Data-analyse met de PC

Arthur Pistorius

February 12, 2007

Voorwoord

Dit dictaat dient als naslagwerk bij de cursus "Data analyse met de PC (PPC)" ten behoeve van studenten in de Moleculaire Levenswetenschappen. Deze cursus beoogt vooral een praktische introductie te zijn in het toepassen van programma's zoals SPSS en MATLAB in een moleculair levenswetenschappelijke context. Het is niet opgezet als een cursus programmeren of ter vervanging van het vak "Wiskunde 3", waarin de lineaire algebra (vectoren en matrices, Fourier analyse) werden behandeld. Enerzijds leert men deze pakketten te gebruiken en anderzijds traint men in het vertalen van natuurwetenschappelijke problemen naar mathematische of statistische problemen en mogelijke oplossingen daarvan.

Nu zijn er vele softwarepakketten die vergelijkbare resultaten opleveren maar aan de keuze van bovengenoemde software pakketten liggen de volgende overwegingen ten grondslag.

1. Beschikbaarheid van licenties binnen campus of Faculteit

2. Universeel bruikbaar op diverse platforms (Unix, Windows PC, Mac) en voor alle denkbare probleemgebieden

3. Met name MATLAB is niet speciaal "gebruikersvriendelijk", dwz. gebruik is niet beperkt tot het plakken van meetgegevens en een analyse met een paar muisklikjes. Je moet zelf een oplossing bouwen voordat er gerekend kan worden. Dit vraagt om kennis van de onderliggende problematiek om tot een goed eindresultaat en een zinvolle interpretatie te komen.

De cursus is onderverdeeld in 2 aparte, op zichzelf staande secties, te weten:

1. Statistiek

2. Niet-lineaire regressie en curve fitting

Als intermezzo worden toch een aantal elementaire begrippen uit de lineaire algebra behandeld omdat deze in allerlei vakgebieden toegepast worden. Een elementaire kennis van vectoren en matrices is zeker gewenst wanneer men het programma MATLAB gaat gebruiken voor een vak als "data analyse".

De eindresultaten van deze cursus vinden hun weerslag in een aantal scripts, welke herbruikbaar zijn tijdens het verdere verloop van de studie en/of latere carriere.

Bij het ontwikkelen van de hier voorliggende cursus is steeds gezocht naar een subtiele balans tussen een "software showcase" (welke pakketten zijn zoal beschikbaar?), "een knoppencursus" (hoe kun je pakket X gebruiken om probleem Y op te lossen? zonder overigens de oplossing voorgekookt te presenteren) en verdieping van "moleculair levenswetenschappelijke vakkennis" door op een meer praktische wijze aan te sluiten op eerder behandelde vakken zoals "Statistiek" en "Programmeren in C++".

Nijmegen, Februari 2007
dr. A.M.A. Pistorius

Contents

Chapter 1

Vectoren en Matrices

1.1 Inleiding

Wiskunde III of Lineaire Algebra maakt geen deel meer uit van het curriculum Moleculaire Levenswetenschappen. De praktijk wijst echter uit dat enige kennis op dit terrein van nut kan zijn bij diverse andere vakgebieden, zoals fysica, biofysische chemie of bio-informatica.

In deze inleiding wordt kort ingegaan op eigenschappen en toepassingsmogelijkheden van vectoren en matrices. Deze inleiding wordt aangevuld met enkele practische oefeningen met MATLAB in het kader van de cursus Data analyse met de PC (SB027B).

1.2 Vectoren

De oorspronkelijke betekenis van het Latijnse woord vector is "drager" en hiermee is het gebruik van de term vector in de moleculaire biologie goed te begrijpen. In de wiskundige context is een vector gedefinieerd als een rij of een kolom van n complexe getallen.

$$a = (a_1, a_2, \ldots a_n) \qquad \text{rijvector}$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_n \end{pmatrix} \qquad \text{kolomvector}$$

hierin zijn a_i resp. b_i de componenten van de vector en n, de dimensie oftewel het aantal componenten van de vector. In de 3-dimensionale ruimte vormen de (reeele) componenten een lijnstuk met een lengte en een richting.

Elementaire vectoralgebra kan gebruikt worden om allerlei processen te beschrijven waarbij krachten, snelheden, electrische velden in het spel zijn. Te denken valt aan "gewone" mechanica maar ook aan moleculaire mechanica (MM/MD) welke de grondslag vormt voor berekeningen aan structuurmodellen op basis van NMR, kristallografie of infraroodspectroscopie (normaalcoordinaat analyse). In de wisselstroomtheorie worden vectordiagrammen gebruikt om faserelaties tussen spanningen en/of stromen inzichtelijk te maken.

Deze kunnen bv. in de electrofysiologie toegepast worden om electrische geleiding en impulsvoortplanting door zenuwcellen te beschrijven.

