

UNIVERZA V LJUBLJANI
PEDAGOŠKA FAKULTETA V LJUBLJANI
ODDELEK ZA MATEMATIKO IN RAČUNALNIŠTVO

MARKO RAZPET
SEDI IN RIŠI S
P_ICT_EXOM!

Druga izdaja

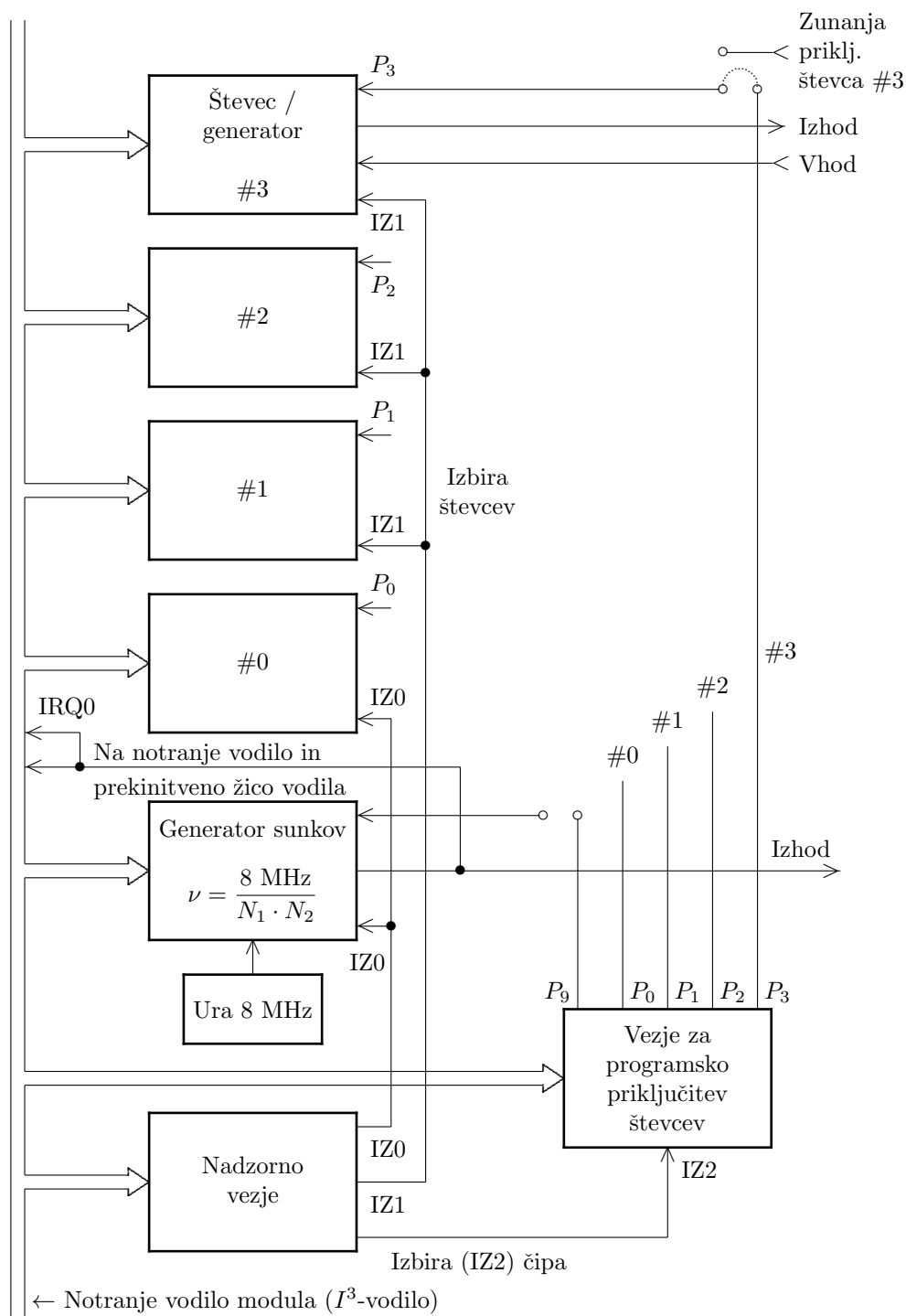
LJUBLJANA 2004

Kazalo

Uvod	3
Priprava okolja za P _T CT _E X	5
Koordinatni sistemi	5
Postavljanje besedila	6
Določitev risalnega polja	9
Vodoravne in navpične črte	11
Risanje pravokotnikov	12
Mreže in napisi	13
Risanje krivulj	14
Puščice	16
Vzorčenje črt	17
Kotiranje	20
Senčenje	21
Zasuki	26
Histogrami	27
Označevanje osi	31
Nekatere izboljšave	37
Basic in programiranje krivulj	45
Vključitev slik v besedilo	47
Literatura	48

Uvod

Zadnja leta je kar veliko ljudi začelo uporabljati pri svojem delu $\text{T}_{\text{E}}\text{X}$ oziroma $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, računalniški orodji za postavljanje besedil. Prenekateri uporabnik je tako delo vzel za svoje in nehoti postal avtor, stavec in recenzent v eni osebi. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ omogoča tudi risanje preprostih slik, ki po potrebi izdelek poživijo. Programski paket $\text{e}_{\text{m}}\text{T}_{\text{E}}\text{X}$ ponuja veliko možnosti, ki jih človek morda prezre, če ga kdo nanje ne spomni. Ena od njih je $\text{P}_{\text{I}}\text{C}_{\text{T}_{\text{E}}\text{X}}$, s katerim lahko oblikujemo prav lepe slike, na primer



S časom so postali osebni računalniki vedno bolj hitri in zmogljivi, ne nazadnje tudi cenejši in dostopnejši širšemu krogu uporabnikov. Generaciji osebnih računalnikov s procesorjem tipa 286 so se pridružili oni s procesorji tipa 386 in 486 in pričakovati je še boljše stroje. Velikost RAM-a 640 kBytov že davnaj ne zadošča več, spodobi se imeti vsaj kakih 4 MBytov. Tudi DOS napreduje in aktiviranje tako velikega pomnilnika ne predstavlja večjih težav.

Paket $\text{emT}_{\text{E}}\text{X}$ punuja tudi debelejšo verzijo $\text{T}_{\text{E}}\text{X}$ a in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a, $\text{bT}_{\text{E}}\text{X}$ in $\text{bL}^{\text{A}}\text{T}_{\text{E}}\text{X}$, s katerima lahko obdelujemo besedilo s slikami vred. S $\text{P}_{\text{T}}\text{C}_{\text{T}}\text{E}_{\text{X}}$ om narejene slike se dajo lepo vklopiti v običajno $\text{T}_{\text{E}}\text{X}$ ovo ali $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovo besedilo, pa tudi znaki (črke, številke, ...) so lahko povsod enako oblikovani. Naknadne spremembe slik se dajo hitro opraviti, recimo pomanjšava, sprememba napisov in debelin črt. Poleg tega je opis slik napisan z znaki ASCII, kar omogoča preprost prenos po elektronski pošti praktično po celem svetu. Prenosljivost na večje sisteme, recimo na VAX, tudi ne dela posebnih težav.

DOS 5.0 ali boljši vsebuje tudi *qbasic*, s katerim lahko hitro naredimo program za izračun potrebnih koordinat elementov slike, na primer krivulj. Tako lahko postane avtor poleg ostalega tudi sam svoj risar. Res je, da obstajajo dosti boljši programi za risanje, toda veliko teh zahteva precej prostora na disku. $\text{P}_{\text{T}}\text{C}_{\text{T}}\text{E}_{\text{X}}$ zaseda le slabih 134 kBytov prostora. $\text{P}_{\text{T}}\text{C}_{\text{T}}\text{E}_{\text{X}}$ ove slike se dajo vnesti v besedilo že na manjših računalnikih, natisniti pa se da vse skupaj že s preprostejšimi tiskalniki.

Paket $\text{emT}_{\text{E}}\text{X}$ vsebuje tudi program $\text{T}_{\text{E}}\text{X}$ CAD, ki nadomešča mučno pisanje ukazov za slike v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u in s katerim lahko z miško ustvarjamo preproste slike, ki se dajo vključiti v besedilo. Dobili smo še $\text{P}_{\text{T}}\text{C}_{\text{T}}\text{E}_{\text{X}}$ CAD, ki igra podobno vlogo kot $\text{T}_{\text{E}}\text{X}$ CAD, z miško se izognemo dolgotrajnemu vnašanju ukazov. Primeren je zlasti za slike z geometrijskimi konstrukcijami.

Ljubljana, oktober 2004

Marko Razpet

Priprava okolja za P_TE_X

Pred opisom slike z P_TE_Xovimi ukazi napišemo ukaz

```
\input pictex
```

če bomo slike nameščali v T_EXovo besedilo (to naredimo pred prvo sliko) in

```
\input prepictex,
```

```
\input pictex
```

```
\input postpictex
```

v primeru L^AT_EXovega besedila. Tedaj namreč ukazi v prvi in zadnji datoteki med seboj spravijo nekatere L^AT_EXove in P_TE_Xove definicije. Namesto na pogonu **c**: utegne biti **emt_EX** nameščen na kakem drugem. Temu primerno nadomestimo v zgornjih ukazih **c**: z na primer **d**:

Seznam P_TE_Xovih ukazov za sliko pričnemo z

```
\beginpicture
```

zaključimo pa z

```
\endpicture
```

V nadaljevanju bomo z oklepaji [] označevali tiste dele ukazov, ki se lahko izpustijo (z oklepaji vred), učinek pa bo seveda drug. Z [] pa bomo označevali tiste dele ukazov, ki morajo biti izpisani z oklepaji vred, če smo jih že uporabili.

Koordinatni sistemi

Koordinatni sistem vpeljemo z ukazom

```
\setcoordinatesystem [units <x_enota,y_enota>] [point at x_ref y_ref]
```

Abscisno in ordinatno os, ki ju P_TE_X ne nariše z zgornjim ukazom, si mislimo orientirani tako kot v matematiki. Če izpustimo del ukaza z **units**, s katerim definiramo velikost obeh enot *x_enota* in *y_enota*, bo P_TE_X vzal zanju **1pt**. Sicer pa definiramo enoti na koordinatnih oseh sami, in sicer v veljavnih merskih enotah, ki jih pozna T_EX: **cm**, **mm**, **in**, **pt**.

Če izpustimo del ukaza **point at**, bo P_TE_X smatral, da sta *x_ref* in *y_ref* enaka 0. Z njima definiramo referenčno točko za nov koordinatni sistem v prvotnem. Uporabo tega ukaza pojasnjuje naslednji primer.

Dva koordinatna sistema definiramo na primer z ukazi:

```
\beginpicture
```

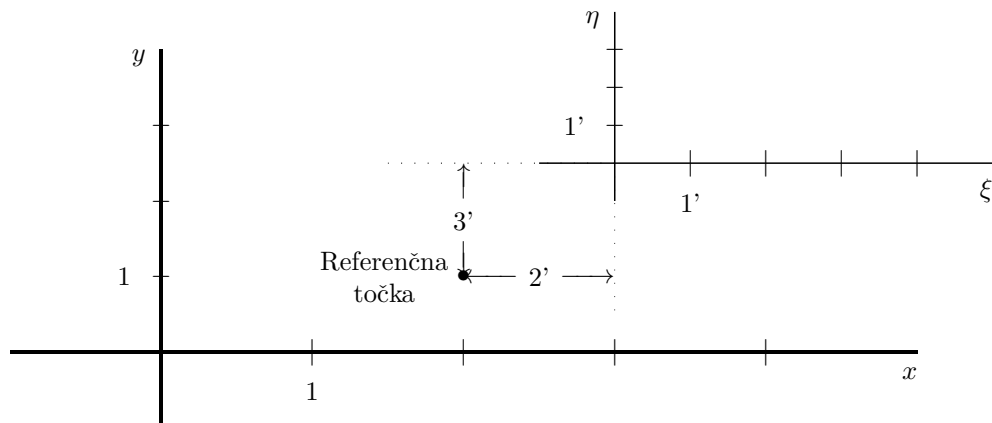
```
\setcoordinatesystem units <2 cm, 1 cm> point at 2 1
```

```
\setcoordinatesystem units <1 cm, 0.5 cm> point at -2 -3
```

```
\endpicture
```

Referenčna točka v prvem koordinatnem sistemu ima koordinati $x = 2$, merjeno v enotah $x_enota = 2$ cm oziroma $y = 1$, merjeno v enotah $y_enota = 1$ cm. Referenčna točka • ima v drugem koordinatnem sistemu koordinati $\xi = -2$ in $\eta = -3$, merjeno v novih enotah $\xi_enota = 1$ cm oziroma $\eta_enota = 0.5$ cm.

Koordinate točk podajamo v P_TE_Xu z urejenim parom števil, med katerima moramo pisati presledek.



Postavljanje besedila

Koordinate točk podajamo v P_TE_Xu z urejenimi pari števil, ločenimi s presledkom. Števila so lahko decimalna, pisana s piko. Koordinate točk na sliki so merjene v enotah, ki smo jih vpeljali v zadnjem ukazu `\setcoordinatesystem`.

Splošna oblika ukaza za postavljanje besedila je

`\put {besedilo} [ $p_x p_y$ ] [ $\langle x\_odmik, y\_odmik \rangle$ ] at  $x y$`

Vedeti je treba, da T_EX razume *besedilo* kot škatlo, ki ima središče in osnovnico. To škatlo lahko potem s parametroma p_x in p_y namestimo v koordinatnem sistemu glede na točko s koordinatama x in y . Če določilo s p_x in p_y izpustimo, bo točka $x y$ v središču škatle *besedilo*.



Sicer pa sta za p_x dovoljena parametra **r** ali **l**, za p_y pa **t**, **b** ali **B**. Pri tem pomeni:

- r** — točka $x y$ je na desnem robu škatle;
- l** — točka $x y$ je na levem robu škatle;
- t** — točka $x y$ je na zgornjem robu škatle;
- b** — točka $x y$ je na spodnjem robu škatle;
- B** — točka $x y$ je na osnovnici škatle.

Z neobveznima parametroma x_odmik in y_odmik v izbranih merskih enotah lahko škatlo po potrebi še nekoliko premaknemo. Delovanje ukaza `\put` pojasnimo z naslednjim opisom slike:

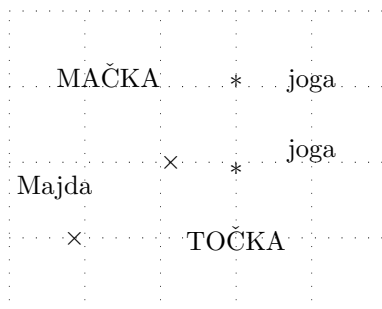
```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\put { $\times$ } [l] at 2 2
\put { $\times$ } [r] at 1 1
\put { $\ast$ } [b] at 3 3
\put { $\ast$ } [t] at 3 2
```

```

\put {T0\v CKA} at 3 1
\put {MA\v CKA} [rb] at 2 3
\put {joga} [B] at 4 3
\put {joga} [b] at 4 2
\put {Majda} [tl] <1mm, -5pt> at 0 2
\endpicture

```

Na sliki tudi vidimo, kakšen je učinek, če katero od določil p_x ali p_y izpustimo.



Besedilo lahko pišemo tudi v stolpcu. V ta namen pozna $\text{\PCTEX}$ ukaze `\lines`, `Lines` in `\stack`. Splošna oblika prvega od teh ukazov je

```

\lines [p] {vrstica_1\cr vrstica_2\cr ...}

```

kjer je **p** način poravnave besedila v stolpcu: **l** za levo in **r** za desno poravnavo. Če to določilo izpustimo, dobimo središčno poravnavo. Če je ukaz `\lines` kombiniran z ukazom `\put` z določilom **B**, potem se ta **B** nanaša na zadnjo vrstico v `\lines`. Ukaz `\Lines` uporabljamo enako, le da se tedaj določilo **B** nanaša na prvo vrstico.

