Solutions - Problem Sheet 12

1. a) Let us denote $w = \exp\left(\frac{2\pi\imath}{n}\right)$. Then the following holds

$$\begin{aligned} w^m &= \exp\left(\frac{2\pi\imath}{n}m\right) \text{ for } m \in \mathbb{Z}, \\ w^n &= \exp\left(\frac{2\pi\imath}{n}n\right) = \exp(2\pi\imath) = 1, \\ \overline{w^m} &= \overline{\exp\left(\frac{2\pi\imath}{n}m\right)} = \exp\left(\overline{\frac{2\pi\imath}{n}m}\right) = \exp\left(-\frac{2\pi\imath}{n}m\right) = w^{-m}. \end{aligned}$$

Now, since $f^n_{kl} = \exp\left(\frac{2\pi\imath}{n}(k-1)(l-1)\right)$ we have

$$f^n_{kl} = f^n_{lk} = w^{(k-1)(l-1)} \text{ for all } 1 \le l, k \le n.$$

In other words $F_n^\top = F_n$, and therefore $F_n^H = \overline{F_n}$. Define $A = F_n^H F_n$. We want to show $A = nI$. The definition of $A = (a_{lj})$, given as a product of two matrices, gives us

$$\begin{aligned} a_{lj} &= \sum_{k=1}^{n} (F_n^H)_{lk}(F_n)_{kj} = \sum_{k=1}^{n} \overline{w^{(l-1)(k-1)}} w^{(k-1)(j-1)} = \sum_{k=1}^{n} w^{(-l+1)(k-1)} w^{(k-1)(j-1)} \\ &= \sum_{k=1}^{n} w^{(k-1)(1-l+j-1)} = \sum_{k=1}^{n} w^{(k-1)(j-l)}. \end{aligned}$$

We have to distinguish two cases.

1. If $l = j$ then
$$a_{ll} = \sum_{k=1}^{n} w^{(k-1)(l-l)} = \sum_{k=1}^{n} w^{(k-1)\cdot 0} = \sum_{k=1}^{n} 1 = n.$$
That is, $a_{ll} = n$, for all $1 \le l \le n$.
2. If $l \neq j$ then we have
$$a_{lj} = \sum_{k=1}^{n} w^{(k-1)(j-l)} = w^{j-l} \sum_{k=1}^{n} w^{(k-1)} = w^{j-l} \sum_{k=0}^{n-1} w^{(k)} = w^{j-l} \frac{1-w^n}{1-w} = w^{j-l} \frac{0}{1-w} = 0.$$
That is, $a_{jl} = 0$ if $j \neq l$.

Combining those results we have $A = nI$, that is, $F_n^H F_n = nI$.

b) If we define $\tilde{F_n} = \frac{1}{\sqrt{n}} F_n$, by **a)** we have

$$\tilde{F_n}^H \tilde{F_n} = \frac{1}{n} F_n^H F_n = \frac{1}{n} nI = I.$$

c) We want to find a diagonal matrix $D = \text{diag}(d_1, d_2, \dots, d_n)$ such that $P_n F_n = F_n D$. Let us denote $A = P_n F_n$, $C = F_n D$ NS $P_n = (p_{ij}^n)_{i,j=1}^n$. We have $p_{1n}^n = 1, p_{i,i-1} = 1$ for $i = 2, \dots, n$ and otherwise $p_{ij}^n = 0$. Multiplication by a permutation matrix from the right side gives a matrix whose rows are a permutation of the set of rows of the original matrix since we have

$$a_{1j} = \sum_{k=1}^{n} p_{1k}^n f_{kj}^n = 1 \cdot \; f_{nj}^n = w^{(n-1)(j-1)}$$

and

$$a_{ij} = \sum_{k=1}^{n} p_{ij}^n f_{kj}^n = 1 \cdot f_{i-1,j}^n = w^{(i-2)(j-1)}$$

