

Evangelion Japanese Font Metric for LuaTeX

<https://github.com/RadioNoiseE/Evangelion-JFM>

<https://www.ctan.org/pkg/evangelion-jfm>

西历 2023 年 黄京

概要

本文档将介绍名为 Evangelion Japanese Font Metric（下简称为“Eva-JFM”）的 JFM 文件。其适用于简体中文（以下简称为「简中」）、繁体中文（以下简称为「繁中」）及日文字体的横直排。旨在提供一个充分利用 LuaTeX-jā 的 `priority` 特性，基于标准 [1] 的同时，支持一些罕用特性的 JFM 文件。文档暂使用中文及西文撰写。

1 背景及略介

TeX 是高德纳教授于 20 世纪末开发的强大排版引擎，能够完全满足西文排版的需求。然因时代局限性^{*1}以及客观原因^{*2}对中日排版支持十分有限。为达成中日排版需求，在宏扩展（如 CJK 等）之外出现了引擎扩展。影响力较大的是 pTeX 系列。

pTeX 系列采用虚拟字体的理念，使用 TFM/VF 映射 TrueType 或 OpenType 字体完成排版。其不支持宏配置字体，也不支持直接生成 PDF 格式文件。但可以满足日本的传统横纵排版需求（工业标准）。

pdfTeX 则是当时另一个 TeX 的引擎扩展，支持不经 DVI 格式直接输出 PDF 格式的文件。然对 Unicode（字符编码）及 TrueType、OpenType（「现代」矢量字体格式）的支持繁琐或有限。

LuaTeX 便是基于 pdfTeX 的引擎扩展，在原生支持 Unicode 下提供 Lua 语言扩展（使能够使用 `fontloader` 等模块）支持现代字体。宏配置字体特性由 `luaotfload` 宏集提供。它 also 支持直接生成 PDF 文件。

LuaTeX-jā 可看作是对两者的合并。这是一个由日本开发者北川弘典首倡的 LuaTeX 下的日文支持项目，即将 pTeX（大部分）移植到 LuaTeX 下。由于 LuaTeX 支持宏配置字体，故不需要 VF 文件为字体提供映射，但为标点挤压等需求保留并扩展^{*3}了 JFM 文件。

本项目就是一个 JFM 文件。使用 LuaTeX 的 `callback`，将简中、繁中、日文及横纵方向、行间标点、悬挂标点、压缩字体等特性集中于 `jfm-eva.lua` 单个文件中。用户可按需调用特性来完成高质量的中日排版。

2 安裝及本地配置

本项目将源文件托管于 GitHub 平台，且已上传至 Comprehensive TeX Archive Net (CTAN)。用户可使用

^{*1} 如没有事实上的统一字符编码等。

^{*2} 如中日字符集较大，以及书写方式的不同（纵书、横书），标点等。

^{*3} 如优先挤压 (`priority`) 特性，及一些特殊字符（如 `parbdd`、`glue`）等。

```
1 tlmgr install evangelion-jfm
```

或使用其他包管理器安装。用户也可使用

```
1 mkdir Evangelion-JFM [ && ] cd Evangelion-JFM
2 git clone https://github.com/RadioNoiseE/Evangelion-JFM
```

获取源文件，再将其放置在本地的 TEXMF 路径中，如

```
1 ~/Library/texlive/2023/texmf-dist/tex/luatex/eva-jfm
```

等。最后运行

```
1 mktexlsr
```

更新本地 \TeX 的 `ls-R` 文件即可。

本文件一般情况下无需用户进行本地配置，但若有特殊需求可见第 5.3 节。

3 使用

以下是在 $\mathbb{L}\mathbb{A}\mathbb{T}\mathbb{E}\mathbb{X}$ 下使用繁体中文字体进行直排的示例

```
1 \usepackage{luatexja-fontspec, luatexja-adjust}
2 \setmainfont{Source Han Serif TC}[Language = Chinese Traditional, TateFeatures = {JFM = eva/{vert, trad,
   nstd}}]
3 \ltjenableadjust[priority = true]
```

