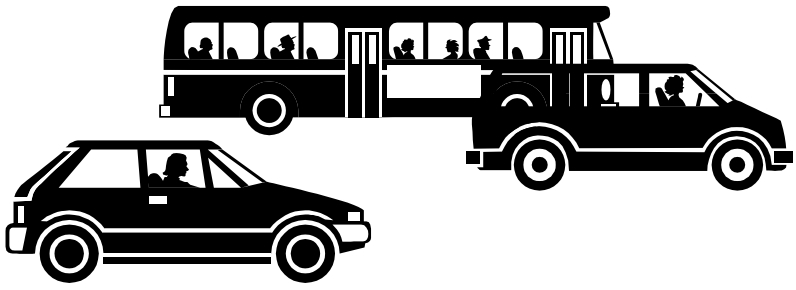
Programvare for trafikkberegninger basert på basiskurvemetoden - versjon pr. 1. april 2005

```
…

nMatrix *xMatrixEstAlphaBeta;
   nMatrix *zMatrix;

   genMatrixX(tCounts->eabDate, nDailyCounts, refYear, &xMatrixEstAlphaBeta,
   &zMatrix);
   (*zMatrix).redim(1, 1);



/*-------------------------------------------------------------------------------
  Estimate traffic curve parameters by means of reduced rank regression
  -------------------------------------------------------------------------------*/

   tcPar = (struct trafficCurveParameters *) malloc(sizeof(struct
   trafficCurveParameters));

   if (estAlphaBeta(tCounts, xMatrixEstAlphaBeta, &tcPar, localMessage)) return(1);
   (*xMatrixEstAlphaBeta).redim(1, 1);

…
```

SAMBA/09/05

Ola Haug
Magne Aldrin



April 2005

Norsk Regnesentral
ANVENDT DATAFORSKNING

NR-notat/NR Note

Tittel/Title: **Programvare for trafikkberegninger basert på basiskurvemetoden – versjon pr. 1. april 2005**

Dato/Date: April
År/Year: 2005
Notat nr: SAMBA/09/05
Note no:

Forfatter/Author: Ola Haug og Magne Aldrin

Sammendrag/Abstract: Basiskurvemetoden er en metode for beregning av trafikkvolum på årsbasis. Implementeringen av basiskurvemetoden utgjøres av to programsystemer.

Hovedmodulen er et Windows-program som beregner trafikkparametere (ÅDT, YDT, …) med usikkerhetsanslag basert på estimerte basiskurver og tidsavgrensede trafikktellinger. Denne modulen er gjort tilgjengelig i form av en Dynamic Link Library (DLL) for bruk på PC (Win32 plattform).

I tillegg er det utviklet en UNIX-basert tilleggsmodul som sørger for oppgradering av basiskurvene som inngår i hovedmodulen. Denne modulen er tilgjengelig i form av en C++-funksjon som er tenkt å skulle inngå i et større programsystem.

Hoved- og tilleggsmodulen spiller sammen ved at resultatfilene fra tilleggsmodulen er direkte lesbare for hovedmodulen. Brukeren må imidlertid selv påse at de oppdaterte basiskurvene blir overført til de regionale vegkontorene og tatt i bruk av det lokale Windows-programmet installert der.

Emneord/Keywords: Basiskurvemetoden, trafikktellinger

Tilgjengelighet/Availability: Åpen

Prosjektnr./Project no.: 220053

Satsningsfelt/Research field: Teknologi, industri og forvaltning

Antall sider/No. of pages: 13

Norsk Regnesentral / Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, 0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

Innhold

1 Innledning

Basiskurvemetoden er en metode for beregning av trafikkvolum på årsbasis. Basert på tellinger av antall kjøretøy per time fra kun en del av året beregnes antall biler per time for de resterende timene i året. Ut fra dette beregnes årsdøgntrafikk (ÅDT) med usikkerhetsanslag. Videre kan det beregnes gjennomsnittlig trafikkvolum med tilhørende usikkerhetsanslag for ulike delperioder av året, så som yrkesdøgntrafikk (YDT), helgedøgntrafikk (HDT), sommerdøgntrafikk (SDT) og julidøgntrafikk (JDT).

Trafikktellingene er gjort i et tellepunkt, og de ulike beregningene gjelder for dette punktet. For biler gjelder at de talte kjøretøyene kan være inndelt i enten 5 separate lengdeklasser, lette (lengdeklasse 1) og tunge (sum lengdeklasse 2-5) eller kun totalt antall kjøretøy (sum lengdeklasse 1-5). De 5 lengdeklassene er 1: 0-5.5m, 2: 5.6-7.6m, 3: 7.7-12.4m, 4: 12.5-15.9m og 5: 16.0m og lengre. Videre kan tellingene være på kjørefeltnivå, for hver enkelt kjøreretning eller for sum av to retninger. Beregninger av trafikkvolum gjøres for samme nivå som de foreliggende tellingene samt for eventuelle aggregerte lengdeklasse- og kjørefeltnivåer. I programversjonen pr. 1. april 2005 er det også mulig å utføre beregninger på sykkeltrafikk. I dette tilfellet opereres det med kun én lengdeklasse (totalt antall kjøretøy). For øvrig utnyttes den fleksibiliteten som ble bakt inn i det opprinnelige programmet, slik at kun mindre justeringer har vært nødvendig for at også denne typen data skal kunne håndteres.