Ukaz `\stack` ima splošno obliko:

```

\stack [p] [<razmik>] {seznam}

```

Pri tem pomeni **p**, enako kot pri ukazu `\lines`, poravnavo elementov v seznamu. Posamezne elemente seznama ločimo z vejicami. Parameter *razmik* lahko sami nastavimo v $\text{\TeX}$ ovih merskih enotah.

Ukazi

```

\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\put {\lines[r]{Desno\cr poravnano\cr besedilo}} [lt] at 3 2
\put {\lines[l]{Levo\cr poravnano\cr besedilo}} [rb] at 2 2
\put {\lines{Sredi\v s\v cno\cr poravnano\cr besedilo}} [t] at 3 4
\put {\lines{Mama\cr moja!}} [Br] at 2.5 1
\put {\Lines{Cara\cr mia!}} [Br] at 1 1
\endpicture

```

nam ustvarijo naslednjo sliko.

		Središčno poravnano besedilo	
Levo poravnano besedilo			
	Mama moja!	Desno poravnano besedilo	
Cara mia!			

Kot primer uporabe ukaza `\stack` vzemimo:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\put {\stack [r] {N,A,M,A}} [rb] at 4 2
\put {\stack {V,a,j,a}} [rt] at 2 4
\put {\stack <4pt> {D,A,I,R,Y,,Q,U,E,E,N}} [rt] at 5 4
\put {\stack <8pt> {HOTEL,SLON}} [t] at 1 2
\endpicture
```

Gornji ukazi ustvarijo sliko:

	V a j a		D A I R Y Q U E E N
		N A M A	
HOTEL SLON			

Če je treba postaviti isto besedilo na več mest, uporabimo ukaz `\multiput`. Ena od oblik tega ukaza je

```
\multiput {besedilo} [[p_xp_y]] [<x_odmik,y_odmik>] at x_1 y_1 x_2 y_2 ... x_n y_n /
```

Pri tem smo uporabili enake oznake kot pri ukazu `\put`. Če je točk veliko, se jih splača zbrati v posebno datoteko (ASCII), recimo z imenom *podatki.dat*. Teda j uporabimo ukaz oblike

```
\multiput {besedilo} [[p_xp_y]] [<x_odmik,y_odmik>] at "podatki.dat"
```

Pogosto pride prav ponavljanje besedila v enakomernih presledkih v predpisani smeri. Povedati je treba, kam besedilo prvič postavimo (x_1 , y_1), kolikokrat ga nato še ponovimo (parameter n) in v kateri smeri (dx , dy). Besedilo bo postavljeno v točkah

$$x_1 \ y_1, x_1 + dx \ y_1 + dy, \dots, x_1 + n \cdot dx \ y_1 + n \cdot dy$$

Ukaz je, z istim pomenom določil kot prej, take oblike:

```
\multiput {besedilo} [[p_xp_y]] [<x_odmik,y_odmik>] at x_1 y_1 *n dx dy /
```

Za pojasnitev ukaza `\multiput` vzemimo na primer opis slike:

```

\beginpicture

\setcoordinatesystem units <10 mm, 10 mm>

\setplotarea x from 0 to 6, y from 0 to 5

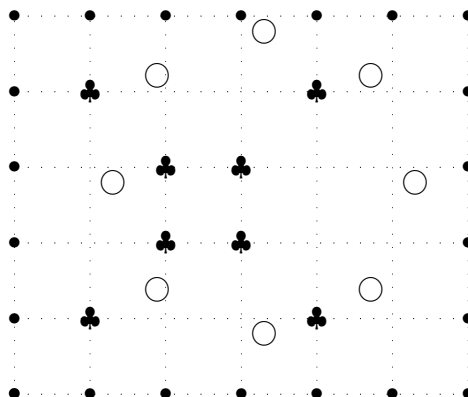
\multiput {$\bigcirc$} <8mm,3mm> at
4.5 2.5
3.914 3.914
2.5 4.5
1.086 3.914
0.5 2.5
1.086 1.086
2.5 .5
3.914 1.086 /

\multiput {$\clubsuit$} at 1 1 *3 1 1 /
\multiput {$\clubsuit$} at 1 4 *3 1 -1 /
\multiput {$\bullet$} at 0 0 *6 1 0 /
\multiput {$\bullet$} at 0 0 *5 0 1 /
\multiput {$\bullet$} at 6 1 *4 0 1 /
\multiput {$\bullet$} at 1 5 *4 1 0 /

\endpicture

```

Našteti ukazi nam naredijo naslednjo sliko:



Določitev risalnega polja

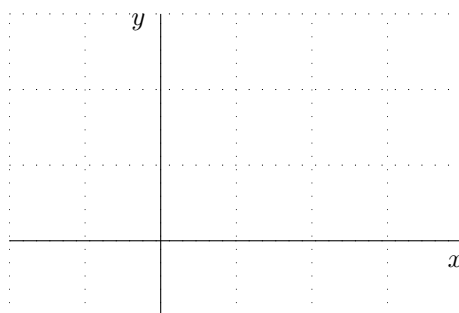
Potem, ko smo uporabili ukaz `\setcoordinatesystem` in z njim definirali enoti v smeri abscise in ordinatne osi, definiramo risalno polje z ukazom

```
\setplotarea x from  $x_{min}$  to  $x_{max}$ , y from  $y_{min}$  to  $y_{max}$ 
```

Ukaz pride zelo prav, če nameravamo sliko integrirati v navadno besedilo ali če rišemo mrežo ali koordinatne osi. Vendar s tem ni rečeno, da določene sestavine slike ne bodo zunaj tega polja. Z ukazi

```
\beginpicture
\setcoordinatesystem units <10 mm, 10 mm>
\setplotarea x from -2 to 4, y from -1 to 3
\endpicture
```

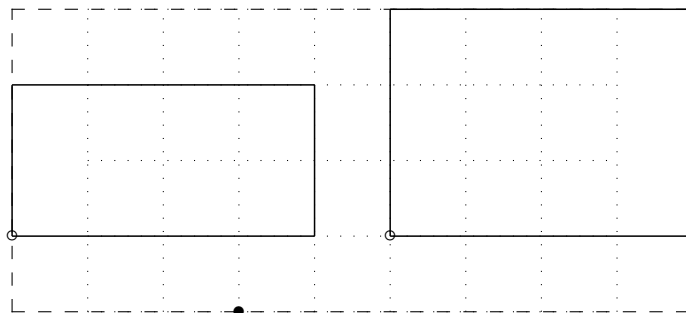
definiramo tako risalno polje:



Več vpeljanih koordinatnih sistemov omogoča večkratno uporabo ukaza `\setplotarea`. Z ukazi

```
\beginpicture
\setcoordinatesystem units <10 mm, 10 mm>
\setplotarea x from -2 to 5, y from 0 to 4
\setcoordinatesystem units <10 mm, 10 mm> point at -2 -1
\setplotarea x from 0 to 4, y from 0 to 3
\setcoordinatesystem units <10 mm, 10 mm> point at 3 -1
\setplotarea x from 0 to 4, y from 0 to 2
\endpicture
```

vpeljemo risalna polja, kot je prikazano na sliki. Pri tem označuje znak $\bullet$ koordinatno izhodišče (referenčno točko) prvega koordinatnega sistema, znaka $\circ$ pa izhodišči preostalih dveh vpeljanih koordinatnih sistemov. Minimalna škatla, ki zajema vsa vpeljana risalna polja, je označena črtkano.



Osnovnica minimalne škatle, v katero spadajo vse škatle nastopajočih gradnikov končnih mer, je določena na nekoliko zapleten način (glej na primer H. Kopka, [2], stran 132). V običajno besedilo se slika vstavi kot minimalna škatla po T_EXovih pravilih. Formiranje podatkov o minimalni škatli vstavimo z ukazom

```
\accountingoff
```

prekličemo pa z

```
\accountingon
```

Pri uporabi ukaza `\accountingoff` se lahko zgodi, da bo minimalna škatla ničelnih razsežnosti. To praktično to pomeni, da bodo lahko nekateri elementi slike štrleli v besedilo. Zato je priporočljivo, da si takoj na začetku s `\setplotarea` definiramo dovolj prostora za sliko.

Vodoravne in navpične črte

S P_TCT_EXom lahko rišemo ravne črte. Najhitreje gre delo od rok pri risanju vodoravnih in navpičnih črt.

Debelino le-teh izberemo z ukazom

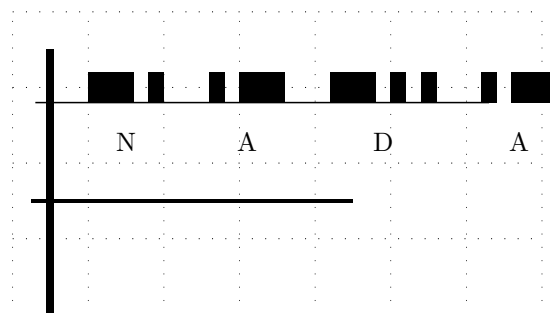
```
\linethickness debelina
```

kjer pomeni *debelina* debelino črt v veljavnih T_EX-ovih enotah. Ponujena debelina je 0.4pt.

Vodoravno ali navpično črto narišemo z ukazom

```
\putrule from x_1 y_1 to x_2 y_2
```

Pri tem je $x_1 = x_2$ pri navpični in $y_1 = y_2$ pri vodoravni črti. Sliko na desni dobimo z ukazi:



```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 7, y from 0 to 4
\linethickness 0.5mm \putrule from 0.25 1.5 to 4.5 1.5
\linethickness 1mm \putrule from 0.5 0 to 0.5 3.5
\linethickness 0.2mm \putrule <3mm,-2mm> from 0 3 to 6 3
\linethickness 4mm
\putrule from 1.0 3 to 1.6 3 \putrule from 1.8 3 to 2.0 3
\putrule from 2.6 3 to 2.8 3 \putrule from 3.0 3 to 3.6 3
\putrule from 4.2 3 to 4.8 3 \putrule from 5.0 3 to 5.2 3
\putrule from 5.4 3 to 5.6 3 \putrule from 6.2 3 to 6.4 3
\putrule from 6.6 3 to 7.2 3
\put {N} [t] at 1.5 2.4 \put {A} [t] at 3.1 2.4
\put {D} [t] at 4.9 2.4 \put {A} [t] at 6.7 2.4
\endpicture
```

Podobno kot ukaz `\put` dopušča tudi ukaz `\putrule` pomik črte za x_{odmik} y_{odmik} . V splošnem je ukaz oblike

```
\putrule [ $\langle x_{odmik}, y_{odmik} \rangle$ ] from  $x_1$   $y_1$  to  $x_2$   $y_2$ 
```

Primer uporabe je razviden iz prejšnje slike.

Risanje pravokotnikov

Pravokotnik, ki ima stranice vzporedne s koordinatnimi osmi narišemo v P_TE_Xu z ukazom

```
\putrectangle corners at  $x_1$   $y_1$  and  $x_2$   $y_2$ 
```

kjer sta si točki x_1 y_1 in x_2 y_2 diagonalno nasprotni. Debelina črt je določena s predhodnim ukazom `\linethickness`. Podobno kot pri ukazu `\put` lahko dosežemo pomik z ukazom

```
\putrectangle [ $\langle x_{odmik}, y_{odmik} \rangle$ ] corners at  $x_1$   $y_1$  and  $x_2$   $y_2$ 
```

Pravokotnike lahko rišemo tudi z ukazom `\putbar`. V tem primeru je treba podati bodisi njegovo vodoravno srednjico skupaj z višino bodisi navpično srednjico skupaj z širino. Tedaj je možno pravokotnik tudi premakniti za x_{odmik} y_{odmik} . Splošni ukaz je v prvem primeru oblike

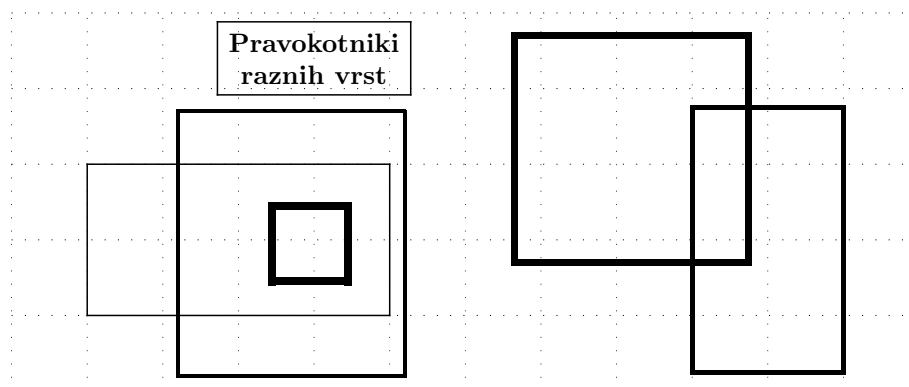
```
\putbar [ $\langle x_{odmik}, y_{odmik} \rangle$ ] breadth  $\langle višina \rangle$  from  $x_1$   $y_s$  to  $x_2$   $y_s$ 
```

v drugem primeru pa oblike

```
\putbar [ $\langle x_{odmik}, y_{odmik} \rangle$ ] breadth  $\langle širina \rangle$  from  $x_s$   $y_1$  to  $x_s$   $y_2$ 
```

Ukaz `\rectangle` lahko v kombinaciji z ukazom `\put` tudi uporabljamo za risanje pravokotnikov dane širine in višine, ki ju navajamo v veljavnih T_EXovih enotah med oklepaji `<>`, kot je razvidno iz naslednjega primera.

Prav tako uporabljamo za okvirjanje besedil v kombinaciji z ukazom `\put` ukaz `\frame`. Za tem ukazom navedemo razdaljo okvira od besedila in v zavitih oklepajih besedilo samo. Tudi uporaba tega ukaza je najboljše razvidna na sliki in njenem opisu.



Za gornjo sliko je treba napisati naslednje ukaze:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 12, y from 0 to 5 \linethickness 0.2mm
\put {\frame <1.5mm> {\bf \lines{Pravokotniki\cr raznih vrst}}}} at 4 4.4
```

```

\putrectangle corners at 1 1 and 5 3
\linethickness 0.4mm \putrectangle <2mm,-3mm> corners at 2 4 and 5 0.5
\linethickness 0.6mm
\putbar breadth <3.5cm> from 9 2 to 11 2
\linethickness 0.8mm
\putbar <2mm,-3mm> breadth <3.1cm> from 8 2 to 8 5
\linethickness 1mm
\put{\rectangle <10mm> <10mm> } [tr] at 4.5 2.5
\endpicture

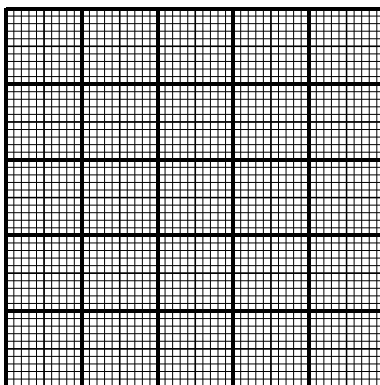
```

Mreže in napisi

P_TE_X pozna zelo učinkovit način za razdelitev polja, določenega z ukazom `\setplotarea`. To je ukaz

```
\grid n m
```

ki nam risalno polje razdeli na $n \times m$ pravokotnikov (n vodoravnih in m navpičnih pasov). Debelina črt je določena s predhodnim ukazom `\linethickness`. Primer milimetrskega papirja:



Dobljen je z ukazi:

```

\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 5
\linethickness 0.1mm \grid 50 50
\linethickness 0.2mm \grid 10 10
\linethickness 0.4mm \grid 5 5
\endpicture

```

Napis nad sliko, nad poljem, ki je določeno z ukazom `\setplotarea`, dobimo z ukazom

```
\plothead {napis}
```

Risanje krivulj

P_lCT_EX omogoča zelo natančno risanje krivulj. Kakovost končnega izdelka je odvisna od števila podanih točk in tiskalnika. Pri slikah z večjim številom krivulj utegnejo nastati težave pri uporabi osebnega računalnika z njegovim pomnilnikom. Pogosto ne zadošča več T_EX, ampak se moramo zateči k njegovi debelejši verziji bT_EX. Tedaj je zelo koristno imeti okrog 4 MBytov velik RAM, uporabljen kot podaljšan pomnilnik (EMS). Brez tega bT_EX sicer predela našo sliko, toda to lahko traja tudi nekaj ur, pogosto uporabljajoč trdi disk.

