```
CGLS
Select $\mathbf{x}_0 \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{u} = \mathbf{0}$, $\rho = 1$
while $\|\mathbf{r}\|_2 > tol$ do
   $\mathbf{s} = \mathbf{A}^*\mathbf{r}$
   $\rho' = \rho$, $\rho = \mathbf{s}^*\mathbf{s}$, $\beta = -\rho/\rho'$
   $\mathbf{u} \leftarrow \mathbf{s} - \beta\,\mathbf{u}$, $\mathbf{c} = \mathbf{A}\mathbf{u}$
   $\sigma = \mathbf{c}^*\mathbf{c}$, $\alpha = \rho/\sigma$
   $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
   $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while
```

```
GRAIG'S METHOD
Select $\mathbf{x}_0 \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{u} = \mathbf{0}$, $\rho = 1$
while $\|\mathbf{r}\|_2 > tol$ do
   $\rho' = \rho$, $\rho = \mathbf{r}^*\mathbf{r}$, $\beta = -\rho/\rho'$
   $\mathbf{c} = \mathbf{A}^*\mathbf{r}$
   $\mathbf{u} \leftarrow \mathbf{c} - \beta\,\mathbf{u}$, $\mathbf{c} = \mathbf{A}\mathbf{u}$
   $\sigma = \mathbf{u}^*\mathbf{u}$, $\alpha = \rho/\sigma$
   $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$
   $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while
```

ALGORITHM 8.1. CGLS (left) and Graig's method (right) for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* for a general non-singular matrix $\mathbf{A}$. Both methods apply CG to a 'squared' system: CG applied to $\mathbf{A}\mathbf{A}^*\mathbf{y} = \mathbf{b}$ leads to Graig's method; CG applied to $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ leads to CGLS.

March 26, 2018

Lecture 8 – Fast Iterative Methods for linear systems of equations

Let $\mathbf{A}$ be a non-singular $n \times n$ matrix.
We are interested in methods for numerically solving

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

$\mathbf{b}$ and $\mathbf{x}$ are n-vector. $\mathbf{A}$ and $\mathbf{b}$ are available. We have to solve for $\mathbf{x}$.

A Variants of CG for non-Hermitian systems

In this lecture, we are interested in iterative methods that rely on short recurrence relations, that is, methods that use a limited (fixed) number of AXPYs and DOTs to extend the Krylov search subspace by one dimension (preferable, using one MV only).

The simplest method of this type is obtained by applying CG to the (normal) equations $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ (CGLS, see Exercise 8.2) or to $\mathbf{A}\mathbf{A}^*\mathbf{y} = \mathbf{b}$ & $\mathbf{x} = \mathbf{A}^*\mathbf{y}$ (Graig's method, see Exercise 8.1). These methods are not widely used for solving *square* systems $\mathbf{A}\mathbf{x} = \mathbf{b}$, since, if for instance $\mathbf{A}$ is Hermitian, these methods only explore Krylov subspace generated by $\mathbf{A}^2$ (or, equivalently, residual polynomials of even degree). However, they are useful for non-square systems, as we will learn in the next lecture.

For CG, we need an Hermitian system. The 'normal equations' are a simple way of forming such a system from any general system. Bi-CG exploits yet another way. It extends to system to a 2×2 block system, see Exercise 8.3.

Property 8.1 *Mathematical properties:*

• *Graig's method:*	$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{r}_0)$	*such that* $\|\mathbf{x} - \mathbf{x}_k\|_2$ *minimal*
• *CGLS:*	$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{r}_0)$	*such that* $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2$ *minimal*
• *Bi-CG:*	$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$	*such that* $\mathbf{b} - \mathbf{A}\mathbf{x}_k \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$

Computational properties per step:
All methods require one MV by $\mathbf{A}$ *and one by* $\mathbf{A}^*$ *plus a few AXPYs and a few DOTs*

```
Bi-CG
Select $\mathbf{x}_0, \widetilde{\mathbf{r}} \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{u} = \mathbf{0}$, $\rho = 1$,        $\widetilde{\mathbf{u}} = \mathbf{0}$,
while $\|\mathbf{r}\|_2 > tol$ do
   $\rho' = \rho$, $\rho = \widetilde{\mathbf{r}}^*\mathbf{r}$, $\beta = -\rho/\rho'$
   $\mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$,        $\widetilde{\mathbf{u}} \leftarrow \widetilde{\mathbf{r}} - \overline{\beta}\,\widetilde{\mathbf{u}}$
   $\mathbf{c} = \mathbf{A}\mathbf{u}$,                $\widetilde{\mathbf{c}} = \mathbf{A}^*\widetilde{\mathbf{u}}$
   $\sigma = \widetilde{\mathbf{r}}^*\mathbf{c}$, $\alpha = \rho/\sigma$
   $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$,        $\widetilde{\mathbf{r}} \leftarrow \widetilde{\mathbf{r}} - \overline{\alpha}\,\widetilde{\mathbf{c}}$
   $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
end while
```

ALGORITHM 8.2. Bi-CG for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* for a general non-singular matrix $\mathbf{A}$.

Exercise 8.1. **Graig's method.** **Graig's method** is obtained by applying CG to the 'smallest norm solution' equations

$$\mathbf{A}\mathbf{A}^*\mathbf{y} = \mathbf{b} \quad \& \quad \mathbf{x} = \mathbf{A}^*\mathbf{y}.$$

Note that the solution is the smallest norm solution in case $\mathbf{A}$ is a full (row) rank $n \times k$ matrix with $n < k$.

(a) Since we are interested in $\mathbf{x}_k \equiv \mathbf{A}^*\mathbf{y}_k$ rather than in the iterates $\mathbf{y}_k$, we want to avoid the computation of $\mathbf{y}_k$ (and of the associated update vectors). Show that this leads to ALG. 8.1

(b) Describe the Krylov subspace that is searched for the approximate solution by Graig's method.

(c) Show that Graig's method minimises the error in the 2-norm rather than the residual.[1]

Exercise 8.2. **CGLS.** CGLS is obtained by applying CG to the normal equations

$$\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}.$$

Note that these equations lead to the minimal residual solution (least square) in case $\mathbf{A}$ is a full (column) rank matrix of size $n \times k$ with $k < n$.

The resulting algorithm is rearranged to make both the residual $\mathbf{r}_k \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_k$ from the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ available as well as the residual $\mathbf{s}_k \equiv \mathbf{A}^*\mathbf{b} - \mathbf{A}^*\mathbf{A}\mathbf{x}_k = \mathbf{A}^*\mathbf{r}_k$ from the normal equation. As a result $\sigma = \mathbf{u}^*(\mathbf{A}^*\mathbf{A}\mathbf{u})$ is computed as $\sigma = (\mathbf{A}\mathbf{u})^*(\mathbf{A}\mathbf{u})$. As a side effect, σ is computed more accurate.

(a) Derive (from CG) the CGLS algorithm of ALG. 8.1. Note that the vector $\mathbf{s}$ can stored on the same location as $\mathbf{c}$.

(b) Describe the Krylov subspace that is searched for the approximate solution by CGLS.

