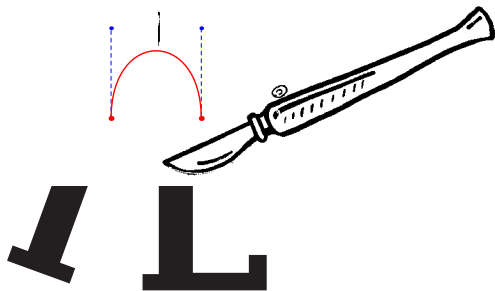


Disseksjon av en font

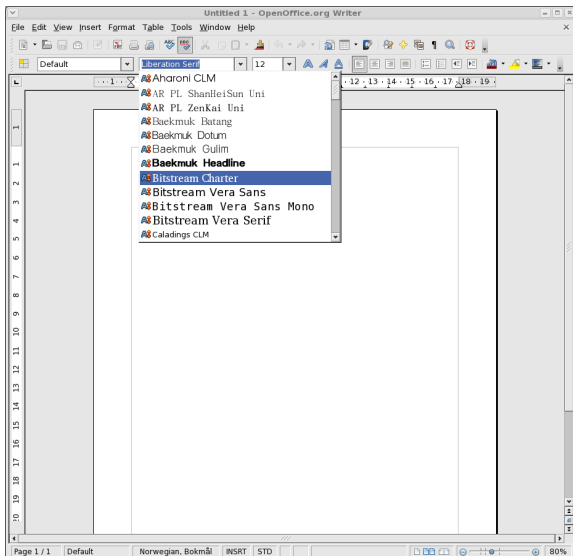
Dag Langmyhr

dag@ifi.uio.no



Alle bruker fonter!

Alle vet hva en font er:



Det finnes tusenvis av fonter:

For bøker:

Garamond

For presentasjoner:

Lucida Fax

Elegante:

Calligra

Morsomme:



... men innmaten er den samme.

Hvorfor?

Det finnes mange eksempler på **vellykket teknologi** som er meget enkel å bruke selv om innmaten er komplisert:

- armbåndsuret
- telefonen
- nettlesere

samt (mener jeg) fonter.

Hva er en font?

En font er en samling bilder som kan plasseres vilkårlige steder på arket eller skjermen.

Geometrisk transformasjon er en stor fordel:

