Lagoon Overflow Project

Introdution

- Def. Let $\mathbf{A} \in \mathbb{C}^{N\times N}$ and $0 \neq \mathbf{v} \in \mathbb{C}^{N}$ then n^{th} Krylov subspace generated by $\mathbf{A}$ from $\mathbf{v}$ is defined by:

$$\mathcal{K}_n = \{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \cdots, \mathbf{A}^{n-1}\mathbf{v}\}.$$

Obviously, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots$.

Introdution

- Def. Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ and $0 \neq \mathbf{v} \in \mathbb{C}^N$ then n^{th} Krylov subspace generated by $\mathbf{A}$ from $\mathbf{v}$ is defined by:

$$\mathcal{K}_n = \{\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v}, \cdots, \mathbf{A}^{n-1}\mathbf{v}\}.$$

  Obviously, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots$.

-

Introdution

- Def. Let $\mathbf{A} \in \mathbb{C}^{N\times N}$ and $0 \neq \mathbf{v} \in \mathbb{C}^N$ then n^{th} Krylov subspace generated by $\mathbf{A}$ from $\mathbf{v}$ is defined by:

$$\mathcal{K}_n = \{\mathbf{v}, \mathbf{Av}, \mathbf{A}^2\mathbf{v}, \cdots, \mathbf{A}^{n-1}\mathbf{v}\}.$$

Obviously, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots$.

Lemma

There is a positive integer $1 \leq \mu \leq N$ such that

$$dim(\mathcal{K}_n(A, v)) = \begin{cases} n & if \quad n \leq \mu \\ \mu & if \quad n \geq \mu \end{cases}$$

μ is called grade of v with respect to A.

-

Introdution

- Def. Let $\mathbf{A} \in \mathbb{C}^{N\times N}$ and $0 \neq \mathbf{v} \in \mathbb{C}^N$ then n^{th} Krylov subspace generated by $\mathbf{A}$ from $\mathbf{v}$ is defined by:

  $$\mathcal{K}_n = \{\mathbf{v}, \mathbf{Av}, \mathbf{A}^2\mathbf{v}, \cdots, \mathbf{A}^{n-1}\mathbf{v}\}.$$

  Obviously, $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots$.

Lemma

There is a positive integer $1 \leq \mu \leq N$ such that

$$dim(\mathcal{K}_n(A, v)) = \begin{cases} n & if \quad n \leq \mu \\ \mu & if \quad n \geq \mu \end{cases}$$

μ is called grade of v with respect to A.

-
- As soon as we hit the invariant subspace of $\mathbf{A}$, we have the exact solution.

Introdution

- Def. A (standard) Krylov space method for solving linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is an iterative method that starts with an initial guess $\mathbf{x}_0$ and generates approximations

$$\mathbf{x}_n = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0).$$

where q_{n-1} is of exact degree $n-1$.

- Def. A (standard) Krylov space method for solving linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is an iterative method that starts with an initial guess $\mathbf{x}_0$ and generates approximations

$$\mathbf{x}_n = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0).$$

where q_{n-1} is of exact degree $n-1$.

Introdution

- Def. A (standard) Krylov space method for solving linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is an iterative method that starts with an initial guess $\mathbf{x}_0$ and generates approximations

$$\mathbf{x}_n = \mathbf{x}_0 + q_{n-1}(\mathbf{A})\mathbf{r}_0 \in \mathbf{x}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0).$$

where q_{n-1} is of exact degree $n-1$.
Therefore,

$$\mathbf{r}_n = \mathbf{r}_0 - \mathbf{A}q_{n-1}(\mathbf{A})\mathbf{r}_0 = p_n(\mathbf{A})\mathbf{r}_0 \in \mathbf{r}_0 + \mathbf{A}\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) \subseteq \mathcal{K}_{n+1}$$

Where degree of p_n is exactly n, i.e. $p_n \in \mathbf{P}_n \backslash \mathbf{P}_{n-1}$.

- Note that $p_n(z) = 1 - zq_{n-1}(z)$ and therefore, $p_n(0) = 1$.

IDR for Solving Linear Systems

IDR is a Krylov subspace solver with one more flavor. The residuals in IDR are forced to be in subspaces $\mathcal{G}_j$ of decreasing dimension.

IDR for Solving Linear Systems

IDR is a Krylov subspace solver with one more flavor. The residuals in IDR are forced to be in subspaces $\mathcal{G}_j$ of decreasing dimension.These nested subspaces are defined by:

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$ and $0 \neq \omega_j$'s are scalars.

IDR for Solving Linear Systems

IDR is a Krylov subspace solver with one more flavor. The residuals in IDR are forced to be in subspaces $\mathcal{G}_j$ of decreasing dimension. These nested subspaces are defined by:

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$ and $0 \neq \omega_j$'s are scalars.

Theorem

Let $\mathbf{A}$ be any matrix in $\mathbb{C}^{N \times N}$ and let $v \neq 0$ be any vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the full Krylov space $\mathcal{K}_{N+1}(\mathbf{A}, \mathbf{v})$. Let $\mathcal{S}$ be any (proper) subspace of $\mathbb{C}^N$ such that $\mathcal{G}_0$ and $\mathcal{S}$ do not share any nontrivial invariant subspace of $\mathbf{A}$ and define $\mathcal{G}_j$ by

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

for $j = 1, 2, 3, \cdots$ where $0 \neq \omega_j$ are scalars then:

IDR for Solving Linear Systems

IDR is a Krylov subspace solver with one more flavor. The residuals in IDR are forced to be in subspaces $\mathcal{G}_j$ of decreasing dimension.These nested subspaces are defined by:

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$ and $0 \neq \omega_j$'s are scalars.

Theorem

Let $\mathbf{A}$ be any matrix in $\mathbb{C}^{N \times N}$ and let $v \neq 0$ be any vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the full Krylov space $\mathcal{K}_{N+1}(\mathbf{A}, \mathbf{v})$. Let $\mathcal{S}$ be any (proper) subspace of $\mathbb{C}^N$ such that $\mathcal{G}_0$ and $\mathcal{S}$ do not share any nontrivial invariant subspace of $\mathbf{A}$ and define $\mathcal{G}_j$ by

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

for $j = 1, 2, 3, \cdots$ where $0 \neq \omega_j$ are scalars then:
(i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$, $\forall j > 0$.

IDR for Solving Linear Systems

IDR is a Krylov subspace solver with one more flavor. The residuals in IDR are forced to be in subspaces $\mathcal{G}_j$ of decreasing dimension.These nested subspaces are defined by:

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

where $\mathcal{S}$ is a fixed proper subspace of $\mathbb{C}^N$ and $0 \neq \omega_j$'s are scalars.

Theorem

Let $\mathbf{A}$ be any matrix in $\mathbb{C}^{N\times N}$ and let $v \neq 0$ be any vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the full Krylov space $\mathcal{K}_{N+1}(\mathbf{A}, \mathbf{v})$. Let $\mathcal{S}$ be any (proper) subspace of $\mathbb{C}^N$ such that $\mathcal{G}_0$ and $\mathcal{S}$ do not share any nontrivial invariant subspace of $\mathbf{A}$ and define $\mathcal{G}_j$ by

$$\mathcal{G}_j = (\mathbf{I} - \omega_j \mathbf{A})(\mathcal{G}_{j-1} \cap \mathcal{S})$$

for $j = 1, 2, 3, \cdots$ where $0 \neq \omega_j$ are scalars then:
(i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}, \forall j > 0.$
(ii) $\mathcal{G}_j = \{\mathbf{0}\}$ *for some* $j \leq N$

Derivation of IDR Algorithm

Usually in Krylov solvers we find $\mathbf{x}_n$ then we find corresponding $\mathbf{r}_n$. In IDR we will do the other way arround, if a recurssion for $\mathbf{r}_n$'s is given, we have to be able to find corresponding $\mathbf{x}_n$'s.