Implementeringen av basiskurvemetoden utgjøres av to programsystemer. Hovedmodulen utviklet tidligere er et Windows-program som beregner trafikkparametere (ÅDT, YDT, …) med usikkerhetsanslag basert på estimerte basiskurver og tidsavgrensede trafikktellinger. I tillegg er det nå utviklet en UNIX-basert modul som sørger for oppgradering av basiskurvene som inngår i hovedmodulen.

Windows-programmet er installert lokalt hos de regionale vegkontorene og brukes til rutinemessige trafikkberegninger. Kjernen i denne programvaren er en C-funksjon som er nærmere beskrevet i kapittel 2.

Den nyutviklede UNIX-modulen for oppgradering av basiskurvene er en C++-funksjon som er tenkt å skulle inngå i et programsystem installert sentralt hos Vegdirektoratet. Oppgraderingen foretas på data for store byer i Norge og for småby/land hver for seg og bør skje hvert 3. - 5. år. Man bør da beregne basiskurver for et passende antall år fram i tid, for eksempel 15 år. Ved hver oppgradering bør det brukes nivå-1 telledata for 3 - 4 år og, hvis mulig, for omlag 100 veglenker eller mer (for hver distriktstype). Normalt vil nivå-1 dataene foreligge for i alt 5 lengdeklasser samt aggregerte nivåer (tunge og totalt antall kjøretøy). Grunnlagsdataene må dessuten være tilstrekkelig kvalitetskontrollert før kjøring av programmet. Brukeren må i etterkant selv påse at de oppdaterte basiskurvene blir overført til de regionale vegkontorene og tatt i bruk av det lokale Windows-programmet installert der. En detaljert beskrivelse av nødvendige inngangsdata til og resultatfiler fra programmodulen for oppgradering av basiskurvene er gitt i kapittel 3.

En kortfattet innføring i basiskurvemetoden er gitt i Aldrin og Haug (2000). Mer omfattende dokumentasjon finnes i Aldrin og Haug (1999), Aldrin og Haug (1998) og Aldrin (1998). Haug og Aldrin (2005) omtaler den nye funksjonaliteten knyttet til sykkeltrafikk.

2 Programvare for beregning av trafikkparametere

Norsk Regnesentral har implementert beregning av trafikkparametere med tilhørende usikkerhetsanslag i programmeringsspråket C. Beregningsmodulen *BeregnTrafikk* er gjort tilgjengelig i form av en Dynamic Link Library (DLL) for bruk på PC (Win32 plattform). I tillegg til selve DLLen krever beregningene installasjon av en del filer som benyttes av DLLen under programkjøring.

Under punkt 2.1 nedenfor er det gitt en beskrivelse av hvordan data skal overføres mellom beregningsmodulen og det kallende programmet. Punkt 2.2 angir hvilke eksterne filer som brukes av DLLen og forklarer hvor de skal installeres.

2.1 Overføring av data mellom beregningsmodul og kallende program

I forbindelse med utveksling av data til/fra beregningsmodulen *BeregnTrafikk* er det nødvendig å ha et omforent format/struktur på dataene som skal gå inn i modulen og resultatene som kommer ut.

Trafikktellingene er registrert på timenivå. For de timene i året hvor tellinger mangler, skal beregningsmodulen fylle inn estimerte verdier basert på den underliggende modellen og de talte dataene. I tillegg skal den beregne et minimumssett av trafikkparametre (ÅDT, YDT, HDT, SDT, JDT) med tilhørende usikkerhetsanslag etter bestemte retningslinjer.

I tillegg til timeindeksen har trafikktellingene følgende parameterisering:

* retning
* kjørefelt
* lengdeklasse

I Vegdatabanken er kun kjørefeltnummer registrert med den underliggende tolkning at odde kjørefeltnummer (1, 3, 5, …) representerer én og samme retning ('inn'), mens like kjørefeltnummer (2, 4, 6, …) representerer motsatt retning ('ut').

I NRs representasjon vil kjørefelt i én og samme retning bli nummerert fortløpende (1, 2, 3, …). Ved samtidig å knytte en retning (1, 2) til hvert kjørefelt ivaretas entydighet. Det kallende program må selv sørge for å konvertere mellom de to ulike representasjonsformene.