Vectoren hebben de volgende eigenschappen tav. optellen en vermenigvuldigen.

$$\begin{aligned} a+b &= (a_1+b_1, a_2+b_2, \ldots a_n+b_n) \\ c\cdot a &= (ca_1, ca_2, \ldots ca_n) \\ c(a+b) &= ca+cb \\ |a| &= \sqrt{a_1^2+a_2^2+\cdots a_n^2} \end{aligned}$$

Hierin is c een scalair, een getal, ofwel een vector met de dimensie n=1. on-line te bekijken).

Naast vermenigvuldiging met een constante zijn voor vectoren ook de volgende vermenigvuldigingen gedefinieerd. Dit zijn respectievelijk het in-product en het uit-product van 2 vectoren. Het in-product is gedefinieerd volgens:

$$\begin{aligned} a\cdot b &= |a||b|cos(\theta) \\ a\cdot b &= a_1\cdot b_1 + a_2\cdot b_2 \ldots a_n\cdot b_n = constante \end{aligned}$$

Hierin is θ de hoek tussen beide vectoren. Berekening van het in-product vindt zijn toepassing in de mechanica, wanneer je de arbeid (W) wilt berekenen welke wordt uitgeoefend door een kracht (F) langs een bepaalde weg, volgens $W = F \cdot s$. F en s zijn beide vectoriele grootheden en θ is de hoek tussen beide vectoren. $|a|$ wordt ook wel de norm van a genoemd. Een veel gebruikte, verkorte schrijfwijze voor het in-product is $< a, b >$. Het resultaat van de berekening van een in-product is een scalair (getal), dat gegeven wordt

door de lengte van de ene vector maal de projectie van de andere vector in dezelfde richting (tekenen).

Ten aanzien van deze vectorvermenigvuldigingen zijn de volgende eigenschappen van toepassing (zonder bewijs).

$$\begin{aligned} a \cdot b &= b \cdot a \\ a \cdot (b + c) &= a \cdot b + a \cdot c \\ c(a \cdot b) &= ca \cdot b + a \cdot cb \end{aligned}$$

Het uit-product of kruisproduct van 2 vectoren is als volgt gedefinieerd:

$$a \times b = |a||b|sin(\theta)c$$

Waarin θ weer de hoek tussen de vectoren representeert en c de eenheidsvector, welke loodrecht staat op zowel a als b . Het resultaat van het uitproduct is een vector, loodrecht op beide oorspronkelijke vectoren met een lengte die gelijk is aan het oppervlakte van het parallellogram, opgespannen door a en b

Op uit-producten zijn de volgende eigenschappen van toepassing (zonder bewijs).

$$\begin{aligned} a \times b &= -b \times a \\ a \times (b + c) &= a \times b + a \times c \\ c(a \times b) &= ca \times b = a \times cb \end{aligned}$$

Het uit-product is van toepassing op diverse fysische grootheden zoals het krachtmoment (fysica, moleculaire mechanica) of magnetisch moment (magnetische resonantiespectroscopie: ESR, NMR).

1.3 Matrices

Een matrix zou je kunnen beschouwen als een meer-dimensionale uitbreiding van vectoren of, in het algemeen, een verzameling van m bij n complexe getallen. Je zou een vector kunnen opvatten als een matrix waarbij de dimensie m of n gelijk is aan 1. De elementen in een matrix zijn gerangschikt

volgens:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & & & \\ . & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Een bijzondere matrix vormt de eenheidsmatrix (Identity matrix in het Engels, aangeduidt met I). Hierin zijn alle niet-diagonaalelementen gelijk aan 0. De elementen op de hoofddiagonaal zijn 1.

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ . & & & \\ . & & & \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Onder de voorwaarde dat de dimensies van matrices gelijk zijn, kun je ze elementsgewijs optellen. Het product van de matrices A en B wordt gedefinieerd als:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj} \tag{1.1}$$

In woorden: elk element in de resulterende matrix C wordt gevormd door de vermenigvuldiging van een rijvector uit A met een kolomvector uit B. Dit betekent dat de rijdimensie van A gelijk moet zijn aan de kolomdimensie van B. Je kunt de volgorde van de matrices A en B niet omdraaien, maar er is een uitzondering.

$$A \cdot I = I \cdot A \tag{1.2}$$

Voor een matrix A is ook een inverse matrix A^{-1} te berekenen, waarvoor geldt dat:

$$A \cdot A^{-1} = A^{-1} \cdot A = I \tag{1.3}$$

De berekening van een inverse matrix valt buiten het bestek van deze inleiding (daar gebruiken we MATLAB voor) maar de inverse matrix speelt oa. een rol bij het oplossen van stelsels van lineaire vergelijkingen. Een andere bewerking die in dit kader nog genoemd moet worden is de transpose matrix. Onder transponeren wordt het omwisselen van de rijen en kolommen van een matrix verstaan. Deze operatie is overigens ook van toepassing op

vectoren. Hierbij wordt simpelweg een rijvector in een kolomvector omgezet (of omgekeerd). Bewezen kan worden dat wanneer een matrix orthogonaal is, de inverse matrix gelijk is aan de getransponeerde matrix.

