

# Τὸ Εὐτυπον

Τεῦχος 3, Ὀκτώβριος 1999

## Σ' ΑΥΤΟ ΤΟ ΤΕΥΧΟΣ:

Ἐπιφυλλίδα .....	iii
Σημειώσεις τοῦ τυπογράφου .....	iv
Ιωάννης Δημάκος	
pdfTEX: Συνδυάζοντας την ευελιξία του PDF	
με την δύναμη του TEX .....	1
Jed Parsons	
How to typeset an archeological book with L <sup>A</sup> T <sub>E</sub> X ....	13
Richard W. D. Nickalls	
MathsPIC: A filter program for use with P <sub>I</sub> C <sub>T</sub> E <sub>X</sub> .....	33
James Killfiger	
Creating device independent PostScript .....	51
Απόστολος Συρόπουλος και Αντώνης Τσολομούτης	
Εγκατάσταση νέων γραμματοσειρών στο L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> .....	57
Φώτης Ανδριτσόπουλος	
Το L <sup>A</sup> T <sub>E</sub> X και άλλοι κειμενογράφοι .....	69
Απόστολος Συρόπουλος	
TEXνικές: Δημιουργία νέων συμβόλων από ήδη	
υπάρχοντα .....	71
Δημήτριος Α. Φιλίππου	
Βιβλίο-Παρουσίαση .....	73



Μία περιοδική έκδοση  
τοῦ Συλλόγου Ἑλλήνων Φίλων τοῦ TEX

ΞΑΝΘΗ

# Τὸ Εὐτυπον

ISSN 1108-4170

Εὐτυπον (*Eutupon*) is a publication of the “Greek T<sub>E</sub>X Friends” (\*εφτ\*) group. It is published twice a year and it is distributed free to the members of the \*εφτ\* group. Its articles deal with T<sub>E</sub>X and, in general, with tools for the electronic typesetting of printed matter.

The address of Εὐτυπον is:

Greek T<sub>E</sub>X Friends  
(c/o A. Syropoulos)  
366, 28th October Street  
GR-671 00 Xanthi  
Greece

URL: [obelix.ee.duth.gr/eft](http://obelix.ee.duth.gr/eft)  
E-mail: [eft@platon.ee.duth.gr](mailto:eft@platon.ee.duth.gr)

The membership fees to the \*εφτ\* Group are:

Admission:  
5.000 GRD (30 USD)  
6-month subscription:  
3.000 GRD (20 USD)  
Annual subscription:  
5.000 GRD (30 USD)

(All cheques should be made payable to Apostolos Syropoulos.)

Articles already published in Εὐτυπον may be copied and redistributed for free, provided their origin is mentioned. However, neither Εὐτυπον nor the \*εφτ\* group assume any responsibility for methods, products, instructions or ideas described or expressed in articles published herein. The content of articles published herein is the sole responsibility of the respective authors.

Τὸ Εὐτυπον ἀποτελεῖ μία περιοδικὴ ἔκδοση τοῦ Συλλόγου Ἑλλήνων Φίλων τοῦ T<sub>E</sub>X (\*εφτ\*). Κυκλοφορεῖ ἀνά ἐξάμηνο καὶ διανέμεται δωρεάν στὰ μέλη τοῦ Συλλόγου. Στὶς σελίδες τοῦ δημοσιεύονται ἄρθρα σχετικὰ μὲ τὸ T<sub>E</sub>X καὶ τὴν ἠλεκτρονικὴ στοιχειοθεσία ἐντύπων γενικώτερα.

Ἡ διεύθυνση τοῦ Εὐτύπου εἶναι:

Σύλλογος Ἑλλήνων Φίλων τοῦ T<sub>E</sub>X  
(ὕπ' ὄψη: Ἄ. Συρόπουλος)  
28ης Ὀκτωβρίου 366  
671 00 Ξάνθη

URL: [obelix.ee.duth.gr/eft](http://obelix.ee.duth.gr/eft)  
E-mail: [eft@platon.ee.duth.gr](mailto:eft@platon.ee.duth.gr)

Συνδρομὲς μέλους στὸν Σύλλογο \*εφτ\*:

Ἐγγραφή:  
5.000 δρχ. (30 δολ. ΗΠΑ)  
Συνδρομὴ ἐξάμηνη:  
3.000 δρχ. (20 δολ. ΗΠΑ)  
Συνδρομὴ ἐτήσια:  
5.000 δρχ. (30 δολ. ΗΠΑ)

(Ἐπιταγὲς θὰ πρέπει νὰ ἀποστέλλονται στὸ ὄνομα τοῦ Ἀποστόλου Συρόπουλου.)

Ἐπιτρέπεται ἡ ἀναδημοσίευση καὶ διανομὴ ἄρθρων ποὺ ἔχουν ἤδη δημοσιευθεῖ στὸ Εὐτυπον, ὑπὸ τὸν ὅρο ὅτι θὰ ἀναφέρεται ἡ προέλευσή τους. Ὡστόσο, τόσο τὸ Εὐτυπον ὅσο καὶ ὁ Σύλλογος \*εφτ\* δὲν ἀναλαμβάνουν καμία εὐθύνη γιὰ μεθόδους, προϊόντα, ὁδηγίες καὶ ιδέες ποὺ περιγράφονται ἢ ἐκφράζονται ἐντὸς ἄρθρων δημοσιευμένων στὸ περιοδικό. Τὸ περιεχόμενο τῶν ἄρθρων ποὺ δημοσιεύονται ἐντὸς τοῦ περιοδικοῦ ἀποτελεῖ εὐθύνη τῶν ἀντιστοιχῶν συγγραφέων.

ΤΕΧνίτριες και ΤΕΧνίτες!

Πρώτα-πρώτα, θά θέλαμε νά ποῦμε δημοσίως — ἔστω καί μέ καθυστέρηση — ἕνα μεγάλο εὐχαριστῶ στὸν Γιάννη Χαραλάμπους, ποὺ ἀνέλαβε τὴν στοιχειοθεσία τοῦ προηγουμένου τεύχους μὲ τὸ Ω. Πιστεύουμε ὅτι τὰ σχετικά ἄρθρα τοῦ Γιάννη, ἀλλὰ καί ἡ γενικώτερη ἐμφάνιση τοῦ προηγουμένου τεύχους σᾶς ἔδωσαν μία καλὴ ἰδέα γιὰ τὶς δυνατότητες τοῦ Ω. (Γιὰ ὅσα παροράματα βρήκατε στὸ προηγούμενο τεῦχος, τὴν εὐθύνη φέρουν οἱ ὑπογράφοντες καὶ ὄχι ὁ Γιάννης.) Ἐπιπλέον ἐλπίζουμε ὅτι τὸ προηγούμενο τεῦχος θά ἀποτελέσει μία καλὴ ἀφορμὴ γιὰ μία εὐρεία συζήτηση γιὰ τὸ Ω καὶ τὴν ἐξέλιξη τοῦ ΤΕΧ.

Στὸ παρὸν τεῦχος, ὁ Ἰωάννης Δημάκος μᾶς παρουσιάζει τὸ pdfΤΕΧ, ἐνῶ Jed Parsons μᾶς λέει πῶς στοιχειοθέτησε ἕνα ἀρχαιολογικὸ βιβλίο μὲ τὸ IATEX. Ὁ Richard Nickalls μᾶς εἰσαγάγει στὸ MathsPIC, καὶ ὁ James Killfiger ἀναλύει τὶς δυσκολίες γιὰ τὴν δημιουργία γραμματοσειρῶν ποὺ δὲν θά ἐξαρτῶνται ἀπὸ τὸ πρόγραμμα ἢ τὸ λειτουργικὸ σύστημα μὲ τὸ ὁποῖο δουλεύουμε. Τέλος, ὁ Ἀπόστολος Συρόπουλος καὶ ὁ Ἀντώνης Τσολομούτης συνεχίζουν τὴν περιγραφή μεθόδων γιὰ χρῆση διαφόρων γραμματοσειρῶν μὲ τὸ IATEX 2<sub>ε</sub>. Σὲ ἕνα ἄλλιώτικο ἄρθρο, ὁ Φώτης Ἀνδριτσόπουλος ἐξηγεῖ τοὺς λόγους ποὺ τὸν ὤθησαν στὸ IATEX. Τὸ τεῦχος κλείνει μὲ τὰ τακτικὰ ἄρθρα τῶν ΤΕΧνικῶν καὶ τῆς Βιβλιοπαρουσίασης.

Πρὶν σᾶς ἀφήσουμε νά ξεφυλλίσετε τὸ τεῦχος, θά θέλαμε νά σᾶς ὑπενθυμίσουμε πῶς «οἱ καλοὶ λογαριασμοὶ κάνουν τοὺς καλοὺς φίλους». Ὁ Σύλλογος ἔχει ξεκινήσει μία σοβαρὴ προσπάθεια γιὰ τὴν ἐξάπλωση τοῦ ΤΕΧ στὸν ἐλληνικὸ χῶρο μὲ τὸ Εὐτύπον καὶ ἄλλες δημοσιεύσεις. Γιὰ νά μπορέσουμε νά συνεχίσουμε τὸ ἔργο τῆς διάδοσης τοῦ ΤΕΧ, χρειάζονται κάποια χρήματα. Ἡ τακτικὴ πληρωμὴ τῶν συνδρομῶν εἶναι συνεπῶς καθῆκον ὅλων μας. Καὶ ὅπως πάντα, σᾶς τονίζουμε ὅτι ὁ Σύλλογος ἔχει τὶς πόρτες ἀνοιχτὲς γιὰ ὅλους καὶ οἱ σελίδες τοῦ Εὐτύπου εἶναι διαθέσιμες σὲ ὅσους θέλουν νά παρουσιάσουν κάτι σχετικὸ μὲ τὸ ΤΕΧ καὶ τὴν ψηφιακὴ τυπογραφία ἢ ἀπλῶς νά ποῦν τὴν γνώμη τους γιὰ κάτι ποὺ διάβασαν, ἄκουσαν ἢ βρῆκαν.

Ἔως τὴν ἐπόμενη ἐπικοινωνία μας, σᾶς εὐχόμαστε ὅ,τι καλύτερο καὶ ΤΕΧνικώτερο!

– Ἀπόστολος Συρόπουλος  
– Δημήτριος Ἄ. Φιλίππου

Τὸ Εὐτυποῦν στοιχειοθετήθηκε μὲ τὸ L<sup>A</sup>T<sub>E</sub>X. Ὡς βασικά πακέτα χρησιμοποιήθηκαν τὸ `cah-gut.cls` (τὸ πακέτο τοῦ γαλλικοῦ περιοδικοῦ *Les Cahiers GUTenberg*) καὶ τὸ `babel` μὲ τὴν ἀγγλικὴ καὶ τὴν ἐλληνικὴ ἐπιλογή. Γιὰ τὶς διαφορὲς γραφικὲς παραστάσεις, χρησιμοποιήθηκαν ἐπίσης τὰ πακέτα `graphics`, `graphicx` καὶ `color`.

Ἡ εἰκόνα στὴν σελ. 76 εἶναι παρμένη ἀπὸ ἓνα χαρακτηριστὸ σὲ ξύλο τοῦ ἀγγλοῦ ζωγράφου καὶ χαράκτη William Holman Hunt (1827–1910) μὲ τίτλο *The Lady of Shalott* (1857).

Ἡ ἐκτύπωση ἔγινε μὲ τὸ πρόγραμμα Ghostscript σὲ ἐκτυπωτὴ HP LaserJet V (1200 dpi), ἐνῶ ἡ βιβλιοδεσία ἔγινε στὸ Τυπογραφεῖο τοῦ Δημοκρατεῖου Πανεπιστημίου Θράκης στὴν Ξάνθη.

# pdfT<sub>E</sub>X: Συνδυάζοντας την ευελιξία του pdf με την δύναμη του T<sub>E</sub>X

---

Ιωάννης Δημάκος

*American College of Thessaloniki*  
idimakos@otenet.gr

## Abstract

pdfT<sub>E</sub>X is a program that directly generates PDF files from T<sub>E</sub>X sources (either plain T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X), without getting into the trouble to generate a DVI file, then a PS file and finally the PDF file. In this article we provide a thorough presentation of the program and its functionality.

Στο πρώτο κεφάλαιο του T<sub>E</sub>Xbook ο Donald Knuth αναφέρει πως σκοπός του T<sub>E</sub>X είναι η παραγωγή όχι απλώς αποδεκτών κειμένων, αλλά κειμένων αρίστης ποιότητας. Στα χρόνια που ακολούθησαν, το T<sub>E</sub>X (αλλά και τα μακρο-πακέτα, όπως το L<sup>A</sup>T<sub>E</sub>X, το A<sub>M</sub>S-T<sub>E</sub>X, το A<sub>M</sub>S-L<sup>A</sup>T<sub>E</sub>X κλπ, αλλά και βοηθητικά προγράμματα, όπως το BIBT<sub>E</sub>X, το METAFONT κ.λπ.) έγιναν το de facto πρότυπο στην παραγωγή παντός τύπου επιστημονικών εντύπων (βιβλία, άρθρα, φυλλάδια κ.λπ.) υψηλότερης ποιότητας.

Η καθιέρωση του Δικτύου και του Παγκόσμιου Ιστού (νεο-ελληνιστί, Internet και World Wide Web) επέτρεψε τη διάθεση, μετάδοση και αποθήκευση εγγράφων ανά τον κόσμο σε μια πληθώρα φορμών ευρείας χρήσεως, όπως απλό κείμενο, HTML, PostScript, PDF<sup>1</sup>. Η φόρμα PostScript (.ps), προϊόν της εταιρείας Adobe ήταν ήδη γνωστό στους κύκλους του T<sub>E</sub>X, χάρις στο πρόγραμμα d<sub>n</sub>ips που επέτρεπε την παραγωγή αρχείων έτοιμων για εκτύπωση. Τα αρχεία αυτά μπορούσαν να εμφανισθούν και στην οθόνη με τη βοήθεια των προγραμμάτων GhostScript και ghostview. Η φόρμα pdf, προϊόν κι αυτό της Adobe, ήταν όμως εκείνο που καθιερώθηκε και χρησιμοποιείται για παντός είδους κείμενα, κυρίως λόγω της δωρεάν διάθεσης του προγράμματος ανάγνωσης Acrobat Reader.

---

<sup>1</sup> Δεν αναφέρομαι σε εμπορικές φόρμες που παράγουν εφαρμογές όπως το MS Word κ.λπ.

Μέχρι τώρα για να πάρουμε στα χέρια μας ένα αρχείο pdf, χρειαζόμασταν ή το Adobe Distiller ή το ghostscript για τη μετατροπή του .ps σε .pdf αρχείο. Το Distiller είναι ένα εμπορικό πρόγραμμα με αρκετά τσουχτερή τιμή που μετατρέπει όλων των ειδών τα κείμενα σε μορφή pdf. Η άλλη εναλλακτική λύση, αυτή του ghostscript και του συνοδευτικού προγράμματος ps2pdf λειτουργεί πολύ καλά όταν έχουμε να μετατρέψουμε κείμενα γραμμένα με τα λεγόμενα base fonts, τις γραμματοσειρές PostScript δηλαδή, αυτές που διατίθενται με κάθε εκτυπωτή Postscript. Η μέθοδος αυτή, όμως, παρουσιάζει προβλήματα ιδίως στην εκτύπωση και παρουσίαση bitmap fonts (όπως η κλασική σειρά των CM του Knuth). Μια τρίτη, και σχετικά πρόσφατη λύση, παρέχεται μέσω του προγράμματος dvi2pdf με δυνατότητα μετατροπής του .dvi σε .pdf.

Η σχετικά νέα λύση του pdfTeX επιτρέπει την απ' ευθείας παραγωγή εγγράφων pdf και τον εμπλουτισμό τους με συνοδευτικά σχόλια, σημειώσεις, παραπομπές σε άλλα σημεία του ίδιου του κειμένου ή και σε εξωτερικές πηγές αναφορών, ακόμη. Συνδυάζει τις εντολές του TeX και της Adobe ώστε ο χρήστης να μπορεί να δημιουργεί κείμενα σε μορφή pdf, τα οποία κανείς μπορεί άμεσα να συμπεριλάβει σε κάποια ιστοσελίδα.

## 1. Καλώς ήρθες, pdfTeX!

Το pdfTeX είναι δημιούργημα του Hàn Thê Thành από την Τσεχία. Η τελευταία διαθέσιμη έκδοση είναι 0.14β. Διατίθεται από όλους τους κόμβους του CTAN, καθώς επίσης και τον κόμβο του συλλόγου χρηστών TeX της Τσεχίας, (cstug) στην διεύθυνση:

`ftp://ftp.cstug.cs/pub/tex/local/cstug/thanh.`

### 1.1. Εγκατάσταση

Το pdfTeX αποτελεί μέρος της διανομής mikTeX, του teTeX, του Web2C. Στους κόμβους του CTAN μπορείτε να βρείτε και τον πηγαίο κώδικα για πολλές άλλες διανομές και πλατφόρμες. Στο τελευταίο TeXLive CD, που μοιράστηκε στο σύλλογο, υπάρχει προεγκατεστημένο το pdfTeX αλλά και το pdfLaTeX.

Θα συνιστούσα, όμως, να προχωρήσετε στην επανεγκατάσταση και των δυο για να προσθέσετε το ελληνικό hyphenation file, ούτως ώστε να έχετε την δυνατότητα να χρησιμοποιείτε πλήρως το σύστημα και με ελληνικά. Οι σχετικές εντολές για να προχωρήσετε στην επανεγκατάσταση είναι:

```
pdftex -ini -fmt=pdfTeX plain \dump
pdftex -ini -fmt=pdfLaTeX latex.ltx
```

Τις εντολές αυτές πρέπει να τις εκτελέσετε στον κατάλογο (ελληνιστί directory) όπου είναι εγκατεστημένες ήδη οι παλαιότερες εκδόσεις των pdftex και pdfflatex. Στο τελευταίο TeX Live CD θα πρέπει να αναζητήσετε τον κατάλογο:/texmf/web2c. Αν αντιμετωπίσετε δυσκολίες, ο οδηγός χρήσης του pdfTeX περιέχει οδηγίες για να μπορέσετε να ολοκληρώσετε την εγκατάσταση.

## 2. Το πρώτο μας κείμενο σε pdfTeX

Θα ξεκινήσουμε με μια παρατήρηση: Εκτός από τις εντολές μορφοποίησης κειμένου που διαφέρουν μεταξύ των TeX, L<sup>A</sup>TeX, αλλά και eplain TeX, οι εντολές που αφορούν στη διαμόρφωση pdf δεν διαφέρουν άσχετα από την πλατφόρμα του TeX που χρησιμοποιούμε. Έτσι οι εντολές που θα παρουσιάσουμε στο άρθρο αυτό θα έχουν τα ίδια αποτελέσματα με το pdfTeX ή το pdfL<sup>A</sup>TeX.

### 2.1. Οι πρώτες εντολές

Ας γράψουμε ένα απλό αρχείο κειμένου (έστω και λίγες γραμμές κειμένου αρχούν προς το παρόν) και ας προσθέσουμε στην πρώτη γραμμή του αρχείου την εντολή:

```
\pdfoutput=1
```

Θα τροφοδοτήσουμε το αρχείο ως ένα κανονικό αρχείο .tex και presto, το πρώτο μας pdf αποτέλεσμα. Τώρα, μπορούμε να προσθέσουμε μιαν εντολή ακόμη και θα ξανα-επεξεργαστούμε το αρχείο:

```
\pdfcompresslevel=9
```

αμέσως μετά την πρώτη pdf εντολή του κειμένου. Το αποτέλεσμα, όπως φαίνεται στο Acrobat Reader δεν διαφέρει μετά την προσθήκη και της δεύτερης εντολής. Τότε γιατί να την προσθέσουμε στο κείμενό μας; Ας δούμε, όμως, τις δύο αυτές εντολές ξανά.

Η πρώτη εντολή με όρισμα «1» (ή μια άλλη θετική τιμή) σημαίνει πως θα δημιουργηθεί αρχείο pdf, ενώ η τιμή «0» θα έχει ως αποτέλεσμα τη δημιουργία αρχείου .dvi, δηλαδή το pdfTeX μπορεί και δημιουργεί και κανονικά αρχεία .dvi.

Η δεύτερη εντολή, που παίρνει τιμές από μηδέν (0) ως κι εννιά (9) μας δίνει διαφορετικούς βαθμούς συμπίεσης όταν χρησιμοποιούμε γραφικά στο αρχείο(0 σημαίνει καθόλου συμπίεση, 1 σημαίνει γρήγορη συμπίεση, 9 μέγιστη και οι υπόλοιπες τιμές ενδιάμεσα επίπεδα συμπίεσης).

## 2.2. Ο τίτλος κι ο συγγραφέας

Μέχρι στιγμής δεν έχουμε δει κάτι το διαφορετικό ή ξεχωριστό στην παραγωγή αρχείων pdf. Ούτε κι έχουμε εισχωρήσει στα άδυτα του pdf. Ένα πρώτο βήμα προς αυτήν την κατεύθυνση είναι οι ακόλουθες εντολές που αφορούν στον συγγραφέα και τον τίτλο του αρχείου που επεξεργαζόμαστε. Η σύνθετη εντολή:

```
\pdfinfo
{
/Author      (Ioannis Dimakos)
/Creator     (pdfTeX / babel / web2c)
/Producer    (pdfTeX)
/CreationDate (D:19990830210000)
/ModDate     (D:19990831234000)
/Title       (eytypoarticle.tex)
/Subject     (Introductory Note on pdfTeX)
/Keywords    (pdf, tex, eytypon)
}
```

θα μας δώσει πληροφορίες σχετικά με το αρχείο που επεξεργαζόμαστε. Οι πληροφορίες αυτές εμφανίζονται στο Acrobat Reader με CTRL-D ή από το μενού File, και την επιλογή Document Info και General. Αν και η εντολή θυμίζει λίγο τις εντολές `\author`, `\date`, και `\title` του L<sup>A</sup>T<sub>E</sub>X που χρησιμοποιούμε για την παρουσίαση τέτοιων στοιχείων, διαφέρει από αυτές σημαντικά. Πρώτα απ' όλα, δεν μπορούμε να βάλουμε δικές μας προτιμήσεις, π.χ., `/address`. Δεύτερον, η εντολή μπορεί να μπει τόσο στον πρόλογο, όσο και στον επίλογο του αρχείου. Τρίτον, όπως φαίνεται και από το παράδειγμα, τα ορίσματα `/author`, `/title` κλπ. συνοδεύονται από «/» κι όχι «\». Ωστόσο, δεν είναι όλα τα στοιχεία απαραίτητα για την εκτέλεση της εντολής. Μπορούμε να παραλήψουμε όσα δεν μας χρειάζονται, και το pdfTeX θα ορίσει προκαθορισμένες τιμές.