(注意需要调用支持直书的文档类或使用 `\tate` 命令)。Lua $\mathbb{T}\mathbb{E}\mathbb{X}$ -ja 的 JFM 语法为：

```
1 jfm = <JFM name>/{\<JFM features>}
```

而一般情况使用 `\setmainfont` 时则为：

```
1 \setmainfont{<font name>}[Language = <language name>, <dir> = {JFM = <JFM name>/{\<JFM features>}}]
```

其中，`` 自然为需要的字体名称。`<language name>` 在使用日文字体时可忽略，而使用简中、繁体中文字体时为必填^{*4}，因 Lua $\mathbb{T}\mathbb{E}\mathbb{X}$ -ja 会默认将其覆盖为 `Japanese` 选项，而这会带来灾难性的后果^{*5}。`<dir>` 选填 `TateFeatures`（直书）或 `YokoFeatures`（横书）。其后的 `<JFM name>` 为调用 JFM 的文件名^{*6}。最后的 `<JFM features>` 选项为选择使用的 JFM 特性，详细请看第 4 章。

对于进阶用户，也推荐用

```
1 \def\ltj@stdyokojfm{eva/{\<JFM features>}}
```

或配合 NFSS 来使用。

其他情况下设置 JFM 及其更多信息请看 Lua $\mathbb{T}\mathbb{E}\mathbb{X}$ -ja 文档 [2]。

^{*4} 简中填 `Chinese Simplified`，繁体中填 `Chinese Traditional` 即可。

^{*5} 比如错误的标点位置：日文为冒号及分号中置、其余偏靠，简中是全部偏靠，而繁体中则是统统中置。

^{*6} Lua $\mathbb{T}\mathbb{E}\mathbb{X}$ -ja 会依 `jfm-<JFM name>.lua` 的格式来查找该文件。

4 支持特性

本章节将介绍 Eva-JFM 的所有特性，分别为：语言特性、方向特性、扩展特性、西文特性及私有特性。

4.1 語言特性

本区特性必填且只可填一个。不然则会报错。

`jp` → (*JaPanese*)

日语特性。当使用日文字体时需调用该特性。其与简中、繁体区别在于问号及感叹号后插入的伸缩胶量。影响特性 `lgp`，且对内部分组有影响。

`trad` → (*TRADitional chinese*)

繁体特性。当使用繁体中文字体时需调用。与简中、日语特性的区别源于中置的标点。故，对于全部标点左右插入的伸缩胶的量都与简中、日语不同。针对句点紧挨闭括号、标点位于句末时等皆有优化。

`simpl` → (*SiMPLified chinese*)

简体特性，使用简体中文字体排版时调用。与日语、繁体特性区别源于分号及冒号等全部偏靠从而影响其左右插入伸缩胶的量。Eva-JFM 对一些（不该出现的）神奇情况（如两个句号同时出现、开括号后出现问号等）进行优化。对问号、感叹号等作了特殊处理。

4.2 方向特性

本分区特性与全部其他特性兼容，可同时调用。

`vert` → (*VERTical writing*)

直书特性。对标点挤压、分组有影响。直书时必须调用。

4.3 擴展特性

本区特性 `hgp` 不依赖 `vert` 特性，其余需同 `vert` 特性同时调用。否则报错。

`extd` → (*EXTenDed font*)

压缩字体特性。默认为横比纵为 100 比 80 的字体压缩^{*7}。可用 `extd=<ratio>` 设置横方向拉伸比例（默认即为 1.25。需同 `extend` (`luaotfload`) 或 `FakeStretch` (`fontspec`) 同时使用。

`lgp` → (*LineGap Punctuations*)

行间标点特性。该特性将部分标点「悬挂」至行间。日文字体时与繁、简中字体时会有区别。详见第 5 章。

`hgp` → (*HanGing Punctuations*)

悬挂标点特性。该特性将部分标点「悬挂」于行末。仅简中、日文字体拥有该特性。

^{*7} 日本新闻字体，如每日新闻明朝体。

4.4 西文特性

本区特性使用时需先使用`\ltjsetparameter` 设置 `jacharrange` 从而调整 JAchar 的范围。

```
\ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, -8, +9}}
```