(c) Show that CGLS minimises the residual $\mathbf{b} - \mathbf{A}\mathbf{x}_k$ in the 2-norm.[1]

(d) The quantity σ can be computed as $\sigma = (\mathbf{A}\mathbf{u})^*(\mathbf{A}\mathbf{u})$ but also as $\sigma = \mathbf{u}^*(\mathbf{A}^*\mathbf{A}\mathbf{u})$. Explain why the first approach is more accurate. Discuss other advantages of the first approach.

[1] Recall that CG minimises residuals in the $\mathbf{A}^{-1}$-norm in case $\mathbf{A}$ is positive definite.

Exercise 8.3. **Bi-CG as CG.** Select an n-vector $\widetilde{\mathbf{b}}$.

(a) Consider the extended system

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \widetilde{\mathbf{b}} \end{bmatrix}.$$

Relate this system to the equation of interest $\mathbf{A}\mathbf{x} = \mathbf{b}$.

The system $\mathbf{A}^*\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$ is called the **shadow system** or **dual system**. Note that the matrix in the extended system is Hermitian. Therefore, CG can be applied. Split the resulting recurrence relations into relations for the first (block) coordinate and the second (block). Give the Krylov subspace that is searched for an approximate solution of $\mathbf{x}$ with the resulting method.

Consider the matrix $\mathbf{J}$ and the extended system

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^* \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \widetilde{\mathbf{b}} \end{bmatrix} \quad \text{and} \quad \mathbf{J} \equiv \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$$

(b) Relate this system to the equation of interest $\mathbf{A}\mathbf{x} = \mathbf{b}$.

The matrix $\mathbf{J}$ is symmetric, but not definite. The assignment $(\widehat{\mathbf{x}}, \widehat{\mathbf{y}}) \rightsquigarrow \widehat{\mathbf{y}}^*\mathbf{J}\widehat{\mathbf{x}}$ defines a "semi inner-product" (i.e., an inner-product except for the positive condition '$\widehat{\mathbf{x}}^*\mathbf{J}\widehat{\mathbf{x}} \geq 0$ and $\widehat{\mathbf{x}}^*\mathbf{J}\widehat{\mathbf{x}} = 0$ iff $\widehat{\mathbf{x}} = \mathbf{0}$').

(c) Show that the matrix in the system is Hermitian with respect to this semi inner product.

(d) Apply CG to this system using the semi inner product instead of the standard inner product, that is, replace $\mathbf{r}^*\mathbf{r}$ by $\widehat{\mathbf{r}}^*\mathbf{J}\widehat{\mathbf{r}}$, etc.. Split the resulting recurrence relations into relations for the first (block) coordinate and the second (block) to obtain **Bi-CG** as in Alg. 8.2. Note that the lines that exclusively are for the computation of $\widetilde{\mathbf{x}}$ have been omitted.

(e) Prove that the residual $\mathbf{r}_k$ as produced in the kth step of Bi-CG (that is, the residual of the first block coordinate) is orthogonal to $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$. Here, $\widetilde{\mathbf{r}}_0$ $(= \widetilde{\mathbf{b}} - \mathbf{A}^*\widetilde{\mathbf{x}}_0)$ is the so-called **initial shadow residual**, that is, the initial residual of the shadow system. Note that the initial guess $\widetilde{\mathbf{x}}_0$ for the shadow system is irrelevant.

In CG (for Hermitian matrices), residuals are mutually orthogonal, while, here in Bi-CG (for general matrices), residuals are orthogonal to a different set of residuals, the shadow residuals. It is said that the residuals satisfy a **bi-orthogonal**ity relation, which is reflected in the naming of the method.

Exercise 8.4. **Bi-CG and the initial shadow residual.**

(a) Show that Bi-CG is equivalent to CG in case $\mathbf{A}$ is Hermitian and $\widetilde{\mathbf{r}}_0 = \mathbf{r}_0$.

(b) Show that Bi-CG minimises the 2-norm of the residual (rather than the A^{-1}-norm) in case $\mathbf{A}$ is Hermitian and $\widetilde{\mathbf{r}}_0 = \mathbf{A}^*\mathbf{r}_0$.

If $\mathbf{A}$ is close to Hermitian (that is, the anti-Hermitian part of $\mathbf{A}$ is relatively small, see Exercise 0.26**??**), then one of the choices for $\widetilde{\mathbf{r}}_0$ as indicated above works well. In general, for general matrices, it appears that a random vector $\widetilde{\mathbf{r}}_0$ gives the best convergence results.

Exercise 8.5. Here we compare, CGLS, Graig's method and Bi-CG.

(a) Compare the computational costs per step.

(b) Compare the memory requirements.

(c) Suppose $\mathbf{A}$ is positive definite, with all eigenvalues in $[\lambda_-, \lambda_+] \subset (0, \infty)$. Use the result of Exercise 5.17(f) to obtain convergence estimates.

In general optimal Krylov subspace methods (as GMRES and GCR) require the least number of iteration steps to find accurate approximate solutions. However, as we will learn in the next exercise, this is not always the case. A method like Graig's method (that finds an approximate solutions in Krylov subspace generated by $\mathbf{A}\mathbf{A}^*$ rather than by $\mathbf{A}$) can be dramatically better.

Exercise 8.6. **Graig's method versus GCR.**

(a) Consider the block diagonal matrix $\mathbf{A}$ with 2×2 diagonal blocks (n is even). The kth block is given by

$$D_k \equiv \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}.$$

Show that GCR needs two steps to compute the solution of the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$. Discuss the convergence of Graig's method for this equation. Discuss the difference with the system with matrix $\mathbf{A} = \mathrm{diag}(1, 2, \ldots, n)$.

(b) Consider the trivial equation $\mathbf{I}\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$. Form a new system $\mathbf{A}\mathbf{x} = \mathbf{b}$ by bringing the first row of the trivial system to the last row position. Show that, for some well selected $\mathbf{b}$, CGR needs n steps to find an accurate approximate solution. How many steps does Graig's method need?

(c) Summarise your conclusions.

B Bi-CG, breakdowns and bi-Lanczos

Breakdowns. CG when applied to a Hermitian non-definite system can break down. For the same reason, Bi-CG can break down: $\sigma_k \equiv \widetilde{\mathbf{r}}_k^* \mathbf{c}_k$ can be zero; the **breakdown of the LU-decomposition**. Note that $\rho_k \equiv \widetilde{\mathbf{r}}_k^* \mathbf{r}_k$ can be zero as well. This is called the **Lanczos breakdown**. The naming will be explained below. An exact breakdown (that is an exact zero value for ρ or σ) does not often happen. If it does, a simple restart taking the approximate solution at breakdown as initial guess for the restarted iteration, is an effective strategy. Unfortunately, **near breakdown**s, that is, relatively small values for ρ and σ, are not uncommon. They lead to amplifications of rounding errors, resulting in

- a delayed convergence (sometimes failed convergence), and
- inaccurate approximations, that is, in a significant difference between the **recursively updated residual** $\mathbf{r}_k$ and the **true residual** $\mathbf{b} - \mathbf{A}\mathbf{x}_k$ (in a difference that is much larger than the required residual tolerance; see Lecture 8.E).

