



HAL
open science

Tiqwah: A Typesetting System for Biblical Hebrew, based on TeX

Yannis Haralambous

► **To cite this version:**

Yannis Haralambous. Tiqwah: A Typesetting System for Biblical Hebrew, based on TeX. Quatrième Colloque International Bible et informatique, matériel et matière, Association Internationale Bible et Informatique et Vrije Universiteit Amsterdam, Aug 1994, Amsterdam, Netherlands. pp.445-470. hal-02100472

HAL Id: hal-02100472

<https://hal.science/hal-02100472>

Submitted on 15 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tiqwah: A Typesetting System for Biblical Hebrew, based on T_EX

Dr. Yannis Haralambous*

Abstract

This paper presents *Tiqwah*: a typesetting system for Biblical Hebrew, which uses the combined efforts of T_EX, METAFONT and GNU Flex. The author describes its use and features, discusses issues relevant to the design of fonts and placement of floating diacritics and gives a list of rare cases and typographical curiosa which can be found in the Bible. The paper concludes with an example of Hebrew Biblical text typeset by *Tiqwah* (the begin of the book of Genesis).

1 Introduction

The *Tiqwah* system uses the possibilities of T_EX, METAFONT and GNU Flex to typeset Biblical Hebrew. This is not a simple task: (a) special fonts are needed, (see section 3); (b) several levels of diacritics are required; they have to be entered in a reasonable way (see 2.2.2, and 2.2.3), and placed correctly under or over the characters (see 3.3). The Bible being the most demanding Hebrew text (from the typographical point of view), *Tiqwah* can trivially be used to typeset any other Hebrew text, classical or modern; besides Tiberian, also Babylonian and Palestinian vowels are included in the font, as well as special characters for Yiddish.

This paper is divided into three parts: the first one, more pragmatical, describes the requirements and use of the *Tiqwah* system; the second one discusses the design of the fonts and the algorithm of placement of floating diacritics; finally, the third part gives a list of rare cases and typographical curiosa found in the Hebrew Bible, and the way to produce them through *Tiqwah*.

But first, for the reader not familiar with T_EX and METAFONT, a short introduction.

*Centre d'Études et de Recherche sur le Traitement Automatique des Langues, Institut National des Langues et Civilisations Orientales, Paris. Private address: 187, rue Nationale, 59800 Lille, France. Fax: (33) 20.40.28.64, ISDN/Numéris: (33) 20.15.81.77, E-mail: Yannis.Haralambous@univ-lille1.fr

1.1 What is T_EX?

Donald E. Knuth's typesetting system T_EX is the achievement of a man who believes that beauty can be found everywhere; even in the way of bringing odd mathematical formulas on paper. He not only proved that this can be done by a machine, but also that it can be done in an *efficient* and *elegant* way. And he donated his work to the academical community by releasing it in the public domain.

T_EX should be pronounced as in modern Greek τέχ; the name comes from the three first letters of words τέχνη (Art) and τεχνολογία (Technology). T_EX is not “what you see's what you get”: the user prepares a file, containing the text and some structural mark up, he/she then executes T_EX which reads the document and produces a separate camera-ready document, which can be previewed, printed, faxed, etc.

The process of executing T_EX is the same under all circumstances; we concentrate on the syntax of the file the user has to prepare (the *input*). T_EX is a programming language and so it necessarily has *commands* and *comments*; but being above all a typesetting system, the main importance is given to *text*. *Commands* are English words, in most cases preceded by the character \ (backslash); everything on the right of a % (percent sign) is a *comment*; the rest is *text*.

Commands can be defined or re-defined by the user to fulfill specific needs. Also portions of text can be *arguments* of commands; for example one could define a macro \hebrewnum with one argument: the number; for the values 1, 15, 900, 2000 it would give respectively א, טו, ת, כ. This would be written in the following way:

```
\hebrewnum{1}, \hebrewnum{15}, \hebrewnum{900}, \hebrewnum{2000},
```

where the curly braces delimit the argument. This feature of T_EX gives the user infinite possibilities of text manipulation.

So, actually, the user is *structuring* the text before typesetting; this additional structure can be used in several ways. The same thing happens also at the level of the document structure. By using a very widespread set of T_EX macros called L^AT_EX 2_ε, the user indicates the different parts of the document, words to be included in the index, bibliographic references, cross-references etc. L^AT_EX 2_ε then produces a document according to ‘style specifications’, which can be chosen by the user or can be supplied by his/her institution to insure uniformity of design. Needless to say that the table of contents, index, bibliography, cross references, etc. are automatically generated and updated by L^AT_EX 2_ε.

T_EX is very well adapted to multilingual texts: by using simple commands anywhere inside a text the user can instantly change hyphenation rules, punctuation spacing, script (if necessary) as well as the direction of text (the latter feature requires a special bidirectional version of T_EX, called T_EX--X_ET).

Returning to the special needs of Hebrew, another characteristic feature of T_EX is that it works by assembling ‘boxes’, in the sense that to every character of a font corresponds an invisible two-dimensional box which T_EX will very accurately place on the page. At every instance, T_EX is able to find out the dimensions of boxes, make arithmetic calculations out of these data, and according to the results, move them around on the page (both horizontally and vertically).

This feature has been used by the author in *Tiqwah*, for the placement of diacritics. As explained in section 3.3, T_EX has to perform a certain number of calculations for every letter with diacritics, before typesetting it. Wide combinations of diacritics will force T_EX to kern

between letters, this will change the global width of the word, and maybe of the line, the paragraph, the page, etc. These calculations are done with a precision of $\frac{1}{65536}$ of a point, which is about $\frac{1}{100}$ of the wavelength of visible light.

Last but not least a very important feature of T_EX is its universality and portability. Written in standard and well documented Pascal, it has been ported to every possible platform, from personal computers to working stations and mainframes. D. E. Knuth has placed severe restrictions to these implementations so that they all remain strictly compatible. Hundreds of thousands of people in the world use T_EX to typeset and to communicate their documents through email and ftp; the fact of being open and public domain software has made T_EX also very widespread in the Eastern and Third World countries.

For further reading on T_EX and L^AT_EX, the reader may consult [?], [?], [?].

1.2 And METAFONT?

METAFONT is a programming language, as innovatory and powerful as T_EX, but specialized in font design and production. In a traditional typeface, the same character can appear in different sizes, weights (light, bold etc.), and shapes (straight, slanted, small capitals etc). All of these must look similar enough to make apparent the fact that they belong to the same font family; still they have their own specific features. Not to mention that the different characters of the same font must also have common characteristics; resemblances inside a font make it homogeneous and harmonic—differences make it lively and witty (just like human relations inside a family).

A computer should be taught about the common characteristics of characters, as well as their differences; this is exactly what METAFONT does: a ‘meta-font’ is a set of instructions by which arbitrarily many fonts can be drawn by varying the values of certain parameters. These parameters will change the size, weight and shape of the characters in different fonts, but the basic properties of the font will remain the same so that the fonts created out of a ‘meta-font’ deserve the term ‘family of fonts’.

Of the utmost importance is the possibility of adding *optical scaling* to characters: characters have thin and fat strokes; in traditional typography, when these characters are scaled, the thin and fat strokes do not behave in the same way. Let’s take for example a Bodoni ‘H’; this famous Italian font designer liked to have very thin strokes (in this case, the horizontal bar) opposed to very fat ones (the two vertical stems). When this character is magnified, the thin stroke must remain thin, otherwise it would lose its elegance. On the other hand, when the character is reduced, again the thin stroke cannot be changed as much as the fat ones, because it could very easily disappear. And this is exactly what happens in nowadays DTP using linearly scalable fonts, where a single shape is reduced and magnified to produce what was specifically and separately drawn in traditional font design.