Voor een vierkante matrix (gelijke rij- en kolomdimensies) is een determinant gedefinieerd als:

$$det(A) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & & & \\ . & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix}$$

Voor een 3x3 matrix is deze nog handmatig uit te rekenen volgens:

$$det(A) = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \quad (1.4)$$

Determinanten spelen een rol bij het exact berekenen van een inverse matrix (niet behandeld) en gelden tevens als een criterium of een stelsel van lineaire vergelijkingen wel een oplossing heeft. Een dergelijk stelsel heeft een niet-lege oplossing wanneer de determinant van de matrix niet gelijk is aan 0. Berekening van een determinant is een alternatieve (maar gerelateerd aan de bovenstaande) methode om vast te stellen of een matrix orthogonaal is. In een dergelijk geval is de determinant +1 of -1.

Een speciaal geval vormt het eigenwaarde-probleem, dat als volgt is omschreven. Stel: een vierkante matrix A, gevraagd een kolomvector x die voldoet aan de vergelijking: $Ax = \lambda x$ met λ een scalair. Dus A toegepast op x levert een constante maal dezelfde x. Dit stelsel heeft een niet-triviale oplossing (een triviale oplossing wordt gevonden voor $x_1 = x_2 = x_n = 0$) wanneer:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} \dots a_{1n} \\ a_{n1} \dots & a_{nn-\lambda} \end{vmatrix} = 0$$

Deze determinant wordt de karakteristieke vergelijking, ofwel een n-de graads vergelijking in λ genoemd, met n reeele (of complexe) wortels $(\lambda_1, \lambda_2, \lambda_n)$: de eigenwaarden van A. Een bekend (of berucht) voorbeeld is de Schrödinger vergelijking

$$H(\Psi) = E \cdot \Psi \quad (1.5)$$

1.3.1 Lineaire vectortransformaties

De functie van een matrix is eigenlijk het geven van een voorschrift of een recept voor een vectortransformatie, zodat aan vector x een beeldvector y wordt toegevoegd, waarvoor geldt:

$$A \cdot x = y \tag{1.6}$$

A toegepast op x levert y. De notatie volgorde is hier van belang alsook het gegeven dat de kolomdimensie van A gelijk is aan de rijdimensie van x. Een dergelijke transformatie is lineair wanneer:

1. $A(\lambda x) = \lambda Ax$ voor elke x en scalair λ
2. $A(x + y) = Ax + Ay$

Wanneer x en y beide vectoren zijn in een assenstelsel, gevormd door eenheidsvectoren (dwz. vectoren die onderling loodrecht op elkaar staan en ieder een lengte 1 hebben) kunnen de coordinaten van y gevonden worden door de matrix A te transponeren (oftewel het omwisselen van rijen en kolommen). Een voorbeeld van een lineaire vectortransformatie is bv. de rotatie van een vector over een bepaalde hoek, beschreven door:

$$A = \begin{pmatrix} cos\phi & sin\phi \\ -sin\phi & cos\phi \end{pmatrix}$$

De beeldvector wordt gevonden via:

$$y = Ax = \begin{pmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{pmatrix} x$$

Zodat $y_1 = x_1 cos\phi - x_2 sin\phi$ en $y_2 = x_1 sin\phi + x_2 cos\phi$.

Tot en met de 3-dimensionale ruimte is dit nog inzichtelijk. Toch is het concept algemener van toepassing zoals bv. in de quantummechanica.

Chapter 2

Een hands-on inleiding MATLAB

2.1 Inleiding

MATLAB oftewel "Matrix Laboratory" is een krachtige, interactieve programma suite voor numerieke analyse en data-visualisatie. Naast het basisprogramma bestaan er diverse toolboxen, welke gebouwd zijn voor de oplossing van specifieke klassen van problemen zoals signaalverwerking, neurale netwerken, simulatie (SimuLink is in het eerste jaar al ter sprake geweest) en vele anderen. MATLAB kan ook gezien worden als een zelfstandige programmeertaal waarmee eenvoudige, herbruikbare programma's (in zg. M-files) kunnen worden ontwikkeld voor repeterende analytische problemen.

2.2 Een eerste verkenning

MATLAB is binnen de FNWI voor diverse platforms (Unix, wintel, Macintosh) beschikbaar. Om op een unix systeem de grafische mogelijkheden te kunnen benutten, kun je gebruik maken van het X-window systeem (X-terminal, werkstation of Windows PC met XWIN32 software). De huidige versie is MATLAB R2006b (oftewel 7.3). Binnen deze cursus wordt gebruik gemaakt van de PC versie. M-files, commando's en datafiles zijn echter uitwisselbaar tussen de diverse platforms.

Het programma start je meestal (vrij triviaal) via een snelkoppeling op het bureaublad van de PC. Mogelijk dat je de applicatie nog moet opzoeken

op de S:-schijf (software). Naast de menubalk bestaat de MATLAB desktop uit 3 vensters, nl.