同时推荐与对应 OpenType 特性同时使用。

`hwid` \rightarrow (*Half WIDTH*)

半宽西文特性。使用此特性（且按上述设置完成后）西文字母排布为严格半宽。本特性不会压缩或拉伸西文字母，故当未使用对应半宽字体特性时，只会简单的重叠，此时不推荐使用。同时也将失去所有 `kern` 以及 `italic correction` 的数据，同时忽略 `xkanjiskip` 参数。请务必谨慎调用。

`fwid` \rightarrow (*Full WIDTH*)

全宽西文特性。描述同上。但，若不调用全宽特性，西文间距会被简单撑开。

4.5 私有特性

使用本区特性前请先确保你清楚地知道你在做什么。

`nstd` \rightarrow (*Non STandard*)

忽略标准特性。字体排印标准 [1] 认为逗号的压缩权重应比句号要低。本特性将句号的压缩优先级与逗号交换，使逗号被优先压缩^{*8}。仅在使用 `luatexja-adjust` 宏集时有效。

5 行間標點特性

本章节将提供更多详细的关于行间标点特性的信息，以及可能出现的问题及其解决方案。

5.1 關於「懸掛」

行间标点可见于古籍之中，是将标点符号与直书结合妥协的产物。

传统上悬挂句号与逗号。而 `Eva-JFM` 特性在繁中、简中特性下会悬挂句号、逗号、顿号、冒号及分号，日文字体下则不悬挂冒号及分号。原因在于日本习惯上将冒号与分号看作「中点类」，直书时横置处理。

本 JFM 将全部标点悬挂于字体右下位置。详见下一节。

5.2 懸掛的位置

标点悬挂的位置有以下考量，可参照图 1。若有特殊需求请看第 5.3 节。优先级由上至下。

- 三种字体风格统一，位置原则上一致（故，繁中字体也悬挂于右下、而非居中）；
- 不同标点中的相同（似）元素位置相同；
- 繁中、简中、日文字体标点触字框右边线；

^{*8} 考虑逗号、句号在文字系统中占的重量，以及「开明」压缩风格。

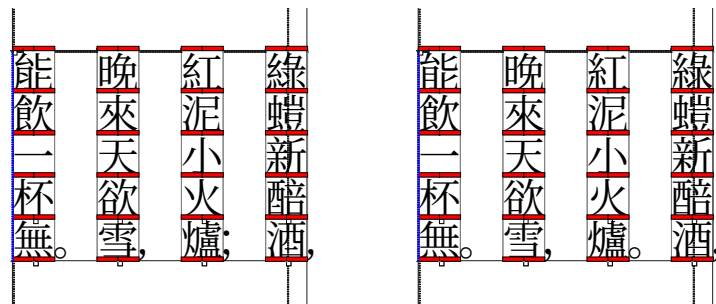


图 1 行间标点特性示意图

- 不同标点符号因形状不同可于字框底线略下沉或上浮；
- 不同标点符号因大小不同可靠近或远离字框右边线；
- 三种字体可分别因字符设计的差异而位置略微区别。

5.3 用戶配置

本特性是以三套思源字体为基准设计的。而由于各字体的标点符号位置不可避免会有不同，故在某些特殊情况下会出现错位影响视觉效果的情况。或是单纯对原设定而言更偏好其他设定等原因，本节提供自定义及调整的两种方法。第一种较简单但可移植性较差，而第二种虽繁琐但一劳永逸。

5.3.1 修改原程式碼

在 Eva-JFM 中，控制行间标点的分区域别为

```
1 [101,2] ==> [1]; [201,2] ==> [2]; [301,2] ==> [3].
```

只需调整其中 `left` 和 `down` 键的值即可。其中 `left` 为向右移动，`down` 为向下移动。具体可参照终章。

5.3.2 使用外掛符號字體

该方法的原理就是使用特殊的仅包含（标点）符号的字体来替换原有字体中的标点符号，从而稳定其表现。可使用字体熔炉等工具将标点符号从整套字体中提取出来并封装为新字体，也可使用开源符号字体。