Rounding errors are amplified for two reasons:

- The scalars ρ and σ are results of inner products. If ρ or σ is relatively small, then an inner product has been computed between two vectors that are nearly mutually orthogonal. This is the situation where inner products are being relatively inaccurately computed (see (8.17))
- If ρ or σ is small, then β or α will be large. The vector update involving such a scalar is one between two large vectors. Nevertheless, the resulting updated vector is usually of modest size, which is precisely the situation where vector updates are inaccurate (see (8.18)).

If the norms of the residuals are plotted (in the **convergence history**), then this phenomenon (assuming some large α) is visible as a peak in the plot.

Bi-Lanczos. CG and the Lanczos process are closely related: Lanczos quantities can be recovered from quantities computed in the CG process, and, conversely, CG can be viewed as an efficient implementation, where on top of the Lanczos process, the projected system has been solved with an LU-decomposition (cf., Exercise 7.5). As Bi-CG is a variant of CG, bi-Lanczos is a variant of Lanczos. It produces projections of general matrices $\mathbf{A}$ on Krylov subspace that are tridiagonal (as Lanczos), rather than (full) Hessenberg (as Arnoldi). As Bi-CG, it relies on bi-orthogonality. Bi-CG and bi-Lanczos are related as CG and Lanczos are.

Exercise 8.7. **Bi-Lanczos.** For some normalised $\mathbf{v}_1$ and $\mathbf{w}_1$, generate $\mathbf{V}_k \equiv [\mathbf{v}_1 \ldots, \mathbf{v}_k]$

and $\mathbf{W}_k \equiv [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ as follows

$$\left\{\begin{array}{l} \widetilde{\mathbf{v}}_k = \mathbf{A}\mathbf{v}_k - \alpha_k \mathbf{v}_k - \sum\limits_{j<k} \beta_k^{(j)} \mathbf{v}_j \perp \mathbf{W}_k, \quad \mathbf{v}_{k+1} = \widetilde{\mathbf{v}}_k / \|\widetilde{\mathbf{v}}_k\|_2 \\ \widetilde{\mathbf{w}}_k = \mathbf{A}^*\mathbf{w}_k - \widetilde{\alpha}_k \mathbf{w}_k - \sum\limits_{j<k} \widetilde{\beta}_k^{(j)} \mathbf{w}_j \perp \mathbf{V}_k, \quad \mathbf{w}_{k+1} = \widetilde{\mathbf{w}}_k / \|\widetilde{\mathbf{w}}_k\|_2 \end{array}\right. \tag{8.1}$$

where the orthogonality restrictions define the coefficients α_j, $\beta_j^{(i)}$, $\widetilde{\alpha}_j$ and $\widetilde{\beta}_j^{(i)}$.

Assume that $D_k \equiv \mathbf{W}_k^* \mathbf{V}_k$ is non-singular.

(a) Prove that $\mathbf{A}\mathbf{V}_{k+1} = \mathbf{V}_k \, \underline{H}_k$ and $\mathbf{A}^*\mathbf{W}_{k+1} = \mathbf{W}_k \, \underline{\widetilde{H}}_k$ for some upper Hessenberg matrices $\underline{H}_k$ and $\underline{\widetilde{H}}_k$. Describe these Hessenberg matrices in terms of the αs and βs.

(b) Prove that D_k is diagonal with diagonal entries $d_{jj} = \delta_j \equiv \mathbf{w}_j^* \mathbf{v}_j$.

(c) Show that $\mathbf{W}_k^* \mathbf{A} \mathbf{V}_k = D_k H_k = \widetilde{H}_k^* D_k$.

(d) Conclude that $\underline{T}_k \equiv \underline{H}_k$ is tri-diagonal,

$$\widetilde{\alpha}_k = \overline{\alpha}_k, \quad \beta_{k+1} = \frac{\delta_{k+1}}{\delta_k} \frac{1}{\|\widetilde{\mathbf{w}}_k\|_2}, \quad \widetilde{\beta}_{k+1} = \frac{\bar{\delta}_{k+1}}{\bar{\delta}_k} \frac{1}{\|\widetilde{\mathbf{v}}_k\|_2}, \tag{8.2}$$

where $\beta_{k+1} = \beta_{k+1}^{(k)}$, $\widetilde{\beta}_{k+1} = \widetilde{\beta}_{k+1}^{(k)}$ and the other βs are 0.

(e) Use the properties from (d) to derive an efficient variant of (8.1): bi-Lanczos.

Note that the matrix $D_k^{-1}\mathbf{W}_k^*$ can be viewed as part of the inverse of $\mathbf{V}_n$, where the columns of $\mathbf{V}_n$ form a Krylov basis (generated by $\mathbf{A}$) for the whole space $\mathbb{C}^n$. Bi-Lanczos computes a basis $\mathbf{V}_n$ for a convenient matrix representation T_n of $\mathbf{A}$. This basis is not orthogonal (as in Lanczos or Arnoldi), but, instead, a partial inverse is computed as well.

We assumed that D_k is non-singular. However, $\delta_k = \mathbf{w}_k^* \mathbf{v}_k$ can be 0. Then the process breaks down (cf., (8.2)): the **Lanczos breakdown**. To avoid breakdowns ($\delta_k = 0$), or to avoid near breakdowns (δ_k is relatively small), the bi-Lanczos process can be extended to allow block diagonal matrices D_k. Strategies that determine the block size depend on how large 'relatively small' is. A process that exploits such a strategy, a **look ahead strategy**, is called a **look-ahead Lanczos** process.

Look-ahead Lanczos processes are more stable than simple bi-Lanczos. But of course, they are more costly. The costs per step are proportional to the block size. Unfortunately, it is not uncommon, that blocks are constructed of very large size (virtually turning look-ahead Lanczos into Arnoldi).

Exercise 8.8. Let $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ and $\mathbf{W}_k = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$. Assume $\mathbf{A}\mathbf{V}_{k+1} = \mathbf{V}_k \, \underline{H}_k$ and $\mathbf{A}^*\mathbf{W}_{k+1} = \mathbf{W}_k \, \underline{\widetilde{H}}_k$ for some upper Hessenberg matrices $\underline{H}_k$ and $\underline{\widetilde{H}}_k$ (cf., Exercise 8.7). Assume that $D_k \equiv \mathbf{W}_k^* \mathbf{V}_k$ is a non-singular block-diagonal matrix. Note that we allow the diagonal blocks to have different dimensions. Actually, the look-ahead strategy determines the dimension dependent on the computed data.

(a) Prove that $\underline{T}_k \equiv \underline{H}_k$ is block tridiagonal.

Put $\Pi_k \equiv \mathbf{I} - \mathbf{V}_k D_k^{-1} \mathbf{W}_k^*$.

(b) Show that Π_k is a (skew) projection, that projects onto $\mathbf{W}_k^\perp$, while Π_k^* projects onto $\mathbf{V}_k^\perp$.