for $i = 2, \dots, n$ and $j = 1, \dots n$. On the other hand

$$c_{ij} = \sum_{k=1}^{n} f_{ik}^n d_{kj} = f_{ij}^n d_j = d_j w^{(i-1)(j-1)}.$$

Therefore $a_{1j} = c_{1j}$ if and only if $d_j = w^{(n-1)(j-1)}$. What is left is to ensure that if we take such a D, that is, if $D = \text{diag}(1, w^{n-1}, w^{2(n-1)}, \dots, w^{(n-1)(j-1)})$ we also have $a_{ij} = c_{ij}$ for $i = 2, \dots, n$ and $j = 1, \dots n$. But, for such i and j we indeed have

$$\begin{aligned} c_{ij} &= d_j w^{(i-1)(j-1)} = w^{(n-1)(j-1)} w^{(i-1)(j-1)} = w^{(i-2+n)(j-1)} = w^{(i-2)(j-1)} w^{n(j-1)} \\ &= w^{(i-2)(j-1)} (w^n)^{j-1} = w^{(i-2)(j-1)} = a_{ij}, \end{aligned}$$

since $w^n = 1$. Therefore, $a_{ij} = c_{ij}$ for all i, j, which means $A = C$, that is, $P_n F_n = F_n D$.

2. Take an $n \in \mathbb{N}$ and an arbitrary $n \times n$ matrix A. We can write this matrix as $A = \begin{pmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix}$,

where $\mathbf{a}_i$ and $\mathbf{a}_i^\top$ is the i-th row of A.

a) For $\alpha \in \mathbb{R}$ we consider

$$\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top + \alpha \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} \overset{Thm}{\underset{5.2.H}{=}} \det(\mathbf{a}_1, \dots, \mathbf{a}_k + \alpha \mathbf{a}_i, \dots, \mathbf{a}_n) \overset{Thm}{\underset{5.2.B}{=}} \det(\mathbf{a}_1, \dots, \mathbf{a}_k, \dots, \mathbf{a}_n)$$

$$\overset{Thm}{\underset{5.2.H}{=}} \det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} = \det A,$$

which is what we wanted to show.

b) For $\alpha \in \mathbb{R}$ we have

$$\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \alpha \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} \overset{Thm}{\underset{5.2.H}{=}} \det(\mathbf{a}_1, \dots, \alpha \mathbf{a}_k, \dots, \mathbf{a}_n) \overset{Thm}{\underset{5.2.C}{=}} \alpha \det(\mathbf{a}_1, \dots, \mathbf{a}_k, \dots, \mathbf{a}_n)$$

$$\overset{Thm}{\underset{5.2.H}{=}} \alpha \det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} = \alpha \det A$$

which is what we wanted to show.

c) From the preceding subproblems we have

$$\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} \overset{\textbf{a)}}{=} \det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top + \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} \overset{\textbf{b)}}{=} -\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top + \mathbf{a}_i^\top \\ \vdots \\ -\mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} \overset{\textbf{a)}}{=} -\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top + \mathbf{a}_i^\top \\ \vdots \\ -\mathbf{a}_i^\top + \mathbf{a}_k^\top + \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix}$$

$$\overset{\textbf{a)}}{=} -\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_k^\top + \mathbf{a}_i^\top - \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} = -\det \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_i^\top \\ \vdots \\ \mathbf{a}_k^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix} = -\det A.$$

3. **a)** From the lectures we know that the solution to the equation $\mathbf{y}'(x) = \mathbf{A}\mathbf{y}(x)$ has the form $\mathbf{y}(x) = \mathbf{S}\begin{pmatrix} \gamma_1 e^{\lambda_1 x} \\ \gamma_1 e^{\lambda_1 x} \end{pmatrix}$ where $\mathbf{A} = \mathbf{S}\mathbf{D}\mathbf{S}^{-1}$ and $\mathbf{D} = \operatorname{diag}(\lambda_1, \lambda_2)$. In our case $\lambda_1 = -1, \lambda_2 = \frac{1}{2}$, and we can write the expression for $\mathbf{y}(x)$ as a linear combination of columns of $\mathbf{S}$ as follows

$$\mathbf{y}(x) = \mathbf{S}\begin{pmatrix} \gamma_1 e^{\lambda_1 x} \\ \gamma_1 e^{\lambda_1 x} \end{pmatrix} = \gamma_1 e^{\lambda_1 x}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \gamma_2 e^{\lambda_2 x}\begin{pmatrix} 2 \\ 3 \end{pmatrix} = \gamma_1 e^{-x}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \gamma_2 e^{\frac{x}{2}}\begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Since $e^{-x} \to 0$ as $x \to \infty$ and $e^{\frac{x}{2}} \to \infty$ as $x \to \infty$, we have that $\mathbf{y}(x) \to \mathbf{0}$ as $x \to \infty$ is possible only if $\gamma_2 = 0$. These parameters γ_1 and γ_2 are determined from the initial value $\mathbf{y}(0)$. So, given a starting vector $\mathbf{y}(0) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ we have