将其放入 TEXMF 并更新 Ls-R 文件后即可使用 Lua_T_EX-j_a 提供的 `AltFont` 键进行替换，例元：

```
1 \setmainfont[
2   Language = <language>,
3   TateFeatures = {
4     JFM = eva/{vert, lgp, <language>},
5     AltFont = {
6       {Range = "<utf-8 code>", Font = <symbol font>}
7     }
8   }
9 ]{<main font>}
```

其中首个 `<language>` 可选填 `Japanese`、`Chinese Traditional` 或 `Chinese Simplified`，第二个则填语言特性

分区的对应 `jp`、`trad` 及 `smp1` 特性。`<utf-8 code>` 则为需要替换的标点符号的 Unicode 编码，如需替换句号 (ideographic full stop, U+3002)⁹。`<symbol font>` 以及 `<main font>` 填符号字体名称、正文字体名称即可。

对于开发者，也建议使用 NFSS 的

```
1 \DeclareAlternateKanjiFont{<base encoding>}{<base family>}{<base series>}{<base shape>}{<alt encoding>}{<alt  
family>}{<alt series>}{<alt shape>}{<range>}
```

进行替换。其中 `<base>` 为正文字体，`<alt>` 则为替换符号字体。

具体语法及示例可看 Lua \TeX -ja 文档 [2]。

6 后發

Eva-JFM 的内部分组受 `min10.tfm` [5] 的启发，支持的 `priority` 特性则取自阿部紀行氏的 `jlreq.lua` [6] 文件。其余可见参考文献。

本 JFM 的名字来源于庵野秀明的『新世紀エヴァンゲリオン』。

参考文献

- [1] W3C Japanese Layout Task Force (ed). Requirements for Japanese Text Layout (W3C Working Group Note), 2022, 2023. <https://www.w3.org/TR/jlreq/>.
- [2] Lua \TeX -ja プロジェクトチーム. Lua \TeX -ja パッケージ, 2022, 2023.
- [3] The Unicode Consortium. The Unicode Standard Version 15.0 - Core Specification, 2022.
- [4] Victor Eijkhout. \TeX by Topic, A \TeX nician's Reference, Addison-Wesley, 1992.
- [5] 乙部徹己. min10 フォントについて. <http://argent.shinshu-u.ac.jp/~otobe/tex/files/min10.pdf>.
- [6] Noriyuki Abe. Jreq Document Class, 2022. <https://github.com/abenori/jlreq>.
- [7] 庵野秀明. 新世紀エヴァンゲリオン.

程式碼

以下为 `jfm-eva.lua` 文件内容，供参考。

```
1 ---- Evangelion Japanese Font Metric for LuaTeX
2 ---- Current Version: 1.0.2 (e)
3 ---- Dev URL: https://github.com/RadioNoiseE/Evangelion-JFM
4 ---- Copyright 2023, RadioNoiseE ©
5
6
7 -- 初始化
8 local lang_jp, lang_tc, lang_sc, dir_vt, font_extd, punc_lg, punc_hg, std_nil, al_hw, al_fw
9
```