Vår beregningsmodul er bygget opp av flere C-funksjoner, men for brukeren vil modulen foreligge i form av en DLL (ett funksjonskall). I C-notasjon skjer overføringen av data til modulen gjennom (i rekkefølge) en 'struct'-konstruksjon, en feilmeldingsstreng av typen *wchar_t*, en heltallsverdi (*int*) som angir en øvre grense for lengden på feilmeldingsbufferet og en tekststreng av typen *wchar_t* som spesifiserer katalogen der de eksterne binærfilene som leses av programmet ligger. 'Struct'en inneholder følgende komponenter (*i* angir at komponentverdien må være satt før kallet til beregningsmodulen mens *o* betyr at komponentens innhold kan endres inne i beregningsmodulen):

geografi	*i*	Type region; dvs. om tellingene kommer fra by (=0), land (=1), sykkel (=2), ...
aar	*i*	Årstall for tellingene, på formen YYYY
nretn	*i*	Antall retninger som er talt (mulige verdier er 1 eller 2)

nlk	*i*	Antall lengdeklasser (mulige verdier er 1 (kun totalt antall kjøretøy, 2 (lette/tunge kjøretøy) eller 5 (full lengdeklasseoppløsning). For sykkeltrafikk er 1 eneste lovlige verdi.)
nextpar	*i*	Antall ekstra parametre som skal beregnes (utover ÅDT, YDT,…). Mulige verdier er 0, 1 , 2, … .
**nflt*	*i*	Antall kjørefelt i hver retning. Dette er en vektor med *nretn* elementer; hvert element angir antall kjørefelt i den tilhørende retningen (mulige verdier er 1, 2, 3, …). Antall kjørefelt i ulike retninger kan være forskjellig.
***extpar*	*i*	Indikatorvektorer for ekstra (brukerdefinerte) parametre som skal beregnes. Dette er ei matrise med 0/1-elementer som forteller hvilke timer som evt. skal inngå i beregningen av ikke-standard parametre (f.eks. kveldsdøgntrafikk). Matrisa har *nextpar* kolonner og like mange rader som det er timer i det aktuelle året. Hver kolonne svarer til én ekstra parameter.
****antall*	*i/o*	Tredimensjonalt array med talte/estimerte kjøretøy. Idet beregningsmodulen kalles, inneholder det kun talte verdier (med kode for manglende verdi innsatt for de timene hvor registreringer ikke er tilgjengelig). Ved retur er alle manglende verdier erstattet med estimerte timeverdier. Følgende unntak gjelder på kjørefelt-nivå hvis det finnes minst to kjørefelt i samme retning: Hvis det for et gitt tidspunkt foreligger tellinger for minst ett av kjørefeltene, samtidig som det mangler tellinger for minst ett av feltene (i samme retning), returneres beregnede verdier også der det foreligger tellinger.
****param*	*o*	Tredimensjonalt array med beregnede parametre (ÅDT, YDT osv.) med tilhørende standardavvik der dette inngår. Estimatene beregnes for alle mulige kombinasjoner av retning, kjørefelt og lengdeklasse på det nivået telledatene er gjort samt for alle aggregerte nivåer som kan avledes fra dette.

Når det gjelder de to siste komponentene i lista over, er disse ordnet på følgende måte.

***antall*

Aksene i det tredimensjonale arrayet med trafikktellinger representerer: retning/kjørefelt, lengdeklasse og tid. Dimensjonen på arrayet avhenger av verdiene på de tre størrelsene *nretn*, *nflt* og *nlk* samt antall timer i det aktuelle året. Innenfor hver akse er dataene ordnet på følgende måte:

i) Retningsparameteren brukes som første skille: Først kommer alle tellingene i retning 1, deretter alle tellingene i retning 2. Denne siste sekvensen faller bort dersom data kun foreligger i én retning eller kun samlet for begge retningene. Legg merke til at beregningsmodulen ikke skiller mellom data fra én retning på veien og sum over begge retningene (i begge tilfellene er *nretn* = 1). Innenfor hver retning organiseres dataene etter kjørefelt. Antall kjørefelt kan være forskjellig for tellinger i to retninger.

ii) Innenfor hvert kjørefelt deles dataene inn etter hvilken lengdeklasse de tilhører. Dette utgjør den andre aksen i arrayet. Antall lengdeklasser kan være 1 (kun totalt antall kjøretøy registrert), 2 (kjøretøy inndelt i lette og tunge) eller 5 (full oppløsning i kjøretøyklasser).

iii) Tidsaksen løper over alle timene i det året registreringen er foretatt.

***param*

****param* organiseres etter delvis samme prinsipp som ****antall*. Det betyr at hovedstrukturen med bruk av retning/kjørefelt og lengdeklasse som to av aksene i arrayet beholdes. I tillegg inkluderes eventuelle aggregeringsnivåer langs disse aksene. Den tredje aksen i ****antall* (tid) byttes ut med en parameterverdiakse i ****param*.

i) Organiseringen langs retning/kjørefeltaksen blir som følger: Først kommer alle resultatene på det nivået som datene er gitt, først for retning 1, deretter for retning 2 (igjen faller denne siste sekvensen bort dersom data kun foreligger i én retning eller kun samlet for begge retningene). Innenfor hver retning deles resultatene inn etter kjørefelt. Hvis dataene foreligger på kjørefeltnivå (*nflt*[i]>1), opprettes i tillegg et aggregert nivå for summen over alle kjørefelt i den aktuelle retningen. Hvis kjøretøy også er talt i begge retninger (*nretn*=2), får vi i tillegg et aggregeringsnivå for summen av kjøretøy i begge retninger summert over felt. Eventuelle aggregeringsnivåer plasseres fortløpende langs retning/kjørefeltaksen i forlengelsen av den opprinnelige strukturen.