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{y}(0) = \mathbf{S}\begin{pmatrix} \gamma_1 \\ \gamma_1 \end{pmatrix} = \gamma_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \gamma_2 \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Seeing that we need $\gamma_2 = 0$ it follows

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \gamma_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

in other words, we have that $\mathbf{y}(x) \to \mathbf{0}$ if and only if $\mathbf{y}(0) \in \text{Span}\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\}$.

b) From the lectures we know that the solution of the recursion $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{y}^{(k)}$ has the form

$$\mathbf{y}^{(k)} = \mathbf{S}\,\text{diag}(\lambda_1^k, \lambda_2^k)\mathbf{S}^{-1}\mathbf{y}^{(0)}$$

where $\mathbf{A} = \mathbf{SDS}^{-1}$ and $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2)$. In our case $\lambda_1 = -1, \lambda_2 = \frac{1}{2}$, and we can write this expression as

$$\begin{aligned}
\mathbf{y}^{(k)} = \mathbf{S}\,\text{diag}((-1)^k, (\frac{1}{2})^k)\mathbf{S}^{-1}\mathbf{y}^{(0)} &= \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} (-1)^k & 0 \\ 0 & 2^{-k} \end{pmatrix} \begin{pmatrix} 3 & -2 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\
&= (-1)^k(3y_1 - 2y_2)\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 2^{-k}(y_2 - y_1)\begin{pmatrix} 2 \\ 3 \end{pmatrix}.
\end{aligned}$$

Analogously to **a)** we can have $\mathbf{y}^{(k)} \to \mathbf{0}$ as $k \to \infty$ if only if $3y_1 - 2y_2 = 0$, that is, $3y_1 = 2y_2$. This is because $2^{-k} \to 0$ as $k \to \infty$ and the limit of $(-1)^k$ as $k \to \infty$ does not exist. Therefore, we have $\mathbf{y}^{(0)} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{y_1}{2}\begin{pmatrix} 2 \\ 3 \end{pmatrix}$, that is, $\mathbf{y}^{(k)} \to \mathbf{0}$ as $k \to \infty$ if only if $\mathbf{y}(0) \in \text{Span}\left\{\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\}$.

c) From **a)** we know

$$\mathbf{y}(x) = \gamma_1 e^{-x}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \gamma_2 e^{\frac{x}{2}}\begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Therefore, $\mathbf{y}(x)$ will be bounded (as $x \to \infty$), that is, we will not have $\|y(x)\| \to \infty$ as $x \to \infty$ if and only if $\gamma_2 = 0$ since if $\gamma_2 \neq 0$ then

$$\left\|\gamma_2 e^{\frac{x}{2}}\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| = e^{\frac{x}{2}}|\gamma_2|\left\|\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| \to \infty, \text{ as } x \to \infty$$

because $|\gamma_2|\left\|\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| \neq 0$ and $e^{\frac{x}{2}} \to \infty$ as $x \to \infty$. In terms of the discussion in **a)**, the solution $\mathbf{y}(x)$ will be unbounded if and only if $\mathbf{y}(0) \notin \text{Span}\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\}$.

d) From **b)** we have

$$\mathbf{y}^{(k)} = (-1)^k(3y_1 - 2y_2)\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 2^{-k}(y_2 - y_1)\begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

We have

$$\begin{aligned}
\|\mathbf{y}^{(k)}\| &= \left\|(-1)^k(3y_1 - 2y_2)\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 2^{-k}(y_2 - y_1)\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| \\
&\leq |(-1)^k||3y_1 - 2y_2|\left\|\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\| + 2^{-k}|y_2 - y_1|\left\|\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| \\
&\leq |3y_1 - 2y_2|\left\|\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\| + |y_2 - y_1|\left\|\begin{pmatrix} 2 \\ 3 \end{pmatrix}\right\| < \infty.
\end{aligned}$$

Therefore, $\mathbf{y}^{(k)}$ is bounded as $k \to \infty$ for all possible initial vectors $\mathbf{y}^{(0)}$. In other words, the set of all $\mathbf{y}^{(0)}$ for which the limit of $\mathbf{y}^{(k)}$ is unbounded is empty.

