}`

   Boldface 12 divided by x as a spatial fraction.

MathematiX supports neither italic nor sans serif type; only boldface English letters, boldface digits, and script English letters are available. In sample 1, the English Letter Indicator is required where shown to differentiate a boldface English letter from a German letter (Part 11). In sample 3, notice that every character in a special typeface requires its own indicator. MathematiX does not support the indicators to change the typeface for an entire phrase. As demonstrated in sample 4, you *do* use the Numeric Indicator for boldface digits in a fraction, which is the single contradiction to our Numeric Indicator prohibition in Part 3.

## 11 NON-ROMAN ALPHABETS

### Limited To Greek, German and Two Hebrew Signs

1. `{.a+.b+.g}`

   alpha plus beta plus gamma

2.
   ```
   { @l ,we see t @m #4.k .k #2.,k
   when .k"] .k #5 and .,k .k #10_4}
   ```

   We see that four lowercase kappa equals two uppercase kappa when lowercase kappa is equal to 5 and uppercase kappa is equal to 10

3. `{_,a-_,b}`

   Capital German A minus capital German B

4. `{,,a0}`

   Aleph null (Hebrew a sub zero)

5. `{,,a_ x ,,xy,,a}`

   aleph, lowercase x, space, uppercase X, uppercase Y, aleph.

MathematiX does not support any Russian letters. You can't combine a special typeface with a non-Roman alphabet: MathematiX won't make a boldface kappa or a script German S.

In Sample 2, notice the tix null tool `{"]}` (Section 4, Part 2) in the second lowercase kappa. MathematiX always interprets space, dots 4-6, k, space as an equal sign. The tix null tool "breaks up" the sequence, allowing MathematiX to recognize the dots 4-6, k as a kappa.

MathematiX only supports two Hebrew letters: aleph and bet. Two dot 6s can be either the Hebrew letter indicator or the double capitalization indicator. When double dot 6 precedes two or more letters, MathematiX interprets `{,,}` as the double capitalization indicator. In sample 5, we use the `{_ }` disappearing space (Section 4, Part 2) to break up the first expression so that the aleph can touch the lowercase x.

## 12 COMMON SYMBOLS USED IN CHEMISTRY

### No support for electron dot patterns

1. `{,h2,s,o4}`

   sulfuric acid

2. `{,s,o;4^-2}`

   sulfate ion

3. `{,c_3],c}`

   carbon single bond carbon

4. `{,n_7],n}`

   nitrogen double bond nitrogen

5. `{,n_=],n}`

   nitrogen triple bond nitrogen

6. `{2,h2+,o2 $[33o 2,h2,o}`

   Two hydrogen molecules plus one oxygen molecule are in equilibrium with 2 water molecules.

The source of the single, double, and triple bond symbols is "Report of the Chemistry Braille Workshop" (Von E. Eulert, 1985). While these symbols are not found in the Nemeth Code Book, they are widely accepted. In Section 9 we demonstrate using BEX **$$p#** commands to create spatially molecular diagrams.

## 13 SYMBOLS USED IN ADVANCED MATH

### Many Non-Standard Requirements

```
1.   {"lim%x$o0] ?1/x# .k ,=}
```

The limit as x approaches 0 of 1 divided by x is equal to infinity

```
2.   {"$c]+}
```

A plus sign superimposed on a circle

```
3.   {"!%:]}
```

Lower integral (an integral sign with underbar)

```
4.   {@b(n%k)}
```

The binomial coefficient with n things taken k at a time

```
5.   {" .,,s%i_ .k_ 1< ,=] ,(?1/i^5"#,)}
```

The summation from i equals 1 to infinity of 1 over i to the fifth power

```
6.   {?@dx/@dz# .k q''}
```

The partial derivative of x with respect to z is equal to q double prime

```
7.   {@& y @=\;x \ xy .k -xy}
```

For all y there exists an x such that xy equals minus xy

```
8.   {!;0^,="f(x, n) dx .k n&}
```

The integral from 0 to infinity of f of x and n with respect to x is equal to n factorial

```
9.   {,a .% ,b .k @_0 ,* ,a /.k ,b @/ ,a and ,b /.k @_0}
```

The intersection of A and B is the empty set therefore A is not equal to B, since A and B are nonempty

```
10.  {.,(?1/2#, ?1/3#, ?1/4#, ?1/5#, '''.,)}
```

The set containing one half, one third, one fourth, one fifth, etc

```
11.  {"x<_<] .1@.1] y:}
```

x hat is very much greater than y overbar

Samples 2 and 3 demonstrate MathematiX's non-standard approach to superimposition (printing two inkprint signs on top of each other). In MathematiX, the pattern is multipurpose indicator, first symbol, termination indicator, second symbol, while Nemeth uses first symbol, dot 4, second symbol, termination indicator. The proper Nemeth for Sample 2 would be {`$c_$+]`}, meaning "take a circle shape and fill-in with a plus shape." MathematiX does not recognize the standard Nemeth {`%!`} for an integral sign with an underbar. The MathematiX {`"!%:]`} literally tells the software: "modify an integral sign by placing an underbar." Section 4, Part 4 explains how you can use this technique to superimpose just about any inkprint signs, while Part 5 addresses a special use to exercise fine control of vertical spacing.

Standard Nemeth for Sample 4 would be {`(n%k)`}. In MathematiX, you *must* start a binomial expression with {`@b(`}.

Sample 5 shows the special techniques for summation signs. The standard Nemeth would be {`".,s%i  .k  1<,=]  ,(?1/i^5"#,)`}. The double dot six makes an enlarged capital sigma. The extra spaces inside the modified expression make the upper limit, the operator, and the lower limit center correctly in inkprint. The disappearing spaces that bracket the equal sign ensure correct inkprint output. See Section 4, Part 4 for an explanation and Section 9 for more samples.

Sample 9 shows your entry for the empty set sign. MathematiX always interprets {`_0`} as closing double quote: only when you add the dot 4 will MathematiX create the empty set sign.

Sample 10 uses enlarged braces to enclose tall inkprint. This deviates from standard Nemeth, where enlarged braces are only used when the enclosed inkprint comprises several independent lines. MathematiX printed output looks best if you use enlarged signs whenever they enclose more than two levels—more samples in Section 4, Part 4.

## 14 UNSUPPORTED SYMBOLS

MathematiX can't tix inkprint from every possible Nemeth symbol. With reference to the rule numbering in *Nemeth Braille Code for Mathematics and Science Notation,* here's what MathematiX can't do:

- Rules 4 and 5: MathematiX does not support Russian letters, most Hebrew letters, italic letters, or sans serif letters. You may be able to substitute script letters for italics. MathematiX does not support word or phrase typefont indicators.
- Rule 11: MathematiX can't handle cancellation, but see Section 9 for a roundabout way to represent cancellation in inkprint.
- Rule 14: MathematiX does not support second order modified expressions, where the modification is modified.
- Rules 16 and 21 describe a very rich array of mathematical shapes and arrows, while MathematiX only supports a limited number—see Part 7 for exactly which ones.

- Rule 18 addresses signs of grouping: MathematiX does not support barred brackets, barred braces, enlarged angle brackets, or enlarged half brackets. No horizontal signs of grouping are supported, but you may able to use underbar or overbar of an entire expression as a substitute.
- Rules 24 and 25: While you can use BEX tabs and **$$** commands to format your inkprint spatially, MathematiX *can't* tix spatially arranged braille. If you follow Rules 24 and 25 in the Nemeth Code to create spatially arranged arithmetic, for example, you are making braille that MathematiX simply chokes on.
- Miscellaneous: No tally marks, curved less than, curved greater than, bar through inclusion sign, bar through reverse inclusion sign, or inclusion through equals sign. You may be able to substitute a vertical bar {\} for a tally mark.

# Section 6: Math Menu Reference

# Section 6
# Math Menu Reference

The MathematiX disk adds a fifth menu to your BEX program. This Section details how the Math Menu options function. The Math Menu shares many qualities common to all BEX menus. Follow the instructions in User Level Section 4 for naming and selecting chapters. Press <CR> to get the list of options. To choose an option, press just its initial letter.

## 1 MOVING TO AND FROM THE MATH MENU

Since MathematiX came after BEX, its Math Menu isn't completely integrated into the BEX menu structure. The Math Menu is *only* available from the Main Menu. MathematiX won't let you move from the Starting Menu directly to the Math Menu. Moving between the Math Menu and the Main and Starting Menus requires swapping disks. Here's the drill:

| From here | To here | This disk in drive | Press this key |
|---|---|---|---|
| Main | Math | MathematiX Menu Disk | spacebar |
| Math | Main | BEX Main Disk | J |
| Math | Starting | BEX Boot Disk | spacebar |
| Main | Starting | BEX Boot Disk | spacebar |

To move from the Math to the Second or Page Menus, you must first switch disks and get to the Main Menu. If you're storing either the BEX Main side or the MathematiX software on RAM drive or the Sider, some disk swapping is still required to move from the Starting Menu to the Math Menu—please see Section 10 for details.

## 2 OPTION M - MATH OUTPUT

Most of the Math Menu options are borrowed from BEX's Main and Second Menu: details in Parts 3 and 4. The heart of MathematiX is option M - Math Output. The Math Output process, which we call *tixing,* is a combination of translating and formatting. For each BEX page, MathematiX loads two software modules from the MathematiX disk. So unlike any other BEX option, the MathematiX Menu Disk must stay in the drive throughout the process—this is why MathematiX *requires* a two-drive Apple system. (Because of this double disk access, you may wish to consider investing in extra memory for your Apple and BEX 3.0. Then you can run MathematiX from a RAM drive, which is *much* faster.)

### *Choosing the Tix Destination*

Once you press **M** on the Math Menu, MathematiX prompts for the chapter to tix. You specify one or more chapters by name or by scanning, and then Math Output prompts **tix where:**. For a reminder of your choices, type **? <CR>** at this prompt. By typing the first letter and pressing <CR>, you choose one of the three destinations:

S Screen preview is appropriate for sighted users wishing to proofread a tix-ready chapter. The material is printed to the 40-column screen. When the screen fills, you hear a low boop—press the space bar for the next screen.

V Verbalize allows both blind and sighted users to proofread the content and structure of a tix-ready chapter. MathematiX interprets the material, expressing it in words, not graphics. When MathematiX detects a mathematical expression with a faulty structure, it presents an error message accompanied by its approximate location in the BEX chapter. Section 7 describes the tix verbalization tools you can use to control what's spoken and the meaning of the structure error messages.

I Inkprint hard copy creates the regular print mathematical output. MathematiX tixes to the first large print printer you've defined in your configuration.

If you don't want to tix after all, press <CR> alone at the **Tix where:** prompt to get back to the Math Menu. You can cancel tixing at any point by pressing <Esc>. For screen or verbalize destinations, <Esc> halts tixing immediately. When you're tixing inkprint hard copy, it can take up to five seconds for MathematiX to return you to the Math Menu.

### *What Happens When Tixing Goes Right*

Once you type a destination letter and press <CR>, MathematiX reads some software from the MathematiX program disk. Then it reads the first BEX page of the tix-ready chapter and starts to tix it: the scratchy noises you hear mean the software's working. The last step before output is to read some more software from the MathematiX disk. This process is repeated for every BEX page of the chapter or chapters you've specified. If you didn't leave the MathematiX disk in the program drive, you would get BEX's standard **Program segment could not be loaded** error and return to the Math Menu.

### *How To Cope When Tixing Goes Wrong*

Three kinds of problems with your tix-ready chapters can prevent MathematiX from completing the Math Output process.

*Overflow shriek* If the tix scratching noise changes into a shrieking whoop, then MathematiX has encountered an overflow error. There are too many characters in the BEX page, and MathematiX has run out of room in the Apple's memory. Press <Esc> to cancel tixing, then use Adjust pages on the Second Menu or **control-C control-P** in the Editor to reduce the number of characters on each BEX page. For general material, your BEX pages must contain 2500 or fewer characters; for graphic-intensive material like calculus, keep your pages under 2000 characters.

*Structure error* Your material contains contradictory information that prevents MathematiX from tixing. With the Inkprint or Screen destinations, MathematiX stops output and provides the approximate location of the first error. See Section 7 for how you use Verbalize to find and fix the problem.

*Program crash* The Apple beeps, then issues some sort of error message like **SYNTAX ERROR IN 1045** or **OUT OF MEMORY ERROR IN 900**. The tix-ready chapter includes very long math expressions that exceed the maximum characters per line. Since you haven't provided MathematiX with an acceptable place to break the line, it's "crashed." Press control-Reset, depress the Caps Lock, and type **RUN START <CR>**. You must edit the chapter and insert spaces or discretionary line break `{@ }` characters—see Section 4, Part 3 for details.

## 3 THREE MODIFIED OPTIONS

Option P - Print chapters on the Math Menu lacks a few features found in its Main Menu counterpart. The options that let you bring tix-ready data from other braille devices have been modified to work better with MathematiX.

### Print Option Differences

When you *print* a BEX chapter, the formatter creates output pages using the printer's carriage width and form length and any format commands in your text. You can print a tix-ready chapter to a brailler to proofread it in hard copy. The tix tools, math, and literary braille appear just as you entered them: all format indicators and commands are executed. MathematiX only *tixes* when you use option M - Math Output.

The Main and Math Menu's Print option format text identically: the difference lies in your range of printing destinations. On the Math Menu, you can only print to the screen or one of the numbered printers defined in your configuration. You can't use **+V** to add voice output; you can't type **N <CR>** to define a new printer on the fly; and you can't specify **L <CR>** to mean the last printer used. Multi-function print on the Main Menu lets you restart a printout on a specified page or make more than one copy of a document. Multi-function print is not available on the Math Menu: switch disks and go to the Main Menu when you need those features.

### Input through Slot Differences

Because MathematiX greatly expands your data as it tixes, option I - Input through slot on the Math Menu creates BEX pages with a maximum of 2500 characters. In addition, the Math Menu's Input through Slot has one new feature. User Level Section 12 explains how you finish the transfer process by pressing Q on the Apple keyboard: this closes the current BEX page and builds a chapter directory. On the Math Menu, you have an alternative way to end the transfer: place the single **<control-Z>** character at the end of the data you're sending. When the Math Menu's Input through slot sees a <control-Z> in the file, it acts like you've pressed **Q** on the Apple keyboard.

### *From VB Differences*

The Math Menu version of option F - From VB creates BEX pages of 2500 or fewer characters, to minimize overflow errors during Math output. Unlike its Main Menu cousin, the Math Menu's From VersaBraille doesn't ask you if you want control characters. Since math material makes heavy use of dots 4-5-6 and dot 4, trying to transfer control characters from the VB would result in massive confusion. All the symbols that control format in MathematiX are printing characters, so you shouldn't need to use control characters anyway.

After you press **F**, the only prompt is **Do you want V B page indicators? N**. As explained in full in User Level Section 11, most of the time you press <CR> to accept the No default. Then you'll enter **chord-X H** at the VB chapter name to begin the transfer of data.

## 4 SEVEN FAMILIAR OPTIONS

Seven of the Math Menu options are identical to their counterparts on your Main and Second Menus. We duplicated them on the Math Menu for your convenience, so you wouldn't have to constantly switch disks. These "all-stars" are:

Copy chapters: Make identical copy of chapter(s) with same name (on different disk) or different name; at Master Level, load or save Ready (Zippy) chapter.

Disk catalog: List BEX chapters on disk; press <space> for DOS catalog of every file on disk.

Editor: Create and modify one BEX chapter (see User Section 5).

Kill chapters: Deletes chapters from disk.

Quit BEX: Issue DOS or Applesoft BASIC commands.

Replace characters: Change characters in chapter(s) using one of three methods: Changes from a basic or contextual transformation chapter stored on disk; changes typed directly; or contextual changes typed directly. (see User Level Section 8 and Master Level Section 6).

# free sectors: Shows free space on disk.

# Section 7: Checking Your Tix-Ready Chapters with Verbalize

# Section 7
# Checking Your Tix-Ready Chapters with Verbalize

When you use option M - Math Output, you have three destinations for your tixed data. This Section focuses on the Verbalize destination. Once you type **V <CR>** at the **Tix where?** prompt, MathematiX tixes your data in words to the 40-column screen and your voice synthesizer. Part 1 explains what happens during verbalization and how to interpret what you hear. Part 2 discusses the tix verbalization tools that control what's printed to the screen and spoken. Part 3 of this Section is relevant to sighted as well as blind MathematiX users. MathematiX can't tix a series of math symbols that don't conform to its guidelines. If a structure error occurs while tixing to the screen or printer, MathematiX halts output entirely. Verbalize provides all users with detailed information about these structure errors before you tix inkprint. In the Section 2 Guided Tour, Parts 7 and 8 provide a step-by-step sample of using Verbalize and understanding its vocabulary.