Derivation of IDR Algorithm

Usually in Krylov solvers we find $\mathbf{x}_n$ then we find corresponding $\mathbf{r}_n$. In IDR we will do the other way arround, if a recurssion for $\mathbf{r}_n$'s is given, we have to be able to find corresponding $\mathbf{x}_n$'s. Assuming this is possible we have the following, Let $\nabla u_k := u_k - u_{k-1}$.

$$
\begin{aligned}
\mathbf{A}\nabla\mathbf{x}_{n+1} = \mathbf{A}(\mathbf{x}_{n+1} - \mathbf{x}_n) &= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n \\
&= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{b} + \mathbf{b} - \mathbf{A}\mathbf{x}_n \\
&= -\mathbf{r}_{n+1} + \mathbf{r}_n \\
&= -p_{n+1}(\mathbf{A})\mathbf{r}_0 + p_n(\mathbf{A})\mathbf{r}_0 \\
&= (p_n(\mathbf{A}) - p_{n+1}(\mathbf{A}))\mathbf{r}_0
\end{aligned}
$$

Derivation of IDR Algorithm

Usually in Krylov solvers we find $\mathbf{x}_n$ then we find corresponding $\mathbf{r}_n$. In IDR we will do the other way arround, if a recurssion for $\mathbf{r}_n$'s is given, we have to be able to find corresponding $\mathbf{x}_n$'s. Assuming this is possible we have the following, Let $\nabla u_k := u_k - u_{k-1}$.

$$\begin{aligned}
\mathbf{A}\nabla\mathbf{x}_{n+1} = \mathbf{A}(\mathbf{x}_{n+1} - \mathbf{x}_n) &= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n \\
&= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{b} + \mathbf{b} - \mathbf{A}\mathbf{x}_n \\
&= -\mathbf{r}_{n+1} + \mathbf{r}_n \\
&= -p_{n+1}(\mathbf{A})\mathbf{r}_0 + p_n(\mathbf{A})\mathbf{r}_0 \\
&= (p_n(\mathbf{A}) - p_{n+1}(\mathbf{A}))\mathbf{r}_0
\end{aligned}$$

We can find $\mathbf{x}_{n+1}$ here without solving the system if $p_n(z) - p_{n+1}(z)$ is divisible by z, as follows:

Derivation of IDR Algorithm

Usually in Krylov solvers we find $\mathbf{x}_n$ then we find corresponding $\mathbf{r}_n$. In IDR we will do the other way arround, if a recurssion for $\mathbf{r}_n$'s is given, we have to be able to find corresponding $\mathbf{x}_n$'s. Assuming this is possible we have the following, Let $\nabla u_k := u_k - u_{k-1}$.

$$\begin{aligned}
\mathbf{A}\nabla\mathbf{x}_{n+1} = \mathbf{A}(\mathbf{x}_{n+1} - \mathbf{x}_n) &= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n \\
&= \mathbf{A}\mathbf{x}_{n+1} - \mathbf{b} + \mathbf{b} - \mathbf{A}\mathbf{x}_n \\
&= -\mathbf{r}_{n+1} + \mathbf{r}_n \\
&= -p_{n+1}(\mathbf{A})\mathbf{r}_0 + p_n(\mathbf{A})\mathbf{r}_0 \\
&= (p_n(\mathbf{A}) - p_{n+1}(\mathbf{A}))\mathbf{r}_0
\end{aligned}$$

We can find $\mathbf{x}_{n+1}$ here without solving the system if $p_n(z) - p_{n+1}(z)$ is divisible by z, as follows:

$$\cancel{\mathbf{A}}\nabla\mathbf{x}_{n+1} = \cancel{\mathbf{A}}\psi_n(\mathbf{A})\mathbf{r}_0$$

Derivation of IDR Algorithm

We saw in any Krylov solvers $\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{n+1}(\backslash\mathcal{K}_n)$ where degree of p_n is exactly n.

Derivation of IDR Algorithm

We saw in any Krylov solvers $\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{n+1}(\backslash\mathcal{K}_n)$ where degree of p_n is exactly n. Therefore, we have:

$$\mathbf{r}_n = \sum_{k=0}^{n} \alpha_k \mathbf{A}^k \mathbf{r}_0 \ , \ \alpha_n \neq 0$$

Derivation of IDR Algorithm

We saw in any Krylov solvers $\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{n+1}(\backslash\mathcal{K}_n)$ where degree of p_n is exactly n. Therefore, we have:

$$\mathbf{r}_n = \sum_{k=0}^{n} \alpha_k \mathbf{A}^k \mathbf{r}_0 \ , \ \alpha_n \neq 0$$

Let $\mathbf{v}_n \in \mathcal{K}_{n+1} \backslash \mathcal{K}_n$ be any arbitrary vector, then $\mathbf{A}\mathbf{v}_n \in \mathcal{K}_{n+2} \backslash \mathcal{K}_{n+1}$.

Derivation of IDR Algorithm

We saw in any Krylov solvers $\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{n+1}(\backslash\mathcal{K}_n)$ where degree of p_n is exactly n. Therefore, we have:

$$\mathbf{r}_n = \sum_{k=0}^{n} \alpha_k \mathbf{A}^k \mathbf{r}_0 \ , \ \alpha_n \neq 0$$

Let $\mathbf{v}_n \in \mathcal{K}_{n+1} \backslash \mathcal{K}_n$ be any arbitrary vector, then $\mathbf{A}\mathbf{v}_n \in \mathcal{K}_{n+2} \backslash \mathcal{K}_{n+1}$. Hence, any arbitrary vector in $\mathcal{K}_{n+2} \backslash \mathcal{K}_{n+1}$ can be written as:

$$-\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$$

Derivation of IDR Algorithm

We saw in any Krylov solvers $\mathbf{r}_n = p_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{n+1}(\backslash\mathcal{K}_n)$ where degree of p_n is exactly n. Therefore, we have:

$$\mathbf{r}_n = \sum_{k=0}^{n} \alpha_k \mathbf{A}^k \mathbf{r}_0 \ , \ \alpha_n \neq 0$$

Let $\mathbf{v}_n \in \mathcal{K}_{n+1} \backslash \mathcal{K}_n$ be any arbitrary vector, then $\mathbf{A}\mathbf{v}_n \in \mathcal{K}_{n+2} \backslash \mathcal{K}_{n+1}$. Hence, any arbitrary vector in $\mathcal{K}_{n+2} \backslash \mathcal{K}_{n+1}$ can be written as:

$$-\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$$

Therefore, we can write:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$$

Derivation of IDR Algorithm

Our goal was computing $\mathbf{x}_{n+1}$ provided that $\mathbf{r}_{n+1}$ is known. Let's replace r_i's by $\mathbf{b} - \mathbf{A}\mathbf{x}_i$ in above formula to get $\mathbf{x}_{n+1}$:

Derivation of IDR Algorithm

Our goal was computing $\mathbf{x}_{n+1}$ provided that $\mathbf{r}_{n+1}$ is known. Let's replace r_i's by $\mathbf{b} - \mathbf{A}\mathbf{x}_i$ in above formula to get $\mathbf{x}_{n+1}$:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} \longrightarrow$$

$$\begin{aligned}
\mathbf{b} - \mathbf{A}\mathbf{x}_{n+1} &= -\alpha \mathbf{A}\mathbf{v}_n + \textstyle\sum_{k=0}^{n} \beta_k (\mathbf{b} - \mathbf{A}\mathbf{x}_{n-k}) \\
&= (\sum_{k=0}^{n} \beta_k)\mathbf{b} - \mathbf{A}(\alpha \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{x}_{n-k})
\end{aligned}$$

Derivation of IDR Algorithm

Our goal was computing $\mathbf{x}_{n+1}$ provided that $\mathbf{r}_{n+1}$ is known. Let's replace r_i's by $\mathbf{b} - \mathbf{A}\mathbf{x}_i$ in above formula to get $\mathbf{x}_{n+1}$:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} \longrightarrow$$

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x}_{n+1} &= -\alpha \mathbf{A}\mathbf{v}_n + \textstyle\sum_{k=0}^{n} \beta_k (\mathbf{b} - \mathbf{A}\mathbf{x}_{n-k}) \\ &= (\sum_{k=0}^{n} \beta_k)\mathbf{b} - \mathbf{A}(\alpha \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{x}_{n-k}) \end{aligned}$$

Consequently $\mathbf{x}_{n+1}$ would be solution of

$$\mathbf{A}\mathbf{x}_{n+1} = (1 - \sum_{k=0}^{n} \beta_k)\mathbf{b} + \mathbf{A}(\alpha \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{x}_{n-k}).$$

Derivation of IDR Algorithm

Our goal was computing $\mathbf{x}_{n+1}$ provided that $\mathbf{r}_{n+1}$ is known. Let's replace r_i's by $\mathbf{b} - \mathbf{A}\mathbf{x}_i$ in above formula to get $\mathbf{x}_{n+1}$:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} \longrightarrow$$

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x}_{n+1} &= -\alpha \mathbf{A}\mathbf{v}_n + \textstyle\sum_{k=0}^{n} \beta_k (\mathbf{b} - \mathbf{A}\mathbf{x}_{n-k}) \\ &= (\sum_{k=0}^{n} \beta_k)\mathbf{b} - \mathbf{A}(\alpha \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{x}_{n-k}) \end{aligned}$$

Consequently $\mathbf{x}_{n+1}$ would be solution of

$$\mathbf{A}\mathbf{x}_{n+1} = (1 - \sum_{k=0}^{n} \beta_k)\mathbf{b} + \mathbf{A}(\alpha \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{x}_{n-k}).$$

$\mathbf{x}_{n+1}$ is computable without solving the system if $(1 - \sum_{k=0}^{n} \beta_k) = 0$.