1. De workspace. Dit geeft een overzicht van gebruikte variabelen, arrays en matrices

2. De command history. Hierin worden alle in gevoerde commando's gelogd. Kan handig zijn om eerder gegeven commando's te herhalen of te wijzigen

3. Het command window. Achter de prompt >> kunnen commando's ingevoerd worden

Deze indeling is naar eigen behoefte aan te passen via de menu-optie View.

In de menubalk is nog een veld voor de current directory. In deze, zelf te kiezen directory kunnen data- en M-files opgeslagen worden. Kies een dergelijke directory bij voorkeur niet in de boomstructuur van MATLAB: dataverlies bij het opnieuw booten van een net-PC zal het gevolg zijn. Foutmeldingen mbt. het niet kunnen vinden van bepaalde files zijn doorgaans terug te voeren op een verkeerde keus van de werkdirectory. Het kiezen van een root-directory, bv. `D:\`, kan ook aanleiding geven tot deze foutmelding. Maak dus gebruik van een sub-directory, bij voorkeur op je home-disk (bv. `H:\PPC`).

Commando's zijn vrij intuitief te geven. Voor een correct resultaat is het echter noodzakelijk dat je weet waar je mee bezig bent(!). Op dit punt onderscheidt MATLAB zich van andere "gebruiksvriendelijke" programmatuur met voorgebakken oplossingen waar je enkel nog je data hoeft in te voeren. De Help Navigator (vraagteken of menu-optie Help) biedt veel informatie over het gebruik van ingebouwde functies (algorithme, randvoorwaarden, toepassingsvoorbeelden). De on-line manuals zijn op inhoudsopgave, trefwoord/functie-index en vrije tekst te doorzoeken.

2.3 Invoer van data

2.3.1 Handmatige invoer

Zonder gegevens valt er natuurlijk niets te berekenen of te analyseren. MATLAB is een interactief programma dus data kun je handmatig invoeren. De mogelijkheid bestaat ook om datafiles (in ASCII of platte tekst) in een enkele

bewerking in te lezen. Voor grotere data-sets en meer-dimensionale problemen is dit de methode bij uitstek.

Op de volgende manier worden waarden aan variabelen toegekend.

```
a=1;
b=2;
c=a+b

c =

    3
```

De ; wordt gebruikt om de echo van de invoer te onderdrukken, waarbij in de derde regel van dit voorbeeld het resultaat van een rekenkundige operatie wordt toegekend aan variable c en tegelijkertijd door de niet-onderdrukte echo op het scherm wordt getoond.

Het basiselement waarop MATLAB berekeningen uitvoert is echter een array, waarvan de dimensie niet vooraf gedefinieerd hoeft te worden. Dit maakt gebruik van het programma een stuk eenvoudiger ten opzichte van andere programmeertalen zoals Fortran of C. Arrays worden als volgt gevuld (opdracht 1):

```
conc=[0.00 0.17 0.33 0.50 0.67 0.83];
ratio=[0.65 0.74 0.80 0.86 0.95 1.01];
```

In feite worden hier 2 rijvectoren met 6 elementen ingevoerd (zie de Size aanduiding 1x6 in het Workspace venster).

Voor veel berekeningen zijn echter kolomvectoren nodig. Deze kun je direct definieren volgens:

```
conc=[0.00;0.17;0.33;0.50;0.67;0.83];
ratio=[0.65;0.74;0.80;0.86;0.95;1.01];
```

Kolomvectoren kunnen echter ook verkregen worden uit rijvectoren dmv. transponeren: het verwisselen van rijen en kolommen. Dit werkt ook in de omgekeerde richting.

```
conc=conc';
ratio=ratio';
```

De size aanduiding in het Workspace venster verandert overeenkomstig de operatie in 6x1. Foutmeldingen van MATLAB zijn vaak terug te voeren op verkeerde dimensies van matrices of vectoren.

2.3.2 File in- en uitvoer

Er vanuit gaande dat je de juiste werkdirectory hebt geselecteerd kun je een ASCII data file importeren met het load() commando. Werkend in een andere directory kun je de volledige padnaam van het file opgeven, bv.:

```
load('cholaat.dat');
load('d:\ir_data\cholaat.dat');
```

MATLAB behandelt dergelijke gegevens niet als XY dataparen maar als een dubbele (6 bij 2) array met de naam cholaat. Dit wordt meteen duidelijk als je deze gegevens plot (later meer hierover):

```
plot(cholaat)
```

Het grafische venster, getiteld Figure No. 1, laat 2 (!) grafieken zien, uitgezet tegen het indexnummer van het element in de array. Om de juiste XY grafiek te maken moeten de kolommen uitgesplitst worden. Dit is mogelijk door een nieuwe variabele te definieren en een range uit de dubbele array op te geven.

```
x=cholaat(:,1);
y=cholaat(:,2);
plot(x,y)
```

Een rij of, in dit geval, een datapaar inlezen gaat op een vergelijkbare manier:

```
x=cholaat(1,:);
x=cholaat(2,:);
```