(c) Show that $\mathbf{v}_{k+1} = \widetilde{\mathbf{v}}_k / \|\widetilde{\mathbf{v}}_k\|_2$ if $\widetilde{\mathbf{v}}_k \equiv \Pi_k \mathbf{A} \mathbf{v}_k$ and $\mathbf{w}_{k+1} = \widetilde{\mathbf{w}}_k / \|\widetilde{\mathbf{w}}_k\|_2$ if $\widetilde{\mathbf{w}}_k \equiv \Pi_k^* \mathbf{A}^* \mathbf{w}_k$.

If $\delta_{k+1} = \mathbf{w}_{k+1}^* \mathbf{v}_{k+1}$ is too small, then an orthonormal basis $\widetilde{\mathbf{V}}_p \equiv [\mathbf{v}_{k+1}, \ldots, \mathbf{v}_{k+p}]$ of $\mathcal{K}_p(\Pi_k \mathbf{A}, \mathbf{v}_{k+1})$, and an orthonormal basis $\widetilde{\mathbf{W}}_p \equiv [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_{k+p}]$ of $\mathcal{K}_p(\Pi_k^* \mathbf{A}^*, \mathbf{w}_{k+1})$ can be generated (with Arnoldi) with p the smallest integer such that the smallest singular value of $\widetilde{D} \equiv \widetilde{\mathbf{W}}_p^* \widetilde{\mathbf{V}}_p$ is sufficiently large. Then, the $p \times p$ matrix $\widetilde{D}$ is the next diagonal block of D_k. Note that δ_{k+1} is the smallest singular value of $\widetilde{D}$ for $p = 1$.

(d) Let $\delta \in (0, 1)$, say $\delta = 10^{-4}$. Derive a look-ahead Lanczos algorithm that selects block sizes such that the diagonal blocks have singular values at least δ.

As CG in relation to Lanczos, Bi-CG can be obtained as an efficient implementation, where on top of bi-Lanczos, the LU-decomposition is used for solving the projected system $\underline{T}_k y = \|\mathbf{r}_0\|_2 e_1$. As in case of CG (for non-definite matrices $\mathbf{A}$), a more stable process, is obtained by solving the projected system with a QR-decomposition, thus avoiding near-breakdowns of the LU-decomposition. For an Hermitian matrix $\mathbf{A}$, this leads to MINRES. For a general matrix, we obtain (simple) **QMR** (**quasi minimal residuals**). Bi-Lanczos can be stabilised with look ahead strategies to avoid also near Lanczos breakdowns. These strategies can be incorporated in QMR, leading to full QMR: QMR can be viewed as a stabilised version of Bi-CG. QMR cures all near breakdowns in Bi-CG, but it still shares its other disadvantages: it needs two MVs per step of which one is by $\mathbf{A}^*$, the multiplication be $\mathbf{A}^*$ does not extend the search subspace $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. It merely helps to detect an (hopefully) appropriate approximation in this space. In a next section, Section D, we discuss approaches to get rid of these weak points. But first, in the next section, we reformulate Bi-CG, to ease the subsequent discussion.

C Bi-CG and the shadow Krylov subspace

As we learnt in Exercise 8.3, Bi-CG can be obtained by applying CG to some 'symmetrised' system. As an alternative, we derive below Bi-CG as a method that computes residuals that are orthogonal to a sequence of growing spaces. Eventually, the final space in this sequence will be the whole space, which forces the final residual to be $\mathbf{0}$. In contrast to the Krylov subspace approach that aims for residuals with minimal norm (as GCR and GMRES), this 'bi-orthogonality' approach allows computational steps with short recurrence relations (i.e., with a small fixed number of AXPYs and DOTs per step). Moreover, as we will see in Section D it allows modifications to Bi-CG to get rid of some disadvantages attached to this method (as MVs with $\mathbf{A}^*$).

Exercise 8.9. **Bi-CG.** For square non-singular matrices $\mathbf{A}$, the **Bi-CG recurrence relations** are determined by the coupled two-term recurrences

$$\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k - \beta_k\, \mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\, \mathbf{A}\, \mathbf{u}_k \end{aligned} \tag{8.3}$$

(with $\mathbf{u}_{-1} \equiv \mathbf{0}$) and the orthogonality requirement

$$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \widetilde{\mathbf{r}}_{k-1}. \tag{8.4}$$

Here, $\widetilde{\mathbf{r}}_0, \ldots, \widetilde{\mathbf{r}}_{k-1}$ form a Krylov basis for the **shadow Krylov subspace** $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$.

(a) Show that

$$\operatorname{span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}) = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \quad \text{and} \quad \operatorname{span}(\mathbf{r}_0, \ldots, \mathbf{r}_k) = \mathcal{K}_{k+1}(\mathbf{A}, \mathbf{r}_0).$$

(b) Prove that

$$\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0). \tag{8.5}$$

$\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$ is also called the **dual Krylov subspace**.

(c) Show that for some $\vartheta_k \in \mathbb{C}$ we have that $\widetilde{\mathbf{r}}_k - \bar{\vartheta}_k\, \mathbf{A}^* \widetilde{\mathbf{r}}_{k-1} \in \mathcal{K}_{k-1}(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$.

(d) Put

$$\rho_k \equiv \widetilde{\mathbf{r}}_k^* \mathbf{r}_k \quad \text{and} \quad \sigma_k \equiv \widetilde{\mathbf{r}}_k^* \mathbf{A} \mathbf{u}_k.$$

Show that

$$\alpha_k = \frac{\rho_k}{\sigma_k} \quad \text{and} \quad \beta_k = \frac{\rho_k}{\vartheta_k\, \sigma_{k-1}}.$$

Conclusion. *It suffices to put $\mathbf{r}_k$ and $\mathbf{A}\mathbf{u}_k$ orthogonal to only one vector* (namely, $\widetilde{\mathbf{r}}_{k-1}$) *at the cost of only two inner products* (ρ_k and σ_k, ρ_{k-1} is available from the previous step), *to get a residual $\mathbf{r}_k$ that is orthogonal to a k-dimensional space $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$*. This space is not equal to $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$. We, therefore, refer to the orthogonality relation as **bi-orthogonality**. *We can efficiently produce a sequence of residuals that are orthogonal to a sequence of 'growing' spaces.*

Exercise 8.10. **Bi-CG, the shadow space.** Here, we use the notation and results from Exercise 8.9.

As we have seen in Exercise 8.9, the Bi-CG process requires the construction of a Krylov basis $\widetilde{\mathbf{r}}_0, \widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_{k-1}$ of the shadow Krylov subspace $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$.

In this exercise, we discuss strategies for constructing a shadow Krylov basis. In practice, only the strategy in (8.7) is used. However, the strategies in the parts of this exercise play a crucial role in hybrid Bi-CG methods (see Exercise 8.11).

(a) Show that $\widetilde{\mathbf{r}}_k$ can be expressed as $\widetilde{\mathbf{r}}_k = \bar{q}_k(\mathbf{A}^*)\widetilde{\mathbf{r}}_0$, where q_k is a polynomial of degree k (if $q_k(\zeta) = \gamma_0 + \gamma_1 \zeta + \ldots + \gamma_k \zeta^k \quad (\zeta \in \mathbb{C})$, then $\bar{q}(\zeta) \equiv \bar{\gamma}_0 + \bar{\gamma}_1 \zeta + \ldots + \bar{\gamma}_k \zeta^k$). Show that, for a unique scalar ϑ_k, the polynomial $q_k(\zeta) - \vartheta_k \zeta q_{k-1}(\zeta)$ is of degree $k-1$.