Derivation of IDR Algorithm

We want $1 - \sum_{k=0}^{n} \beta_k = 0$ or equivalently $\sum_{k=0}^{n} \beta_k = 1$. This equation can be established by:

$$\sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} = \mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}.$$

Derivation of IDR Algorithm

We want $1 - \sum_{k=0}^{n} \beta_k = 0$ or equivalently $\sum_{k=0}^{n} \beta_k = 1$. This equation can be established by:

$$\sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} = \mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}.$$

Because,

$$\begin{aligned}
\beta_0 \mathbf{r}_n + \beta_1 \mathbf{r}_{n-1} + \cdots + \beta_n \mathbf{r}_0 &= \mathbf{r}_n - \gamma_1 \nabla \mathbf{r}_n - \gamma_2 \nabla \mathbf{r}_{n-1} - \cdots \gamma_n \nabla \mathbf{r}_1 \\
&= (1-\gamma_1)\mathbf{r}_n + (\gamma_1 - \gamma_2)\mathbf{r}_{n-1} + \cdots \\
&\qquad + (\gamma_{n-1} - \gamma_n) r_1 + \gamma_n \mathbf{r}_0
\end{aligned}$$

Derivation of IDR Algorithm

We want $1 - \sum_{k=0}^{n} \beta_k = 0$ or equivalently $\sum_{k=0}^{n} \beta_k = 1$. This equation can be established by:

$$\sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k} = \mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}.$$

Because,

$$\begin{aligned} \beta_0 \mathbf{r}_n + \beta_1 \mathbf{r}_{n-1} + \cdots + \beta_n \mathbf{r}_0 &= \mathbf{r}_n - \gamma_1 \nabla \mathbf{r}_n - \gamma_2 \nabla \mathbf{r}_{n-1} - \cdots \gamma_n \nabla \mathbf{r}_1 \\ &= (1 - \gamma_1)\mathbf{r}_n + (\gamma_1 - \gamma_2)\mathbf{r}_{n-1} + \cdots \\ &\qquad + (\gamma_{n-1} - \gamma_n) r_1 + \gamma_n \mathbf{r}_0 \end{aligned}$$

which implies:

$$\begin{aligned} \beta_0 &= 1 - \cancel{\gamma_1} \\ \beta_1 &= \cancel{\gamma_1} - \cancel{\gamma_2} \\ &\vdots \\ \beta_n &= \cancel{\gamma_1} \end{aligned}$$

Derivation of IDR Algorithm

So, in general for any Krylov solver we can write:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A} \mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}.$$

So, in general for any Krylov solver we can write:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}.$$

Now let's replace $\sum\limits_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$ by $\mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}$ to get:

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A}\mathbf{v}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1} \tag{2.1}$$

Derivation of IDR Algorithm

So, in general for any Krylov solver we can write:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}.$$

Now let's replace $\sum\limits_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$ by $\mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}$ to get:

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A}\mathbf{v}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1} \tag{2.1}$$

In equation (2.1) we are using all residuals from 0 to n. Depth of recurssion is so long, in order to get short recurssions we need to have $\gamma_k = 0$ for $k \geq m$ where $m \ll N$.

Derivation of IDR Algorithm

So, in general for any Krylov solver we can write:

$$\mathbf{r}_{n+1} = -\alpha \mathbf{A}\mathbf{v}_n + \sum_{k=0}^{n} \beta_k \mathbf{r}_{n-k}.$$

Now let's replace $\sum\limits_{k=0}^{n} \beta_k \mathbf{r}_{n-k}$ by $\mathbf{r}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1}$ to get:

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A}\mathbf{v}_n - \sum_{k=1}^{n} \gamma_k \nabla \mathbf{r}_{n-k+1} \tag{2.1}$$

In equation (2.1) we are using all residuals from 0 to n. Depth of recurssion is so long, in order to get short recurssions we need to have $\gamma_k = 0$ for $k \geq m$ where $m \ll N$.So, I will change the upper limit from n to l.

Derivation of IDR Algorithm

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A}\mathbf{v}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \qquad (2.2)$$

Equation (2.2) can be written in general for all Krylov solvers. It is time to add IDR flavor.

Derivation of IDR Algorithm

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A} \mathbf{v}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \tag{2.2}$$

Equation (2.2) can be written in general for all Krylov solvers. It is time to add IDR flavor.

We want to force residuals to be in $\mathcal{G}_{j+1} := (\mathbf{I} - \omega_{j+1}\mathbf{A})(\mathcal{G}_j \cap \mathcal{S})$. Therefore, there should be a $\mathbf{v}_n \in \mathcal{G}_j \cap \mathcal{S}$ such that

$$\mathbf{r}_{n+1} = (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \tag{2.3}$$

Derivation of IDR Algorithm

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha \mathbf{A}\mathbf{v}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \tag{2.2}$$

Equation (2.2) can be written in general for all Krylov solvers. It is time to add IDR flavor.

We want to force residuals to be in $\mathcal{G}_{j+1} := (\mathbf{I} - \omega_{j+1}\mathbf{A})(\mathcal{G}_j \cap \mathcal{S})$. Therefore, there should be a $\mathbf{v}_n \in \mathcal{G}_j \cap \mathcal{S}$ such that

$$\mathbf{r}_{n+1} = (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \tag{2.3}$$

Equating (2.2) and (2.3) leads to

$$\mathbf{v}_n := \mathbf{r}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \quad \text{and} \quad \alpha := \omega_{j+1}$$

Derivation of IDR Algorithm

Making this change in (2.2) gives the following:

$$\begin{cases} \mathbf{r}_{n+1} &= (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n &= \mathbf{r}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \in \mathcal{G}_j \cap \mathcal{S} \end{cases} \tag{2.4}$$

Making this change in (2.2) gives the following:

$$\left\{ \begin{array}{lcl} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \in \mathcal{G}_j \cap \mathcal{S} \end{array} \right. \tag{2.4}$$

Finding $\mathbf{v}_n$ and ω_j would be a great step toward our goal. So, let $\mathcal{S} := \mathcal{N}(\mathbf{P}^*)$ where $\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_s]$. $\mathcal{S}$ has codimension s.

Derivation of IDR Algorithm

Making this change in (2.2) gives the following:

$$\begin{cases} \mathbf{r}_{n+1} &= (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n &= \mathbf{r}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \in \mathcal{G}_j \cap \mathcal{S} \end{cases} \tag{2.4}$$

Finding $\mathbf{v}_n$ and ω_j would be a great step toward our goal. So, let $\mathcal{S} := \mathcal{N}(\mathbf{P}^*)$ where $\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_s]$. $\mathcal{S}$ has codimension s.

Since $\mathbf{v}_n \in \mathcal{S}$, we have to have $\mathbf{P}^*\mathbf{v}_n = 0$, and consequently,

$$\mathbf{P}^*(\sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1}) = \mathbf{P}^*\mathbf{r}_n$$

Derivation of IDR Algorithm

Making this change in (2.2) gives the following:

$$\begin{cases} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1} \in \mathcal{G}_j \cap \mathcal{S} \end{cases} \tag{2.4}$$

Finding $\mathbf{v}_n$ and ω_j would be a great step toward our goal. So, let $\mathcal{S} := \mathcal{N}(\mathbf{P}^*)$ where $\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_s]$. $\mathcal{S}$ has codimension s.

Since $\mathbf{v}_n \in \mathcal{S}$, we have to have $\mathbf{P}^*\mathbf{v}_n = 0$, and consequently,

$$\mathbf{P}^*(\sum_{k=1}^{l} \gamma_k \nabla \mathbf{r}_{n-k+1}) = \mathbf{P}^*\mathbf{r}_n$$

where γ_k's are unknowns, so, we have a linear system of s equations in l unknowns, which is uniquely solvable if $l = s$.