4. a) In order to show the statement of the problem it is sufficient to show that $\mathbf{AS} = \mathbf{S}\,\mathrm{diag}\left(2, \frac{1}{3}\right)$ since if this equation holds, then multiplying it by $\mathbf{S}^{-1}$ from the right gives the statement of the problem. We have

$$\mathbf{S}\,\mathrm{diag}\left(2, \frac{1}{3}\right) = \mathbf{S}\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 4 & -\frac{1}{3} \\ -2 & \frac{1}{3} \end{pmatrix}.$$

On the other hand, we have

$$\mathbf{AS} = \begin{pmatrix} \frac{11}{3} & \frac{10}{3} \\ -\frac{5}{3} & -\frac{4}{3} \end{pmatrix}\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{22-10}{3} & \frac{-11+10}{3} \\ \frac{-10+4}{3} & \frac{5-4}{3} \end{pmatrix} = \begin{pmatrix} 4 & -\frac{1}{3} \\ -2 & \frac{1}{3} \end{pmatrix}.$$

Therefore, $\mathbf{AS} = \mathbf{S}\,\mathrm{diag}\left(2, \frac{1}{3}\right)$ and the statement follows.

b) From the lectures for an arbitrary $\mathbf{y}^{(0)} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ we have

$$\mathbf{y}^{(k)} = \mathbf{S}\begin{pmatrix} 2 & 0 \\ 0 & 3^{-k} \end{pmatrix}\mathbf{S}^{-1}\mathbf{y}^{(0)} = \begin{pmatrix} (2^{k+1} - 3^{-k})y_1 + (2^{k+1} - 2\cdot 3^{-k})y_2 \\ (-2^k + 3^{-k})y_1 + (-2^k - 2\cdot 3^{-k})y_2 \end{pmatrix}$$
$$= \begin{pmatrix} 2^k(2y_1 + 2y_2) - 3^{-k}(y_1 + 2y_2) \\ -2^k(y_1 + y_2) + 3^{-k}(y_1 + 2y_2) \end{pmatrix} = 2^k(y_1 + y_2)\begin{pmatrix} 2 \\ 1 \end{pmatrix} + 3^{-k}(y_1 + 2y_2)\begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Taking $\mathbf{y}^{(0)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ we get

$$\mathbf{y}^{(k)} = \begin{pmatrix} 2^{k+1}\cdot 0 - 3^{-k}(-1+2) \\ -2^k \cdot 0 + 3^{-k}(-1+2) \end{pmatrix} = 3^{-k}\begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Since $3^{-k} \to 0$ as $k \to \infty$ we have $\mathbf{y}^{(k)} \to \mathbf{0}$ as $k \to \infty$.

c) Plugging $\mathbf{y}^{(0)} = \begin{pmatrix} -1 + 2\epsilon \\ 1 - \epsilon \end{pmatrix}$ in the expression for $\mathbf{y}^{(k)}$ we get

$$\mathbf{y}^{(k)} = \begin{pmatrix} \epsilon 2^{(1+k)} - 3^{-k} \\ -\epsilon 2^k + 3^{-k} \end{pmatrix} = \begin{pmatrix} 2^{(k-19)} - 3^{-k} \\ -2\cdot 2^{k-19} + 3^{-k} \end{pmatrix}$$

for $\epsilon = 2^{-20}$. We can notice that 2^{k-19} decreases until $k = 19$, but then it begins to increase again, and in fact $2^{k-19} \to \infty$ as $k \to \infty$. Since $3^{-k} \to 0$ as $k \to \infty$, we have $\mathbf{y}^k$ is unbounded in the limit with respect to k.

d) The results presented in the MATLAB code and supported by the plot of $\mathbf{y}^{(k)}$ are not unexpected. This is because of the imperfections that problems related to finite arithmetic bring to the table. Similarly to **c)**, the norm of our vector decreases for some time, but at some point it starts to increase to infinity.