Voor het lezen en schrijven van platte ASCII databestanden zijn de volgende functies van belang. Naast filenaam en variabele kan nog een veldscheidingsteken opgegeven worden:

```
A=dlmread('meting.dat', ';');
dlmwrite('results.dat', A, ';');
```

MATLAB kent vele mogelijkheden om diverse bestandstypen (spreadsheets, beeldinformatie, geluidsbestanden) in te lezen of weg te schrijven. Geformatteerde files kunnen gelezen of geschreven worden met I/O functies die ook in de programmeertaal C worden gebruikt. In deze categorie valt ook de besturing en uitlezing van meetinstrumenten, welke via een RS232-poort

(COM1:, COM2: ...) met de PC verbonden zijn. Behandeling hiervan valt echter buiten het bestek van dit college.

De workspace, welke alle gedefinieerde arrays bevat, kan afzonderlijk worden opgeslagen in een file met een MAT-extensie. Het format is specifiek voor MATLAB en kan dus niet zonder meer door andere programma's ingelezen worden. Voor grotere, langlopende projecten kan dit een handige optie zijn (voorkomt veel overbodig, herhaald type-werk). De workspace kun je op elk moment opslaan dmv. het diskette icoontje in het workspace venster of via het file-menu en vervolgens de optie "Save workspace as". In beide gevallen wordt je geprompt om een filenaam op te geven. Om de workspace of een afzonderlijke array (conc) schoon te vegen of te verwijderen, gebruik je het commando:

```
clear all;
clear conc;
```

Een eerder opgeslagen workspace kun je weer inlezen met het load('opgave1.mat') commando, zoals eerder is beschreven.

2.4 Vectoren, matrices en lineaire algebra

2.4.1 Algemene eigenschappen

In de gebruikspraktijk van MATLAB worden termen als array, vector en matrix vaak losjes door elkaar gebruikt. Een matrix is in ieder geval gedefinieerd door een 2-dimensionale rangschikking (array) van reeele of complexe getallen, welke een voorschrift voor een lineaire transformatie geeft. De verschillen komen binnen MATLAB tot uiting in de aanduiding in de Workspace. Een scalair heeft een dimensie van 1 rij bij 1 kolom (1x1), een rijvector heeft de dimensie van 1 bij n (bv. 1x6), een kolomvector n bij 1 (bv. 6x1) en een matrix n bij m (bv. 6x6). Het is gebruikelijk om vectoren aan te duiden met kleine letters en matrices met hoofdletters. MATLAB heeft (natuurlijk) vele functies welke toepasbaar zijn op matrices of welke matrices produceren. Dit college beoogt geen vervanging te zijn voor een cursus "Lineaire Algebra" dus een aantal matrix- en vector-operaties binnen MATLAB welke van belang zijn voor de oplossing van talrijke analytische problemen worden zonder bewijs gepresenteerd.

Optelling en subtractie van vectoren en matrices gaat gewoon, volgens de regels van de kunst, element voor element mits de dimensies maar overeenstemmen. Een bijzondere optelling welke niet bestaat in de lineaire algebra is het verhogen van alle matrixelementen met een constante, volgens:

```
X =
    9  2  7
    4  7 10
    5 12  8

a=3;
X + a

anw =

   12  5 10
    7 10 13
    8 15 11
```

Rij- en kolomvectoren zijn 1-dimensionele arrays en kunnen in elkaar omgezet worden dmv. transponeren: het omwisselen van rijen en kolommen. Dit is in de vorige paragraaf al naar voren gekomen. Een inproduct van 2 vectoren is vermenigvuldigen van een rij- met een kolomvector: dit levert een scalair (getal) op. Het uitproduct is een vermenigvuldiging van een kolom- met een rijvector.

```
u = [3 1 4]';
v = [2 0 -1];
x = v*u

x =

    2

X = u*v

X =

   6  0  -3
```

```
   2   0   -1
   8   0   -4
```

Een belangrijk onderscheid moet gemaakt worden tussen een arrayproduct en een matrixproduct. Zo kun je 2 kolomvectoren niet met elkaar vermenigvuldigen volgens:

```
u= [3 1 4];
v=[2 0 -1];
u*v
```

Een foutmelding ("Inner matrix dimensions must agree") is het resultaat. Het arrayproduct is echter wel gedefinieerd als de vermenigvuldiging van element voor element. Een arrayproduct wordt aangegeven met .* zoals in:

```
u= [3 1 4];
v=[2 0 -1];
u .* v

ans =

     6
     0
    -4
```

Het matrixproduct $C = AB$ is alleen gedefinieerd wanneer de kolomdimensie van A even groot is als de rijdimensie van B, oftewel: wanneer A de dimensie m bij p heeft, en B de dimensie p bij n, dan heeft C de dimensie m bij n. Het product wordt als volgt berekend:

```
for i=1:m
   for j=1:n
      C(i,j)=A(i,:)*B(:,j);
   end
end
```

Een matrixvermenigvuldiging in MATLAB wordt gewoon uitgevoerd met een product operatie:

```
C = A*B
```

De volgorde van A en B kan niet omgewisseld worden (niet-commutatief). Een matrix kan ook nog links vermenigvuldigd worden met een rijvector en rechts met een kolomvector. Het resultaat is een rijvector resp. een kolomvector. Op deze manier kunnen simultane stelsels van lineaire vergelijkingen compact worden weergegeven.