## 2.3. Βασικές εντολές μορφοποίησης κειμένου

Εκτός από τις συνηθισμένες εντολές μορφοποίησης του κειμένου που προσφέρονται από το T<sub>E</sub>X ή την πλατφόρμα της δικής σας προτίμησης, το pdfTeX παρέχει πρόσθετες δυνατότητες μορφοποίησης του τελικού προϊόντος. Οι εντολές:

```
\pdfpagewidth=xx
\pdfpageheight=yy
```



όπου τα `xx` και `yy` είναι κάποια μήκη, π.χ., 3 cm, επιτρέπουν την μορφοποίηση της εμφάνισης μιας σελίδας pdf (το πλάτος και το ύψος της σελίδας, αντίστοιχα) όπως και οι γνωστές μας TeX εντολές:

```
\textwidth \textheight
```

Αν και το pdfTeX διαθέτει ένα αρχείο συστήματος με βασικές τιμές (θα το βρείτε στον κατάλογο `config` της δικής σας εγκατάστασης στο αρχείο `pdftex.cfg`), ωστόσο οι τιμές αυτές μπορούν να αλλάξουν κατά βούληση με τη βοήθεια των προαναφερθέντων εντολών pdf. Παραδείγματα τέτοιων εντολών ακολουθούν:

```
\pdfpagewidth=190mm  
\pdfpageheight=277mm
```

Δυο ακόμη εντολές που επηρεάζουν την εμφάνιση της σελίδας του αρχείου pdf είναι και οι εξής:

```
\pdfpagesattr={/MediaBox xx yy zz tt}  
\pdfpageattr={/MediaBox xx yy zz tt}
```

Η βασική διαφορά των δυο εντολών είναι πως ενώ η πρώτη έχει γενικό χαρακτήρα και μπορεί να χρησιμοποιηθεί για την διαμόρφωση ολόκληρου του κειμένου, η δεύτερη χρησιμοποιείται για να αναστείλει προσωρινά την δράση της πρώτης, και το πεδίο δράσης της αφορά στην συγκεκριμένη σελίδα που τυπώνεται εκείνη την στιγμή.

Τα `xx yy` καθορίζουν τις συντεταγμένες του κάτω αριστερά άκρου και τα `zz tt` τις αντίστοιχες του άνω δεξιά άκρου της σελίδας. Αντί του ορίσματος `/MediaBox xx yy zz tt` που αφορά στο φυσικό μέγεθος της σελίδας, μπορούμε να ορίσουμε `/CropBox xx yy zz tt` για να «σημαδέψουμε» μόνον το τμήμα εκείνο της σελίδας που θα τυπωθεί. Ένα τρίτο εναλλακτικό όρισμα είναι και το `/Rotate 90` που δίνει διάφορα γραφικά εφφέ προσανατολισμού. Αντί για 90 μοίρες περιστροφή μπορούμε να ορίσουμε οποιοδήποτε πολλαπλάσιο (ακέραιο ή μη) του 90 για περιστροφή κατά τη φορά των δεικτών του ρολογιού.

## 2.4. Εικόνες και γραφικά

Μια εικόνα αξίζει όσο χίλιες λέξεις, σύμφωνα με το γνωστό Κινέζικο γνωμικό, και το pdfTeX μας δίνει την δυνατότητα να εκφράσουμε τις λέξεις (και τις εικόνες αυτές) ποικιλοτρόπως. Η εντολή:

```
\pdfimage height xx width yy depth zz {filename.ext}
```

επιτρέπει την εισαγωγή εικόνας και την (προαιρετική) διαμόρφωση των διαστάσεων της στην σελίδα. Σε αντίθεση με το  $\TeX$  και το  $\LaTeX$ , το  $\text{pdf}\TeX$  μπορεί να χειρισθεί τέσσερα (4) διαφορετικά είδη αρχείων εικόνας:

- JPEG (.jpg),
- PNG (.png),
- TIFF (.tif),
- PDF (.pdf).

Η προκαθορισμένη επιλογή είναι τα αρχεία της μορφής JPEG. Το  $\text{pdf}\TeX$  δεν μπορεί να επεξεργασθεί εικόνες και γραφικά σε αρχεία PostScript (.ps) ή Encapsulated PostScript (.eps). Για τις περιπτώσεις αυτές θα πρέπει να γίνει μια μικρή προεργασία και μετατροπή τους σε ένα από τα παραπάνω είδη αρχείων.

Με τα ορίσματα `height xx width yy depth zz` μπορούμε να καθορίσουμε τις ακριβείς διαστάσεις της εικόνας που θα εισάγουμε. Σε περίπτωση που δεν ορίσουμε κάποια διάσταση η εικόνα θα εισαχθεί στο φυσικό μέγεθός της:

```
\pdfimage {image.jpg} % default dimensions
\pdfimage height 2cm width 2cm {image.png} % image at 2x2 cm
```

Αν, όμως, ορίσουμε μόνο μια διάσταση (π.χ., ύψος), τότε οι άλλες διαστάσεις θα υπολογισθούν κατά τέτοιο τρόπο ώστε ο λόγος `width:(height+depth)` να είναι ο ίδιος όπως αυτός των φυσιολογικών διαστάσεων της εικόνας.

Ένας έξυπνος τρόπος για την επίτευξη διαφόρων οπτικών εφέ είναι η εισαγωγή της εικόνας σε ένα  $\TeX$  «κουτί» και μετά να αυξομειώσουμε ανάλογα τις διαστάσεις του «κουτιού» αυτού. Το ακόλουθο παράδειγμα είναι διαφωτιστικό:

```
\setbox0=\hbox{%
  \pdfimage{image.jpg}
}
\pdfimage width x\wd0 height y\ht0 {image.png}
```

Οι μεταβλητές `\wd0` και `\ht0` δεσμεύουν για μας τις διαστάσεις της εικόνας. Στη συνέχεια, μπορούμε να τροποποιήσουμε το μέγεθος της εικόνας πολλαπλασιάζοντας τις διαστάσεις αυτές με κάποιους συντελεστές  $x, y$ .

Μια ακόμη εντολή καθορίζει την ανάλυση (resolution) της εικόνας σε περίπτωση που θέλουμε ανάλυση διαφορετική από την προκαθορισμένη ανάλυση των 72 dpi:

```
\pdfimageresolution=xx
```

Πειραματιστείτε κι εσείς με διαφορετικές τιμές και ορίσματα στις εντολές που παρουσιάστηκαν στο άρθρο αυτό και δείτε τα διάφορα εφέ που δημιουργείτε.

## 2.5. Από σημείο σε σημείο

Ερχόμαστε τώρα στο πιο δυναμικό ίσως σημείο του pdfTeX που φανερώνει την πρακτικότητα και την ευκολία που παρέχει τόσο στον συγγραφέα όσο και στον αναγνώστη. Με την εντολή:

```
\pdfcatalog {/PageMode /UseThumbs
              /URI (http://obelix.ee.duth.gr/eft/pdfarticle.pdf)
              } openaction goto page 2 {/fit}
```

καθορίζουμε την πλοήγηση του κειμένου μέσα από το Acrobat Reader. Ας δούμε την εντολή αναλυτικότερα. Η εντολή αποτελείται από δυο μέρη. Στις αγκύλες, το όρισμα /URI προσδιορίζει τον δικτυακό τόπο του αρχείου που επεξεργαζόμαστε. Το όρισμα /PageMode προσδιορίζει συγκεκριμένες λειτουργίες του Acrobat Reader. Όταν ανοίγουμε το Acrobat Reader, στο αριστερό τμήμα της οθόνης, υπάρχει κάποιος δεσμευμένος χώρος για την πλοήγηση στο κείμενο είτε σελίδα - σελίδα (thumbs mode είτε με τη χρήση σελιδοδεικτών (outline mode)). Οι πιθανές τιμές του ορίσματος εμφανίζονται στον πίνακα 1 που ακολουθεί:

Πίνακας 1: Είδη πλοήγησης στο Acrobat Reader

Εντολή	Ενέργεια
/UseThumbs	σελίδα-σελίδα
/UseOutlines	σελιδοδείκτες
/FullScreen	χωρίς μενού επιλογών
/UseNone	τίποτε από τα παραπάνω

Έξω από τις αγκύλες, το όρισμα `openaction goto page 2 /fit` καθορίζει πως η πρώτη σελίδα που θα εμφανιστεί θα είναι η σελίδα νο. 2 και η σελίδα θα καταλαμβάνει όλο το ευρύς της οθόνης (στην πραγματικότητα, η σελίδα καταλαμβάνει το εύρος του «παραθύρου» στο οποίο εμφανίζεται το κείμενο pdf κι όχι ολόκληρη την οθόνη. Περισσότερα στοιχεία για το όρισμα `goto page` και `/fit` παρακάτω στην σελίδα 9.

## 2.6. Σχόλια επί σχολίων

Είπαμε στην εισαγωγή του άρθρου αυτού πως το pdfTeX επιτρέπει την εισαγωγή σχολίων και σημειώσεων που διευκολύνουν τον συγγραφέα αλλά και τους αναγνώστες. Το pdfTeX μετατρέπει έτσι ένα «στατικό» κείμενο σε ένα δυναμικό σύνολο που μπορεί να περιέχει ηχητικά αποσπάσματα, video-clips, online σημειώσεις αλλά και παραπομπές σε άλλα τμήματα του κειμένου ή και σε διαφορετικά αρχεία. Οι παραπομπές παρουσιάζονται στην επόμενη ενότητα του άρθρου (σελ. 9).

Η εντολή `\pdfannot` θα δημιουργήσει στο σημείο ορισμού της μια online σημείωση (που θυμίζει αρκετά τα γνωστά αυτοκόλλητα post-it). Πιο συγκεκριμένα:

```
\pdfannot
width xx
height yy
depth zz
{/subtype /text
/open /true
/contents (A simple pop-up note)
}
```

η παραπάνω εντολή θα δημιουργήσει ένα πλαίσιο διαστάσεων που καθορίζονται από τα γνωστά πια ορίσματα `width`, `height`, `depth` στο οποίο θα εμφανίζεται το κείμενο που εμείς θέλουμε. Τα ορίσματα εντός της αγκύλης είναι προφανή. Το σχόλιο είναι τύπου κειμένου (`text`), γιατί μπορεί να είναι και τύπου `movie` ή `sound`<sup>2</sup>), ενώ όταν ανοίγει το αρχείο στο Acrobat Reader, η σημείωση είναι κι αυτή ανοικτή (εμφανίζεται το περιεχόμενό της). Αν αλλάξουμε την τιμή του `/open` από `/true` σε `/false`, τότε η σημείωση θα εμφανιστεί «κλειστή» και θα θυμίζει πράγματι τα γνωστά αυτοκόλλητα. Μπορούμε έτσι να πλουτίσουμε το κείμενό μας με οδηγίες, επεξηγήσεις (που πιθανόν να μην χωρούν αλλού μέσα στο κείμενο), τύπους, και φόρμουλες.

## 2.7. Παραπομπές και υπερ-δεσμοί

Για την δημιουργία παραπομπών και υπερ-δεσμών (ελληνιστί `hyperlinks`) μεταξύ σημείων του κειμένου που επεξεργαζόμαστε χρειαζόμαστε **τρεις** (3) εντολές.

Η **πρώτη** εντολή «σημαδεύει» τον προορισμό της παραπομπής, το σημείο εκείνο του κειμένου στο οποίο παραπέμπουμε τον αναγνώστη. Η σύνταξή της έχει ως εξής:

<sup>2</sup> Στο εγχειρίδιο PDF Reference Manual της Adobe περιγράφονται λεπτομερώς οι δυνατότητες για παραπομπές πολυμέσων

```
\pdfdest num x|name y fit|fith|...
```

Η εντολή ορίζει μ' έναν αριθμό (ή και με όνομα) τον σημείο στο οποίο παραπέμπουμε. Θυμίζει λίγο την εντολή `\label` αν και στην περίπτωση που εξετάζουμε μπορούμε να επιλέξουμε μεταξύ αριθμών και ονομάτων. Το δεύτερο μέρος της εντολής αφορά στον τρόπο παρουσίασης της παραπομπής στο Acrobat Reader. Οι επιλογές είναι πολλές και παρουσιάζονται στον πίνακα 2.

Πίνακας 2: Πώς εμφανίζονται οι παραπομές στο Acrobat Reader

Εντολή	Εμφάνιση
<code>fit</code>	Ολόκληρης της σελίδας στην οθόνη
<code>fith</code>	Του εύρους της σελίδας στην οθόνη
<code>fitv</code>	Του ύψους της σελίδας στην οθόνη
<code>fitb</code>	Με βάση τις διαστάσεις του BoundingBox της σελίδας
<code>fitbh</code>	Με βάση το εύρος του BoundingBox
<code>fitbv</code>	Με βάση το ύψος του BoundingBox
<code>xyz</code>	Καθορίζει το βαθμό εστίασης zoom της σελίδας

Μια παρατήρηση στις παραπάνω εντολές. Η εντολή `fith` ορίζει το εύρος της σελίδας (κι όχι το ύψος της, όπως ίσως να περίμενε κανείς λόγω του `h` (`height`)). Το ίδιο ισχύει και για την εντολή `fitbh`.

Η **δεύτερη** εντολή «καθορίζει» την αρχή της παραπομπής (το σημείο του κειμένου από το οποίο παραπέμπουμε προς το σημείο που ορίσαμε με την εντολή `\pdfdest`. Η εντολή συντάσσεται ως εξής:

```
\pdfannotlink width xx height yy depth zz
      attr {...} action
```

Τα ορίσματα `width`, `height`, και `depth` είναι γνωστά. Το `attr{...}` καθορίζει τα `attributes` (χαρακτηριστικά) της παραπομπής, το χρώμα και το πάχος του πλαισίου γύρω από το κείμενο της παραπομπής. Η παραπομπή μπορεί να απλώνεται σε αρκετές γραμμές κειμένου οι οποίες θα διακρίνονται για το διαφορετικό τους χρώμα και τις διαχωριστικές γραμμές του πλαισίου. Πιο συγκεκριμένα:

```
attr {/C [x y z] /Border [p q r]}
attr {/C [0.9 0 0] /Border [0 0 2]}
```

ορίζουμε πως η παραπομπή θα είναι γραμμένη με βαθύ κόκκινο χρώμα και θα έχει πάχος 2 pt. Αν δεν ενδιαφέρεστε να ορίσετε εσείς κάποιο χρώμα ή το πλαίσιο

που θα περιβάλλει την παραπομπή, μπορείτε να προσπεράσετε το σημείο αυτό. Το pdfTeX θα ορίσει από μόνο του προκαθορισμένες τιμές. Τι μπορούμε να κάνουμε με την παραπομπή τώρα; Οι επιλογές για το όρισμα `action` της εντολής είναι πολλές:

- `page n` μεταφορά στη σελίδα  $n$  του κειμένου.
- `goto num n|name y` μεταφορά στο σημείο που ορίσαμε με την εντολή `\pdfdest`.
- `goto file` μεταφορά σε κάποιο άλλο αρχείο. Μπορεί να συνδυαστεί και με την εντολή `page n` και να μας οδηγήσει απ' ευθείας σε κάποια συγκεκριμένη σελίδα του νέου αρχείου.
- `thread num n|name y` μεταφορά στο νήμα υπ' αριθ.  $n$  ή με ονομασία  $y$ . Για νήματα θα μιλήσουμε παρακάτω.
- `user action` κάποια ελεγχόμενη από τον χρήστη κίνηση. Συνήθως χρησιμοποιείται για την ενεργοποίηση ενός υπερδεσμού στο διαδίκτυο, π.χ.:

```
/Subtype /URI /URI (http://obelix.ee.duth.gr/eft/).
```

Το διπλό `/URI` δεν είναι τυπογραφικό λάθος ή οπτική απάτη. Το πρώτο `/URI` συνοδεύει το `/Subtype` (ορίζει, δηλαδή, το είδος της ενέργειας του χρήστη), και το δεύτερο προσδιορίζει την δικτυακή διεύθυνση που θέλουμε.

Η **τρίτη** εντολή για τον ορισμό μιας παραπομπής «κλείνει» το κείμενο της παραπομπής. Στη συνέχεια θα ακολουθήσει φυσιολογικά το κείμενο. Η σύνταξή της:

```
\pdfendlink
```

χωρίς να υπάρχουν πρόσθετα ορίσματα ή άλλα χαρακτηριστικά. Ας δούμε όλες τις εντολές μαζί σε ένα απλό παράδειγμα.

```
The purpose of this example is to show the use
of the \pdfannotlink command.
\pdfannotlink attr{/C [0.7 0 0] /Border [0 0 2]} goto num 1
I will take you from here
\pdfendlink
...
...
more lines of text
```

```

...
...
\pdfdest num 1
And I will get you here. This is the destination
of the \pdfannotlink command.

```

### 3. Περισσότερες πληροφορίες

Για τους χρήστες που θα ήθελαν να μάθουν περισσότερα για τις δυνατότητες του pdfTeX, ο κόμβος <http://www.tug.org/applications/pdftex/> περιέχει αρκετές κατατοπιστικές πηγές πληροφοριών. Εκεί θα βρείτε το FAQ το εισαγωγικό εγχειρίδιο του pdfTeX, αλλά και μια σειρά εγγράφων που έχουν δημιουργηθεί με το pdfTeX και παρουσιάζουν τις δυνατότητες του προγράμματος.

Πολύ καλή πηγή πληροφοριών είναι και ο πηγαίος κώδικας του αρχείου `example.tex` που υπάρχει σε κάθε διανομή του pdfTeX. Αξίζει τον κόπο να τον μελετήσετε και να δοκιμάσετε παραλλαγές των παραδειγμάτων του.

Επίσης, για περισσότερες πληροφορίες σχετικά με το pdf format, αρκετά εγχειρίδια (φυσικά, σε pdf format) διατίθενται από τον δικτυακό τόπο της Adobe:

**Adobe Glossary:** Ένα κατατοπιστικό και συνάμα σύντομο λεξικό από τον χώρο του pdf.

**Portable Document Format Reference Manual:** Το οριστικό εγχειρίδιο της Adobe, στο οποίο περιγράφεται λεπτομερώς το pdf. Έκδοση 1.3 του Μαρτίου 1999.

Τέλος, στον δικτυακό τόπο του συλλόγου θα βρείτε το αρχείο `expdf.tex` και `expdf.pdf` με τα παραδείγματα των εντολών που παρουσιάσαμε στο άρθρο αυτό.

### 4. Επίλογος

Τελικά, χρειάζεται κανείς το pdfTeX; Πιστεύω πως ναι. Αν και είναι ακόμη σε beta έκδοση, το pdfTeX βοηθά να ξεπεραστούν προβλήματα που συχνά-πυκνά εμφανίζονται στον παραδοσιακό κύκλο επεξεργασίας του TeX. Αναφέρομαι, φυσικά, στο πρόβλημα με τα λεγόμενα base fonts που δεν μετατρέπονται εύκολα με το ghostscript (μέσω του προγράμματος `ps2pdf`).

Ένας άλλος λόγος είναι η ταχύτητα του κύκλου επεξεργασίας. Αντί για 3 στάδια επεξεργασίας (`.tex` → `.dvi` → `.ps` → `.pdf`) ή έστω 2 στάδια αν υπολογίσουμε το νέο πρόγραμμα-οδηγό `dvipdfm`, με το pdfTeX αρκεί ένα στάδιο επεξεργασίας. Έτσι μειώνεται ο χρόνος του κύκλου συγγραφής - διόρθωσης ενός

κειμένου. Προσοχή όμως γιατί αν έχετε το `.pdf` αρχείο ανοικτό στο Acrobat Reader και προσπαθήσετε να επεξεργαστείτε εκ νέου τον κώδικα που το δημιουργήσε, θα λάβετε ένα προειδοποιητικό μήνυμα από το σύστημα πως δεν μπορεί να γράψει στο αρχείο (γιατί αυτό φυσικά είναι ανοικτό και δεσμευμένο από μίαν άλλη εφαρμογή).

Κυρίως όμως, το `pdfTeX` επιτρέπει την δημιουργία κειμένων που δεν έχουν μόνον φυσική ροή, αλλά στα οποία ο συγγραφέας καθορίζει (μέσω των σχολίων, σημειώσεων, και υπερ-δεσμών που εισάγει) τη νοηματική ροή του κειμένου. Παράλληλα, το `pdfTeX` (και για να είμαστε πιο ακριβείς, το `pdfLaTeX`) επιτρέπει την δημιουργία ολοκληρωμένων επαγγελματικών παρουσιάσεων. Αρκετά πακέτα, όπως το `graphicx`, το `hyperref`, το `color`, το `pdfscreen` δίνουν την ευχέρεια στον χρήστη να προετοιμάσει μέσω του `LaTeX` (και του `pdfLaTeX`) κείμενα και παρουσιάσεις χωρίς να καταφεύγει σε ακριβά, εμπορικά προγράμματα (τρέμε, PowerPoint!!!). Όμως, για τα προγράμματα αυτά θα μιλήσουμε στο επόμενο τεύχος του περιοδικού.



# How to Typeset an Archaeological Book with L<sup>A</sup>T<sub>E</sub>X

---

Jed Parsons

*Department of Classics  
University of California, Berkeley*

*e-mail: jed@socrates.berkeley.edu*

## 1. Introduction

The creation of the Camera-Ready Copy (CRC) for a book on ancient Greek technology and archaeology was prepared with the document preparation system L<sup>A</sup>T<sub>E</sub>X. The book, entitled *Hysplex*, by Panos Valavanis and translated into English by Stephen Miller, will be available from the University of California Press in the fall of 1999 (ISBN 0-520-09829-3). This article is intended to demonstrate that the freely-available document preparation systems T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, while primarily known for their use in mathematical and scientific publication, are suitable tools for Classicists and other humanists for producing well-organized, maintainable, and beautiful texts.