- skalering
- rotering
- vridning
- reversering

P

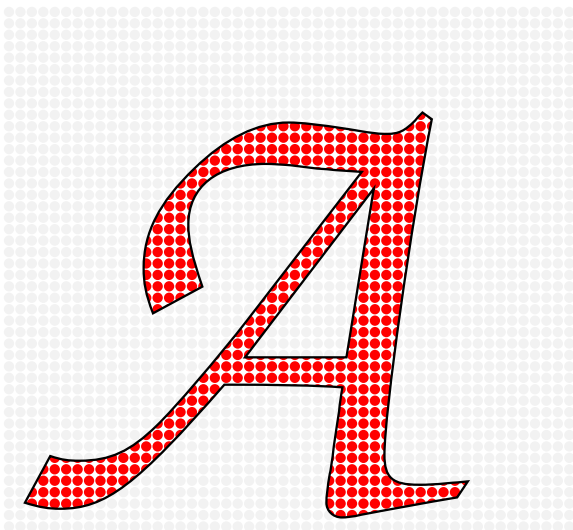
P

P

P

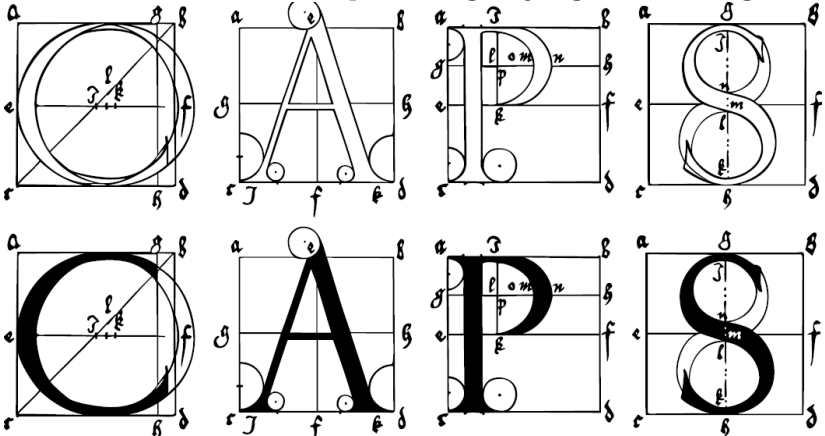
Hva er en font?

Løsningen er
omriss av hvert
tegn på et egnet
format.



Hvordan tegne et godt omriss?

Mange har lenge prøvd å få til dette, som *Albrecht Dürer* rundt år 1500. Men med passer og linjal går det dårlig.



Glatte kurver er en lek!

Løsningen

I 1912 definerte *Sergeï Bernshteïn* en interpolasjonsformel

$$B(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3, t \in [0, 1]$$

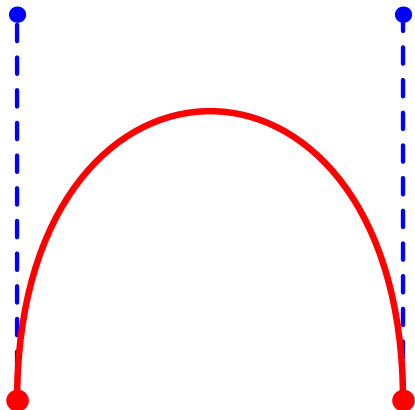
I 1959 fikk *Paul de Casteljou* ideen om å benytte denne formelen til et grafisk formål: Han designet karosseriplater for Citroën.

Kurvene ble kjent som **Bézier-kurver** etter *Pierre Bézier* som lånte ideen og brukte den hos Renault; han publiserte de første artiklene om denne bruken.

Glatte kurver er en lek!

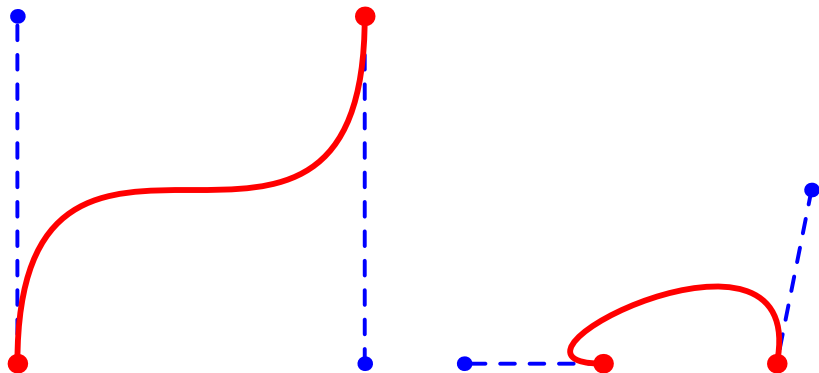
Bézier-kurver er enkle å forstå:

- Vi skal trekke en *pen* kurve fra A til B .
- Fra A skal den starte i retning av kontrollpunktet P_1 .
- Når kurven går mot B , skal den ha retning fra det andre kontrollpunktet P_2 .



Glatte kurver er en lek!