Zato je priporočljivo čim več uporabljati ukaze kot so `\putrule`, `\putrectangle`, ki rišejo vodoravne in navpične črte. Predelava poteka v tem primeru zelo hitro.

Ukaz `\plot` povezuje točke, ki so naštetje za njim: $x_1 y_1 x_2 y_2 \dots x_n y_n$. P_lCT_EX pozna dva načina povezovanja: linearnega in kvadratnega. Ponujen je linearni način. Med seboj ju preklapljam z ukazoma

```
\setlinear in \setquadratic
```

Pri linearnem načinu povezovanja dobimo povezave z daljicami po dve in dve točki: *prva*, *druga*; nato, če so navedene, *druga*, *tretja* in tako naprej. Pri kvadratnem načinu povezovanja pa dobimo povezave s parabolami po tri in tri točke: *prva*, *druga*, *tretja*; nato, če so navedene, *tretja*, *četrt*, *peta* in tako naprej. Pri kvadratnem načinu moramo navesti 3, 5, 7, ..., pri linearnem pa 2, 3, 4, ... točk. Navajanje točk zaključimo s presledkom in znakom /. Popoln ukaz je oblike:

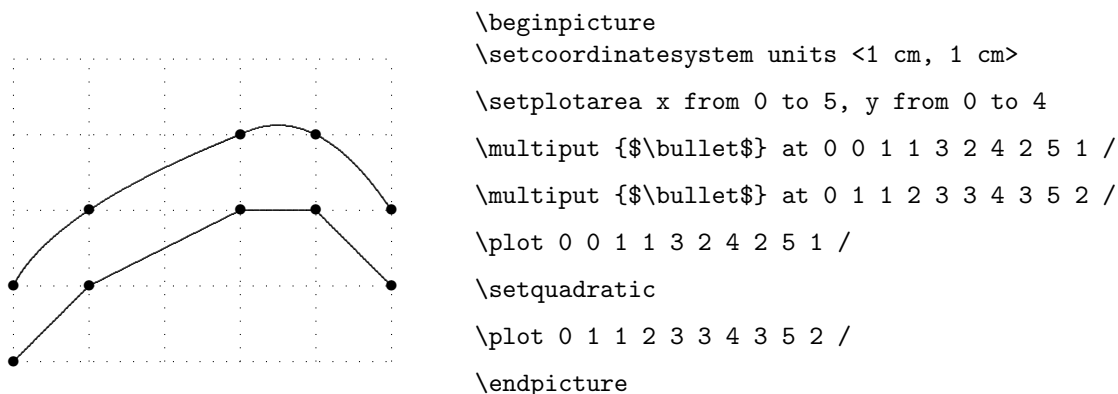
```
\plot x_1 y_1 x_2 y_2 \dots x_n y_n /
```

Pri večjem seznamu točk raje uporabljamo ukaz

```
\plot "seznam"
```

kjer so na datoteki *seznam* v obliki ASCII zbrane koordinate točk (podobno kot pri ukazu `\multiput`).

Točka, s katero P_lCT_EX riše z ukazom `\plot`, je pika iz nabora `\fiverm`, če nismo ukazali drugače. Uporaba ukaza `\plot` je razvidna iz naslednjega primera:



Za risanje krožnih in eliptičnih lokov ima P_lCT_EX posebna ukaza. Krožni lok s središčnim kotom α v stopinjah, z znakom $-$ v primeru negativno orientiranega loka, z začetkom v točki (x_0, y_0) in središčem v točki (x_s, y_s) , dobimo z ukazom

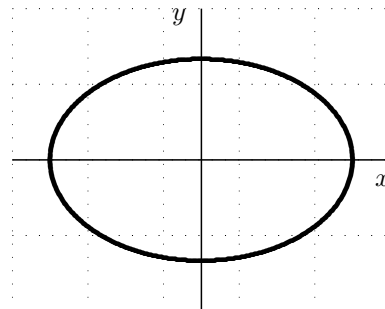
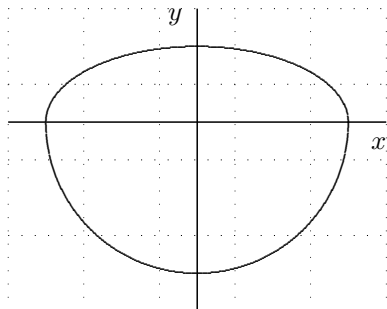
```
\circulararc  $\alpha$  degrees from  $x_0 y_0$  center at  $x_s y_s$ 
```

Podoben je ukaz za risanje eliptičnega loka, le da moramo dodati še razmerje njenih osi $a : b$ (a v smeri x in b v smeri y):

```
\ellipticalarc axes ratio  $a : b$   $\alpha$  degrees from  $x_0 y_0$  center at  $x_s y_s$ 
```

Za celo krožnico oziroma elipso vzamemo $\alpha = 360$. Oglejmo si to na primeru. Zlepek polovice elipse in

krožnice na spodnji sliki levo



in označeni koordinatni osi dobimo z naslednjimi ukazi:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\linethickness 0.2mm
\putrule from 0 2.5 to 5 2.5 \put {$x$} [tr] at 5 2.3
\putrule from 2.5 0 to 2.5 4 \put {$y$} [tr] at 2.3 4
\circulararc -180 degrees from 4.5 2.5 center at 2.5 2.5
\ellipticalarc axes ratio 2:1 180 degrees from 4.5 2.5 center at 2.5 2.5
\endpicture
```

Debelino krivulj lahko spremenimo tako, da zamenjamo *simbol*, s katerim riše ukaz `\plot`. Ukaz za to je

```
\setplotsymbol ({simbol})
```

Pri tem lahko posežemo tudi po naboru, ki ga sami definiramo s $\text{\TeX}$ ovim ukazom `\font`. Za debelo izvlečeno elipso na zgornji sliki smo uporabili ukaze:

```
\font\debeli cmr12 scaled \magstep2
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\linethickness 0.2mm \putrule from 0 2 to 5 2 \put {$x$} [tr] at 5 1.8
\putrule from 2.5 0 to 2.5 4 \put {$y$} [tr] at 2.3 4
\setplotsymbol ({\debeli.})
\ellipticalarc axes ratio 3:2 360 degrees from 4.5 2 center at 2.5 2
\endpicture
```

Krivulje, narejene z ukazom `\plot`, pa tudi krožni in eliptični loki ter puščice, dobljene z ukazom `\arrow`, lahko štrlijo iz risalnega polja, ki je določeno s `\setplotarea`. Z ukazom

`\inboundscheckon`

namreč prepovemo risanje izven tega polja, z ukazom

`\inboundscheckoff`

pa lahko to prepoved prekličemo. Vendar je le dobro vedeti, da se predelava slike lahko precej zavleče, če smo vključili `\inboundscheckon` pri velikem številu točk, segajočih izven risalnega območja. Črte, dobljene s `\putrule` in `\putrectangle` se pri tem ne porežejo. Prav tako ne napisi, ki smo jih postavili z ukazom `\put`.

Pri eliptičnih in krožnih lokih bo $\text{P}\text{T}\text{E}\text{X}$ vzel enaki enoti v smeri obeh koordinatnih osi, in sicer enoto v smeri x , vpeljano z ukazom `\setcoordinatesystem units`.

Puščice

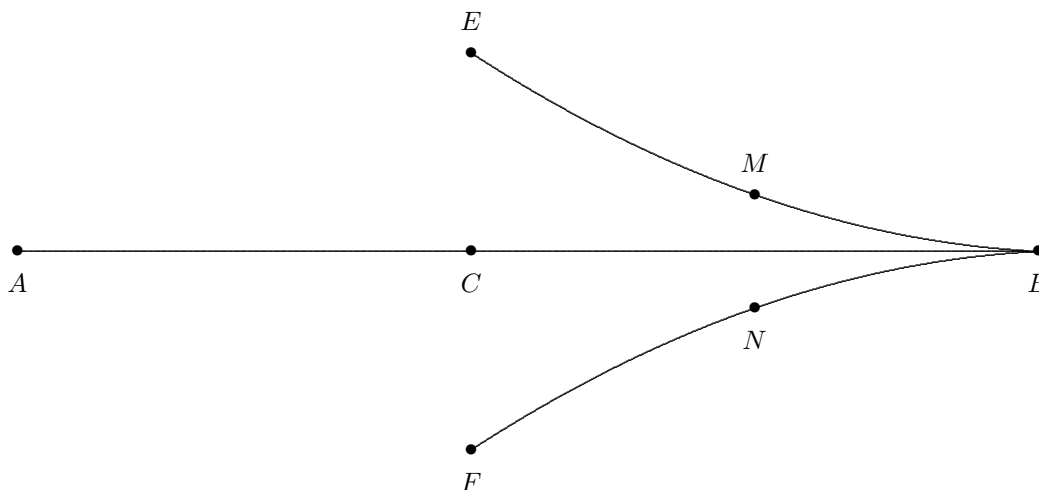
Puščice lahko rišemo s $\text{P}\text{T}\text{E}\text{X}$ om v poljubnih smereh in tudi njihovo obliko lahko izberemo. Upoštevati pa moramo, da poteka risanje slik, na kateri so puščice, prav tako počasi kot risanje krivulj z ukazom `\plot`. Če hočemo pridobiti na času, uporabljamo pri risanju vodoravnih in navpičnih puščic raje ukaz `\putrule` in znake $<$, $>$, $\vee$, $\wedge$. Če je le mogoče, uporabljamo puščice iz bogate zakladnice TEX ovega puščičja (`\longrightarrow...`).

Puščico narišemo z ukazom

`\arrow <dolžina_glave> [a,b] from x_1 y_1 to x_2 y_2`

kjer sta $x_1 y_1$ koordinati začetne točke A , $x_2 y_2$ koordinati končne točke B , $dolžina_glave$ razdalja med točkama C in B , medtem ko je $dolžina_glave \times a$ razdalja med točkama M in N ter $dolžina_glave \times b$ razdalja med točkama E in F . Z ukazom `\arrow` določimo torej smer in velikost puščice ter obliko njene glave. Razmere so prikazane na pretirano povečani puščici.

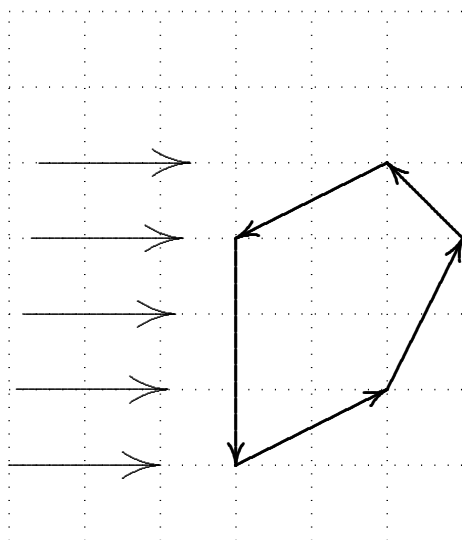
Debelino črt pri puščicah lahko spremenimo z izbiro drugega simbola z ukazom `\setplotsymbol`.



Puščico lahko še po želji premaknemo z dodatkom `<x_odmik y_odmik>` pred besedo `from` kakor na primer pri ukazu `\put`, tako da je splošni ukaz za puščico oblike

`\arrow <dolžina_glave> [a,b] [<x_odmik y_odmik>] from x_1 y_1 to x_2 y_2`

Puščičje na sliki dobimo na primer z ukazi, ki so napisani zraven.



```

\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from -2 to 4, y from -1 to 6
\arrow <5mm> [0.2,0.7] from -2 0 to 0 0
\arrow <5mm> [0.2,0.7] <1mm,10mm> from -2 0 to 0 0
\arrow <5mm> [0.2,0.7] <2mm,20mm> from -2 0 to 0 0
\arrow <5mm> [0.2,0.7] <3mm,30mm> from -2 0 to 0 0
\arrow <5mm> [0.2,0.7] <4mm,40mm> from -2 0 to 0 0
\setplotsymbol ({\tenrm.})
\arrow <3mm> [0.2,0.7] from 1 0 to 3 1
\arrow <3mm> [0.2,0.7] from 3 1 to 4 3
\arrow <3mm> [0.2,0.7] from 4 3 to 3 4
\arrow <3mm> [0.2,0.7] from 3 4 to 1 3
\arrow <3mm> [0.2,0.7] from 1 3 to 1 0
\endpicture

```

Vzorčenje črt

P_{CTE}X omogoča tudi risanje črt pikčasto, črtkano in celo z lastnim vzorcem. To velja tako za črte, dobljene z ukazi `\putrule` in `\putrectangle` kot tudi za `\plot`, `\circulararc`, `\ellipticalarc`, `\grid`, `\frame` in nekatere druge. Z ukazom

```
\setdots
```

preidemo na pikčaste (ponujena razdalja med pikami je 5pt), z ukazom

```
\setdashes
```

na črtkane (ponujena dolžina črtic in presledkov je 5pt) in z ukazom

```
\setsolid
```

na scela izvlečene črte. Za ukazom `\setdots` lahko navedemo še razdaljo med pikami, za `\setdashes` pa dolžino črtic in presledkov:

```
\setdots <razdalja_pik> \setdashes <dolžina_črtic>
```

Z ukazom

```
\setdashpattern <dc_1,dp_1,dc_2,dp_2,...>
```

definiramo lasten vzorec, kjer bo

dc_1 dolžina prve črtice,

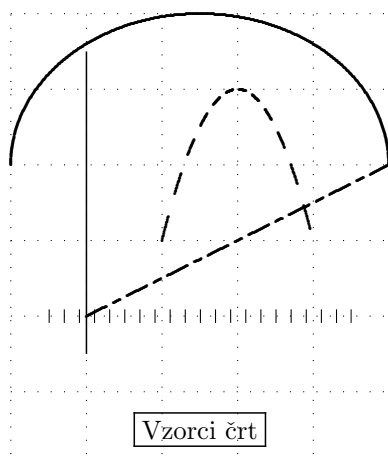
dp_1 dolžina prvega presledka,

dc_2 dolžina druge črtice,

dp_2 dolžina drugega presledka, ...,

periodično vzdolž črte ali krivulje, dokler pač gre. Lahko se tudi ne izide in v takem primeru se pač zadnja perioda temu primerno odreže.

Primer uporabe novih ukazov naj bo naslednja slika.



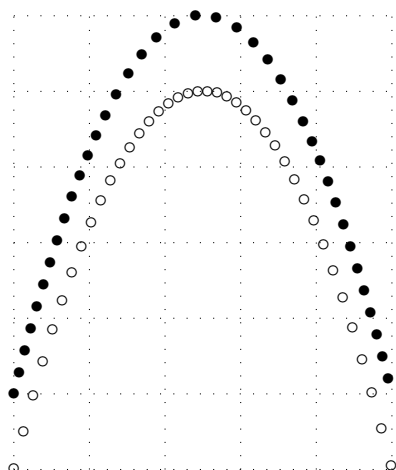
```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from -1 to 5
\linethickness 1.8mm \setdots <2mm>
\putrule from 0.5 1 to 4.5 1
\setsolid \linethickness 0.2mm
\putrule from 1 0.5 to 1 4.5
\setdashes <2mm> \setplotsymbol ({\tenrm.})
\setquadratic \plot 2 2 3 4 4 2 /
\setdashpattern <3mm,1mm,1mm,1mm>
\setlinear \plot 1 1 5 3 / \setsolid
\ellipticalarc axes ratio 5:4 180
degrees from 5 3 center at 2.5 3
\put {\frame <1mm> {Vzorci \v crt}} at 2.5 -0.5
\endpicture
```

Vzorčaste krivulje z ukazom `\plot` lahko dobimo tudi z uporabo ukaza

`\plotsymbolspacing razdalja`

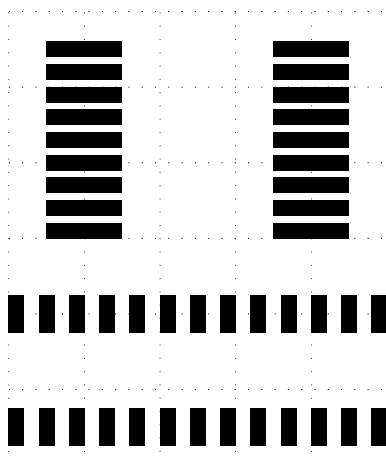
s katerim dobimo simbole, definirane s `\setplotsymbol`, postavljene vzdolž krivulje približno na razdalji *razdalja*. Ponujena vrednost za `\plotsymbolspacing` je 0.4pt.

Pri krivulji, ki jo dobimo z uporabo `\setquadratic` pri majhnem številu točk, lahko pride do precejšnje neenakomernosti, kar je ilustrirano na sliki s parabolama (○), ki je podana le s tremi točkami, in parabolo (●), ki je podana gosteje.