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form:

Derivation of IDR Algorithm

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form: Let

$\nabla \mathbf{R}_{n:n-s+1} := [\nabla \mathbf{r}_n \quad \nabla \mathbf{r}_{n-1} \quad \cdots \quad \nabla \mathbf{r}_{n-s+1}]$ and $\mathbf{c}_n = [\gamma_1^{(n)} \quad \gamma_2^{(n)} \cdots \gamma_s^{(n)}]^T$ gives the following:

Derivation of IDR Algorithm

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form: Let

$\nabla \mathbf{R}_{n:n-s+1} := [\nabla \mathbf{r}_n \quad \nabla \mathbf{r}_{n-1} \quad \cdots \quad \nabla \mathbf{r}_{n-s+1}]$ and $\mathbf{c}_n = [\gamma_1^{(n)} \quad \gamma_2^{(n)} \cdots \gamma_s^{(n)}]^T$ gives the following:

$$\begin{cases} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \nabla \mathbf{R}_{n:n-s+1}\mathbf{c}_n \end{cases} \tag{2.5}$$

Derivation of IDR Algorithm

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form: Let

$\nabla \mathbf{R}_{n:n-s+1} := [\nabla \mathbf{r}_n \quad \nabla \mathbf{r}_{n-1} \quad \cdots \quad \nabla \mathbf{r}_{n-s+1}]$ and $\mathbf{c}_n = [\gamma_1^{(n)} \quad \gamma_2^{(n)} \cdots \gamma_s^{(n)}]^T$ gives the following:

$$\begin{cases} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \nabla \mathbf{R}_{n:n-s+1}\mathbf{c}_n \end{cases} \tag{2.5}$$

For computing $\mathbf{v}_n$ we need $s+1$ residuals, therefore, computing the first residual in $\mathcal{G}_{j+1}$ needs $s+1$ residuals in $\mathcal{G}_j$. So, we can expect $\mathbf{r}_{n+1}$ be in $\mathcal{G}_{j+1}$ if $n+1 \geq (j+1)(s+1)$.

Derivation of IDR Algorithm

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form: Let

$\nabla \mathbf{R}_{n:n-s+1} := [\nabla \mathbf{r}_n \quad \nabla \mathbf{r}_{n-1} \quad \cdots \quad \nabla \mathbf{r}_{n-s+1}]$ and $\mathbf{c}_n = [\gamma_1^{(n)} \quad \gamma_2^{(n)} \cdots \gamma_s^{(n)}]^T$ gives the following:

$$\begin{cases} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \nabla \mathbf{R}_{n:n-s+1}\mathbf{c}_n \end{cases} \tag{2.5}$$

For computing $\mathbf{v}_n$ we need $s+1$ residuals, therefore, computing the first residual in $\mathcal{G}_{j+1}$ needs $s+1$ residuals in $\mathcal{G}_j$. So, we can expect $\mathbf{r}_{n+1}$ be in $\mathcal{G}_{j+1}$ if $n+1 \geq (j+1)(s+1)$.

Therefore the relationship between n and j will be $j = \lfloor \frac{n}{s+1} \rfloor$.

Derivation of IDR Algorithm

By letting $l = s$ and introducing the following new notaions equation (2.4) can be written in more compact form: Let

$\nabla\mathbf{R}_{n:n-s+1} := [\nabla\mathbf{r}_n \quad \nabla\mathbf{r}_{n-1} \quad \cdots \quad \nabla\mathbf{r}_{n-s+1}]$ and
$\mathbf{c}_n = [\gamma_1^{(n)} \quad \gamma_2^{(n)} \cdots \gamma_s^{(n)}]^T$ gives the following:

$$\begin{cases} \mathbf{r}_{n+1} & = & (\mathbf{I} - \omega_{j+1}\mathbf{A})\mathbf{v}_n \\ \mathbf{v}_n & = & \mathbf{r}_n - \nabla\mathbf{R}_{n:n-s+1}\mathbf{c}_n \end{cases} \tag{2.5}$$

For computing $\mathbf{v}_n$ we need $s+1$ residuals, therefore, computing the first residual in $\mathcal{G}_{j+1}$ needs $s+1$ residuals in $\mathcal{G}_j$. So, we can expect $\mathbf{r}_{n+1}$ be in $\mathcal{G}_{j+1}$ if $n+1 \geq (j+1)(s+1)$.

Therefore the relationship between n and j will be $j = \lfloor \frac{n}{s+1} \rfloor$. Since $\mathcal{G}_{j+1} \subset \mathcal{G}_j$ repeating these calculations will produce new residuals $r_{n+2}, r_{n+3}, \cdots$ in $\mathcal{G}_{j+1}$.

Choice of ω_j's: The ω_{j+1} can be chosen "freely" in calculation of first residual in $\mathcal{G}_{j+1}$ but the same value must be used in calculation of the subsequent residuals in $\mathcal{G}_{j+1}$. We choose ω_{j+1} such that $\mathbf{r}_{n+1}$ has minimum norm.

This choie will leads to $\omega_{j+1} = \frac{\mathbf{t}^*\mathbf{v}_n}{\mathbf{t}^*\mathbf{t}}$ where $\mathbf{t} = \mathbf{A}\mathbf{v}_n$

The algorithm for solving linear system will be:

- Compute first $s+1$ residuals using any (OrthoRres) method like GMRES.

IDR Algorithm

The algorithm for solving linear system will be:

- Compute first $s + 1$ residuals using any (OrthoRres) method like GMRES.
- Compute $\mathbf{c}_n = (\mathbf{P}^* \nabla \mathbf{R_{n:n-s}})^{-1} \mathbf{P}^* \mathbf{r}_n$

IDR Algorithm

The algorithm for solving linear system will be:

- Compute first $s+1$ residuals using any (OrthoRres) method like GMRES.
- Compute $\mathbf{c}_n = (\mathbf{P}^* \nabla \mathbf{R}_{\mathbf{n:n-s}})^{-1} \mathbf{P}^* \mathbf{r}_n$
- Compute $\mathbf{v}_n = \mathbf{r}_n - \nabla \mathbf{R}_{n:n-s} \mathbf{c}_n$

IDR Algorithm

The algorithm for solving linear system will be:

- Compute first $s+1$ residuals using any (OrthoRres) method like GMRES.
- Compute $\mathbf{c}_n = (\mathbf{P}^*\nabla\mathbf{R}_{\mathbf{n:n-s}})^{-1}\mathbf{P}^*\mathbf{r}_n$
- Compute $\mathbf{v}_n = \mathbf{r}_n - \nabla\mathbf{R}_{n:n-s}\mathbf{c}_n$
- Compute $\omega_{j+1} = \frac{(\mathbf{A}\mathbf{v}_n)^*\mathbf{v}_n}{||\mathbf{A}\mathbf{v}_n||_2^2}$ (if $\mathbf{r}_{n+1}$ is first residual in $\mathcal{G}_{j+1}$)

IDR Algorithm

The algorithm for solving linear system will be:

- Compute first $s+1$ residuals using any (OrthoRres) method like GMRES.
- Compute $\mathbf{c}_n = (\mathbf{P}^*\nabla\mathbf{R_{n:n-s}})^{-1}\mathbf{P}^*\mathbf{r}_n$
- Compute $\mathbf{v}_n = \mathbf{r}_n - \nabla\mathbf{R}_{n:n-s}\mathbf{c}_n$
- Compute $\omega_{j+1} = \frac{(\mathbf{A}\mathbf{v}_n)^*\mathbf{v}_n}{||\mathbf{A}\mathbf{v}_n||_2^2}$ (if $\mathbf{r}_{n+1}$ is first residual in $\mathcal{G}_{j+1}$)
- Compute $\mathbf{r}_{n+1}$

IDR Algorithm

The algorithm for solving linear system will be:

- Compute first $s+1$ residuals using any (OrthoRres) method like GMRES.
- Compute $\mathbf{c}_n = (\mathbf{P}^* \nabla \mathbf{R}_{\mathbf{n:n-s}})^{-1} \mathbf{P}^* \mathbf{r}_n$
- Compute $\mathbf{v}_n = \mathbf{r}_n - \nabla \mathbf{R}_{n:n-s} \mathbf{c}_n$
- Compute $\omega_{j+1} = \frac{(\mathbf{A}\mathbf{v}_n)^* \mathbf{v}_n}{||\mathbf{A}\mathbf{v}_n||_2^2}$ (if $\mathbf{r}_{n+1}$ is first residual in $\mathcal{G}_{j+1}$)
- Compute $\mathbf{r}_{n+1}$
- Compute $\mathbf{x}_{n+1}$

Eigenvalues based on IDR

So far we have

$$\left\{ \begin{array}{lcl} \mathbf{r}_n & = & (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} \\ \mathbf{v}_{n-1} & = & \mathbf{r}_{n-1} - \nabla \mathbf{R}_{n-1:n-s}\mathbf{c}_{n-1} \end{array} \right. \quad (2.5)$$

Where $\nabla \mathbf{R}_{n-1:n-s} = [\nabla \mathbf{r}_{n-1} \nabla \mathbf{r}_{n-2} \cdots \nabla \mathbf{r}_{n-s}]$ and $\mathbf{c}_{n-1}^T = [\gamma_1^{(n-1)} \cdots \gamma_s^{(n-1)}]$.