2.4.2 Bijzondere matrices en operaties

De eenheidsmatrix is een bijzondere, vierkante (dimensie nxn) matrix, welke gevuld is met nullen op de niet-diagonaal elementen en enen op de diagonaal elementen, bv.:

```
I =
    1 0 0 0
    0 1 0 0
    0 0 1 0
    0 0 0 1
```

Zodat alleen voor deze matrix geldt dat $A * I = I * A$. Een dergelijke matrix creeer je snel met:

```
eye(4,4)
eye(4)
```

Voor een eenheidsvector staat echter de ones() functie ter beschikking. Het aantal elementen kun je laten afhangen van de grootte van een andere array zoals bv. in:

```
t = [1 2 3 4];
v = ones(size(t))
v =

   1  1  1  1
```

Een array of matrix vullen met willekeurige getallen kan ook wel eens van pas komen. Dit gaat analoog aan de eerder beschreven voorbeelden met:

```
c=rand(1,4)
C=rand(4,4)
```

Voor een vierkante, niet-singuliere matrix A is ook een inverse matrix A^{-1} gedefinieerd, welke ervoor zorgt dat $A^{-1} * A = I$ en $A * A^{-1} = I$ waarbij I de eenheidsmatrix is. De inverse matrix wordt berekend met:

```
X = inv(A)
```

De inverse matrix heb je nodig bij het oplossen van stelsels van lineaire vergelijkingen. Berekening van een inverse matrix is "rekentechnisch" een kostbare aangelegenheid (tijd). Voor deze toepassing staat binnen MATLAB de backslash operator (\) ter beschikking.

2.4.3 Het oplossen van lineaire vergelijkingen

De oplossing van het eerder aangehaalde voorbeeld van de kwantitatieve analyse van een enkele component (detergentia in proteoliposomen: zie ook de opgaven) komt neer op de bepaling van de vergelijking van een ijklijn welke een concentratie voorspelt, gegeven een gemeten optische dichtheid (Lambert-Beer). Dit kan uitgebreid worden naar meerdere componenten, bv. het eiwit-, suiker- en vetgehalte in melk. In dit geval wordt gesproken over multiple lineaire regressie (MLR). De wiskundige oplossing verloopt via dezelfde route.

In eerste instantie zoeken we een matrix die, toegepast op een serie bekende concentraties, een serie absorptiewaarden geeft. Bij de bepaling van een onbekend sample wordt de omgekeerde weg gevolgd. Intuitief moet ergens een inversie operatie worden uitgevoerd. De berekening in MATLAB wordt echter uitgevoerd met de backslash operator \ . De delingsoperator slash (/) of backslash (\) geeft dus aan aan welke kant de onbekende matrix (X) staat tov. de coefficientenmatrix (A). Gegeven twee matrices A en B. Dan is:

- X = A\B de oplossing van de matrixvergelijking AX = B
- X = B/A de oplossing van de matrixvergelijking XA = B

Om X = A\B te berekenen moeten A en B hetzelfde aantal rijen hebben. De oplossing X heeft dan hetzelfde aantal kolommen als B en hetzelfde aantal rijen als A. Bij de "linkerdeling" X = A/B moeten de kolom- en rijdimensies omgewisseld zijn.

De coefficientenmatrix A hoeft niet vierkant te zijn. MATLAB zoekt aan de hand van de samenstelling van A een geschikt algorithme. In het voorbeeld van de regressie analyse heb je meestal te maken met een overgedetermineerd

systeem (meer dan 2 punten worden gebruikt om de vergelijking van een rechte ijklijn te bepalen). In zo'n situatie wordt een kleinste kwadraten oplossing berekend.

2.5 Grafieken

In de index van "MATLAB, The Language of Technical Computing" komt de term "plot" niet voor, maar toch beschikt MATLAB over goede faciliteiten om grafieken van wiskundige functies of van ingelezen en bewerkte meetresultaten te genereren (zoek plot eens op in de MATLAB help index). De eenvoudigste vormen van een lijn- resp. een puntengrafiek (plusteken als markering voor een meetpunt) worden verkregen via:

```
plot(x,y)
plot(x,y, '+')
```

Een tweede plotcommando veegt echter de eerste grafiek weg. Meervoudige plots kunnen gemaakt worden middels de "hold on" functie. "hold off" schakelt deze eigenschap weer uit.

```
plot(x,y)
hold on
plot(x,y, '+')
hold off
```