In METAFONT we reach the other extreme: every size involved in the design of a character is parametrizable, and the designer can specify the behaviour of every stroke when the font is scaled, or when its weight or shape are modified. These methods have been applied to the Biblical Hebrew font of the *Tiqwah* system (see section 3 for more details on *Tiqwah* fonts).

METAFONT is as portable and universal as T_EX: it can produce fonts for every device (printer, fax or screen), from cheap dot-matrix printers to professional phototypesetting devices. The code written by the font designer is internally converted into Bézier curves (third

degree polynomials); then METAFONT adequately rasterizes these curves according to certain parameters concerning the output device. METAFONT runs on every possible platform; special versions of it produce PostScript output as well (cf. [?], [?]).

For further reading on METAFONT, the reader may consult [?], [?].

2 Using *Tiqwah*

2.1 Requirements

To typeset in Biblical Hebrew using *Tiqwah*, one needs a decent T_EX system (in this context, we need a T_EX implementation featuring Peter Breitenlohner's T_EX-_X_ET as well as a METAFONT implementation with user-configurable parameters¹), a relatively powerful machine (being able to run BigT_EX) and the *Tiqwah* package², consisting of a preprocessor written in GNU Flex, fonts written in METAFONT, and T_EX macros. The preprocessor is written entirely in GNU Flex, without using any system-dependent subroutines; hence it can be compiled in a straightforward manner on any platform having a GNU Flex executable and an ANSI C compiler.

Once *Tiqwah* installed, typesetting is done in two steps: an input file is prepared using the syntax we will describe below; the preprocessor then reads this file, and produces a L^AT_EX 2_ε (or plain T_EX) file which then can be run through T_EX, in the usual way.³

2.2 Preparing the input file

If you wish to write your file in L^AT_EX 2_ε, you have to include the line

```
\usepackage{tiqwah}
```

in the preamble. Plain T_EX users will write `\input tiqwamac.tex` at the beginning. However, the author recommends the use of L^AT_EX 2_ε, because of its powerful font selection scheme.

A *Tiqwah* input file contains text, T_EX/L^AT_EX macros, and *preprocessor directives*. The latter concern only Hebrew script. To type Hebrew text you need to enter *Hebrew mode*; this is done by the preprocessor directive `<H>`. To leave Hebrew mode, enter the directive `</H>`. For Yiddish, the directives are `<Y>` and `</Y>`. The directives `<H>` and `<Y>` are the only ones recognized by the preprocessor *outside* Hebrew/Yiddish mode.⁴

¹The internal parameter `max_font_dimen` of METAFONT has to take a value of at least 53, to be able to generate *Tiqwah* fonts.

²*Tiqwah* will be included in ScholarT_EX; it is part of the long awaited version 1 of this package, together with *new* Greek, Arabic, Syriac, Coptic and Akkadian cuneiform fonts.

³An adaptation of the *Tiqwah* system to Ω (the T_EX extension prepared by John Plaice and the author) is under preparation: it will allow typesetting in vowelized non-Biblical Hebrew, without preprocessor.

⁴A few other similar directives concern other languages of scholarly interest: `<A>` for Arabic Naskhi, `<S>` for Syriac Serto, `<E>` for Syriac Estrangelo, `<C>` for East Syriac, `<K>` for Akkadian

Once you are inside Hebrew/Yiddish mode, you type Hebrew text in Latin transcription, from left-to-right. No special indication needs to be given to T_EX about font or writing direction switching, this is done automatically by T_EX. The following sections describe the transcription you have to use as well as all other features of the preprocessor.

2.2.1 Letters

The Hebrew transcription of letters (consonants and long vowels) is given in table B of the appendix (page 20); the Yiddish one will be given together with all other features of the Yiddish part of *Tiqvah*, in a forthcoming paper, dedicated entirely to this language.

Here is a simple example of code producing non-vowelized Hebrew text:

```
<H>ym hm*lx hw*' hm*qwm hn*mw*k b*ywtr b*'wlm</H> will produce
יום המלח הוא המקום הנמוך ביותר בעולם.
```

Some notes concerning the transcription of letters of table B: there is no distinction between medial and final forms; these are automatically applied by T_EX. The asterisk * transcribes the *dagesh*, *mappiq* or *šureq* dot.⁵ The broken lamed ם is used automatically whenever no upper diacritic is present,⁶ this feature can be turned off by the command line option `-nobroken` of the preprocessor. The character ם is a ligature of letters *ayin* and *lamed*; a variant form of it is ם. This ligature is not used in the Bible, and hence is not applied automatically by the preprocessor: it has to be explicitly requested by the code ' /1 (instead of ' 1 which will produce the normal ם).

In the same table, the reader will also encounter the symbol ך; it is called “*nun invers*” and is used in (Nu 10,35–36) and (Ps 107). The raised *ayin* occurs in (Ps 80) and (Hi 38), see section 4.2 for more details.

2.2.2 Vowels

Hebrew vowels and their transcriptions are displayed in tables C and D (p. 21). Table C displays the three available vowelization systems: Tiberian (the most frequent one), Palestinian and Babylonian. Tiberian vowels are used by default. To switch to Palestinian or Babylonian, one uses the directives <PALESTINIAN> and <BABYLONIAN>. The directive for Tiberian is <TIBERIAN>. The same text can be typeset in either one of the three systems, just by adding/removing one of these directives; here is an example of the same text, written in the three vowel systems:

ימלא פי תהלתך היום תפארתך
ימלא פי תהלתך היום תפארתך
ימלא פי תהלתך היום תפארתך

cuneiform. Other writing systems provided by ScholarT_EX do not require a preprocessor.

⁵Letters *aleph*, *ayin*, and *resh* with *dagesh* are very rare, see section 4.3 for more details.

⁶With one exception: the *holem*.

Most vowels can be entered in two different ways: either by a “phonetic” one- (or two-) letter code (a for *pataḥ*, A for *qames*, etc.) or by a three-letter code in uppercase form, surrounded by < and > (<PAT> for *pataḥ*, <QAM> for *qames*, etc.) Both methods are equivalent and can be arbitrarily mixed.

Vowels are entered *after* letters, except in the case of the *pataḥ furtivum*, where the code <PTF> has to be entered before x, h*, or ‘ (א, ה and ש are the only letters which can take a *pataḥ furtivum*⁷). The *rafe* accent can be found in table E.

Follows the same example of simple Hebrew text with its transcription, this time vowelized:

```
<H>yAm ham*Elax hw*' ham*Aqw^om han*Amw*k" b*"yw^oter b*A'w^olAm</H>
```

will produce

יום המלח הוא המקום הנמוך ביותר בעולם

2.2.3 Masoretic accents and other symbols

Tables E and F (p. 22, 23) display Masoretic cantillation marks and miscellaneous symbols: the Sephardic *varika*, and punctuation marks *maqeph*, *setuma*, *petuḥa*, *soph pasuq*. Two styles of Masoretic accents are provided: oldstyle (as found in BHK⁸ and [?], [?]) and modern (as in BHS⁸). The distinction is done on the T_EX level, by macros (\modernmasoretic and \oldstylemasoretic), which can be used inside or outside Hebrew mode; the default style is oldstyle. Table F shows the glyphs of modern Masoretic accents. The same remark as in the previous section, concerning alternative input of codes applies in this case as well.