Eigenvalues based on IDR

So far we have

$$\begin{cases} \mathbf{r}_n & = & (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} \\ \mathbf{v}_{n-1} & = & \mathbf{r}_{n-1} - \nabla \mathbf{R}_{n-1:n-s}\mathbf{c}_{n-1} \end{cases} \tag{2.5}$$

Where $\nabla \mathbf{R}_{n-1:n-s} = [\nabla \mathbf{r}_{n-1} \nabla \mathbf{r}_{n-2} \cdots \nabla \mathbf{r}_{n-s}]$ and $\mathbf{c}_{n-1}^T = [\gamma_1^{(n-1)} \cdots \gamma_s^{(n-1)}]$.

I want to change the order of columns in $\nabla \mathbf{R}_{n-1:n-s}$ and re-write it as $\nabla \mathbf{R}_{n-s:n-1} = [\nabla \mathbf{r}_{n-s} \nabla \mathbf{r}_{n-s+1} \cdots \nabla \mathbf{r}_{n-1}]$, consequently we have to change order of γ's and re-write $\mathbf{c}_{n-1}^T := [\gamma_s^{(n-1)} \cdots \gamma_1^{(n-1)}] =$

Eigenvalues based on IDR

So far we have

$$\begin{cases} \mathbf{r}_n & = & (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} \\ \mathbf{v}_{n-1} & = & \mathbf{r}_{n-1} - \nabla \mathbf{R}_{n-1:n-s}\mathbf{c}_{n-1} \end{cases} \tag{2.5}$$

Where $\nabla \mathbf{R}_{n-1:n-s} = [\nabla \mathbf{r}_{n-1} \nabla \mathbf{r}_{n-2} \cdots \nabla \mathbf{r}_{n-s}]$ and $\mathbf{c}_{n-1}^T = [\gamma_1^{(n-1)} \cdots \gamma_s^{(n-1)}]$.

I want to change the order of columns in $\nabla \mathbf{R}_{n-1:n-s}$ and re-write it as $\nabla \mathbf{R}_{n-s:n-1} = [\nabla \mathbf{r}_{n-s} \nabla \mathbf{r}_{n-s+1} \cdots \nabla \mathbf{r}_{n-1}]$, consequently we have to change order of γ's and re-write $\mathbf{c}_{n-1}^T := [\gamma_s^{(n-1)} \cdots \gamma_1^{(n-1)}] = [\gamma_1^{(n-1)} \cdots \gamma_s^{(n-1)}]$.

Eigenvalues based on IDR

So far we have

$$\begin{cases} \mathbf{r}_n & = & (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} \\ \mathbf{v}_{n-1} & = & \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-1:n-s}\mathbf{c}_{n-1} \end{cases} \tag{2.5}$$

Where $\nabla\mathbf{R}_{n-1:n-s} = [\nabla\mathbf{r}_{n-1}\nabla\mathbf{r}_{n-2}\cdots\nabla\mathbf{r}_{n-s}]$ and $\mathbf{c}_{n-1}^T = [\gamma_1^{(n-1)}\cdots\gamma_s^{(n-1)}]$.

I want to change the order of columns in $\nabla\mathbf{R}_{n-1:n-s}$ and re-write it as $\nabla\mathbf{R}_{n-s:n-1} = [\nabla\mathbf{r}_{n-s}\nabla\mathbf{r}_{n-s+1}\cdots\nabla\mathbf{r}_{n-1}]$, consequently we have to change order of γ's and re-write $\mathbf{c}_{n-1}^T := [\gamma_s^{(n-1)}\cdots\gamma_1^{(n-1)}] = [\gamma_1^{(n-1)}\cdots\gamma_s^{(n-1)}]$.

With this cosmetic change (2.5) turns into (2.6):

$$\begin{cases} \mathbf{r}_n & = & (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1} \\ \mathbf{v}_{n-1} & = & \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1} \end{cases} \tag{2.6}$$

Eigenvalues based on IDR

$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla \mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1}$. Let's re-write $\mathbf{v}_{n-1}$ in terms of residuals rather than residual differences:

Eigenvalues based on IDR

$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1}$. Let's re-write $\mathbf{v}_{n-1}$ in terms of residuals rather than residual differences:

$$\begin{aligned}
\mathbf{v}_{n-1} &= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1} \\
&= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1} \nabla\mathbf{r}_{n-s+k}\gamma_{k+1}^{(n-1)} \\
&= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1} (r_{n-s+k} - r_{n-s+k-1})\gamma_{k+1}^{(n-1)}
\end{aligned}$$

Eigenvalues based on IDR

$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1}$. Let's re-write $\mathbf{v}_{n-1}$ in terms of residuals rather than residual differences:

$$\begin{aligned}\mathbf{v}_{n-1} &= \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1} \\ &= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1} \nabla\mathbf{r}_{n-s+k}\gamma_{k+1}^{(n-1)} \\ &= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1} (r_{n-s+k} - r_{n-s+k-1})\gamma_{k+1}^{(n-1)}\end{aligned}$$

$$= \mathbf{r}_{n-1} - \left(- \gamma_1 r_{n-s-1} + \sum_{k=1}^{s-1} r_{n-s-1+k}(\gamma_k^{(n-1)} - \gamma_{k+1}^{(n-1)}) + \gamma_s \mathbf{r}_{n-1}\right)$$

Eigenvalues based on IDR

$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1}$. Let's re-write $\mathbf{v}_{n-1}$ in terms of residuals rather than residual differences:

$$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \nabla\mathbf{R}_{n-s:n-1}\mathbf{c}_{n-1}$$

$$= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1} \nabla\mathbf{r}_{n-s+k}\gamma_{k+1}^{(n-1)}$$

$$= \mathbf{r}_{n-1} - \textstyle\sum_{k=0}^{s-1}(r_{n-s+k} - r_{n-s+k-1})\gamma_{k+1}^{(n-1)}$$

$$= \mathbf{r}_{n-1} - \Big(-\gamma_1 r_{n-s-1} + \sum_{k=1}^{s-1} r_{n-s-1+k}(\gamma_k^{(n-1)} - \gamma_{k+1}^{(n-1)}) + \gamma_s \mathbf{r}_{n-1}\Big)$$

$$= \mathbf{r}_{n-1} - \left(R_{n-s-1:n-1}\begin{bmatrix} -\gamma_1^{(n-1)} \\ \gamma_1^{(n-1)} - \gamma_2^{(n-1)} \\ \vdots \\ \gamma_s^{(n-1)} \end{bmatrix}\right)$$

Eigenvalues based on IDR

So far we got,

$$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \left(\mathbf{R}_{n-s-1:n-1} \begin{bmatrix} -\gamma_1^{(n-1)} \\ \gamma_1^{(n-1)} - \gamma_2^{(n-1)} \\ \vdots \\ \gamma_s^{(n-1)} \end{bmatrix}\right)$$

$$= \mathbf{r}_{n-1} + \left(\mathbf{R}_{n-s-1:n-1} \begin{bmatrix} \gamma_1^{(n-1)} \\ \gamma_2^{(n-1)} - \gamma_1^{(n-1)} \\ \vdots \\ -\gamma_s^{(n-1)} \end{bmatrix}\right)$$

So far we got,

$$\mathbf{v}_{n-1} = \mathbf{r}_{n-1} - \left(\mathbf{R}_{n-s-1:n-1} \begin{bmatrix} -\gamma_1^{(n-1)} \\ \gamma_1^{(n-1)} - \gamma_2^{(n-1)} \\ \vdots \\ \gamma_s^{(n-1)} \end{bmatrix}\right)$$

$$= \mathbf{r}_{n-1} + \left(\mathbf{R}_{n-s-1:n-1} \begin{bmatrix} \gamma_1^{(n-1)} \\ \gamma_2^{(n-1)} - \gamma_1^{(n-1)} \\ \vdots \\ -\gamma_s^{(n-1)} \end{bmatrix}\right)$$

$$= \mathbf{R}_{n-s-1:n-1} \begin{bmatrix} \gamma_1^{(n-1)} \\ \gamma_2^{(n-1)} - \gamma_1^{(n-1)} \\ \vdots \\ 1 - \gamma_s^{(n-1)} \end{bmatrix} := \mathbf{R}_{n-s-1:n-1} \ \text{diff} \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix}$$