## 1 VERBALIZE PARROTS FORMAT AND TRANSLATES THE CONTENT

Verbalize doesn't provide you with a preview of the final inkprint format. As explained in Section 4, MathematiX itself is in charge of most inkprint format decisions. When MathematiX encounters any **$$** commands in the chapter during Verbalize, it simply parrots them back to you. As detailed in Section 4, Part 3, MathematiX ignores format commands that it doesn't support. When you verbalize a command like **$$vl8**, MathematiX tells you that it's not supported. Like BEX, MathematiX allows you to enter nonsensical format values. Even if you set left and right margins larger than the carriage width with **$$ml100 $$mr100**, Verbalize won't complain. However, MathematiX may crash when you attempt to tix to inkprint. When it comes to Math Output of format commands, *garbage in, garbage out*.

What Verbalize does well is tell you the inkprint result of the data you've entered. Verbalize uses a "picky" style for math material, and a more "conversational" approach for literary text. Every space, digit, symbol, letter, and punctuation mark is announced individually in math mode. MathematiX does its best to conventionally read the math aloud. However, MathematiX *doesn't* understand the underlying meaning of the symbols you've entered, so it can't always read things as a person would. In many cases, MathematiX simply announces the name of the signs that would appear in inkprint, leaving the interpretation up to you. You can check Appendix A, Verbalize Vocabulary, for the meaning of any unfamiliar terms.

After you turn on literary mode by typing `{ @l }`, you hear full words and no explicit spaces or capitalization. Set the voice device for "Most" punctuation mode when you want to check your literary punctuation. Use this change between "picky" and "conversational" styles to check the accurate placement of the tix translation tools. It's easy to recognize if you've neglected to precede literary material with `{ @l }`, since MathematiX attempts to interpret the literary contractions as math symbols. You'll hear patent nonsense that quickly degenerates into a structure error. Similarly, if you neglect to switch back into math mode with `{ @m }`, then your math is verbalized as a hodgepodge of unrelated contractions.

### *Verbalize Sample*

Here's a quick demonstration of the "picky" and "conversational" styles. A tix-ready chapter containing just: `{`**`3./8 is smaller than 3./4`**`}` is verbalized as **3 divided by 8 space i s space s m a l l e r space t h a n space 3 divided by 4**. Since the entire sample is in math mode, you hear every letter and space in the phrase *is smaller than*.

On the other hand, consider a tix-ready chapter containing: `{ `**`$p #4_?3/8_# @l is small] ?an @m #4_?3/4_#_4`**`}`. When you verbalize this, you hear:

```
new paragraph 4 start fraction 3
fraction line 8 end fraction space is
smaller than 4 start fraction 3 fraction
line 4 end fraction period
```

In this case, the phrase is in literary mode, so it's pronounced as words. If you omitted the `{ @m }` that turns math mode back on, you'd hear:

```
new paragraph 4 start fraction 3 fraction
line 8 end fraction space is smaller than
dd_thccstdd_ble_.
```

And if you remembered the `{ @m }` but omitted the `{ @l }`, you'd hear:

```
new paragraph 4 start fraction 3
fraction line 8 end fraction space i s
space s m a l l
ERROR: extra termination sign ...
space start fraction a n space
ERROR: fraction not finished ...
ERROR: fraction line missing ...
end fraction 4 start fraction 3
ERROR: fraction not started ...
fraction line 4
ERROR: fraction line missing ...
end fraction period
```

Part 3 explores structure errors in great detail.

## 2 CONTROLLING WHAT'S SPOKEN WITH TIX VERBALIZATION TOOLS

When you verbalize a chapter, you may not wish to hear the entire text. You determine what's spoken with the three *tix verbalization tools:*

`{@talk}` Begin speech and screen output, pausing when the screen fills. (Default)

`{@express}` Begin continuous speech and screen output

`{@notalk}` Suppress speech and screen output unless there's a structure error during verbalization

For inkprint or screen destinations, all the tix verbalization tool characters are suppressed: you don't get `{@talk}` in your output. You can type one space after these tools to make them "words;" if you do, it disappears when tixed. On the other hand, a space before the tool is significant. The following three samples tix identically: `{,a_4 #2 "k #3}` `{,b_4 #2 "k @talk #3}` `{,c_4 #2 "k @talk#3}`.

When you answer **V <CR>** to the **Tix where:** prompt, MathematiX always begins in regular "talk" mode. When the Apple's screen is full of verbalized words, you hear a low boop. You can use screen review to go over what's been said, or you can press the spacebar to continue. When you want to save wear on your thumb, use the "express" talk mode. To freeze output and enter screen review for proofreading, press **control-L**.

The "notalk" lets you zero in on a portion of a chapter for close review. It's also handy when you need to confirm that you've corrected an error found previously. When you verbalize one or more perfect tix-ready chapters that start out with `{@notalk}`, then you only hear the tix scratch and disk access. When you return to the Math menu, you know that all is well.

You can use BEX's Editor to add the `{@notalk}` tool where you want it—remember, it's entirely suppressed when you tix to your printer. Alternatively, you can specify the **HUSH#** "set-up" chapter as first in your list of chapters to tix. The **HUSH#** chapter just contains the `{@notalk}` tool—it's on your MathematiX Menu Disk.

## 3 UNDERSTANDING STRUCTURE ERRORS

Several mathematical expressions require the use of more than one sign in a particular order. For example, a simple fraction must begin with `{?}`, include a fraction line `{/}`, and end with `{#}`. When MathematiX reports a *structure error,* it means that one or more of these signs is missing or present in the wrong order.

### *Structure Error Format*

When MathematiX notices a structure error while tixing to the screen or the inkprint printer, output halts. You hear a beep, followed by:

```
Structure error found.
Please use Verbalize to check
chapter NAME page #
```

MathematiX replaces the **NAME** with the actual BEX chapter name, and the **page #** with the specific BEX page number.

During Verbalize, MathematiX announces structure errors in greater detail (even when you're in `{@notalk}` mode):

```
ERROR: [error description] in
chapter NAME page # paragraph #
```

Check below for the meaning of the error descriptions. To emphasize the error, MathematiX raises the pitch of your synthesizer. MathematiX uses the actual BEX chapter name and BEX page number. MathematiX lists the paragraph number by counting `{ $p }` indicators in your chapter. When the error occurs before the first `{ $p }` in a BEX page, you hear **paragraph 0**.

After the error message, MathematiX restores the pitch to normal and continues tixing. If you were in "notalk" mode, the structure error message acts like a `{@talk}` tool: you hear all the material up to the next `{@notalk}` tool. You can cancel tixing by pressing <Esc> and go fix the error immediately, or you can continue through the list of chapters, noting all errors to hunt them down later. When you're configured at the Master Level, you can use the **control-B P #** command to get a hard-copy version of the entire verbalize screen output—details in Section 9, Part 1.

## Meaning of Structure Error Messages

In the following list, we present all the error messages in alphabetical order. For each one, there's a brief summary of the data pattern that causes MathematiX to emit the message. As discussed further below under "Hints for Interpreting Errors," the fact that MathematiX mentions "fractions" doesn't necessarily mean that the problem lies with a fraction in your material. There will be times when an error message seems completely unrelated to what you're writing. To help you track down problems, use BEX's Locate command to move to the first item in the data pattern.

| Error Message | Data Pattern Generating It |
|---|---|
| Binomial expression not finished | `{@b(}` not followed by `{)}`. |
| Complex fraction line missing | `{,#}` not preceded by `{,/}` whether or not preceded by `{,?}`. |
| Complex fraction not finished | `{,?}` not followed by `{,/}` and `{,#}`. |
| Complex fraction not started | `{,/}` not preceded by `{,?}`. |
| Cube root not terminated | `{<3>}` not followed by `{]}`. |

| | |
|---|---|
| Extra termination sign | an unbalanced {]} that does not match an appropriate starting indicator. MathematiX may add further location information, such as "Extra termination sign in numerator of complex fraction." One cause is neglecting to place the {"} at the beginning of the modified expression; it can also mean a radical that lacks its initial {>}. |
| Fraction line missing | {?} followed by {#} but with no intervening {/}. |
| Fraction not finished | {?} followed by {/} but not followed by {#} |
| Fraction not started | {?} followed by a chunk boundary (explained below) or structurally incorrect radical or modified expression. |
| Hypercomplex fraction line missing | {,,#} not preceded by {,,/} whether or not preceded by {,,?}. |
| Hypercomplex fraction not finished | {,,?} not followed by {,,/} and {,,#}. |
| Hypercomplex fraction not started | {,,/} not preceded by {,,?}. |
| Indexed radical missing radical sign | {]} preceded by {<} with no intervening {>}. |
| Indexed radical not terminated | {<} and {>} not followed by {]}. |
| Modified expression not started | {%} not preceded by {"}. |
| Modified expression not terminated | {"} not followed by {]}. |
| New fraction but previous complex fraction not finished | {,?} or {,,?} preceded by {,?}. |
| New fraction but previous hypercomplex fraction not finished | {,,?} preceded by {,,?}. |
| New fraction but previous simple fraction not finished | {?} or {,?} or {,,?} preceded by {?}. |
| Square root not terminated | {>} or {<2>} not followed by {]}. |

By definition, a complex fraction's numerator or denominator (or both) contains a simple fraction, while a hypercomplex fraction includes a complex fraction in its numerator and/or denominator. However, MathematiX does not check to see if your complex and hypercomplex fractions strictly satisfy these definitions. When you enter {,?a;4,/45,#} to allow room for a subscripted numerator, MathematiX tixes the inkprint with more vertical space than if you entered the standard Nemeth `{?a;4/45#}`.

## Hints for Interpreting Errors

MathematiX can't know what you mean to write; when the conditions listed above are met, you get the error message. When you *intend* to write a hypercomplex fraction and omit one of the required symbols, then the error messages are self-explanatory. However, there will be times when an error message seems completely unrelated to what you're writing.

The most common cause of structure errors is writing grade 2 contractions in math mode. You *must* introduce all literary material with either the one-space `{ @l }` or disappearing `{_ @l }` tix translation tool. When you don't, every *th* or *st* sign causes a fraction-related error, while *ar* and *gh* generate radical-related errors. In addition to missing translation tools, missing spaces can cause havoc. The spaces around dot 5 are crucial to its meaning: If you slipped a space in the middle of the "less than" `{"k}` you'd get an "Modified expression not terminated" error.

Several Editor commands can help you track down problems. (Section 2, Part 8 demonstrates a host of techniques.) Let's say you intended to write `{/,*}` "it does not follow that," yet you reverse the position of the dots 3-4 and dot 6. You get a message **Complex fraction not started in chapter JUNK page 1 paragraph 4**. Edit page 1 of the **JUNK** chapter. Then enter **control-A 4 control-P** to advance four `{ $p }` indicators. Checking the error message list, you see that this message happens when MathematiX finds `{,/}` that's not preceded by `{,?}`. Once you're in the fourth paragraph, type **control-L ,/ control-A** to zero in on the problem data.

### *Timing and Placement of Structure Errors Depends on "Chunking"*

MathematiX checks the structural integrity of your material in chunks, one BEX page at a time. To decide where an expression ends within each BEX page, MathematiX can use hard <CR>s, new line `{ $l }`, new paragraph `{ $p }`, and new page `{ $f }` indicators. MathematiX's "chunking" techniques can influence the timing and placement of structure errors. For some expressions, the error description pops up immediately. But generally, MathematiX works backwards from the chunk boundary, reporting the problems in reverse order. A chunk boundary in the middle of an expression can cause several, accumulating structure errors.

For example, suppose you verbalize a **LOUSY#** chapter containing: `{,?24,/>? $p 1/2#],#}`. You would hear:

```
  start complex fraction 2 4
  complex fraction line square root of
  start fraction
  ERROR: fraction not finished in
  chapter LOUSY# page 1 paragraph 0
  ERROR: square root not terminated in
  chapter LOUSY# page 1 paragraph 0
  ERROR: complex fraction not finished in
```

```
chapter LOUSY# page 1 paragraph 0
new paragraph 1 slash 2
ERROR: extra termination sign in
chapter LOUSY# page 1 paragraph 1
ERROR: complex fraction line missing
chapter LOUSY# page 1 paragraph 1
end complex fraction
```

When MathematiX encounters the paragraph indicator in the middle of this otherwise correct complex fraction, it *first* reports a {?} that's not followed by {/} and {#}. Then it checks to see what else has been started in this paragraph. Working backwards, it complains about the {>} that's not followed with {]}, and finally the {,?} that's not followed by {,/} and {,#}. When MathematiX announces the new paragraph, it's completely finished with the first chunk.

That's why {1/2} is announced as "one slash two" instead of "one fraction line two." When MathematiX gets to the termination indicator {]}, it realizes that nothing has been started, so it reports "Extra termination sign." Finally, it reaches the BEX page chunk boundary, and complains about the {,#} which has not been preceded by {,\}.

Structure errors are reported with paragraph numbers, so liberal use of paragraph indicators makes your material easier to navigate. When you use Input through slot, it's possible that BEX would move to a new page in the middle of an expression, causing structure errors in correct material. Use the clipboard to copy the material from the end of one page to the start of the next. When you hear a spate of error messages in a row, you need to peel back layers of the onion to locate the culprit. Until you're a MathematiX expert, we recommend you confirm that you have located and fixed all problems by verbalizing again after you hear and correct errors.

# Section 8: Troubleshooting

# Section 8
# Troubleshooting

When you encounter problems using MathematiX, check here to see if there's a simple solution. Next, take a look at the **KNOWN MATHEMATIX BUGS** chapter on the MathematiX Samples Disk to see if we have supplied a workaround. If not, please see the end of this section for how to contact us for help!

## General Math Output Problems

I don't hear the tix scratch

The Apple speaker isn't connected or is turned all the way down.

On an Apple IIc, rotate the lefthand volume knob towards the back. On an Apple IIgs, use the Control Panel to increase the volume setting. If you have an external speaker (like the Echo's), check the volume setting and that the wires are secure.

The tix scratch changes to a whooping siren and nothing's printed or verbalized.

An *overflow error* caused by too many characters in the BEX page being tixed.

Your tix-ready chapter must contain 2500 or fewer characters for each BEX page. Limit your pages to 2000 characters for graphic-intensive material—lots of integrals, fractions, superscripts, and subscripts. Use **control-C control-P** in the Editor or Adjust pages on the Second Menu. Details in Section 3, Part 1.

## Problems When Verbalizing with Math Output

After I enter **V** <CR> at the **Tix where:** prompt, all I hear is the tix scratch.

There's a `{@notalk }` tool at the start of the chapter, and your material contains no structure errors.

Find the `{@notalk }` and delete it, then try again.

I hear the punctuation characters in my math material, but not in my literary material.

Verbalize speaks every single sign and space when in math mode. In literary mode, it just sends the inkprint characters to the screen.

Set your voice device so it pronounces the punctuation you wish to hear.

When I wrote the material on my external device, I know it was perfect. But when I Verbalize it, there are structure errors.

There's a BEX page break at the wrong place, like in the middle of a modified expression. Input through slot keeps your BEX pages under 2500 characters, but can't recognize that it's interrupting math entities.

Edit the chapter and use the clipboard to move material between BEX pages. It's a good habit to start each BEX page with a new-line or new paragraph indicator.

## Problems When Tixing Inkprint with Math Output

I typed **I <CR>** at the **Tix where:** prompt but I don't get inkprint output.

Printer's not paying attention to the Apple.

Many minor problems can prevent the printer from accepting signals from the Apple. Check all of the following things: that the printer is "on-line" or "selected;" that the printer's power cord is plugged in at the wall and at the printer; that the printer interface cable is connected at the Apple and the printer; that the printer has enough paper; that the paper or ribbon has not jammed.

Still no inkprint output when I specify **I <CR>** at the **Tix where:** prompt.

Printer or interface card is incompatible with BEX large print.

Read Section 4 in the BEX Interface Guide on interfacing large print printers. Use option P - Print chapters on the Main Menu to test the large print capabilities of the printer. If you can't get large print output, see the instructions at the end of this Section on calling for help.

My **$$ub** and **$$ub** commands aren't underlining.

Commands appear within math material.

MathematiX only underlines literary material. You can simulate underlining of math notation by using a modified expression with an underbar below—see Section 5, Part 8.

Portions of the material from my tix-ready chapter weren't printed.

You have written a very long math expression without spaces that exceeds the carriage width (default 76 for ImageWriter, 58 for Epson). MathematiX doesn't know where to break the text into lines, so it throws it away.

Edit the chapter and insert spaces or the discretionary line break `{@ }`. Read more in Section 3, Part 2.

Output halts, there's a beep, followed by a message like **SYNTAX ERROR IN 1045** or **OUT OF MEMORY ERROR IN 900**.