Masoretic accents are entered after the letter to which they belong; they can be placed before or after vowels belonging to the same letter, their order is not important. Prepositive accents are placed before the first letter of the word. Postpositive accents, like *paṣta*, placed *inside* a word, will be typeset *between* letters.

Finally, table G (p. 23) displays a collection of typographical curiosa: symbols used in various contexts and for various purposes. The single and double quotes ‘ and ’ are used for numerals and abbreviations. The upper two dots are also used for numerals: the indicate thousands. The asterisk * is used both as an editorial mark (like the circellus, but apparently with slightly different meaning), and as a replacement character for missing letters (see section 4.8). The zero-like symbol 0 is used to indicate a missing word in (Jdc 20,13 [?], [?] only). The isolated *dageš* is used to indicate a missing letter with *dageš*, in (Jes 54,16 BHS only). Finally, the dotted circle ◌ is used in textbooks as a basis for diacritics.

⁷The combination “letter *ayin* with *pataḥ furtivum*” is not displayed in the table, because it is graphically indistinguishable from the normal *ayin* with *pataḥ*. This can be changed if there is a demand for differentiation of the two *pataḥ* types.

⁸Throughout this paper, BHS will be the “Biblia Hebraica Stuttgartensia” [?], and BHK the “Biblia Hebraica” [?], edited by Rudolf Kittel.

2.2.4 Other preprocessor directives

A few directives do not produce glyphs, and hence are not included in the tables:

1. `<NIL>` placed after a letter, will prevent the final form to be applied to it (for example, for numerals or stand-alone letters which have to be in medial form: `<H>k<NIL> k</H>` will produce `ך כ`);
2. `<EOW>` placed after a letter will force it to be in final form, even if other letters follow (for example, in (Jes 54,16) one reads `לְמַרְבֵּה`; this word has been entered as `<H>lima<EOW>r"b*e<AZL>h</H>`). See section 4.4 for more details;
3. `<EMPTY>` will produce an invisible character of normal width. It can be used as a basis for stand-alone diacritics in the case of missing words (see 4.9);
4. `<SMALL>` and `<BIG>` will produce small and big letters, see section 4.1 for more details; they act only on one letter at a time.

More features may be added to the preprocessor if necessary.

2.3 Running the preprocessor

Once you have prepared the input file, for example `genesis.inp`, you run the preprocessor by writing

```
tiqwah options < genesis.inp > genesis.tex
```

where *options* can be the following:

1. `-h` displays a few lines describing the command line options;
2. `-p` produces plain `TEX` instead of `LATEX` output (typesetting with *Tiqwah* in plain `TEX` is not recommended by the author);
3. `-l` followed by a number, indicates the maximum line length of code produced by the preprocessor; default is 80. This applies only to commands inside Hebrew/Yiddish mode, the remaining of the file is not modified;
4. `-nobroken` disables the automatic broken lamed insertion. With this option, `<H>w"l<SIL>'ô=yAla<RBM>d"t*iy</H>` will produce `וְלֹא־יִלְדְתִי` (Jes 23,4), without it, you would get `וְלֹא־יִלְדְתִי`. It should be noted that the *holem* vowel fits on the broken *lamed*: a special “broken-lamed-with-*holem*” glyph is provided in the font (5);
5. `-d` produces debugging output, sent to the `stderr` stream [for those who want to modify the code of the preprocessor].

2.4 Running T_EX/L_AT_EX

As usual. T_EX has to be T_EX--X_ET, otherwise you will get an error message about unknown commands `\beginR` and `\endR`.

If you are using L_AT_EX 2_ε, you have to include the line `\usepackage{tiqwah}` in the preamble; if you are using plain T_EX (not recommended), write `\input tiqwamac.tex` instead.

We have completed the description of the preprocessor's use and features. Now we will turn ourselves to issues concerning the design of fonts and the placement of floating diacritics.

3 Fonts for typesetting the Holy Bible in Hebrew

Designing fonts for Biblical typesetting is quite a challenge: on the one hand, one has to face centuries of tradition, and the inevitable comparison with masterpieces of typography; on the other hand, unlike Western typography, there is no room for innovation: modern Hebrew typefaces are widely used in Israel and elsewhere, but certainly *not* for Biblical text! Working under such tight restrictions can be compared to composing fugas or painting Byzantine Icons: there are very strict rules to struggle with, and you can't avoid being hooked by the masterpieces others have done and which fatally are out of reach...

Fortunately, digital font creation does not always need to be original and innovative (although at the end it always *will* have new features, since the phototypesetting machines are fundamentally different from the traditional presses). After all, we are in the age of *reproduction*...

The author started with the idea in mind to reproduce as faithfully as possible, the most beautiful Hebrew font he could find. There seems to be a consensus among a large group of scholars that one of the most beautiful Hebrew types ever done was the one of the *Biblia Hebraica*, edited by Kittel and printed in Germany in the early twenties. Unfortunately the molds were lost in the bombing of Leipzig, so only printed copies of that book could be studied by the author to get the necessary information for reproducing the font.

Doing this, and studying other books as well, such as a Haggadah by Saul Raskin ([?]), printed in New York in 1941, and old Talmudic books printed in Vienna in the late XIXth century, the author realized that Hebrew fonts have a fascinating feature: there is a remarkable deviation between different point sizes (in T_EXnical terms: they have a high degree of metaness). While for Latin typefaces the changes between small and large point sizes affect mostly the width of strokes, in Hebrew, letter shapes often change considerably. And what's even more unusual: changes that occur when going from small sizes to the normal size often occur the other way around when going from normal to large size: for example, one can see, in figure 1, the letter *gimel* at 6, 11, and 36 points (magnified so that they all have the same physical size). While the right tail of the letter moves more to the right when going

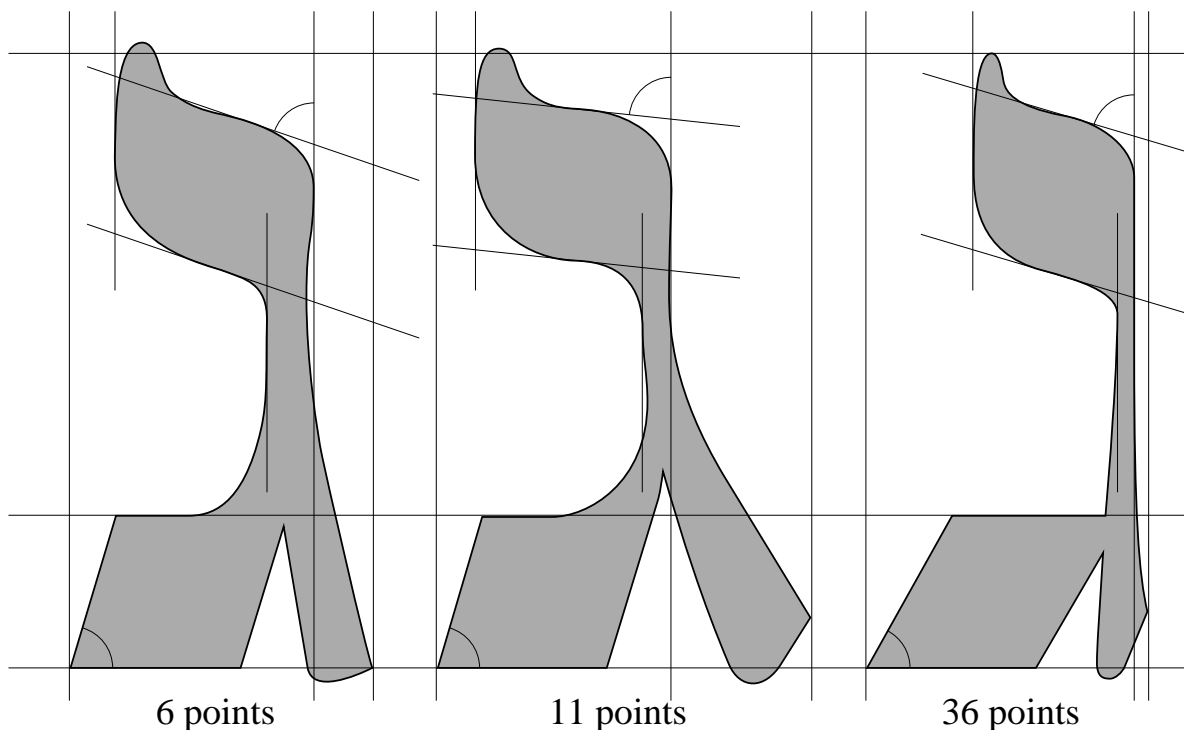


Figure 1: The letter λ at point sizes 6, 11, 36

from 6 to 11 points, it retracts again when going from 11 to 36, and almost becomes vertical.