(b) Show that a Krylov basis of $\widetilde{\mathbf{r}}_0, \ldots, \widetilde{\mathbf{r}}_k$ can be obtained with the update formulae

$$\begin{aligned} \widetilde{\mathbf{u}}_k &= \widetilde{\mathbf{r}}_k - \bar{\beta}_k \widetilde{\mathbf{u}}_{k-1}, \quad \widetilde{\mathbf{c}}_k = \mathbf{A}^* \widetilde{\mathbf{u}}_k \\ \widetilde{\mathbf{r}}_{k+1} &= \widetilde{\mathbf{r}}_k - \bar{\alpha}_k \widetilde{\mathbf{c}}_k. \end{aligned} \tag{8.6}$$

How do you initialise this *coupled two-term recurrence* relation? Give an expression for ϑ_k.

Note that $\widetilde{\mathbf{u}}_k$ is not needed in the update formulae for $\mathbf{r}_k$ and $\mathbf{c}_k$. Show that for a suitable $\mathbf{c}_0$ (which one?) (8.6) is equivalent to

$$\begin{aligned} \widetilde{\mathbf{c}}_k &= \mathbf{A}^* \widetilde{\mathbf{r}}_k - \bar{\beta}_k \widetilde{\mathbf{c}}_{k-1}, \\ \widetilde{\mathbf{r}}_{k+1} &= \widetilde{\mathbf{r}}_k - \bar{\alpha}_k \widetilde{\mathbf{c}}_k. \end{aligned} \tag{8.7}$$

Show that (8.7) is equivalent to the *three-term recurrence* relation

$$\widetilde{\mathbf{r}}_{k+1} = (1 - \bar{\gamma}_k)\widetilde{\mathbf{r}}_k - \bar{\alpha}_k \mathbf{A}^* \widetilde{\mathbf{r}}_k + \bar{\gamma}_k \widetilde{\mathbf{r}}_{k-1}, \quad \text{where} \quad \gamma_k \equiv \frac{\alpha_k}{\alpha_{k-1}} \beta_k. \tag{8.8}$$

How does the formula read for $k = 0$? Show that for this for this choice of $\widetilde{\mathbf{r}}_k$ we have that

$$q_{k+1}(\zeta) = (1 - \gamma_k - \alpha_k \zeta)\, q_k(\zeta) + \gamma_k\, q_{k-1}(\zeta) \qquad (\zeta \in \mathbb{C}). \tag{8.9}$$

The classical **Bi-CG algorithm** [Fletcher 1976] incorporates the update formulae of (8.7).

(c) As a variant of (8.8), the scalars can be selected to minimise the norm of $\widetilde{\mathbf{r}}_{k+1}$:

$$\widetilde{\mathbf{r}}_{k+1} = (1 - \bar{\nu}_k)\, \widetilde{\mathbf{r}}_k - \bar{\mu}_k\, \mathbf{A}^* \widetilde{\mathbf{r}}_k + \bar{\nu}_k\, \widetilde{\mathbf{r}}_{k-1}, \tag{8.10}$$

with

$$(\mu_k, \nu_k) \equiv \operatorname{argmin}_{(\mu,\nu)} \|(1 - \bar{\nu})\, \widetilde{\mathbf{r}}_k - \bar{\mu}\, \mathbf{A}^* \widetilde{\mathbf{r}}_k + \bar{\nu}\, \widetilde{\mathbf{r}}_{k-1}\|_2.$$

How do you compute (μ_k, ν_k)? (See also Exercise 5.20.) Give an expression for ϑ_k.

(d) Show that a Krylov basis of $\widetilde{\mathbf{r}}_0, \ldots, \widetilde{\mathbf{r}}_k$ can be obtained with the update formulae

$$\widetilde{\mathbf{r}}_{k+1} = \widetilde{\mathbf{r}}_k - \bar{\omega}_k\, \mathbf{A}^* \widetilde{\mathbf{r}}_k, \quad \text{where} \quad \bar{\omega}_k \equiv \operatorname{argmin}_\omega \|\widetilde{\mathbf{r}}_k - \bar{\omega}\, \mathbf{A}^* \widetilde{\mathbf{r}}_k\|_2. \tag{8.11}$$

Give an expression for ϑ_k and an update formula for the associated polynomials q_k.
Check that (8.11) follows the LMR (local minimal residual) approach.

(e) Select an $\ell \in \mathbb{N}$. Show that a Krylov basis of $\{\widetilde{\mathbf{r}}_j\}$ can be obtained with the update formulae

$$\begin{aligned} &\widetilde{\mathbf{r}}_{k+1} = \mathbf{A}^* \widetilde{\mathbf{r}}_k \quad \text{if} \quad k \notin \{m\ell - 1 \,\big|\, m \in \mathbb{N}\} \\ &\widetilde{\mathbf{r}}_{m\ell} = \widetilde{\mathbf{r}}_{m\ell-\ell} - (\gamma_1 \widetilde{\mathbf{r}}_{m\ell-\ell+1} + \ldots + \gamma_{\ell-1} \widetilde{\mathbf{r}}_{m\ell-1} + \gamma_\ell \mathbf{A}^* \widetilde{\mathbf{r}}_{m\ell-1}), \end{aligned} \tag{8.12}$$

where the $(\gamma_j) = (\gamma_j^{(m)})$ minimise the 2-norm of the new shadow residual $\widetilde{\mathbf{r}}_{m\ell}$.
Give an expression for ϑ_k and update formulae for the associated polynomials q_k.
Check that (8.12) follows an approach that (in exact arithmetic) is equivalent to GCR(ℓ), that is, restarted GCR with restart length ℓ. Note that for $\ell = 1$, this approach is the LMR approach.

(f) Discuss pros and cons (efficiency, use of memory, stability) of the above choices for the shadow Krylov basis (some speculation is allowed).

D Bi-CG and hybrid Bi-CG methods

Except for the Bi-CG residuals and update vectors, which are denoted by $\mathbf{r}_k^{\mathrm{BiCG}}$ and $\mathbf{u}_k^{\mathrm{BiCG}}$, respectively, we use the notation and results from Exercise 8.9 and Exercise 8.10. In addition, we put $\mathbf{Q}_k \equiv q_k(\mathbf{A})$ and

$$\mathbf{r}_k \equiv \mathbf{Q}_k\mathbf{r}_k^{\mathrm{BiCG}}, \quad \mathbf{u}_k \equiv \mathbf{Q}_k\mathbf{u}_{k-1}^{\mathrm{BiCG}} \quad \text{and} \qquad \mathbf{r}'_k \equiv \mathbf{Q}_k\mathbf{r}_{k+1}^{\mathrm{BiCG}}, \quad \mathbf{u}'_k \equiv \mathbf{Q}_k\mathbf{u}_k^{\mathrm{BiCG}}.$$

The polynomials q_k here will be **residual polynomials**, i.e., $q_k(0) = 1$, of exact degree k. $\mathbf{x}_k$ and $\mathbf{x}'_k$ are the associated approximate solutions: $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$, $\mathbf{r}'_k = \mathbf{b} - \mathbf{A}\mathbf{x}'_k$.
The residuals $\mathbf{r}_k$ here have been expressed as $q_k(\mathbf{A})\mathbf{r}_k^{\mathrm{BiCG}}$, a product of the Bi-CG residual and a residual polynomial q_k in $\mathbf{A}$.
Iterative methods that compute residuals of this type are called **Hybrid Bi-CG** methods or **Bi-CG type of product methods** (or **Lanczos type of product methods LTP**). The auxiliary polynomials q_k are called **stabilisation polynomial**s or **acceleration polynomial**s. The naming is clearly inspired by the effect that the polynomials q_k are supposed (hoped) to have.