5. a) Let $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. Then, for every $i = 1, \ldots, 15$ we have

$$\begin{pmatrix} p_1^i \\ p_2^i \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix} = \begin{pmatrix} a_{11}x_1^i + a_{12}x_2^i \\ a_{21}x_1^i + a_{22}x_2^i \end{pmatrix}.$$

Since the entries of $\mathbf{A}$ are our unknowns, we want to rewrite this a linear system of the form $\mathbf{p}^i = \mathbf{X}^i\mathbf{a}$ where $\mathbf{a} = \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{pmatrix}$ and $\mathbf{X}^i$ is some matrix. We can do that in the following manner

$$\begin{pmatrix} p_1^i \\ p_2^i \end{pmatrix} = \begin{pmatrix} x_1^i & x_2^i & 0 & 0 \\ 0 & 0 & x_1^i & x_2^i \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{pmatrix}. \tag{1}$$

Now we can build our overdetermined system. Since an equation of the form (1) holds for all $i = 1, \ldots, 15$ we can write

$$\underbrace{\begin{pmatrix} p_1^1 \\ p_2^1 \\ p_1^2 \\ p_2^2 \\ \vdots \\ p_1^{15} \\ p_2^{15} \end{pmatrix}}_{\mathbf{p}} = \underbrace{\begin{pmatrix} x_1^1 & x_2^1 & 0 & 0 \\ 0 & 0 & x_1^1 & x_2^1 \\ x_1^2 & x_2^2 & 0 & 0 \\ 0 & 0 & x_1^2 & x_2^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{15} & x_2^{15} & 0 & 0 \\ 0 & 0 & x_1^{15} & x_2^{15} \end{pmatrix}}_{\mathbf{B}} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{pmatrix}.$$

Hence, $\mathbf{B}$ is a 30×4 matrix, $\mathbf{p}$ is a 30×1 vector and $\mathbf{a}$ is a 4×1 vector.

b) Rank-nullity theorem tells us that $\text{Rang}(B) \leq 4$. Therefore, $\mathbf{B}$ will have full rank if its rank is equal to 4. The rank of $\mathbf{B}$ will be < 4 if and only if its columns are linearly dependent which in our case means that the rank of $\mathbf{B}$ will be strictly smaller than 4 if there exists a $c \in \mathbb{R}$ such that $x_2^i = cx_1^i$ for all $i = 1, \ldots, 15$. Therefore, we would have $\mathbf{x}^i = x_1^i \begin{pmatrix} 1 \\ c \end{pmatrix}$ for all $i = 1, \ldots, 15$. In other words, $\mathbf{B}$ does not have full rank if and only if all points $\mathbf{x}^i$ lie on the same line, i.e., if all points are collinear.

Listing 1: shapeidentmat

c)

```
% Problem 5c)
% INPUT
% X - points
% OUTPUT
% B - matrix of the overdetermined system

function B = shapeidentmat(X)

% Preset B to its size
B = zeros(30, 4);

% Assign B's entries aptly
a1 = [1 0 0 0; 0 0 1 0];
a2 = [0 1 0 0; 0 0 0 1];
for i = 1:15
    B(2*i-1:2*i,:) = X(1,i)*a1+X(2,i)*a2;
end
```

d)

Siehe nächstes Blatt!

Listing 2: complinmap

e)

```
% Problem 5d)-e)
% INPUT
% X - original, non-transformed, points of the image
% P - data about a figure from the tree
% OUTPUT
% A - approximation of the matrix A
% err - error in approximation
function [A, err] = complinmap(X, P)

% Compute the matrix of the overdetermined system
B = shapeidentmat(X);
% Adjust P so that according to the system
P = reshape(P, 30, 1);

% Find the solution of the said system and adjust its form
A = reshape( B\P, 2, 2)';

% Compute the error
err = norm(reshape(P, 2, 15) - A*X);
```

f) Let us assume that we `StarPts` are points of a star, that is, the $\mathbf{x}^i$ vectors. Then, if `PPts` are points given by a photo, and if they denote a star then the code