## 2. Getting Started with T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X

The typesetting program T<sub>E</sub>X, and the collection of sophisticated T<sub>E</sub>X macros called L<sup>A</sup>T<sub>E</sub>X, were originally designed with mathematicians and scientists in mind. From the start, they were powerful renderers of complex mathematical equations, but their capacity for rendering languages with non-Latin alphabets was limited. Those limitations have been tremendously reduced by the development of numerous ancillary packages which facilitate the creation of multilingual, structured documents. One can create a document that switches between languages (following hyphenation rules for each language separately), uses cross-references whose pagination is automatically generated, creates custom-formatted bibliographies from bibliographic databases, structures the presentation of the document in a controllable and logical way, and much more.

The functionality of  $\text{\TeX}$ ,  $\text{\LaTeX}$ , and the accompanying packages written for them, makes them a logical choice for humanists now as well as scientists. For scholars of ancient languages, there are even macro packages for managing manuscript collations and generating critical editions like those of the Oxford Classical Texts or Teubner, with automatic line numbering, line referencing in the critical apparatus(es), etc. Finally,  $\text{\TeX}$  and  $\text{\LaTeX}$  are freely available on the Internet, or on CDROM for a small cost. The programs can be run on UNIX (and Linux), Macintosh, Windows, and DOS, to name a few.

$\text{\TeX}$  and  $\text{\LaTeX}$  (which I shall refer to collectively as  $\text{\LaTeX}$  when the distinction is unimportant) are not easy to use, it is true. The user does not edit a document as he or she would when using a word processor. Rather, one writes markup instructions in a text-only document. This document is then processed by an interpreter. The resulting output can be viewed or converted to another file format, such as PostScript or PDF. This requires most writers to adjust the way they think about composition. It is not my purpose to introduce the reader to the fundamentals of  $\text{\LaTeX}$ . There are many good books and online tutorials for this. (See the references on page 31 below.) If you have had no exposure to  $\text{\LaTeX}$ , browse this document, see if illustrates the sorts of things you want to do, and if it does, consider looking into some introductory material (see below). If you have some previous experience, I hope you will find some value in the more technical aspects of this text.

### 3. Some Reasons to Use $\text{\LaTeX}$

Why go to the trouble? Above all, for a beautiful book. Word processors, as they are available today, are little more than glorified typewriters. They have the ability to generate footnotes and switch fonts, yes, but they have no taste. Word processors do not perform letterfitting operations and kerning well. They do not try to balance the gray value (the ratio of black ink to white paper) of a whole paragraph, shuffling line breaks and minimizing hyphenation on the way. They do not automatically implement the ligatures intended by the designer for the font used (e.g., fl, ffi, and perhaps ij, sc, etc.). They do not speak the language of typographers, and cannot reliably be called upon to *accurately* set text in points and picas (as any printing house will specify). They do not manage cross references well (if at all) to text, footnotes, headings, figures, etc. They do not generally interface well with bibliographic databases and the like.  $\text{\LaTeX}$  does all these things, and does them well. In sum, word processors have forgotten (thanks probably both to the intervention of the typewriter and a market-driven disinterest in relearning) the state of the art of printing as it was in Gutenberg's day, and they have forgotten the skills developed by

centuries of scribes and calligraphers. The authors of T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X have done their homework: they have consulted with type designers, calligraphers, and typographers alike. The result speaks for itself. Yes, it is harder to use L<sup>A</sup>T<sub>E</sub>X than a word processor, but if you care about beauty, there is every reason to make the effort.

## 4. Platforms

I run L<sup>A</sup>T<sub>E</sub>X on the Linux operating system (which, like L<sup>A</sup>T<sub>E</sub>X, is freely available with source code). If you have a commercial operating system, like MacOS or some variety of Windows, you are also in luck. The programs comprised by L<sup>A</sup>T<sub>E</sub>X have been ported to many platforms. Indeed, L<sup>A</sup>T<sub>E</sub>X documents themselves are inherently portable, since the document encoding is the same for every system. If you have ever received an email attachment formatted by some word processor you don't have, you will appreciate this.

The following sections briefly introduce some basic components of a L<sup>A</sup>T<sub>E</sub>X system, with some examples of their use or usefulness.

## 5. Software Components of a L<sup>A</sup>T<sub>E</sub>X System

L<sup>A</sup>T<sub>E</sub>X resides on your system in a directory or folder hierarchy just as do any other applications you have. There are some key components to a basic L<sup>A</sup>T<sub>E</sub>X system, which I will briefly introduce here. As a Linux user, I am most qualified to talk about the UNIX utilities which I am familiar with. Similar utilities exist for other platforms. They will probably have slightly different names, but they will function in essentially the same way.

### 5.1. `tex` and `latex`

These are the executable programs that interpret your source document and the markup instructions in it. They generate as output a DVI (“device independent”) file. This file contains raw instructions for typesetting the page. A further interpreter is necessary to convert the data in this file into a printed output format, such as PostScript or PDF. DVI files are portable between people who have the same font libraries. These files contain specifications such as page size and the location and metrics of every glyph to appear on the page. Those glyphs themselves must be pulled from the font library and set by the final interpreter. (On my system, this is a program called `dvips` which turns DVI files into PostScript.)

$\LaTeX$  is easily invoked at the shell prompt, taking your input file name as an argument. Input files can instruct other files to be loaded and merged. This makes it easy to keep large books divided into manageable chunks. For *Hysplex*, I created the following files containing parts of the book:

```
half-title.tex
cip.tex
dedication.tex
abbr+bib.tex
prologue.tex
introduction.tex
part1.tex
part2.tex
conclusion.tex
appendix.tex
glossary.tex
summary.tex
```

(There were also other parts of the document that were automatically generated along the way, like the table of contents, list of figures, and index.) All of these files, and my own macro sets, were loaded and arranged in the single file `hx_book.ltx`, so all I had to do to generate the book was type, at my shell prompt,

```
latex hx_book.ltx
```

## 5.2. Style Files and Macro Packages

$\TeX$  is extensible by anyone who writes macros for it.  $\LaTeX$  is really a (huge) set of  $\TeX$  macros. Macro packages which provide logical formatting commands may be called Style Files. I regularly use style files that enable me to set polytonic Greek and generate critical editions of texts for teaching. I also often need to format complex tables, include graphic images, construct linguistic trees and diagrams, faithfully represent a fragment of papyrus, etc. There are style files freely available to do all these things.

## 5.3. xdvi

After  $\LaTeX$  is finished processing your input file (and assuming there are no errors for you to deal with), you receive a message like:

(see the transcript file for additional information)  
Output written on hx\_book.dvi (203 pages, 624220 bytes).  
Transcript written on hx\_book.log.

You probably want to see what your document looks like without having to print it. I do this with a program called `xdvi`, which is the UNIX X-Window system DVI file viewer. There are DVI viewers for other platforms as well. By invoking

```
xdvi hx_book.dvi &
```

I can see, but not edit, the generated output. If there is something to fix, I must edit the L<sup>A</sup>T<sub>E</sub>X source and recompile it. (You don't have to restart `xdvi` or reload the file; it's sensitive to the change.)

#### 5.4. dvips

To generate printable output, a utility like the UNIX `dvips`, which outputs PostScript code, is useful. To print on my (slightly broken) printer at home, which can only achieve a resolution of 300dpi, I type

```
dvips -D 300 hx_book.dvi
```

which spools a 300dpi PostScript file to the printer. If I want to output a copy at higher resolution, I might type

```
dvips -D 600 -o hx_book.ps hx_book.dvi
```

which instructs `dvips` to create the PostScript file `hx_book.dvi` at 600dpi. I do this regularly with all my documents. To print, I upload the PostScript file to the server at my university (you can use `ftp` for this, or `Fetch` if you are a Mac user; I use `scp`). In my office, I download the file to a Mac, and print it using the Apple LaserWriter Utility on one of my department's spiffy new printers. This way I don't have to mess around with floppy disks. If you have PostScript versions of each font you use, you can embed them in your document. This makes it scalable to any resolution. On my Linux-based L<sup>A</sup>T<sub>E</sub>X distribution (t<sub>e</sub>L<sub>A</sub>T<sub>E</sub>X), I type

```
dvips -Pcmz -o outfile.ps infile.dvi
```

In our case, we could not do this, since the Greek fonts we used to not yet exist in PostScript format.

### 5.5. Adobe Distiller and PDF $\TeX$

Another alternative is to generate a PDF file from your DVI file. PDF is Adobe's Portable Document Format, which is rapidly becoming popular for online publishing. The program required to read PDF files, Acrobat Reader, is freely available from Adobe (<http://www.adobe.com>). Adobe's program to encode PDF files is not free, however, but can be purchased for something like US \$50 for academic use. A nice alternative for  $\TeX$  users is to get the free program PDF $\TeX$ ; for  $\LaTeX$  there is PDF $\LaTeX$ . These programs work very well and make web publishing easy. For more on PDF with  $\LaTeX$ , see <http://www.math.uakron.edu/~dpstory/latx2pdf.html>.

## 6. Logical Document Structure

$\LaTeX$  enables you to structure your document in a logical way. That is, you may make abstract divisions and entities, such as “section” and “figure” which are invoked by simple markup commands, and which will all be rendered in the final output according to the rules you specify. Thus, your document might have some encoding like:

```
\section{Observations on the Starting Lines of Various Stadia}
```

```
From the data that we have studied to this point and from the
conclusions that we have reached concerning the shape and the
operation of the Classical \emph{hysplex}, we may now proceed
to observations about some characteristics in the starting
lines in other stadia that seem to be relevant to the
\emph{hysplex} mechanism.
```

```
\subsection{Mainland Greek Stadia}
```

```
\subsubsection{Olympia}
```

```
Two constructions were added at the ends of the last phase
of the western \emph{balbis}. These project and make a right
angle with the \emph{balbis}, and they have created problems
for the excavators of the most famous stadium of the ancient
world.\grfn{E. Curtius, F. Adler, \emph{et al}., \emph{Die
Ausgrabungen zu Olympia}~V (1881) 25, pl.~35. Cf.~Romano
fig.~100.} At one end ...
```

which, when it is parsed by LaTeX, produces the following output (here reduced to 80%).

## Observations on the Starting Lines of Various Stadia

From the data that we have studied to this point and from the conclusions that we have reached concerning the shape and the operation of the Classical *hysplex*, we may now proceed to observations about some characteristics in the starting lines in other stadia that seem to be relevant to the *hysplex* mechanism.

### Mainland Greek Stadia

#### OLYMPIA

Two constructions were added at the ends of the last phase of the western *balbis*. These project and make a right angle with the *balbis*, and they have created problems for the excavators of the most famous stadium of the ancient world.<sup>176</sup> At one end

If you're curious about the markup commands, `\section`, `\subsection`, and so forth, are macros that have been defined elsewhere (in my collection of macros associated with the book). When invoked, they format the text as you see, and also generate an entry for the table of contents, keyed to the correct page number. (The table of contents is also generated by macros.) This makes it easy to adjust layout. For example, if the subsection heading needs to be rendered in a larger point size, I can easily implement this change throughout the document by simply changing the declaration of the `\subsection` command slightly. The command `\emph` places the text in braces in italics. It has the added feature of adding a small space to correct for the slant of the italics before following words and tall punctuation marks, but not before periods and commas. You may be see this in the document you are reading now. `\grfn` is my footnote command that is sensitive to the use of Greek. You can't see the footnote in this example, but you can see that the command left a proper footnote mark above the text. Within the braces after the footnote command, you will see the tilde character; that's L<sup>A</sup>T<sub>E</sub>X-speak for a non-breakable space. I put these in the text wherever I want to ensure that there is no line break. I would normally put these between initials in peoples' names, too, but here the names are right at the beginning of the note, so I didn't bother.

## 6.1. Frontmatter and Backmatter

The University of California Press has its own format that we had to adhere to. For example, front matter was to be numbered with lowercase Roman numerals; the main text and backmatter with Arabic. Furthermore, certain parts of the text were to begin on a new right page; others simply on a new page. A few LaTeX commands made it simple to put this in order:

```

\pagestyle{empty}
\pagenumbering{roman}
\input dedication                % Dedication, no headers
        \cleardoublepage

\pagestyle{fancy}

\tableofcontents                % TOC
        \cleardoublepage

                                % LOF
\listof{hxfloat}{List of Figures} % Our own float style
                                % (see millermacs.tex)
        \cleardoublepage

\input prologue
        \clearpage

\input abbr+bib
        \clearpage

```

I've taken out a few formatting trivialities, but that's the bulk of the first part of the main input file `hx_book.ltx`. Simple commands specify the mode of page numeration. The `\input` commands merge in the other documents that comprise the whole book. The table of contents and list of figures are generated by a single command. LaTeX works out all the pagination itself. Finally, the simple commands `\clearpage` and `\cleardoublepage` ensure either a simple page break, or that the next section will start on a new right page. Backmatter is formatted in a similar way.



### 6.2. Table of Contents

When L<sup>A</sup>T<sub>E</sub>X comes across sectioning commands, it inserts the section name in a file it generates for you, wrapped in the proper formatting commands, and the proper page number. In our case, the command `\tableofcontents` generated a page starting like this (again, reduced to 80%):

#### Contents

<b>List of Figures</b>	<b>ix</b>
<b>Prologue</b>	<b>xv</b>
<b>Abbreviations and Bibliography</b>	<b>xvii</b>
<b>Introduction</b>	<b>1</b>
<b>Documents and Documentation of the Mechanism</b>	<b>3</b>
Written Sources and Terminology . . . . .	3
Architectural Features of Stadia . . . . .	10
The New Panathenaic Amphora . . . . .	20
The Parts of the <i>Hysplex</i> and Its Means of Operation . . . . .	31
Reconstruction of the Form of the <i>Hysplex</i> . . . . .	35
The Operation of the Mechanism . . . . .	44

### 6.3. List of Figures

L<sup>A</sup>T<sub>E</sub>X also possesses a `\listoffigures` command. In our case, the normal L<sup>A</sup>T<sub>E</sub>X method for handling figures had to be altered, so we used a package called `float.sty` to customize both our floating figures and our list of figures. I'll say more about this below (page 26).

### 6.4. Page Numbering

Frontmatter uses Roman numerals, while the rest of the text uses Arabic numerals. L<sup>A</sup>T<sub>E</sub>X makes it easy to switch between the two: `\pagenumbering{roman}` and `\pagenumbering{arabic}` are the commands. Macros that generate lists of page numbers, like the table of contents, are given the correct numerals (see above). If you don't want to start with page number 1, reset the page counter.

For Hellenists who want to go the whole nine yards, it is possible to do page numbering using Ionian numerals.

## 6.5. Headings

This is one area where L<sup>A</sup>T<sub>E</sub>X is wrongly considered difficult to customize. L<sup>A</sup>T<sub>E</sub>X provides its own heading macros. The press gave us specific instructions, which we followed by redefining L<sup>A</sup>T<sub>E</sub>X's sectioning commands. For example, the `\section` macro was redefined as follows:

```
\renewcommand{\section}{\thispagestyle{plain}
    \@startsection%
    {section}%
    {1}%
    {0mm}%
    {-2\baselineskip}%
    {4\baselineskip}%
    {\normalfont\Large\centering}
}
```

A quick glance at a good L<sup>A</sup>T<sub>E</sub>X guide, like Goosens, Mittelbach, and Samarin's *The L<sup>A</sup>T<sub>E</sub>X Companion*, and you will easily understand the parameters of this declaration. (Backmatter sections have a slightly different definition in our book.)

## 6.6. Running Headers

*Hysplex* is being published as a volume in a series which has its own heading format as well. Headings may be complicated by the constraint that pages that contain only figures normally do not have running headers. We oscillated between two ways of handling headings, and ultimately opted for the `fancyheadings` style package. We persuaded our editor to let us suppress running headers atop pages that contained only figures. Here are two ways to do this: First, our home-brewed approach, then the method provided by `fancyheadings`.

### 6.6.1. The Home-Brewed Approach

First, the declaration of our even and odd page headers:

```
\makeatletter
```

```

\def\ps@textheadings{\ps@headings
  \def\@evenhead{\if@fcolmade%
    \else \thepage\hfil{\itshape\rightmark}\hfil\fi}%
  \def\@oddhead{\if@fcolmade%
    \else\hfil{\itshape\leftmark}\hfil\thepage\fi}%
}

```

This checks to see if any text has been placed on the page. If not, it suppresses the headers. It's followed by the next instructions, which reduce the space allotted to the header to 0 so that the figures may take up the entire available space of the page.

```

\pagestyle{textheadings}

% change spacing on float pages (\@fptop, \@fpsep, \@fpbot)

\setlength{\@fptop}{-1\headheight plus 1fil}
\addtolength{\@fptop}{-\headsep}
\setlength{\@fpbot}{-1\footskip plus 1fil}

\makeatother

```

### 6.6.2. Using fancyheadings.sty

It turned out to be easier to work with the built-in macros of `fancyheadings`. If you use this package, you may find the following declarations useful.

```

\pagestyle{fancy}
\setlength{\headrulewidth}{0pt}
\lhead[\fancyplain{}{\thepage}]%      % even pages, left
  {\fancyplain{}{}}%                  % odd pp, left
\rhead[\fancyplain{}{}]%              % even pp, right
  {\fancyplain{}{\thepage}}%         % odd pp, right
\chead[\fancyplain{}{\itshape\leftmark}]% section on left
  {\fancyplain{}{\it Hysplex}}
\cfoot{}

```

Periodically, in the document, we had to redefine `\chead` when the running header had to change. (You can define your sectioning macros to do this for you, but for various reasons we opted to do it by hand.)

The `fancyheadings` method for suppressing headers on a float page is well documented. Use the `\iffloatpage` mechanism in your float definitions.

## 7. Fonts

One of the notorious hassles with  $\text{\LaTeX}$  is the use of fonts.  $\text{\TeX}$  was written before the advent of PostScript, and is designed to work with its own ancillary font language, METAFONT. METAFONT is itself an extremely powerful tool. Like  $\text{\TeX}$ , it too is an interpreter that parses instructions for drawing fonts, and generates collections of glyphs as output. The METAFONT language has not, as far as I know, been surpassed in the world of font generation tools. Its design language offers the programmer as many user-defined variables as he or she might want; in the modern, PostScript world, such variables are known as *axes*. The recently developed Multiple Master fonts from Adobe, which are regarded as the most versatile PostScript fonts yet, offer three axes of variation.

These days it is possible, though it's sometimes not simple, to use professional PostScript fonts with  $\text{\TeX}$  and  $\text{\LaTeX}$ . It's trivial to use cheap knock-off fonts that have no ligatures, small caps, hanging figures, etc.; it's also easy to use one of the thirty-five standard PostScript fonts (with good ligatures and kerning tables), for which helpful people have provided publicly all the necessary macros and metric files  $\text{\LaTeX}$  needs (see your local CTAN site). Still, to use a professional font that follows the original calligrapher's design is considerably more difficult. In our case, we were lucky enough that all the necessary fonts exist in METAFONT form.

### 7.1. Choosing the Right Font

Font selection should be deliberate. The press, when it specified the constraints for the CRC, suggested two possible fonts for us to use: Times New Roman and Palatino. Neither of these fonts appealed to me, though, for use in this project. Considering that the text is a scientific account informed by Classical Greek engineering texts and modern archaeological evidence, it seemed inappropriate to give the page the mixed Baroque and Neo-Humanist texture of Times New Roman, and no better to give it the Humanist face of Palatino. (These fonts are freely available for use with  $\text{\LaTeX}$ , so it would have been no problem to use them.) Happily,  $\text{\TeX}$ 's default font, Computer Modern, seemed an excellent choice. This typeface could be described as having a rationalist pen axis, and Neo-Classical design. Just the thing, I think, for the subject. The Greek font we used, similar to the famed Didot Greek, is perfectly compatible in my opinion.

This example shows the Roman face combined with the Greek orthotic and Greek oblique in a footnote. Note that the eight point footnote fonts are not simply scaled versions of the ten point text face. They have been recompiled by METAFONT with variations in design axis. They are most notably a little wider, by comparison, and darker, with the result that they balance visually

on the page with their bigger, ten-point siblings. Scaled fonts, in contrast, tend to look anemic in footnotes.

91. F. Dürrbach and P. Roussel, *Inscriptions de Délos* (1935) 3.29. In addition the word *hysplex* appears also in other Delian inscriptions: in *IG* XII 199.23, a the year 274 BC there is mention among other expenses of the payment of or ὕσπληγας ἐπισκευάσαντι” (to Philotas for the construction of the *hysplex*) and in an expense of 5<sup>1</sup>/<sub>2</sub> drachmas “ὀπλάριον τῆ ὕσπληγι” (for a small part for the *hysplex*) in *IG* XII 203.96 that is dated to 269 BC, and in the fragmentary *IG* XII 260.4, a “ (torchrace *hysplex*) is mentioned.

The book contains a summary in Greek. Here is a snippet. (Wedge-shaped monotonic accents are also available for use with L<sup>A</sup>T<sub>E</sub>X.) NB: Greek typographers traditionally indent the first paragraph of a chapter, in contrast with standard American practice.

Το πρόβλημα των πρώιμων εκκινήσεων στους αγώνες δρόμου, πα βρισκόμαστε με μεταφορική χρήση σε όλη την αρχαία ελληνική γραμματεία. μέχρι το τέλος της αρχαιότητας, αντιμετωπίστηκε αποτελεσματικά τους αξιωματούχους των αγώνων, με την τοποθέτηση συστημάτων αι στις balbdec των σταδίων, μπροστά από τους δρομείς. Σε όλα αυτά που ξεκινούν από απλούς ξύλινους μηχανισμούς για να καταλήξουν και στην εμφάνιση και στη λειτουργία τους κατασκευές, θα πρέπει

## 7.2. Advocacy for Fonts: A Plea for Responsibility

Tragically, typefaces are not legally defensible by copyright law. One can, in my country at least, copyright only the name of a font, and (as has recently been shown in a lawsuit between Adobe and a plagiarizer) the computer algorithm that describes a font. The letterforms themselves, which may have taken years for a skilled designer to create, are not regarded by law to be the works of art they are, and are not given the protection they deserve.