Exercise 8.11. **Hybrid Bi-CG.**

(a) Show that

$$\rho_k = \widetilde{\mathbf{r}}_0^*(q_k(\mathbf{A})\mathbf{r}_k^{\mathrm{BiCG}}) = \widetilde{\mathbf{r}}_0^*\mathbf{r}_k \quad \text{and} \quad \sigma_k = \mathbf{r}_0^*(\mathbf{A}q_k(\mathbf{A})\mathbf{u}_k^{\mathrm{BiCG}}) = \widetilde{\mathbf{r}}_0^*\mathbf{A}\mathbf{u}'_k.$$

(b) Show that

$$\begin{aligned} \mathbf{u}'_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}'_k &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}'_k \end{aligned} \tag{8.13}$$

Give an update formula for the associated approximate solutions.

(c) Now, we take the strategy of (8.11) as inspiration and compute $\mathbf{r}_{k+1}$ and $\mathbf{u}_{k+1}$ as follows

$$\begin{aligned} &\omega_k \equiv \mathrm{argmin}_\omega \|\mathbf{r}'_k - \omega\,\mathbf{A}\mathbf{r}'_k\|_2, \\ &\mathbf{r}_{k+1} = \mathbf{r}'_k - \omega_k\,\mathbf{A}\mathbf{r}'_k, \\ &\mathbf{u}_{k+1} = \mathbf{u}'_k - \omega_k\,\mathbf{A}\mathbf{u}'_k. \end{aligned} \tag{8.14}$$

Show that this corresponds to the choice $q_{k+1}(\zeta) = (1 - \omega_k\zeta)q_k(\zeta)$ for q_{k+1}.
Give an update formula for the associated approximate solutions.

(d) Give an algorithm for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ iteratively by combining (8.13) and (8.14). Give the algorithm in a form that minimises memory use and computational costs.

This algorithm derived here is called **Bi-CGSTAB** (Bi-CG stabilised, see Alg. 8.3 at the left). The Bi-CGSTAB residuals $\mathbf{r}_k$ can be expressed as a product of the Bi-CG residual and a residual polynomial q_k (in $\mathbf{A}$) that comes from a LMR strategy: Bi-CGSTAB = LMR × Bi-CG.

Exercise 8.12. **Hybrid Bi-CG, 2.** We continue Exercise 8.11.

(a) Use the strategy in (8.8) to derive Alg. 8.3 at the right, an algorithm for the hybrid method called **CGS** (**Conjugate Gradients Squared**): CGS = Bi-CG × Bi-CG.

(b) Use the strategy in (8.10) to derive an hybrid method, called **GPBi-CG** (**general product Bi-CG**): GPBi-CG = 1-GCR × Bi-CG, where 1-GCR is truncated GCR with truncation length 1 (cf., Lecture 5.G); for details, see Exercise 8.13.

(c) Use the strategy in (8.12) to derive Alg. 8.4, an algorithm for the hybrid method called **BiCGstab(ℓ)** (Hint: first cycle ℓ times through the Bi-CG loop (8.13) to form $\mathbf{A}^i\mathbf{Q}_k\mathbf{r}_{k+j}^{\mathrm{BiCG}}$ and $\mathbf{A}^i\mathbf{Q}_k\mathbf{u}_{k+j-1}^{\mathrm{BiCG}}$ ($j = 1, \ldots, \ell$, $i \le j$) and use a strategy as in Exercise 5.21): BiCGstab(ℓ) = GCR(ℓ) × Bi-CG, where GCR(ℓ) is restarted GCR with restart length ℓ (cf., Lecture 5.G).

Exercise 8.13. **Bi-CG and GPBiCG.** We use the following convention. If $\mathbf{x}$, $\mathbf{y}$ and $\widetilde{\mathbf{r}}$ are n-vectors, then

$$\mathbf{z} = \mathbf{x} - \alpha\mathbf{y} \perp \widetilde{\mathbf{r}}$$

```
Bi-CGSTAB
Select $\mathbf{x}_0, \widetilde{\mathbf{r}} \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = 0$, $\omega = \sigma = 1$.
While $\|\mathbf{r}\| > tol$ do
  $\sigma \leftarrow -\omega\sigma$, $\rho = \widetilde{\mathbf{r}}^*\mathbf{r}$, $\beta = \rho/\sigma$
  $\mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$, $\mathbf{c} = \mathbf{A}\mathbf{u}$
  $\sigma = \widetilde{\mathbf{r}}^*\mathbf{c}$, $\alpha = \rho/\sigma$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$,
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{s} = \mathbf{A}\mathbf{r}$, $\omega = \mathbf{s}^*\mathbf{r}/\mathbf{s}^*\mathbf{s}$
  $\mathbf{u} \leftarrow \mathbf{u} - \omega\,\mathbf{c}$
  $\mathbf{x} \leftarrow \mathbf{x} + \omega\,\mathbf{r}$
  $\mathbf{r} \leftarrow \mathbf{r} - \omega\,\mathbf{s}$
end while
```

```
CGS
Select $\mathbf{x}_0, \widetilde{\mathbf{r}} \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0$, $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = \mathbf{w} = \mathbf{0}$, $\rho = 1$.
While $\|\mathbf{r}\| > tol$ do
  $\sigma = -\rho$, $\rho = \widetilde{\mathbf{r}}^*\mathbf{r}$, $\beta = \rho/\sigma$
  $\mathbf{w} \leftarrow \mathbf{u} - \beta\,\mathbf{w}$
  $\mathbf{v} = \mathbf{r} - \beta\,\mathbf{u}$
  $\mathbf{w} \leftarrow \mathbf{v} - \beta\,\mathbf{w}$, $\mathbf{c} = \mathbf{A}\mathbf{w}$
  $\sigma = \widetilde{\mathbf{r}}^*\mathbf{c}$, $\alpha = \rho/\sigma$
  $\mathbf{u} = \mathbf{v} - \alpha\,\mathbf{c}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{A}(\mathbf{v} + \mathbf{u})$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,(\mathbf{v} + \mathbf{u})$
end while
```

ALGORITHM 8.3. Bi-CGSTAB [van der Vorst, '92] (at the left) and Conjugate Gradients Squared [Sonneveld, '89] (at the right) for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* for a general non-singular matrix $\mathbf{A}$.

defines the scalar α and the n-vector $\mathbf{z}$: $\alpha \in \mathbb{C}$ is such that $\mathbf{z} \perp \widetilde{\mathbf{r}}$.