`[A err] = complinmap(StarPts,PPts)`

gives us a very small residual, that is, the value of `err` is very small. In fact, if there is no noise in `PPTs`, then the error should be of the order 10^{-15} On the other hand, if `PPts` are points given by a photo, and if they denote an Xmas tree then calling the same code will give a much larger error, of the order 10^{1}. Therefore, given `PPts` we can compute `[A err1] = complinmap(StarPts,PPts)` and `[A err2] = complinmap(XmasPts,PPts)`, where `XmasPts` are points of a Xmas tree as given by the code `Xmastree.m`, compare `err1` and `err2` and frome the size of the error infer whether `PPts` are points of a star or of a Xmas tree.

Listing 3: countXmasdecoration

g)

```
% Problem 5g)
% Load the data
load fotodaten.dat;

% Set the initial values of the counters to 0
StarCnt=0;
XmasCnt=0;

% Compute the original points of the star and of the xmas tree
StarPts = star;
XmasPts = Xmastree;
close all;

for i = 1 : 20
    % Load the i-th collection of 15 points denoting a figure
    PtsFromData = fotodaten(2*i-1:2*i, :);
    XmasCn
```

Bitte wenden!

```
    % Compute both errors
    [~, err1] = complinmap(StarPts, PtsFromData);
    [~, err2] = complinmap(XmasPts, PtsFromData);

    % Error comparison yields the results
    if err1 > err2
        XmasCnt = XmasCnt+1;
    else
        StarCnt = StarCnt+1;
    end
end
```

6. **a)** We need to show that $\sum_{i=1}^{N} p_{ij} = 1$, for all $j = 1, \ldots, N$. Let us distinguish two cases.

1. Site j has no links.
   In this case $p_{ij} = \frac{1}{N}$ for all $i = 1, \ldots N$ by the statement of the problem. Therefore

$$\sum_{i=1}^{N} p_{ij} = \sum_{i=1}^{N} \frac{1}{N} = \frac{1}{N} N = 1.$$

2. Site j has $n_j \geq 1$ links.
   If j has links to n_j sites then consequently, j does not have a link to all of the remaining sites, that is, $N - n_j$ of them. Denote by $i_1, \ldots, i_{n_j}$ the sites to which there is a link from j and by $\tilde{i}_1, \ldots, \tilde{i}_{N-n_j}$ all other sites. By the statement of the problem we have $p_{i_k j} = \frac{1-\delta}{n_j} + \frac{\delta}{N}$ for $k = 1, \ldots n_j$ and $p_{\tilde{i}_l j} = \frac{\delta}{N}$ for $l = 1, \ldots, N - n_j$. Now, we have

$$\sum_{i=1}^{N} p_{ij} = \sum_{k=1}^{n_j} p_{i_k j} + \sum_{l=1}^{N-n_j} p_{\tilde{i}_l j} = \sum_{k=1}^{n_j} \left(\frac{1-\delta}{n_j} + \frac{\delta}{N} \right) + \sum_{l=1}^{N-n_j} \frac{\delta}{N}$$
$$= \frac{1-\delta}{n_j} n_j + \sum_{i=1}^{N} \frac{\delta}{N} = 1 - \delta + \frac{\delta}{N} N = 1 - \delta + \delta = 1.$$

We conclude that the sum of entries in any column of $\mathbf{P}$ is indeed 1, hence, $\mathbf{P}$ is a stochastic matrix.

Listing 4: countlinks

b)

```
% Problem 6b)
% INPUT
% G - incidence matrix
% OUTPUT
% nl - vector whose i-th entry is the number of
% links from the i-th site

function nl = countlinks(G)

nl = sum(G)';
```

Listing 5: buildP

Siehe nächstes Blatt!

```
% Problem 12c)
% INPUT
% G - incidence matrix, showing the connectedness between websites
% d - parameter delta
% OUTPUT
% P - transition-probability matrix

function P = buildP(G,d)

N = size(G,1);
% Determine the nj's
l = full(sum(G));
% Determine which sites have at least one link to another site
idx = find(l>0);
s = zeros(N,1);
s(idx) = 1./l(idx);
ds = ones(N,1)/N;
ds(idx) = d/N;

% Building the matrix by using the expression in part f) of the problem
P = ones(N,1)*ones(1,N)*diag(ds) + (1-d)*G*diag(s);
```