The widespread distribution of cheap knock-off fonts and plagiarism itself hinders the digitization of a font by a firm that has paid for the rights to a good artist’s design. A firm can simply not afford to spend its time and effort producing a typeface that will be (legally!) stolen as soon as it is made available. Let me give an example from my own experience. I was searching for a Greek face to accompany a certain Neo-Humanist typeface. In the specimen books, I came across a font by Hermann Zapf named *Heraklit*. The letterforms were once drawn by the calligrapher by hand for a book of Heraclitus’ fragments, and then were cut in metal for an edition of the New Testament. I cannot find the words to describe the beauty of this typeface. No digitized version exists,

though it obviously should. I wrote to Mr. Zapf and asked if *Heraklit* would ever be issued in digital form. He forwarded my letter to the Managing Director of Linotype. Here is an excerpt from the response I received from him:

Unfortunately, Heraklit is not available in digital format. It was produced by Linotype as a hard metal typeface but it was never digitized. These typefaces are amongst the most pirated ones . . . . We are not planning to produce Heraklit in the foreseeable future. We do like this type face, we have the originals, and we have Hermann Zapf's authorization. But sadly the economics are just wrong.

Of course, the violence does not stop with the plagiarism itself. Plagiarizers leave out all the “extra” characters: the ligatures, the lining figures, the small capitals, the swash characters, and whatever else was created by the designer as part of the work of art as a whole. Plagiarizers do a rough job with contours, and a rougher job with kerns. So what we are left with are artists and foundries put out of business by crooks, and shabby effigies of the original design installed with every word processor in the world. This is a tragedy. I encourage any who read this to treat fonts as works of art, and respect their designers. Use fonts that are produced by firms that licensed the designs from the designer. You can also write a letter, urging your government to regard typefaces as the works of art they are and give them according legal protection.

## 8. Figures

In the production of *Hysplex*, the figures will be pasted in by the publisher in spaces left in the CRC. Thus all we need to do is leave a space on the page with the correct caption below it, and the appropriate (probably shorter) caption in the list of figures. We declared a new type of floating figure, dubbed `hxfloat`.

```
\newfloat{hxfloat}{tp}{lox}
\floatstyle{boxed}
\floatname{hxfloat}{\footnotesize Figure}
```

Let me explain these instructions. First,  $\LaTeX$  must try to place the new float either at the top of the page `hxfloat` is invoked on, or on a subsequent page containing figures only (the meaning of `tp`). Information regarding the location of figures will be written to a file with the suffix `.lox`. This float will be surrounded by a box (useful at the draft stage, where we want to see the virtual outline of the figure—in the final version for the printer, the linewidth of the

box will be set to 0). Finally, captions will begin with the word “Figure” in the same type size as the footnote text.

Another command, this one also called `hxfloat`, governs the appearance of the float.

```

\makeatletter
\newcommand\hxfig{%
  \@ifnextchar[%
    \threeargfig
    \twoargfig}
\makeatother

\def\twoargfig#1#2{\threeargfig[#2]{#1}{#2}}

\def\threeargfig[#1]#2#3{%
  \begin{hxfloat}
  \begin{center}
  \fbox{\parbox[b][#2in][s]{30pc}%
    { \vfil
      { \hfil Figure height\ #2 in \hfil }
      \vfil }}
  \end{center}
  \caption[#1]{\footnotesize $\greightpoint$#3%
    $\grtenpoint$}
  \end{hxfloat}}

```

This defines a command that takes two or three parameters, the first (optional) parameter being a short caption for the float which will be placed in the list of figures, the second (required) being the height of the figure (in inches), and the third (required) being the full caption to appear below the figure in the text. If you call `\hxfig` with just two arguments, it will behave as though you supplied the value for the second required argument (the full caption) to the optional argument as well. Thus, only if the list of figures needs a different version of the caption do you need to specify it by means of the optional argument. The `\fbox` command generates the box with the right height, with text inside stating the height of the figure. The `\caption` command supplies the text both for the list of figures and for the caption below the figure itself. Elsewhere, we have redefined `\caption` to center the caption horizontally if it is narrower than the width of the textblock, and to format it as a paragraph if it contains more than one line.

## 9. Cross References

$\LaTeX$  and  $\TeX$  (using `eplain` with the latter) have powerful tools for labelling and addressing cross references. Using automatic cross references, we don't have to write specifically

```
... see above, page~83, ...
and hope that the document's pagination will never
change---because it will.
Rather, we can write
\begin{verbatim}
... see above, page~\pageref{myref}, ...
```

This will produce in the output the page number associated with the location in the source file where the command `\label{mylabel}` was made. The more general command `\ref` is context sensitive; it knows if the `\label` is in a footnote, on a figure, or in a section or subsection, etc., and will produce the appropriate number. You can use the same label for as many `\ref` and `\pageref` commands as you want. For example,

```
... see above, note~\ref{myref} on page~\pageref{myref}, ...
```

will do what you think it does.

To make it easier for an author or editor to be sure that cross references are pointing at the right block of text, a useful package `showref` may be used. When included, `showref` causes a box to be printed in the margin with the keyword in it wherever a `\label` is invoked, and the keyword to be printed above the resolved reference in the text. When everyone is confident your labels are all properly placed, you can simply `uninclude showref`, and the marginal marks go away.

## 10. Bibliography

The author did not provide a bibliography for this book. (There was instead a list of important bibliographic references at the beginning of the book, and numerous cross references between footnotes to help the reader find other references.) For the sake of illustration, I've included bibliographic examples here from another source. There are three aspects to bibliographic references in  $\LaTeX$  (and `eplain \TeX`). First, you must have a database of sources; second, you can use various citation commands to produce references of your stylistic



choosing in the text; third, you can use any number of bibliographic style files, or create your own, to govern the format of your printed bibliography. There are other capabilities I won't go into here, such as the production of multiple bibliographies, the production of bibliographic reference cards for yourself, etc.

The bibliography database has entries like this:

```
@book{vendryes:1902,
  author = {J. Vendryes},
  title = {Recherches sur l'histoire et les effets
           de l'intensit\`e initiale en latin},
  year = 1902,
  publisher = {C. Klincksieck},
  address = {Paris},
  annote = {PA~2119 .V45 1902}
}
```

The database itself is a file terminating in the suffix `.bib`. The first string after the open quote is the keyword. I like keywords of the format `author:year`, because they are easy to remember or guess.

In the text, you use citation commands that reference the keyword of the bibliographic entry, e.g., `\citeN{vendryes:1902}`.

Somewhere in your document source, you will have commands like these.

```
\bibliographystyle{tapa}
\bibliography{bib-file-name}
\nocite{*}      % print all bib entries, not just those cited
```

This will cause the bibliography to be printed here, formatted according to the rules in the bibliographic style file (terminating in `.bst`) which you specify. In this example, I have created a style file with the `makebst` program (freely available) which emulates the bibliography format of the Classics journal TAPA.

The result is text that looks like this (here, a footnote):

<sup>7</sup>The standard bibliography includes, for a quantitative solution, Leo (1905), Burger (1928), de Groot (1934), and Cole (1969); for an accentual solution, Lindsay (1893); for a syllable-counting solution, Pighi (1957) (see below note 24). For a recent discussion and assessment of the bibliography, see Kloss (1993).

Some of the bibliography (with `tapa.bst`) looks like this:

- Prodocimi, A. L., 1986. “Sull’accento latino e italico.” In A. Etter, ed., *o-o-pe-ro-si, Festschrift für Ernst Risch zum 75. Geburtstag*, pages 601–618. Berlin: de Gruyter.
- Soubiran, J., 1988. *Essai sur la versification dramatique des Romains*. Paris: Centre Nationale de la Recherche Scientifique.
- Steriade, D., 1988. “Greek Accent: A Case for Preserving Structure.” *Linguistic Inquiry* 19(2): 271–314.

## 11. Index

Indices are easy to generate with L<sup>A</sup>T<sub>E</sub>X and eplain T<sub>E</sub>X using the simple command `\index`. You can arrange a hierarchical index, with nested entries, sorted in any order you want. This last feature was particularly important in this book, where we needed Greek words in the index alphabetized according to English rules.

One problem we encountered was that `\index`, as implemented, does not behave differently when called from within a footnote. Happily, the Usenet group `comp.text.tex` is very active and helpful. Hours after we posted a question to the group, Stefan Ulrich responded with this definition for `\footindex`.

```
\makeatletter
\newcommand\asfoot[1]{\@printfoot#1\end}
\def\@printfoot#1-#2\end{#1 {\it n}\/#2}%

\newcommand*\@wrfootindex[1]{%
\protected@edef\@tempa{\noexpand\protected@write\noexpand\@indexfile{}}%
{\string\indexentry{#1|asfoot}{\noexpand\thepage-\thefootnote}}}%
\@tempa
\endgroup\@esphack}

\newcommand\footindex{\@bsphack\begingroup
\@sanitize
\@wrfootindex}
\makeatother
```

Here is a sample of the book’s two-column index. Greek readers will find the alphabetization strange, but it was done this way to meet the expectations of English-speaking readers.

- Priene, 20 *n*68, 144
  - Stadium I, 67
  - Stadium II, 113–119, 125, 137
- proclamation, 2
- propylon*, 80
- φύφος, 169
- πτελείνος, 36
- πτέρνα, 39
- pulleys, 69

When looking at a printed copy, just as you might want to know where you invoked your label and reference commands, you (or someone else) might want to see exactly where you invoked indexing commands. Use the style file `showidx` and with no extra instructions, L<sup>A</sup>T<sub>E</sub>X will print your index keywords in the margin on the page in which they were declared.

## 12. Crop Marks

The printer who is photsetting your CRC will need cropmarks to center the camera with. There are many macro packages for generating these. We used `cropmark.sty` and set the following parameters:

```
\setlength{\cropwidth}{.2pt}  
\setlength{\cropsep}{1pc}  
\setlength{\croppadlr}{1pc}
```

## 13. Further T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X Resources

### 13.1. Online

<http://www.tug.org/>

<http://www.ctan.org/>

Usenet: `comp.text.tex`

### 13.2. Printed

Don Knuth, *The T<sub>E</sub>Xbook*. Addison-Wesley.

Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley.

Michel Goossens, Frank Mittelbach, and Alexander Samarin, *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley.

Michel Goossens, Sebastian Rahtz, and Frank Mittelbach, *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*. Addison-Wesley.

Alan Hoenig, *T<sub>E</sub>X Unbound*. Oxford University Press.

Robert Bringhurst, *The Elements of Typographic Style*. Hartley & Marks.

# MathsPIC: A filter program for use with PICTEX

---

Richard W. D. Nickalls

*Department of Anaesthesia,  
City Hospital, Nottingham, UK.*

*dicknickalls@compuserve.com*

## Abstract

This article presents an overview of the mathsPIC utility for the PICTEX drawing engine. mathsPIC facilitates the drawing of mathematical diagrams by allowing the manipulation of points by name rather than by coordinates. Some familiarity with the PICTEX package is assumed.

*This article is a slight modification/update of the original article presented at the EuroTEX99 meeting in Heidelberg, Germany (September, 1999).*

## 1. Introduction

mathsPIC<sup>1</sup> is a filter program for use with the PICTEX drawing engine and its various addons<sup>2</sup>. It parses a plain text input-file (the mathsPIC file), and generates a plain text output-file containing PICTEX and TEX commands, which can then be TEXed (or LATEXed) in the usual way. Spaces and the comment % symbol are used in the same way as TEX, although unlike TEX, mathsPIC commands are *not* case-sensitive; PICTEX commands can also be freely used in the mathsPIC file. mathsPIC also returns various parameter values in the output file, e.g. angles, distances between points, center and radius of inscribed and exscribed circles etc., since such values can be useful when making adjustments to a diagram.

---

<sup>1</sup> CTAN/tex-archive/graphics/pictex/mathspic/

<sup>2</sup> See particularly pictexwd.sty and other related files by Andreas Schrell in CTAN/tex-archive/graphics/pictex/addon/.

The motivation for `mathsPIC` stems from the fact that, while `PiCTeX` is an extremely versatile system for drawing figures, and offers the convenience and advantages of having the graphics code within the `TEX` document itself (e.g. printer-independence), it does require you to specify the coordinates of most points. This can make it quite awkward to use with complicated diagrams, particularly if several coordinates have to be re-calculated manually each time the diagram is adjusted.

For example, the `PiCTeX` commands for drawing Figure1, namely a triangle  $ABC$  with  $AB$  5cm,  $AC$  3cm, and included angle  $BAC$  40 degrees, and its incircle, are as follows.

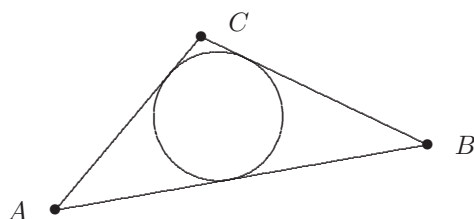


Figure 1:

```
\put {$\bullet$} at 0 0 % draw point A
\put {$\bullet$} at 4.9240 0.8682 % draw point B
\put {$\bullet$} at 1.9283 2.2981 % draw point C
\plot 0 0 4.9240 0.8682 1.9283 2.2981 0 0/ % draw triangle
% draw incircle
\circulararc 360 degrees from 3.0086 1.245 center at 2.1568 1.245
```

Although point  $A$  can be placed at the origin for convenience it is then necessary, since in this particular example  $AB$ ,  $AC$ , and the included angle are defined (see above), to resort to geometry and a calculator to determine points  $B$  and  $C$ . It is then necessary to recall the coordinates of all the points in order to write the `\plot` command. Finally, the `\circulararc` command requires even more geometry and calculation to figure out the radius of the incircle, the coordinates of its center, and the coordinates of the starting point of the arc drawing routine. Furthermore, if the initial diagram is not a suitable shape or size, the calculator has to be used again for any adjustments. In practice, therefore, `PiCTeX` requires a certain amount of planning and calculation for all but the simplest of diagrams.

`mathsPIC` provides an environment for manipulating named points, which makes `PiCTeX` convenient to use and mathematical diagrams easy to create.

For example, the equivalent mathsPIC commands for drawing Figure 1 are as follows.

```
point(A){0,0}           % point A at origin
point(B){A,polar(5,10deg)} % point B 5cm from A
point(C){A,polar(3,50deg)} % point C 3cm from A, BAC = 40 deg
drawPoints(ABC)        % draw points A B C
drawTriangle(ABC)
drawIncircle(ABC)
```

## 2. Points

Each point is associated with a point-name which is defined using a `point` command. Once defined, points can be referred to by name.

A point-name *must* begin with a *single* letter, and may have up to a *maximum* of two following digits. The following are valid point-names:  $A$ ,  $B$ ,  $C3$ ,  $d45$ . Since mathsPIC is not case sensitive the points  $d45$  and  $D45$  are regarded as being the same point. Point-names can be either separated by spaces, or simply run together. Where coordinates need to be specified, the point is defined using the following command: `point(name){ $x$ , $y$ }`. Existing point-names can be re-allocated new coordinates using the equivalent `point*` command.

The default point-character is the symbol  $\bullet$  (`\bullet`). However, mathsPIC allows the optional use of any T<sub>E</sub>X character to represent a point, by including it in a square bracket. For example, the point  $A(5,10)$  can be represented by the  $\square$  symbol<sup>3</sup> by defining it as follows.

```
point(A){5,10}[\Box]
```

Note that when lines are drawn to a point, the line will (unless otherwise instructed) extend to the point location. However, this can be prevented by allocating an optional circular line-free zone to a point by specifying the radius in the square bracket. For example, lines to a  $\square$  symbol at point  $A$  (above) can be prevented from being drawn through the box to its center, by allocating a 5 unit line-free zone to the point, as follows.

```
point(A){5,10}[\Box$,5]
```

If only the radius is specified (e.g. [5]) then the line-free zone will be applied to the default point-character (`\bullet`). However, if there is no character

<sup>3</sup> Available in the `latexsym` package.

before the comma (e.g. `[,5]` or `[ ,5]`) then the point-character will be the space character.

Points can also be defined in relation to other points or lines; for example, as the midpoint of a line, or the intersection of two lines.

Points are interpreted according to their grouping and context. Thus two points represent either a line or its length. For example, `drawCircle(C3,AB)` means draw a circle, center  $C3$ , with radius the length of line  $AB$ . Three points represent either a triangle, an angle, or a line, depending on the circumstances.

The order of points in `mathsPIC` commands is often significant. For example, the command `point(D){PointOnLine(AB,23)}` defines the point  $D$  as being 23 units *from A in the direction of B*. Some examples of valid `mathsPIC` point commands are as follows.

```
point(A){5,5}
point(B2){22,46}[$\triangle$,6]
point(C){3,8}[4]
point*(A){A,shift(3,0)}
point(D){midpoint(AB)}
point(e2){intersection(AB,CD)}
point(f){PointOnLine(AB,+5)}
point(g){perpendicular(C,AB)}
point(h){D,shift(3,-2)}
point(j){e2,polar(3,122 deg)}
point(k){F,polar(4, 0.5 rad)}
point(m){circumcircleCenter(ABC)}
point(n){incircleCenter(ABC)}
point(p){excircleCenter(ABC,BC)}
```

The point-symbol is drawn using the `drawPoint` or `drawPoints` command; for example, `drawPoint(A)`, or `drawPoints(ABCD)`.

### 3. Variables

Although magnitude can be specified by defining two points, numeric variables can be more conveniently defined using the `variable(name){value}` command. The *value* in the braces can be either a numeric (e.g. 4.32), a variable name (e.g. `r3`), or two points (e.g. `AB`) meaning the distance between the two points. Thus the command `variable(r3){20}` allocates 20 to the variable `r3`, which could then be used, for example, in the circle command



`drawcircle(C3,r3)`. New values can be re-allocated to an existing variable-name using the equivalent `variable*` command. Variables can also be manipulated mathematically using the terms `advance`, `multiply`, `divide`, and `mod`. Thus the command `variable(t1){t0,advance(-4)}` is equivalent to  $t_1 = t_0 - 4$ .

However, one must be careful not to mistake variables for points, and vice versa, since they both have the same name structure. When both are necessary, using upper case for points and lower case for variables is a convenient strategy. Some examples of valid mathsPIC `variable` commands are as follows.

```
variable(r){5}
variable*(r23){-6}
variable(p3){x}
variable(y4){y1,divide(9)}
variable(a2){angle(ABC)}
variable(x){area(ABC)}
variable(y1){Ypoint(A)}
```

## 4. Lines

Lines are drawn using the `drawLine` or `drawLines` command. For example, a line from  $P_1$  to  $P_2$  is drawn with the command `drawLine(P1P2)`. If a line is to be drawn through several points (say,  $J_1, J_2, J_3, J_4, J_5$ ) and can all be drawn ‘without lifting the pen’, then this can be achieved in one go with `drawLine(J1J2J3J4J5)`. Several unconnected lines can be drawn using one command by separating the line segments with commas; for example, `drawLines(J1J2,J3J4J5,J1J3)`.

## 5. Text

Text is typeset using the `text` command and, by default, is centered both horizontally and vertically at a defined point. For example, the box symbol  $\square$  would be placed at the point-location  $Z$  using the command `text($\Box$){Z}`. Optionally, text can be placed relative to a point using appropriate combinations of the P<sub>1</sub>CT<sub>E</sub>X options `B t b l r` to align the **B**aseline, **t**op, **b**ottom, **l**eft, **r**ight edges respectively of the text box with the point-location (see Wichura, 1992). For example, the text box This is point P would be aligned such that the right edge of the text box would be centered vertically at the point  $P$ , using `text(This is point P){P}[r]`.

Alternatively, text can be located relative to the point-location using the optional `shift(dx, dy)` or `polar(r,  $\theta$ )` commands. For example, points  $P_1P_2P_3$  could have their labels located 4 units from each point as follows.

```
text($P_1$){P1,shift(-4,0)}
text($P_2$){P2,polar(4,10 deg)}
text($P_3$){P3,polar(4,0.29088 rad)}
```

## 6. Using $\text{P}\text{T}\text{E}\text{X}$ and $\text{T}\text{E}\text{X}$ commands

In order to protect  $\text{P}\text{T}\text{E}\text{X}$  commands within the `mathsPIC` file from being acted on by `mathsPIC` they are included as the argument for the `pictex{...}` command, which simply copies its argument unchanged into the output file. For example, drawing with dashed lines is enabled using the  $\text{P}\text{T}\text{E}\text{X}$  command `\setdashes`, which is used in the `mathsPIC` file as `pictex{\setdashes}`.

In practice, it is also useful to include in the `mathsPIC` file any  $\text{T}\text{E}\text{X}$  or  $\text{L}\text{A}\text{T}\text{E}\text{X}$  headers and footers which would otherwise have to be added manually to the output file before  $\text{L}\text{A}\text{T}\text{E}\text{X}$ ing the file. This is achieved using the similar `tex{...}` and `latex{...}` commands. For example, when using  $\text{L}\text{A}\text{T}\text{E}\text{X}$  a typical format for a `mathsPIC` file might be as follows.

```
latex{\documentclass[a4paper]{article}}
latex{\usepackage{pictexwd,latexsym,amssymb}}
latex{\begin{document}}
pictex{\beginpicture}
paper{units(mm),xrange(0,50),yrange(0,50),axes(LB)}
...
...
pictex{\endpicture}
latex{\end{document}}}
```

## 7. Input and output files

The following example `mathsPIC` file (`triangle.m`) illustrates how some of these commands are used to draw Figure 2. Setting up the  $\text{P}\text{T}\text{E}\text{X}$  plotting area and axes is greatly simplified by the `mathsPIC` one-line `paper` command. Note that the dashed line  $AD$  is drawn after invoking the  $\text{P}\text{T}\text{E}\text{X}$  `\setdashes` command, and then `\setsolid` is used before drawing the right-angle symbol.

Also, the points  $A$ ,  $B$ ,  $C$  are defined using the  $\text{\TeX}$   $\odot$  symbol, in conjunction with a line-free zone of 1.2 mm in order to make the lines go to the edge of the symbol—the radius of such  $\text{\TeX}$  symbols has to be determined by trial and error.