Here is a (possible) explanation for this behaviour: the reasons which cause metaness are different when in the small-to-normal range, from those in the normal-to-large range: in the former case, the problem to solve is legibility. As a matter of fact, many Hebrew letters look quite similar in normal size: compare samekh and final mem, or kaph and bet, at 11 (or higher) points in the table of Appendix A. Their distinctive features are so discreet that they could as well disappear if the normal size was linearly reduced; a well drawn small point Hebrew font has to bring these distinctive features to the foreground: compare these letters again at 6 points: kaph and final mem are round while samekh and bet remain quadratic. On the other hand, when going from normal to large, one follows purely esthetic criteria: elegance is the main goal; in that context, Hebrew letters follow “Bodoni-like” esthetics: they have very important fat strokes and very fine thin ones. Hebrew letters use—even more than Latin letters—the effect of contrast between fat and thin strokes.

Being hooked by the beauty of this script, the author decided not only to produce a most decent Hebrew font, but also to cover the whole range of optical METAMorphoses of the types. On table A of the appendix (p. 19), the reader can see the first results of this adventure; they are by no means final! The author hopes to be able to improve these characters to meet the outcome of the Hebrew typograph-

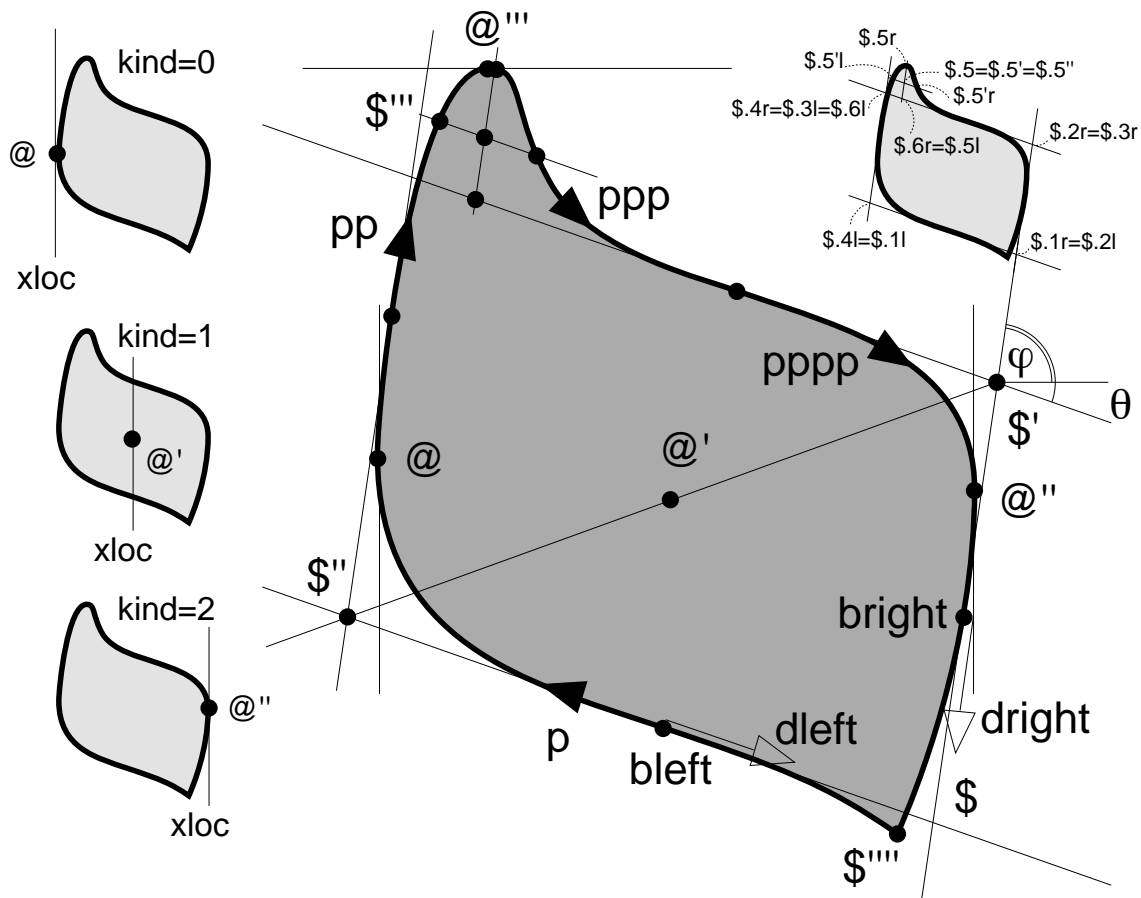


Figure 2: The METAFONT description of the character part “flame”

ical tradition.

3.1 Technical details

Drawing a font with such a high amount of metaness is a process not far from *morphing*, a technique used more and more in video and cinema⁹. Nevertheless there is an important difference between METAFONT “morphing” and the usual morphing we see in movies. To *morph* two images, we are not changing the grayscale (or color) weight of each pixel, but the coordinates of Bézier curve control points. Interpolation becomes very uncertain, since it is by no means trivial that the set of interpolated Bézier curves will still produce a decent character shape.

The solution to this problem is to detect “tendencies” in the letter shape metaness and to be guided by these while morphing: for example, the lower left stroke of the letter aleph, has the tendency of protruding to the left when pointsize

⁹*Morphing*, is the continuous interpolation between two pictures; it has been used in special effects, for example to show people being transformed into other people.

gets small. This has to be taken into account for all paths of this stroke, so that the transformation is homogeneous. The best way to do this is to determine “centers of gravity”, which will move during the transformation; then it suffices to define all the important control points of the stroke with respect to a center of gravity: in this way the movement of the latter will produce an homogeneous move (and hence, transformation) of the whole stroke.

An important precaution is to limit the metaness of certain quantities to a certain range of point sizes. For example, the width of fat strokes can vary arbitrarily (after all it is directly related to the letter point size), but other characteristics should not “vary too much”, in other words, they should remain stable outside of a certain point size range. That is the case, for example, of the “hanging left stroke” of letter final pe, in small point sizes; this stroke extrudes already to the left at point size 8; for point sizes lower than 8, the amount of extrusion remains stable, otherwise the character shape would be deformed; same phenomenon for the height of the intersection point of the vertical and the oblique stroke of letter final sade: after point size 24 the intersection height remain stable, since at this point size it has reached an extremal point. The idea of this paragraph could be stated as: “morphing should be applied only for interpolations inside the regular range; for extrapolations, the usual metaness (stroke widths etc.) is applied.”