Listing 6: HarvardPlot

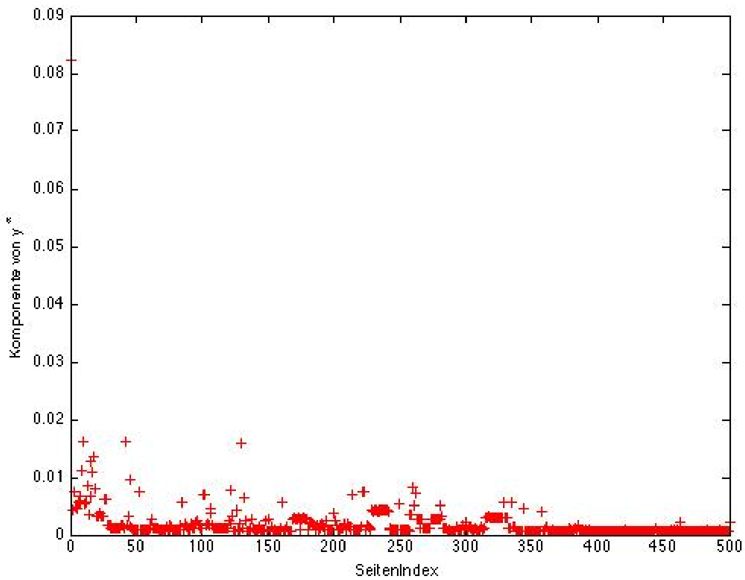
d)

```

% Load the data
load harvard500.mat;

% Build the transition matrix P
d = 0.15;
P = buildP(G,d);
N = size(G, 1);
% Take an arbitrary inital values
y = rand(size(G, 1), 1);
% Normalise y so that the sum of its elements is 1 (so that it represents
% probabilities)
y = y/sum(y);
yold = ones(size(y));
% Tolerance level
tol = 1e-6;

while norm(yold-y)/norm(yold) > tol
    yold = y;
    % Straightforward computation of the next iterant
    y = P*y;
end

% Plotting
plot((1:N)',y,'r+');
xlabel('SeitenIndex');
ylabel('Komponente von y^{\infty}');
```

This code produces figure 1.

e) In each iteration we compute $\mathbf{P}\mathbf{y}^{(k)}$, so we want to know the computational complexity of this operation. The fact that the total number of links is $\leq 5N$ means that the incidence

Abbildung 1: HarvardPlot



matrix **G** is linearly sparse. That means that only linearly many entries of **P** are not $\frac{1}{N}$. Therefore, the computational cost is $\mathcal{O}(N^2)$.

NB: The computational cost of computing a matrix-vector product with full (i.e. non-sparse) matrix is usually considered to be $\mathcal{O}(N^3)$.

f) We can see from the definition of **P**, that we can write **P** as a sum of two matrices, **A** and **B**, such that

$$a_{ij} = \begin{cases} \frac{1}{N}, & n_j = 0 \\ \frac{\delta}{N}, & \text{otherwise} \end{cases} .$$

and

$$b_{ij} = \begin{cases} \frac{1-\delta}{n_j}, & \text{if there is a link from } j \text{ to } i \\ 0, & \text{otherwise} \end{cases} .$$

We now have $\mathbf{P} = \mathbf{A} + \mathbf{B}$.

The definition of **A** states that **A** has constant columns, i.e., for a given column j of **A** all the entries in that column are equal, $a_{ij} = a_{kj}$ for all $i, k = 1, \ldots, N$. On the other hand, the matrix $\mathbf{1}\mathbf{1}^\top$ has all entries equal to 1, thus, for $\mathbf{D}_1 = \operatorname{diag} d_1, \ldots, d_n)$ the entries of $\mathbf{1}\mathbf{1}^\top \mathbf{D}_1$ are

$$(\mathbf{1}\mathbf{1}^\top \mathbf{D}_1)_{ij} = d_j$$

In other words, it has constant columns and all entries in the j-th column are equal to d_j. Therefore, by defining

$$d_j = \begin{cases} \frac{1}{N}, & n_j = 0 \\ \frac{\delta}{N}, & \text{otherwise} \end{cases}$$

We have $\mathbf{A} = \mathbf{1}\mathbf{1}^\top \mathbf{D}_1$ by the preceding arguments.