ii) Innenfor hvert kjørefelt deles dataene inn etter hvilken lengdeklasse de tilhører. Dette utgjør den andre aksen i arrayet. I tillegg til lengdeklassene slik de ble gitt for telledataene, beregnes aggregerte nivåer der dette er mulig. Totalt antall elementer langs denne aksen er *nsize*. For *nlk*=1 blir *nsize*=1 (kun total), for *nlk*=2 blir *nsize*=3 (lette/tunge + total) og for *nlk*=5 blir *nsize*=7 (lengdeklasse 1-5, tunge (sum 2-5) og total (sum 1-5)).

iii) Til hvert element i det todimensjonale planet spent ut av retning/kjørefeltaksen og lengdeklasseaksen (talte eller aggregerte nivåer) hører et sett med estimerte verdier for ulike trafikkparametre. Disse danner resultataksen. Totalt antall elementer langs denne aksen er 2*(5+*nextpar*) verdier: ÅDT, SD(ÅDT), YDT, SD(YDT), HDT, SD(HDT), SDT, SD(SDT), JDT, SD(JDT) + evt. ekstra parametre som angitt av brukeren. SD(ÅDT) betegner standardavviket til ÅDT-estimatet osv.. På kjørefeltnivå beregnes ikke standardavvik. I stedet returneres kode for manglende verdi i disse elementene. Et unntak er hvis det er bare ett kjørefelt i en retning (*nflt*[i]=1); i dette tilfellet beregnes også standardavvik.

Nedenfor følger en oversikt over de nødvendige deklarasjonene slik de kan inngå i det kallende programmet (i C syntaks).

…

```
#define DMISS          -999.   // kode for manglende verdi
#define NFIXPAR         5      // antall parametre som alltid skal beregnes
#define MAXSTRLEN    2000      // øvre grense for lengde på tekststreng

struct TrafikkData {
  int geografi;
  int aar;
  int nretn;
  int nlk;
```

```
  int nextpar;
  int *nflt;
  int **extpar;
  double ***antall;
  double ***param;
}

…

/* Definer feilmeldingsbuffer */

int maksLengdePaaFeilmelding;
wchar_t *pFeilmeldingsBuffer;
maksLengdePaaFeilmelding = MAXSTRLEN;
pFeilmeldingsBuffer = (wchar_t *) malloc(maksLengdePaaFeilmelding*sizeof(wchar_t));



/* Definer tekststreng for datafiler */

wchar_t *pVBFilkatalog;
pVBFilkatalog = (wchar_t *) malloc(maksLengdePaaFeilmelding*sizeof(wchar_t));
wcscpy(pVBFilkatalog, (wchar_t *) "d:\\TrafikkBeregninger\\data_binary");


/* Sett av plass til TrafikkData-objekt */

struct TrafikkData *Tdata;
Tdata = (struct TrafikkData *) malloc(sizeof(struct TrafikkData));


/* Fyll inn nødvendige størrelser */

Tdata->geografi = <geografikode>;
Tdata->aar = <årstall for tellingene>;
Tdata->nretn = <antall retninger>;
Tdata->nlk = <antall lengdeklasser>;
Tdata->nextpar = <antall ekstra parametre>;


/* Sett av plass til og fyll vektor med antall kjørefelt for hver retning */

Tdata->nflt = (int *) malloc(Tdata->nretn*sizeof(int));
for (i=0;i<Tdata->nretn;i++)
  Tdata->nflt[i] = <antall felt i retning i>;


/* Fyll matrise med antall kjøretøy (talte eller manglende verdier) */

anttimer = 24*aardager[Tdata->aar];          // antall timer i året; aardager = 365(366)
                                               == antall elementer langs tidsaksen
nakse_retn=0;
for (i=0;i<Tdata->nretn;i++)
```

```
  nakse_retn += Tdata->nflt[i];                 // antall elementer langs retning/kjørefeltaksen

nakse_lk=Tdata->nlk;                            // antall elementer langs lengdeklasseaksen

Tdata->antall = (double ***) malloc(nakse_retn*sizeof(double **));
for (i=0; i<nakse_retn; i++) {
  Tdata->antall[i] = (double **) malloc(nakse_lk*sizeof(double *));
  for (j=0; j<nakse_lk; j++) {
    Tdata->antall[i][j] = (double *) malloc(anttimer*sizeof(double));
    for (k=0; k<anttimer; k++)
      Tdata->antall[i][j][k] = (data foreligger for aktuell time?) ? <antall kjøretøy> : DMISS;
  }
}


/* Fyll matrise med indeksvektorer for eventuelle ekstra parametre */

if (Tdata->nextpar > 0) {
  Tdata->extpar = (int **) malloc(anttimer*sizeof(int*));
  for (i=0;i<anttimer;i++) {
    Tdata->extpar[i] = (int *) malloc(Tdata->nextpar*sizeof(int));
    for (j=0;i<Tdata->nextpar;j++)
      Tdata->extpar[i][j] = (inngår  time i i ønsket parameter?) ? 1 : 0;
  }
}


/* Sett av plass til parameterne som skal beregnes */

nakse_retn=0;                                   // antall elementer langs retning/kjørefeltaksen
for (i=0;i<Tdata->nretn;i++)
  nakse_retn += Tdata->nflt[i];
if (Tdata->nflt[0]>1) nakse_retn += 1;
if (Tdata->nretn >1) {
  nakse_retn += 1;
  if (Tdata->nflt[1]>1) nakse_retn += 1;
}

switch(Tdata->nlk) {
  case 1:
    nsize = 1;
    break;
  case 2:
    nsize = 3;
    break;
  case 5:
    nsize = 7;
}
nakse_lk=nsize;                                 // antall elementer langs lengdeklasseaksen

nakse_res = 2*(NFIXPAR+Tdata->nextpar);   // antall elementer langs resultataksen

Tdata->param = (double ***) malloc(nakse_retn*sizeof(double **));
```

```
for (i=0; i<nakse_retn; i++) {
  Tdata->param[i] = (double **) malloc(nakse_lk*sizeof(double *));
  for (j=0; j<nakse_lk; j++)
    Tdata->param[i][j] = (double *) malloc(nakse_res*sizeof(double));
}
```