One of the most important parts of many Hebrew letters is the “flame” (or “crown”). Figure 2 shows the different METAFONT reference points and paths used for the definition of a standard METAFONT “flame”-subroutine.

3.2 Rashi

Besides the “quadratic” Hebrew font, which is shown in table A of the Appendix, the author has also developed a *Rashi* font. This type was used in Synagogue books for comments of the Biblical text. Synagogal books, which often are masterpieces of typography combine several point sizes of *Rashi* and *quadratic* in various page setups. On the other hand, Rashi is not used in scholarly editions. Rashi is not diacriticized (neither vowels, nor cantillation marks); nevertheless, the author has included vowels, semi-vowels and šewa in the Rashi font, just in case somebody want’s to break that rule...

The Rashi font has only the usual metaness (optical scaling through variation of stroke width). Here are the characters of pointsize 10, in their real physical size, compared to the quadratic font at the same size:

Rashi:
 Quadratic: אבגדהוזחטיכךלםמננסםפףצקרתש

In a forthcoming article, the author will give examples taken from Synagogal books with comments in Rashi.

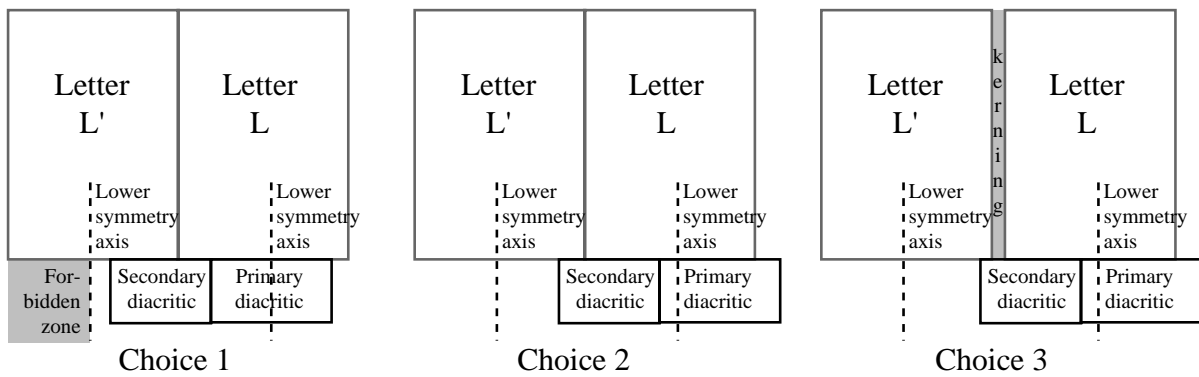


Figure 3: Three possible choices of diacritics positioning, in order of preference.



Figure 4: Upper and lower symmetry axes for Hebrew characters

3.3 An algorithm for placing floating diacritics

After a close study of the typesetting of diacriticized text in the Hebrew Bibles, and numerous discussions with Johannes de Moor, the author was led to the following considerations:

1. We divide the set of diacritics into two categories: *primary* and *secondary*. Primary diacritics are vowels, semi-vowels and šewa (stratum 2); all others are secondary ones (strata 3–4). Secondary diacritics are always appended to the left of primary diacritics belonging to the same letter.
2. Every consonant has an upper and a lower symmetry axis, on which diacritics are centered: these are not necessarily identified with the symmetry axis of the character's box: for example, the lower symmetry axis of character ך is going through the middle of the *vertical stroke*, and not through the middle of the imaginary character box (as its upper symmetry axis).

On fig. 4, the reader can see the choice of upper and lower symmetry axis for each character, as well as the “forbidden zones”, which should be avoided by diacritics¹⁰.

¹⁰The reader will notice two letters *he* on fig. 4, with different lower symmetry axes: the second one shows the axis used for the *pataḥ furtivum*. The same process is applied to letter *het*.

3. Suppose that:

- we have a letter L followed by letter L' ;
- letter L carries a primary diacritic P and a secondary diacritic S , both under it, S being necessarily appended to the left of P ;
- the lower symmetry axis of letter L is A and the one of L' is A' .

Then we have *three possible choices, in the following order of preference*:

- (a) P is centered on A (see fig. 3, 1);
- (b) the group of diacritics SP is centered on A as a whole (see fig. 3, 2);
- (c) the group of diacritics SP is centered on A as a whole, furthermore a kern is added between L' and L (see fig. 3, 3) so that diacritic S does not overlap on L' or it's diacritics.

Once these choices as well as their order of preference have been determined, the algorithm for placing diacritics under (or over) a word is the following:

```
for (letters of the word starting from the left) {  
  try choice (a)  
  if ((a) not succesful) {  
    try choice (b)  
    if ((b) not succesful) {  
      apply choice (c)  
    }  
  }  
  go to next letter  
}
```

where the criterium of “success” is the fact that the diacritics of the current letter do not overlap with the following letter (if this letter has a descender part) or it's diacritics, or it's lower symmetry axis. Here is an example of such a situation. On the following line, the reader can see a few (imaginary) words illustrating the three choices described above:¹¹

‡ בְּבֵב (4) † בְּבֵב (3) † בְּבֵב (2) † בְּבֵב (1)

In case (1), we have three letters *bet*, the medial one having a primary diacritic *segol* and a secondary diacritic *atnaḥ*. On the left side there is no diacritic, and the lower symmetry axis of the left *bet* is far enough from the *atnaḥ* of the medial letter, to allow placement of the diacritics according to choice 1: the *segol* is centered under the letter, and the *atnaḥ* concatenated to it.

¹¹These words are displayed in a magnified 8-point font, so that diacritics are larger, relatively to characters and the three choices become more obvious.

In case (2), instead of *bet* we have placed a *zayin* at the end of the word. There is no diacritic under that letter, but it's lower symmetry axis is much closer to the medial *bet* than it was in case (1); so that now, the diacritics of the medial letter, placed as before, would inevitably touch the symmetry axis of the *zayin*. T_EX automatically switches to choice 2, and checks that, without additional kerning, the diacritics remain indeed inside the authorized area.

In case (3), we add a diacritic *segol* to the letter *zayin*. Choice 2 is not valid anymore, and T_EX automatically kerns letters *zayin* and the *bet* so that the *atnah* is at safe distance of the *segol* to the left of it. This is choice 3, and it always works, because there are no limitations to T_EX's operation of kerning.

Word (4) has been included to show T_EX's reaction in front of a punctuation mark: (a) T_EX does not float the diacritic under the punctuation mark as in case (1), and (b) it doesn't either switch to choice 2, like in (2). The reason is that both operations (a) and (b) are reserved to letters which are considered as part of a whole (the *word*); the punctuation mark belonging to a different entity must be placed independently, and should not participate in the algorithm of floating diacritics placement. As the reader can see in (4), T_EX kerns between the punctuation mark and the letter until the *atnah* is clearly not under it anymore.

NOTES:

1. certain characters have descenders: ק ך ן ף ץ or ascenders: ל; these parts of characters are considered “forbidden zones”—no diacritic should overlap or even touch them (forbidden zones are visible on fig. 4 as shaded areas);
2. the algorithm only concerns diacritics that are *centered* over or under the character, with respect to the symmetry axes shown in fig. 4; not centered diacritics (like the *holem*) obtain fixed positions before applying the algorithm. The region they occupy becomes a *forbidden zone*, just like letter descenders or ascenders;¹²
3. if there are both upper and lower diacritics, the algorithm has to be applied twice, once for each case. Choices are independent, but a possible kerning due to application of choice 3 to one of the two parts, could modify the choice applied to the other part;
4. if there is already a kern between two letters, it must be taken into consideration before applying the algorithm;
5. while inside a line, T_EX is typesetting by counting blank space with respect to character boxes (and not diacritic boxes), at begin of line the maximum between width of diacritic box and width of character box must be taken into account [so that diacritics do not protrude over line begin].