Eigenvalues based on IDR

On the last page we derived:

$$\mathbf{v}_{n-1} = \mathbf{R}_{n-s-1:n-1} diff \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix}$$

Eigenvalues based on IDR

On the last page we derived:

$$\mathbf{v}_{n-1} = \mathbf{R}_{n-s-1:n-1} diff \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix}$$

Moreover,

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1}.$$

Therefore,

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1} diff \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix} := (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \tag{2.7}$$

Eigenvalues based on IDR

On the last page we derived:

$$\mathbf{v}_{n-1} = \mathbf{R}_{n-s-1:n-1} diff \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix}$$

Moreover,

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{v}_{n-1}.$$

Therefore,

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1} diff \begin{bmatrix} 0 \\ \mathbf{c}_{n-1} \\ 1 \end{bmatrix} := (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \tag{2.7}$$

Equation (2.7) brings us the following theorem:

Eigenvalues based on IDR

THM: The generalized Hessenberg decomposition for IDR(s) is given by

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_w^{(n)} = \mathbf{R}_{n+1}\underline{\mathbf{Y}}_n^o$$

where $\mathbf{R}_{n+1} = [\mathbf{r}_0 \cdots \mathbf{r}_n]$. For $s < k \leq n$ the k^{th} column of upper triangular matrix $\mathbf{Y}_n \in \mathbb{C}^{n\times n}$ and of the extended Hessenberg matrix $\underline{\mathbf{Y}}_n^o \in \mathbb{C}^{(n+1)\times n}$ are defined by:

$$\mathbf{Y}_n\mathbf{e}_k = \begin{bmatrix} \mathbf{o}_{k-(s+1)} \\ \mathbf{y}_k \\ \mathbf{o}_{n-k} \end{bmatrix}, \underline{\mathbf{Y}}_n^o\mathbf{e}_k = \begin{bmatrix} \mathbf{o}_{k-(s+1)} \\ \mathbf{y}_k \\ -1 \\ \mathbf{o}_{n-k} \end{bmatrix}$$

where $\mathbf{y}_k = \begin{bmatrix} \gamma_1^{(k)} \\ \gamma_2^{(k)} - \gamma_1^{(k)} \\ \vdots \\ 1 - \gamma_s^{(k)} \end{bmatrix}$ and diagonal matrix $\mathbf{D}_\omega^{(n)}$ is given by $\mathbf{e}_k^T\mathbf{D}_\omega^{(n)}\mathbf{e}_k = \omega_j$, $j = \lfloor \frac{k}{s+1} \rfloor$.

Eigenvalues based on IDR

The leading portions of matrices $\mathbf{Y}_n, \underline{\mathbf{Y}}_n^o$ and $\mathbf{D}_o mega^{(n)}$ are given by Hessenberg decomposition of starting procedure.

Eigenvalues based on IDR

The leading portions of matrices $\mathbf{Y}_n, \underline{\mathbf{Y}}_n^o$ and $\mathbf{D}_omega^{(n)}$ are given by Hessenberg decomposition of starting procedure.

Proof: We sort the terms in equation (2.7) according to occurance of the matrix $\mathbf{A}$ and obtain:

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \tag{2.7}$$

$$\omega_j \mathbf{A}\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n = \mathbf{R}_{n-s-1:n-1} - \mathbf{r}_n = \mathbf{R}_{n-s-1:n} \begin{bmatrix} \mathbf{y}_n \\ -1 \end{bmatrix} \tag{2.8}$$

Which is the n^{th} column:

Eigenvalues based on IDR

The leading portions of matrices $\mathbf{Y}_n, \underline{\mathbf{Y}}_n^o$ and $\mathbf{D}_o mega^{(n)}$ are given by Hessenberg decomposition of starting procedure.

Proof: We sort the terms in equation (2.7) according to occurance of the matrix $\mathbf{A}$ and obtain:

$$\mathbf{r_n} = (\mathbf{I} - \omega_j \mathbf{A})\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n \tag{2.7}$$

$$\omega_j \mathbf{A}\mathbf{R}_{n-s-1:n-1}\mathbf{y}_n = \mathbf{R}_{n-s-1:n-1} - \mathbf{r}_n = \mathbf{R}_{n-s-1:n} \begin{bmatrix} \mathbf{y}_n \\ -1 \end{bmatrix} \tag{2.8}$$

Which is the n^{th} column:

$$\omega_j \mathbf{A}\mathbf{R}_n \begin{bmatrix} \mathbf{o}_{n-(s+1)} \\ \mathbf{y}_n \end{bmatrix} = \mathbf{R}_{n+1} \begin{bmatrix} \mathbf{o}_{n-(s+1)} \\ \mathbf{y}_n \\ -1 \end{bmatrix}$$

Restart of IDR

We know convergence of all these methods depends on how good our initial guess is. We do n iterations then we can use current information to start the procedure with a better initial vector.

Restart of IDR

We know convergence of all these methods depends on how good our initial guess is. We do n iterations then we can use current information to start the procedure with a better initial vector.

Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ Using IDR(s)Eig after n steps we get:

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega^{(n)} = \mathbf{R}_n\underline{\mathbf{Y}}_n^0 - \mathbf{r}_{n+1}\mathbf{e}_n^T$$

Let's call $\mathbf{Y}_n\mathbf{D}_\omega^{(n)}$ just $\mathbf{Y}_n$ for simplicity.

Restart of IDR

We know convergence of all these methods depends on how good our initial guess is. We do n iterations then we can use current information to start the procedure with a better initial vector.

Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ Using IDR(s)Eig after n steps we get:

$$\mathbf{A}\mathbf{R}_n\mathbf{Y}_n\mathbf{D}_\omega^{(n)} = \mathbf{R}_n\underline{\mathbf{Y}}_n^0 - \mathbf{r}_{n+1}\mathbf{e}_n^T$$

Let's call $\mathbf{Y}_n\mathbf{D}_\omega^{(n)}$ just $\mathbf{Y}_n$ for simplicity.
$\mathbf{Y}_n$ and $\underline{\mathbf{Y}}_n^0$ have some structure like the structure given below for $n = 7$ and $s = 3$:

$$Y_n = \begin{bmatrix} * & * & * & * & 0 & 0 & 0 \\ 0 & * & * & * & * & 0 & 0 \\ 0 & 0 & * & * & * & * & 0 \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}, \quad Y_n^0 = \begin{bmatrix} * & * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & 0 & 0 \\ 0 & * & * & * & * & * & 0 \\ 0 & 0 & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix}.$$

Restart of IDR

We can transfer the pencil $(\mathbf{Y}_n^0, \mathbf{Y}_n)$ to $(\mathbf{T}_n, \mathbf{S}_n)$ where $\mathbf{T}_n$ and $\mathbf{S}_n$ are (quasi) triangular while defending the structure using LZ algorithm.

Restart of IDR

We can transfer the pencil $(\mathbf{Y}_n^0, \mathbf{Y}_n)$ to $(\mathbf{T}_n, \mathbf{S}_n)$ where $\mathbf{T}_n$ and $\mathbf{S}_n$ are (quasi) triangular while defending the structure using LZ algorithm. Then we will have the following equations:

$$\mathbf{T}_n = \mathbf{L}\mathbf{Y}_n^0\mathbf{Z}, \quad \mathbf{S}_n = \mathbf{L}\mathbf{Y}_n\mathbf{Z} \tag{3.1}$$

Therefore, we have:

$$\mathbf{A}\mathbf{R}_n\mathbf{L}^{-1}\mathbf{S}_n\mathbf{Z}^{-1} = \mathbf{R}_n\mathbf{L}^{-1}\mathbf{T}_n\mathbf{Z}^{-1} - \mathbf{r}_{n+1}\mathbf{e}_n^T$$

Restart of IDR

We can transfer the pencil $(\mathbf{Y}_n^0, \mathbf{Y}_n)$ to $(\mathbf{T}_n, \mathbf{S}_n)$ where $\mathbf{T}_n$ and $\mathbf{S}_n$ are (quasi) triangular while defending the structure using LZ algorithm. Then we will have the following equations:

$$\mathbf{T}_n = \mathbf{L}\mathbf{Y}_n^0\mathbf{Z}, \quad \mathbf{S}_n = \mathbf{L}\mathbf{Y}_n\mathbf{Z} \tag{3.1}$$

Therefore, we have:

$$\mathbf{A}\mathbf{R}_n\mathbf{L}^{-1}\mathbf{S}_n\mathbf{Z}^{-1} = \mathbf{R}_n\mathbf{L}^{-1}\mathbf{T}_n\mathbf{Z}^{-1} - \mathbf{r}_{n+1}\mathbf{e}_n^T$$

Letting $\mathbf{R}_n\mathbf{L}^{-1} = \hat{\mathbf{R}}_n$ and applying $\mathbf{Z}$ on right we get:

$$\mathbf{A}\hat{\mathbf{R}}_n\mathbf{S}_n = \hat{\mathbf{R}}_n\mathbf{T}_n - \mathbf{r}_{n+1}\mathbf{z}^T$$

where $\mathbf{z}^T$ is the last row of $\mathbf{Z}$.