Two possible causes: a very long math expression with no place to break the line (see previous) or there's a contradictory format command. For example, a **$$mr100** would create an effective carriage width of zero, because the right margin is larger than the entire carriage width.

Press control-Reset, depress the Caps Lock, and type **RUN START <CR>**. Check for places needing spaces or discretionary line break {@ } characters. Make sure your format commands make sense. Sometimes the program crashes so badly, you have to reboot.

MathematiX tells me to use Verbalize to find a structure error, but Verbalize doesn't report any structure errors.

Program bug.

Inkprint and Verbalize are two separate programs. We attempted to make them function identically, but we're only human. Contact us for help—see the instructions at the end of this Section on how.

## ■ RAM Drive Problems

After I press **control-D** at the Math Menu to load the RAM drive, the program hangs.

There isn't room on the RAM drive for all the MathematiX software.

Establish a new configuration where virtual drive 2 is slot 3 drive 2 and try again.

When I'm at the Math Menu RAM drive, the chapters on my Main side RAM drive have disappeared!

BEX has changed the meaning of "drive 1" to the "Math Menu RAM drive."

Use the asterisk * as a drive number to access the Main Side RAM drive—details in Section 10, Part 2.

### *Calling for Technical Support*

To help us help you better, we ask that you:

- Make sure that the data disks containing problem chapters are on hand.
- Dig out your MathematiX Manual so you can refer to it during our conversation.
- Position the telephone near your Apple, with MathematiX up and running, if possible.
- Note down the name and model of your printer and interface card.
- Use option W - What is in the computer on BEX's Starting Menu to find out which cards are in what slots.
- Call our Technical Helpline at 608-257-8833.

# Section 9: Putting MathematiX to Work

# Section 9
# Putting MathematiX to Work

In this Section we have gathered together an interesting miscellany of hints, tips, and techniques. Part 1 discusses three BEX Master Level features that supercharge MathematiX. In Part 2 we describe how to produce more closely-conforming Nemeth braille from your tix-ready chapters; Part 3 explains how to enlist the aid of the Grade 2 translator in creating the literary parts of a tix-ready chapter. One way that blind users can be come more familiar with the appearance of inkprint mathematics is detailed in Part 4. Calculus users will find Part 5 of particular interest. As Parts 6 and 7 briefly demonstrate, you can use MathematiX to output very complicated materials. RDC wants you to make best use of MathematiX. If you're unsure how to create a particular format, please get in touch so we can work together to figure out if it's possible.

## 1 MASTER LEVEL BEX TECHNIQUES

You can take advantage of a number of BEX Master Level features to make your MathematiX use more enjoyable. First and foremost is RAM drives, discussed in Section 10. The section numbering in the BEX Master Level changed dramatically from version 2.2 to version 3.0, so we just refer to section titles here.

### ■ Chapters in Memory Tix Faster

Both BEX 2.2 and 3.0 provide a special BEX chapter that's in memory, not on disk. In BEX 2.2, it's called the Zippy chapter and its name is the single <Del> character. In BEX 3.0, the same thing is known as the Ready chapter, referred to with the single ] right bracket character. Whichever version of BEX you have, Verbalizing goes much faster when your tix-ready chapter is stored in this Zippy/Ready chapter.

### ■ Getting Hardcopy of the Verbalize Information

The Master Level **control-B** I/O commands let you send data to more than one channel at a time. You can take advantage of this to get a simultaneous hardcopy record of what shows on the screen during Verbalize. This can be very helpful when you're trying to track down thorny structure errors. For example, when your configuration defines printer 4 as a paperless brailler, issue the command **control-B P 4** immediately before you press **V** at the **tix where:** prompt. From that point on, all the information that goes to the voice channel is also sent to the paperless brailler. When Verbalize is finished, use **control-B P D** to turn the printer channel off again.

### *Two Cautions on control-B P #*

There are two points to keep in mind when you use this trick. As always, the speed of the slowest channel determines the pace of output. If you ask BEX to send the verbalize information to a Cranmer Brailler, it will take a *very* long time. If you turn the printer off-line, then you "freeze" BEX. To recover, simply turn the device on-line again, then issue the command **control-B P D**.

Secondly, BEX has a disconcerting but minor bug that you may encounter the *first time* you activate a printer channel, depending on how the printer is configured. The first time you issue the **control-B P #** command, the program crashes into the Apple monitor. You get random numbers like **00/378E: 00 24** and finally an asterisk * prompt. Type **control-C <CR>** to leave the monitor and get the BASIC prompt, then type **RUN <CR>** to get back to the menu. Now you can issue the **control-B P #** command and it works fine.

## Speeding up MathematiX with Automatic Procedure Chapters

Automatic procedure chapters, detailed in the BEX Master Level, let you perform multi-step operations unattended. You can establish an auto chapter that answers all the prompts for the three basic MathematiX steps:

1 Bringing the data from an external device with From VB or Input through Slot

2 Checking the data for structure errors with the Verbalize destination in Math Output

3 Tixing the data to your inkprint printer

To make sure that untixable data won't cause havoc, when you run an auto chapter from the Math Menu, a Verbalize structure error halts execution of the stored keystrokes. (As always, a disk error also stops an auto chapter in progress.) The `{@notalk }` tix tool can streamline the process: when your data is free of errors, the Verbalize step is silent except for the tix scratch. To ensure the presence of `{@notalk }`, you can specify the **HUSH#** chapter on the MathematiX Menu Disk as the first tix-ready chapter you tix. (The **HUSH#** chapter just contains the eight characters `{@notalk }`.)

Here's a step-by-step example of how you could create an auto chapter that performs those three steps. While the sample uses Input through slot, it works equally well for From VB.

```
Math: control-R
Start remember mode
Math: I
Input through slot
Target chapter: HOMEWORK# <CR>
Apple is ready to receive, begin sending from remote device
Issue appropriate commands on remote device to send
a file that ends with <control-Z>
Math: M
Math output
Chapter: 1HUSH# <CR>
Chapter: HOMEWORK# <CR>
```

```
Chapter: <CR>
Tix where: V <CR>
Tix scratching, hopefully followed by total silence. Make
sure that your printer's top-of-form is set correctly and
you have enough paper.
Math: M
Math output
Chapter: HOMEWORK# <CR>
Chapter: <CR>
Tix where: I <CR>
Math: control-S
Auto chapter: 1TIX HOMEWORK-A <CR>
Math:
```

In order to include Input through slot in the auto chapter, you must type a <control-Z> character at the end of the file in your external braille device. Since the auto chapter always uses the name **HOMEWORK#** for the received file, if you repeat the procedure with the same data disk, you overwrite the chapter each time. When you want to save an earlier version, simply rename the chapter before repeating the process.

The **TIX HOMEWORK-A** chapter that's saved on the program drive contains all the keystrokes needed to duplicate the process at a later time. To use the auto chapter, put the MathematiX Menu Disk in drive 1, a data disk with sufficient room in your default data drive, get your external device and printer ready, and press **control-A** at the **Math:** prompt. MathematiX prompts you for the auto chapter name: type **1DO HOMEWORK-A <CR>** and sit back and relax.

## 2 CREATING BRAILLE OUTPUT WITH MATHEMATIX

The **T2B-T** transformation chapter on the MathematiX Menu Disk helps you change a tix-ready chapter to more standard Nemeth Code. **T2B-T** finds the MathematiX-only symbols like { **@m** } or {_ } and replaces them with nothing. Once this replacement is complete, you can emboss the modified chapter, using option P - Print chapters on the Math or Main menus.

Suppose you have a chapter named **LESSON2#** on a data disk in drive 2. Here's how you prepare a hard-copy braille version without any of the MathematiX-only symbols.

```
Math Menu: R
Replace characters
Drive or chapter: LESSON2# <CR>
Drive or chapter: <CR>
Target chapter: LESSON2X <CR>
Use transformation chapter: 1T2B-T <CR>
Continue? Y <CR>
```

You hear some interesting clicks and pops as the MathematiX symbols are deleted. Notice that the target chapter has a different name! If you entered the same name for the target chapter, you would no longer have the tix-ready version. You're now ready to proofread the **LESSON2X** chapter. When you are satisfied, you emboss it by specifying the printer number for your brailler.

## 3 PREPARING TIX-READY CHAPTERS WITH HELP FROM THE GRADE 2 TRANSLATOR

While MathematiX is geared towards the production of inkprint from braille input, it can also be used to assist in the production of Nemeth braille. Sighted people can tix to the Screen to quickly check the accuracy of Nemeth Code entry. This technique works best when the document you're transcribing is mostly literary, but makes some use of Nemeth Code. (Many of the samples in this Manual were prepared this way.)

1 Enter the material in inkprint and manual Nemeth Code. You must precede all literary inkprint with { **@l** }. When you switch to manual Nemeth entry, start off with { **@-** }. (As introduced in Section 3, Part 2, MathematiX interprets { **@-** } and { **@m** } identically.) You must follow all the guidelines in Sections 3 and 5 when it comes to entering the math symbols. This is your **ORIGINAL** chapter.

2 Translate. Use option G - Grade 2 translation on the **ORIGINAL** chapter, creating a **ORIGINAL2** chapter. Grade 2 translation will only occur between the { **@l** } and { **@-** } controls.

3 Tix **ORIGINAL2** to the Screen. Use option M - Math Output to proofread, specifying the Screen tix destination.

4 Correct errors in **ORIGINAL2**. If any problems pop up when proofing, edit the chapter, fix them, and proof again to confirm.

At this point, you have a tix-ready chapter. Use option N - Name change on the Second Menu to rename **ORIGINAL2** as **HOMEWORK#** or whatever.

When you want to prepare a more standard Nemeth braille hardcopy of this data, follow the instructions in Part 2 on the use the **T2B-T** transformation chapter.

### *Forcing the Letter Sign to Ensure Better MathematiX Back-translation*

As mentioned in Section 3, Part 2, MathematiX needs more letter signs in literary mode than good grade 2 requires. Option G - Grade 2 translator adheres more closely to the grade 2 rules. You can force the Grade 2 translator to place a letter sign by preceding the letter with {**>;** }. Lettered outlines are an example. Grade 2 doesn't require a letter sign before an isolated letter that is followed by a period. When you enter

```
   @l b. GENERAL REACTION: @- ,a + ,b $o ,,ab @l
```

then the translator creates:

```
  { @l b4 ,,g5]al ,,reac;n3 @- ,a + ,b $o ,,ab @l }
```

When you tix this, the initial letter becomes **but**..

Slip {**>;** } before the letter to force the translator to place a letter sign: **@l >;b. GENERAL REACTION: @- ,a + ,b $o ,,ab @l** . Don't do this in math material—

if you want an English letter sign or any other braille character, you have to enter it yourself.

## 4 CREATING TACTILE VERSIONS OF MATHEMATIX'S GRAPHICS OUTPUT

When you specify the Screen preview tix destination, Math Output produces inkprint on the Apple's HI-RES screen. Using the techniques described here, you can save one screen's worth of output as an Apple HI-RES graphic file. You can then use RDC's pixCELLS software (or Lorin Software's Illustrations program) to send this graphic file to an embosser, providing a tactile image of MathematiX's graphics output. Each brailler dot in the tactile graphic corresponds to one inkprint dot in MathematiX. This can show blind users the MathematiX version of a specific inkprint sign, as well as the spacing decisions that MathematiX makes.

### *Tailoring the Image to One or Two Braille Pages*

pixCELLS makes one brailler dot from every lit pixel on the Apple HI-RES screen; a full screen requires six braille pages. Whether the graphics file requires one or two brailler pages vertically depends on how much information is in your tix-ready chapter. When you tix to the screen, MathematiX uses an effective carriage width of 40, which would require more than one braille page horizontally.

Use a right margin command to narrow MathematiX's output so that it fits within one braille page horizontally. When you're embossing the graphic on a VersaPoint, Ohtsuki, or Romeo, begin your tix-ready chapter with **$$mr25**. For a Cranmer Brailler, use **$$mr26**. Since less information fits on each screen line, very long math expressions won't be accurately represented.

You can only save one screen's worth of information at a time. This means that after you type **S <CR>** at the **Tix where:** prompt, you should hear one low boop after the tix scratch. When you press the spacebar after this boop, you return to the Math Menu. If you hear another low boop, it means there's more than one screen worth of stuff.

### ■ Capturing the Screen Image Step-by-Step

We supply an automatic procedure chapter on the MathematiX Menu Disk named **PIXGRAB-A** which issues the commands that save the graphics file. To use it, you must have two floppy disk drives—see below for how to cope when you don't.

Prepare a short tix-ready chapter that begins with the appropriate **$$mr#** command. Tix the chapter to the Screen preview destination to make sure that it's just one screen long. The **GREEK#** chapter on the MathematiX Sample Data disk is an example; it contains all of MathematiX's Greek characters. You must have a data disk in drive 2 with at least 34 free sectors. Now you're ready to go:

```
Math Menu: M
Math Output
Drive or chapter: GREEK# <CR>
Drive or chapter: <CR>
Tix where: S <CR>
```

You hear the tix scratch, followed by brief silence and a low boop. Press the spacebar to return to the Math Menu. You're ready to run the auto chapter:

```
Math Menu: control-A
Auto chapter: 1PIXGRAB-A <CR>
```

Now BEX goes through the process of saving the last HI-RES image under the name **MATH.IMAGE** on the data disk in drive 2. Once the file is saved, you are back at the Math Menu.

### *Manually Saving the Image*

When you don't have two floppy disk drives, then the **PIXGRAB-A** chapter won't work. But you can still issue the appropriate commands manually. Once you have tixed to the screen and pressed space, here's what you do:

```
Math Menu: Q
] BSAVE  MATH.IMAGE,A$2000,L$2000,S6,D1  <CR>
] RUN <CR>
Math Menu:
```

You must type the address **A$2000,L$2000** exactly as shown: this captures the HI-RES image. You can substitute any name you prefer where we show **MATH.IMAGE**. When you wish to save the image on a different disk drive, change the # in **S#,D#**. (When you're comfortable editing auto chapters, you can copy **PIXGRAB-A** and modify the disk drive references there.)

### *Capturing A Series of Images*

If you repeat this process with the same data disk, the new **MATH.IMAGE** file replaces the old. When you want to capture a series of graphics, use the DOS **RENAME** command after each "grab." Suppose you're capturing two images. You automatically or manually save the first image, then do this:

```
Math Menu: Q
] RENAME  MATH.IMAGE,IMAGE.1,S6,D2  <CR>
] RUN <CR>
Math Menu:
```

Now when you capture the second image, the **IMAGE.1** file is preserved.

## 5 FORMATTING LIMITS CORRECTLY

In Section 4, Part 4, we discuss how MathematiX positions the upper and lower limits around a central operator in a modified expression. MathematiX can't center the limits automatically. If both your lower and upper limits are one or two characters (inkprint signs) long, then you can't control centering anyway. But when either limit is three signs or longer, you must add spaces *inside* the modified expression. Each space after the above symbol, dots 1-2-6, nudges the upper limit to the right, while each space after the below symbol, dots 1-4-6, nudges the lower limit to the right. Each space you put after the initial dot 5 nudges the central operator one character to the right.

In the following samples, we use "n" to stand in for any lower limit, and "x" to stand in for any upper limit. The operator is always shown as the sigma, but the techniques hold true for other operators as well. Take special notice of where spaces appear:

| **Limit Lengths** | **What You Enter** |
|---|---|
| One or two | `{".,,s%n<x]}`<br>`{".,,s%nn<x]}`<br>`{".,,s%n<xx]}`<br>`{".,,s%nn<xx]}` |
| Lower limit of three | `{" .,,s%nnn< x]}`<br>`{" .,,s%nnn<xx]}`<br>`{" .,,s%nnn<xxx]}` |
| Upper limit of three | `{" .,,s% n<xxx]}`<br>`{" .,,s%nn<xxx]}`<br>`{" .,,s%nnn<xxx]}` |
| Upper limit of four | `{"  .,,s%  n<xxxx]}`<br>`{"  .,,s% nn<xxxx]}`<br>`{"  .,,s% nnn<xxxx]}` |

Sample 1 doesn't use any "padding" spaces, because the lower and upper limits are two or fewer characters. In the other samples, you always pad the sigma. Of course, when your limit produces an even number of signs, no amount of tweaking can create perfectly balanced centering.

MathematiX requires spaces on both sides of the equals, greater than, and less than signs of comparison. For correct inkprint, signs of comparison in the upper and lower limit must not be spaced. It's important to use the disappearing space for signs of comparison in a limit, and to only enter "padding" spaces exactly where we show them. The following chart shows what can go wrong:

1. `{" .,,s%n_ .k_ #0< ,=]a;n}`

   Best result: large sigma with infinity centered above and (unspaced) "n equals 0" centered below, followed by "a sub n".