⁹ 编码可至 <https://www.unicode.org/charts/unihanrsindex.html> 查询。

```

10 if luatexja.jfont.jfm_feature then
11     lang_jp = luatexja.jfont.jfm_feature.jp
12     lang_tc = luatexja.jfont.jfm_feature.trad
13     lang_sc = luatexja.jfont.jfm_feature.smpl
14     dir_vt = luatexja.jfont.jfm_feature.vert
15     font_extd = luatexja.jfont.jfm_feature.extd
16     punc_lg = luatexja.jfont.jfm_feature.lgp
17     punc_hg = luatexja.jfont.jfm_feature.hgp
18     std_nil = luatexja.jfont.jfm_feature.nstd
19     al_hw = luatexja.jfont.jfm_feature.hwid
20     al_fw = luatexja.jfont.jfm_feature.fwid
21 end
22
23 -- 預處理及容錯
24 if font_extd == true and dir_vt == false then
25     tex.error('JFM feature "extd" only works with feature "vert".\n' ..
26             'For now I\'ll ignore it.')
27 end
28
29 if punc_lg == true and dir_vt == false then
30     tex.error('JFM feature "lgp" only works with feature "vert".\n' ..
31             'For now I\'ll ignore it.')
32 end
33
34 if al_hw == true and al_fw == true then
35     tex.error('JFM feature "hwid" cannot be used with "fwid".')
36 end
37
38 if not ((lang_jp and not (lang_tc or lang_sc)) or
39         (lang_tc and not (lang_jp or lang_sc)) or
40         (lang_sc and not (lang_jp or lang_tc))) then
41     tex.error('Specify one and only one feature from three language specific features\n' ..
42             '"jp", "trad" or "smpl"\n' ..
43             'is required.\n' ..
44             'For now I\'ll use "lang_jp" for japanese by default.')
45 end
46
47 -- 壓縮比例設定
48 if font_extd == true then
49     local extd_ratio = (type(font_extd) == 'string') and tonumber(font_extd) or 1.25
50 end
51
52 -- 行間標點字間距補足
53 local lgp_kanjiskip = {kanjiskip_natural = 0, kanjiskip_stretch = 1, kanjiskip_shrink = 1}
54
55 -- 定義函數宏
56 local function logic_anif(f1, f2, r1, r2)
57     local rta = f1 and (f2 and r1) or r2
58     return rta

```

```

59 end
60
61 local function logic_if(f1, r1, r2)
62     local rti = f1 and r1 or r2
63     return rti
64 end
65
66 local function context_height()
67     local rth = dir_vt and (font_extd and extd_ratio/2 or 0.5) or 0.88
68     return rth
69 end
70
71 local function context_depth()
72     local rtd = dir_vt and (font_extd and extd_ratio/2 or 0.5) or 0.12
73     return rtd
74 end
75
76 -- 主體
77 local eva = {
78     version = 3,
79     dir = logic_if(dir_vt, 'tate', 'yoko'),
80     zw = 1,
81     zh = logic_anif(dir_vt, font_extd, extd_ratio, 1),
82     kanjiskip = {0, 0.25, 0},
83     xkanjiskip = {0.25, 0.125, 0.125},
84
85     [0] = { -- 缺省類
86         width = 1,
87         height = context_height(),
88         depth = context_depth(),
89         italic = 0,
90         left = 0,
91         down = 0,
92         align = 'middle',
93         glue = {
94             [1] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, 0}, {-1,
-2}}), {priority = logic_if(std_nil, {-1, 0}, {-1, -2}})),
95             [2] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = logic_if(std_nil, {-1, -2}, {-1,
0}}), {priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
96             [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1,
priority = {-1, -1}}, {priority = {-1, -1}})),
97             [7] = {0.5, 0, 0.25, ratio = 1, priority = {-1, -2}},
98             [9] = {0.25, 0, 0.125, ratio = 1, priority = {-1, -1}}
99         },
100         round_threshold = 0.01
101     },
102
103     [1] = { -- 読点類
104         chars = logic_anif(dir_vt, punc_lg, {}, {'\ ', ' '}),

```



```

105     width = 0.5,
106     height = context_height(),
107     depth = context_depth(),
108     italic = 0,
109     left = 0,
110     down = 0,
111     align = logic_if(lang_tc, 'middle', 'left'),
112     glue = {
113         [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
114 -2}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2}})),
115         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
116         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25}),
117         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5,
118 0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
119 ratio = 0, priority = {0, -1}})),
120         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
121 -2}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2}})),
122         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0,
123 -2}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, 0}, {0, -2}})),
124         [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
125 -2}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2}})),
126         [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1,
127 -2}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, 0}, {-1, -2}})),
128         [8] = logic_if(lang_tc, {0.25, 0, 0.125}, {}),
129         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
130 priority = {0, -1}})
131     },
132     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0.5, 0}, {0, 0}))
133 },
134
135 [101] = { -- 読点類 (行間a)
136     chars = logic_anif(dir_vt, punc_lg, {'\'}, {}),
137     width = 0,
138     height = context_height(),
139     depth = context_depth(),
140     italic = 0,
141     left = 0.38,
142     down = -0.34,
143     align = 'left',
144     glue = {
145         [0] = lgp_kanjiskip
146     }
147 },
148
149 [102] = { -- 読点類 (行間b)
150     chars = logic_anif(dir_vt, punc_lg, {'\'}, {}),
151     width = 0,
152     height = context_height(),
153     depth = context_depth(),