…

BeregnTrafikk(&Tdata, pFeilmeldingsBuffer, maksLengdePaaFeilmelding, pVBFilkatalog)

…

2.2 Eksterne datafiler

Beregningene som gjøres i *BeregnTrafikk* er basert på informasjon som er lagret i eksterne datafiler og som leses av DLLen under kjøring av programmet. Filene har binært format tilpasset bruk på PC (Win32 plattform). For at DLLen skal finne igjen filene, må de installeres i henhold til en fastlagt katalogstruktur som er relativ til katalogen spesifisert gjennom *pVBFilkatalog* i kallet til *BeregnTrafikk* (*pVBFilkatalog* skal ikke avsluttes med katalogskille ("\")). Strukturen er som følger:

Katalogen spesifisert gjennom *pVBFilkatalog* må inneholde fire underkataloger: en katalog for storbydata (kalt *by*), en for småby/landdistrikt (kalt *land*), en for sykkeldata (kalt *sykkel*) og en som inneholder filer som er felles for de tre områdetypene (kalt *felles*). Hver av katalogene *by* og *land* inneholder følgende filer:

- *basis1995_lk1.dat, basis1995_lk2.dat, …, basis1995_lk7.dat* (7 filer)
  Tilsvarende filer for årene 1996-2009 ligger også på samme sted (totalt 105 filer).
- *koefusikkerhet.dat*
- *optridge_lk1.dat, optridge_lk2.dat, …, optridge_lk7.dat* (7 filer)

Tilsvarende filer finnes på katalogen *sykkel*, men siden det her kun er snakk om totalt antall kjøretøy, eksisterer kun filene som inneholder lengdeklasseangivelsen '*lk7*'.

Katalogen *felles* inneholder filene

- *z1995.dat, z1996.dat, … , z2009.dat* (15 filer)

I programversjonen av 1. april 2005 har det (grunnet innføringen av sykkeldata) vært nødvendig å gjøre en endring i en skaleringsfaktor som berører '*optridge*'-filene. Dette betyr at også de eksterne datafilene som følger den oppgraderte programvaren må installeres sammen med denne!

Installasjon av eksterne datafiler og DLLen tar til sammen ca. 464MB diskplass. En full installasjon gir mulighet for trafikkberegninger for alle årene i perioden 1995 til 2009. Filene på *felles*-katalogen samt filene *koefusikkerhet.dat* og *optridge_lkx.dat*-filene må alltid installeres. Men for å gjøre beregninger for et bestemt år trenger man ikke legge inn annet enn de basiskurvefilene som svarer til dette året (f.eks. *basis1995_lk1.dat, …, basis1995_lk7.dat* for 1995). Dette kan være nyttig hvis programmet kun kjøres sporadisk og man ikke ønsker å legge beslag på unødig stor diskplass.

3 Programvare for oppgradering av basiskurver

Norsk Regnesentral har, som et ledd i basiskurvemetoden, utviklet programvare for oppgradering av basiskurver basert på nye nivå-1 tellinger. Beregningsmodulen er tilgjengelig som en C++-funksjon og skal etterhvert integreres i Vegdirektoratets egen programvare. I tillegg til selve beregningsmodulen leveres også et hovedprogram skrevet i C++. Dette programmet er utelukkende tenkt brukt til testformål, og det er derfor ikke lagt vekt på brukervennlighet i form av grafisk grensesnitt og liknende. All programvare er tilpasset kjøring under UNIX operativsystem.

Under punkt 3.1 nedenfor er det gitt en overordnet beskrivelse av funksjonaliteten i programvaren, mens punkt 3.2 omhandler detaljene i C++-funksjonen, så som grensesnitt, krav til maskinvare og eksterne filer samt resultatene som produseres. Punkt 3.3 inneholder en kort beskrivelse av hovedprogrammet.