2. `{".,,s%n_ .k_ #0<,=]a;n}`

   Acceptable result: large "sigma," "infinity" above, unspaced "n equals 0" below all aligned on left, followed by "a sub n."

3. `{".,,s%n .k #0<,=]a;n}`

   Problematic result: large "sigma" with "infinity" centered above and "n" centered below; followed by "a" with "equals" immediately below, followed by subscript "n," followed by "0" on same baseline as "n" and "equals."

4. `{".,s%n  .k  #0<,=]a;n}`

Correct Nemeth but Illegible result: Standard "sigma" next to lowercase "a." "Infinity" touches top bar of "sigma;" "n" is immediately below "sigma," "equals" is superimposed on subscript "n," "zero" stands alone to right of obliterated subscript.

## 6 DEEP INSIDE MATHEMATIX'S SPACING

The tix vertical tools let you get deep into MathematiX's format. As we stressed when we introduced them in Section 4, Part 5, the tix vertical tools override MathematiX's automatic spacing decisions. For success, you need to know what inkprint math *should* look like and be able to think like a software program. Thanks to long lists of exceptions deep inside MathematiX, 97% of your output will be just swell. This Part provides the background so diligent users can fix up that last three percent; Part 7 explains how we use these techniques in the chapters from MathematiX Sample Data disk. Verbalize doesn't provide feedback on the format created by the tix vertical tools. Either a sharp-eyed sighted assistant or a graphics program like pixCELLS (see Part 4) is required to see the tools' effects.

### Background on Relative Vertical Positioning of Characters

While MathematiX does a good job of making draft quality output, it occasionally makes some poor choices. If you desire to improve the output, you need to know how MathematiX places characters on the page. Some parts of MathematiX make smarter decisions than others.

For the relative vertical placement of many expressions, MathematiX takes into account the actual width and height of the signs involved. No manual intervention is required when MathematiX creates:

- radical signs
- overbars above modified expressions
- underbars below modified expressions
- the horizontal centering of fractions
- the vertical position of the denominators in fractions

For example, when you tix `{>67^3"]}` MathematiX raises the top line of the radical so that it doesn't obscure the cube exponent. When you tix `{"!<:]}`, MathematiX keeps track of the height and width of the integral sign. When it places the overbar, it's clearly above the integral.

On the other hand, sometimes the vertical placement of signs is based on MathematiX's assumption that the enclosed character is "normal size." When the enclosed character is taller, then the relative position is too close together. (A list of the tall characters appears below.) You can use tix vertical tools or other techniques to improve the presentation of:

- numerators in fractions
- single character overbar
- signs positioned above a modified expression (except for overbar)
- signs positioned below a modified expression (except for underbar)

## *Strategies for Improving Vertical Spacing*

When creating spatial fractions, MathematiX always places the horizontal fraction bar at the same position relative to the baseline. The denominator material is automatically dropped down as needed. MathematiX assumes that the lowest character in the numerator does not drop below the baseline. One way to make room for droopy numerators is to write the fraction as complex, even if it's simple. Or you can use the tix vertical tools. When the numerator has a subscript, raise the numerator one-quarter character with `{[<q}`. When the numerator has a modified expression with one sign printed below, raise the numerator one character with `{[<o}`.

When you tix `{.,,s:}`, MathematiX assumes the summation operator is as tall and as wide as a capital letter. The single character overbar appears too low, and follows the sigma. Since MathematiX does an excellent job of placing overbars and underbars with modified expressions, the simple solution is to write `{".,,s:]}`.

When positioning signs above or below a modified expression, MathematiX generally assumes the central sign(s) are one character high. The exceptions are the tall operators (summation, product, intersection, and union) and the integral sign. Suppose you wish to print a right-pointing arrow below a quantity enclosed within enlarged braces. As the chart below shows, enlarged braces are two characters high, straddling the baseline; they extend one-half character above and below the baseline. To ensure legibility in the lower limit, drop it one full character for a total of two characters below: `{".,(?1/3#.,)%[%ox$o0]}`.

MathematiX can't tix an *indexed radical* in the upper or lower limit. Fortunately, there's an alternative math notation that MathematiX *can* output correctly. Show the indexed radical with a linear fractional exponent. To show an upper limit of the cube root of two, use
`{"  .,,s%n_  .k_  #0<2^1/3"]}` instead of
`{"  .,,s%n_  .k_  #0<<3>2]]}`.

## *The Thirteen Tall Characters*

The majority of signs are a standard MathematiX "character": eight pixels high and seven pixels wide. There are a handful of "tall" exceptions, in three sizes:

| | | |
|---|---|---|
| Three characters high: | `{!} {!!} {!!!}`<br>`{,,\}` | integral; double-height enlarged vertical bar |
| Two characters high, straddling baseline: | `{.,,s}`<br>`{.,,p}`<br>`{,(} {,)}`<br>`{.,(} {.,)}`<br>`{@,(} {@,)}`<br>`{,\}` | summation operator;<br>product operator;<br>enlarged parentheses;<br>enlarged braces;<br>enlarged brackets;<br>enlarged vertical bar |
| One-and-one-half characters high: | `{.%} {.+}` | intersection (half character below); union; (half character above) |

The **CHARACTER SCALE** chapter on the MathematiX Sample Data disk contains a hairy collection of tix vertical tools that tix a stack of horizontal lines at the left margin. You can clipboard the contents of this chapter before any characters into a test chapter. When output in inkprint or as brailler graphics (Part 4), the scale helps you count the height of any character combination.

### *Background on Superscripts and Subscripts*

Unlike typeset mathematics, when MathematiX tixes superscripts or subscripts, it uses the standard-size character. It uses the following values to shift away from the baseline:

| | |
|---|---|
| first level subscript or superscript | one half character; four pixels |
| subscript of subscript or superscript of superscript | additional quarter character or two pixels; six pixels from reference baseline |
| subscript of subscript of subscript or superscript of superscript of superscript | additional eighth character or one pixel; seven pixels from reference baseline |

These values apply to many, but not all superscripts and subscripts. The exceptions are integral sign or a closing enlarged symbol of enclosure: in those cases, MathematiX increases these values automatically to ensure legibility.

You can take advantage of these exceptions to improve the clarity of some superscripts and subscripts. For example, when you want to show the *quantity* the fourth root of 89 cubed, enter `{,(<4>89],)^3}`. When you use the right bracket notation to evaluate an integral between the limits, use an enlarged right bracket, not a standard right bracket: `{x^3"-2x+1@,);1^4}` instead of `{x^3"-2x+1@);1^4}`.

### *Guidelines for Counting Length*

MathematiX can get cranky when asked to tix very long expressions without spaces or discretionary line breaks. The default carriage width is 76 characters for an ImageWriter and 58 for the Epson. When you're lining up material on two or more lines, you may need to know how long expressions are. In general, each Nemeth *symbol* stands for one inkprint *sign,* and produces one MathematiX character when tixed to inkprint. Most Nemeth *indicators* are interpreted as format information, and don't create a specific character when tixed. Here are the only exceptions:

**Six double-width characters** {`$r`} rectangle; {`$e`} ellipse; {`$a`} concave up arc; {`$'`} concave down arc; {`.,,s`} summation operator; {`.,,p`} product operator. The last two are also two characters high.

**Multiple character sequences** When the math symbols mean one character *followed* by another, then the result is several characters. For example, {`+"-`} results in two characters: one plus sign followed by one minus sign. {`+-`}, on the other hand, results in a one-character-wide minus sign with a plus sign above it.

**Radicals** The radical sign adds one character to the length of the expression: {`<18>478]`} is six characters.

**Horizontal arrows** Take the number of shaft indicators {`3`} and divide by two, rounding up. Then add the number of arrow heads to find the total arrow length. {`$[33333`} is four characters: (5 divided by 2 rounded up is 3, plus one for the arrowhead); {`$[33o`} is three characters (2 divided by 2 is 1 plus 2 arrowheads).

### *Cautions for Using the Tix Vertical Tools*

When you use tix vertical tools within a dot 5, save dots 1-2-4-5-6 restore pair, you must carefully plan both vertical and horizontal spacing. If you save and restore carelessly, you can end up never advancing horizontally.

Here's an example of how *not* to write a short array: {`,\"[<h,x]"[%h,x],\`}. In this version, the closing vertical bar overwrites the two stacked X's. Here's why: MathematiX outputs the first vertical bar and saves the position at one character after the vertical bar at the baseline. The first tix vertical tool moves up from the baseline to print the upper X. The termination indicator takes you back to the baseline vertical position and the post-bar horizontal position. This position is saved again, then MathematiX goes down a half character and tixes the X, and finally restores the position. Since both X's have been bracketed by {`"  ]`}, MathematiX has never advanced one character horizontally. When it comes time for the closing vertical bar, it overprints the X's.

Here are two alternatives that tix correctly: {`,\"[<h,x]"[%h,x]  ,\`} adds a space before the closing vertical bar, advancing the horizontal position one character. {`,\"[<h,x][%h,x[<h,\`} is a little trickier. The first X is bracketed with {`"  ]`}, resulting in no horizontal advance. The second X is *not* bracketed: you manually move down a half character, tix and advance the X, then manually move back up a half character.

## 7 FORMATTING MATH OUTPUT WITH $$ COMMANDS AND TIX VERTICAL TOOLS

The MathematiX Sample Data disk contains several tix-ready chapters that demonstrate a few of the many ways you can format your MathematiX output.

### *Spatially Arranged Chemistry Diagrams*

You can use MathematiX to print up chemical structures. Take a look at the ACETIC# chapter, which shows the chemical structure of acetic acid. The trick is to work out the structure in advance to ensure that the rows align. Since the braille and inkprint versions of the chemical bond symbols are the same horizontal width, you can check the accuracy by printing your trials in hardcopy braille.

Because a chemical structure is very fluid, tabs are not very useful. Instead, we use the absolute position command **$$p#**. Remember that the first character on the line is **$$p0**. The sequence {`_//`} is an oblique double bond. The sequence {`_3]`} is a horizontal bond. The character {`\`} is a vertical single bond.

### *Formatting a Calculus Test*

The CALC# chapter demonstrates a host of the techniques enumerated here, as well as some others. You can emboss this chapter to see how we handled the margins. We used a left margin of five and a right margin of 10 for the problem statements; this makes it easier for inkprint reader to distinguish between the problems and solutions.

This chapter contains many long equations, where we liberally use **$$kb $$kf** pairs. Notice that when showing an integral with limits, it's clearer to place a space after the upper limit and before the integrand. For example, {`!;3^4 ?1/x^2"#  dx`} is preferred to {`!;3^4"?1/x^2"#  dx`}. To ensure that the integrand appears on the output same line as the integral, use a **control-S** sticky space.

## ■ Tabs and Spatial Arrangements

When you want to present spatially arranged material with MathematiX, you can use both tabs and tix vertical tools. We refer to the TAB DEMO# and TVT DEMO# chapters from the MathematiX Sample Data disk as we explain how you do it. The first sample in TAB DEMO# is a matrix with three rows and three columns. The general approach for formatting matrices is also useful for determinants, systems of simultaneous equations, and other spatial material.

## *Formatting Matrices*

A matrix is an array of elements arranged in rows and columns, enclosed within a single pair of very large braces, brackets, or parentheses. In the standard inkprint notation, these very large symbols of enclosure span all the lines in the matrix. Printing a matrix in MathematiX presents two challenges: creating the very large symbols of enclosure, and aligning the elements in each column. As the chart in Part 6 shows, MathematiX's "enlarged" symbols of enclosure are only two characters tall. As a substitute for these inkprint symbols of enclosure, use MathematiX's enlarged symbols of enclosure at the beginning and end of each line. This legibly suggests the standard inkprint notation. If your inkprint readers will be encountering this for the first time, include a note alerting them to the substitution.

To align the material in each column, you use BEX tabs and horizontal position commands. Here's the general approach we took for sample number 1 in **TAB DEMO#**. Before setting any tab stops, we clear all previous tab stops with **$$tc**. We set all the tab stops while entering the top row. Of course, the top row is not always an ideal model for the remaining tab stops: some elements below the top element may be longer. You must figure out the longest element in each column and take this into account when establishing the columns.

To set the tab for a column, we place **$$t*** right before the beginning of that column's material, establishing a tab at the current horizontal position on the line. We set the first tab stop for the position of the enlarged left bracket. We set the other tab stops for one space beyond the longest element in the previous column. To leave room for the longest element in a column, we use **$$p+#** to skip ahead some number of characters. The # value is equal to the number of additional characters in the column's longest element. If the elements in the final column vary greatly in length, it helps to set a tab stop for the close bracket, but we did not do that here.

On subsequent lines we use a disappearing space followed by a tab command **_ $$** (dots 4-5-6, space, dots 1-2-4-6, dots 1-2-4-6, space), to move to the beginning of a column. This novel combination ensures that the tab command moves where you want it. If the space after an element already brings you to the next tab stop, then the standard **$$** skips over to the *next* tab stop. For example, when your input is:

```
$l 111 $$t* 20 $$t* 3333 $$t* 44 $$t* 55
$l aaa_ $$ bb_ $$ c_ $$ ddd_ $$ e
$l aaa $$ bb $$ c $$ ddd $$ e
```

Then the tixed result is:

```
111 20 3333 44 55
aaa bb c    ddde
aaa    bb   c  ddd e
```

This combination of the disappearing space and tab command only works for Math Output. If you printed a tix-ready chapter that includes _ $$ , you would see the single dots 4-5-6 or underbar followed by however many spaces the tab stop creates.

In greater detail, the longest element in the first column is two characters: `{-1}`. This appears in the second row, not the top row where we're setting tab stops. That's why we put **$$p+1** to move ahead one character after the 3 in the first column, top row. We then use **$$t*** to set the tab stop for the second column. In the second column, the longest element is the `{-5}` in the top row—we don't need any **$$p+#** command to leave room for additional characters. We just leave one space between the second and third columns, and then use **$$t*** to set the tab stop for the third column. The longest element in the third column is the **-3** in the second row; since the element in the first row has only one character, we leave a space before the enlarged right bracket.

We move to the second line and then use **$$** to position the enlarged left bracket. _ **$$** takes us to the start of the second column and then the third column. After the 2 in the second column, we do not really need the disappearing space before the tab, as there is one space to spare. But because it does not hurt, we use the disappearing space-tab combination everywhere to be free from thinking about it. Since the `{-3}` in the third column has two characters, the length we allocated for the column, we don't need to leave a space before the enlarged right bracket. The third row is similar. Because the fraction in the third column has the width of just one character and we allocated two characters for the third column, we leave a space before the enlarged right bracket.

### *Other Applications of the Matrix Technique*

A determinant is written just like a matrix, except the elements are enclosed by vertical bars instead of brackets. This is no problem for MathematiX: begin and end each row with `{,\}`, an enlarged vertical bar. When you write systems of simultaneous equations, you may want to line up the coefficients or the variables in each line. You can use tab stops just as we did for lining up elements in a matrix.

Sample 2 in **TAB DEMO#** demonstrates the use of tabs in formatting a piecewise defined function, which gives different rules for defining the function values on different parts of the domain. In print each rule is written on a separate line, followed by a comma and then the applicable part of the domain. The entire arrangement is enclosed on the left by a very large left brace. In MathematiX we recommend beginning each row in the pattern with the enlarged left brace `{.,(}`. We clear out tab stops with **$$tc**, then set a tab at the position of the first enlarged left brace. We write the top row, then move down for the next row with `{ $1 }`. We use the **$$** tab command to position the enlarged left brace on the second row, write the second row, and repeat it all again for the third row.

Sample 3 in **TAB DEMO#** shows an even simpler approach to formatting the same thing. Since the definition follows some introductory text, we used **$$ml*** to establish the left margin at the position of the first enlarged left brace. Then we write the top row and move down for the next row with `{ $1 }`. Because the top row left margin has been established at the enlarged left brace, we are in position to write the second and third rows without messing with tabs at all. After the period at the end of the definition, we reset the left margin to zero to correctly position the question on the next line.

## ■ Formatting Spatial Arrangements with Tix Vertical Tools

Sample 1 in **TVT DEMO#** shows how to construct the by-now familiar piecewise defined function with tix vertical tools. This approach requires the extra work of deciding how much to move up and down, since you are bypassing MathematiX's automatic routines that ensure adequate vertical space between lines. However, tix vertical tools make it possible to line up the arrangement with other text before and after it.

Our goal was to align the preceding and following text with the middle row of the piecewise function definition. We begin with the introductory text. Then we save the current position (where the middle row will begin) and move up to place the top row. After we write the top row out, we restore the saved position—we're back at the middle-row starting point. Now we save it again, move down and write the bottom row. Then we restore the saved position to return to the beginning of the middle row. We then write this middle row and continue with a period and the next sentence. MathematiX is smart enough to leave room for the top and bottom rows when it positions the middle row and moves down for the line after the spatial arrangement.