```

```

146     italic = 0,
147     left = logic_if(lang_tc, 0.62, 0.40),
148     down = logic_if(lang_tc, -0.58, -0.26),
149     align = 'left',
150     glue = {
151         [0] = lgp_kanjiskip
152     }
153 },
154
155 [2] = { -- 句點類
156     chars = logic_anif(dir_vt, punc_lg, {}, {'.', 'o'}),
157     width = 0.5,
158     height = context_height(),
159     depth = context_depth(),
160     italic = 0,
161     left = 0,
162     down = 0,
163     align = logic_if(lang_tc, 'middle', 'left'),
164     glue = {
165         [0] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
166             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
167         [1] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
168         [2] = logic_if(lang_tc, {0.5, 0, 0.25}, {0.5, 0, 0.25, ratio = 0}),
169         [3] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5,
170             0, 0.25, priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25,
171             ratio = 0, priority = {0, -1}})),
172         [4] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
173             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0}})),
174         [5] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0,
175             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {0, -2}, {0, 0}})),
176         [6] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
177             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
178         [7] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1,
179             0}}), {0.5, 0, 0.25, ratio = 0, priority = logic_if(std_nil, {-1, -2}, {-1, 0}})),
180         [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0}, {}),
181         [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
182             priority = {0, -1}})
183     },
184     end_adjust = logic_if(lang_tc, {0.25, 0}, logic_if(punc_hg, {-0.5, 0.5, 0}, {0, 0}))
185 },
186
187 [201] = { -- 句點類 (行間a)
188     chars = logic_anif(dir_vt, punc_lg, {'.'}, {}),
189     width = 0,
190     height = context_height(),
191     depth = context_height(),
192     italic = 0,
193     left = logic_if(lang_tc, 0.68, 0.34),
194     down = logic_if(lang_tc, -0.58, -0.28),

```

```

187     align = 'left',
188     glue = {
189         [0] = lgp_kanjiskip
190     }
191 },
192
193 [202] = { -- 句點類 (行間b)
194     chars = logic_anif(dir_vt, punc_lg, {'。'}, {}),
195     width = 0,
196     height = context_height(),
197     depth = context_height(),
198     italic = 0,
199     left = 0.42,
200     down = -0.35,
201     align = 'left',
202     glue = {
203         [0] = lgp_kanjiskip
204     }
205 },
206
207 [3] = { -- 兩點類
208     chars = logic_if(lang_jp, {}, (logic_anif(dir_vt, punc_lg, {}, {'：', ';' }))),
209     width = logic_if(dir_vt, 1, 0.5),
210     height = context_height(),
211     depth = context_depth(),
212     italic = 0,
213     left = 0,
214     down = 0,
215     align = logic_if(lang_tc, 'middle', 'left'),
216     glue = {
217         [0] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
218 priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
219         [1] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
220 priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0,
221 priority = {0, -1}})),
222         [2] = logic_if(dir_vt, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 1, priority = {0, -1}}, {
223 priority = {0, -1}}), logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0,
224 priority = {0, -1}})),
225         [3] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.5, 0, 0.25, priority = {0,
226 -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
227         [4] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
228 priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
229         [5] = logic_if(dir_vt, {priority = {0, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
230 priority = {0, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {0, -1}})),
231         [6] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
232 priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),
233         [7] = logic_if(dir_vt, {priority = {-1, -1}}, logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0,
234 priority = {-1, -1}}, {0.5, 0, 0.25, ratio = 0, priority = {-1, -1}})),