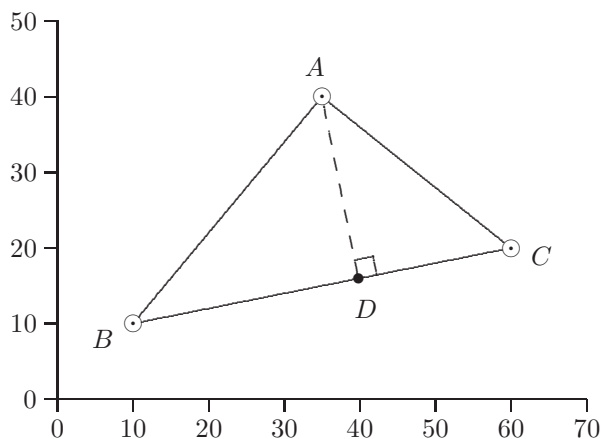


Figure 2:

```

pictex{\beginpicture}
paper{units(mm),xrange(0,70),yrange(0,50),axes(LB),ticks(10,10)}
point(A){35,40}[$\odot$,1.2]
point(B){10,10}[$\odot$,1.2]
point(C){60,20}[$\odot$,1.2]
point(D){perpendicular(A,BC)} %% from point A to line BC
drawPoints(ABCD)
drawLine(ABCA)
pictex{\setdashes}
drawLine(AD)
pictex{\setsolid}
drawRightangle(ADC,2.5)
text($A$){A,shift(-1,4)}
text($B$){B,shift(-4,-2)}
text($C$){C,shift(4,-1)}
text($D$){D,shift(1,-4)}
showLength(AD)
showLength(BC)
showArea(ABC)
pictex{\endpicture}

```

When the above file is processed by `mathsPIC` the output file is as follows. Note how the `PiCTEX` commands are accompanied by their `mathsPIC` commands (commented out), some of which have additional information added (e.g. the coordinates of a derived point—see `%% point(D)...`). Being able to compare the `mathsPIC` and resulting `PiCTEX` commands in the output file is particularly useful when debugging. Note also how the `show` commands return the lengths  $AD$ ,  $BC$ , and the area  $ABC$ .

```
%% triangle.m (Figure 2)
\beginpicture
%% paper{units(mm),xrange(0,70),yrange(0,50),axes(LB),ticks(10,10)}
\setcoordinatesystem units <1mm,1mm>
\setplotarea x from 0 to 70 , y from 0 to 50
\axis left ticks numbered from 0 to 50 by 10 /
\axis bottom ticks numbered from 0 to 70 by 10 /
%% point(A){35,40}[$\odot$,1.2]
%% point(B){10,10}[$\odot$,1.2]
%% point(C){60,20}[$\odot$,1.2]
%% point(D){perpendicular(A,BC)} ( 39.80769 , 15.96154 )
%% drawPoints(ABCD)
\put {$\odot$} at 35 40 %% A
\put {$\odot$} at 10 10 %% B
\put {$\odot$} at 60 20 %% C
\put {$\bullet$} at 39.80769 15.96154 %% D
%% drawLine(ABCA)
\plot 34.23178 39.07813 10.76822 10.92187 / %% AB
\plot 11.1767 10.23534 58.8233 19.76466 / %% BC
\plot 59.06296 20.74963 35.93704 39.25037 / %% CA
\setdashes
%% drawLine(AD)
\plot 35.23534 38.8233 39.80769 15.96154 / %% AD
\setsolid
%% drawRightangle(ADC,2.5)
\plot 42.25914 16.45183 41.76886 18.90328 /
\plot 39.3174 18.41299 41.76886 18.90328 /
%% text($A$){A,shift(-1,4)}
\put {$A$} at 34 44
%% text($B$){B,shift(-4,-2)}
\put {$B$} at 6 8
%% text($C$){C,shift(4,-1)}
\put {$C$} at 64 19
%% text($D$){D,shift(1,-4)}
```

```
\put {$D$} at 40.80769 11.96154
%% showLength(AD) 24.51452
%% showLength(BC) 50.9902
%% showArea(ABC) 625.0001
\endpicture
```

## 8. Arrows

Arrows can be drawn in all possible orientations, will *stretch* between points, and arrowheads are readily customised using the `mathsPIC setArrowshape` command (see also Salomon, 1992). Curved arrows are drawn using the `drawAngleArrow` command, which takes parameters for the angle, radius of arc, direction, and whether the angle is internal or external (see Figures 3 and 4).

```
%% arrow1.m (Figure 3)
pictex{\beginpicture}
paper{units(mm),xrange(5,45),yrange(5,45)}
point(A){30,30}
point(P){10,10}
point(B){30,10}
drawPoints(APB)
drawLine(APBA)
text($A$){A,shift(1,5)}
text($B$){B,shift(5,0)}
text($P$){P,shift(-5,0)}
drawAngleArrow{angle(BPA),radius(11),anticlockwise,internal}
text($\psi$){P,polar(7,22.5 deg)}
drawRightangle(ABP,2.5)
pictex{\endpicture}
```

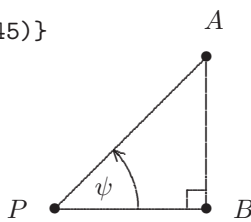


Figure 3:

Arrows are also used to link elements in a diagram, as shown in Figure 4. The right-hand diagram uses the `drawArrow` command; the small gap between the arrows and the letters  $P, Q, R, T$  being due to the 5 unit line-free radius associated with these points (see `arrow2.m`). The arrows are easily ‘stretched’ to accommodate their labels simply by adjusting the separation of the nodes in the `polar( $r, \theta$ )` commands (cf. Feruglio, 1994).

```
%% arrow2.m (Figure 4)
pictex{\beginpicture}
```

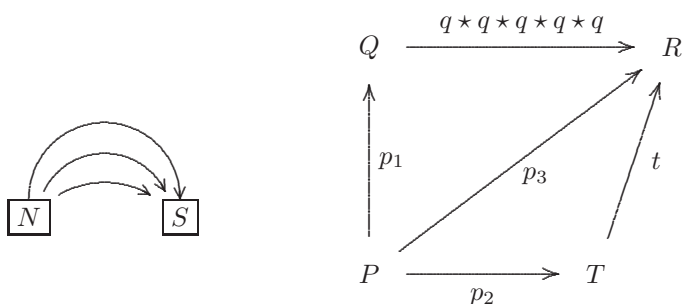


Figure 4:

```

paper{units(mm),xrange(10,110),yrange(0,45)}
%%-----Left-hand diagram-----
point(N){15,20}
point(S){N,shift(20,0)}
text(\framebox{N}){N,shift(0,-2.5)}
text(\framebox{S}){S,shift(0,-2.5)}
point(Z){midpoint(NS)}
drawAngleArrow{angle(NZS),radius(NZ),clockwise,internal}
point(N1){N,shift(2,1)}
point(S1){S,shift(-2,1)}
point(Z1){Z,shift(0,-3)}
drawAngleArrow{angle(N1Z1S1),radius(N1Z1),clockwise,internal}
point(N2){N1,shift(2,-0.5)}
point(S2){S1,shift(-2,-0.5)}
point(Z2){Z,shift(0,-10)}
drawAngleArrow{angle(N2Z2S2),radius(N2Z2),clockwise,internal}
%%-----Right-hand diagram-----
point(P){60,10}[$P$,5]
point(Q){P,polar(30,90 deg)}[$Q$,5]
point(R){Q,polar(40,0 deg)}[$R$,5]
point(T){P,polar(30,0 deg)}[$T$,5]
drawPoints(PQRT)
drawArrows(PQ,QR,PT,TR,PR)
point(P1){midpoint(PQ)}
text($p_1$){P1,shift(3,0)}
point(P2){midpoint(PT)}
text($p_2$){P2,shift(0,-3)}
point(P3){midpoint(PR)}
text($p_3$){P3,shift(2,-2)}

```

```
point(T1){midpoint(TR)}
text($t$){T1,shift(3,0)}
point(Q1){midpoint(QR)}
%% use a macro for the label
latex{\newcommand{\q}{\q \star q \star q \star q \star q}}
text(\q){Q1,shift(-1,3)}
pictex{\endpicture}
```

## 9. Circles

mathsPIC allows the point-symbol to be designated as a circle using the optional [circle,r] to the point command, which then allows the circles not only to have a line-free zone, but also to be drawn using the `drawPoints` command as shown in Figure 5. This greatly simplifies the drawing of directed graphs, trees and equivalent structures.

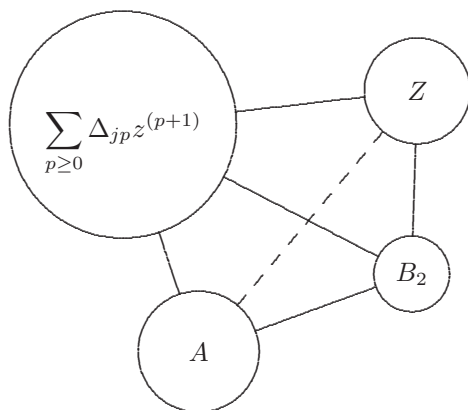


Figure 5:

```
%% points4.m (Figure 5)
pictex{\beginpicture}
paper{units(mm),xrange(0,65),yrange(0,55)}
point(A){30,10}[circle,8]
point(B1){A,shift(-10,30)}[circle,15] %% big circle
point(B2){A,polar(30,20 deg)}[circle,5]
point(Z){A,polar(45,50 deg)}[circle,7]
```

```

drawPoints(AB1B2Z)
drawLines(AB1,AB2,B1B2,B1Z,ZB2)
pictex{\setdashes}
drawLine(AZ)
text($A$){A}
% use a macro for the maths
latex{\newcommand{\BB}{\sum_{p\ge 0}\Delta_{jp}z^{(p+1)}}}
text(\vbox{\BB}){B1,shift(0,-2)}
text($B_2$){B2}
text($Z$){Z}
pictex{\endpicture}

```

Points on circles (and their labels) are most easily defined and positioned using the `polar` parameter, as shown in the `mathsPIC` file for Figure 6. Note the use of the `variable` command to define the radius  $r_2$ .

```

% circle.m (Figure 6)
pictex{\beginpicture}
paper{units(mm),xrange(0,55),yrange(0,55)}
point(C){30,30}[$\odot$,1.2]
variable(r2){20} % radius
drawCircle(C,r2) %% center C
point(P){C,polar(r2,250 deg)}
point(Q){C,polar(r2,120 deg)}
point(R){C,polar(r2,-30 deg)}
drawPoints(CPQR)
drawLines(PCRQP)
showAngle(PQR) % alpha
showAngle(PCR) % beta
text($P$){P,polar(5,30 deg)}
text($Q$){Q,polar(5,160 deg)}
text($R$){R,polar(5,-30 deg)}
drawAngleArrow{angle(PCR),radius(8),anticlockwise,internal}
text($\beta$){C,polar(5,285 deg)}
drawAngleArrow{angle(PQR),radius(12),anticlockwise,internal}
text($\alpha$){Q,polar(8,-65 deg)}
pictex{\endpicture}

```

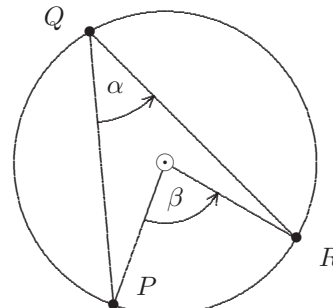


Figure 6:

Note that the returned values in the output file from the `showAngle` commands (see below) for Figure 6 indicate that  $\beta$  is twice  $\alpha$  as expected.

```

% showAngle(PQR) 40 degrees / radians 0.6981316 % alpha

```



```
%% showAngle(PCR) 79.99999 degrees / radians 1.396263 % beta
```

mathsPIC offers a range of circle commands (`drawIncircle`, `drawCircumcircle`, `drawExcircle`) specifically for geometry diagrams (cf. Cameron, 1992), as shown in Figure 7. This example starts with the triangle  $ABC$ , and later points  $D$  and  $E$  are constructed in order to extend lines  $AC$  and  $AB$ .

```
%% incircle.m (Figure 7)
```

```
pictex{\beginpicture}
paper{units(mm),xrange(0,70),yrange(25,70)}
point(A){10,40}
point(B){30,35}
point(C){40,55}
drawPoints(ABC)
drawTriangle(ABC)
drawCircumcircle(ABC)
drawIncircle(ABC)
%% extend AC and AB
point(D){pointOnLine(AC,50)}
point(E){pointOnLine(AB,50)}
drawLines(CD,BE)
drawExcircle(ABC,BC)
text($A$){A,shift(-3,-4)}
text($B$){B,shift(0,-5)}
text($C$){C,shift(2,5)}
point(P1){circumcircleCenter(ABC)}[$\odot$,1.2]
point(P2){incircleCenter(ABC)}[$\odot$,1.2]
point(P3){excircleCenter(ABC,BC)}[$\odot$,1.2]
drawPoints(P1P2P3)
pictex{\setdashes}
drawLine(P1P2P3P1)
pictex{\endpicture}
```

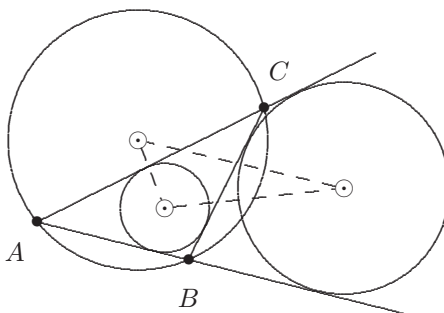


Figure 7:

## 10. Functionally connected diagrams

When constructing diagrams it is often useful to write the mathsPIC file in such a way that new points are related to earlier points, since the structure of the diagram is then maintained even when points are moved. This is demonstrated in Figure 8, where the mathsPIC code for the two diagrams differs *only*

in the angle of the line  $AB$  (left diagram, 60 degrees; right diagram, 5 degrees) as defined in the `point(B){...}` command as follows.

- Left-hand diagram: `point(B){A,polar(45,60 deg)}`
- Right-hand diagram: `point(B){A,polar(45,5 deg)}`

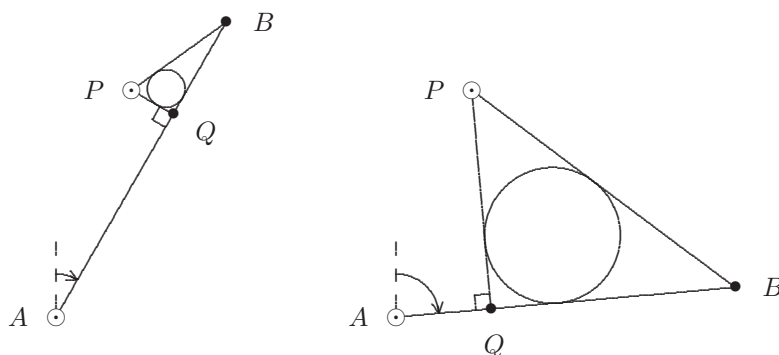


Figure 8: The mathsPIC code for the two diagrams differs *only* in the angle of the line  $AB$  as defined in the `point(B){...}` command (see `dynamic.m`).

```

%% dynamic.m (Figure 8)
pictex{\beginpicture}
paper{units(mm),xrange(5,120),yrange(0,45)}
%% left-hand diagram
point(A){15,5}
point(P){A,shift(10,30)}
point(B){A,polar(45,60 deg)}
point(Q){perpendicular(P,AB)}
drawRightangle(PQA,2)
drawPoints(ABPQ)
drawLine(ABPQ)
drawIncircle(PQB)
text($A$){A,shift(-5, 0)}
text($B$){B,shift(5, 0)}
text($P$){P,shift(-5, 0)}
point(S){pointOnLine(QP,-5)}
text($Q$){S}
%% now draw vertical line and angle

```

```

point(N){A,shift(0,12)}
pictex{\setdashes}
drawLine(AN)
pictex{\setsolid}
drawAngleArrow{angle(NAB),radius(7),clockwise,internal}
pictex{\endpicture}

```

## 11. Inputting files and recursion

mathsPIC allows the recursive input of blocks of commands as files, using the `inputfile` command to input files containing mathsPIC commands. In practice, this facility is equivalent to a ‘DO-LOOP’ in a program. For example, the file `myfile.dat` would be input six times sequentially using the command `inputfile(myfile.dat) [6]`.

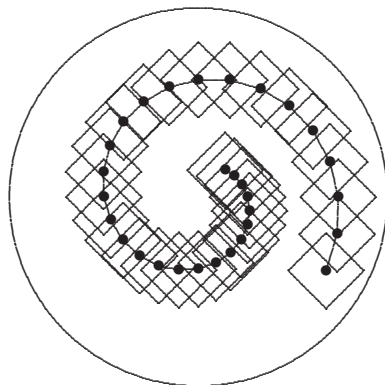


Figure 9:

Figure 9 was produced by the following code (`spiral.m`), which inputs a small file of mathsPIC code (`spiral.dat`) recursively 30 times using the `inputfile(spiral.dat) [30]` command. Note the use of the `variable*` and `point*` commands to re-allocate variables and points recursively, as well as the use of the `advance` term.

```

%% spiral.m
pictex{\beginpicture}

```

```

paper{units(mm), xrange(0,60), yrange(0,60)}
point(C){30,30} % circle center
drawcircle(C,25)
variable(a){315} % angle deg
variable(r){20} % radius of spiral
variable(s){5} % square, semi-diagonal
point(T){C,polar(r,330 deg)}
inputfile(spiral.dat)[30]
pictex{\endpicture}

% spiral.dat
variable*(r){r,advance(-0.5)} % let r=r-0.5
variable*(a){a,advance(15)} % let a=a+15 deg
point*(P){C,polar(r,a deg)}
drawpoint(P)
drawline(TP)
point*(T){P} % let T=P
% make a square centered on P
point*(Q1){P,polar(s,0)}
point*(Q2){P,polar(s,90)}
point*(Q3){P,polar(s,180)}
point*(Q4){P,polar(s,270)}
drawline(Q1Q2Q3Q4Q1)

```

Data-files which do not contain `mathsPIC` commands can be input using the `inputfile*` command. This command inputs files verbatim, and so can be used for inputting files containing points for plotting curves and `PTCTEX` commands.

## 12. Conclusion

`mathsPIC` facilitates the drawing of `PTCTEX` diagrams mainly because it provides an environment for manipulating named points, and makes it easy for points to be defined in terms of other points. Geometry diagrams can therefore be constructed in an intuitive way, much as one might with a compass and ruler; for example, constructing a point at a certain position simply to allow some other point to be constructed, perhaps to draw a line to. In other words, `mathsPIC` offers the freedom to create ‘hidden’ points having a sort of scaffolding function. In particular, this facility allows diagrams to be constructed in such a way that they remain functionally connected even when points are moved.

Note that `mathsPIC` can also be viewed as a handy tool for exploring elementary geometry since its `show` commands return the values of various parameters; for example, angles, the distance between points, coordinates of derived points, areas of triangles etc. A detailed 37-page manual on `mathsPIC` is available at [CTAN/graphics/pictex/mathspic/](http://CTAN/graphics/pictex/mathspic/).

Finally, I would like to thank Apostolos Syropoulos and Bob Schumacher for beta-testing this program, and for their many ideas and suggestions.

### 13. References

- Cameron P. J. (1992): *Geometric diagrams in L<sup>A</sup>T<sub>E</sub>X*. *TUGboat* **13** (No. 2), 215–216.
- Feruglio G. V. (1994): *Typesetting commutative diagrams*. *TUGboat* **15** (No. 4), 466–484.
- Salomon D. (1992): *Arrows for technical drawing*. *TUGboat* **13** (No. 2), 146–149.
- Wichura M. J. (1992): *The P<sub>r</sub>CT<sub>E</sub>X manual*. Pub: Personal T<sub>E</sub>X Inc., 12 Madrona Avenue, Mill Valley, CA 94941, USA. <http://www.pctex.com/>



## Creating device independent PostScript

---

James Kilfiger

*Department of Mathematics  
University of Warwick  
Coventry, UK  
e-mail:mapdn@csu.warwick.ac.uk*

**T<sub>E</sub>X and fonts** T<sub>E</sub>X is a program with a heritage. When it was first written computer typesetting was in it's infancy. There were no high quality, publicly available fonts which it could use. To overcome this problem Knuth wrote METAFONT [3]. This allows the user to create their own fonts by describing them with a fairly simple computer language. Knuth himself designed the computer modern families of fonts with METAFONT and since then a wide range of other metafonts have been created for scripts as varied as Devanāgarī and Klingon.

Since that time many other font formats have been invented. These may be broadly classified into 'bitmap' and 'vector' formats (see figure 10). Type 1 and Type 3 fonts are PostScript fonts developed by Adobe. Type 3 fonts may contain arbitrary PostScript commands, whereas Type 1 fonts can only describe the outlines of characters. Truetype is another outline font format developed by Apple Computers. The .pk and .gf formats are generated by METAFONT<sup>1</sup> and the .bdf and .fon formats are mainly used for on-screen fonts.

**Why so many formats!?** This plethora of formats can be annoying, 'How do I convert...' is probably the most frequently asked question on `comp.text.fonts`. Each manufacturer made their own format, often the details of these formats were kept secret. Adobe encrypted their fonts in an attempt to prevent unauthorized sharing of their fonts. Only with the publication of the Black Book [1] did they allow others to make Type 1 fonts. It would be nice if there was one international standard for vector fonts and another for bitmaps, but despite encouraging moves, such as the development of OpenType fonts,<sup>2</sup> for the moment we have to rely on conversion programs.

---

<sup>1</sup> Actually each .pk file is a compressed .gf file generated by `gftopk`.