```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from -1 to 5
\plotsymbolspacing 3mm
\setplotsymbol ({\circ})
\setquadratic
\plot 0 -1 2.5 4 5 -1 /
\setplotsymbol ({\bullet})
\plot 0.0 0.0 0.5 1.8 1.0 3.2 1.5 4.2 2.0 4.8
2.5 5.0 3.0 4.8 3.5 4.2 4.0 3.2 4.5 1.8 5.0 0.0 /
\endpicture
```

Z hkratno uporabo ukazov `\linethickness`, `\plotsymbolspacing`, `\setdots` in `\putrule` lahko dosežemo zanimive vzorce. Pri tem je treba vedeti, da $\text{\LaTeX}$ postavlja zapolnjen pravokotnik s stranicama, ki sta določeni z `\linethickness` in `\plotsymbolspacing`, vzdolž vodoravne ali navpične črte, tako na gosto, kot velewa *razdalja-pik* pri `\setdots`.

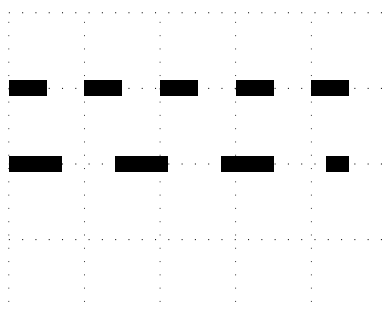


```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from -1 to 5
\plotsymbolspacing 2mm \setdots <4mm>
\linethickness 5mm
\putrule from 0 1 to 5 1
\putrule from 0 -0.5 to 5 -0.5
\plotsymbolspacing 2mm \setdots <3mm>
\linethickness 10mm
\putrule from 1 2 to 1 4.7
\putrule from 4 2 to 4 4.7
\endpicture
```

Pri uporabi ukaza `\setdashes` se rado zgodi, da se vzorec ne izide lepo. V takem primeru lahko sliko izboljšamo, če uporabimo ukaz

```
\setdashesnear <približna_dolžina_črtic> for <celotna_dolžina_črte>
```

Tedaj bo $\text{\PCTE}$ sam razporedil črtice na približno *približna_dolžina_črtic* vzdolž *celotna_dolžina_črte*. Praktični primer:



```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\linethickness 2mm \setdashes <7mm>
\putrule from 0 2 to 4.5 2
\setdashesnear <5mm> for <45mm>
\putrule from 0 3 to 4.5 3
\endpicture
```

$\text{\PCTE}$ celo omogoča, da krivulji izračuna njeno dolžino, kar lahko upoštevamo pri vzorcih. To naredi ukaz oblike

```
\findlength {ukazi_za_krivuljo}
```

ki nam vrne celotno dolžino krivulje `\totalarclength`, ki jo lahko s pridom uporabimo v kasnejšem ukazu `\setdashesnear`.

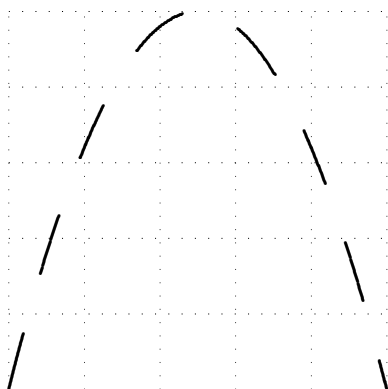
Podobno uporabljamo ukaz

```
\setdotsnear <približen_razmik_pikic> for <celotna_dolžina_črte>
```

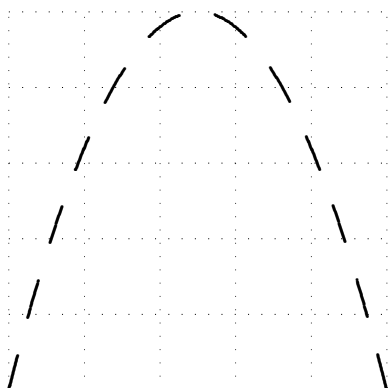
v primeru, ko se postavljanje točk vzdolž krivulje ne izide.

Lahko pa tudi izrabimo `\totalarclength` in definiramo svoj vzorec s `\setdashpattern`. Več primerov za to najdemo v [2].

Oglejmo si kot primer kar parabolo. Prvič se namreč črtkanje parabole ni lepo izšlo, drugič pa uporabimo prej opisan postopek. Rezultat je očitno boljši.



```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 5
\setplotsymbol ({\tenrm.})
\setdashes <8mm>
\setquadratic
\plot "parabola.dat"
\endpicture
```



```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 5
\setplotsymbol ({\tenrm.})
\setquadratic
\findlength {\plot "parabola.dat"}
\setdashesnear <5mm> for <\totalarclength>
\plot "parabola.dat"
\endpicture
```

Pri zgornjih dveh slikah je bila kvadratna funkcija tabelirana s korakom 0.5 cm. Podatki zanjo so shranjeni na datoteki *parabola.dat*. Zakaj toliko točk, ko pa je parabola določena že s tremi? Razlog je v tem, da tedaj ne bi dobili z ukazom `\setdashes` enako dolgih črtic vzdolž parabole.

Opisani postopek lahko uporabimo tudi pri daljicah, pa tudi pri krožnicah in elipsah, če pač želimo imeti čim popolnejšo sliko.

Kotiranje

Včasih je treba na sliki tudi kaj kotirati. To bi lahko naredili z ukazi `\arrow` in ustreznim besedilom. Da pa nam ni treba na roko pisati ukazov za vse to, je dobro vedeti, da nam $\text{\P\TeX}$ omogoča vodoravno in navpično kotiranje tudi enostavneje.

Ukaz za to je

```
\betweenarrows {besedilo} from  $x_1$   $y_1$  to  $x_2$   $y_2$ 
```

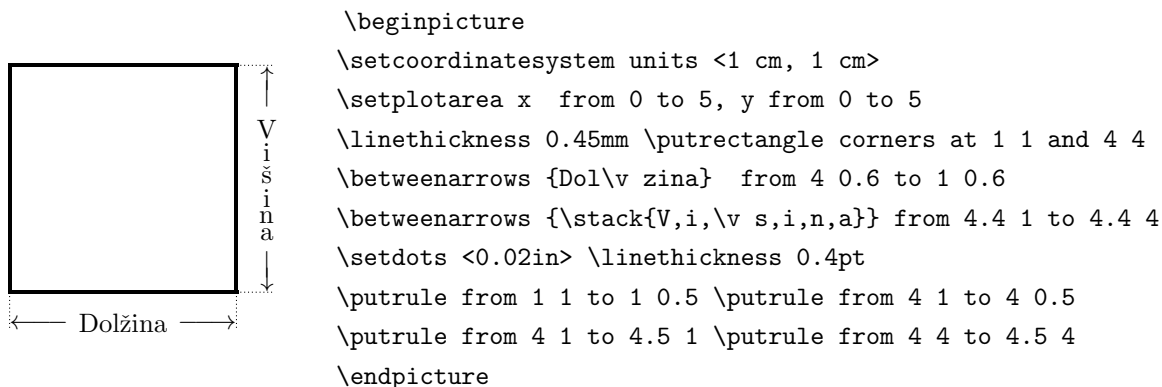
Najsplošnejši ukaz, ki ga morda sem ter tja potrebujemo, je oblike

```
\betweenarrows {besedilo} [ $p_x p_y$ ] [ $x_{odmik}$ ,  $y_{odmik}$ ] from  $x_1$   $y_1$  to  $x_2$   $y_2$ 
```

Pri tem je bodisi $x_1 = x_2$ v primeru navpičnega kotiranja ali pa $y_1 = y_2$ pri vodoravnem kotiranju. Parametri p_x, p_y , x_{odmik} in y_{odmik} imajo isti pomen kot pri ukazu `\put`.

Debelina črt je pri kotiranju fiksirana. Ker $\text{\P\TeX}$ pravzaprav povleče črte med *besedilo* in puščici $\leftarrow$ in $\rightarrow$ oziroma $\downarrow$ ter $\uparrow$, se lahko včasih zgodi, da prehodi črt v puščice niso najlepši. Takrat poskusimo sliko izboljšati z majhnim premikom kotirnih puščic.

Paziti je treba tudi na to, da *besedilo* ni previsoko ali preširoko glede na dolžino, ki jo kotiramo.



Senčenje

S $\text{P}\text{T}\text{E}\text{X}$ om lahko senčimo pravokotnike pa tudi like, ki jih sami definiramo. Pri tem lahko izbiramo simbol, s katerim senčimo in gostoto senčenja. Dobro je vedeti, da senčenje lahko zelo upočasni obdelavo slike. Ukaz za izbiro simbola za senčenje je

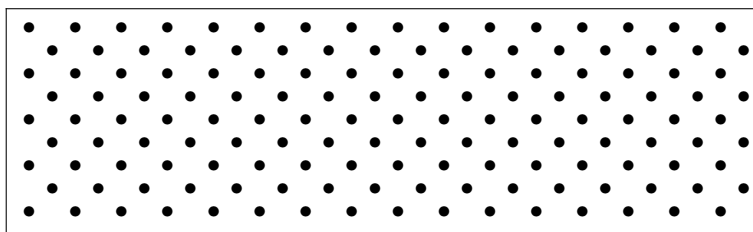
```
\setshadesymbol ({simbol})
```

Punujeni *simbol* je pika iz nabora `\fiverm`, ponujena vodoravna oziroma navpična razdalja med simboli, *gostota_mreže*, pa je 5pt. Po želji lahko med `setshadesymbol` in `(simbol)` vrinemo še dopolnilo

```
<\epsilon_{lev}, \epsilon_{des}, \epsilon_{sp}, \epsilon_{zg}>
```

Pri tem pomenijo po vrsti ϵ_{lev} , ϵ_{des} , ϵ_{sp} , ϵ_{zg} , koliko se s senčenjem lahko približamo levemu, desnemu, spodnjemu, zgornjemu robu senčenega polja. Te količine navedemo v veljavnih TEX ovih merskih enotah. Za razdaljo 0 enot pišemo samo črko z. Če katere od navedenih količin ne navajamo, pišemo samo vejice.

Način senčenja je tak:



Gostoto mreže pri senčenju nastavimo z ukazom

```
\setshadegrid span <gostota_mreže>
```

Najlažje je senčiti pravokotnike, dobljene z ukazi `\putrectangle`, `\rectangle`, `\putbar`, `\frame` ali `\sethistograms`. Z ukazom

```
\shaderectangleson
```

vkjučimo njihovo senčenje, z

```
\shaderectanglesoff
```

pa senčenje pravokotnikov izključimo. Primer: z ukazi

```

\beginpicture

\setcoordinatesystem units <1 cm, 1 cm>

\setplotarea x from 0 to 15, y from 0 to 3

\setshadesymbol ({\$-$}) \setshadegrid span <0.12in>

\shaderectangleson

\putrectangle corners at 0 0 and 3 3

\setshadesymbol ({\$+$}) \setshadegrid span <0.08in>

\putrectangle corners at 4 0 and 7 3

\setshadesymbol ({\tenrm.}) \setshadegrid span <0.04in>

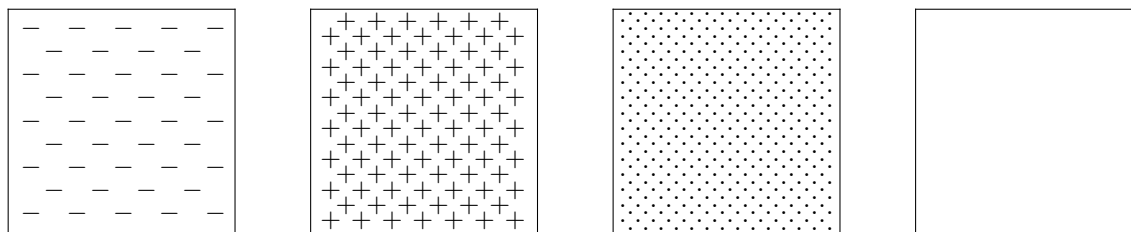
\putrectangle corners at 8 0 and 11 3

\shaderectanglesoff \putrectangle corners at 12 0 and 15 3

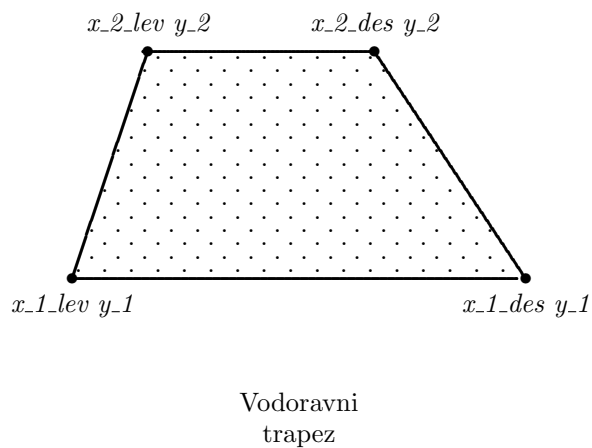
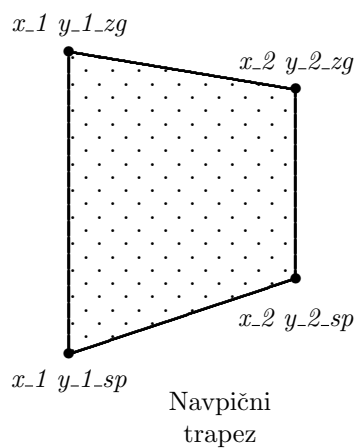
\endpicture

```

dobimo naslednjo sliko:



P_TE_X pozna preprost postopek za senčenje trapezov. Ukaz `\vshade` uporabljamo za senčenje navpičnih, `\hshade` pa za senčenje vodoravnih trapezov. Pri navpičnih trapezih sta stranici vzporedni z osjo y , pri vodoravnih pa z osjo x .



Ukaz za senčenje navpičnih trapezov je oblike