```

```

225     [8] = logic_if(lang_tc, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, {0.5, 0, 0.25, ratio =
226     0, priority = {0, -1}}),
227     [9] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.75, 0, 0.25, ratio = 1/3,
228     priority = {0, -1}})
229     }
230 },
231
232 [301] = { -- 兩點類 (行間a)
233     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {':'}, {})),
234     width = 0,
235     height = context_height(),
236     depth = context_depth(),
237     italic = 0,
238     left = logic_if(lang_tc, 0.94, 0.72),
239     down = logic_if(lang_tc, -0.58, -0.34),
240     align = 'left',
241     glue = {
242         [0] = lgp_kanjiskip
243     }
244 },
245
246 [302] = { -- 兩點類 (行間b)
247     chars = logic_if(lang_jp, {}, logic_anif(dir_vt, punc_lg, {';'}, {})),
248     width = 0,
249     height = context_height(),
250     depth = context_depth(),
251     italic = 0,
252     left = logic_if(lang_tc, 0.96, 0.78),
253     down = logic_if(lang_tc, -0.58, -0.34),
254     align = 'left',
255     glue = {
256         [0] = lgp_kanjiskip
257     }
258 },
259
260 [4] = { -- 小書きの仮名類
261     chars = {
262         'あ', 'い', 'う', 'え', 'お', 'っ', 'や', 'ゆ', 'よ', 'わ', 'か',
263         'け', 'く', 'ぐ', 'ア', 'イ', 'ウ', 'エ', 'オ', 'ツ', 'ヤ', 'ユ',
264         'ヨ', 'ワ', 'カ', 'ケ', 'ハ', 'ヘ', 'ベ', 'ク', 'シ', 'ス', 'ト', 'ヌ',
265         'ハ', 'ヒ', 'フ', 'ヘ', 'ホ', 'ム', 'ラ', 'リ', 'ル', 'レ', 'ロ'
266     },
267     width = 1,
268     height = context_height(),
269     depth = context_depth(),
270     italic = 0,
271     left = 0,
272     down = 0,
273     align = 'middle',

```



```

362 },
363
364 [9] = { -- 中點類
365     chars = logic_if(lang_jp, {'・', '・', '・'}, {'・', '・'}),
366     width = 0.5,
367     height = context_height(),
368     depth = context_depth(),
369     italic = 0,
370     left = 0,
371     down = 0,
372     align = 'middle',
373     glue = {
374         [0] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
375         [1] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}),
376         [2] = logic_if(lang_tc, {0.5, 0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, ratio = 0,
priority = {0, -1}}),
377         [3] = logic_if(dir_vt, {0.25, 0, 0.125, ratio = 0, priority = {0, -1}}, logic_if(lang_tc, {0.5,
0, 0.25, priority = {0, -1}}, {0.25, 0, 0.125, priority = {0, -1}})),
378         [4] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
379         [5] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
380         [6] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
381         [7] = {0.25, 0, 0.125, ratio = 0, priority = {-1, -1}},
382         [8] = {0.25, 0, 0.125, ratio = 0, priority = {0, -1}},
383         [9] = {0.5, 0, 0.25, priority = {0, -1}}
384     },
385     end_adjust = {0.25, 0}
386 },
387
388 [10] = { -- 西文
389     chars = {},
390     width = 0,
391     height = context_height(),
392     depth = context_depth(),
393     italic = 0,
394     left = 0,
395     down = 0,
396     align = 'middle',
397     glue = {}
398 },
399
400 [11] = { -- 行頭
401     chars = {'parbdd', 'boxbdd'},
402     glue = {
403         [7] = {0, 0, 0}
404     }
405 }
406 }
407

```

```

408 if al_hw == true or al_fw == true then
409     eva[10].chars = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
410                     'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
411                     'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M',
412                     'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'}
413     eva[10].glue = table.fastcopy(eva[0].glue)
414     eva[10].glue[0] = {0.25, 0.125, 0.125, ratio = 0, priority = {0, -1}}
415     eva[0].glue[10] = {0.25, 0.125, 0.125, ratio = 1, priority = {0, -1}}
416     for _, catnum in ipairs({1, 2, 3, 5, 8, 9}) do
417         eva[catnum].glue[10] = table.fastcopy(eva[catnum].glue[0])
418     end
419 end
420
421 if al_hw == true and al_fw == false then
422     eva[10].width = 0.5
423 end
424
425 if al_fw == false and al_fw == true then
426     eva[10].width = 1
427 end
428
429 luatexja.jfont.define_jfm(eva)

```