In greater detail, just before the spatial arrangement we used a disappearing space after the equal sign so that we could save the position one space before the enlarged left brace. This improves the readability of the tix-ready chapter, as it allows us to begin each row in the arrangement with a space. Checking the chart in Part 6, we find that the top half of the middle enlarged brace and the bottom half of the top enlarged brace total two characters high. To add some gap between them for readability, we moved up two-and-three-quarters characters for the top row; this leaves enough room for the fraction in the top row. We did this movement in two steps, moving up three characters and then down one-quarter character. Similarly we moved down two-and-three-quarters characters for the bottom row, in two steps.

Using the tix vertical tools in this order requires you to write the actual math material out of order. There's another way to do it. Save the position at the middle row starting point, move up, write the top row, then restore. Back at the middle row, save the position, write the middle row, establish a tab stop at the end, then restore. Finally, back at the middle row one last time, save the position, write the bottom row, restore, then issue a tab command to take you past the end of the middle row. Establishing and using the tab stop is critical in this method, otherwise you won't advance horizontally. As we cautioned in Part 6, using the tix vertical tools within save-restore pairs requires planning to ensure that you actually move somewhere.

### *Tix Vertical Tools that Improve Vertical Positioning*

As discussed in Part 6, MathematiX places the fraction line as if the numerator has nothing below its baseline. A numerator with subscripts or with a directly below indicator requires special handling. In **TVT DEMO#**'s Sample 2, the first fraction has a subscript in the numerator. After the begin fraction indicator we use the tix vertical tool { **[<h** } to move the numerator up one-half character, to leave room for the subscript. Because we moved up inside of the numerator, the fraction line restores the correct vertical position.

Similarly, in Sample 3 the last fraction has a numerator which includes a directly below indicator. After the begin fraction indicator we use a tix vertical tool to move the numerator up by one character. This leaves room for the below expression—x, right arrow, a—which requires the height of one character. Also notice the use of **$$kb $$kf** pairs to ensure that this last equation is not split between two lines.

As also mentioned in Part 6, when tixing an above or below expression (other than an overbar or underbar), MathematiX places it as if its height is one character. In Sample 4 the below expression has a superscript and therefore has an extra half a character in its height. Therefore, after the below indicator we use the tix vertical tool { **[%h** } to move the below expression down by half a character. Because we moved down inside of a below expression, the termination indicator restores the correct vertical position.

### *Combining Tabs and Tix Vertical Tools*

The number line shown as sample 5 in **TVT DEMO#** uses tabs to line up the zeros of { **f' (x)** } with the appropriate x values. To draw the number line at its proper length, we use an underbar modified expression for the row of plus and minus signs and zeros. We place an arrow head on both ends of this line by using the tix vertical tools to move down three-quarters of a character (in two steps).

# Section 10: MathematiX and Extended Disk Systems

# Section 10
# MathematiX and Extended Disk Systems

You can use the Sider hard disk system, manufactured by First Class Peripherals, with both BEX 2.2 and 3.0. Beginning with version 3.0, BEX lets you use a variety of storage devices for the BEX software and your data. When your Apple has more than 128K memory, BEX 3.0 lets you configure the memory as super-fast electronic disks, or *RAM drives*. BEX 3.0 lets you store data on 3.5-inch disks, and you can also use these handy disks for MathematiX data. You can't copy the MathematiX software to a 3.5-inch disk.

## 1 RUNNING MATHEMATIX FROM A SIDER VOLUME

There are several drawbacks to running MathematiX from a Sider volume. You must create an automatic procedure chapter that tells BEX where to look for the "program drive." Every time you boot BEX on the Sider, you have to use this auto chapter to get access to MathematiX. Since Sider disk access is no faster than floppy disk access, the main advantage of including the Sider in your use of MathematiX is not having to insert and remove floppy disks. If you can stand to run BEX from floppy disks, then simply define a BEX configuration that doesn't reference the Sider, and follow the floppy disk procedures for MathematiX. But if you insist, this Part explains how to do it.

### *Copy MathematiX Software to Single Sider Volume*

The first step is copying every file from the MathematiX program disk to one Sider volume, using FID. Make note of the Sider volume number. For the sake of this example, you copy the MathematiX software on to Sider volume 9. When you copy the MathematiX software to a different Sider volume, substitute that number for the *9* in the samples that follow.

### *Creating the* MTX *Automatic Procedure*

Boot up from the Sider and get to BEX's Main Menu. In order to use MathematiX, you must create an automatic procedure chapter that tells BEX to go to the Math Menu when you press the spacebar. You only create the auto chapter once; you then *use* the auto chapter each time you want to run MathematiX from the Sider.

The auto chapter begins with a "magic character," which depends on your BEX version. Then comes the number 486, followed by one <CR>, the Sider volume number, finished up with another <CR>. For BEX 2.2, this "magic character" is **control-6**; here's the step by step for creating the auto chapter:

```
Main: control-R
Start remember mode
Main: control-6 486 <CR> 9 <CR>
Main: control-S
Auto chapter: 1MTX <CR>
```

```
Main:
```

For BEX 3.0, you use **control-backslash** as your "magic character:"

```
Main: control-R
Start remember mode
Main: control-\ 486 <CR> 9 <CR>
Main: control-S
Auto chapter: 1MTX <CR>
Main:
```

The keystrokes you've stored in the **MTX** chapter on virtual drive 1 tell BEX that pressing the spacebar means move to the Math Menu on Sider volume 9. When you copy the MathematiX software to a different Sider volume, substitute that volume number for the digit *9* in the sample above.

### *Using MathematiX on the Sider*

Every time you boot BEX from the Sider, you must run the **MTX** auto chapter in order to get to the MathematiX software. At the Main Menu, press **control-A**, then type **1MTX <CR>** at the **Auto chapter:** prompt. Here's a summary of how you navigate the BEX and MathematiX menus on the Sider:

| From here | To here | Key(s) to press |
|---|---|---|
| Starting | Main | spacebar |
| Main | Math | (after you've run **MTX**) spacebar |
| Math | Main | J |
| Math | Second or Page | J then S or Z |
| Main | Starting | spacebar to Math, then spacebar <CR> to Starting |

When you're at the Math Menu, virtual drive 1 is the Sider volume where the MathematiX software resides. To access the Sider volume where the Main side software resides, use the asterisk as a drive "number"—details below under "Redirecting Virtual Drive 1."

## 2 LOADING MATHEMATIX ON A RAM DRIVE

This feature requires BEX 3.0. As introduced in Section 6, during tixing, MathematiX must read software from the program disk twice for each BEX page. Since disk access to RAM drives is electronic, loading the MathematiX software on a RAM drive dramatically increases MathematiX's speed. To run *both* the Main side of BEX and MathematiX on RAM, you must have two RAM drives in your configuration. When you have just one large RAM drive from a slot 1-7 memory card, you must choose which set of programs to run from RAM: either the Main side or the MathematiX software.

In Section 1, we showed the "JOHN" configuration for an Apple IIgs 512K with three RAM drives, one 5.25-inch floppy drive, and one 3.5-inch disk drive. The virtual drives are set up like this:

```
Which drive: 6 ? <CR>
1 is slot 3 drive 1
2 is slot 6 drive 1
3 is slot 5 drive 1
4 is slot 5 drive 3
5 is slot 3 drive 3
6 is slot 3 drive 2
```

In order to load the Main side software on RAM drive, virtual drive 1 must be a RAM drive, in this case, slot 3, drive 1. To also load the MathematiX software on RAM drive, all you have to do is define another RAM drive in your configuration. It doesn't matter what virtual drive number you assign to this drive: MathematiX looks for the second RAM drive in your list of virtual drives when it comes time to copy the software.

The numbering and size of your RAM drives depends on how much memory you've installed in your Apple. The final RAM drive—the highest number on the drive list—may be a little smaller than the others. You can't store the MathematiX software on the final RAM drive if you have 256K or 1024K of memory, since it holds only one-third as much.

In this sample configuration, the second RAM drive is slot 3, drive 3, which BEX addresses as virtual drive 5. Notice that the full addresses of virtual drives 5 and 6 are out of order. When you have 512K or 768K memory, then the last RAM drive holds two-thirds as much as a standard one, and you can copy the MathematiX software to it with room to spare. Since the *second* RAM drive always holds the MathematiX software, the "out of order" arrangement makes more room for data on virtual drive 6.

### *Copy from Disk with* Control-D

The Main side software is automatically copied to virtual drive 1 when you move from the Starting to the Main Menus. The loading of the MathematiX software on to RAM drive doesn't happen automatically. You follow these four steps to copy the software:

1 Get to the Main Menu.

2 Place MathematiX disk in the booting drive, generally slot 6, drive 1.

3 Press the spacebar to move to the Math Menu.

4 At the Math Menu, press **control-D**.

MathematiX begins copying all the files and chapters from the MathematiX disk to the RAM drive; when it's finished, you are switched over to the newly-copied software.

### *Navigating the RAM Drive Menus*

As you can see from the following table, moving between the Main and Math Menus is straightforward. However, you can no longer move directly from the Main Menu to the Starting Menu; you have to detour through the Math Menu:

| From here | To here | This disk in booting drive | Key(s) to press |
|---|---|---|---|
| Main | Math | none required | spacebar |
| Math | Main | none required | J |
| Main | Starting | BEX Boot Disk | spacebar to Math, then spacebar <CR> to Starting |
| Starting | MathematiX on RAM | MathematiX Menu Disk | spacebar to Main, then spacebar to MathematiX on disk, then **control-D** |

Please note that you *always* use **control-D** when you go from the Starting Menu to the MathematiX RAM drive. If you don't use **control-D** after a trip to the Starting Menu, then MathematiX uses the software from the floppy disk. The first time you use **control-D**, MathematiX copies software from disk. Until you turn off the Apple, subsequent times, you're just telling BEX where to find the MathematiX software.

## Redirection of Virtual Drive 1: The Asterisk Drive Specification

As stressed in Master Level Section 3, the current program drive is always virtual drive 1. The Main side RAM drive is virtual drive 1 when you're at the Main Menu. When you move to the Starting Menu, BEX redirects virtual drive 1 to the floppy disk in slot 6, drive 1. At the Starting Menu, you can catalog the Main Side RAM drive with the letter **M** at the **Which drive:** prompt.

When you're at the Math Menu, the MathematiX RAM drive becomes virtual drive 1. MathematiX introduces a new drive specification to let you catalog or work with chapters on the Main Side RAM drive. When you're at the Math Menu, use the asterisk * character to refer to the Main side RAM drive. You can use asterisk alone for a catalog, or as a "drive number" when scanning or naming chapters. For example, ***SAVE** refers to a SAVE chapter on the Main side RAM drive. Entering ***A#** at the **Naming method:** prompt would add the number sign character to chapters written on the Main side RAM drive.

# Appendices

# Appendix A
# Verbalize Vocabulary

When Verbalize uses a term you're unfamiliar with, check here. This alphabetical list includes the terminology Verbalize uses when interpreting your tix-ready chapters, as well as cross-references from the standard Nemeth terminology to the Verbalize version.

## A

**Above** Precedes material that will be printed directly over other material, created from the directly-over indicator `{<}` within a modified expression. (Examples in Section 5, Part 8.) See also "start modified expression," "below," and "terminate modified expression"; contrast with "superscript."

**Aleph** The first letter of the Hebrew alphabet, entered as `{,,a}` (Examples in Section 5, Part 11.)

**Alpha** Greek letter a, entered as `{.a}`.

**Angstrom Units** Units of measurement shown in inkprint as a capital A with a hollow dot above it, entered as `{@,a}`.

## B

**Backslash** An inkprint diagonal line from the upper left to the lower right, created from `{_*}`.

**Baseline** Indicates the end of a superscript or subscript. Most inkprint signs share a common vertical reference; signs appearing above or below this baseline are interpreted as superscripts or subscripts. Created by a space, multipurpose indicator, or punctuation indicator. (Examples in Section 5, Part 4.)

**Below** Precedes material that will be printed directly under other material, created from the directly-under indicator `{%}` within a modified expression. (Examples in Section 5, Part 8.) See also "start modified expression," "above," and "terminate modified expression"; contrast with "subscript."

**Bet** The second letter of the Hebrew alphabet, entered as `{,,b}`. (Examples in Section 5, Part 11.)

**Beta** Greek letter b, entered as `{.b}`.

**Binomial of** Precedes a binomial expression, created from `{@b(}`. A binomial is a mathematical expression shown as one quantity written above another quantity, enclosed within enlarged parenthesis. See also "close binomial" and "over."

**Bold face** Inkprint letters that are thicker and darker than normal. Create a bold face capital x from `{_;,x}`.

**Brace** Symbol of enclosure, also known as "curly brace"; open and close braces are entered with `{.(}` and `{.)}`.

**Bracket** Symbol of enclosure, also known as "square bracket"; open and close brackets are entered with `{@(}` and `{@)}`.

## C

**Chi** A Greek letter, entered as {.&}.

**Close binomial** Follows a binomial expression, created from {)} which has been preceded by {%} which in turn has been preceded by {@b(}. See also "binomial of" and "over."

**Close brace** See "brace."

**Close bracket** See "bracket."

**Close paren** See "paren."

**Complex** Adjective that may modify a fraction's start, line, or end, created from {,?}, {,/}, and {,#}; results in greater vertical spacing when tixed.

**Cubed** A superscript of 3, created from any character followed by {^3 }. Compare with "superscript" and "squared."

**Cube root of** An index of radical value of three, tixed as an elevated 3 followed by the inkprint radical sign, created from {<3>}. See also "terminate cube root."

## D

**Degree** A hollow dot in the superscript position. For example, 90 degrees is entered as {#90^.*}.

**Delta** Greek letter d, entered as {.d}.

**Directly over** See "above."

**Directly under** See "below."

**Divided by** The inkprint symbol used in arithmetic to show division, entered as {./}.

**Dot** A centered dot which often represents multiplication, entered as {*}.

**Double prime** Two angled strokes; often used to represent seconds or inches, entered as {''}. See also "prime."

**Double tilde** Two wavy horizontal lines stacked vertically, entered as {@:@:}. Often means "approximately equal to." Compare "tilde."

## E

**End fraction** Follows the denominator material and implies a return to the vertical position before the fraction. Created from {#} when preceded by start fraction indicator and fraction line. See also "complex" and "hypercomplex."

**Enlarged** A taller version of a symbol of enclosure. For example, "enlarged braces" are suitable for enclosing spatial fractions. (Examples in Section 4, Part 4.)

**Epsilon** Greek letter e, entered as {.e}.

**Equals sign above** An equals sign above a sign of comparison. Created by placing {.k} before a sign of comparison.

**Equals sign under** An equals sign below a sign of comparison, created by placing {.k} after a sign of comparison.

**Eta** A Greek letter, entered as { . : }.

## F

**Factorial** An inkprint sign that looks like an exclamation mark, entered as {&}.

**Fraction line** Indicates end of numerator material and beginning of denominator material. Tixed as horizontal line slightly longer than the longest sign in numerator or denominator. Created from {/} when preceded by start fraction indicator and followed by end fraction indicator. See also "complex" and "hypercomplex."

**From** A subscript that follows an integral sign. Entered as { ; } that is preceded by { ! }. See also "upto" and "subscript."

## G

**Gamma** Greek letter g, entered as { . g}.

## H

**Hollow dot** A centered small circle, entered as { . * }. Also see "degree."

**Horizontal bar above** A horizontal bar above a sign of comparison. Created by placing { : } before a sign of comparison.

**Hypercomplex** Adjective that may modify a fraction's start, line, or end. Created from { , , ?}, { , , / }, and { , , #}; results in greater vertical spacing when tixed.

## I

**Index of radical** Precedes the value defining which mathematical root is being extracted from a radical. Tixed as an elevated value followed by the inkprint radical sign, entered as {<}. See also "of" and "terminate radical."

**Inner radical** A radical inside a radical, created by { . } placed before one of the following: {>}, {]}, or {<}.

**Integral** An infinite sum represented by a tall, thin, stylized "S"; created by { ! }.

**Iota** Greek letter entered as { . i }.

## K

**Kappa** Greek letter k; created by { . k }. MathematiX uses spaces or tix null tool to distinguish between equals sign and kappa. (Examples in Section 5, Part 11.)

**Koph** Obsolete Greek letter entered as { . q}.

## L

**Lambda** Greek letter l, entered as { . l }.

**Literary braille** English text encoded with grade 2 braille contractions. MathematiX requires the literary translation tools { @l } or {_ @l } before it can interpret literary braille as such. (Examples in Section 3, Part 2.) Compare with "math braille."

## ■ M

**Math braille** Mathematical notation encoded in Nemeth braille. MathematiX starts out interpreting your material as math braille. After a transition to literary braille, MathematiX requires the math translation tools { **@m** } or {_ **@m** } before it interprets math braille as such. Compare with "literary braille."