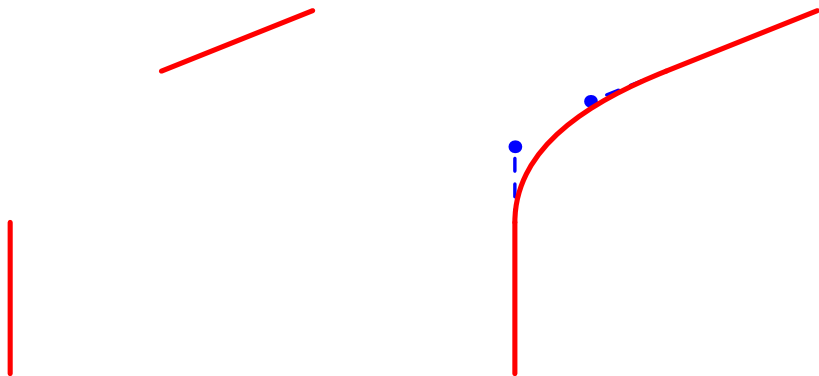
Dette gjelder uansett hvor «rare» kontrollpunkter vi velger:



Glatte kurver er en lek!

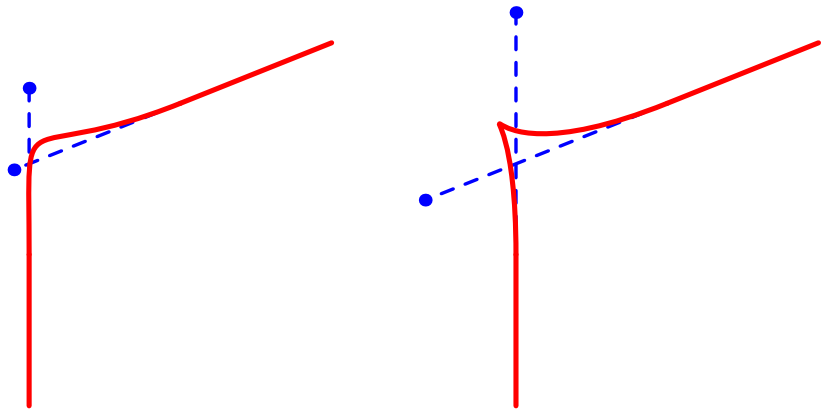
Bézier-kurver har to store fordeler:

- 1 Det er trivielt enkelt å få en glatt overgang mellom to streker:



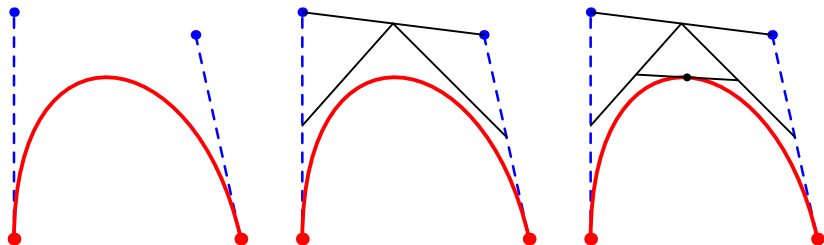
Glatte kurver er en lek!

Men vi må passe litt på hvor vi setter kontrollpunktene:



Glatte kurver er en lek!

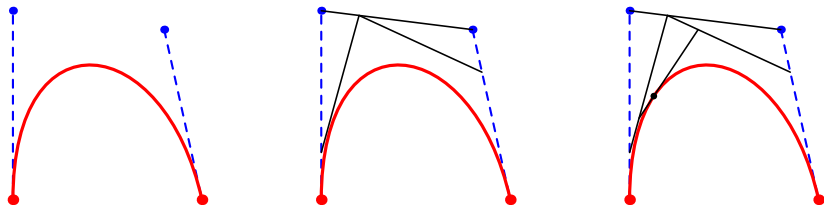
- 1 Bézier-kurver gir glatte overganger.
- 2 Bézier-kurver er lette (og raske!) å tegne:



Dette tar 12 addisjoner og 12 divisjoner.