(a) Show that $\alpha = (\widetilde{\mathbf{r}}^*\mathbf{x})/(\widetilde{\mathbf{r}}^*\mathbf{y})$.

(b) Show that the Bi-CG vectors are also defined by the coupled two-term recurrences

$$\begin{aligned}
\mathbf{r}_{k+1}^{\mathrm{BiCG}} &= \mathbf{r}_k^{\mathrm{BiCG}} - \alpha_k\,\mathbf{c}_k^{\mathrm{BiCG}} \perp \widetilde{\mathbf{r}}_k,\\
\mathbf{c}_{k+1}^{\mathrm{BiCG}} &= \mathbf{A}\mathbf{r}_{k+1}^{\mathrm{BiCG}} - \beta_{k+1}\,\mathbf{c}_k^{\mathrm{BiCG}} \perp \widetilde{\mathbf{r}}_k,\\
\mathbf{x}_{k+1}^{\mathrm{BiCG}} &= \mathbf{x}_k^{\mathrm{BiCG}} + \alpha_k\,\mathbf{u}_k^{\mathrm{BiCG}},\\
\mathbf{u}_{k+1}^{\mathrm{BiCG}} &= \mathbf{r}_{k+1}^{\mathrm{BiCG}} - \beta_{k+1}\,\mathbf{u}_k^{\mathrm{BiCG}}.
\end{aligned} \tag{8.15}$$

Again, $\widetilde{\mathbf{r}}_0, \ldots, \widetilde{\mathbf{r}}_{k-1}$ form a Krylov basis for the shadow Krylov subspace $\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)$.
The first two relation determine the convergence of Bi-CG (they form the 'engine' of Bi-CG), the third and fourth relation allow the approximate solution to be updated ($\mathbf{x}_k$ 'gets an almost free ride': there are no additional inner products involved, and no additional MVs, only two extra vector updates).

(c) For a residual polynomial q_k, i.e., $q_k(0) = 1$, of exact degree k, we put $\mathbf{Q}_k \equiv q_k(\mathbf{A})$ and

$$\begin{array}{lll}
\mathbf{r}_k \equiv \mathbf{Q}_k\mathbf{r}_k^{\mathrm{BiCG}}, & \mathbf{r}'_{k+1} \equiv \mathbf{Q}_k\mathbf{r}_{k+1}^{\mathrm{BiCG}}, & \mathbf{r}''_{k+1} \equiv \mathbf{Q}_{k-1}\mathbf{r}_{k+1}^{\mathrm{BiCG}},\\
\mathbf{u}_k \equiv \mathbf{Q}_k\mathbf{u}_k^{\mathrm{BiCG}}, & \mathbf{u}'_{k+1} \equiv \mathbf{Q}_k\mathbf{u}_{k+1}^{\mathrm{BiCG}}, & \mathbf{u}''_{k+1} \equiv \mathbf{Q}_{k-1}\mathbf{u}_{k+1}^{\mathrm{BiCG}},\\
\mathbf{c}_k \equiv \mathbf{A}\mathbf{Q}_k\mathbf{u}_k^{\mathrm{BiCG}}, & \mathbf{c}'_{k+1} \equiv \mathbf{A}\mathbf{Q}_k\mathbf{u}_{k+1}^{\mathrm{BiCG}}, & \mathbf{c}''_{k+1} \equiv \mathbf{A}\mathbf{Q}_{k-1}\mathbf{u}_{k+1}^{\mathrm{BiCG}}.
\end{array}$$

Show that

$$\begin{aligned}
\mathbf{r}'_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{c}_k \perp \widetilde{\mathbf{r}}_0,\\
\mathbf{c}'_{k+1} &= \mathbf{A}\mathbf{r}'_{k+1} - \beta_{k+1}\,\mathbf{c}_k \perp \widetilde{\mathbf{r}}_0,\\
\mathbf{x}'_{k+1} &= \mathbf{x}_k + \alpha_k\,\mathbf{u}_k,\\
\mathbf{u}'_{k+1} &= \mathbf{r}'_{k+1} - \beta_{k+1}\,\mathbf{u}_k.
\end{aligned} \tag{8.16}$$

(d) With q_k and q_{k-1} a residual polynomials of exact degree k, $k-1$, respectively, we update the polynomials as

$$q_{k+1}(\zeta) = (1 - \nu_k - \omega_k\,\zeta)\,q_k(\zeta) + \nu_k\,q_{k-1}(\zeta),$$

```
BiCGStab($\ell$)
$\mathbf{x}=\mathbf{0}$,  $\mathbf{r}=[\mathbf{b}]$.  Choose $\widetilde{\mathbf{r}}$.
$\mathbf{u}=[\mathbf{0}]$,  $\gamma_\ell=\sigma=1$.
While $\|\mathbf{r}\|>tol$ do
   $\sigma\leftarrow-\gamma_\ell\,\sigma$
   For $j=1,\ldots,\ell$ do
      $\rho=\widetilde{\mathbf{r}}^*\mathbf{r}_j$,  $\beta=\rho/\sigma$
      $\mathbf{u}\leftarrow\mathbf{r}-\beta\,\mathbf{u}$,  $\mathbf{u}\leftarrow[\,\mathbf{u},\mathbf{A}\mathbf{u}_j\,]$
      $\sigma=\widetilde{\mathbf{r}}^*\mathbf{u}_{j+1}$,  $\alpha=\rho/\sigma$
      $\mathbf{r}\leftarrow\mathbf{r}-\alpha\,\mathbf{u}_{2:j+1}$,  $\mathbf{r}\leftarrow[\,\mathbf{r},\mathbf{A}\mathbf{r}_j\,]$
      $\mathbf{x}\leftarrow\mathbf{x}+\alpha\,\mathbf{u}_1$
   end for
   $M=\mathbf{r}^*\mathbf{r}$
   Solve $M_{2:\ell+1,2:\ell+1}\vec{\gamma}=M_{2:\ell+1,1}$ for $\vec{\gamma}$
   $\mathbf{u}\leftarrow\mathbf{u}_1-\mathbf{u}_{2:\ell+1}\,\vec{\gamma}$
   $\mathbf{x}\leftarrow\mathbf{x}+\mathbf{r}_{1:\ell}\,\vec{\gamma}$
   $\mathbf{r}\leftarrow\mathbf{r}_1-\mathbf{r}_{2:\ell+1}\,\vec{\gamma}$
end while
```

Algorithm 8.4. BiCGstab(ℓ) [Sleijpen–Fokkema, '93] for solving $\mathbf{A}\mathbf{x}=\mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* for a general non-singular matrix $\mathbf{A}$.
Notation: $\mathbf{u}=[\mathbf{u}_1,\ldots,\mathbf{u}_{j+1}]$ and $\mathbf{r}=[\mathbf{r}_1,\ldots,\mathbf{r}_{j+1}]$ are $n\times(j+1)$ matrices at the end of step j in the 'j-loop', $\mathbf{x}$ and $\widetilde{\mathbf{r}}$ are n-vectors, $\mathbf{u}_{2:j+1}\equiv[\mathbf{u}_2,\ldots,\mathbf{u}_{j+1}]$, etc., γ_ℓ is the last coordinate of the ℓ-vector $\vec{\gamma}$: $\vec{\gamma}=(\gamma_1,\ldots,\gamma_\ell)^{\mathrm{T}}$. At the end and at the start of the 'while-loop', $\mathbf{u}=[\mathbf{u}_1]$ and $\mathbf{r}=[\mathbf{r}_1]$.

where ν_k, $\omega_k\neq0$ are appropriate selected scalars (for a discussion, see below).
Show that q_{k+1} is a residual polynomial of exact degree $k+1$.
Derive the following update formulae for the $\mathbf{r}_k$ and $\mathbf{c}_k$.