**Modified expression** See "start modified expression" and "terminate modified expression."

**Mu** Greek letter m, entered as {**.m**}.

## ■ N

**New line** Indicates presence of { **$l** } or hard <CR> while in math mode.

**New page** Indicates presence of { **$f** } new page indicator while in math mode.

**New paragraph** Indicates presence of { **$p** } paragraph indicator while in math mode. Compare with "paragraph mark."

**Nu** Greek letter n, entered as {**.n**}.

## ■ O

**Of** The separator between the index of a radical and the radical, entered as a {**>**} preceded by a {**<**}. See also "index of radical" and "terminate radical."

**Omega** Greek letter w, entered as {**.w**}.

**Omicron** Greek letter o, entered as {**.o**}.

**Open brace** See "brace."

**Open bracket** See "bracket."

**Open paren** See "paren."

**Output move down** Created from the tix vertical tool that shifts the inkprint vertical position. For example, {**[%h**} is verbalized as "output move up half character." (Examples in Section 4, Part 5 and Section 9, Parts 6 and 7.)

**Output move up** Created from the tix vertical tool that shifts the inkprint vertical position. For example, {**[<q**} is verbalized as "output move up quarter character." (Examples in Section 4, Part 5 and Section 9, Parts 6 and 7.)

**Over** Separates the two quantities in a binomial expression. A binomial is a mathematical expression shown as one quantity written above another quantity, enclosed inside of enlarged parentheses, Created from {**%**} which has been preceded by {**@b(**}. See also "binomial of" and "close binomial."

**Overbar** See "with overbar" and "terminate overbar expression."

## P

**Paragraph indicator** The four characters { `$p` } that create a new output paragraph when printing or tixing. Verbalized as "dollar sign, p" in literary mode and "new paragraph" in math mode.

**Paragraph mark** An inkprint sign like an uppercase P with two parallel vertical strokes, created from {`@,p`}. Contrast with "paragraph indicator" and "new paragraph."

**Parallel to** Two parallel vertical lines, created from {`$1`}. MathematiX uses spaces or tix null tool to distinguish between parallel to symbol and new output line indicator. (Examples in Section 4, Part 2.)

**Paren** Short for parenthesis. Open and close parens are entered with {`(`} and {`)`}.

**Perpendicular to** Inkprint sign like upside down uppercase T, created from {`$p`}. MathematiX uses spaces or tix null tool to distinguish between perpendicular to symbol and new paragraph indicator. (Examples in Section 4, Part 2.)

**Phi** Greek letter f, entered as {`.f`}.

**Pi** Greek letter p, entered as {`.p`}. Pi often represents the quantity 3.14159...

**Point** The decimal point, entered as {`.`}.

**Pounds Sterling** An inkprint symbol used to represent English currency, created from {`@1`}. MathematiX uses spaces or tix null tool to distinguish between pounds Sterling symbol and literary translation tool. (Examples in Section 4, Part 2.)

**Prime** An angled stroke; often used to represent minutes or feet, entered as {`'`}. See also "double prime."

**Product** An inkprint symbol consisting of an enlarged capital pi, entered as {`.,,p`}.

**Psi** Greek letter y, entered as {`.y`}.

## R

**Radical** An inkprint mathematical symbol that shows a mathematical root. See "square root of," "terminate square root," "cube root of," "terminate cube root," "index of radical," "of," and "terminate radical."

**Rho** Greek letter r, entered as {`.r`}.

## S

**Sampi** An obsolete Greek letter, entered as {`.c`}.

**Sigma** Greek letter s, entered as {`.s`}.

**Slash** An inkprint diagonal line from the lower left to the upper right, entered as {`_/`}.

**Squared** A superscript of 2, created from any character followed by {`^2 `}. Compare with "superscript" and "cubed."

**Square root of** The default radical, implies an index of 2, created from {>}. See also "terminate square root."

**Start fraction** Precedes the numerator of a fraction. Material up to the "fraction line" keyword is the numerator, tixed above the fraction line; numerator and denominator are centered horizontally relative to the fraction line. Created from {?} followed by fraction line and end fraction indicator. See also "fraction line," "end fraction," "complex," and "hypercomplex."

**Start modified expression** Indicates the beginning of some material which has other signs printed above and/or below it, created from {"}. (Examples in Section 5, Part 8.) See also "above," "below," "terminate overbar expression," "terminate underbar expression," and "terminate modified expression."

**Subscript** Begins an expression which is written below the base line, entered as {;}.

**Subscript with subscript** Begins a subscript of a subscript, an expression which is written below the subscript level, entered as {;;}.

**Subscript with superscript** Begins a superscript of a subscript, an expression which is written above the subscript level, entered as {;^}.

**Summation** An inkprint symbol consisting of an enlarged capital sigma, entered as {.,,s}.

**Superscript** Begins an expression which is written above the base line, entered as {^}.

**Superscript with subscript** Begins a subscript of a superscript, an expression which is written below the superscript level, entered as {^;}.

**Superscript with superscript** Begins a superscript of a superscript, an expression which is written above the superscript level, entered as {^^}.

## ■ T

**Tau** Greek letter t, entered as {.t}.

**Terminate cube root** The end of a cube root, created from {]} preceded by {<3>}. See also "cube root of."

**Terminate modified expression** The end of a modified expression, created from the {]} termination indicator.

**Terminate overbar expression** The end of a modified expression that has a horizontal bar over it. Created from {<:]} that is preceded by a {"}. Also see "start modified expression."

**Terminate radical** The end of a radical, created from {]} preceded by {>}, which in turn is preceded by {<}. See also "index of radical," and "of."

**Terminate square root** The end of a square root, created from {]} preceded by {>}. See also "square root of."

**Terminate underbar expression** The end of a modified expression that has a horizontal bar under it. Entered as {%:]} that is preceded by a {"}. Also see "start modified expression."

**Theta** A Greek letter, entered as {.?}.

**Tilde** A wavy horizontal line, slightly longer than a dash, created from `{@:}`. Often used to mean "similar"; may modify a sign of comparison.

## U

**Underbar** See "terminate underbar expression."

**Unknown quantity** Created from the general omission sign, entered as `{=}`. When tixing to inkprint, the result is a question mark.

**Upper case lock** Precedes an entire capitalized word. Uppercase lock is ended with a space, number, or punctuation. Entered as `{,,}`.

**Upsilon** Greek letter u, entered as `{.u}`.

**Upto** A superscript that follows an integral sign, entered as `{^}` that is preceded by `{!}`. See also "from" and "superscript."

## V

**Vau** Obsolete Greek letter, entered as `{.v}`.

## W

**With equals sign below it** An equals sign below a sign of comparison, created by placing `{.k}` after a sign of comparison.

**With horizontal bar below it** A horizontal bar below a sign of comparison, created by placing `{:}` after a sign of comparison.

**With overbar** A horizontal bar placed over a character, created by placing `{:}` after a character. For example, x with overbar is created by `{x:}`.

## X

**Xi** Greek letter x, entered as `{.x}`.

## Z

**Zeta** Greek letter z, entered as `{.z}`.

# Appendix B
# Nemeth Symbols Supported by MathematiX in Transcriber Order

The following pages show every Nemeth symbol that MathematiX supports. This information parallels Appendix B in the Nemeth Braille Code for Mathematics and Science Notation, presenting the symbols "alphabetized" in transcriber order. MathematiX-specific requirements, in italics, follow the symbol's name(s). To save space, we omit function names.

■ **1** {a} **(dot 1)**

{a} ------------------ English letter a, also (with appropriate preceding indicator) German ah, Greek alpha, and Hebrew aleph

■ **2** {b} **(dots 1-2)**

{b} ------------------ English letter b, also German beh, Greek beta, and Hebrew bet

■ **3** {c} **(dots 1-4)**

{c} ------------------ English letter c, also German tseh and Greek sampi

■ **4** {d} **(dots 1-4-5)**

{d} ------------------ English letter d, also German deh and Greek delta

■ **5** {e} **(dots 1-5)**

{e} ------------------ English letter e, also German eh and Greek epsilon

■ **6** {f} **(dots 1-2-4)**

{f} ------------------ English letter f, also German eff and Greek phi

■ **7** {g} **(dots 1-2-4-5)**

{g} ------------------ English letter g, also German gheh and Greek gamma

■ **8** {h} **(dots 1-2-5)**

{h} ------------------ English letter h, also German hah

■ **9** {i} **(dots 2-4)**

{i} ------------------ English letter i, also German ee and Greek iota

■ **10** {j} **(dots 2-4-5)**

{j} ------------------ English letter j, also German yaht

■ **11** {k} **(dots 1-3)**

{k} ------------------ English letter k, also German kah and Greek kappa

■ **12** {l} **(dots 1-2-3)**
{l}------------------ English letter l, also German ell and Greek lambda

■ **13** {m} **(dots 1-3-4)**
{m}------------------ English letter m, also German em and Greek mu

■ **14** {n} **(dots 1-3-4-5)**
{n}------------------ English letter n, also German en and Greek nu

■ **15** {o} **(dots 1-3-5)**
{o}------------------ English letter o, also German oh and Greek omicron
{o}------------------ right arrow head *part of arrow*

■ **16** {p} **(dots 1-2-3-4)**
{p}------------------ English letter p, also German peh and Greek pi

■ **17** {q} **(dots 1-2-3-4-5)**
{q}------------------ English letter q, also German koo and Greek koph

■ **18** {r} **(dots 1-2-3-5)**
{r}------------------ English letter r, also German err and Greek rho

■ **19** {s} **(dots 2-3-4)**
{s}------------------ English letter s, also German ess and Greek sigma

■ **20** {t} **(dots 2-3-4-5)**
{t}------------------ English letter t, also German teh and Greek tau

■ **21** {u} **(dots 1-3-6)**
{u}------------------ English letter u, also German oo and Greek upsilon

■ **22** {v} **(dots 1-2-3-6)**
{v}------------------ English letter v, also German fao and Greek vau

■ **23** {x} **(dots 1-3-4-6)**
{x}------------------ English letter x, also German iks and Greek xi

■ **24** {y} **(dots 1-3-4-5-6)**
{y}------------------ English letter y, also German ypsilon and Greek psi

■ **25** {z} **(dots 1-3-5-6)**
{z}------------------ English letter z, also German tset and Greek zeta

■ **26** {&} **(dots 1-2-3-4-6)**

{&} ------------------ factorial

■ **27** {=} **(dots 1-2-3-4-5-6)**

{=} ------------------ omission sign *tixed as question mark*

■ **28** {(} **(dots 1-2-3-5-6)**

{(} ------------------ left parenthesis

■ **29** {!} **(dots 2-3-4-6)**

{!} ------------------ integral

{!!}----------------- double integral

{!!!}--------------- triple integral

■ **30** {)} **(dots 2-3-4-5-6)**

{)} ------------------ close binomial expression

{)} ------------------ right parenthesis

■ **31** {*} **(dots 1-6)**

{*} ------------------ dot end of arrow *part of arrow*

{*} ------------------ dot, or times

■ **32** {<} **(dots 1-2-6)**

{<} ------------------ directly-over indicator (first order)

{<} ------------------ index-of-radical indicator

{<lim}------------- upper limit

{<limit} --------- upper limit

{<:]}--------------- terminate modified expression with overbar

■ **33** {%} **(dots 1-4-6)**

{%} ------------------ directly-under indicator (first order)

{%lim}------------- lower limit

{%limit} --------- lower limit

{%:]}--------------- terminate modified expression with underbar

■ **34** {?} **(dots 1-4-5-6)**

{?} ------------------ opening simple-fraction indicator

## ■ 35 {:} (dots 1-5-6)

| | |
|---|---|
| {:} | horizontal bar, or macron |
| {:@%} | bar over logical product |
| {:@%:} | bar over and bar under logical product |
| {:@%.k} | bar over and equals sign under logical product |
| {:@:} | bar over tilde |
| {:@:@:} | bar over double tilde |
| {:@+} | bar over logical sum |
| {:@+:} | bar over and bar under logical sum |
| {:@+.k} | bar over and equals sign under logical sum |
| {:_"k} | bar over inclusion sign |
| {:_.1} | bar over reverse inclusion sign |
| {:"k} | bar over less than sign |
| {:.1} | bar over greater than sign |

## ■ 36 {$} (dots 1-2-4-6)

| | |
|---|---|
| {$} | shape indicator |
| {$a} | concave up arc |
| {$c} | circle |
| {$e} | ellipse, or oval |
| { $f } | new output page *BEX format indicator* |
| {$l} | parallel to |
| { $l } | new output line *BEX format indicator* |
| {$o} | right arrow (contracted form) |
| {$p} | perpendicular to |
| { $p } | new output paragraph *BEX format indicator* |
| {$r} | rectangle |
| {$s} | star |
| {$t} | triangle |
| {$<o} | up arrow (contracted form) |
| {$<[o} | bidirectional vertical arrow (contracted form) |
| {$<[33o} | bidirectional vertical arrow |
| {$<33o} | up arrow |
| {$%o} | down arrow (contracted form) |
| {$%33o} | down arrow |
| {$$} | begin BEX format command |
| {$[} | angle |
| {$[o} | bidirectional horizontal arrow (contracted form) |
| {$[33} | left arrow |
| {$[33o} | bidirectional horizontal arrow |
| {$33o} | right arrow |
| {$4} | square |

| | |
|---|---|
| {$7o} | right double arrow |
| {$'} | concave down arc |
| {$_4} | filled square |

■ **37** {]} **(dots 1-2-4-5-6)**

| | |
|---|---|
| {]} | terminate modified expression |
| {]} | terminate radical or square root |
| {]} | restore saved output position *tix vertical tool* |

■ **38** {\} **(dots 1-2-5-6)**

| | |
|---|---|
| {\} | vertical bar |
| {\} | vertical bond *chemistry bond diagram* |
| {\\} | double vertical bar |
| {\\} | vertical double bond *chemistry bond diagram* |
| {\\\} | triple vertical bar |
| {\\\} | vertical triple bond *chemistry bond diagram* |
| {\\"\\} | double vertical bar followed by double vertical bar |
| {\"\} | vertical bar followed by vertical bar |

■ **39** {[} **(dots 2-4-6)**

| | |
|---|---|
| {[} | mathematical comma at subscript or superscript level |
| {[} | left arrow head *part of arrow* |
| {[<d} | move up two characters *tix vertical tool* |
| {[<f} | move up four characters *tix vertical tool* |
| {[<h} | move up half of a character *tix vertical tool* |
| {[<o} | move up one character *tix vertical tool* |
| {[<q} | move up quarter of a character *tix vertical tool* |
| {[<t} | move up three characters *tix vertical tool* |
| {[<x} | move up one and a half characters *tix vertical tool* |
| {[%d} | move down two characters *tix vertical tool* |
| {[%f} | move down four characters *tix vertical tool* |
| {[%h} | move down half of a character *tix vertical tool* |
| {[%o} | move down one character *tix vertical tool* |
| {[%q} | move down quarter of a character *tix vertical tool* |
| {[%t} | move down three characters *tix vertical tool* |
| {[%x} | move down one and a half characters *tix vertical tool* |

■ **40** {w} **(dots 2-4-5-6)**

| | |
|---|---|
| {w} | English letter w, also German veh and Greek omega |

■ **41** {1} **(dot 2)**

| | |
|---|---|
| {1} | numeral 1 |

■ **42** {2} **(dots 2-3)**

| | |
|---|---|
| {2} | numeral 2 |

■ **43** {3} **(dots 2-5)**

| | |
|---|---|
| {3} | numeral 3 |
| {33} | arrow shaft *part of arrow* |

■ **44** {4} **(dots 2-5-6)**

| | |
|---|---|
| {4} | numeral 4 |

■ **45** {5} **(dots 2-6)**

| | |
|---|---|
| {5} | numeral 5 |

■ **46** {6} **(dots 2-3-5)**

| | |
|---|---|
| {6} | numeral 6 |

■ **47** {7} **(dots 2-3-5-6)**

| | |
|---|---|
| {7} | numeral 7 |

■ **48** {8} **(dots 2-3-6)**

| | |
|---|---|
| {8} | numeral 8 |

■ **49** {9} **(dots 3-5)**

| | |
|---|---|
| {9} | numeral 9 |

■ **50** {0} **(dots 3-5-6)**

| | |
|---|---|
| {0} | numeral 0 |

■ **51** {/} **(dots 3-4)**

| | |
|---|---|
| {/} | simple fraction line |
| {/$l} | not parallel to |
| {/$p} | not perpendicular to |
| {/@e} | not an element of |
| {/,*} | it does not follow |

■ **52** {+} **(dots 3-4-6)**

| | |
|---|---|
| {+} | plus |
| {+-} | plus or minus |
| {+"-} | regular plus followed by regular minus |
| {+"_-} | regular plus followed by boldface minus |

