Mathematical Modelling in the Natural Sciences
SS16, Exercises, Sheet 8

Solutions to be presented on 13. May 2016

1. Let `uexact` be the (exact) output from the program `convdiff.m`

   `http://imsc.uni-graz.at/keeling/modII_ss16/convdiff.m`

   with (exact) input parameters $a_1 = 10$, $k_1 = 2$, $k_2 = 1$ and $k_3 = 0.1$. Develop a computational tool, calling the program convdiff.m, to determine these (exact) parameters as if they were unknown. (Solved by)

2. The following data (2176 and 109051904 bytes respectively)

   `http://imsc.uni-graz.at/keeling/modII_ss16/aif.txt`
   `http://imsc.uni-graz.at/keeling/modII_ss16/ct.txt`

   can be read into matlab as follows:

```
fa   = fopen('aif.txt','r');
AIF  = reshape(fscanf(fa,'%g',inf), 2,[])';
t    = AIF(:,1);
t    = t-t(1);
cAIF = AIF(:,2);
Nt   = length(t);
clear AIF;
fa = fclose(fa);

fc = fopen('ct.txt','r');
Nx = 256;
Ny = 256;
cT = reshape(fscanf(fc,'%g',inf),Nt,Nx,Ny);
cT = flipdim(permute(cT,[2,3,1]),1);
fclose(fc);
```

   and then visualized as follows:

```
subplot(1,2,1)
plot(t,cAIF)
subplot(1,2,2)
for k=1:Nt
    imagesc(reshape(cT(:,:,k),Nx,Ny)); axis image; axis off; colormap('gray');
    drawnow;
end
```

To establish relationships between these arrays and the constructions in the lectures notes, let `cAIF(k)` be $C_{\text{AIF}}(t_k)$ for t_k = `t(k)`, $k = 1, \ldots,$ `Nt`, and let `cT(i,j,k)` be $C_{\text{T}}(x_i, y_j, t_k)$ for $x_i = ih$, $i = 1, \ldots,$ `Nx`, $y_j = jh$, $= 1, \ldots,$ `Ny`, $h > 0$ and t_k = `t(k)`, $k = 1, \ldots,$ `Nt`.

For each pixel (x_i, y_j), solve the deconvolution problem of determining the kernel function K from known C_{T} and C_{AIF} in the convolution $C_{\text{T}} = K * C_{\text{AIF}}$, and use truncated singular value decomposition. Specifically, let the matrix A be constructed from C_{AIF} as given in the lecture notes. Let $\boldsymbol{C}_{\text{T}}$ be a vector of tissue concentrations at advancing times and for a given pixel. Let $\boldsymbol{K}$ be a vector of unknown kernel values at advancing times and for a given pixel. The problem to minimize $\|A\boldsymbol{K} - \boldsymbol{C}_{\text{T}}\|^2$ with respect to $\boldsymbol{K}$ should be solved by performing truncated singular value decomposition to obtain a regularized pseudo-inverse of A.

For a fixed pixel (x_i, y_j) let $\{K(x_i, y_j, t_k)\}_{k=1}^{\texttt{Nt}}$ be the estimated kernel function. Plot $V(\cdot,\cdot) = \sum_{k=1}^{\texttt{Nt}} K(\cdot,\cdot,t_k)$ as an image,

```
imagesc(V); axis image; axis off; colormap('gray');
```

and interpret your result. (Solved by – see also the solution in

http://imsc.uni-graz.at/keeling/modII_ss16/deconv.m

and the visualization in

http://imsc.uni-graz.at/keeling/modII_ss16/browse.m.)

3. Now consider the method of deconvolution using an exponential basis

$$K(t;\boldsymbol{k}) = \sum_{k=1}^{M} k_m \exp[-\lambda_m t]$$

where the estimated kernel $K(t;\boldsymbol{k})$ is constrained to be non-increasing. Assume that data are given as in the previous exercise. Let the time scales be harmonically distributed:

```
M = 20;
lambda = (1:M)/t(end);
```

Construct a matrix E as follows, where $E\boldsymbol{k} \leq \boldsymbol{0}$ insures that K is non-increasing,

```
E = -diag(lambda);
for m=1:(M-1)
    Dm = eye(M);
    for j=1:(M-m)
        i = m+j;
        Dm(j,j) = 1/(lambda(i)-lambda(j));
        Dm(j,j+1) = -Dm(j,j);
    end
    E = Dm' \ E;
end
```

Add the following row to E to insure that $-K(0) \leq 0$.

```
E = [E;-ones(1,M)];
```

The monotonicity constraint is $E\boldsymbol{k} \leq \boldsymbol{z}$ where $\boldsymbol{z} = \boldsymbol{0}$,

```
z = zeros(M+1,1);
```

The constrained optimization problem is to minimize $\|\boldsymbol{C}_{\mathrm{T}} - AK(\boldsymbol{t};\boldsymbol{k})\| = \|\boldsymbol{C}_{\mathrm{T}} - B\boldsymbol{k}\|$ under the constraint $E\boldsymbol{k} \leq \boldsymbol{z}$, where

```
B = A*exp(-kron(t,lambda));
```

and A depends upon C_{AIF} as in the previous exercise and as given in the lecture notes. Once the solution $\boldsymbol{k}$ is determined for a given pixel (x_i, y_j),

```
opts = optimset('LargeScale','off','Display','off','MaxIter',100000);
max = 1.0e5; k = zeros(M,1); cTij = reshape(cT(i,j,:),Nt,1);
k = lsqlin(B,cTij,E,z,[],[],-mx,mx,k,opts);
```

the kernel function is given by

```
K = exp(-kron(t,lambda))*k;
```

and the physiological parameters are given by

```
VolFrac(i,j)  = dot(k,1./lambda);
FlowRate(i,j) = dot(k,ones(1,M));
MeanTime(i,j) = VolFrac(i,j)/FlowRate(i,j);
```

Use these hints to repeat the previous exercise to perform the deconvolution using the exponential basis approach instead of truncated singular value decomposition. (Solved by – see also the solution in

`http://imsc.uni-graz.at/keeling/modII_ss16/deconv_exp.m`

and the visualization in

`http://imsc.uni-graz.at/keeling/modII_ss16/browse_exp.m`.)