<sup>2</sup> OpenType is a format which aims to be compatible with both Type 1 and Truetype

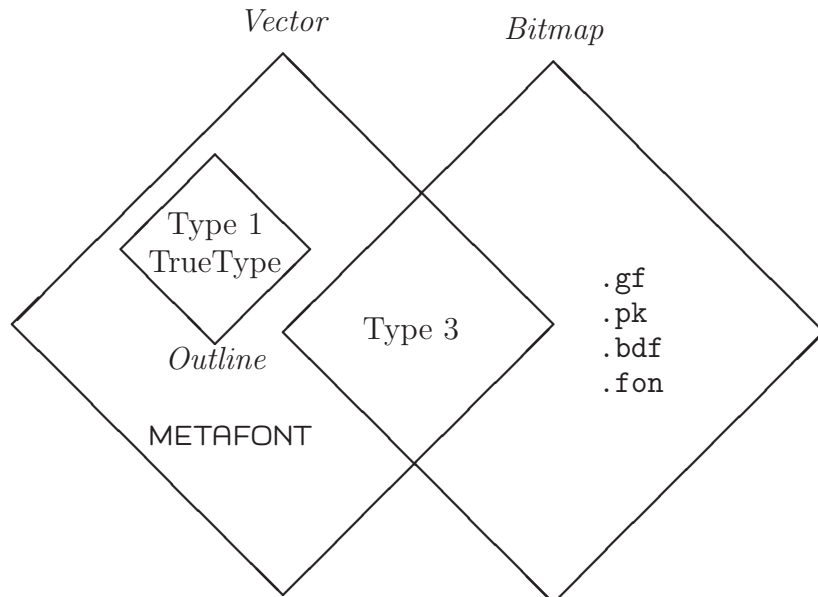


Figure 10: Bitmap and Vector font formats

**Easy and hard conversions** Converting any font format to a bitmap is usually a simple task. Converting between outline formats is harder, functions used to define the outlines differ between formats, and it may become necessary to approximate the outline. This is what `ttf2pt1` does. Conversion from a bitmap to a vector format requires the font to be traced, either by hand or by computer. Hand tracing usually produces good results, computer tracing can be very good, or can be very poor. As Type 3 fonts can contain any postscript command, direct conversion to an outline format is probably impossible. Often the most cost-effective way of making an outline font from a Type 3 font is to trace a high resolution bitmap.

$\text{\TeX}$  itself is not bound to any font format, it does not need to know about the shapes of the glyphs in a font, only their dimensions. This information is stored in a separate file, a `.tfm` file. It is only when the device driver (such as `dvips` or `xdvi`) is used that the actual fonts loaded.

As the fonts generated by METAFONT are bitmaps they cannot be scaled, or printed at a different resolution to that for which they were designed. This creates problems when one wants to share documents created by  $\text{\TeX}$  electronically.



**In defence of bitmaps** Eventually nearly all fonts are converted into bitmaps. Even native PostScript printers will internally convert fonts to a pattern of dots when printing. This process is known as *rasterization*. At low resolutions a hand crafted bitmap font will be far superior than an automatically rasterized font. Both Type 1 and TrueType fonts allow the font designer to give *hints* to the rasterizing program about where to place dots at low resolution. METAFONT has it's own system in which the rasterizing can be tuned to a particular output device. The bitmaps generated by METAFONT will usually be superior to those generated by your printer. And if resolution independence is not important this is how you will get the best output.

Bitmap fonts can also be generated from a variety of other sources, most notably from TrueType fonts by `ttf2pk`

**Using resolution independent fonts** Very often one wishes to generate resolution independent postscript. To do this one need to use scalable PostScript fonts. There are a number of options:

- Use fonts such as Times-Helvetica-Courier throughout your document. These are standard PostScript fonts which are available on all PostScript printers. However this restricts you to a particular system of font families.
- Use Type 1 versions of Computer Modern. Several people have converted the Computer Modern fonts to Type 1, either analytically or by tracing the bitmaps generated by METAFONT.
- Buy a font family. One can purchase high quality fonts from several suppliers—if one is willing to pay the high prices demanded.
- Convert a metafont to PostScript. There are two programs which attempt to do this, `MetaFog` [2] and `mf2pt3` [4]. Both make use of `METAPOST` to make the initial conversion to PostScript.

`MetaFog` is a commercial program that (its maker claims) can produce quality Type 1 fonts from METAFONT sources. This is a hard (in fact insoluble) problem. The conversion is, by necessity, approximate. However the fonts produced in this way are high quality, visually identical to the output of METAFONT and can be used in PDF and by ATM.

On the other hand `mf2pt3` attempts nothing so hard. It produces a Type 3 font, which is scalable, and resolution independent. Figure 12 shows the results of conversion of the METAFONT source to a bitmap at 360dpi, to Type 1 by tracing, and to scalable Type 3.

One problem with `mf2pt3` at the moment are the large PostScript files it generates. Each font will add about 300kB to the size of the PostScript file. Some

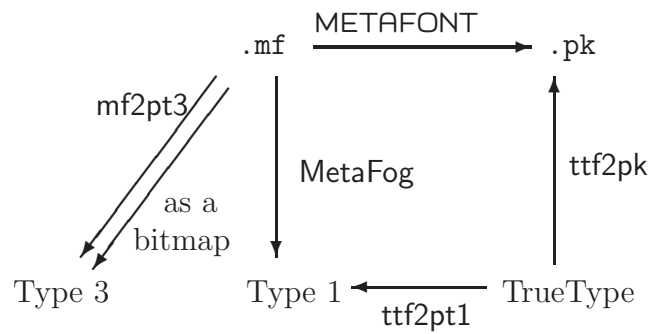
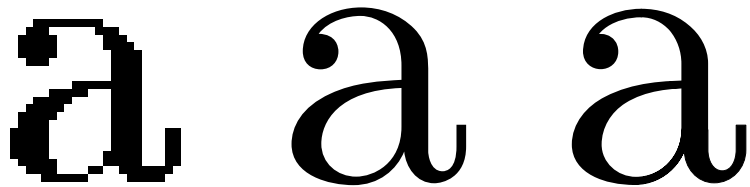


Figure 11: Various conversions that can be done

Figure 12: The Computer Modern 'a'. On the left as a scaled bitmap, in the centre the BlueSky Type 1 conversion, on the right the `mf2pt3` conversion.

form of compression or partial downloading is required to make the PostScript a reasonable size.

## Bibliography

- [1] *The Black Book* available from [http://www.adobe.com/supportservice/devrelations/PDFS/TN/T1\\_SPEC.PDF](http://www.adobe.com/supportservice/devrelations/PDFS/TN/T1_SPEC.PDF).
- [2] Richard Kinch *MetaFog: Converting METAFONT Shapes to Outlines* Paper presented at the 1995 T<sub>E</sub>X Users Group Conference, St. Petersburg, Florida. Available from <http://www.truetex.com>.
- [3] Donald E. Knuth *The METAFONT Book* Addison-Wesley, (1986) ISBN 0201134446.
- [4] Apostolos Syropoulos. *mf2pt3 The Program* from CTAN:fonts/utilities/mf2pt3



# ΕΓΚΑΤΑΣΤΑΣΗ ΝΕΩΝ ΓΡΑΜΜΑΤΟΣΕΙΡΩΝ ΣΤΟ L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

---

ΑΠΟΣΤΟΛΟΣ ΣΥΡΟΠΟΥΛΟΣ<sup>¶</sup> ΚΑΙ ΑΝΤΩΝΗΣ ΤΣΟΛΟΜΥΤΗΣ<sup>§</sup>

<sup>¶</sup>28ης Οκτωβρίου 366  
671 00 Ξάνθη  
E-mail: [apostolo@obelix.ee.duth.gr](mailto:apostolo@obelix.ee.duth.gr)

<sup>§</sup>Πανεπιστήμιο Αιγαίου  
Τμήμα Μαθηματικών  
Καρλόβασι, Σάμος  
E-mail: [atsol@iris.math.aegean.gr](mailto:atsol@iris.math.aegean.gr)

## Abstract

In this article we show how one can enhance his/her T<sub>E</sub>X installation with either Greek TrueType or Greek Type 1 fonts. The reader is provided with all information needed to use these “new” fonts with the `greek` option of the `babel` package. This includes the generation of Font Definition files, virtual fonts, etc.

## 1. Γραμματοσειρές Type 1

Η εγκατάσταση των γραμματοσειρών αυτών γίνεται χρησιμοποιώντας τον μηχανισμό των εν δυνάμει γραμματοσειρών (virtual fonts).

Ας υποθέσουμε ότι έχουμε τη γραμματοσειρά στο αρχείο `font.pfb` και το αρχείο `font.afm` που περιέχει τις μετρικές (τις διαστάσεις) των χαρακτήρων. Θα περιγράψουμε αναλυτικά και με κάποιες διορθώσεις τη διαδικασία που περιγράφεται στο [1]. Αν απλά μας ενδιαφέρει μόνο η αγγλική γλώσσα τα πράγματα είναι απλά. Εκτελούμε τις ακόλουθες εντολές:

```
afm2tfm font.afm -v font7t font7a >> psfonts.map
vptovf font7t font7t.vf font7t.tfm
```

Το font7a που βάλουμε στην πρώτη εντολή είναι απλά ένα ψευδώνυμο (alias) του αρχείου font.pfb. Αν δούμε το αρχείο psfonts.map θα περιέχει μία γραμμή της μορφής:

```
font7a FontName
```

όπου το FontName είναι το εσωτερικό όνομα της γραμματοσειράς, π.χ., AvantGarde-Book. Μεταβάλλουμε το αρχείο psfonts.map κατάλληλα:

```
font7a FontName <font.pfb
```

Δηλαδή καθοδηγούμε τον οδηγό dvips ότι όταν το T<sub>E</sub>X ζητάει την γραμματοσειρά font7a, να συμπεριλαμβάνει στο τελικό αρχείο PostScript το αρχείο font.pfb. Θα μπορούσαμε να χρησιμοποιήσουμε κατευθείαν το όνομα font αντί του font7a που χρησιμοποιήσαμε στο τελευταίο όρισμα της πρώτης εντολής αλλά όπως θα δούμε παρακάτω αυτό δημιουργεί προβλήματα αφού είναι δυνατόν από ένα pfb αρχείο να παράγουμε γραμματοσειρές με διαφορετικά χαρακτηριστικά (όπως κεκλιμένη γραφή, πεζοκεφαλαία, κ.λπ.): οπότε και αλλάζουμε το όνομα font7a σε, π.χ., fonto7a για την κεκλιμένη γραφή, ενώ τροφοδοτούμε αυτό το ψευδώνυμο με το ίδιο αρχείο γραμματοσειράς, το font.pfb.

Επιπλέον δημιουργούμε ένα αρχείο με όνομα Fontmap στο οποίο προσθέτουμε την παρακάτω αράδα:

```
/FontName (font.pfb) ;%
```

Τώρα έχουμε όλα τα απαραίτητα αρχεία για να χρησιμοποιήσουμε τη γραμματοσειρά με το plain T<sub>E</sub>X. Για να χρησιμοποιηθεί με το L<sup>A</sup>T<sub>E</sub>X θα πρέπει να δημιουργήσουμε τους κατάλληλους πίνακες που θα επιτρέπουν στο NFSS (βλέπε [1, 2, 3, 4]) ώστε να διαλέγει σωστά μέγεθος και μορφή. Για να το πετύχουμε αυτό δημιουργούμε ένα αρχείο ορισμού γραμματοσειράς με όνομα OT1xxx.fd, όπου το xxx είναι θέμα δικής μας επιλογής. Το αρχείο αυτό θα πρέπει να περιέχει τουλάχιστον τα εξής:

```
\DeclareFontFamily{OT1}{xxx}{}
\DeclareFontShape{OT1}{xxx}{m}{n}{<-> font7t}{}
\endinput
```

(Είναι σχεδόν πάντα απαραίτητο να υπάρχουν και ορισμοί αντικατάστασης γραμματοσειράς, αλλά για λόγους απλότητας παραλείπουμε το σκέλος αυτό.) Έτσι αν θέλουμε να καλέσουμε τοπικά σ' ένα αρχείο L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> που ετοιμάζουμε αυτή την γραμματοσειρά, πρέπει να ορίσουμε μία εντολή που θα καλεί την γραμματοσειρά. Για παράδειγμα βάζοντας στον πρόλογο του αρχείου την εντολή:

```
\newcommand{\myfont}[1]{\usefont{OT1}{xxx}{m}{n} #1\relax}
```

ενώ στο αρχείο μας όταν θέλουμε να καλέσουμε αυτή τη γραμματοσειρά, απλά δηλώνουμε:

```
\myfont{This is test}
```

Εναλλακτικά αν θέλουμε όλο το κείμενό μας να στοιχειοθετηθεί με αυτή τη γραμματοσειρά αντί για τα παραπάνω γράφουμε στον πρόλογο του αρχείου μας την εντολή

```
\renewcommand{\rmdefault}{xxx}
```

Όμως με αυτό τον τρόπο έχουμε στη διάθεσή μας χαρακτήρες που αρκούν μόνο για αγγλικό κείμενο. Χαρακτήρες όπως οι  $\mathcal{E}$ ,  $\mathfrak{a}$ , κ.λπ., που απαιτούνται για άλλες λατινογενείς γλώσσες δεν είναι διαθέσιμοι. Αν κανείς θέλει να έχει πρόσβαση και σε αυτούς τους χαρακτήρες (εάν αυτοί είναι διαθέσιμοι στο font.pfb) θα πρέπει εκτός του να βάλει στον πρόλογο αρχείου L<sup>A</sup>T<sub>E</sub>X<sub>2</sub> $\epsilon$  που δημιουργεί την εντολή

```
\usepackage[T1]{fontenc}
```

να παράγει εν δυνάμει γραμματοσειρές που περιέχουν πληροφορίες για τη θέση των χαρακτήρων στη γραμματοσειρά font.pfb. Αυτό αλλιώς λέγεται *ανακωδικοποίηση*. Κάθε σύγχρονη εγκατάσταση του T<sub>E</sub>X περιέχει το αρχείο EC.enc. Αντιγράφουμε αυτό το αρχείο στον κατάλογο που εργαζόμαστε και εκτελούμε τις εντολές:

```
afm2tfm font.afm -T EC.enc -v font8t font8a >> psfonts.map
vptovf font8t font8t.vf font8t.tfm
```

Ανοίγουμε το αρχείο psfonts.map. Θα παριέχει μία ακόμα αράδα:

```
font8a FontName " EC ReEncodeFont " <EC.enc
```

Την οποία τροποποιούμε σε

```
font8a FontName " EC.enc ReEncodeFont " <EC.enc <font.pfb
```

Το αρχείο Fontmap δεν χρειάζεται παραπέρα τροποποίηση. Εδώ είναι σαφής η ανάγκη να έχουμε τα διαφορετικά ονόματα font7a και font8a, αφού αλλιώς δεν θα μπορούσε ο οδηγός dvips που διαβάζει το αρχείο psfonts.map να αποφασίσει αν θα ανακωδικοποιήσει τη γραμματοσειρά ή όχι.

Από πλευράς γραμματοσειράς είμαστε έτοιμοι και για την κωδικοποίηση T1 αλλά χρειάζεται βέβαια και ένα αρχείο ορισμού γραμματοσειράς που τώρα θα πρέπει να λέγεται T1xxx.fd. Αυτό το αρχείο θα πρέπει να περιλαμβάνει τα εξής:

```
\DeclareFontFamily{T1}{xxx}{}
\DeclareFontShape{T1}{xxx}{m}{n}{<-> font8t}{}
\endinput
```

Αυτή η διαδικασία καλύπτει πλήρως τις περισσότερες γραμματοσειρές που διανέμουν διάφορες εταιρείες. Θα δούμε αμέσως τώρα πως να φτιάξουμε παραλλαγές μιάς γραμματοσειράς από το font.pfb όπως κεκλιμένη γραφή, καπιταλάκια, κ.ά.

Αν αναζητήσει κανείς σε μία σύγχρονη εγκατάσταση του T<sub>E</sub>X τη γραμματοσειρά Bookman θα δει ότι υπάρχουν μόνο 4 αρχεία (τα ubkd8a.pfb, ubkdi8a.pfb, ubkl8a.pfb, ubkli8a.pfb). Παρ' όλα αυτά δουλεύουν σωστά οι εντολές

```
\normalfont, \itshape, \slshape, \bfseries, \bfseries\itshape,
\bfseries\scshape, \scshape
```

δηλαδή 7 διαφορετικές σειρές/μορφές. Αυτό επιτυγχάνεται παράγοντας τις κεκλιμένες (slanted) και τα καπιταλάκια (SmallCaps) τεχνητά. Θα παρουσιάσουμε τη διαδικασία για το font.pfb και τη κωδικοποίηση OT1 (για τη κωδικοποίηση T1 τα πράγματα είναι ανάλογα).

Για να πάρουμε κεκλιμένη γραμματοσειρά εκτελούμε τις εντολές:

```
afm2tfm font.afm -s 0.167 -v fonto7t fonto7a >> psfonts.map
vptovf fonto7t fonto7t.vf fonto7t.tfm
```

ενώ προσθέτουμε στο τέλος της νέας αράδας του psfonts.map το γνωστό <font.pfb (το Fontmap πάλι δεν θέλει καμία τροποποίηση). Η παράμετρος 0.167 είναι η κλίση που ζητάμε για την κεκλιμένη μορφή. Ενώ είναι η συνήθης τιμή τίποτα δεν μας εμποδίζει να χρησιμοποιήσουμε οποιοδήποτε άλλο αριθμό στο διάστημα (-1, 1). Αν διαλέξουμε αρνητική τιμή η γραμματοσειρά θα έχει κλίση στα αριστερά! Βέβαια στο αρχείο OT1xxx.fd θα πρέπει να προσθέσουμε τη γραμμή

```
\DeclareFontShape{OT1}{xxx}{m}{sl}{<-> fonto7t}{}>
```



Τέλος αν στο L<sup>A</sup>T<sub>E</sub>X<sub>2</sub>ε αρχείο μας έχουμε δηλώσει

```
\renewcommand{\rmdefault}{xxx}
```

τότε η συνήθης εντολή για αλλαγή σε κεκλιμένη γραφή θα δουλέψει σωστά.

Ανάλογη είναι η διαδικασία για καπιταλάκια με λίγες τροποποιήσεις. Οι εντολές είναι:

```
afm2tfm font.afm -V fontc7t fontc7a >> psfonts.map
vptovf fontc7t fontc7t.vf fontc7t.tfm
```

Προσέξτε ότι στην πρώτη εντολή άλλαξε το -v σε -V. Προσθέτουμε το <font.pfb στη νέα ομάδα του psfonts.map αφήνουμε το Fontmap ως έχει και προσθέτουμε στο OT1xxx.fd τη γραμμή:

```
\DeclareFontShape{OT1}{xxx}{m}{sc}{<-> fontc7t}{}
```

Υπάρχει ακόμα μία ενδιαφέρουσα παραλλαγή. Αυτή είναι η παράμετρος -e στο πρόγραμμα afm2tfm για την παραγωγή των λεγομένων «πεπλατυσμένων» (extended) και «εκλεπτυσμένων» (condensed) γραμματοσειρών. Σαν παράδειγμα δίνουμε τις δύο εντολές:

```
afm2tfm font.afm -e 1.2 -v fonte7t fonte7a >> psfonts.map
vptovf fonte7t fonte7t.vf fonte7t.tfm
```

Αυτό παράγει αρχεία για την χρήση της γραμματοσειράς πεπλατυσμένης κατά τον παράγοντα 1,2. Παράμετρος μικρότερη του 1 παράγει μικρότερου πλάτους (εκλεπτυσμένη) γραμματοσειρά.

Κλείνουμε αυτό το κεφάλαιο με το να πούμε ότι αν θέλουμε να εγκαταστήσουμε την γραμματοσειρά στο L<sup>A</sup>T<sub>E</sub>X<sub>2</sub>ε σε μόνιμη βάση απλά μεταφέρουμε όλα τα αρχεία που έχουν παραχθεί κατά την διαδικασία αυτή (εκτός των .vpl) στο δέντρο του T<sub>E</sub>X. Βάζουμε τα .pfb σε υποκατάλογο του texmf/fonts/type1/ τα .vf σε υποκατάλογο του texmf/fonts/vf/, τα .tfm σε υποκατάλογο του texmf/fonts/tfm/ και τα .fd στο texmf/tex/generic/babel/. Τέλος, αντιγράφουμε το περιεχόμενο του Fontmap στο αρχείο Fontmap του συστήματος (συνήθως στο ghostscript/version/Fontmap) και το περιεχόμενο του psfonts.map στο αρχείο psfonts.map του συστήματος (συνήθως στο texmf/dvips/base/psfonts.map) και η εγκατάσταση είναι πλήρης (τα αρχεία .vpl δεν χρειάζονται πια και μπορούν να διαγραφούν).

Τα πράγματα είναι λίγο διαφορετικά όταν κανείς θέλει να δουλέψει με ελληνικά. Περιγράφουμε τη μέθοδο στην επόμενη ενότητα.

### 1.1. Ελληνικές Type 1 Γραμματοσειρές

Για να δουλέψουμε με μία ελληνική γραμματοσειρά πρέπει πρώτα απ' όλα το αρχείο της γραμματοσειράς να περιέχει τους ελληνικούς χαρακτήρες στις σωστές θέσεις. Αν έχουμε κάνει τις ενέργειες που περιγράφονται στην προηγούμενη ενότητα για το αγγλικό μέρος της γραμματοσειράς και τη κωδικοποίηση T1 (με τη βοήθεια του αρχείου `EC.enc`) τότε είναι εύκολο να ελέγξουμε κάτι τέτοιο. Απλά τρέχουμε την εντολή

```
tex testfont
```

και στην προτροπή του TeX δίνουμε το όνομα του αρχείου δηλαδή το `font8a`. Έτσι θα παραχθεί το αρχείο `testfont.dvi`. Αν ανοίξουμε αυτό το αρχείο με ένα πρόγραμμα όπως το `xdvi` θα δούμε έναν πίνακα της γραμματοσειράς. Αν περιέχονται σε αυτόν οι περισσότεροι ελληνικοί χαρακτήρες τότε μπορούμε να προχωρήσουμε με τη μέθοδο A (δες παρακάτω). Αυτό που μόλις περιγράψαμε έχει νόημα (σε σχέση με το `EC.enc`) γιατί οι «σωστές» θέσεις για τα ελληνικά γράμματα καταλαμβάνουν τις θέσεις των επιπλέον χαρακτήρων για τις εν γένει λατινογενείς γλώσσες. Άρα αν η γραμματοσειρά μας έχει ελληνικά σε σωστές θέσεις θα πρέπει να εμφανιστούν στον πίνακα του `font8a`.

Αν αυτό δεν συμβεί, είτε η γραμματοσειρά μας δεν περιέχει ελληνικά είτε τα περιέχει σε άλλες θέσεις (μπορεί να είναι Unicode γραμματοσειρά) οπότε θα ακολουθήσουμε τη μέθοδο B.