### ■ 53 {#} (dots 3-4-5-6)

{#} ------------------ closing simple fraction indicator

{#} ------------------ numeric indicator *optional in math mode*

### ■ 54 {>} (dots 3-4-5)

{>} ------------------ radical, or square root

### ■ 55 {'} (dot 3)

{'} ------------------ prime, or minutes, or feet

{''}----------------- double prime, or seconds, or inches

{'''}--------------- triple prime

{  '''}------------- ellipsis

### ■ 56 {-} (dots 3-6)

{-} ------------------ hyphen

{-} ------------------ minus

{-+}----------------- minus or plus

{--}----------------- dash

{----}------------ long dash

{-"+}--------------- regular minus followed by regular plus

{-"_+}------------- regular minus followed by boldface plus

### ■ 57 {@} (dot 4)

{@  }----------------- discretionary line break *tix spacing tool*

{@a}----------------- at sign

{@b(}--------------- opening of binominal expression *MathematiX-only symbol*

{@c}----------------- cents sign

{@d}----------------- partial derivative

{@e}----------------- membership, or is an element of

{@express  }---- restore speech & suppress pauses when screen is full *tix verbalize tool*

{@h}----------------- crossed h

{@l}----------------- pounds Sterling

{  @l  }------------- begin literary translation mode *tix translation tool*

{  @m  }------------- restore math translation mode *tix translation tool*

{@notalk  }------ suppress speech output unless structure error *tix verbalize tool*

{@s}----------------- dollar sign

{@talk  } --------- restore speech and pause when screen is full *tix verbalize tool*

{@&}----------------- universal quantifier, or for every

{@=}----------------- existential quantifier, or there exists, or for some

{@=\}--------------- there exists uniquely, or there exists exactly one, or for exactly one

{@(}----------------- left square bracket

### *dot 4 continued*

| | |
|---|---|
| {@)} | right square bracket |
| {@*} | cross product, or times, or multiplication sign |
| {@<,<} | equivalence |
| {@%} | logical product |
| {@%:} | bar under logical product |
| {@%.k} | equals sign under logical product |
| {@:} | tilde |
| {@::} | bar under tilde |
| {@:@:} | double tilde |
| {@:@::} | bar under double tilde |
| {@:@:.k} | equals sign under double tilde |
| {@:"@:} | tilde followed by tilde |
| {@:.k} | equals sign under tilde |
| {@$} | crossed d |
| {@]} | general reference indicator |
| {@5} | reverse membership |
| {@0} | percent sign |
| {@/} | since, or because |
| {@+} | logical sum |
| {@+:} | bar under logical sum |
| {@+.k} | equals sign under logical sum |
| {@#} | asterisk |
| {@>} | check mark |
| {  @-  } | restore math translation mode *tix translation tool* |
| {@^(} | upper left half-bracket |
| {@^)} | upper right half-bracket |
| {@_0} | empty set *MathematiX-only symbol* |
| {@.l} | crossed lambda |
| {@;} | script typefont English letter *must precede each scripted letter* |
| {@;(} | lower left half-bracket |
| {@;)} | lower right half-bracket |
| {@;,} | uppercase script typefont English letter *must precede each scripted letter* |
| {@,a} | angstrom unit |
| {@,p} | paragraph symbol (inkprint sign) |
| {@,r} | crossed R |
| {@,s} | section mark |
| {@,s@,s} | double section mark |
| {@,(} | enlarged left square bracket |
| {@,)} | enlarged right square bracket |
| {@,:} | extended tilde |

## ■ 58 {^} (dots 4-5)

| | |
|---|---|
| {^} ------------------ | superscript indicator |
| {^^}----------------- | superscript with superscript indicator |
| {^^^}--------------- | superscript with superscript with superscript indicator |
| {^^; }--------------- | superscript with superscript with subscript indicator |
| {^; }----------------- | superscript with subscript indicator |
| {^;^}--------------- | superscript with subscript with superscript indicator |
| {^; ; }--------------- | superscript with subscript with subscript indicator |

## ■ 59 {_} (dots 4-5-6)

| | |
|---|---|
| {_} ------------------ | German letter indicator *must precede each German letter* |
| {_} ------------------ | punctuation indicator *required in math mode as listed here* |
| {_ }----------------- | disappearing space *tix spacing tool* |
| {_g] }--------------- | horizontal solid bond *chemistry bond diagram* |
| {_l}----------------- | identity, or is identical with |
| {_&}----------------- | ampersand |
| {_=}----------------- | variation, or varies as |
| {_=] }--------------- | horizontal triple bond *chemistry bond diagram* |
| {_*}----------------- | back slash |
| {_*}----------------- | oblique single bond (upper left to lower right) *chemistry bond diagram* |
| {_**}--------------- | oblique double bond (upper left to lower right) *chemistry bond diagram* |
| {_<}----------------- | caret, or circumflex |
| {_%}----------------- | inverted caret |
| {_] }----------------- | dagger |
| {_\}----------------- | boldface vertical bar |
| {__\}------------- | double boldface vertical bar |
| {_1}----------------- | literary comma *punctuation indicator required ir math mode* |
| {_2}----------------- | semicolon *punctuation indicator required in math mode* |
| {_3}----------------- | colon *punctuation indicator required in math mode* |
| {_3] }--------------- | horizontal single bond *chemistry bond diagram* |
| {_4}----------------- | period *punctuation indicator required in math mode* |
| {_6}----------------- | exclamation point *punctuation indicator required in math mode* |
| {_7] }--------------- | horizontal double bond *chemistry bond diagram* |
| {_8}----------------- | question mark *punctuation indicator required in math mode* |
| {_8}----------------- | open quote *punctuation indicator required in math mode* |
| {_0}----------------- | closing quote *punctuation indicator required in math mode* |
| {_0' }--------------- | closing inner quote *punctuation indicator required in math mode* |
| {_/}----------------- | oblique single bond (lower left to upper right) *chemistry bond diagram* |
| {_/}----------------- | slash, or diagonal line |

### *dots 4-5-6 continued*

{_//} -------------- oblique double bond (lower left to upper right) *chemistry bond diagram*

{_+} ---------------- boldface plus

{_+"-}------------- boldface plus followed by regular minus

{_+"_-}----------- boldface plus followed by boldface minus

{_#} ---------------- boldface numeric indicator *must precede each boldface digit*

{_'} ---------------- apostrophe *punctuation indicator required in math mode*

{_-} ---------------- boldface minus

{_-"+}------------- boldface minus followed by regular plus

{_-"_+}----------- boldface minus followed by boldface plus

{_@(} -------------- boldface left square bracket

{_@)} -------------- boldface right square bracket

{__]} -------------- double dagger

{_"k} -------------- inclusion sign, or is contained in, or is a subset of

{_"k:}------------- bar under inclusion sign

{_"k.k}----------- equals sign under inclusion sign

{_.k} -------------- boldface equals sign

{_.$} -------------- boldface del

{_.1} -------------- reverse inclusion sign, or contains

{_.1:}------------- bar under reverse inclusion sign

{_.1.k}----------- equals sign under reverse inclusion sign

{_;} ---------------- boldface English letter indicator *must precede each boldface letter*

{_;,} -------------- uppercase boldface English letter indicator *must precede each boldface letter*

{_,} ---------------- uppercase German letter indicator *must precede each German letter*

{_,8} -------------- open inner quote *punctuation indicator required in math mode*

## ■ 60 {"} (dot 5)

{"}------------------ base-line indicator

{"}------------------ start modified expression

{"}------------------ multipurpose indicator

{"}------------------ save current output position *tix vertical tool*

{ "k }------------- less than sign

{"k:} -------------- bar under less than sign, or is less than or equal to

{"k@"k]}--------- less than sign nested with less than sign, or is small compared with

{"k".k".1}------ less than sign followed by equals sign followed by greater than sign

{"k".1}----------- less than sign followed by greater than sign

{"k.k}------------- less than or equal to

{"x]y}------------- superimpose any symbol **y** on any symbol **x** *MathematiX-only symbol*

{"!<:]}---------- upper integral *MathematiX-only symbol*
{"!%:]}---------- lower integral *MathematiX-only symbol*
{":<_<]}--------- caret over horizontal bar
{":%*]}---------- dot under horizontal bar
{":%_<]}--------- caret under horizontal bar
{"$c]!}---------- integral with superimposed circle *MathematiX-only symbol*
{"$r]  !}--------- integral with superimposed rectangle *MathematiX-only symbol*
{"$4]!}---------- integral with superimposed square *MathematiX-only symbol*
{"]}---------------- nothing *tix null tool*
{"1}---------------- ratio sign
{"@:%*]}--------- dot under tilde
{".k<*]}--------- dot over equals sign
{".k<$t]}-------- triangle over equals sign
{".k<\]}--------- vertical bar over equals sign
{".k<_<]}-------- caret over equals sign
{".k<_%]}-------- inverted caret over equals sign
{".k<_8]}-------- question mark over equals sign
{".k<.*]}-------- degree sign over equals sign
{".k<;<]}-------- left-pointing caret over equals sign
{".k<;%]}-------- right-pointing caret over equals sign
{".k%**<**]}-- two dots over and two dots under equals sign
{".k%*<*]}------ dot over and dot under equals sign
{".k%_<]}-------- caret under equals sign
{",=]!}---------- integral with superimposed infinity *MathematiX-only symbol*

## ■ 61 {.} (dots 4-6)

{.}----------------- decimal point
{.}----------------- Greek letter indicator
{.}----------------- first inner radical indicator *part of begin and terminate radical indicators*
{ .k }------------ equals sign
{.k@%}------------ equals sign over logical product
{.k@%:}---------- equals sign over and bar under logical product
{.k@%.k}--------- equals sign over and equals sign under logical product
{.k@:}------------ equals sign over tilde
{.k@:@:}--------- equals sign over double tilde
{.k@+}------------ equals sign over logical sum
{.k@+:}---------- equals over and bar under logical sum
{.k@+.k}--------- equals sign over and equals sign under logical sum
{.k_"k}---------- equals sign over inclusion
{.k_.1}---------- equals sign over reverse inclusion
{.k"k}------------ equals sign over less than sign
{.k.1}------------ equals sign over greater than sign

### *dots 4-6 continued*

| | |
|---|---|
| {.&} | Greek chi |
| {.(} | left curly brace |
| {.)} | right curly brace |
| {.*} | hollow dot end of arrow *part of arrow* |
| {.*} | hollow dot, or degree sign in superscript |
| {.%} | intersection sign |
| {.%:} | bar under intersection |
| {.%.k} | equals sign under intersection |
| {.?} | theta (Greek letter) |
| {.:} | eta (Greek letter) |
| {.$} | del, or nabla, or gradient |
| { .1 } | greater than sign |
| {.1:} | bar under greater than sign, or is greater than or equal to |
| {.1@.1]} | greater than sign nested with greater than sign, or is large compared with |
| {.1""k} | greater than sign followed by less than sign |
| {.1".k""k} | greater than sign followed by equals sign followed by less than sign |
| {.1.k} | equals sign under greater than, or is greater than or equal to |
| {./} | division sign |
| {.+} | union sign |
| {.+:} | bar under union sign |
| {.+.k} | equals sign under union sign |
| {.#} | number sign, or crosshatch, or pounds |
| {.-} | proper difference |
| {.@f} | phi (alternate form of Greek letter) |
| {.@s} | sigma (alternate form of Greek letter) |
| {..} | second inner radical indicator *part of begin and terminate radical indicators* |
| {..(} | left angle bracket |
| {..)} | right angle bracket |
| {...} | third inner radical indicator *part of begin and terminate radical indicators* |
| {.,} | uppercase Greek letter indicator |
| {.,(} | left enlarged curly brace |
| {.,)} | right enlarged curly brace |
| {.,,p} | product operator (enlarged uppercase Greek pi) *MathematiX-only symbol* |
| {.,,s} | summation operator (enlarged uppercase Greek sigma) *MathematiX-only symbol* |

## ■ 62 {;} (dots 5-6)

| | |
|---|---|
| {;} | English-letter indicator |
| {;} | subscript indicator |
| {;<} | left caret |
| {;%} | right caret |
| {;2} | proportion sign |
| {;^} | subscript with superscript indicator |
| {;^^} | subscript with superscript with superscript indicator |
| {;^;} | subscript with superscript with subscript indicator |
| {;;} | subscript with subscript indicator |
| {;;^} | subscript with subscript with superscript indicator |
| {;;;} | subscript with subscript with subscript indicator |
| {;,} | uppercase English-letter indicator |

## ■ 63 {,} (dot 6)

| | |
|---|---|
| {,} | capitalization indicator |
| {,} | mathematical comma |
| {,=} | infinity |
| {,(} | left enlarged parenthesis |
| {,)} | right enlarged parenthesis |
| {,*} | therefore |
| {,?} | opening complex-fraction indicator |
| {,\} | enlarged vertical bar |
| {,\,\} | double enlarged vertical bar |
| {,/} | complex-fraction line |
| {,#} | closing complex-fraction indicator |
| {,'} | ditto mark |
| {,,} | double capitalization indicator |
| {,,a} | Hebrew aleph |
| {,,b} | Hebrew bet |
| {,,?} | opening hypercomplex-fraction indicator |
| {,,\} | double-height enlarged vertical bar *MathematiX-only symbol* |
| {,,\,,\} | double double-height enlarged vertical bar *MathematiX-only symbol* |
| {,,/} | hypercomplex-fraction line |
| {,,#} | closing hypercomplex-fraction indicator |

# Appendix C
# Nemeth Symbols Supported by MathematiX in Alphabetical Order

Use this Appendix when you know what it's called but you don't know the symbol. We list the symbols in alphabetical order, followed by any MathematiX-specific requirements in italics, finishing with the symbol itself. Symbols with several names show up several times. For example, {''} appears under "I," "P," and "S." To save space, we omit function names.

## A

| | |
|---|---|
| aleph, lowercase Hebrew letter | {..a} |
| alpha, lowercase Greek letter | {.a} |
| ampersand | {_&} |
| angle | {$[} |
| angstrom unit | {@,a} |
| apostrophe *punctuation indicator required in math mode* | {_'} |
| arc, concave down | {$'} |
| arc, concave up | {$a} |
| arrow shaft *part of arrow* | {33} |
| arrow, bidirectional horizontal (contracted form) | {$[o} |
| arrow, bidirectional horizontal | {$[33o} |
| arrow, bidirectional vertical (contracted form) | {$<[o} |
| arrow, bidirectional vertical | {$<[33o} |
| arrow, down (contracted form) | {$%o} |
| arrow, down | {$%33o} |
| arrow, left | {$[33} |
| arrow, right | {$33o} |
| arrow, right double | {$7o} |
| arrow, up (contracted form) | {$<o} |
| arrow, up | {$<33o} |
| asterisk | {@#} |
| at sign | {@a} |

## B

| | |
|---|---|
| backslash | {_*} |
| bar over and bar under logical product | {:@%:} |
| bar over and bar under logical sum | {:@+:} |
| bar over and equals sign under logical product | {:@%.k} |
| bar over and equals sign under logical sum | {:@+.k} |
| bar over double tilde | {:@:@:} |

| | |
|---|---|
| bar over greater than sign | {:.1} |
| bar over inclusion sign | {:_"k} |
| bar over less than sign | {:"k} |
| bar over logical product | {:@%} |
| bar over logical sum | {:@+} |
| bar over reverse inclusion sign | {:_.1} |
| bar over tilde | {:@:} |
| bar under double tilde | {@:@::} |
| bar under greater than sign | {.1:} |
| bar under inclusion sign | {_"k:} |
| bar under intersection | {.%:} |
| bar under less than sign | {"k:} |
| bar under logical product | {@%:} |
| bar under logical sum | {@+:} |
| bar under reverse inclusion sign | {_.1:} |
| bar under tilde | {@::} |
| bar under union sign | {.+:} |
| base-line indicator | {"} |
| because | {@/} |
| bet, lowercase Hebrew letter | {..b} |
| beta, lowercase Greek letter | {.b} |
| BEX format command, start of | {$$} |
| binomial expression, close | {)} |
| binominal expression, opening of *MathematiX-only symbol* | {@b(} |
| brace, left curly | {.(} |
| brace, left enlarged curly | {.,(} |
| brace, right curly | {.)} |
| brace, right enlarged curly | {.,)} |
| bracket, boldface left square | {_@(} |
| bracket, boldface right square | {_@)} |
| bracket, enlarged left square | {@,(} |
| bracket, enlarged right square | {@,)} |
| bracket, left angle | {..(} |
| bracket, left square | {@(} |
| bracket, right angle | {..)} |
| bracket, right square | {@)} |