Restart of IDR

By discarding the last $k = n - m$ columns of each side we get:

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m = \hat{\mathbf{R}}_m\mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Where $\hat{\mathbf{R}}_m$ consist of first m columns of $\hat{\mathbf{R}}_n$, $\mathbf{S}_m$ and $\mathbf{T}_m$ are $m \times m$ leading submatrix of $\mathbf{S}_n$ and $\mathbf{T}_n$ respectively and $\mathbf{z}_0^T$ consist of first m components of $\mathbf{z}^T$.

Restart of IDR

By discarding the last $k = n - m$ columns of each side we get:

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m = \hat{\mathbf{R}}_m\mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Where $\hat{\mathbf{R}}_m$ consist of first m columns of $\hat{\mathbf{R}}_n$, $\mathbf{S}_m$ and $\mathbf{T}_m$ are $m \times m$ leading submatrix of $\mathbf{S}_n$ and $\mathbf{T}_n$ respectively and $\mathbf{z}_0^T$ consist of first m components of $\mathbf{z}^T$.

Now, we have to turn this equation to a *generalized Hessenberg decomposition.* We start doing that by creating zero at the first component of $\mathbf{z}_0^T$.

Restart of IDR

By discarding the last $k = n - m$ columns of each side we get:

$$\mathbf{A}\hat{\mathbf{R}}_m \mathbf{S}_m = \hat{\mathbf{R}}_m \mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Where $\hat{\mathbf{R}}_m$ consist of first m columns of $\hat{\mathbf{R}}_n$, $\mathbf{S}_m$ and $\mathbf{T}_m$ are $m \times m$ leading submatrix of $\mathbf{S}_n$ and $\mathbf{T}_n$ respectively and $\mathbf{z}_0^T$ consist of first m components of $\mathbf{z}^T$.
Now, we have to turn this equation to a *generalized Hessenberg decomposition.* We start doing that by creating zero at the first component of $\mathbf{z}_0^T$.

There will be different number of elimination matrices at each step chasing the bulge both up and down, so, to avoid having tons of notifications I will use "hat" for showing there is a bulge in $S_m^{(i)}$ and $T_m^{(i)}$. "hat" will be used on L_i, R_i and also there will be "hat" on $\hat{R}_m^{(i)}$ which is not indicating existence of bulge.

Restart of IDR

We start by the following equation:

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m = \hat{\mathbf{R}}_m\mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Restart of IDR

We start by the following equation:

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m = \hat{\mathbf{R}}_m\mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Applying $\mathbf{G}_1$ to create zero at the first place of $\mathbf{z}_0^T$ on right that acts on first and second columns, it adds proper multiple of second column to first column so that we defend the zero structure of upper-triangular parts of $\mathbf{S}_m$ and $\mathbf{T}_m$, we have

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m\mathbf{G}_1 = \hat{\mathbf{R}}_m\mathbf{T}_m\mathbf{G}_1 - \mathbf{r}_{n+1}\mathbf{z}_1^T.$$

This equation can be re-written as:

Restart of IDR

We start by the following equation:

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m = \hat{\mathbf{R}}_m\mathbf{T}_m - \mathbf{r}_{n+1}\mathbf{z}_0^T$$

Applying $\mathbf{G}_1$ to create zero at the first place of $\mathbf{z}_0^T$ on right that acts on first and second columns, it adds proper multiple of second column to first column so that we defend the zero structure of upper-triangular parts of $\mathbf{S}_m$ and $\mathbf{T}_m$, we have

$$\mathbf{A}\hat{\mathbf{R}}_m\mathbf{S}_m\mathbf{G}_1 = \hat{\mathbf{R}}_m\mathbf{T}_m\mathbf{G}_1 - \mathbf{r}_{n+1}\mathbf{z}_1^T.$$

This equation can be re-written as:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(1)}\hat{\mathbf{S}}_m^{(1)} = \hat{\mathbf{R}}_m^{(1)}\mathbf{T}_m^{(1)} - \mathbf{r}_{n+1}\mathbf{z}_1^T \tag{3.2}$$

(Note that I did not put hat on T_m because there is no bulge in T_mG_1 !)

Restart of IDR

The followings are evident: $\mathbf{z}_0^T \mathbf{G}_1 = [0 * * \cdots *] = \mathbf{z}_1^T$

$$\mathbf{T}_m^{(1)} = \begin{bmatrix} t & t & t & t & 0 & 0 & 0 \\ + & t & t & t & t & 0 & 0 \\ 0 & 0 & t & t & t & t & 0 \\ 0 & 0 & 0 & t & t & t & t \\ 0 & 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & 0 & t \end{bmatrix}, \quad \hat{\mathbf{S}}_m^{(1)} = \begin{bmatrix} s & s & s & s & 0 & 0 & 0 \\ \oplus & s & s & s & s & 0 & 0 \\ 0 & 0 & s & s & s & s & 0 \\ 0 & 0 & 0 & s & s & s & s \\ 0 & 0 & 0 & 0 & s & s & s \\ 0 & 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & s \end{bmatrix}$$

Restart of IDR

The followings are evident: $\mathbf{z}_0^T \mathbf{G}_1 = [0 * * \cdots *] = \mathbf{z}_1^T$

$$\mathbf{T}_m^{(1)} = \begin{bmatrix} t & t & t & t & 0 & 0 & 0 \\ + & t & t & t & t & 0 & 0 \\ 0 & 0 & t & t & t & t & 0 \\ 0 & 0 & 0 & t & t & t & t \\ 0 & 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & 0 & t \end{bmatrix}, \quad \hat{\mathbf{S}}_m^{(1)} = \begin{bmatrix} s & s & s & s & 0 & 0 & 0 \\ \oplus & s & s & s & s & 0 & 0 \\ 0 & 0 & s & s & s & s & 0 \\ 0 & 0 & 0 & s & s & s & s \\ 0 & 0 & 0 & 0 & s & s & s \\ 0 & 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & s \end{bmatrix}$$

"$\oplus$" is a bulge that needs to be eliminated. This can be done by elimination matrix $\mathbf{L}_1$ applied on left to both $\hat{\mathbf{S}}_m^{(1)}$ and $\mathbf{T}_m^{(1)}$ adding proper multiple of row 1 to row 2 turning $\hat{\mathbf{S}}_m^{(1)}$ to upper-triangular form. This will not disturb anything in $\mathbf{T}_m^{(1)}$.

Restart of IDR

Therefore, we will have:

$$\mathbf{L}_1 \hat{\mathbf{S}}_m^{(1)} := \mathbf{S}_m^{(1)}, \quad \mathbf{L}_1 \mathbf{T}_m^{(1)} := \mathbf{T}_m^{(1)},$$

Restart of IDR

Therefore, we will have:

$$\mathbf{L}_1\hat{\mathbf{S}}_m^{(1)} := \mathbf{S}_m^{(1)}, \quad \mathbf{L}_1\mathbf{T}_m^{(1)} := \mathbf{T}_m^{(1)},$$

or equivalently

$$\hat{\mathbf{S}}_m^{(1)} = \mathbf{L}_1^{-1}\mathbf{S}_m^{(1)}, \quad \mathbf{T}_m^{(1)} = \mathbf{L}_1^{-1}\mathbf{T}_m^{(1)}$$

Restart of IDR

Therefore, we will have:

$$\mathbf{L}_1 \hat{\mathbf{S}}_m^{(1)} := \mathbf{S}_m^{(1)}, \quad \mathbf{L}_1 \mathbf{T}_m^{(1)} := \mathbf{T}_m^{(1)},$$

or equivalently

$$\hat{\mathbf{S}}_m^{(1)} = \mathbf{L}_1^{-1} \mathbf{S}_m^{(1)}, \quad \mathbf{T}_m^{(1)} = \mathbf{L}_1^{-1} \mathbf{T}_m^{(1)}$$