Η διαφορά ανάμεσα στις δύο μεθόδους είναι το αρχείο `.enc` που χρησιμοποιούμε κάθε φορά και πρακτικά δεν διαφέρει από την δουλειά που κάναμε με το `EC.enc`. Τα αρχεία `TeXLGREncoding.enc` για τη μέθοδο A και `TeXLGRUEncoding.enc` για τη μέθοδο B αναλαμβάνουν να λύσουν το πρόβλημα του τονισμού καταγράφοντας στο αρχείο των εν δυνάμει γραμματοσειρών τα απαιτούμενα πολλαπλά στοιχεία (ligatures). Παραπάνω δουλειά χρειάζεται μόνο για την τεχνητή δημιουργία πεζοκεφαλαίων μιά και το κεφαλαίο αντίστοιχο των `σ` και `ς` θα πρέπει να είναι και στις δύο περιπτώσεις το `Σ` ενώ ανάλογο πρόβλημα υπάρχει και με το `χ`. Πριν προχωρήσουμε στην αναλυτική παρουσίαση των δύο μεθόδων, αξίζει να σημειωθεί ότι τα δύο αρχεία πρέπει να τοποθετηθούν στον κατάλογο `texmf/dvips/base/` του δένδρου της υλοποίησης του TeX.

#### Μέθοδος A (Τα ελληνικά εμφανίστηκαν στον πίνακα του `font8a`)

Με τη βοήθεια του αρχείου `TeXLGREncoding.enc` που διανέμεται μαζί με το πακέτο `TeXgreek` εκτελούμε τις εντολές:

```
afm2tfm font.afm -T TeXLGREncoding.enc -v gfont7t \
```

```
gfont7a >> psfonts.map
vptovf gfont7t gfont7t.vf gfont7t.tfm
```

(Το σύμβολο \ σημαίνει απλά ότι η εντολή δε χωράει σε μία γραμμή και έτσι την γράφουμε σε δύο.) Βάλαμε το γράμμα g πριν από το όνομα των αρχείων για να δηλώσουμε απλά ότι πρόκειται για ελληνική γραμματοσειρά (το αρχείο που περιέχει τη γραμματοσειρά παραμένει το font.pfb). Τροποποιούμε κατάλληλα το αρχείο psfonts.map προσθέτοντας κατά τα γνωστά στη νέα του αράδα το <font.pfb, ενώ και πάλι το αρχείο Fontmap δεν θέλει καμία τροποποίηση. Τέλος φτιάχνουμε το αρχείο ορισμού γραμματοσειράς με όνομα lgrxxx.fd και περιεχόμενο το:

```
\DeclareFontShape{LGR}{xxx}{m}{n}{<-> gfont7t}{}
```

(προσέξτε το LGR στο αρχείο να είναι γραμμένο με κεφαλαία.) Πάλι υπάρχουν δύο τρόποι για χρήση της γραμματοσειράς. Αν θέλουμε τοπική χρήση στο αρχείο μας γράφουμε στον πρόλογο του κώδικά μας τα εξής:

```
\documentclass{article}
\usepackage[greek]{babel}
\usepackage[iso-8859-7]{inputenc}
\newcommand{\myfont}[1]{\usefont{LGR}{xxx}{m}{n} #1\relax}
```

και στο σώμα του αρχείου μας γράφουμε ελληνικά με αυτή την γραμματοσειρά με την εντολή \myfont.

Για ολική αλλαγή ώστε τα ελληνικά και τα αγγλικά να παράγονται με το font.pfb γράφουμε:

```
\documentclass{article}
\usepackage[greek]{babel}
\usepackage[iso-8859-7]{inputenc}
\renewcommand{\rmdefault}{xxx}
```

**Μέθοδος Β** (Τα ελληνικά δεν εμφανίστηκαν στον πίνακα του font8a αλλά είμαστε σίγουροι ότι η γραμματοσειρά μας περιέχει ελληνικούς χαρακτήρες)

Κάνουμε ακριβώς την ίδια διαδικασία με πιο πριν αλλά τώρα χρησιμοποιούμε το αρχείο TeXLGRUEncoding.enc αντί του TeXLGREncoding.enc.

Η παραγωγή γυρτής γραφής γίνεται με τον ίδιο τρόπο που έγινε και με τα αγγλικά, δηλαδή προσθέτοντας την επιλογή -s 0.167 στην εκτέλεση του προγράμματος afm2tfm. Ομοίως για τις πεπλατυσμένες και τις εκλεπτυσμένες παραλλαγές.

Εδώ πρέπει να κάνουμε δύο παρατηρήσεις.

**Παρατήρηση 1.** Αν η γραμματοσειρά μας προέρχεται ή προορίζεται από/για τον κόσμο των Windows της εταιρείας Microsoft είναι πολύ πιθανό να πρέπει να κάνουμε μία αλλαγή στο αρχείο `TeXLGREncoding.enc`. Θα πρέπει να αλλάξει η δήλωση `/cent` σε `/paragraph`.

**Παρατήρηση 2.** Για να είναι εφικτός ο τονισμός υπάρχει ένας μηχανισμός πολλαπλών στοιχείων (ligatures) που ορίζονται στο αρχείο `TeXLGRUEncoding.enc` και στο `TeXLGRUEncoding.enc`. Αυτές τις δηλώσεις τις περνάει το πρόγραμμα `afm2tfm` στο αρχείο `font7t.vpl` και με τη σειρά του το `vptovf` στο σχετικό αρχείο `.tfm`. Έχει παρατηρηθεί ότι σε μερικές περιπτώσεις το `afm2tfm` δεν τα καταφέρνει να γράψει τα πολλαπλά στοιχεία στο αρχείο `.vpl`. Έτσι λοιπόν αν δεν δουλεύει σωστά ο τονισμός ανοίγουμε σε έναν επεξεργαστή κειμένου το αρχείο `.vpl` και αντικαθιστούμε το «LIGTABLE» με το περιεχόμενο του αρχείου `ligtable.vpl` που διανέμεται με το πακέτο `TeXgreek`. Τέλος ξανατρέχουμε το πρόγραμμα `vptovf`.

## 1.2. Πεζοκεφαλαία Γραφή

Εδώ η διαδικασία είναι ανάλογη με τη διαδικασία για τα αγγλικά. Δηλαδή δίνουμε τις εντολές:

```
afm2tfm font.afm -T TeXLGRUEncoding.enc -V gfontc7t \
gfontc7a >> psfonts.map
vptovf gfontc7t gfontc7t.vf gfontc7t.tfm
```

Σημειώστε ότι σε περίπτωση που χρησιμοποιούμε το διακόπτη `-e 1.25` σε συνδυασμό με τον `-V`, προσεγγίζουμε καλύτερα τα πραγματικά πεζοκεφαλαία. Υπάρχουν μόνο δύο διορθώσεις που πρέπει να γίνουν στο αρχείο `.vpl` πριν εκτελεστεί η δεύτερη εντολή `vptovf`. Ανοίγουμε το αρχείο `gfontc7t.vpl` με έναν επεξεργαστή κειμένου καθώς και το αρχείο `gfont7t.vpl`. Στο `gfontc7t.vpl` πηγαίνουμε στη θέση που γράφει:

```
(CHARACTER q
  (CHARWD R 496)
  (CHARHT R 447)
  (CHARDP R 208)
)
```

όπου βέβαια οι αριθμοί 496, 447 και 208 είναι (εδώ) ενδεικτικοί. Αυτές τις γραμμές με τους αριθμούς τις σβήνουμε και τις αντικαθιστούμε με τις γραμμές που βρίσκουμε στο αρχείο `gfont7t.vpl` στη θέση `CHARACTER C Q` πολλαπλασιασμένους με 0,8. Πριν κλείσει και η τελευταία παρένθεση του `CHARACTER C q` παρεμβάλλουμε τα εξής:

```
(MAP
  (SELECTFONT D 1)
  (SETCHAR O 121)
)
```

όπου ο χαρακτήρας μετά το `SETCHAR` είναι το κεφαλαίο αγγλικό `o` και όχι το μηδέν. Ο παράγοντας 0,8 προέρχεται από το ότι η σμίκρυνση που γίνεται για την παραγωγή των πεζοκεφαλαίων είναι στο 80%. Αυτό μπορούμε να το αλλάξουμε αν θέλουμε με την παράμετρο `-c`:

```
afm2tfm font.afm -T TeXLGREncoding.enc -c 0.75 \
-V gfontc7t gfontc7t
```

Μετά πηγαίνουμε στο `gfontc7t.vpl` στη θέση

```
(CHARACTER C c
  (CHARWD R 496)
  (CHARHT R 447)
  (CHARDP R 208)
)
```

και αλλάζουμε τις γραμμές με τους αριθμούς στις γραμμές με τους αριθμούς από το αρχείο `gfont7t.vpl` στη θέση `CHARACTER C S`, ενώ πριν κλείσει η τελευταία παρένθεση για τον χαρακτήρα `c` παρεμβάλλουμε τα εξής:

```
(MAP
  (SELECTFONT D 1)
  (SETCHAR O 123)
)
```

Τώρα τρέχουμε το πρόγραμμα `vptovf` και μετά την κατάλληλη τροποποίηση των `psfonts.map` και `LGRxxx.fd` μπορούμε να χρησιμοποιήσουμε την `\sc`. Είναι πιθανό ο χαρακτήρας `X` να μην εμφανίζεται στα κεφαλαία με σωστή απόσταση δεξιά του (δοκιμάστε τη λέξη `\sc Αχορταγος`). Αν συμβαίνει αυτό μειώστε την μεταβλητή `CHARWD` στο αρχείο `gfontc7t.vpl` στον χαρακτήρα `CHARACTER C q` και ξαναδοκιμάστε.

### 1.3. Ρυθμίσεις απόστασης ειδικών ζευγών (Kerning)

Πολλές φορές κάποιοι χαρακτήρες λόγω της γεωμετρίας τους για να φαίνεται ότι είναι σε σωστή απόσταση μεταξύ τους πρέπει να έρθουν πιο κοντά. Αυτό βέβαια δεν σημαίνει ότι θα πρέπει να μειώσουμε την τιμή της μεταβλητής CHARWD γιατί αυτή θα επηρεάσει την απόσταση του χαρακτήρα από όλους τους άλλους ενώ εμείς θέλουμε να το κάνουμε για συγκεκριμένους χαρακτήρες. Για να γίνει αντιληπτό τι εννοούμε κοιτάξτε τη λέξη

AVIATION

και την

AVIATION.

Λόγω της γεωμετρίας των A και V το ανθρώπινο μάτι θα τους ήθελε πιο κοντά (όπως είναι στην πρώτη φορά) και όχι με την συνήθη απόστασή τους (όπως εμφανίζονται την δεύτερη φορά). Παράδειγμα στα ελληνικά είναι η λέξη

ΠΥΛΗ και ΠΥΛΗ.

Το  $\TeX$  —ως πρόγραμμα στοιχειοθεσίας— μπορεί αυτόματα να κάνει τέτοιες ρυθμίσεις αρκεί να έχουμε ενημερώσει τα αρχεία των γραμματοσειρών ποιά είναι εκείνα τα ζεύγη των οποίων η απόσταση αλλάζει όταν εμφανίζονται διαδοχικά και κατά πόσο θέλουμε να έρθουν πιο κοντά. Αυτό το τελευταίο, δηλαδή το ποσοτικό πρόβλημα, είναι κάτι με το οποίο κανείς πρέπει να πειραματιστεί γιατί έχει να κάνει και με τον σχεδιασμό (την γεωμετρία) της εκαστοτε γραμματοσειράς. Αναφέρουμε εδώ ότι για τα ελληνικά κάνουμε τέτοιες ρυθμίσεις για τα ζευγάρια ΑΥ, ΔΥ, ΛΥ, ΥΑ, ΥΔ, ΥΛ, ΑΥ, ΥΑ, ΥΔ, ΥΛ, ΑΥ, ΥΑ, ΥΔ, ΥΛ, ΑΥ, ΥΑ και ΥΔ.

Εδώ θα περιγράψουμε το πώς γίνεται αυτό. Ανοίγουμε το .vpl αρχείο που προέκυψε από την εντολή afm2tfm και πηγαίνουμε στο τέλος του LIGTABLE και πρίν κλείσει η τελευταία παρένθεση αυτού του πίνακα προσθέτουμε τις γραμμές:

```
(LABEL C U)
(KRN C A R -64)
(KRN C D R -64)
(KRN C L R -64)
(STOP)
```

Εδώ λέμε ότι όταν ο χαρακτήρας Υ ακολουθείται από Α ή Δ ή Λ τότε ο δεύτερος χαρακτήρας θα πρέπει να πλησιάσει τον Υ κατά 64 μονάδες. Παρατηρήστε ότι η αντιστοιχία των γραμμάτων στο LIGTABLE είναι αυτή του babel (δες [3]). Ομοίως κάνουμε επεμβάσεις για τα υπόλοιπα ζεύγη και μετά τρέχουμε το πρόγραμμα vptovf.

Στα αγγλικά τώρα τέτοιες επεμβάσεις γίνονται συνήθως για τα ζεύγη AV, AW, AY, VA, WA, YA ή και αλλού αν κρίνεται απαραίτητο.

**Παρατήρηση.** Σημειώστε εδώ ότι οι χαρακτήρες Ά, Έ και Ύ δεν υπάρχουν στην αντιστοιχία του babel γιατί παράγονται με τον μηχανισμό των συνδέσμων. Έτσι για την παραπάνω εργασία για αυτούς τους χαρακτήρες χρησιμοποιούμε την οκταδική τους αναπαράσταση. Το Ά είναι ο χαρακτήρας στη θέση 200 (θέση του babel και όχι του font.pfb). Το Έ στη θέση 314 και το Ύ στη θέση 337 (τρέξτε latex nfssfont.tex στο gfont7t και τυπώστε το αποτέλεσμα για να δείτε όλες τις θέσεις των χαρακτήρων (δες [4])). Μέ βάση αυτό ο παραπάνω μηχανισμός για το Ά γίνεται:

```
(LABEL 0 200)
(KRN C L R -64)
(KRN C Y R -64)
(STOP)
```

## 2. Γραμματοσειρές TrueType

Για τις γραμματοσειρές αυτές είτε τις μετατρέπουμε σε METAFONT (δες [2]) είτε τις μετατρέπουμε σε Type 1 με το πρόγραμμα ttf2pt1 ή σε Type 3 με το πρόγραμμα ttf2pfa. Το δεύτερο έχει το πλεονέκτημα ότι παράγει και το .afm αρχείο. Και τα δύο προγράμματα είναι διαθέσιμα στο

<http://www.netspace.net.au/~mheath/ttf2pt1/>

Κλείνουμε με την παρατήρηση ότι όταν δεν έχουμε το αρχείο .afm ενώ έχουμε το .pfb μπορούμε να παράγουμε το .afm με το πρόγραμμα getafm και την εντολή

```
getafm font.pfb | gsnd - > font.afm
```

Έχει αναφερθεί όμως ότι αυτό δεν δουλεύει με κάποιες πρόσφατες εκδόσεις του ghostscript. Σημειώστε ότι αυτός είναι ουσιαστικά ο λόγος του «πειράζουμε» το αρχείο Fontmap.

## Βιβλιογραφία

- [1] Α. Συρόπουλος, Ν. Σωφρονίου, Δ. Φιλίππου, *TEX και ψηφιακή τυπογραφία*, Εκδόσεις Ζήτη, (1998).
- [2] Α. Συρόπουλος, Α. Τσολομούτης, *Γραμματοσειρές True Type και L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*, Το Εύτυπον, **2**, (1999), 17–22.
- [3] Α. Συρόπουλος, *L<sup>A</sup>T<sub>E</sub>X*, Εκδόσεις Παρατηρητής, (1998).
- [4] M. Goossens, F. Mittelbach, A. Samarin, *The L<sup>A</sup>T<sub>E</sub>X companion*, Addison-Wesley Co., (1994).



## Το L<sup>A</sup>T<sub>E</sub>X και άλλοι κειμενογράφοι

---

Φώτης Ανδριτσόπουλος

Αρχιμήδους 34

123 51 Αθήνα

e-mail: [fandrit@telecom.ntua.gr](mailto:fandrit@telecom.ntua.gr)

### Abstract

The preparation of my thesis for the degree of Diploma in Electrical Engineering, lead me to adopt L<sup>A</sup>T<sub>E</sub>X over common word processors. In this short note I report what lead me to this decision and how I managed to become an average L<sup>A</sup>T<sub>E</sub>X user in less than a week.

Όλοι όσοι έχουν ασχοληθεί με υπολογιστές, από τους προγραμματιστές μέχρι τους απλούς χρήστες, έχουν έρθει κάποια στιγμή στην ανάγκη να γράψουν ένα κείμενο, είτε αυτό είναι επίσημο είτε πρόχειρο. Μία μερίδα των χρηστών αυτών χρησιμοποιούν τους υπολογιστές αποκλειστικά και μόνο για την συγγραφή κειμένων. Κατά την περίοδο των φοιτητικών μου χρόνων ήταν αρκετές οι φορές που χρειάστηκα κι εγώ να δακτυλογραφήσω κείμενα για διάφορες εργασίες που μου ανατέθηκαν από τη σχολή αλλά ακόμα και για προσωπική χρήση.

Την ύπαρξη του L<sup>A</sup>T<sub>E</sub>X την έμαθα πριν από τρία περίπου χρόνια από έναν ειδικό του είδους θα έλεγα, χωρίς όμως να το έχω χρησιμοποιήσει ποτέ. Έτσι είδα τις δυνατότητές του και εντυπωσιάστικα κυρίως από την ποιότητα του αποτελέσματος που παρείχε. Επειδή όμως δεν είναι ένα περιβάλλον WYSIWYG<sup>1</sup>, φαντάστηκα πως για να γράψει κανείς κείμενα στο L<sup>A</sup>T<sub>E</sub>X ήταν κάτι το τρομακτικά δύσκολο αλλά ταυτόχρονα και πολύ χρονοβόρο, (συγκρίνοντας πάντα με WYSIWYG κειμενογράφους). Η ποιότητα που έπαιρνα από άλλους κειμενογράφους όπως για παράδειγμα από το Microsoft Word ήταν κάτι το ανεκτό αλλά όχι και ικανοποιητικό.

Όταν ξεκίνησα να γράφω την διπλωματική μου εργασία, δακτυλογράφησα τα αρχικά μου κείμενα στο Microsoft Word. Όμως από την αρχή κιόλας πρόσεξα πως οι πολλοί μαθηματικοί τύποι που έπρεπε να γράψω και με δυσκόλευαν αρκετά στην διατύπωσή τους αλλά και κάποια προβλήματα όπως η στοίχιση των σχέσεων και προπάντως η ποιότητα του συνολικού αποτελέσματος γινόταν όλο και

---

<sup>1</sup> What you see is what you get

πιο αντιαισθητική. Ταυτόχρονα αντιμετώπιζα και το πρόβλημα ότι δεν μπορούσα να γράφω κομμάτια των κειμένων μου σε διαφορετικούς υπολογιστές. Όταν ο υπολογιστής στο εργαστήριο είχε την έκδοση του WYSIWYG κειμενογράφου A, στο σπίτι ο υπολογιστής μου την έκδοση B και ο φορητός πιθανόν καμμία από αυτές δημιουργούνταν τρομακτικά προβλήματα ασυμβατότητας μεταξύ των εγγράφων όταν δακτυλογραφούσα ένα μέρος στον έναν και το υπόλοιπο σε κάποιον άλλον. Παρά ταύτα, παρουσιάζοντας κάποια στιγμή ένα μέρος από τα κείμενα που είχα γράψει μέχρι τότε, στον καθηγητή με τον οποίο εκπονούσα την διπλωματική αυτή εργασία, μου είπε πως το περιεχόμενο ήταν καλό αλλά σαν εμφάνιση ήταν πολύ κακό! Θα μπορούσε κάποιος να πει ότι η εμφάνιση δεν θα τον ενδιέφερε και τόσο. Η προσωπική μου γνώμη πάνω σε αυτό το θέμα είναι πως από τη στιγμή που ξεκινάμε να γράψουμε ένα κείμενο στον υπολογιστή (και όχι χειρόγραφα σε ένα κομμάτι χαρτί) το κάνουμε για να πάρουμε ένα πραγματικά ποιοτικό αποτέλεσμα.

Τότε ήταν η στιγμή που ξεκίνησα να μάθω το  $\text{\LaTeX}$  φοβούμενος όμως πως θα απαιτήσει πολύ χρόνο τη στιγμή που πιεζόμουν χρονικά για την συγγραφή του κειμένου. Τώρα είμαι σίγουρος πως αν δοκίμαζα να γράψω το κείμενο αυτό σε έναν WYSIWYG κειμενογράφο θα σπαταλούσα πολύ περισσότερο χρόνο από αυτόν που ξόδεψα για το  $\text{\LaTeX}$ . Με τη βοήθεια ενός βιβλίου<sup>2</sup> κατάφερα μέσα σε μία εβδομάδα να γράψω ακόμα και τις πιο δύσκολες μαθηματικές σχέσεις και προπάντων χωρίς να ενδιαφέρομαι για την στοίχιση του κειμένου, τον συλλαβισμό των λέξεων κ.λπ., εφ'όσον αυτό το αναλαμβάνει το  $\text{\LaTeX}$ . Επίσης είχε λυθεί και το πρόβλημα της ασυμβατότητας μεταξύ των κειμενογράφων, εφόσον για τη συγγραφή κειμένων του  $\text{\LaTeX}$  χρειαζόμουν έναν οποιονδήποτε κειμενογράφο απλού κειμένου (plain text editor) όπως ο vi του UNIX, το Edit του DOS ή ακόμα και το γνωστό στους περισσότερους Notepad των Windows. Ο καθηγητής έμεινε έκπληκτος όταν είδε το αποτέλεσμα των κειμένων μου που είχα γράψει με το  $\text{\LaTeX}$ . Ο ίδιος χρησιμοποιούσε τον κειμενογράφο WordPerfect τότε παρά το γεγονός ότι είχε ακουστά την ύπαρξη του  $\text{\LaTeX}$ . Πίστευε λανθασμένα και αυτός (όπως και πολλοί άλλοι) πως είναι ένα περιβάλλον στριφνό. Του έκανε τόσο μεγάλη εντύπωση η ποιότητα του αποτελέσματος που με ρώτησε αν θα μπορούσε να αρχίσει να το χρησιμοποιεί και αυτός στο PC του (οπώς και κάνει τώρα!). Η αμφιβολία του αυτή προήλθε από έναν μύθο που πολλοί πιστεύουν ότι το  $\text{\LaTeX}$  μπορεί να λειτουργήσει μόνο σε περιβάλλον τύπου UNIX. Ένα ακόμα πλεονέκτημα του  $\text{\LaTeX}$  στο οποίο πολλοί δεν δίνουν ιδιαίτερη σημασία, είναι ότι το  $\text{\LaTeX}$  είναι δωρεάν! Σε συνδυασμό λοιπόν με ένα δωρεάν λειτουργικό σύστημα (Linux, FreeBSD, Solaris x86 κ.λπ.) μπορούμε να έχουμε ένα άψογο σύστημα στοιχειοθεσίας χωρίς να ξοδέψουμε ούτε μία δραχμή!