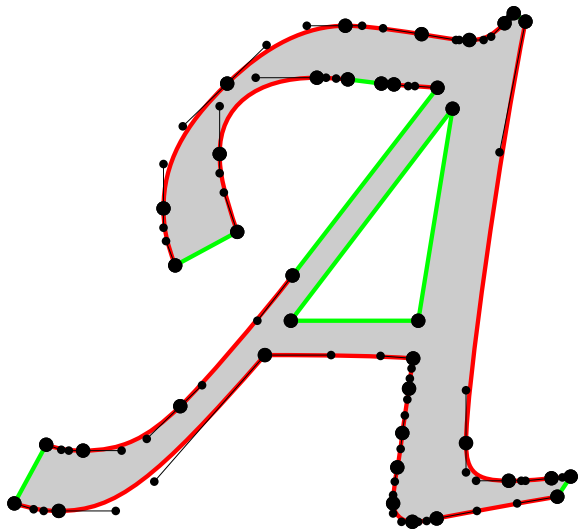
Glatte kurver er en lek!

Det er like lett å finne andre punkter på kurven:



Glatte kurver er en lek!

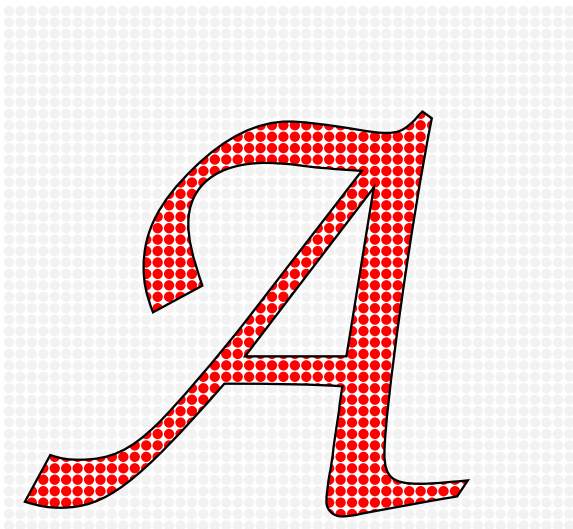
Da har vi
verktøyet vi
trenger: grønne
streker er rette
linjer og røde
streker er
Bézier-kurver.



Glatte kurver er en lek!

Da er problemet
løst.

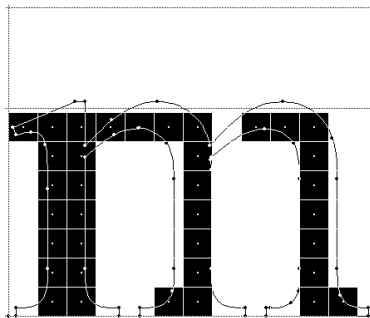
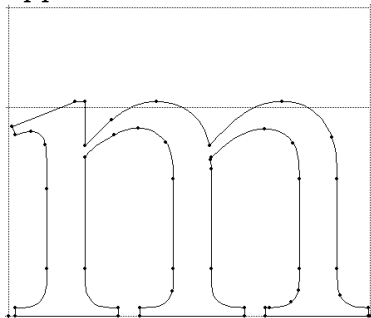
Eller er det det?



Er da problemet løst?

Men hva med avrundingsfeil?

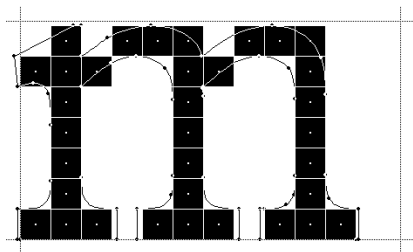
Omrisset benytter reelle tall (double-tall) og skal legges oppå et rutenett med fast størrelse.



Er da problemet løst?

Løsningen er **hinting** som må med i fonten:

- Hvilke streker skal være like tykke?
- Hvilke avstander skal være like store?
- Hvilke elementer skal ikke fjernes uansett hvor små de blir?



Hvordan passer bokstavene sammen?

Passer bokstavene sammen?

W A V E R L Y

W A V E R L Y

Løsningen er å legge inn **utligning** («kerning») som er en avstandsjustering mellom visse bokstavpar:

W A V E R L Y

W A V E R L Y

Utligningen kan beregnes automatisk, men det beste er å gjøre det manuelt. Dette er en *stor* jobb.