```
\vshade x_1 y_1-sp y_1-zg x_2 y_2-sp y_2-zg /
```

za senčenje vodoravnih trapezov pa oblike

```
\hshade y_1 x_1-lev x_1-des y_2 x_2-lev x_2-des /
```

Robove trapezov je vselej treba narisati posebej, če jih seveda želimo imeti. Pri tem mora biti pri navpičnem senčenju

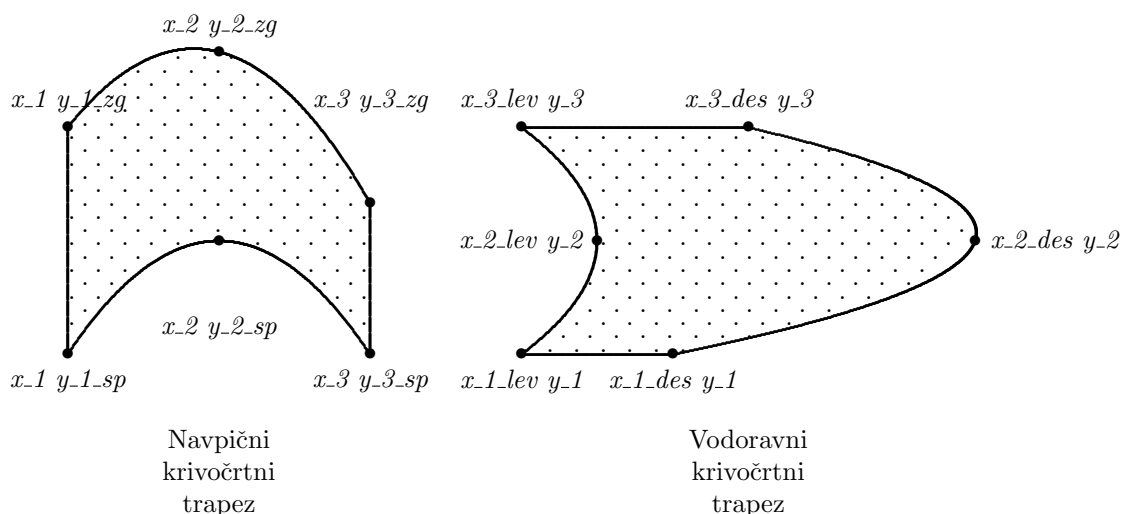
$$x_1 < x_2, y_1-sp \leq y_1-zg, y_2-sp \leq y_2-zg$$

oziroma

$$y_1 < y_2, x_1-lev \leq x_1-des, x_2-lev \leq x_2-des$$

pri vodoravnem senčenju.

Kar je bilo doslej povedano, velja, če velja ukaz `\setlinear`. Če pa velja ukaz `\setquadratic`, potem lahko senčimo navpične oziroma vodoravne krivočrtnne trapeze, ki so omejeni od spodaj in od zgoraj oziroma z leve in desne strani s parabolami. V tem primeru je treba podati tri točke za spodnjo (oziroma za levo) in tri točke za zgornjo (oziroma za desno) parabolo.



Ukaz za senčenje navpičnih krivočrtnih trapezov je oblike

```
\vshade x_1 y_1-sp y_1-zg x_2 y_2-sp y_2-zg x_3 y_3-sp y_3-zg /
```

in za senčenje vodoravnih krivočrtnih trapezov oblike

```
\hshade y_1 x_1-lev x_1-des y_2 x_2-lev x_2-des y_3 x_3-lev x_3-des /
```

Pri tem morajo veljati omejitve

$$x_1 < x_2 < x_3, y_1-sp \leq y_1-zg, y_2-sp \leq y_2-zg, y_3-sp \leq y_3-zg$$

oziroma

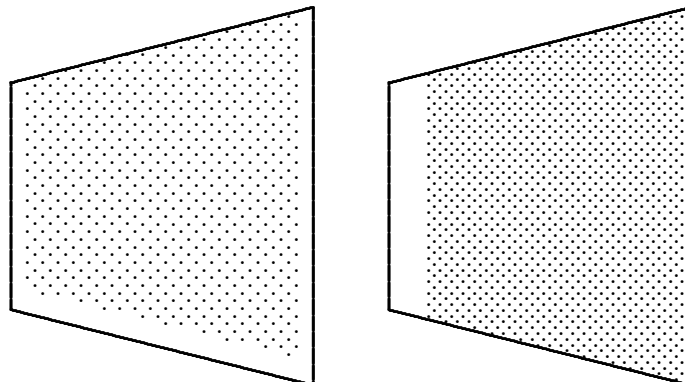
$$y_1 < y_2 < y_3, x_1-lev \leq x_1-des, x_2-lev \leq x_2-des, x_3-lev \leq x_3-des$$

Pri senčenju trapezov ima $\text{P}_\text{CTE}\text{X}$ na razpolago tudi dodatek, s katerim se robovom senčenega polja po potrebi približamo ali oddaljimo. Če tega dodatka ni, uporabi $\text{P}_\text{CTE}\text{X}$ svoje mere. Dodatek je oblike

< ε_{lev} , ε_{des} , ε_{sp} , ε_{zg} >

Pri tem imajo nastopajoče količine enak pomen kot pri tovrstnem dodatku ukaza `\setshadesymbol`. Opisani dodatek vrinemo pri ukazih `\vshade` in `\hshade` za prvo trojko koordinat.

Primer uporabe: deloma osenčena trapeza



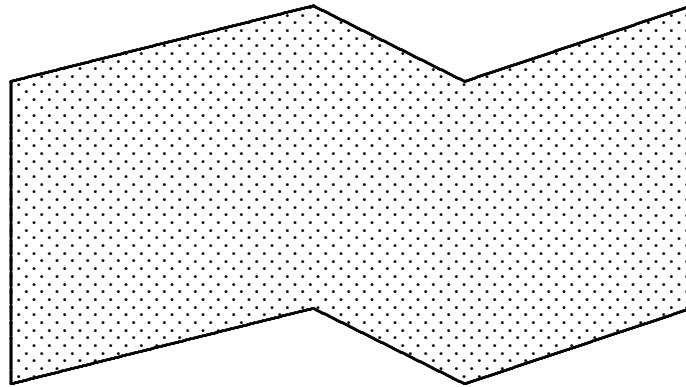
dobimo z ukazi:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 1 to 10, y from 0 to 5
\setplotsymbol ({\tenrm.})
\setshadesymbol ({\tenrm.})
\setshadegrid span <0.04in>
\plot 1 1 1 4 5 5 5 0 1 1 /
\vshade 1 1 4 <2mm,2mm,3mm,z> 5 0 5 /
\setcoordinatesystem units <1 cm, 1 cm> point at -5 0
\setplotsymbol ({\tenrm.})
\setshadesymbol ({\tenrm.})
\setshadegrid span <0.03in>
\plot 1 1 1 4 5 5 5 0 1 1 /
\vshade 1 1 4 <0.2in,,,> 5 0 5 /
\endpicture
```

Senčimo lahko tudi več trapezov iste sorte hkrati, če se pravilno stikajo. Lomljeni črti, ki jih omejujeta spodaj (levo) in zgoraj (desno), morata imeti lomišča pri istih abscisah (ordinatah). Z dodatki

< ε_{lev} , ε_{des} , ε_{sp} , ε_{zg} >

dosežemo, da sence trapezov neopazno prehajajo s trapeza na trapez. Če velja `\setlinear`, vrivamo ta dodatek za vsako trojko koordinat za `\vshade` in `\hshade`, razen za zadnjo, pri `\setquadratic` pa za prvo, tretjo, peto... trojko koordinat, razen za zadnjo. Primer bo pojasnil vse. Slika



je nastala z naslednjim opisom:

```
\beginpicture

\setcoordinatesystem units <1 cm, 1 cm>

\setplotarea x from 0 to 9, y from 0 to 5

\setplotsymbol ({\tenrm.}) \setshadesymbol ({\tenrm.})

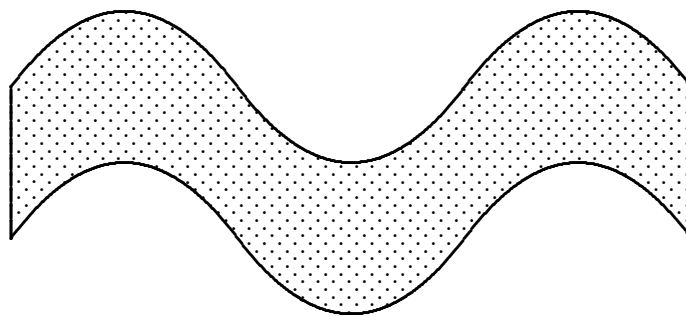
\setshadegrid span <0.04in>

\plot 0 0 0 4 4 5 6 4 9 5 9 1 6 0 4 1 0 0 /

\vshade 0 0 4 <z,z,,> 4 1 5 <z,z,,> 6 0 4 <z,z,,> 9 1 5 /

\endpicture
```

Pa še en primer senčenja krivočrtnih trapezov:



Za to potreben opis je naslednji:

```
\beginpicture

\setcoordinatesystem units <1 cm, 1 cm>

\setplotarea x from 0 to 9, y from 1 to 5
```

```

\setplotsymbol ({\tenrm.}) \setshadesymbol ({\tenrm.})

\setshadegrid span <0.04in>

\plot 0 2 0 4 / \plot 9 2 9 4 /

\setquadratic

\plot 0 4 1.5 5 3 4 4.5 3 6 4 7.5 5 9 4 /

\plot 0 2 1.5 3 3 2 4.5 1 6 2 7.5 3 9 2 /

\vshade 0 2 4 <z,z,,> 1.5 3 5

        3 2 4 <z,z,,> 4.5 1 3

        6 2 4 <z,z,,> 7.5 3 5

        9 2 4 /

\endpicture

```

Včasih je dobro vedeti, da predelava slike, kjer uporabljamo `\setquadratic` in senčenje poteka relativno počasi, zlasti na starejših osebni računalnikih s procesorjem tipa 286. Prav tako se hitro zgodi, da zmanjka spomina. Ena od možnih rešitev je uporaba debele verzije $\text{\TeX}$ a. Tedaj lahko traja predelava precej dolgo, če ni dovolj RAMa. Najbolje se po dosedanjih izkušnjah izkažejo v kategoriji osebni računalnikov tisti s procesorji tipa 386 ali 486 in kakšnih 4 MB RAMa z uporabo debele verzije $\text{\TeX}$ a, ki je narejen za tovrstne procesorje (na primer 3c-beta8).

Če smo torej šibkeje strojno in programsko opremljeni, je bolje uporabljati kar senčenje v linearnem načinu, toda s primerno ozkimi trapezi.

Zasuki

S $\text{\PCTEX}$ om lahko nekatere gradnike na sliki tudi zasukamo za izbrani kot φ okrog izbrane točke $x_s y_s$. Ukaz je oblike

```
\startrotation by cos  $\varphi$  sin  $\varphi$  about  $x_s y_s$ 
```

Pri tem se moramo potruditi in poiskati $\cos \varphi$ in $\sin \varphi$ v decimalni obliki. Po ukazu `\startrotation` se bodo zasukali tisti deli slike, ki so dobljeni z ukazi `\plot`, `\circulararc`, `\ellipticalarc` in `\arrow`. Točke v ukazih `\put` in `\multiput` se tudi zasukajo, napisi pa se le vzporedno premaknejo spremljajoč te točke. Z ukazi `\vshade` in `\hshade` senčeni liki se tudi zasukajo, znak, s katerim senčimo, se ne zasuka. Prav tako je z pravokotniki in črtami, ki so dobljeni z ukazi `\putrectangle` in `\putrule`. Sukanje izklopimo z ukazom

```
\stoprotation
```

Primer zasuka ellipse: ukazi

```

\beginpicture

\setcoordinatesystem units <1 cm, 1 cm>

\setplotarea x from 0 to 9, y from 0 to 5

\setplotsymbol ({\tenrm.}) \put {\$ \circ \$} at 5 3

```

```

\ellipticalarc axes ratio 3:1 360 degrees from 8 3 center at 5 3

\setdashes <0.75mm>

\startrotation by 0.866 0.5 about 5 3

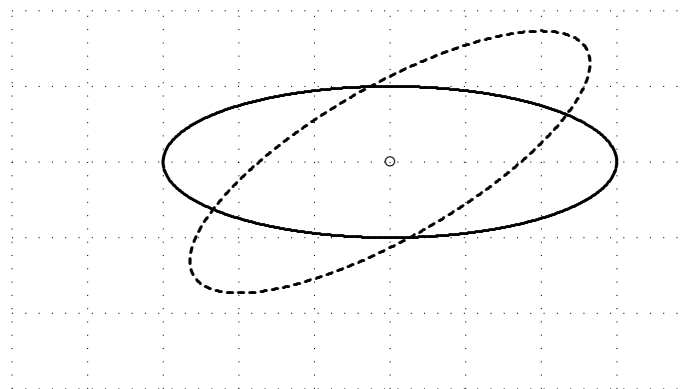
\ellipticalarc axes ratio 3:1 360 degrees from 8 3 center at 5 3

\stoprotation

\endpicture

```

ustvarijo sliko



Sukanje lahko po `\stoprotation` ponovno vklopimo. Če so parametri zasuka isti kot pri prvem vklopu, zadošča že ukaz `\startrotation` (brez parametrov).

Histogrami

Za nazoren prikaz številskih rezultatov, zlasti v statistiki, uporabljajo histograme. P_TE_X pozna preprosto metodo za njihovo izdelavo, in sicer si lahko privoščimo navpične in vodoravne histograme.

Navpične histograme najavimo z ukazom

```
\sethistograms
```

Za tem ukazom stoječi ukaz `\plot` vsebuje podatke o stolpcih za histogram. Prvi par števil sta koordinati točke prvega, to se pravi najbolj levega stolpca, ki leži na osi histograma. Naslednji pari števil pa so koordinate desnih oglišč stolpcev. Ordinate so lahko tudi negativne. Osenčene histograme dobimo s predhodnim ukazom `\shaderectangleson`. Ukaz `\sethistograms` lahko prekličemo z `\setlinear` ali `\setquadratic`.