¹²An exception to this rule is the letter ן (*waw* with *holem magnum*), where the right dot is sufficiently lower than the standard diacritic height for additional diacritics to be placed as if the dot wasn't there.

4 Rare cases and typographical curiosa

It is not—and has never been—allowed to change the text of the Hebrew Bible. The Masorets and other Bible commentators have proposed some modifications to the text, which had to be made apparent without changing the text itself. For this reason, many (typo-)graphical tricks have been used to indicate potential modifications of the text. These may differ from one Bible edition to the other (although they seem to be quite stable between rabbinical editions), and may not appear in modern study editions of the Bible, like BHS. Here is a list of such curiosa, after a short search by the author¹³, as well as the way to achieve them with *Tiqwah*.

4.1 Bigger and smaller letters

Letters bigger or smaller than ordinary text. They can appear at any location inside a word. They are vertically justified at the upper bar of Hebrew letters (and not at the baseline), so that big letters are protruding downwards only, and small letters are “hanging”. The eventual *dagesh* dot belongs to the pointsize of the letter itself (bigger or smaller than ordinary text), while the eventual diacritics are typeset in the same size as ordinary text. In the case of big letters, lower diacritics are lowered so that keep the same distance to the letter as in the case of ordinary letters; in the case of small letters they are *not* raised, and remain at their default position.

Here is all possible occurrences the author could detect:

וּבְהִצְטִיף (Gn 27,46), קָצַתִּי (Gn 23,2) וְלִבְיָתָהּ (Gn 2,4) בְּהִבְרָאם (Gn 1,1) בְּרֵאשִׁית (Gn 30,42), הַבְּזוֹנָה (Gn 34,31), שְׁלֹשִׁים (Gn 50,23), נֶצֶר (Ex 34,7) אַחֲרַי (Ex 34,14), מִיִּקְרָא (Lv 1,1), מִיִּקְרָה (Lv 6,2), עַל־גְּהוֹן (Lv 11,42) [big *waw*], וְהִתְנַלַּח (Lv 13,33), וַיִּשְׁרָבֵם (Dt 9,24), מִמְרִים (Dt 9,24), שָׁמַע ... אַחֲרַי (Dt 6,4) [big *yod*], יִגְדֹּל־נָא (Nu 14,17), בְּסוּפָהּ (Jer 14,2) וַיְזַחַת (Jes 56,10) צִפּוֹ (Jos 14,11) כִּכְחִי (Dt 32,6) הִלְיָהוּהָ (Na 1,4), לְשִׂיא (Ps 24,4) [small *waw*], וְכִנָּה (Ps 80,16), מִשְׁלֵי (Prv 1,1) [small final *nun*], וְנִרְגָּו (Prv 16,28), אָדָם (Prv 28,17), הֵב | אֶבֶב (Prv 30,15), וַיִּישַׁשׁ (Hi 7,6), שְׁבִטּוֹ (Hi 9,34), עַל־פְּנֵי־פָרָז (Hi 16,14) [small final *sade*], חָף (Hi 33,9), לְעִיר (Cant 1,1), חִוּר (Qoh 12,13), סוּף (Qoh 7,1), טוֹב (Thr 2,9), מְבַעֵו (Thr 1,12), לְיָנִי (Ru 3,13), וַיִּזְתָּא (Est 9,9) [big *waw* and small *zayin*], פֶּרֶם־שֶׁתָּא (Est 9,9), פֶּרֶם־שֶׁתָּא (Est 9,7), וַתִּכְתֹּב (Est 9,29), בְּשִׁפְרָא (Da 6,20), אָדָם (1Ch 1,1).

To produce big and small letters, one uses the preprocessor directives <BIG> and <SMALL> respectively. These affect only the first letter following them. For example, to obtain

בְּשִׁפְרָא (Da 6,20), one writes <H>b*sI" <SMALL>par" <BIG>p*ArA<TIP>' </H>.

¹³The author would be grateful for any help or suggestion on completing this list.

4.2 Raised ‘ayin

At three locations in the Bible, the author encountered raised letters ‘ayin. Contrary to small letters as described in the previous section, these are typeset in the regular pointsize. The diacritics remain under the normal baseline except in the case of a *patāh* diacritic, which was raised as well, in BHS and BHK.

Here are all occurrences of raised ‘ayin the author could find: מִיֶּעֶר [in [?], [?]] or מִיֶּעֶר [in BHS and BHK] (Ps 80,14), רְשָׁעִים (Hi 38,13), מְרִשָּׁעִים (Hi 38,15).

The raised ‘ayin is a regular character of the *Tiqwah* font and can be produced by the input code ‘/ as shown in table B of the appendix.

4.3 Letters *aleph*, *resh* and ‘ayin with *dageš* dot

The author has found three locations in the Hebrew Bible, where the letter *aleph* takes an *dageš* dot: in BHS the dots are placed in the lower part of the letter, in BHK they are ignored, while in [?], [?] they are placed in the upper or in the lower part of the letter. Here are these occurrences, as they appear in [?], [?]: וַיִּבְאֹן (Gn 43,26), תִּבְאֹן (Lv 23,17), וַיִּבְאֹן (Esr 8,18).

At a single location in the Bible the author found the letter ‘ayin with *dageš*: בְּעַפְרָיִם (1S 5,12). This letter appears in [?], [?] but not in BHS. In BHK a large dot is placed *over* the character.

Finally, the letter *resh* with *dageš* occurs in מְרִתָּה (Prv 14,10).

To produce these letters with *Tiqwah*, use codes ’*, ’/, r* and ‘* as shown in table B of the appendix.

4.4 Letters not obeying rules of contextual analysis

In some cases a letter does not appear in final form as it should, and inversely a letter inside a word is written in final form (for example to indicate a contraction of two words). Here are two cases the author has detected: לְמִרְיָהּ (Jes 9,6), with a final *mem* inside the word, and מִן אֲהַסְעֶרְהָ (Hi 38,1), where the *nun* of the first word is not in final form.

To impose a final form one uses the preprocessor directive <EOW> (EOW stands for “end of word”), after the letter: <H>1"ma<EOW>r"b*e<AZL>h</H> to obtain the example above; to avoid a final form one uses the directive <NIL>, after the letter as well: <H>mi<MER><NIL></H> for the example. More technically, in the first case, the preprocessor considers it is at the end of a word and treats the two parts of the word as distinct—but concatenated—words, in the second case an invisible character of zero width makes it think it is *not* at the end of the word.

4.5 Letters with more than one vowel

Because again of contractions or other grammatical phenomena, a letter can carry more than one vowel: here is an example: כֶּאֱשֶׁר (Ez 9,11), where the letter *kaph* carries both a *šewa* and a *holem*.

Input of such letters is straightforward.

4.6 Isolated *dagesh*

The author encountered an isolated, vowelized *dagesh* in BHS: הַיְיָ (Jes 54,16).

To obtain this character with *Tiqwah*, use the directive <DAGESH>. The invisible box of this character is sufficiently wide to carry vowels and/or other diacritics. It is treated as any other letter, so you have to use the directive <EOW> (see section 2.2.4) to obtain the example beyond (otherwise the letter *nun* will not be final). Here is the *Tiqwah* code of our (unique) example: <H>hin<EOW><DAGESH>e<MEH></H>.