3.1 Funksjonalitet - overordnet beskrivelse

Beregningsmodulen tilpasser et sett med åtte basiskurver til de nivå-1 tellingene som foreligger ut fra en nærmere angitt modell som blant annet tar hensyn til årstids- og døgnvariasjoner i trafikkmønsteret. Det er opp til brukeren å spesifisere hvilke nivå-1 tellinger som skal inngå i beregningene, og det er også brukeren som må påse at dataene er av tilfredsstillende kvalitet.

Programmet tillater tre ulike inndelinger av de talte kjøretøyene etter lengdeklasse: 1 lengdeklasse (kun totalt antall kjøretøy), 2 lengdeklasser (lette/tunge kjøretøy) eller 5 lengdeklasser (full lengdeklasse-oppløsning). Den første inndelingen kan gjelde både bil- og sykkeltrafikk, mens det i de to sistnevnte tilfellene kun er snakk om biler. Her forutsettes det dessuten at telledata foreligger også for alle mulige aggregerte nivåer; det vil si totalt antall kjøretøy i tilfellet lette/tunge, og både totalt antall kjøretøy og tunge kjøretøy i tilfellet med full lengdeklasseoppløsning. Basiskurvene beregnes for de samme lengdeklassene som det foreligger tellinger for, inkludert aggregerte nivåer. Normalt vil det være aktuelt å kjøre programmet kun for 5 lengdeklasser, samt aggregering til tunge og totalt antall kjøretøy. Typisk vil nivå-1 dataene omfatte 100 veglenker eller mer over en periode på 3-4 år.

Programvaren forutsetter at trafikktellingene er registrert på timenivå og genererer basiskurver med tilsvarende tidsoppløsning. Dette er i tråd med gjeldende praksis og i overensstemmelse med de krav som stilles til basiskurvene i tidligere levert programvare for beregning av trafikkparametere med tilhørende usikkerhetsanslag. Basiskurvene beregnes for hele år, og det brukerspesifiserte tidsvinduet for generering av basiskurver er ikke avhengig av tidsperioden som det foreligger nivå-1 tellinger for.

De resulterende filene med basiskurver blir skrevet på et format som er direkte lesbart for programmet som beregner trafikkparametere (*BeregnTrafikk*). Det er imidlertid opp til brukeren av programvaren å sørge for at filene blir overført til riktig filkatalog i forhold til bruk i dette programmet (underkatalogene *by*, *land*, *sykkel* eller *felles* relativt til katalogen spesifisert i kallet til *BeregnTrafikk*).

Beregningene som utføres av programvaren er omfattende og tidkrevende. Programmet legger beslag på betydelig internminne og prosessorkraft under kjøring og stiller dermed store krav til maskinvaren.

3.2 Beregningsmodul

Beregningsmodulen *estTrafficCurves* er skrevet som en funksjon i programmeringsspråket C++. I tillegg til en rekke nyutviklede rutiner benytter funksjonen seg også av rutiner fra C++-biblioteket VIIM utviklet ved Norsk Regnesentral. For å gi brukeren størst mulig grad av fleksibilitet, er de aktuelle biblioteksrutinene tatt ut og levert som kildekode.

All programutvikling er utført under Solaris operativsystem. Syntaksen er standard C++ for nyutviklede rutiner, og kildekoden skal derfor være mulig å overføre også til andre plattformer. På grunn av syntaksen i VIIM-rutinene må imidlertid all kompilering og lenking gjøres med GNU C++ kompilatoren g++ (fra utviklingsarbeidet vet vi at versjon 2.95.2 fungerer). I den totale leveransen inngår i tillegg til filer med ren funksjonskode også et sett med headerfiler.

3.2.1 Parametere i funksjonskall

Beregningsmodulen *estTrafficCurves* kalles med følgende parametere som argument (*i* angir at variabelverdien må være satt før kallet til beregningsmodulen mens *o* betyr at variabelens innhold kan endres inne i beregningsmodulen):

**tCounts* *i* C-'*struct*' av typen *trafficCountsLevel1*, nærmere beskrivelse nedenfor

**tcYear* *i* Heltallsvektor (*int*) med to elementer: Start- og sluttår for generering av basiskurver, hvert av de på formen YYYY. Typisk vil startår være inneværende år, mens sluttår for eksempel kan være 15 år fram i tid.

indDOS *i* Heltallsverdi (*int*) som bestemmer binærformatet til resultatfilene.
indDOS = 0: binærformat for bruk av programmer under UNIX operativsystem
indDOS = 1: binærformat for bruk av programmer under Windows operativsystem. Dette er det normale ettersom basiskurvene fortrinnsvis vil brukes av Windowsprogrammet som kaller *BeregnTrafikk*.

inclASCIIformat *i* Heltallsverdi (*int*) som styrer utskrift av resultatene til ASCII-formaterte filer.
inclASCIIformat = 0: ingen utskrift til ASCII-filer
inclASCIIformat = 1: ASCII-formaterte filer skrives

**fileDirOut* *i* Tekststreng (*char*) som angir filkatalogen hvor resultatfilene skal skrives. Avsluttes med katalogskille ("/").