Hva med andre spesielle tegnkombinasjoner?

Noen tegnkombinasjoner bør spesialbehandles:

fi fl

De tar seg bedre ut om vi bruker en **ligatur**:

fi fl

Ligaturer samt opplysninger om hvilke tegnkombinasjoner som gir en ligatur, må også med i fonten.



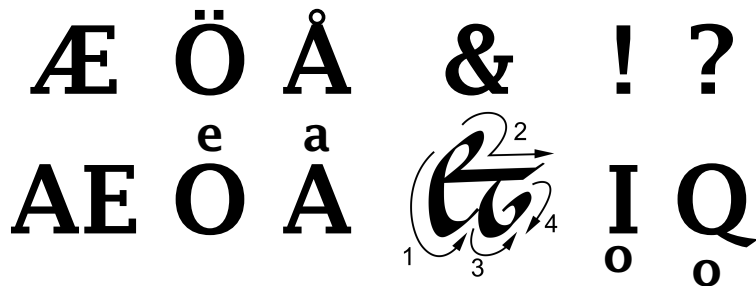
Hva med andre spesielle tegnkombinasjoner?

De fleste fonter tilbyr **fi** og **fl** men noen har mange fler:

Stylistic			Letters	
fi		fh	Æ	æ
fj	ffi	fi	Œ	œ
fl	ffj	fl		
ff	ffl	ff		
		ft		
	ct	st		
		Th		

Hva med andre spesielle tegnkombinasjoner?

Noen ligaturer er blitt så vanlige at vi bare kjenner dem som eget tegn:

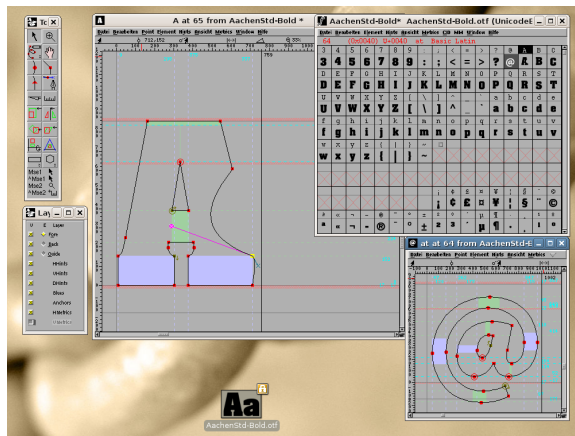


Kan alle lage sin egen font?

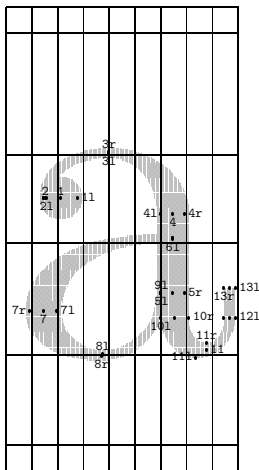
Kan jeg lage min egen font?

Det finnes mange brukbare programmer, blant annet *FontForge*.

Å lage en god font tar flere måneder!



Et alternativ
Folk som liker
L^AT_EX og å jobbe
på en deklarativ
måte, vil kanskje
like METAFONT,
laget av *Donald
Knuth*, mannen
bak T_EX.



3 = 3r + (0, -0.5)
5 = 5r + (-9, 0)
6 = 6l + (0, 0.2)
8 = 8l + (0, -0.5)
9 = 9l + (0, 0)
10 = 10l + (11, 0)
12 = 12l + (-4.5, 0)
13 = 13l + (-4.5, 0)
1r = 2 + (-1, 0)
2r = 2 + (-1, 0)
6r = 6l + (0, 0.3)
9r = 9l + (0, 0)
12r = 12l + (-9, 0)

Takk for meg!

Nå kan dere bruke fonter i vilden sky i tillit til at den innebygde teknologien fungerer som den skal.