Let us switch our attention to **B** now. Notice that $b_{ij} \neq 0$ if and only if $g_{ij} = 1$, where $\mathbf{G} = (g_{ij})_{i,j=1}^N$, because of the definitions of **B** and **G**. Now, multiplying a matrix $\mathbf{D}_2 =$

$\text{diag}(c_1, \dots, c_N)$ with a diagonal matrix from the right hand side only scales the columns of the given matrix, that is

$$(\mathbf{G}\mathbf{D}_{ij}) = g_{ij}c_j.$$

Therefore, if we define

$$c_j = \begin{cases} \frac{1}{n_j}, & \text{there is no link from j to i} \\ 0, & \text{otherwise} \end{cases}$$

then by the preceding we have $(1-\delta)\mathbf{G}\mathbf{D}_2 = B$.

Combining the expressions for $\mathbf{A}$ and $\mathbf{B}$ we have

$$\mathbf{P} = \mathbf{1}\mathbf{1}^\top \mathbf{D}_1 + (1-\delta)\mathbf{G}\mathbf{D}_2.$$

g) Using the expression from subproblem **f)** we have that $\mathbf{P}\mathbf{v}$, for $\mathbf{v} \in \mathbb{R}^N$ is split into two parts. First, we have the expression $\mathbf{1}\mathbf{1}^\top \mathbf{D}_1\mathbf{v}$. Since $\mathbf{D}_1$ is a diagonal matrix, $\mathbf{D}_1\mathbf{v}$ is a vector and it is computed in linear time, as we only need N multiplications. That is, the computational complexity of that operation is $\mathcal{O}(N)$. Let us denote that vector with $\mathbf{w}_1 = \mathbf{D}_1\mathbf{v}$ Then, $\mathbf{1}^\top\mathbf{w}_1$ is a scalar, that is $\mathbf{1}^\top\mathbf{w}_1 = \langle \mathbf{1}, \mathbf{w}_1 \rangle \in \mathbb{R}$, and as such, it is again computed in linear time. i.e., $\mathcal{O}(N)$. Lastly, since $\alpha = \mathbf{1}^\top\mathbf{w}$ is just a number, then $\mathbf{1}\alpha$ is a scalar-vector operation, which can again be computed in linear time. Therefore, the computational complexity of $\mathbf{1}\mathbf{1}^\top\mathbf{D}_1\mathbf{v}$ is $\mathcal{O}(N)$.

Secondly, we need to compute $(1-\delta)\mathbf{G}\mathbf{D}_2\mathbf{v}$. Again, since $\mathbf{D}_2$ is a diagonal matrix, $\mathbf{w}_2 = \mathbf{D}_2\mathbf{v}$ is a vector and its computational complexity is $\mathcal{O}(N)$. Since $\mathbf{G}$ is linearly sparse, i.e., since it has only up to $5N$ non-zero entries, the product $\mathbf{G}\mathbf{w}_2$ has linear computational complexity, that is $\mathcal{O}(N)$. The last part is a mulitplication by a scalar $(1-\delta)$ which has compuational complexity $\mathcal{O}(N)$. Therefore, the computational complexity of $(1-\delta)\mathbf{G}\mathbf{D}_2\mathbf{v}$ is $\mathcal{O}(N)$.

What remains is to add two vectors, $\mathbf{1}\mathbf{1}^\top\mathbf{D}_1\mathbf{v}$ and $(1-\delta)\mathbf{G}\mathbf{D}_2\mathbf{v}$. The computational complexity of this is again $\mathcal{O}(N)$. Finally, we have that the computation complexity of multiplying $\mathbf{1}\mathbf{1}^\top\mathbf{D}_1 + (1-\delta)\mathbf{G}\mathbf{D}_2$ with a vector is $\mathcal{O}(N)$.