4.7 Inverted letters

In (Nu 10,35–36) as well as in (Ps 107), one encounters the horizontally inverted letter *nun* 𐤎. In the critical apparatus of BHK one can read “𐤎 invers: [editio Bombergiana Jacobi ben Chajjim anni 1524/25] בַּסֵּעַ et כַּמְתֵּאֲנִים”. Both in BHS and BHK the types used for this character are not very satisfactory, while in [?], [?] a type of the same quality as the ordinary *nun* is used.

This character is obtained by the code n/. It seems that other inverted characters may exist (an inverted *lamed* seems to be hidden in the Bible text...). They will be added to the *Tiqwah* system, whenever necessary.

4.8 Missing letters

The treatment of missing letters is typical of the work and restrictions of Masorets: they were not allowed to add letters that were missing, so while vowelizing the consonants they did so also for the missing letters, and by that action made their existence apparent.¹⁴

In [?], [?] an asterisk is used to denote a missing letter. This asterisk is vowelized just like any ordinary letter. In BHS and BHK different methods are used: in some cases, empty space is left; in other cases no empty space is left and the diacritics of the missing character are just squeezed between those of (not missing) letters [a phenomenon occurring also in [?], [?]; for example in the word יְרוּשָׁלַם (Ps 137,6) where a *hireq* is squeezed between the *lamed* and the final *mem*].

Here are the missing letters detected by the author, as printed in [?], [?]: יִשְׂאֵל־* (2S 16,23), עַל־* (2S 18,20), וְתִגְעַשׁ* (2S 22,8), שִׁבְכָה* (1R 7,20), וְאֶת־תִּמָּר (1R 9,18),

¹⁴The reader can compare this with the glasses or gloves worn by the invisible man of the homonymous novell of H. G. Wells.

מִלֶּדֶד (1R 15,18), מְעַל (1R 20,41), מֵאֵין (2R 5,25), רִגִּיר (2R 9,15), מְלֶדֶד (2R 11,20),
 בֵּית־מְלֶדֶד (2R 15,25), וְכַרְמֵל (Jes 32,15), תַּחַת (Jes 55,13), אָרֶץ (Jer 10,13), עָם (Jer 17,19),
 וְהָיוּ (Jer 18,23), דַּבַּר (Jer 40,3), רָשָׁע (Ez 18,20), וְקַרְיַת־מָה (Ez 25,9), רַשְׁכוֹת (Ez 42,9),
 בִּירְכַתָּם (Ez 46,19), יִשְׁנוּ (Prv 4,16), וְיָוֵד (Prv 23,25), עַד (Hi 2,7), יָגִאֵל (Ru 3,13),
 כֹּל־עַמִּים (Thr 1,18), אֵל (Thr 2,2), אֵין (Thr 5,3), הַנִּמְנֵתוֹן (Da 2,9), דִּי (Da 2,43).

In *Tiqwah* one writes <AST> to obtain the letter-like asterisk (attention, the ASCII asterisk * is used only for the *dagesh*, *mappiq* and *šureq* dot!). If one prefers to leave an empty space, one can use the directive <EMPTY>. Contrarily to <NIL>, this one produces an invisible character with *non-zero* width; it can be vowelized just like any character. Finally, <NIL> can be used if we want to squeeze the diacritics of the missing character between the existing characters/diacritics.

Here is an example: <H>b"<AST>ag*iyd</H> will give the (imaginary) word בְּגִיד; by replacing <AST> by <EMPTY> in the code, one would get בֵּי גִיד and finally, by using <NIL> instead of <AST> or <EMPTY>, the result would be בגִיד.

4.9 Missing words

To indicate the location of missing words, all combinations of the preceding techniques are used. In BHK and BHS, empty vowelized characters are used; in [?], [?], a single asterisk, in the middle of the diacritics of the missing word. In a single case, a digit zero is used instead of asterisk. Here are the missing words detected by the author, as printed in [?], [?]: ֶׁ0 (Jdc 20,13), ֶׁ* (2S 8,3), ֶׁ* (2R 19,31), ֶׁ* (2R 19,37), ֶׁ* (Jer 31,37), ֶׁ* (Jer 50,29), ֶׁ* (Ru 3,6), ֶׁ* (Ru 3,17).

To obtain the digit zero in *Tiqwah*, use the directive <ZERO>. For the remaining examples, the directives explained in section 4.8 are used, in a straightforward manner.

5 Conclusion

As hinted by its name (*Tiqwah* means “hope” in Hebrew), the author made this system hoping that it will lead to a revival of Biblical Hebrew typography. Its three main axes (fonts, typesetting, user interface) are based on three powerful programming languages: METAFONT for font creation, T_EX for typesetting, and GNU Flex for preprocessing. The openness and flexibility of these languages guarantees the platform independence and consistency of the *Tiqwah* system.

The author would like to express his gratitude to Prof. Johannes de Moor of the Theologische Universiteit van de Gereformeerde Kerken (Kampen), for his constant and friendly guidance and support. Also he would like to thank Jean Kahn (Paris) for his help in the chase of rare cases and typographical curiosa, and Alan Hoenig (New York), Daniel Navia (Paris) and Reinhard Wonneberger (Mainz) for their warm response and friendly advice.

References

6 pt	8 pt	9 pt	10 pt	11 pt	12 pt	17 pt	24 pt	36 pt
א	א	א	א	א	א	א	א	א
ב	ב	ב	ב	ב	ב	ב	ב	ב
ג	ג	ג	ג	ג	ג	ג	ג	ג
ד	ד	ד	ד	ד	ד	ד	ד	ד
ה	ה	ה	ה	ה	ה	ה	ה	ה
ו	ו	ו	ו	ו	ו	ו	ו	ו
ז	ז	ז	ז	ז	ז	ז	ז	ז
ח	ח	ח	ח	ח	ח	ח	ח	ח
ט	ט	ט	ט	ט	ט	ט	ט	ט
י	י	י	י	י	י	י	י	י
כ	כ	כ	כ	כ	כ	כ	כ	כ
ל	ל	ל	ל	ל	ל	ל	ל	ל
מ	מ	מ	מ	מ	מ	מ	מ	מ
נ	נ	נ	נ	נ	נ	נ	נ	נ
ס	ס	ס	ס	ס	ס	ס	ס	ס
ע	ע	ע	ע	ע	ע	ע	ע	ע
פ	פ	פ	פ	פ	פ	פ	פ	פ
צ	צ	צ	צ	צ	צ	צ	צ	צ
ק	ק	ק	ק	ק	ק	ק	ק	ק
ר	ר	ר	ר	ר	ר	ר	ר	ר
ש	ש	ש	ש	ש	ש	ש	ש	ש
ת	ת	ת	ת	ת	ת	ת	ת	ת

ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג
ה	ו	ז	ח	ט	י	יא	יב	יג

Figure 5: Table A. Hebrew characters in point sizes 6–36 points

א	ב	ג	ד	ה	ו	ז	ח
'	b	g	d	h	w	z	x
ט	י	כך	ל	מם	נן	ס	ע
T	y	k	l	m	n	s	'
פּ	צ׳	ק	ר	שׁ	שׂ	ש׃	ת
p	Y	q	r	W	S	W/	t
א׃	ב׃	ג׃	ד׃	ה׃	ו׃	ז׃	ח׃
'*	b*	g*	d*	h*	w*	z*	x*
ט׃	י׃	כ׃	ל׃	מ׃	נ׃	ס׃	ע׃
T*	y*	k*	l*	m*	n*	s*	'*
פ׃	צ׃	ק׃	ר׃	שׁ׃	שׂ׃	ש׃׃	ת׃
p*	Y*	q*	r*	W*	S*	W/*	t*
	ל	ל׃	׳/l	׳/L	׳/	n/	
	L	L*	'/l	'/L	'/	n/	