## ■ C

| | |
|---|---|
| capitalization indicator | {,} |
| capitalization indicator, double | {,,} |
| caret | {_<} |
| caret over equals sign | {".k<_<]} |
| caret over horizontal bar | {":<_<]} |

| | |
|---|---|
| caret under equals sign | {".k%_<]} |
| caret under horizontal bar | {":%_<]} |
| caret, inverted | {_%} |
| caret, inverted over equals sign | {".k<_%]} |
| caret, left | {;<} |
| caret, left-pointing over equals sign | {".k<;<]} |
| caret, right | {;%} |
| caret, right-pointing over equals sign | {".k<;%]} |
| cent sign | {@c} |
| check mark | {@>} |
| chi, lowercase Greek letter | {.&} |
| circle | {$c} |
| circumflex | {_<} |
| colon *punctuation indicator required in math mode* | {_3} |
| comma, literary *punctuation indicator required in math mode* | {_1} |
| comma, mathematical | {,} |
| comma, mathematical at subscript or superscript level | {[} |
| complex-fraction indicator, closing | {,#} |
| complex-fraction indicator, opening | {,?} |
| complex-fraction line | {,/} |
| contained in | {_"k} |
| contains | {_.1} |
| cross product | {@*} |
| crossed d | {@$} |
| crossed h | {@h} |
| crossed lambda | {@.1} |
| crossed R | {@,r} |
| crosshatch | {.#} |

## D

| | |
|---|---|
| dagger | {_]} |
| dagger, double | {__]} |
| dash | {--} |
| dash, long | {----} |
| decimal point | {.} |
| degree sign *in superscript* | {.*} |
| degree sign over equals sign | {".k<.*]} |
| del | {.$} |
| del, boldface | {_.$} |
| delta, lowercase Greek letter | {.d} |
| diagonal line | {_/} |
| directly-over indicator (first order) | {<} |
| directly-under indicator (first order) | {%} |

| | |
|---|---|
| disappearing space *tix spacing tool* | {_ } |
| discretionary line break *tix spacing tool* | {@ } |
| ditto mark | {,'} |
| division sign | {./} |
| dollar sign | {@s} |
| dot | {*} |
| dot end of arrow *part of arrow* | {*} |
| dot over and dot under equals sign | {".k%*<*]} |
| dot over equals sign | {".k<*]} |
| dot under horizontal bar | {":%*]} |
| dot under tilde | {"@:%*]} |

### E

| | |
|---|---|
| element of | {@e} |
| ellipse | {$e} |
| ellipsis | { '''} |
| empty set *MathematiX-only symbol* | {@_0} |
| English letter indicator, boldface *must precede each boldface letter* | {_;} |
| English letter indicator,<br>uppercase boldface *must precede each boldface letter* | {_;,} |
| English-letter indicator | {;} |
| English-letter indicator, uppercase | {;,} |
| epsilon, lowercase Greek letter | {.e} |
| equals over and bar under logical sum | {.k@+:} |
| equals sign | { .k } |
| equals sign over and bar under logical product | {.k@%:} |
| equals sign over and equals sign under logical product | {.k@%.k} |
| equals sign over and equals sign under logical sum | {.k@+.k} |
| equals sign over double tilde | {.k@:@:} |
| equals sign over greater than sign | {.k.1} |
| equals sign over inclusion | {.k_"k} |
| equals sign over less than sign | {.k"k} |
| equals sign over logical product | {.k@%} |
| equals sign over logical sum | {.k@+} |
| equals sign over reverse inclusion | {.k_.1} |
| equals sign over tilde | {.k@:} |
| equals sign under double tilde | {@:@:.k} |
| equals sign under greater than | {.1.k} |
| equals sign under inclusion sign | {_"k.k} |
| equals sign under intersection | {.%.k} |
| equals sign under logical product | {@%.k} |
| equals sign under logical sum | {@+.k} |
| equals sign under reverse inclusion sign | {_.1.k} |

| | |
|---|---|
| equals sign under tilde | {@:.k} |
| equals sign under union sign | {.+.k} |
| equals sign, boldface | {_.k} |
| equivalence | {@<,<} |
| eta, lowercase Greek letter | {.:} |
| exclamation point *punctuation indicator required in math mode* | {_6} |
| existential quantifier | {@=} |

## F

| | |
|---|---|
| factorial | {&} |
| feet | {'} |
| for every | {@&} |
| for exactly one | {@=\} |
| for some | {@=} |
| fraction indicator, closing simple | {#} |
| fraction indicator, opening simple | {?} |
| fraction line, simple | {/} |

## G

| | |
|---|---|
| gamma, lowercase Greek letter | {.g} |
| German letter indicator *must precede each German letter* | {_} |
| German letter indicator, uppercase *must precede each German letter* | {_,} |
| gradient | {.$} |
| greater than sign | { .1 } |
| greater than sign followed by equals sign followed by less than sign | {.1".k""k} |
| greater than sign followed by less than sign | {.1""k} |
| greater than sign nested with greater than sign | {.1@.1]} |
| Greek letter indicator | {.} |
| Greek letter indicator, uppercase | {.,} |

## H

| | |
|---|---|
| half-bracket, lower left | {@;(} |
| half-bracket, lower right | {@;)} |
| half-bracket, upper left | {@^(} |
| half-bracket, upper right | {@^)} |
| Hebrew aleph | {,,a} |
| Hebrew bet | {,,b} |
| hollow dot | {.*} |
| hollow dot end of arrow *part of arrow* | {.*} |
| horizontal bar | {:} |
| horizontal double bond *chemistry bond diagram* | {_7]} |

| | |
|---|---|
| horizontal single bond *chemistry bond diagram* | {_3]} |
| horizontal solid bond *chemistry bond diagram* | {_g]} |
| horizontal triple bond *chemistry bond diagram* | {_=]} |
| hypercomplex-fraction indicator, closing | {,,#} |
| hypercomplex-fraction indicator, opening | {,,?} |
| hypercomplex-fraction line | {,,/} |
| hyphen | {-} |

## ■ I

| | |
|---|---|
| identical with | {_l} |
| identity | {_l} |
| inches | {''} |
| inclusion sign | {_"k} |
| index-of-radical indicator | {<} |
| infinity | {,=} |
| integral | {!} |
| integral with superimposed circle *MathematiX-only symbol* | {"$c]!} |
| integral with superimposed infinity *MathematiX-only symbol* | {",=]!} |
| integral with superimposed rectangle *MathematiX-only symbol* | {"$r] !} |
| integral with superimposed square *MathematiX-only symbol* | {"$4]!} |
| integral, double | {!!} |
| integral, lower *MathematiX-only symbol* | {"!%:]} |
| integral, triple | {!!!} |
| integral, upper *MathematiX-only symbol* | {"!<:]} |
| intersection sign | {.%} |
| iota, lowercase Greek letter | {.i} |
| it does not follow | {/,*} |

## ■ K

| | |
|---|---|
| kappa, lowercase Greek letter | {.k} |
| koph, lowercase Greek letter | {.q} |

## ■ L

| | |
|---|---|
| lambda, lowercase Greek letter | {.l} |
| large compared with | {.1@.1]} |
| left arrow head *part of arrow* | {[} |
| less than or equal to | {"k.k} |
| less than sign | { "k } |
| less than sign followed by equals sign followed by greater than sign | {"k".k".1} |
| less than sign followed by greater than sign | {"k".1} |
| less than sign nested with less than sign | {"k@"k]} |

| | |
|---|---|
| line, new output *BEX format indicator* | {$l} |
| literary translation mode *tix translation tool* | {@l} |
| logical product | {@%} |
| logical sum | {@+} |
| lower limit | {%lim} |
| lower limit | {%limit} |

## M

| | |
|---|---|
| macron | {:} |
| math translation mode *tix translation tool* | {@m} |
| math translation mode *tix translation tool* | {@-} |
| membership | {@e} |
| membership, reverse | {@5} |
| minus | {-} |
| minus or plus | {-+} |
| minus, boldface | {_-} |
| minus, boldface followed by boldface plus | {_-"_+} |
| minus, boldface followed by regular plus | {_-"+} |
| minus, regular followed by boldface plus | {-"_+} |
| minus, regular followed by regular plus | {-"+} |
| minutes | {'} |
| modified expression, start | {"} |
| modified expression, terminate | {]} |
| move down four characters *tix vertical tool* | {[%f} |
| move down half of a character *tix vertical tool* | {[%h} |
| move down one and a half characters *tix vertical tool* | {[%x} |
| move down one character *tix vertical tool* | {[%o} |
| move down quarter of a character *tix vertical tool* | {[%q} |
| move down three characters *tix vertical tool* | {[%t} |
| move down two characters *tix vertical tool* | {[%d} |
| move up four characters *tix vertical tool* | {[<f} |
| move up half of a character *tix vertical tool* | {[<h} |
| move up one and a half characters *tix vertical tool* | {[<x} |
| move up one character *tix vertical tool* | {[<o} |
| move up quarter of a character *tix vertical tool* | {[<q} |
| move up three characters *tix vertical tool* | {[<t} |
| move up two characters *tix vertical tool* | {[<d} |
| mu, lowercase Greek letter | {.m} |
| multiplication sign | {@*} |
| multipurpose indicator | {"} |

## N

| | |
|---|---|
| nabla | {.$} |
| not an element of | {/@e} |
| not parallel to | {/$1} |
| not perpendicular to | {/$p} |
| nothing *tix null tool* | {"]} |
| nu, lowercase Greek letter | {.n} |
| number sign | {.#} |
| numeral 0 | {0} |
| numeral 1 | {1} |
| numeral 2 | {2} |
| numeral 3 | {3} |
| numeral 4 | {4} |
| numeral 5 | {5} |
| numeral 6 | {6} |
| numeral 7 | {7} |
| numeral 8 | {8} |
| numeral 9 | {9} |
| numeric indicator *optional in math mode* | {#} |
| numeric indicator, boldface *must precede each boldface digit* | {_#} |

## O

| | |
|---|---|
| oblique double bond (lower left to upper right) *chemistry bond diagram* | {_//} |
| oblique double bond (upper left to lower right) *chemistry bond diagram* | {_**} |
| oblique single bond (lower left to upper right) *chemistry bond diagram* | {_/} |
| oblique single bond (upper left to lower right) *chemistry bond diagram* | {_*} |
| omega, lowercase Greek letter | {.w} |
| omission sign | {=} |
| omicron, lowercase Greek letter | {.o} |
| oval | {$e} |

## P

| | |
|---|---|
| page, new output *BEX format indicator* | { $f } |
| paragraph symbol (inkprint sign) | {@,p} |
| paragraph, new output *BEX format indicator* | { $p } |
| parallel to | {$1} |
| parenthesis, left | {(} |

| | |
|---|---|
| parenthesis, left enlarged | {,(} |
| parenthesis, right | {)} |
| parenthesis, right enlarged | {,)} |
| partial derivative | {@d} |
| percent sign | {@0} |
| period *punctuation indicator required in math mode* | {_4} |
| perpendicular to | {$p} |
| phi, standard lowercase Greek letter | {.f} |
| phi, alternate lowercase form | {.@f} |
| pi, lowercase Greek letter | {.p} |
| plus | {+} |
| plus or minus | {+-} |
| plus, boldface | {_+} |
| plus, boldface followed by boldface minus | {_+"_-} |
| plus, boldface followed by regular minus | {_+"-} |
| plus, regular followed by boldface minus | {+"_-} |
| plus, regular followed by regular minus | {+"-} |
| pound Sterling | {@l} |
| pounds | {.#} |
| prime | {'} |
| prime, double | {''} |
| prime, triple | {'''} |
| product operator, enlarged<br>uppercase Greek pi *MathematiX-only symbol* | {.,,p} |
| proper difference | {.-} |
| proportion sign | {;2} |
| punctuation indicator *required in math mode as listed here* | {_} |
| psi, lowercase Greek letter | {.y} |

## Q

| | |
|---|---|
| question mark *punctuation indicator required in math mode* | {_8} |
| question mark over equals sign | {".k<_8]} |
| quote, closing *punctuation indicator required in math mode* | {_0} |
| quote, closing inner *punctuation indicator required in math mode* | {_0'} |
| quote, open *punctuation indicator required in math mode* | {_8} |
| quote, open inner *punctuation indicator required in math mode* | {_,8} |

## R

| | |
|---|---|
| radical | {>} |
| radical indicator, first inner<br>*part of begin and terminate radical indicators* | {.} |
| radical indicator, second inner<br>*part of begin and terminate radical indicators* | {..} |

| | |
|---|---|
| radical indicator, third inner *part of begin and terminate radical indicators* | {...} |
| radical, terminate | {]} |
| ratio sign | {"1} |
| rectangle | {$r} |
| reference indicator, general | {@]} |
| restore output position *tix vertical tool* | {]} |
| restore speech and pause when screen is full *tix verbalize tool* | {@talk } |
| reverse inclusion sign | {_.1} |
| rho, lowercase Greek letter | {.r} |
| right arrow (contracted form) | {$o} |
| right arrow head *part of arrow* | {o} |

## S

| | |
|---|---|
| sampi, lowerrase Greek letter | {.c} |
| save output position *tix vertical tool* | {"} |
| script typefont English letter *must precede each scripted letter* | {@;} |
| script typefont English letter, uppercase *must precede each scripted letter* | {@;,} |
| seconds | {''} |
| section mark | {@,s} |
| section mark, double | {@,s@,s} |
| semicolon *punctuation indicator required in math mode* | {_2} |
| shape indicator | {$} |
| sigma, standard lowercase Greek letter | {.s} |
| sigma (alternate lowercase form) | {.@s} |
| since | {@/} |
| slash | {_/} |
| small compared with | {"k@"k]} |
| square | {$4} |
| square root | {>} |
| square root, terminate | {]} |
| square, filled | {$_4} |
| star | {$s} |
| subscript indicator | {;} |
| subscript with subscript indicator | {;;} |
| subscript with subscript with subscript indicator | {;;;} |
| subscript with subscript with superscript indicator | {;;^} |
| subscript with superscript indicator | {;^} |
| subscript with superscript with subscript indicator | {;^;} |
| subscript with superscript with superscript indicator | {;^^} |
| subset of | {_"k} |

| | |
|---|---|
| summation operator, enlarged uppercase Greek sigma *MathematiX-only symbol* | {.,,s} |
| superimpose any symbol y on any symbol x *MathematiX-only symbol* | {"x]y} |
| superscript indicator | {^} |
| superscript with subscript indicator | {^;} |
| superscript with subscript with subscript indicator | {^;;} |
| superscript with subscript with superscript indicator | {^;^} |
| superscript with superscript indicator | {^^} |
| superscript with superscript with subscript indicator | {^^;} |
| superscript with superscript with superscript indicator | {^^^} |
| suppress pauses when screen is full *tix verbalize tool* | {@express } |
| suppress speech output unless structure error *tix verbalize tool* | {@notalk } |

## T

| | |
|---|---|
| tau, lowercase Greek letter | {.t} |
| terminate modified expression with overbar | {<:]} |
| terminate modified expression with underbar | {%:]} |
| there exists | {@=} |
| there exists exactly one | {@=\} |
| there exists uniquely | {@=\} |
| therefore | {,*} |
| theta, lowercase Greek letter | {.?} |
| tilde | {@:} |
| tilde followed by tilde | {@:"@:} |
| tilde, double | {@:@:} |
| tilde, extended | {@,:} |
| times | {*} |
| times | {@*} |
| triangle | {$t} |
| triangle over equals sign | {".k<$t]} |
| two dots over and two dots under equals sign | {".k%**<**]} |

## U

| | |
|---|---|
| union sign | {.+} |
| universal quantifier | {@&} |
| upper limit | {<lim} |
| upper limit | {<limit} |
| upsilon, lowercase Greek letter | {.u} |

## V

| | |
|---|---|
| variation (varies as) | {_=} |
| vau, lowercase Greek letter | {.v} |
| vertical bar | {\} |
| vertical bar followed by vertical bar | {\"\} |
| vertical bar over equals sign | {".k<\]} |
| vertical bar, boldface | {_\} |
| vertical bar, double | {\\} |
| vertical bar, double boldface | {__\} |
| vertical bar, double double-height enlarged *MathematiX-only symbol* | {,,\,,\} |
| vertical bar, double enlarged | {,\,\} |
| vertical bar, double followed by double vertical bar | {\\"\\} |
| vertical bar, double-height enlarged *MathematiX-only symbol* | {,,\} |
| vertical bar, enlarged | {,\} |
| vertical bar, triple | {\\\} |
| vertical bond *chemistry bond diagram* | {\} |
| vertical double bond *chemistry bond diagram* | {\\} |
| vertical triple bond *chemistry bond diagram* | {\\\} |

## X & Z

| | |
|---|---|
| xi, lowercase Greek letter | {.x} |
| zeta, lowercase Greek letter | {.z} |