**localMessage* *o* Tekststreng (*char*) som gir beskjed om hva som gikk galt ved eventuell unormal terminering. Strengen må ha en lengde på minimum 200 tegn. Feilmeldingstjenesten omfatter bare utvalgte situasjoner og utgir seg på ingen måte for å være komplett.

C-'*struct*'en *trafficCountsLevel1* inneholder følgende variable som må fylles før *estTrafficCurves* kalles:

eabDate[2] *i* Heltallsvektor (*int*) med to elementer: Start- og sluttime for nivå-1 tellingene, hver av de på formen YYYYMMDDHH

nlClass	*i*	Heltallsverdi (*int*): antall lengdeklasser i nivå-1 tellingene, lovlige verdier: 1, 2 eller 5
nRoadLinks	*i*	Heltallsverdi (*int*): antall veglenker som inngår i nivå-1 tellingene
**missing*	*i*	VIIM-objekt av typen *nMatrix*: Matrise med indikatorvariable (0/1; 0=manglende eller mangelfull telling, 1=godkjent telling) for hvert tidspunkt og hver veglenke med nivå-1 tellinger. Antall rader i matrisa = antall telletidspunkt, antall kolonner = antall veglenker. Matrisa har gyldighet for alle lengdeklasser.
***lClass*	*i*	Vektor av VIIM-objekter av typen *nMatrix*: Matrisene inneholder nivå-1 tellingene. Hver matrise representerer én lengdeklasse (evt. på aggregert nivå). Vektoren har lengde bestemt av *nlClass*, dvs. lovlige verdier er 1, 3 (2+1) eller 7 (5+2). Antall rader i hver av matrisene = antall telletidspunkt, antall kolonner = antall veglenker.

For matrisene som inneholder nivå-1 tellingene gjelder følgende: Alle timer mellom start- og sluttidspunktet (henholdsvis *eabDate[0]* og *eabDate[1]*) må være med. Likeledes må samtlige veglenker ha en representativ verdi for hver telletime. Manglende registreringer kan for eksempel angis med verdien -999, men selve verdien er uten betydning siden den tilsvarende posisjonen i *missing* **skal** være satt til verdien 0.

Variabelen **missing* er et VIIM-objekt av typen *nMatrix*, mens ***lClass* er en vektor av slike objekter. I det følgende beskrives to ulike måter å fylle slike objekter på med utgangspunkt i data som enten foreligger i form av en ASCII-fil organisert i rader og kolonner, eller som et todimensjonalt C-array. I begge tilfeller må header-fila *matrix.h* inkluderes i C++-koden.

Vi illustrerer bruken av *nMatrix* gjennom følgende eksempel: La *m* og *n* være henholdsvis antall rader og kolonner i datakilden (ASCII-fila eller C-arrayet) (for eksempel: i **missing* vil *m* angi antall timer med nivå-1 tellinger, mens *n* er antall veglenker). I C++ kodelinjene nedenfor fylles *nMatrix*-objektet *aMatrix* først med verdier fra ASCII-fila *ylk1.dat* og deretter med verdier fra C-arrayet *aArray*.

```
…
#include "matrix.h";
…

double aArray[m][n];
nMatrix aMatrix(m, n);
int i, j:

…

aMatrix.scan("ylk1.dat");                  // Fyller aMatrix med data fra fil

for (i=1; i<=m; i++)                       //
    for (j=1; j<=n; j++)                   // Fyller aMatrix med data fra aArray (elementvis)
        aMatrix.el(i,j) = aArray[i-1][j-1]; //

…
```

Legg spesielt merke til at *nMatrix*-objektet indekseres fra 1 og oppover.

3.2.2 Resultatfiler

Hovedresultatet fra beregningsmodulen er filer med basiskurver splittet etter årstall og lengdeklasse. Disse har benevnelser på formen *basis<YYYY>_lk<L>.dat* hvor *YYYY* angir årstallet og *L* indikerer lengdeklassen (*L*=6 angir tunge kjøretøy, *L*=7 totalt antall kjøretøy). I tillegg til basiskurvefilene lages det også filer hvor enhver time i et gitt år kategoriseres etter nærmere regler og hvor indikatorvariable for blant annet yrkesdøgntrafikk og sommerdøgntrafikk settes tilsvarende. Disse filene, som er nødvendige for programsystemet som estimerer ulike trafikkparametere og deres usikkerhet, genereres også på årsbasis og har benevnelsen *z<YYYY>.dat* hvor *YYYY* angir årstallet. Alle filer skrives til katalogen **fileDirOut* som spesifisert av bruker.

Avhengig av hvilke tellepunkter nivå-1 registreringene representerer, må de genererte basiskurvefilene flyttes til en underkatalog for enten land- (*land*), by- (*by*) eller sykkelkurver (*sykkel*). *z*-filene flyttes til underkatalogen *felles* (underkatalogene refererer seg her til filkatalogen som går inn i kallet av *BeregnTrafikk*).