Figure 6: Table B. Hebrew letters and their input codes

Name	Hireq	Sere	Segol	Pataḥ	Qameṣ	Holem	Qibbuṣ
Tiberian							
Palestinian							
Babylonian							
Input	i	e	E	a	A	o	u
Alt. input	<HIR>	<SER>	<SGL>	<PAT>	<QAM>	<HOL>	<QIB>

Figure 7: Table C. Hebrew vowels (Tiberian, Palestinian & Babylonian) and their input codes

Šewa	Ḥaṭeph-			Pataḥ		Mater	
	Pataḥ	Segol	Qameṣ	furtivum		lectionis	
"	Ha	HE	HA	+x	+h	w^o	'^o
<SWA>	<HPA>	<HSE>	<HQA>	<PTF>x	<PTF>h*	w<RHO>	'<RHO>
	<UHE>	<USH>	<USA>	<LHE>	<LSH>	<LSA>	

Figure 8: Table D. Special Hebrew vowels, special characters and their input codes









































Silluq	Atnaḥ	Tiphḥa	Mereka	Munaḥ	Mehupak	Mereka kepula	Darga
							
<SIL>	<ATN>	<TIP>	<MER>	<MUN>	<MEH>	<MEK>	<DAR>
Xsil	Xatn	Xtip	Xmer	Xmun	Xmeh	Xmek	Xdar
Galgal	Zaqeph		Rebia magnum	Gereš	Garšayim	Pazer	Pazer magnum
	parvum	magnum					
							
<GAL>	<ZQP>	<ZQM>	<RBM>	<GER>	<GAR>	<PAZ>	<PZM>
Xgal	Xzqp	Xzqm	Xrbm	Xger	Xgar	Xpaz	Xpzm
Azla	'Ole weyored	'Illuy	Šalšelet	Šinnorit	Circellus	Tebir	Rafe
							
<AZL>	<OLE>	<ILL>	<SHP>	<SIN>	<CIR>	<TEB>	<RAF>
Xazl	Xole	Xill	Xshp	Xsin	Xcir	Xteb	Xraf
Punctum extraordi- narium	Postpositivi						
	Varika	Circellus	Lineola (paseq)	Segolta	Sinnor, Zarqa	Pašta	Teliša parvum
							
<PUN>	<VAR>	<PCR>	<LIN>	<SEG>	<ZAR>	<PAS>	<TLP>
Xpun	Xvar	Xpcr	Xlin	Xseg	Xzar	Xpas	Xtlp
Rebia mugrash	Praepositivi			Setuma	Petuḥa	Maqqeph	Soph pasuq
	Dehi	Yetib	Teliša magnum				
							
<REM>	<DEH>	<YET>	<TLM>	<SET>	<PET>	=	:
Xrem	Xdeh	Xyet	Xtlm	Xset	Xpet		

Figure 9: Table E. Oldstyle Hebrew ²⁴isoretic accents and their input codes

Atnaḥ	Ṭiphḥa	Mereka	Mereka kepula	Darga	Galgal	Gereš	Garšayim
<ATN>	<TIP>	<MER>	<MEK>	<DAR>	<GAL>	<GER>	<GAR>
Xatn	Xtip	Xmer	Xmek	Xdar	Xgal	Xger	Xgar
Pazer	Azla	Šinnorit	Postpositivi		Praepositivi		
			Tebir	Sinnor, Zarqa	Pašta	Rebia mugrash	Dehi
<PAZ>	<AZL>	<SIN>	<TEB>	<ZAR>	<PAS>	<REM>	<DEH>
Xpaz	Xazl	Xsin	Xteb	Xzar	Xpas	Xrem	Xdeh

Figure 10: Table F. Modern Hebrew masoretic accents and their input codes

'	''		*		0	.	
!	!!	<MIL>	<AST>	<ASA>	<ZERO>	<DAGESH>	<XXX>
		Xmil	Xast	Xasa	Xzer	Xdag	Xxxx

Figure 11: Table G. Miscellaneous symbols and their input codes

בראשית

GENESIS

Caput I. א

בְּרֵאשִׁית בָּרָא אֱלֹהִים אֶת הַשָּׁמַיִם וְאֶת הָאָרֶץ׃ וְהָאָרֶץ
הָיְתָה תֵהוֹ וּבְהוֹ וְחֹשֶׁךְ עַל־פְּנֵי תְהוֹם וְרוּחַ אֱלֹהִים
מְרַחֶפֶת עַל־פְּנֵי הַמַּיִם׃ וַיֹּאמֶר אֱלֹהִים יְהִי אֹר וַיְהִי־אֹר׃
וַיֵּרָא אֱלֹהִים אֶת־הָאֹר כִּי־טוֹב וַיְבָרֶךְ אֱלֹהִים בֵּין הָאֹר
וּבֵין הַחֹשֶׁךְ׃ וַיִּקְרָא אֱלֹהִים | לְאֹר יוֹם וְלַחֹשֶׁךְ קָרָא לַיְלָה
וַיְהִי־עֶרֶב וַיְהִי־בֹקֶר יוֹם אֶחָד׃

וַיֹּאמֶר אֱלֹהִים יְהִי רָקִיעַ בְּתוֹךְ הַמַּיִם וַיְהִי מְבֹדֵל בֵּין מַיִם
לְמַיִם׃ וַיַּעַשׂ אֱלֹהִים אֶת־הַרְקִיעַ וַיְבָרֶךְ בֵּין הַמַּיִם אֲשֶׁר
מִתַּחַת לַרְקִיעַ וּבֵין הַמַּיִם אֲשֶׁר מֵעַל לַרְקִיעַ וַיְהִי־כֵן׃
וַיִּקְרָא אֱלֹהִים לַרְקָע שָׁמַיִם וַיְהִי־עֶרֶב וַיְהִי־בֹקֶר יוֹם
שֵׁנִי׃

וַיֹּאמֶר אֱלֹהִים יִקְוּ הַמַּיִם מִתַּחַת הַשָּׁמַיִם אֶל־מְקוֹם
אֶחָד וְתֵרָאֵה תִיבְשָׁה וַיְהִי־כֵן׃ וַיִּקְרָא אֱלֹהִים | לְיִבְשָׁה
אָרֶץ וּלְמִקְוֵה הַמַּיִם קָרָא יַמִּים וַיֵּרָא אֱלֹהִים כִּי־טוֹב׃
וַיֹּאמֶר אֱלֹהִים תְּדַשּׁוּא הָאָרֶץ דָּשָׂא עֵשֶׂב מִזְרִיעַ זֶרַע
עֵץ פְּרִי עֹשֶׂה פְרִי לְמִינוֹ אֲשֶׁר זֶרְעוֹ־בּוֹ עַל־הָאָרֶץ וַיְהִי־
כֵן׃ וַתּוּצֵא הָאָרֶץ דָּשָׂא עֵשֶׂב מִזְרִיעַ זֶרַע לְמִינֵהוּ וְעֵץ
עֹשֶׂה־פְרִי אֲשֶׁר זֶרְעוֹ־בּוֹ לְמִינֵהוּ וַיֵּרָא אֱלֹהִים כִּי־טוֹב׃
וַיְהִי־עֶרֶב וַיְהִי־בֹקֶר יוֹם שְׁלִישִׁי׃

Figure 12: The book of Genesis, as printed in an 1889 Viennese Bible