Z P_TE_Xovimi ukazi

```

\beginpicture

\setcoordinatesystem units <1 cm, 0.75 cm>

\setplotarea x from 0 to 10, y from 0 to 6

\setplotsymbol ({\tenrm.})

```

```

\setshadegrid span <0.03in>

\shaderectangleson \sethistograms

\plot 0 0 1 2 2 3 4 5 4.5 3 5 2 7 3 9 1 10 2 /

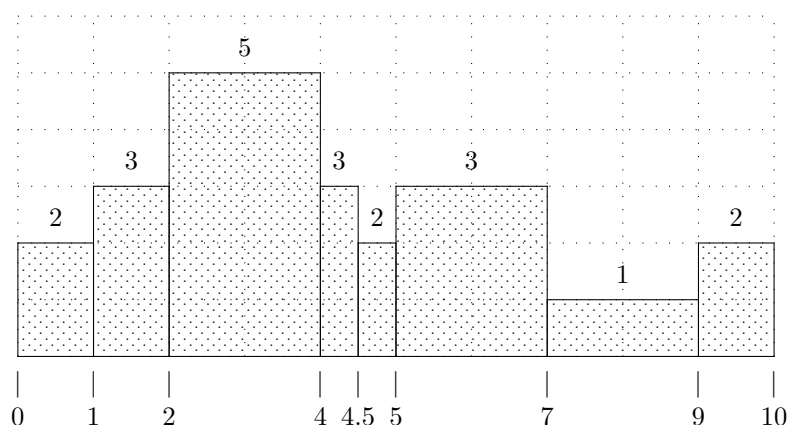
\multiput {$|}$ at 0 -0.5 1 -0.5 2 -0.5 4 -0.5 4.5 -0.5 5 -0.5 7 -0.5 /
9 -0.5 10 -0.5 /

\put {0} [t] at 0 -0.9 \put {1} [t] at 1 -0.9 \put {2} [t] at 2 -0.9
\put {4} [t] at 4 -0.9 \put {4.5} [t] at 4.5 -0.9 \put {5} [t] at 5 -0.9
\put {7} [t] at 7 -0.9 \put {9} [t] at 9 -0.9 \put {10} [t] at 10 -0.9
\put {2} [b] at 0.5 2.3 \put {3} [b] at 1.5 3.3 \put {5} [b] at 3 5.3
\put {3} [b] at 4.25 3.3 \put {2} [b] at 4.75 2.3 \put {3} [b] at 6 3.3
\put {1} [b] at 8 1.3 \put {2} [b] at 9.5 2.3

\endpicture

```

dobimo naslednji histogram:



Vodoravne oziroma navpične diagrame z isto širino lahko rišemo z ukazom

```
\setbars breadth <širina> baseline at x=x_zacetka
```

oziroma z ukazom

```
\setbars breadth <širina> baseline at y=y_zacetka
```

za katerim stoji ukaz `\plot` s koordinatami točk, do katerih pravokotniki segajo. Pri tem pomeni *širina* širino pravokotnikov, *x_zacetka* in *y_zacetka* pa absciso oziroma ordinato, od katere vlečemo pravokotnike v levo oziroma navzgor.

Ukaz `\setbars` pozna še nekatere neobvezne dodatke: dodatek

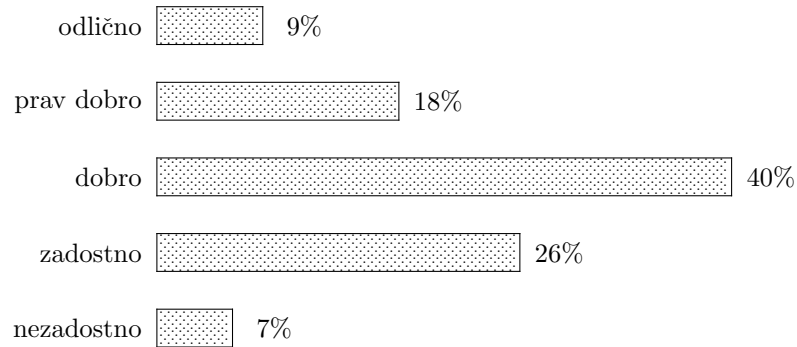
```
<x_odmik,y_odmik>
```

ki ga postavimo med `\setbars` in `breadth` ima podoben pomen kot pri ukazu `\put`, stolpce po potrebi odmakne za izbrani vrednosti *x_odmik* in *y_odmik*. Dodataka

```
baselabels ([p_x p_y] < x_s, y_s>)
```

```
endlabels ([p_x p_y] < x_s, y_s>)
```

omogočata dodajanje napisov na začetek in konec stolpcev. Samo vsebino napisov dodamo v ukazu `\plot` v navednicah " " za koordinatami. Parametra p_x, p_y povesta poravnavo, parametra x_s in y_s pa odmik napisov od stolpcev. Uporabo pojasnimo na primeru iz šole.



Sliko smo dobili z ukazi:

```
\beginpicture
\setcoordinatesystem units <2 mm, 1 cm>
\setplotarea x from -10 to 45, y from 0 to 4.5
\setplotsymbol ({\tenrm.})
\setshadegrid span <0.02in>
\shaderectangleson
\setbars <0pt,2.5mm> breadth <5mm> baseline at x = 2
baselabels ([r] <-2mm,2.5mm>) endlabels ([l] <2mm,2.5mm>)
\plot 9 4 "odlično" "9%"
      18 3 "prav dobro" "18%"
      40 2 "dobro" "40%"
      26 1 "zadostno" "26%"
      7 0 "nezadostno" "7%" /
\endpicture
```

Oglejmo si še navpični histogram take sorte s področja proizvodnje neke surovine v neki tovarni po letih. Sistem ukazov

```
\beginpicture
\setcoordinatesystem units <1 cm, 1mm>
```

```

\setplotarea x from 0 to 10, y from -6 to 60

\setplotsymbol ({\tenrm.}) \setshadegrid span <0.02in>

\shaderectangleson

\setbars <-2.5mm,0pt> breadth <5mm> baseline at y = 0

baselabels ([B] <-2.5mm,-6mm>)

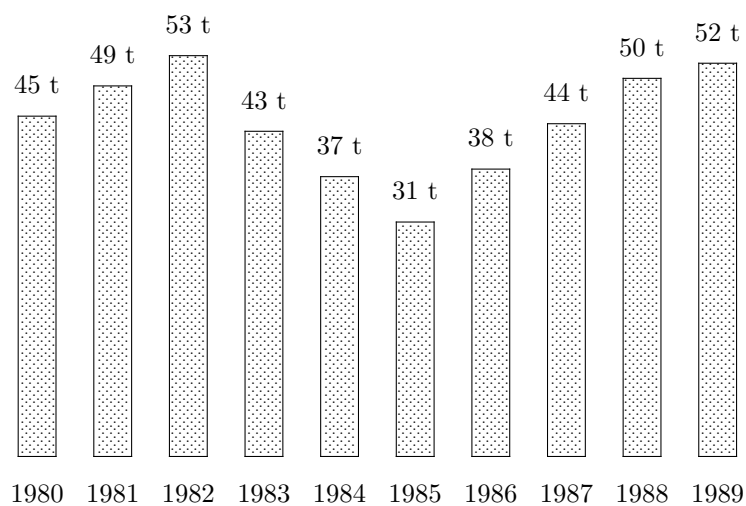
endlabels ([B] <-2.5mm,3mm>)

\plot 1 45 "1980" "45 t"
      2 49 "1981" "49 t"
      3 53 "1982" "53 t"
      4 43 "1983" "43 t"
      5 37 "1984" "37 t"
      6 31 "1985" "31 t"
      7 38 "1986" "38 t"
      8 44 "1987" "44 t"
      9 50 "1988" "50 t"
     10 52 "1989" "52 t" /

\endpicture

```

nam ustvari naslednjo sliko:



Uporabnik pa bo sam presodil, katero varianto risanja histogramov bo uporabil v konkretnem primeru.

Označevanje osi

Za izris in označevanje osi pozna $\text{\P\TeX}$ ukaz

```
\axis
```

s celo paleto dodatkov. Ukaz `\axis` je v tesni zvezi z zadnjim, pred njim stoječim ukazom `\setplotarea`, s katerim definiramo risalno polje. Ukaz `\axis` mora obvezno imeti podatek, kje naj potegne osi. Za os lahko izberemo levo, desno, spodnjo ali zgornjo stranico risalnega polja, tako da ustrezno izberemo eno od dopolnil

```
left    right    bottom    top
```

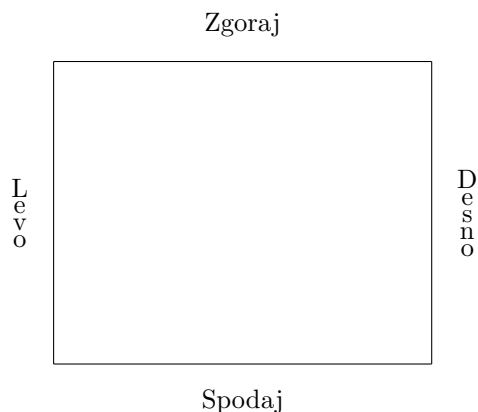
Ukazov `\axis` je lahko več. Napise k osem dobimo z dopolnilom

```
label {besedilo}
```

Naštevanje dopolnil moramo zaključiti s presledkom in z znakom `/`.

Primer: dajmo risalno polje okvir in dodajmo napise.

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\axis left label {\stack{L,e,v,o}} /
\axis right label {\stack{D,e,s,n,o}} /
\axis bottom label {Spodaj} /
\axis top label {Zgoraj} /
\endpicture
```

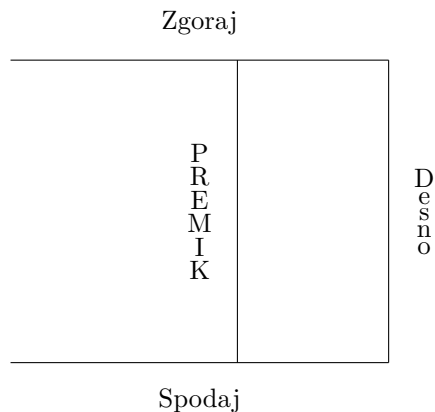


Dopolnili

```
shiftedto x= $x_{premik}$  in shiftedto y= $y_{premik}$ 
```

v ukazu `\axis` povesta premik osi: leve ali desne za x_{premik} , spodnje ali zgornje za y_{premik} . Primer:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\axis left label {\stack{P,R,E,M,I,K}}
shiftedto x=3 /
\axis right label {\stack{D,e,s,n,o}} /
\axis bottom label {Spodaj} /
\axis top label {Zgoraj} /
\endpicture
```

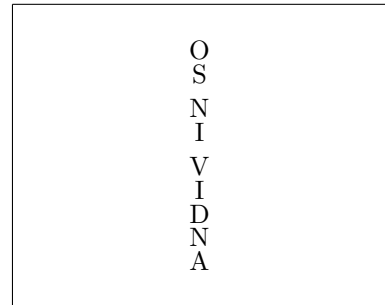


Osi so lahko vidne ali nevidne. Ustrezni dopolnili sta

visible in invisible

Začetna oblika je visible. Primer uporabe:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\axis left /
\axis right
label {\stack{0,S,,N,I,,,V,I,D,N,A}}
shiftedto x=2 invisible /
\axis bottom /
\axis top /
\endpicture
```



Najbolj pestro dopolnilo ukaza `\axis` je `ticks`, s katerim razporedimo črtice in druge oznake ob oseh. Ukaz `ticks` pozna še veliko poddopolnil.

Poddopolnili

out in in

obrnete črtice ven oziroma v risalno polje. Začetna varianta je `out`.

S poddopolnili

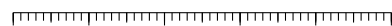
short long length <dolžina>

lahko izbiramo dolžino črtic. Vgrajena oblika je `long`. Z `length` pa lahko sami izberemo dolžino črtic.

Os označimo z n črticami, od začetka do konca osi, s poddopolnilom

quantity n

Da se nam ne bi teh poddopolnil nabralo preveč, si oglejmo primer. Merilce



je nastalo z ukazi:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 0
\axis bottom ticks
quantity 6 short quantity 51 length <1.3mm> quantity 11 /
\endpicture
```

Debelino črtic lahko nastavimo z poddopolnilom

width <širina>

Dosežemo lahko tudi, da so črtice podaljšane v risalno polje. To naredimo s poddopolnilom

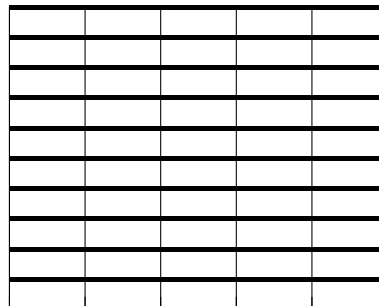
`andacross`

Nasprotno naredi

`butnotacross`

Ta izbira je začetna. Tako imamo novo možnost za izdelavo mrež s $\text{P}\text{T}\text{E}\text{X}$ om. Primer:

```
\beginpicture
\setcoordinatesystem units <1 cm, 1 cm>
\setplotarea x from 0 to 5, y from 0 to 4
\axis bottom ticks in
  andacross quantity 6 /
\axis left ticks in width <0.55mm>
andacross quantity 11 /
\endpicture
```



Črtice so lahko označene ali ne. To dosežemo s poddopolnili

`numbered` `unlabeled` oziroma `withvalues` $n_1 n_2 \dots n_k$ /

Kje naj oznake stojijo, povemo z dodatkom

`from` a `to` b `by` d

da bo pri črtici s koordinato a pisalo a , $a + d$ pri črtici s koordinato $a + d$ in tako naprej; z dodatkom

`at` $a_1 a_2 \dots$ /

pa naštejemo koordinate, pri katerih naj bodo oznake. Pozor! $\text{P}\text{T}\text{E}\text{X}$ ima svoje muhe: če uporabljamo decimalna števila v dodatku `from ... to ... by ...`, morajo a , b in d imeti **enako število decimalk** za decimalno piko. Z zahtevo

`unlabeled` dobimo neoznačene črtice.

Svoje oznake $n_1, n_2, \dots$ vnesemo z opisom

`withvalues` $n_1 n_2 \dots$ /

Primer: s preprostim opisom

```
\beginpicture
\setcoordinatesystem units <8 mm, 1 cm>
\setplotarea x from 0 to 17, y from 0 to 4
\axis left /
\axis right /
\axis bottom ticks in length <0pt> andacross quantity 18
  butnotacross
  withvalues c d e f g a h c$^1$ d$^1$
```

```

e$^1$ f$^1$ g$^1$ a$^1$ h$^1$ c$^2$ d$^2$ e$^2$ /

from 0.5 to 16.5 by 1.0 /

\axis top ticks in length <2.5cm> width <4mm>

at 1 2 4 5 6 8 9 11 12 13 15 16 /

withvalues cis dis fis gis ais cis$^1$ dis$^1$

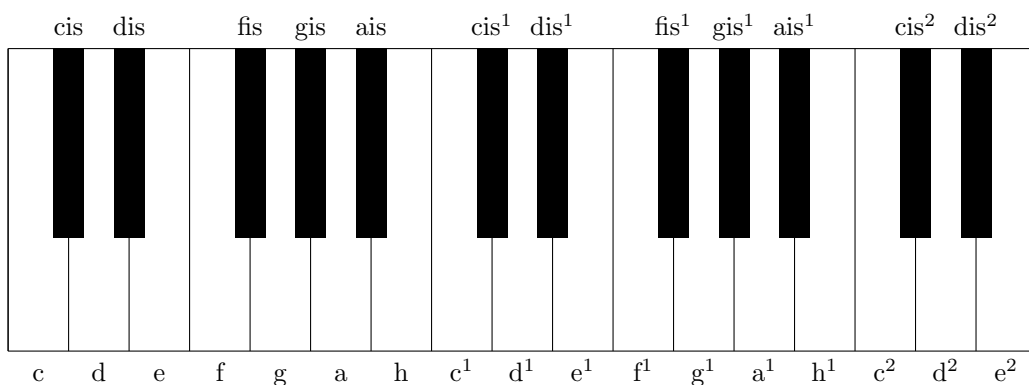
fis$^1$ gis$^1$ ais$^1$ cis$^2$ dis$^2$ /

at 1 2 4 5 6 8 9 11 12 13 15 16 / /

\endpicture

```

ustvarimo čisto spodobno klaviaturo:



P_TE_X pozna tudi logaritemsko razdelitev koordinatnih osi. Dopolnila `from ... to ... by ...` in `quantity` ukaza `\axis` dasta enakomerno ali linearno razdelitev. Poddopolnilo

```
logged
```

dopolnila `ticks` ukaza `\axis` pove, da prehajamo na logaritemsko razdelitev. Prehod nazaj na linearno razdelitev dobimo z

```
unlogged
```

Z `logged` dosežemo oznake in črtice, ki se nanašajo na koordinate x_k v `at in from ... to ... by ...`, namesto pri koordinati x_k pri njenem desetiškem logaritmu: $\log_{10} x_k$.