By substituting this in equation (3.2) we have:

$$\mathbf{A} \hat{\mathbf{R}}_m^{(1)} \mathbf{L}_1^{-1} \mathbf{S}_m^{(1)} = \hat{\mathbf{R}}_m^{(1)} \mathbf{L}_1^{-1} \mathbf{T}_m^{(1)} - \mathbf{r}_{n+1} \mathbf{z}_1^T$$

By letting $\hat{\mathbf{R}}_m^{(1)} \mathbf{L}_1^{-1} = \hat{\mathbf{R}}_m^{(2)}$ we get

$$\mathbf{A} \hat{\mathbf{R}}_m^{(2)} \mathbf{S}_m^{(1)} = \hat{\mathbf{R}}_m^{(2)} \mathbf{T}_m^{(1)} - \mathbf{r}_{n+1} \mathbf{z}_1^T \qquad (3.3)$$

Restart of IDR

Next step will create zero in second position of $\mathbf{z}_1^T$ by $\mathbf{G}_2$ on right. Applying this transformation we have:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(2)}\mathbf{S}_m^{(1)}\mathbf{G}_2 = \hat{\mathbf{R}}_m^{(2)}\mathbf{T}_m^{(1)}\mathbf{G}_2 - \mathbf{r}_{n+1}\mathbf{z}_2^T.$$

Where $\mathbf{z}_2^T := \mathbf{z}_1^T\mathbf{G}_2 = [0\ 0 * * \cdots *]$.

Restart of IDR

Next step will create zero in second position of $\mathbf{z}_1^T$ by $\mathbf{G}_2$ on right. Applying this transformation we have:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(2)}\mathbf{S}_m^{(1)}\mathbf{G}_2 = \hat{\mathbf{R}}_m^{(2)}\mathbf{T}_m^{(1)}\mathbf{G}_2 - \mathbf{r}_{n+1}\mathbf{z}_2^T.$$

Where $\mathbf{z}_2^T := \mathbf{z}_1^T\mathbf{G}_2 = [0\ 0 * * \cdots *]$.

Let $\hat{\mathbf{S}}_m^{(2)} := \mathbf{S}_m^{(1)}\mathbf{G}_2$, $\mathbf{T}_m^{(2)} := \mathbf{T}_m^{(1)}\mathbf{G}_2$. By this notation we have:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(2)}\hat{\mathbf{S}}_m^{(2)} = \hat{\mathbf{R}}_m^{(2)}\mathbf{T}_m^{(2)} - \mathbf{r}_{n+1}\mathbf{z}_2^T. \quad (3.4)$$

$\hat{\mathbf{S}}_m^{(2)}$ and $\mathbf{T}_m^{(2)}$ have the following form:

$$\mathbf{T}_m^{(2)} = \begin{bmatrix} t & t & t & t & 0 & 0 & 0 \\ + & t & t & t & t & 0 & 0 \\ 0 & + & t & t & t & t & 0 \\ 0 & 0 & 0 & t & t & t & t \\ 0 & 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & 0 & t \end{bmatrix}, \quad \hat{\mathbf{S}}_m^{(2)} = \begin{bmatrix} s & s & s & s & 0 & 0 & 0 \\ 0 & s & s & s & s & 0 & 0 \\ 0 & \oplus & s & s & s & s & 0 \\ 0 & 0 & 0 & s & s & s & s \\ 0 & 0 & 0 & 0 & s & s & s \\ 0 & 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & s \end{bmatrix}$$

Restart of IDR

"$\oplus$" has to be eliminated by a Gauss transform on left, L_2, adding proper multiple of row 2 to row 3. This will create a bulge in (3,1) position of $T_m^{(2)}$ disturbing the Hessenberg structure. So far we have:

$$\hat{\mathbf{T}}_m^{(2)} := \mathbf{L}_2 \mathbf{T}_m^{(2)}, \quad \mathbf{S}_m^{(2)} := \mathbf{L}_2 \hat{\mathbf{S}}_m^{(2)}$$

OR equivalently:

$$\mathbf{T}_m^{(2)} = \mathbf{L}_2^{-1} \hat{\mathbf{T}}_m^{(2)}, \quad \hat{\mathbf{S}}_m^{(2)} := \mathbf{L}_2^{-1} \mathbf{S}_m^{(2)}$$

Restart of IDR

"$\oplus$" has to be eliminated by a Gauss transform on left, L_2, adding proper multiple of row 2 to row 3. This will create a bulge in (3,1) position of $T_m^{(2)}$ disturbing the Hessenberg structure. So far we have:

$$\hat{\mathbf{T}}_m^{(2)} := \mathbf{L}_2 \mathbf{T}_m^{(2)}, \quad \mathbf{S}_m^{(2)} := \mathbf{L}_2 \hat{\mathbf{S}}_m^{(2)}$$

OR equivalently:

$$\mathbf{T}_m^{(2)} = \mathbf{L}_2^{-1} \hat{\mathbf{T}}_m^{(2)}, \quad \hat{\mathbf{S}}_m^{(2)} := \mathbf{L}_2^{-1} \mathbf{S}_m^{(2)}$$

$\hat{\mathbf{T}}_m^{(2)}$ and $\mathbf{S}_m^{(2)}$ have following form:

$$\hat{\mathbf{T}}_m^{(2)} = \begin{bmatrix} t & t & t & t & 0 & 0 & 0 \\ + & t & t & t & t & 0 & 0 \\ \oplus & + & t & t & t & t & 0 \\ 0 & 0 & 0 & t & t & t & t \\ 0 & 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & 0 & t \end{bmatrix}, \quad \mathbf{S}_m^{(2)} = \begin{bmatrix} s & s & s & s & 0 & 0 & 0 \\ 0 & s & s & s & s & 0 & 0 \\ 0 & 0 & s & s & s & s & 0 \\ 0 & 0 & 0 & s & s & s & s \\ 0 & 0 & 0 & 0 & s & s & s \\ 0 & 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & s \end{bmatrix}$$

Restart of IDR

As of now, we have:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-1}\mathbf{S}_m^{(2)} = \hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-1}\hat{\mathbf{T}}_m^{(2)} - \mathbf{r}_{n+1}\mathbf{z}_2^T$$

Letting $\hat{\mathbf{R}}_m^{(3)} := \hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-2}$ we get:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(3)}\mathbf{S}_m^{(2)} = \hat{\mathbf{R}}_m^{(3)}\hat{\mathbf{T}}_m^{(2)} - \mathbf{r}_{n+1}\mathbf{z}_2^T \qquad (3.5)$$

Restart of IDR

As of now, we have:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-1}\mathbf{S}_m^{(2)} = \hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-1}\hat{\mathbf{T}}_m^{(2)} - \mathbf{r}_{n+1}\mathbf{z}_2^T$$

Letting $\hat{\mathbf{R}}_m^{(3)} := \hat{\mathbf{R}}_m^{(2)}\mathbf{L}_2^{-2}$ we get:

$$\mathbf{A}\hat{\mathbf{R}}_m^{(3)}\mathbf{S}_m^{(2)} = \hat{\mathbf{R}}_m^{(3)}\hat{\mathbf{T}}_m^{(2)} - \mathbf{r}_{n+1}\mathbf{z}_2^T \qquad (3.5)$$

The bulge in $\hat{\mathbf{T}}_m^{(2)}$ has to be eliminated by a Gauss transform on right, let's call it $\mathbf{R}_2$, adding proper multiple of column 2 to column 1. But, this transformation will create a bulge in $\mathbf{S}_m^{(2)}$.

Restart of IDR

$$\hat{\mathbf{T}}_m^{(2)}\mathbf{R}_2 = \begin{bmatrix} t & t & t & t & 0 & 0 & 0 \\ + & t & t & t & t & 0 & 0 \\ 0 & + & t & t & t & t & 0 \\ 0 & 0 & 0 & t & t & t & t \\ 0 & 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & 0 & t \end{bmatrix},$$

$$\mathbf{S}_m^{(2)}\mathbf{R}_2 = \begin{bmatrix} s & s & s & s & 0 & 0 & 0 \\ \oplus & s & s & s & s & 0 & 0 \\ 0 & 0 & s & s & s & s & 0 \\ 0 & 0 & 0 & s & s & s & s \\ 0 & 0 & 0 & 0 & s & s & s \\ 0 & 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & s \end{bmatrix}.$$

Continuing this procedure after $m-1$ steps we end up with $z_{m-1}^T = \alpha e_m^T$ and following form:

$$AR_mS_m = R_mT_m - r_{n+1}\alpha e_m^T$$

Where R_m is $N-by-m$ matrix of m residuals, S_m will be upper-triangular, T_m will be upper-Hessenberg.