<sup>2</sup>  $\text{\LaTeX}$ , Απόστολος Συρόπουλος, Εκδόσεις Παρατηρητής, 1998.

## ΤΕΧνικές: Δημιουργία νέων συμβόλων από ήδη υπάρχοντα

---

Απόστολος Συρόπουλος

28ης Οκτωβρίου 366

671 00 Ξάνθη

E-mail: [apostolo@obelix.ee.duth.gr](mailto:apostolo@obelix.ee.duth.gr)

Σ' αρκετές περιπτώσεις τα μαθηματικά σύμβολα που παρέχει το ΤΕΧ δεν είναι αρκετά. Για παράδειγμα, το σύμβολο  $\cup$  δεν συμπεριλαμβάνεται στον κατάλογο συμβόλων που παρέχει το ΤΕΧ. Εντούτοις το ΤΕΧ παρέχει εκείνους τους μηχανισμούς με τους οποίους μπορούμε να δημιουργήσουμε νέα σύμβολα με απλούς αλλά και πολύπλοκους τρόπους. Υπάρχουν δύο βασικοί τρόποι δημιουργίας νέων συμβόλων: χρησιμοποιώντας υπάρχοντα σύμβολα τα οποία τα τοποθετούμε το «ένα επάνω στο άλλο» ή χρησιμοποιώντας τις σχεδιαστικές ικανότητες του ΤΕΧ σχεδιάζουμε το νέο σύμβολο. Ας πάρουμε την πρώτη περίπτωση. Το ΤΕΧ θεωρεί τον κάθε χαρακτήρα, την κάθε λέξη και την κάθε παράγραφο ως ένα κουτί (box στην ορολογία του Knuth) το οποίο μπορεί να μετακινηθεί κάθετα και οριζόντια. Με τον τρόπο αυτό μπορούμε να δημιουργήσουμε νέα σύμβολα επικαλύπτοντας το κουτί ενός χαρακτήρα με το κουτί ενός άλλου χαρακτήρα. Για

παράδειγμα το σύμβολο  $\cap$  δημιουργήθηκε με την εντολή

```
\cap\kern-.69em\cdot
```

Εδώ βλέπουμε μία τυπική εφαρμογή της εντολής `\kern` η οποία μπορεί να μετακινεί οριζόντια κατά κάποιο μήκος το κουτί (χαρακτήρα) που ακολουθεί. Με άλλα λόγια μετακινούμε αριστερά το κουτί του χαρακτήρα  $\cdot$  κατά  $0,69\text{ em}$  και έτσι ο χαρακτήρας αυτός εμφανίζεται πάνω στον χαρακτήρα  $\cap$ . Όπως γίνεται φανερό αυτή η μέθοδος έχει το μειονέκτημα ότι απαιτεί να πειραματιστούμε με διάφορα μήκη μέχρι να βρούμε το ιδανικό. Όμως αν πρόκειται για κάποιο σχετικά απλό σύμβολο όπως το  $\cup$ , μπορούμε να χρησιμοποιήσουμε την εντολή `\mathaccent` η οποία τοποθετεί το σύμβολο που ακολουθεί ακριβώς κεντραρισμένο πάνω από το επόμενο σύμβολο. Πιο συγκεκριμένα το σύμβολο  $\cup$  δημιουργήθηκε με την εντολή

```
\mathaccent\cdot\cup
```

Ας δούμε ένα ακόμη παράδειγμα το οποίο δείχνει τα μειονεκτήματα της χρήσης της εντολής `\kern`: συγκρίνεται τα σύμβολα  $\uparrow$  και  $\uparrow$ , ποιο νομίζετε ότι είναι καλύτερο; Προσέξτε ότι στο δεύτερο σύμβολο η οριζόντια γραμμή είναι ελαφρά πιο ψηλά από ότι πρέπει. Δείτε τώρα ποιο είναι το τμήμα που πρέπει να πληρώσει κανείς ώστε να έχει ορθά τοποθετημένη την οριζόντια γραμμή:

```
\DeclareRobustCommand\upplus{%
  \mathbin{\mathpalette\upl@p-}}
\def\upl@p#1#2{\leavevmode%
  \vbox{\baselineskip 0.2ex
  \lineskiplimit-\maxdimen\m@th
  \ialign{\hfil$##$\hfil\cr
    #1\uparrow\cr
    #1#2\cr}}}
```

Τι κάνουν όμως οι διάφορες εντολές που ορίζουν τη νέα εντολή; Καταρχήν με την εντολή `\mathbin` ορίζουμε ότι το  $\TeX$  πρέπει να χειρίζεται ως δυαδικό τελεστή το όρισμά της. Η εντολή `\mathpalette` χρησιμοποιείται για τη σωστή τοποθέτηση μαθηματικού κειμένου στις διάφορες δυνατές μορφές του, π.χ., `display style`, `script style`, κ.λπ. Ειδικότερα δέχεται δύο ορίσματα:

μια μακροεντολή που καθορίζει πως θα στοιχειοθετηθεί το δεύτερο όρισμα που είναι ένα σύμβολο. Η μακροεντολή `\up@p` δέχεται δύο ορίσματα και κάνει χρήση της μακροεντολής `\ialign` η οποία είναι μια ειδική μορφή της εντολής `\halign`. Η μακροεντολή `\leavevmode` είναι απαραίτητη κάθε φορά που πάμε να δημιουργήσουμε ένα νέο σύνθετο χαρακτήρα. Με την εντολή `\vbox` δημιουργούμε ένα νέο κάθετο κουτί στο οποίο θα τοποθετήσουμε τον νέο χαρακτήρα. Καθορίζουμε οι γραμμές βάσης να απέχουν μεταξύ των 0,2ex. Η εντολή `\m@th` καθορίζει ότι δεν υπάρχει καθόλου οριζόντιος κενός χώρος πριν και μετά το σύμβολο, γιατί απλά αυτός ο χώρος θα μπει αυτόματα από την εντολή `\mathbin`. Για τον τρόπο χρήσης της εντολής `\halign` μπορείτε να διαβάσετε ένα οποιοδήποτε βιβλίο για το  $\TeX$ .

Αν τώρα κάποιος επιθυμεί κάνει ακόμη πιο πολύπλοκα σχήματα, μπορεί να χρησιμοποιήσει ένα πακέτο όπως το `sprite` το οποίο παρέχει ένα καμβά για το σχεδιασμό ενός συμβόλου. (Περισσότερες πληροφορίες για το πακέτο αυτό μπορείτε να βρείτε στο βιβλίο  $\LaTeX$  του συγγραφέα του παρόντος.)

## ΒΙΒΛΙΟ-ΠΑΡΟΥΣΙΑΣΗ

---

Δημήτριος Φιλίππου

Κάτω Γατζέα  
385 00 Βόλος

**Δυὸ λόγια γιὰ τὴν ΒΙΒΛΙΟ-ΠΑΡΟΥΣΙΑΣΗ** — Ἐτοῦτες οἱ σελίδες σκοπὸ ἔχουν νὰ παρουσιάσουν ὀρισμένα βιβλία ποὺ θὰ ἐνδιέφεραν τὶς ΤΕΧνίτριες καὶ τοὺς ΤΕΧνίτες. Κάθε συνδρομὴ στὴν παρουσίαση νέων βιβλίων (ἀλλὰ καὶ ὄχι τόσο νέων) γύρω ἀπὸ τὸ ΤΕΧ καὶ τὴν Τυπογραφία εἶναι καλοδεχούμενη. (Οἱ περίεργοι ἀναγνώστες μποροῦν ἐπίσης νὰ σημειώσουν ὅτι τὸν τίτλο «Βιβλίο-Παρουσίαση» τὸν δανειστήκαμε ἀπὸ κάποιες καθημερινὲς ἐφημερίδες, ἀλλὰ ἐλπίζουμε ὅτι — μιᾶς καὶ πρόκειται γιὰ ἓνα τόσο μικρὸ δάνειο — θὰ μᾶς τὸ χαρίσουν.)

\*  
\*  
\*

**David Salomon, *The Advanced T<sub>E</sub>Xbook***, Springer-Verlag, Berlin, Heidelberg, London 1995. Σελ. xx + 490. ISBN 0-387-94556-3. Τιμὴ 48,95 δολ. ΗΠΑ (περίπου 16.000 δρχ.). Διατίθεται ἀπὸ βιβλιοπωλεῖα ξένων ἐπιστημονικῶν βιβλίων.

Ὅσοι ἤδη ξεφύλλισαν τὸ *T<sub>E</sub>Xbook* τοῦ Knuth καὶ τὸ βρῆκαν δύσκολο, ἢ ὅσοι δὲν ἱκανοποιοῦνται μὲ τὶς ἔτοιμες λύσεις τύπου L<sup>A</sup>T<sub>E</sub>X ἀλλὰ δὲν ἀνοιζαν ποτὲ τὸ *T<sub>E</sub>Xbook*, τώρα μποροῦν νὰ μάθουν τὰ ἐνδότερα μυστικά τοῦ ΤΕΧ ἀπὸ τὸ *προχωρημένο* ἐγχειρίδιο τοῦ David Salomon.

Πιὸ ὀγκῶδες ἀπὸ τὸ *T<sub>E</sub>Xbook*, τὸ βιβλίο τοῦ Salomon καλύπτει σχεδὸν τὰ πάντα γύρω ἀπὸ τὸ ΤΕΧ: ἀπὸ τὸν τρόπο λειτουργίας τοῦ προγράμματος καὶ τὸ πῶς διαβάζει τὸν κώδικα, ἕως τὴν δημιουργία πολὺπλοκων συνόλων ὀρισμῶν (macro) καὶ ρουτινῶν ἐξόδου (output routines) γιὰ προβλήματα στοιχειοθεσίας ποὺ ἐπιλύονται μόνο μὲ μεθόδους πολλαπλῆς ἐπεξεργασίας (ὅ,τι δηλαδὴ κάνει τὸ L<sup>A</sup>T<sub>E</sub>X προκειμένου νὰ διασταυρώσει τὶς ἀναφορὲς στὸ κείμενο μὲ τοὺς ἀριθμοὺς τῆς βιβλιογραφίας).

Στὰ καλὰ τοῦ βιβλίου διακρίνουμε τὴν ἐκταταμένη κάλυψη πολλῶν θεμάτων κατὰ τρόπο συστηματικὸ. Παρ' ὅλα αὐτά, μερικὰ ἀπὸ τὰ πλέον δύσκολα θέματα στοιχειοθεσίας ἀφήνονται στὸ κεφ. 8 ὡς ἀσκήσεις γιὰ τὸν ἀναγνώστη — π.χ., ἡ σωστὴ τοποθέτηση περιθωριακῶν σημειώσεων στὸ δεξιὸ καὶ

στο άριστερό περιθώριο κατά την εκτύπωση διπλής όψης, κ.ά. Στα μειονεκτήματα του βιβλίου μπορούμε επίσης να συμπεριλάβουμε κάποιες επιλογές του συγγραφέα που δεν είναι ιδιαίτερας αισθητικές· κακές κοπές αράδων, πολλά παροράματα (μπορείτε να βρείτε έναν πλήρη κατάλογο παροραμάτων του βιβλίου στην ιστοσελίδα του συγγραφέα: <http://www.ecs.csun.edu/~dxs/tatb.advertis/tatbErrata.html>) και στριμωγμένη διάταξη που μάλλον πνίγει τον αναγνώστη. Μερικές σελίδες παραπάνω ίσως να μεγάλωναν τον όγκο του βιβλίου, αλλά θα έδιναν περισσότερο άερα κατά την ανάγνωση. Άς ελπίσουμε ότι αυτές οι άτέλειες θα πάψουν να ύφίστανται σε μία επόμενη έκδοση, γιατί πρόκειται για ένα πολύ καλό βιβλίο.

\*  
\*  
\*

**Alan Hoenig, *TEX Unbound: L<sup>A</sup>T<sub>E</sub>X and TEX Strategies for Fonts, Graphics and More***, Oxford University Press, New York, USA 1998. Σελ. xxiv + 580. ISBN 0-19-509686-X (χαρτόδετο), 0-19-509685-1 (πανόδετο). Τιμή χαρτόδετου 39,50 δολ. ΗΠΑ (περίπου 12.800 δρχ.), τιμή πανόδετου 65 δολ. ΗΠΑ (περίπου 20.800 δρχ.). Διατίθεται από βιβλιοπωλεία ξένων επιστημονικών βιβλίων.

Άκόμα ένα βιβλίο για το TEX, το L<sup>A</sup>T<sub>E</sub>X και, κυρίως, την χρήση γραμματοσειρών διαφόρων τύπων. Πιθανώτατα κανένα άλλο βιβλίο δεν καλύπτει σε τόση έκταση την χρήση γραμματοσειρών με το TEX· πέντε από τα δεκαπέντε κεφάλαια του βιβλίου είναι αφιερωμένα στην χρήση διαφόρων γραμματοσειρών με το TEX ή το L<sup>A</sup>T<sub>E</sub>X. Ο Hoenig έξηγεῖ με πολύ λεπτομέρεια πώς ορίζεται μία νέα οικογένεια γραμματοσειρών, πώς γίνεται η εγκατάσταση νέων γραμματοσειρών, την χρήση τῶν ἐν δυνάμει γραμματοσειρών (virtual fonts), καθώς και την ἀλλαγή γραμματοσειρών στους μαθηματικούς τύπους. Περίπου τριάντα σελίδες με παραδείγματα συμπληρώνουν την συζήτηση περί γραμματοσειρών. Άς σημειωθῆῖ ὅτι τὸ κύριο κείμενο ἔχει στοιχειοθετηθῆῖ με βασική γραμματοσειρά τὴν Adobe Garamond, μία ἐπιλογή τοῦ συγγραφέα που ἔδωσε ἕναν ξεχωριστὸ χαρακτήρα στὸ ὅλο βιβλίο.

Όπως φανερώνει και ὁ τίτλος τοῦ βιβλίου, ὁ Hoenig ἀναφέρεται ἐκτεταμένως και στὸ θέμα τῆς ἐνσωμάτωσης ἀρχείων με γραφικὲς παραστάσεις στὰ ἔντυπα που παράγονται με τὸ TEX ἢ τὸ L<sup>A</sup>T<sub>E</sub>X. Στα σχετικά κεφάλαια ἐξηγεῖται τὸ METAFONT, τὸ METAPOST, τὸ PICTEX, τὸ MFpic και τὸ πακέτο PSTricks. Ὡς πρὸς τὸ θέμα τῶν γραφικῶν, τὸ βιβλίο αὐτὸ μπορεῖ νὰ συγκριθῆῖ μόνο με τὸ *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion* τῶν Goosens, Rahtz και Mittelbach, παρ' ὅτι τὸ τελευταῖο εἶναι πιὸ πλήρες.

Τὰ θέματα που δὲν καλύπτονται καλά στὸ *TEX Unbound* εἶναι κυρίως τὰ βασικά· οἱ κύριες ἐντολὲς τοῦ TEX και τοῦ L<sup>A</sup>T<sub>E</sub>X και ἡ δυνατότητα δημιουργίας νέων

όρισμῶν. Ένας ἀρχάριος στὸ T<sub>E</sub>X μπορεῖ εὐκόλα νὰ χαθεῖ ἀνάμεσα στὴν ἐντολὴ `\hskip` τοῦ T<sub>E</sub>X καὶ στὴν ἐντολὴ `\hspace` τοῦ L<sup>A</sup>T<sub>E</sub>X, ποὺ εἶναι μὲν ἀντίστοιχη ἀλλὰ ὄχι ἐντελὼς ὅμοια. Συνεπὼς πρόκειται γιὰ ἓνα βιβλίον γιὰ προχωρημένους T<sub>E</sub>Xνόβιους μὲ κλίση στὶς γραμματοσειρές!

[Έκτεταμένες κριτικὲς γιὰ τὸ παραπάνω βιβλίον ἔχουν δημοσιευθεῖ ἀπὸ τὸν Michael Doob στὸ περιοδικὸ TUGboat (vol. 19, 1998, σελ. 113–114, στὰ ἀγγλικά) καὶ ἀπὸ τὸν Ulrik Vieth στὸ περιοδικὸ Die T<sub>E</sub>Xnische Komödie (no. 3, 1998, σελ. 54–57, στὰ γερμανικά). Τὴν κριτικὴ τοῦ Doob μπορεῖτε νὰ τὴν βρεῖτε στὴν ἱστοσελίδα: <http://www.tug.org/TUGboat/Contents/contents19-2.html>. Τὴν κριτικὴ τοῦ Vieth θὰ τὴν βρεῖτε στὴν ἱστοσελίδα: <http://www.thphy.uni-duesseldorf.de/vieth/subjects/tex/publist.html>.]

\*  
\* \*  
\*

**Adobe Systems Inc., PostScript<sup>®</sup> Language Reference**, 3rd edition, Addison-Wesley, Reading, Massachusetts, USA 1999. Σελ. xii + 897. ISBN 0-201-37922-8. Τιμὴ 49,95 δολ. ΗΠΑ (περίπου 16.000 δρχ.). Διατίθεται ἀπὸ βιβλιοπωλεῖα ξένων ἐπιστημονικῶν βιβλίων.

Ἡ PostScript, ἡ γλῶσσα «περιγραφῆς σελίδων» ποὺ χρησιμοποιεῖται τόσο σὲ ἐκτυπωτὲς laser ὥστε καὶ σὲ φωτοστοιχειοθετικὲς μηχανές ὑψηλῆς εὐκρίνειας, πρωτοεμφανίσθηκε τὸ 1985. Ἐκτοτε ἡ PostScript ἐπιβλήθηκε ὡς τὸ βιομηχανικὸ πρότυπο τῆς σύγχρονης τυπογραφίας.

Οἱ προγραμματιστὲς ποὺ ἀσχολοῦνται μὲ τὴν ψηφιακὴ τυπογραφία ἔχουν τώρα στὴν διάθεσή τους τὴν τρίτη ἔκδοση τοῦ ἐγχειριδίου τῆς PostScript, τοῦ βιβλίου ποὺ στοὺς μυημένους εἶναι γνωστὸ μὲ τὸ ὄνομα: *Τὸ κόκκινο βιβλίον*. Ἡ νέα ἔκδοση τοῦ ἐγχειριδίου τῆς PostScript περιέχει τὴν ἐπέκταση τῆς PostScript στὸ ἐπίπεδο 3 (LanguageLevel 3). Τὸ ἐγχειρίδιον συνοδεύεται ἀπὸ ἓνα CD-ROM ποὺ περιέχει ὅλο τὸ κείμενον τοῦ βιβλίου σὲ μορφή PDF (Portable Document Format).

*Τὸ κόκκινο βιβλίον* καλύπτει τὰ πάντα γύρω ἀπὸ τὴν PostScript, ἀπὸ τὴν ἱστορία καὶ τὶς βασικὲς ἐντολὲς τῆς γλῶσσας ἕως τὶς ἐφαρμογὲς τῆς σὲ συγκεκριμένα μηχανήματα ἐκτύπωσης. Ὅπως εἶναι φυσικόν, ἓνα τέτοιο ἐγχειρίδιον ἀποτελεῖ μάλλον βιβλίον ἐργασίας καὶ ἀναφορᾶς. Οἱ γνωστὲς τῆς γλῶσσας θὰ τὸ ἐκτιμήσουν ἰδιαιτέρως, ἀλλὰ ὅσοι δὲν κατέχουν τὴν PostScript ἄς ἀναζητήσουν καλύτερα κάποιον ἄλλο εἰσαγωγικὸ βιβλίον.

[Θερμές ευχαριστίες στον Απόστολο Συρόπουλο για την βοήθειά του στην παρουσίαση αυτού του βιβλίου.]

\*  
\* \*  
\*

**Donald E. Knuth, *Digital Typography***, CSLI Publications, Stanford, California, USA 1999. Σελ. xvi + 685. ISBN 1-57586-011-2 (πανόδετο), 1-57586-010-4 (χαρτόδετο). Τιμή χαρτόδετου 39,95 δολ. ΗΠΑ (περίπου 12.800 δρχ.), τιμή πανόδετου 90,00 δολ ΗΠΑ (περίπου 28.500 δρχ.). Διατίθεται από βιβλιοπωλεία ξένων επιστημονικών βιβλίων.

Για τον Donald Knuth δέν χρειάζονται ιδιαίτερες συστάσεις. Ο δημιουργός του  $\TeX$  αυτή την φορά επανέρχεται με μία συλλογή τριάντα περίπου άρθρων, που δημοσίευσε κατά καιρούς σε διάφορα περιοδικά, αλλά και προσωπικών σημειώσεων. Άγαπημένο του θέμα παραμένει πάντα η σχέση της τυπογραφίας με τα μαθηματικά και τους υπολογιστές. Ορίστε μερικοί τίτλοι από τα περιεχόμενα του βιβλίου: «Ψηφιακή τυπογραφία», «Μαθηματική τυπογραφία», «Τὸ γράμμα S», «Προβλήματα για ένα κυριακάτικο πρωινό», «Τὸ μέλλον τοῦ  $\TeX$  καὶ τοῦ METAFONT», κ.ἄ.

Με λίγα λόγια πρόκειται για ένα βιβλίο για ὅσους ἀγαποῦν τὴν ψηφιακὴ τυπογραφία στὴν λεπτομέρειά της, ἀλλὰ καὶ τὸν ξεχωριστὸ τρόπο γραφῆς τοῦ Knuth.