Prgišče ukazov

```

\beginpicture

\setcoordinatesystem units <2 cm, 0 mm>

\setplotarea x from 0 to 6, y from 0 to 0

\axis bottom ticks logged numbered

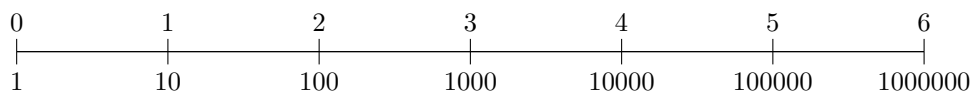
at 1 10 100 1000 10000 100000 1000000 / /

\axis top ticks    numbered    from 0 to 6 by 1 /

```

`\endpicture`

nam pomaga narisati hkratno linearno in logaritemsko skalo:



V splošnem je struktura ukaza `\axis` taka:

```

\axis bottom ali top ali left ali right
    shiftedto x= x_premik ali shiftedto y= y_premik
    visible ali invisible
    label {besedilo}
    ticks
        out ali in
        long ali short ali length <dolžina>
        width <širina>
        butnotacross ali andacross
        unlabeled ali numbered ali withvalues  $n_1 n_2 \dots n_k$  /
        logged ali unlogged
        quantity  $n$  ali from  $a$  to  $b$  by  $d$  ali at  $a_1 a_2 \dots$  / /

```

Pri tem pomeni veznik *ali* izključujoči *ali*: samo en dodatek, zapisan zgoraj v posamezni vrstici, lahko izberemo.

Za primer si pogledajmo iz radiotehnike primer grafa v logaritemski mreži. Opis naslednje slike je tak:

```

\font\debeli cmr12 scaled \magstep2

\beginpicture

\setcoordinatesystem units <6 cm, 4 cm>

\setplotarea x from -1 to 1.05, y from 0 to 1.95

\axis bottom label {Du\v senje v kablu (dB)}

    ticks andacross logged

width <0.6pt> numbered from 0.1 to 0.4 by 0.1
                    from 0.6 to 1.0 by 0.2
                    from 2 to 4 by 1
                    from 6 to 10 by 2

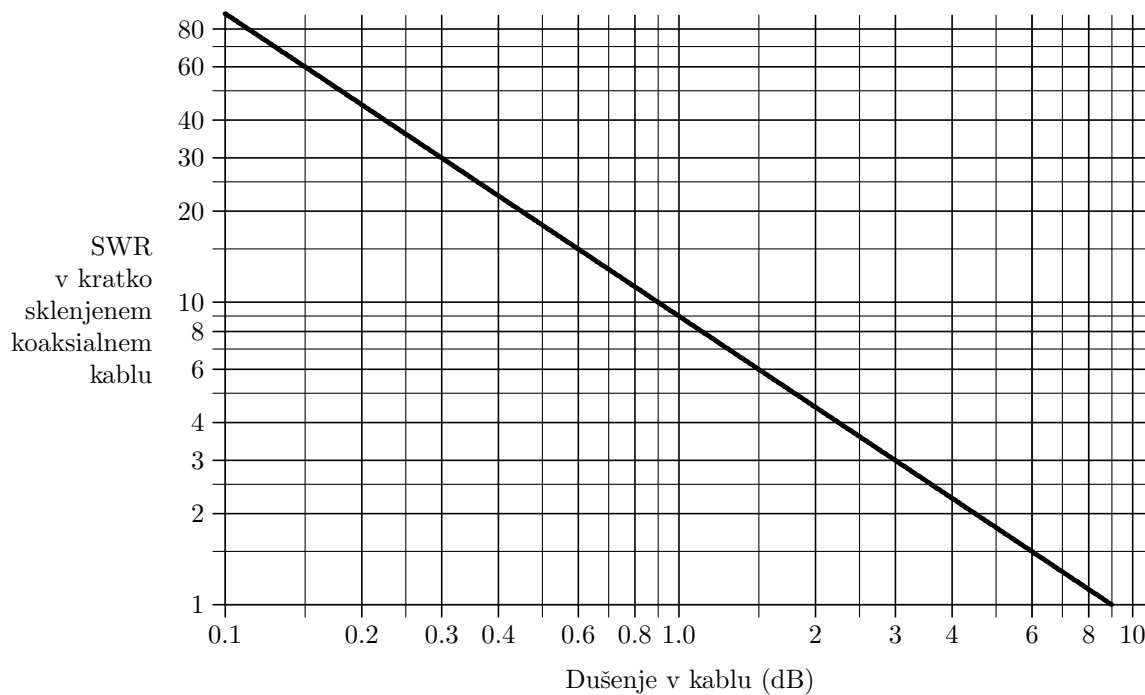
width <0.4pt> unlabeled from 0.15 to 0.25 by 0.10

```

```

from 0.5 to 0.9 by 0.2
from 1.5 to 2.5 by 1.0
from 5 to 9 by 1 /
\axis left label {\lines[r]{SWR\cr v kratko\cr sklenjenem\cr
koaksialnem\cr kablu}}
ticks and across logged
width <0.6pt> numbered from 1 to 4 by 1
from 6 to 10 by 2
from 20 to 40 by 10
from 60 to 80 by 20
width <0.4pt> unlabeled from 1.5 to 2.5 by 1.0
from 5 to 9 by 2
from 15 to 25 by 10
from 50 to 70 by 20 /
\setplotsymbol ({\debeli.})
\plot -1 1.95424250944 0.954242509439 0 /
\endpicture

```



Nekatere izboljšave

Rekli smo že, da poteka obdelava slike kar nekaj časa, čim smo pri opisu slike uporabili ukaze `\plot`, `\arrow`, `\circulararc` in `\ellipticalarc`, ker bi z njimi hoteli narediti pestro sliko, z večjim številom gradnikov. P_TEX pozna ukaz

```
\savelinesandcurves on "datoteka"
```

s katerim shranimo krivulje, ki nastanejo s prej navedenimi ukazi v zapis *datoteka*. Ukaz prekličemo z

```
\donsavelinesandcurves
```

Z ukazom `\savelinesandcurves`

lahko shranimo v posebnem formatu podatke o vseh točkah krivulj, tako da nam jih v naslednji obdelavi ni več treba znova obdelovati. Ukaze za risanje krivulj lahko tedaj ukinemo, na primer tako, da jih spremenimo z znakom `%` v komentar. Namesto ukinjenih ukazov napišemo

```
\replot "datoteka"
```

Pri tem je treba paziti na naslednje: vsako okolje za sliko mora imeti drugo ime *datoteka*. Črte, ki nastanejo s `\sethistrograms`, se ne shranijo s `\savelinesandcurves`. V zapisu *datoteka* so shranjeni podatki o krivuljah, ki so bile narejene z istim simbolom za risanje.

Na začetku ima lahko *datoteka* po potrebi kratek *komentar*, ki ga dobimo z ukazom

```
\writesavefile {komentar}
```

ki ga postavimo za ustreznim `\savelinesandcurves`. Pred komentar doda ukaz znak `%`, tako da na izris krivulj nič ne vpliva.

Primer: z ukazi

```
\font\debeli cmr12 scaled \magstep2

\beginpicture

\setcoordinatesystem units <5 mm, 5 mm>

\setplotarea x from 0 to 10, y from 0 to 6

\setquadratic

\savelinesandcurves on "vaja41a.plt"

\writesavefile {Cela parabola pri vaji 41}

\setplotsymbol ({\debeli .})

\plot 0 0 5 6 10 0 /

\donsavelinesandcurves

\setdots

\setplotsymbol ({\tenrm.})

\savelinesandcurves on "vaja41b.plt"
```

```

\writesavefile {Pik\v casta parabola pri vaji 41}

\plot 0 6 5 0 10 6 /

% \replot "vaja41a.plt"

% \replot "vaja41b.plt"

\endpicture

```

ustvarimo sliko in datoteki vaja41a.plt in vaja41b.plt, kjer se shranijo podatki o parabolah, ki sta povlečeni z neprekinjeno oziroma s pikčasto črto. Z ukazi

```

\font\debeli cmr12 scaled \magstep2

\beginpicture

\setcoordinatesystem units <5 mm, 5 mm>

\setplotarea x from 0 to 10, y from 0 to 6

\setquadratic

% \savelinesandcurves on "vaja41a.plt"

% \writesavefile {Cela parabola pri vaji 41}

\setplotsymbol ({\debeli .})

% \plot 0 0 5 6 10 0 /

% \donsavelinesandcurves

\setdots

\setplotsymbol ({\tenrm.})

% \savelinesandcurves on "vaja41b.plt"

% \writesavefile {Pik\v casta parabola pri vaji 41}

% \plot 0 6 5 0 10 6 /

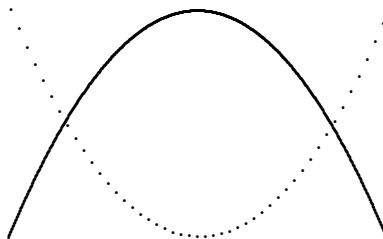
\replot "vaja41a.plt"

\replot "vaja41b.plt"

\endpicture

```

dobimo isto sliko, namreč



kjer pa P_TCT_EX pri drugi varianti vnese že pripravljene podatke o obeh parabolah.

Ukaza `\put` in `\multiput` omogočata ponavljanje slik na različnih mestih. Namesto 7-kratne uporabe ukaza `\circulararc` pri naslednji sliki lahko isto precej hitreje dobimo z ukazi, v katerih definiramo delne slike:

```
\beginpicture

\setcoordinatesystem units <5 mm, 5 mm>

\setplotarea x from -4.8 to 4.8, y from -4.5 to 4.5

\setplotsymbol ({\tenrm.})

\multiput {\beginpicture

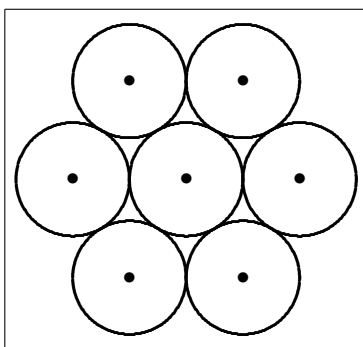
  \circulararc 360 degrees from 1.5 0 center at 0 0

  \put {$\bullet$} at 0 0

\endpicture} at

0 0 3 0 1.5 2.598 -1.5 2.598 -3 0 -1.5 -2.598 1.5 -2.598 /

\grid 1 1 \endpicture
```



Vendar je včasih dobro vedeti, da ima P_TCT_EX tukaj svojo logiko. Na krivulje, ki jih dobimo s `\plot`, `\circulararc` in `\ellipticalarc` pri veljavnem ukazu `\setlinear` ali `\setquadratic`, gleda P_TCT_EX tako, kot da so brez končnih mer, za razliko od na primer pravokotnikov, ki imajo neke mere. Zato v delnih slikah utegne P_TCT_EX krivulje drugače postavljati, kot smo pričakovali.

Za primer si pogledjmo primer ukazov, med katerimi v `\setcoordinatesystem` ni podana referenčna točka, ki jo lahko podamo z dodatkom `point at`:

```
\beginpicture
```

```

\setcoordinatesystem units <1 mm, 1 mm>

\setplotarea x from -40 to 40, y from -10 to 20

\multiput {\beginpicture

  \putrectangle corners at -10 -10 and 10 10

  \circulararc 360 degrees from 8 0 center at 0 0

  \ellipticalarc axes ratio 2:1 360 degrees from 0 20 center at 0 15

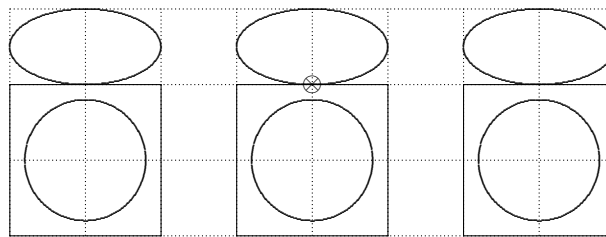
\endpicture} at -30 -10 *2 30 0 /

\put {\otimes} at 0 0 \setdots <0.02in> \grid 8 3

\endpicture

```

ki dajo rezultat, ki smo ga morda pričakovali:



Sedaj pa ukaze nekoliko spremenimo tako, da dodamo referenčno točko:

```

\beginpicture

\setcoordinatesystem units <1 mm, 1 mm> point at 20 10

\setplotarea x from -60 to 30, y from -30 to 10

\put {\bullet} at 20 10

\multiput {\odot} at -30 -10 *2 30 0 /

\multiput {\beginpicture

  \circulararc 360 degrees from 8 0 center at 0 0

  \ellipticalarc axes ratio 2:1 360 degrees from 0 20 center at 0 15

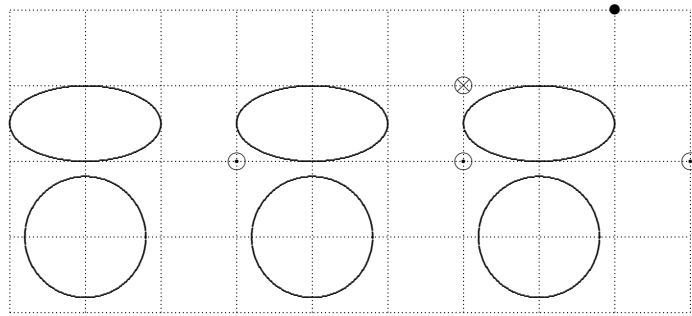
\endpicture} at -30 -10 *2 30 0 /

\put {\otimes} at 0 0 \setdots <0.02in> \grid 9 4

\endpicture

```

in dobimo morda nekoliko nepričakovano reč:

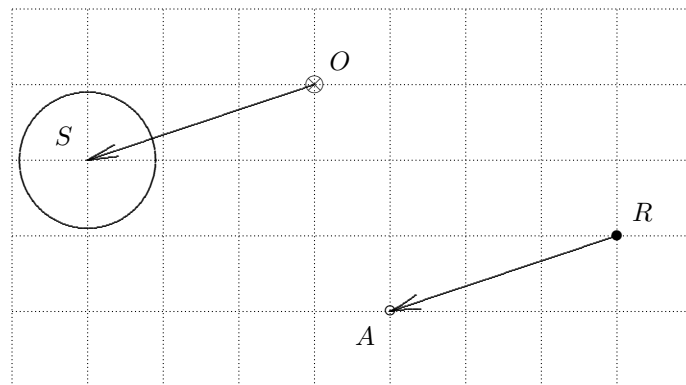


Pri tem smo z znakom $\otimes$ označili koordinatno izhodišče, z $\bullet$ referenčno točko in z $\odot$ točke, kjer smo morda pričakovali središča krogov.