Det lages alltid filer på binært format, mens ASCII-filer kun genereres dersom brukeren har gitt beskjed om dette gjennom å sette variabelen *inclASCIIformat* til 1. ASCII-variantene av basiskurve- og *z*-filene skrives også år for år og inneholder like mange linjer som det er timer i det aktuelle året. Basiskurvefilene inneholder åtte kolonner, en for hver basiskurve. *z*-filene inneholder fem kolonner: Kolonne 1 angir kategorikoden til timene (se Aldrin og Haug (1998)), mens kolonnene 2-5 inneholder indikatorverdier (0/1) for henholdsvis yrkesdøgntrafikk (YDT), helgedøgntrafikk (HDT), sommerdøgntrafikk (SDT) og julidøgntrafikk (JDT).

3.2.3 Diskplass og internminne

Resultatfilene stiller følgende krav til tilgjengelig diskplass. Dersom man kun velger binærformaterte filer, vil disse kreve omlag 455 kB per lengdeklasse og årstall. Ønsker man i tillegg ASCII-formatert utskrift, trengs ytterligere 820 kB per lengdeklasse og årstall. Kravet til internminne under programkjøring er tungt og avhenger av mengden med nivå-1 telledata (lengden av måleperioden, antall veglenker samt antall lengdeklasser). Hvorvidt det genereres basiskurver for ett eller flere år har imidlertid ingen betydning. Tabell 1 viser omtrentlig internminnebruk for utvalgte kombinasjoner av inngangsvariablene (alle beregningene er basert på tre år med nivå-1 telledata fra biltrafikk).

Oppsett	***nlClass***	***nRoadLinks***	**Internminne**
1	1	128	345 MB
2	5	128	499 MB
3	5	160	570 MB

Tabell 1 Bruk av internminne for utvalgte kjøreoppsett.

3.3 Hovedprogram

Til slutt skal vi kort omtale funksjonaliteten til hovedprogrammet *updateTrafficCurves* som kaller beregningsmodulen *estTrafficCurves*. Programmet er kun ment for utviklings- og testformål, men det kan likevel være nyttig å se et eksempel på hvordan beregningsmodulen kalles med ulike funksjonsparametere. Spesielt kan oppbyggingen av elementene i **tCounts* klargjøres gjennom dette eksemplet.

Testprogrammet er også skrevet i C++. Nødvendige programparametere leses fra fil som angis sammen med programkallet:

> *updateTrafficCurves <test.par>*

Parameterfila er en formatert ASCII-fil med følgende innhold:

```
        1995        1995
           5         160           1           1
/nr/minirisk/samba/gsa/Basis2D2data/nivaa1data/
/nr/project/stat/ohaug/Basiskurver/Basis2001/basiskurver/land/
```

I linje 1 spesifiseres start- og sluttår for genereringen av basiskurvene (*tcYear*). De to heltallene leses fra 12 tegn lange felt. Linje 2 inneholder følgende fire heltallsparametere (i rekkefølge): antall lengdeklasser (*nlClass*) og antall veglenker (*nRoadLinks*) i nivå-1 tellingene samt indikatorvariablene *indDOS* og *inclASCIIformat*. Også disse tallene leses fra 12 tegn lange felt. Linje 3 spesifiserer filkatalogen hvor filene med nivå-1 tellinger er å finne, mens linje 4 inneholder navnet på filkatalogen der resultatfilene skal skrives (*fileDirOut*).

Katalogen med nivå-1 telledata forutsettes å inneholde de nødvendige tellefilene i henhold til den verdien som er angitt for *nlClass* (husk aggregerte lengdeklasser!). Indikatorfila skal ha benevnelsen *miss.dat*, mens telledatafilene følger navnekonvensjonen *ylk<L>.dat* hvor *L* angir lengdeklassen.

De vedlagte filene *ylk7.dat* og *miss.dat* representerer henholdsvis nivå-1 telledata og tilhørende indikatorfil over mangelfulle/godkjente data. Formatet på disse filene er kompatibelt med det som leses av hovedprogrammet og gir således et eksempel på hvordan data til beregningsmodulen kan tenkes organisert.

Fila *trafficCurves.lnk* som også er vedlagt, gir et eksempel på hvordan kildekode- og headerfilene kan kompileres og lenkes til et kjørbart testprogram, *updateTrafficCurves*. *trafficCurves.lnk* skal være en eksekverbar fil som kjøres fra kommandolinja i UNIX.

Referanser

Haug, Ola og Aldrin, Magne (2005), "Beregning av trafikkvolum for sykler basert på basiskurvemetoden", Norsk Regnesentral, NR-notat SAMBA/08/05.

Aldrin, Magne og Haug, Ola (2000), "Beregning av trafikkvolum ved hjelp av basiskurvemetoden - En innføring", Norsk Regnesentral, NR-notat SAMBA/05/00.

Aldrin, Magne og Haug, Ola (1999), "Basiskurvemetoden - videreutvikling ved hjelp av ridge-regresjon", Norsk Regnesentral, NR-notat SAMBA/12/99.

Aldrin, Magne og Haug, Ola (1998), "Basiskurvemetoden for ÅDT-beregninger - Kalibrert for de største byområdene i Norge", Norsk Regnesentral, NR-notat SAMBA/25/98.

Aldrin, Magne (1998), "Traffic volume estimation from short-period traffic counts", Traffic Engineering + Control, **39**, 656-660.