Een alternatief om meerdere curves tegen een vaste horizontale as te plotten is juist door gebruik te maken van de eigenschap dat de kolommen van een meerdimensionale array als afzonderlijke curves worden weergegeven. Dit is in de vorige paragraaf al aangestipt. Het volgende voorbeeld is ontleend aan het dictaat "Matlab for Chemists" (P.E.S. Wormer april 2003). Let op de enkele quote aan het eind van de eerste regel!:

```
phi=[0:pi/100:2*pi]';
y(:,1)=cos(phi);
y(:,2)=sin(phi);
plot(phi,y)
title('Graph of the sine and cosine functions')
```

Vaak komt het voor dat je meerdere subplots wilt weergeven in een figuur. Een bladzijde uit een stripboek met 6 plaatjes in 3 rijen en 2 kolommen definieer je als volgt:

```
figure;
subplot(3,2,1);
plot(x,y)
```

In dit voorbeeld wordt plaatje 1, linksboven, gevuld met de grafiek van y versus x.

Met de functies

```
xlabel('Equivalenten cholaat/lipiden')
ylabel('A1397/A1233 ratio')
title('IJklijn cholaat in fosfolipiden')
```

kan een grafiek geannoteerd worden. Dit kan ook in het venster Figure No. 1 zelf. Kies het Edit menu en vervolgens Axes Properties. Per as kunnen titel, bereik en andere eigenschappen van de assen ingesteld worden. Onder het menu File en Page setup kunnen de definitieve afmetingen van een hardcopy plot ingesteld worden.

2.6 M-files

De M-files maken in feite de universele toepasbaarheid van MATLAB mogelijk. Er zijn 2 categorieen te onderscheiden, tw.:

1. scripts of verzamelingen van MATLAB opdrachten voor het uitvoeren van repeterende bewerkingen op data in de workspace

2. functie m-files die input (arrays ...) vragen, hier een bewerking op uitvoeren en tenslotte de resultaten terugplaatsen in de workspace.

M-files maak je met een ASCII tekst editor (notepad, edit, vi, emacs .... maar vooral niet met MS Word) en hebben een .m extensie. Aan de hand van een simpel voorbeeld wordt het gebruik van een functie-m-file snel duidelijk.

Het bestand SPEED.M converteert snelheid in km/h naar m/s. Dit bestand ziet er als volgt uit:

```
function v=speed(s)
% SPEED rekent snelheid in km/h om in m/s
v=1000*s/3600;
```

De eerste regel definieert de functie (script M-files hebben deze regel niet), de input argumenten en de retourwaarde of -variabele. Het is gebruikelijk dat de m-filenaam en de functienaam gelijk gekozen worden. De tweede regel, beginnend met % is een commentaarregel (meerdere regels kan ook) en geeft de on-line help informatie weer. Maak er een goede gewoonte van om functie- en script files (bij welk softwarepakket dan ook) goed te documenteren. Deze voorzienig is ook van toepassing op de script m-files en concreet betekent dit:

```
help speed

SPEED rekent snelheid in km/h om in m/s
```

Speed kun je nu in een gewone MATLAB opdracht gebruiken, dus:

```
y=speed(50)

y=

   13.8889
```

MATLAB kent een aparte set functie-functies, die toegepast kunnen worden op functie m-files, zoals plotting (fplot()), numerieke integratie (quad()), het zoeken van minima (fmin()) en het zoeken van nulpunten (fzero(). Raadpleeg de help informatie). Een speciale set van functie-functies voor het oplossen van gewone differentiaalvergelijkingen vallen buiten het bestek van deze cursus. Voor de omrekening van 50 m/s naar km/h kunnen we speed en fzero als volgt gebruiken:

```
z=fzero('speed(x)-50',10)

z=

   180.0000
```

fzero() gebruikt dus de functienaam tussen enkel quotes en een beginwaarde als extra argument. In plaats van de functienaam kun je ook een in-line functie ter plaatse definieren, zoals het volgende voorbeeld laat zien.

```
z=fzero('1000*x/3600-50',10)
```

```
z=

    180.0000
```

Bibliography

[1] R. Wehrens (2002) *Dictaat Statistics for Chemists*

[2] P.E.S. Wormer (2003) *Dictaat Matlab for Chemists*

[3] P.E.S. Wormer (red.) (2001) *Dictaat Computers in de Chemie I* Ihb. hoofdstukken 4, 5 en 6.

[4] Beckers, M.L.M. (1997) *Chemisch Magazine* mei, 171–181.

[5] Bowen, W.P. en Jerman, J.C. (1995) *Trends in Pharmacological Sciences* (Vol. 16) december, 413–417.

[6] Lauwerier, H. (1994) *Spelen met graphics en Fractals: 80 programma's in Q(uick)BASIC en PowerBASIC* Academic Service, Schoonhoven ISBN 90 395 0092 4.