Če postavljamo delne slike, ki nimajo končnih mer, s `\put in \multiput` v točki A , se opisi slik nanašajo na koordinatni sistem z izhodiščem v točki S tako, da je

$$\overrightarrow{OS} = \overrightarrow{OA} - \overrightarrow{OR} = \overrightarrow{RA}$$

Pri tem je O koordinatno izhodišče nazadnje vpeljanega koordinatnega sistema in R referenčna točka,



ki ju dobimo z ukazi

```
\beginpicture
\setcoordinatesystem units <1 mm, 1 mm> point at 40 -20
\setplotarea x from -40 to 50, y from -40 to 10
\put {$\bullet$} at 40 -20
\put {\beginpicture
\circulararc 360 degrees from 9 0 center at 0 0
\endpicture} at 10 -30
\put {$\otimes$} at 0 0 \put {$\circ$} at 10 -30
\put {$O$} [lb] at 2 2 \put {$A$} [rt] at 8 -32
\put {$R$} [lb] at 42 -18 \put {$S$} [rb] at -32 -8
```

```

\arrow <4mm> [0.2,0.5] from 40 -20 to 10 -30

\arrow <4mm> [0.2,0.4] from 0 0 to -30 -10

\setdots <0.02in> \grid 9 5

\endpicture

```

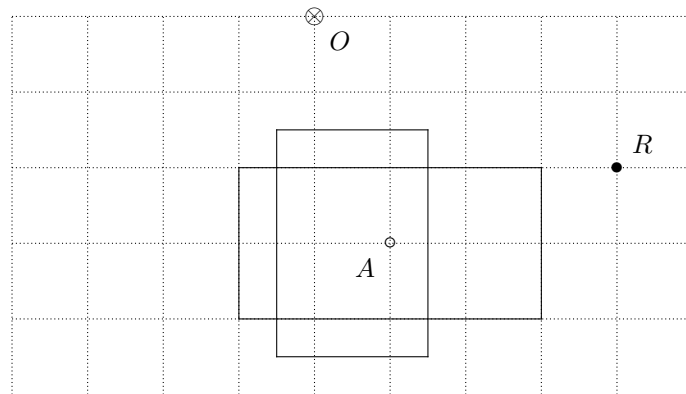
V primeru, ko ima delna slika končne dimenzije, na primer pravokotniki, dobljeni s `\putrectangle`, so razmere nekoliko bolj zapletene. Koordinatno izhodišče sistema, v katerem so opisani gradniki slike, bo v točki S , ki smo jo izračunali po prej opisanem postopku, če podatke o delni sliki shranimo v registra, recimo `\xreg` in `\yreg` in nato dopolnimo z `at` takole:

```
[B1] <\xreg,\yreg> at
```

Da se $\text{\texttt{P}\text{\texttt{I}\text{\texttt{C}\text{\texttt{T}\text{\texttt{E}\text{\texttt{X}}}} zapomne podatke, uporabimo tedaj namesto ukaza `\endpicture` pri opisu delne slike ukaz$

```
\endpicturere save <\xreg,\yreg>
```

Za primer vzemimo sliko



ki je nastala z opisom

```

\beginpicture

\setcoordinatesystem units <1 mm, 1 mm> point at 40 -20

\setplotarea x from -40 to 50, y from -50 to 0

\put {$\bullet$} at 40 -20

\put {\beginpicture

\putrectangle corners at -10 -20 and 10 10

\putrectangle corners at -15 -15 and 25 5

\endpicture} at 10 -30

\put {$\otimes$} at 0 0 \put {$\circ$} at 10 -30

\put {$O$} [lt] at 2 -2 \put {$A$} [rt] at 8 -32

```

```
\put {$R$} [lb] at 42 -18
```

```
\setdots <0.02in> \grid 9 6
```

```
\endpicture
```

Z nekaterimi popravki dobimo sliko, v kateri se bolje znajdemo. S $\text{\TeX}$ ovim ukazom `\newdimen` definiramo razdalji `\xreg` in `\yreg`, ki sta koristni pri postavitvi delne slike. Opis

```
\newdimen\xreg \newdimen\yreg
```

```
\beginpicture
```

```
\setcoordinatesystem units <1 mm, 1 mm> point at 40 -20
```

```
\setplotarea x from -50 to 50, y from -40 to 10
```

```
\put {$\bullet$} at 40 -20
```

```
\put {\beginpicture
```

```
\putrectangle corners at -10 -20 and 10 10
```

```
\putrectangle corners at -15 -15 and 25 5
```

```
\endpicturesave <\xreg,\yreg> [B1] <\xreg,\yreg> at 10 -30
```

```
\put {$\otimes$} at 0 0 \put {$\circ$} at 10 -30
```

```
\put {$O$} [lb] at 2 2 \put {$A$} [rt] at 8 -32
```

```
\put {$R$} [lb] at 42 -18
```

```
\put {$S$} [rb] at -32 -8
```

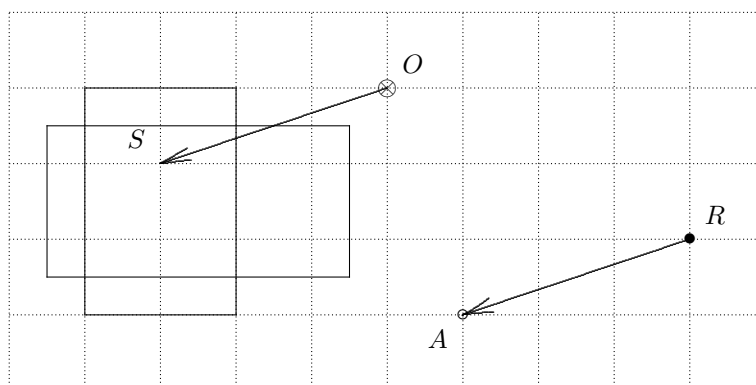
```
\arrow <4mm> [0.2,0.5] from 40 -20 to 10 -30
```

```
\arrow <4mm> [0.2,0.4] from 0 0 to -30 -10
```

```
\setdots <0.02in> \grid 10 5
```

```
\endpicture
```

ustvari naslednjo sliko:



Bolj elegantno lahko delamo z delno sliko, če definiramo poseben ukaz zanjo, na primer za vzmet. Za vzmet je dovolj definirati en njen zavoje z ukazom `\vzmet`. Podatki o točkah so spravljani na datoteki `vaja49.dat`.

```
\newdimen\xreg \newdimen\yreg
\def\vzmet{\hbox{%
\beginpicture \plot "vaja49.dat" \endpicturesave <\xreg,\yreg> }}
\beginpicture
\setcoordinatesystem units <1 mm, 1 mm>
\setplotarea x from -10 to 10, y from -140 to 30
\setplotsymbol ({\tenrm.})
\setshadegrid span <0.02in>
\vshade -10 24 28 10 24 28 /
\multiput {\vzmet} [Bl] <\xreg,\yreg> at 0 10 *19 0 -6.283187 /
\plot 0 -106.283187 0 -115 / \plot 0 19.283187 0 24 /
\putrule from -10 24 to 10 24
\circulararc 360 degrees from 0 -115 center at 0 -125
\ellipticalarc axes ratio 5:2 -180 degrees from 10 -125
center at 0 -125
\setdots <0.03in>
\ellipticalarc axes ratio 5:2 180 degrees from 10 -125
center at 0 -125
\endpicture
```

En zavoje je dovolj na gosto tabeliran od točke $(0, 3)$ do točke $(0, 3 + 2\pi)$, in sicer kot krivulja

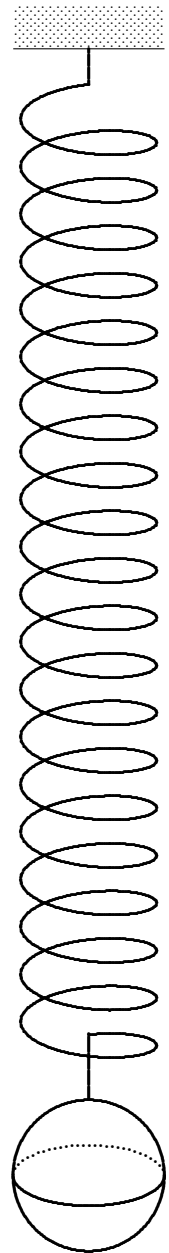
$$\vec{r} = (a \sin t, b \cos t + ht), \quad a = 9, b = 3, h = 1$$

za $t \in [0, 2\pi + \Delta t]$ s korakom $\Delta t = \pi/20$.

Preizkus bi takoj pokazal, da `PiCTEX` izdela sliko na ta način precej hitreje, kot če bi tabelirali vseh 20 zavojev vzmeti in jo nato risali z ukazom `\plot`.

Če se neka delna slika večkrat ponavlja v drugih slikah, se jo lahko shrani v `TEX`ov register z ukazoma `\newbox` in `\setbox`. Potem jo postavimo na pravo mesto z ukazi `\put`, `\multiput`, `\copy` in `\box`. Ukaz `\copy` lahko uporabimo večkrat, ukaz `\box` pa register izprazni, kar se izplača uporabiti pri zadnji uporabi.

Za register si lahko sami izmislimo ime s samimi črkami, na primer `\slikaA`. Kako to uporabljamo, si oglejmo na primeru.



Denimo, da se nam v besedilu pogosto ponavlja znak . Definiramo ga takole:

```
\newbox\slikaA
\setbox\slikaA=\hbox{%
\beginpicture
```

```

\setcoordinatesystem units <1 mm, 1 mm>

\setplotarea x from -5 to 5, y from 0 to 4

\grid 10 4

\endpicture}

```

Znak  je med besedilo vstavljen z ukazom

```
\copy\slikaA
```

V sliko bi ga namestili z ukazom

```
\put{\copy\slikaA} ali \multiput{\copy\slikaA} at
```

Register `\slikaA` pa izpraznimo z ukazom `\box\slikaA` in znaka  se ne bi več dalo uporabljati v nadaljevanju istega besedila.

Basic in programiranje krivulj

Operacijski sistem DOS 5.0 vsebuje tudi *qbasic*, s katerim lahko hitro napišemo program, ki nam bo izračunal točke, ki jih rabimo v $\text{P}\text{T}\text{E}\text{X}$ ovih ukazih `\plot "datoteka"`, `\multiput "datoteka"` in pri podobnih ukazih, ki dopuščajo podatke, ki so zbrani na neki datoteki. Podatki za vzmet na prejšnji strani so na datoteki `vaja49.dat`. Ker basic pozna tudi eksponentni zapis števil, na primer $5.5\text{E}-12$, $\text{P}\text{T}\text{E}\text{X}$ pa ne, moramo z majhno zvijačo poskrbeti, da bodo števila v *datoteka* pisana v standardni decimalni obliki. Z ukazom

```
OPEN "0", #1, "datoteka"
```

to datoteko odpremo, rezultata `xx` in `yy` pa pišemo vanjo z ukazom

```
PRINT #1, xx, yy
```

Za vzmet je program v basicu tak:

```

10 OPEN "0", #1, "vaja49.dat"

20 a = 9

30 b = 3

40 h = 1

45 pi = 3.141593

47 s = pi / 20

50 FOR t = 0 TO 2 * pi + s STEP s

60 x = a * SIN(t)

70 y = b * COS(t) + h * t

72 xx = INT(1000 * x + 0.5) / 1000

75 yy = INT(1000 * y + 0.5) / 1000

```

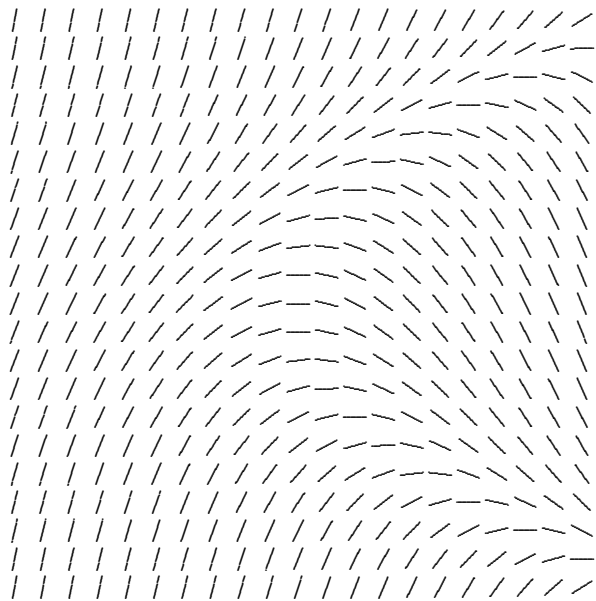
```
80 PRINT #1, xx, yy
```

```
90 NEXT t
```

Tak program je dober tako za qbasic kot tudi za starejši *gwbasic*. Podobno sprogramiramo tudi točke za krivuljo, ki je graf eksplisitno dane funkcije. Lahko naredimo tudi program, ki bo pisal v datoteko cele ukaze, recimo puščice ali črtice. Za primer vzemimo polje smeri diferencialne enačbe

$$y' = \frac{1}{2}y^2 - x$$

Polje smeri je izračunano v točkah $(n/4, m/4)$ za $n, m = 0, \pm 1, \pm 2, \dots, \pm 10$. Na sliki je parabola ena od izoklin, povezuje točke, v katerih je $y' = 0$.



```
\beginpicture
\setcoordinatesystem
  units <1.5 cm, 1.5 cm>
\setplotarea
  x from -2.5 to 2.5,
  y from -2.5 to 2.5
\input vaja50.dat
\setplotsymbol ({\tenrm.})
\setquadratic
\plot 2.5 2.236 0 0 2.5 -2.236 /
\endpicture
```

Na datoteki *vaja50.dat* so podatki o krajših črtic. Datoteko smo naredili z basicom:

```
5 OPEN "0", #1, "vaja50.dat"

10 DEF FNf (x, y) = .5 * y ^ 2 - x

12 DEF FNg (x) = INT(1000 * x + .5 * SGN(x)) / 1000

15 d = .1

20 FOR x = -2.5 TO 2.5 STEP .25

30 FOR y = -2.5 TO 2.5 STEP .25

40 k = FNf(x, y)

50 dx = d / SQR(1 + k ^ 2)

60 dy = d * k / SQR(1 + k ^ 2)

70 x0 = FNg(x - dx)
```

```

75 y0 = FNg(y - dy)

80 x1 = FNg(x + dx)

85 y1 = FNg(y + dy)

90 PRINT #1, "\plot", x0, y0, x1, y1, "/"

100 NEXT y

110 NEXT x

```

Vključitev slik v besedilo

Slike lahko vključimo v besedilo tako, kot je navedeno v [1], lahko pa tudi celotni opis slike na pravem mestu vnesemo z ukazom `\input`. Za začetek naredimo posamezno sliko za $\text{\TeX}$ po vzorcu:

```

\input pictex

\nopagenumbers

\beginpicture

\setcoordinatesystem units <1.5 cm, 1.5 cm>

\setplotarea x from -2.5 to 2.5, y from -2.5 to 2.5

\input vaja50.dat

\setplotsymbol ({\tenrm.})

\setquadratic

\plot 2.5 2.236 0 0 2.5 -2.236 /

\endpicture

\bye

```

Če sliko vnašamo v $\text{\LaTeX}$ ovo besedilo, zberišemo ukaza `\nopagenumbers` in `\bye`, ukazi

```
\input prepictex, \input pictex in \input postpictex
```

pa morajo stati samo pred prvo sliko. Slike lahko usrediščimo z `$$ \dots $$` ali s `\centerline`.

Literatura

- [1] V. Batagelj, B. Golli, *T_EX. Povabilo v T_EX, L^AT_EX, B_IB_TE_X, P_TC_TE_X*, DMFA Slovenije, Ljubljana 1990.
- [2] H. Kopka, *L^AT_EX Erweiterungsöglichkeiten mit einer Einführung in METAFONT*, Addison–Wesley, Bonn in drugje, 1991.
- [3] L. Lamport, *L^AT_EX: A Document Preparation System*, Addison–Wesley, Reading, 1986.
- [4] M. J. Wichura, *The P_TC_TE_X Manual*, T_EX Users Group, Providence, RI, 1987.
- [5] B. Šfiligoj, M. Željko, *P_TC_TE_X CAD. Navodila za uporabo programa*, DMFA Slovenije, Ljubljana 1992.

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.42

RAZPET, Marko

Sedi in riši s PiCTeXom! [Elektronski vir] / Marko Razpet. -2.
izd. - Besedilni podatki. - [Domžale : samozal.], 2004

Način dostopa (URL): <http://www2.arnes.si/~supmrzp/seminar.pdf>.
- Opis temelji na verziji z dne 10.02.2006

ISBN 961-6589-29-6

225022208