Appendix A

Niet-lineaire regressie met MS Excel

Bij het overweldigende aanbod van computerprogramma's voor het oplossen van curve fitting problemen op allerlei terreinen (Gnuplot/Gnufit, Matlab, GraphPad, Peakfit, MathCad ....) zou je bijna over het hoofd zien dat een spreadsheetprogramma zoals Microsoft Excel ook ingebouwde functies heeft voor het oplossen van regressie-vraagstukken. Niet iedereen heeft de beschikking over kostbare pakketten zoal MATLAB , maar Excel is beschikbaar op praktisch iedere PC. Afhankelijk van de geinstalleerde versie zal misschien de zg. Solver Add-in nog geinstalleerd moeten worden.

A.1 Configuratie van Excel voor curve fitting

Aan de hand van NONLINREG.XLS wordt een uitgewerkt voorbeeld gegeven. Dit voorbeeld is ontleend aan het artikel "Nonlinear regression using spreadsheets" (W.P. Bowen en J.C. Jerman in Trends in Pharmacological Sciences (16) 413–417, 1995) [5]. Dit voorbeeld behandelt de analyse van competitieve bindingsdata, gebruikmakend van een (standaard) logistieke vergelijking met 4 parameters:

$$y = min + (max - min)/(1 + (10^{logIC_{50}}/10^{log[I]})^{slope}) \qquad \text{(A.1)}$$

In deze vergelijking zijn de volgende parameters van belang:

- max, maximale specifieke binding

- min, minimale specifieke binding
- $logIC_{50}$, de logarithme van de inhibitorconcentratie die 50 % van de specifieke binding tot gevolg heeft
- slope, de richtingscoefficient in het buigpunt op het stijle stuk van de curve.

Binnen de Blackboard omgeving kun je Excel gebruiken door NONLINREG.XLS te openen in een "New window". Dit is mogelijk via de rechter muisknop, waarbij de cursor op de hyperlink staat. Een andere optie is het opslaan van dit bestand op een lokale harde schijf en Excel apart op te starten. Controleer of de Solver geinstalleerd is via: Tools >> Add-ins ... Vink zonodig de Solver Add-in aan en druk op OK. Na installatie is de Solver als menu item zichtbaar in het Tools menu.

In kolom B is de specifieke binding (als percentage) van een radio-actief ligand uitgezet tegen de logarithme van de inhibitor-concentratie (kolom A, log(mol/l)). Vul de cellen in kolom C met de vergelijking van het gekozen model. In Excel typ je dan:

```
=min+((max-min)/(1+((10^logIC50/10^logI))^slope))
```

Merk op dat de cel wordt gevuld met #NAME?. Excel werkt normaal met cel-referenties en niet met variabelen. Namen van variabelen worden later gedefinieerd.

Zoals bij MATLAB al ter sprake is geweest, willen we de parameters bepalen door de som van de gekwadrateerde verschillen tussen gemeten data en het model (kolom B minus kolom C) te minimaliseren. Kolom D wordt gevuld met de vergelijking

```
=(B2-C2)^2
```

(celnummers in de kolom aanpassen). De som van de kwadraten is nu:

```
=SUM(D2:D11)
```

Kolom F gaat de eerste parameterschattingen bevatten. Ook hier geldt: hoe beter de schatting, hoe sneller het fitproces tot een goed einde komt. Definieer de namen van gebruikte variabelen en fitparameters. Cel F3=max, F4=min, F5=logIC50, F6=slope, D13=SS. Namen definieer je via Insert >> Name >> Define. Vul een naam en een celreferentie in, en druk op Add.

Markeer met de muis de log[I] waarden in kolom A en definieer op de zelfde manier als hiervoor logI als de naam voor de array met log[I] waarden. Nu springen de getalletjes in kolom C en D in het gelid.

Start in het Tools menu de Solver. Stel de SS cel (D13) in als de te minimalizeren doel-cel. Markeer de cellen F2, F3, F4 en F5 als de te varieren cellen/parameters. Klik op Solve en merk op hoe de fitparameters, de voorspelde waarden, de gekwadrateerde verschillen en de kwadratensom veranderen.

A.2 Plotjes in Excel

Een grafische weergave van de data helpt bij het opstellen van een model en de beginschatting van de fit-parameters. Met al zijn voorgekookte oplossingen is Excel geen geweldig plotprogramma (daar is het niet voor gemaakt) maar desalniettemin ...

Een grafiek maak je via Insert >> Chart. Kies als type een scatterplot met gemarkeerde datapunten, verbonden door lijntjes. Trap er niet in om intuitief een lijngrafiek te kiezen: in dat geval kun je de X-as niet meer aanpassen. Druk op Next en selecteer met de muis de kolommen A, B en C als data-bereik. De eerste kolom wordt nu de X-as. In de volgende stappen kun je allerlei as-eigenschappen, titels en legenda aanpassen. Uiteindelijk kun je de plot mbv. de muis (klikken en slepen) naar wens op het werkblad plaatsen. Door met de rechter muisknop op onderdelen van de plot te klikken kun je eigenschappen in detail aanpassen (Format optie). Probeer altijd de printertoner-vretende grijze achtergrond te verwijderen en zoek maar eens uit hoe je in dit geval, met negatieve waardes op de X-as, de Y-as links kan plaatsen.