From the previous loop: $\rho_k,\mathbf{c}_k,\mathbf{r}_k,\mathbf{c}'_k,\mathbf{r}'_k,\mathbf{A}\mathbf{c}'_k,\mathbf{A}\mathbf{r}'_k$

1 $\sigma_k=\widetilde{\mathbf{r}}_0^*\mathbf{c}_k,\quad\alpha_k=\rho_k/\sigma_k$

2 $\mathbf{r}'_{k+1}=\mathbf{r}_k-\alpha_k\,\mathbf{c}_k,\quad$ compute $\mathbf{A}\mathbf{r}'_{k+1}$

2.1 $\mathbf{r}''_{k+1}=\mathbf{r}'_k-\alpha_k\,\mathbf{c}'_k,\quad\mathbf{A}\mathbf{r}''_{k+1}=\mathbf{A}\mathbf{r}'_k-\alpha_k\,\mathbf{A}\mathbf{c}'_k,$

3 Select ν_k and $\omega_k\neq0$

4 $\mathbf{r}_{k+1}=(1-\nu_k)\,\mathbf{r}'_{k+1}-\omega_k\,\mathbf{A}\mathbf{r}'_{k+1}+\nu_k\,\mathbf{r}''_{k+1}$

5 $\rho_{k+1}=\widetilde{\mathbf{r}}_0^*\mathbf{r}_{k+1},\quad\beta_{k+1}=-\rho_{k+1}/(\omega_k\sigma_k)$

6 $\mathbf{c}'_{k+1}=\mathbf{A}\mathbf{r}'_{k+1}-\beta_{k+1}\,\mathbf{c}_k,\quad$ compute $\mathbf{A}\mathbf{c}'_{k+1}$

6.1 $\mathbf{c}''_{k+1}=\mathbf{A}\mathbf{r}''_{k+1}-\beta_{k+1}\,\mathbf{c}'_k$

7 $\mathbf{c}_{k+1}=(1-\nu_k)\,\mathbf{c}'_{k+1}-\omega_k\,\mathbf{A}\mathbf{c}'_{k+1}+\nu_k\,\mathbf{c}''_{k+1}$

Note that the α_k and β_k are such that $\mathbf{r}'_{k+1}$ and $\mathbf{c}'_{k+1}$ are orthogonal to $\widetilde{\mathbf{r}}_0$.

(e) Derive formulae to initialise the iteration (to compute $\mathbf{r}_1$, ...).

(f) Complete the scheme with update formulae for $\mathbf{x}_k$ and $\mathbf{u}_k$

(g) Select $\nu_k=0$ for all k and select ω_k to minimise the 2-norm of the residual $\mathbf{r}_{k+1}$ in line 4 (from $\mathbf{r}'_{k+1}$ and $\mathbf{r}''_{k+1}$. Check that the Lines 2.1 and 6.1 are superfluous. Derive a Bi-CGSTAB algorithm based on the approach here (try to minimise memory requirements and computational costs per step). Lists the costs per MV.

(h) Select ν_k and ω_k to minimise the 2-norm of the residual $\mathbf{r}_{k+1}$ in Line 4. This leads to (a variant of) **GPBi-CG (general product Bi-CG)**: 1-GCR × Bi-CG (1-GCR is truncated GCR with truncation length 1, cf., Lecture 5.G. Give formulae to compute ν_k and ω_k (cf., Exercise 5.20). List the costs per MV of this scheme.

Exercise 8.14. We use the notation of the previous exercise. In the previous exercise, we updated the Bi-CG part before updating the polynomial part: $\mathbf{Q}_k\mathbf{r}_k^{\text{BiCG}} \rightsquigarrow \mathbf{Q}_k\mathbf{r}_{k+1}^{\text{BiCG}} \rightsquigarrow \mathbf{Q}_{k+1}\mathbf{r}_{k+1}^{\text{BiCG}}$. The update order can be reversed: $\mathbf{Q}_k\mathbf{r}_k^{\text{BiCG}} \rightsquigarrow \mathbf{Q}_{k+1}\mathbf{r}_k^{\text{BiCG}} \rightsquigarrow \mathbf{Q}_{k+1}\mathbf{r}_{k+1}^{\text{BiCG}}$.

(a) Show that, with

$$\begin{aligned} \mathbf{r}_k^+ &\equiv \mathbf{Q}_{k+1}\mathbf{r}_k^{\text{BiCG}}, & \mathbf{r}_k &\equiv \mathbf{Q}_k\mathbf{r}_k^{\text{BiCG}}, & \mathbf{r}_k^- &\equiv \mathbf{Q}_{k-1}\mathbf{r}_k^{\text{BiCG}}, \\ \mathbf{c}_k^+ &\equiv \mathbf{Q}_{k+1}\mathbf{c}_k^{\text{BiCG}}, & \mathbf{c}_k &\equiv \mathbf{Q}_k\mathbf{c}_k^{\text{BiCG}}, & \mathbf{c}_k^- &\equiv \mathbf{Q}_{k-1}\mathbf{c}_k^{\text{BiCG}}, \end{aligned}$$

this leads to the following update formulae for the residuals $\mathbf{r}$ and their update vectors $\mathbf{c}$:

$$\begin{aligned} &\textit{From the previous loop: } \mathbf{r}_k, \mathbf{A}\mathbf{r}_k, \mathbf{r}_k^-, \mathbf{c}_k, \mathbf{c}_k^-, \mathbf{A}\mathbf{c}_k \\ &\text{Select } \nu_k \text{ and } \omega_k \neq 0 \\ &\mathbf{r}_k^+ = (1-\nu_k)\,\mathbf{r}_k - \omega_k\,\mathbf{A}\mathbf{r}_k + \nu_k\,\mathbf{r}_k^- \\ &\mathbf{c}_k^+ = (1-\nu_k)\,\mathbf{c}_k - \omega_k\,\mathbf{A}\mathbf{c}_k + \nu_k\,\mathbf{c}_k^- \\ &\mathbf{r}_{k+1}^- = \mathbf{r}_k - \alpha_k\,\mathbf{c}_k \perp \widetilde{\mathbf{r}}_0, \quad \mathbf{A}\mathbf{r}_{k+1}^- = \mathbf{A}\mathbf{r}_k - \alpha_k\,\mathbf{A}\mathbf{c}_k, \\ &\mathbf{r}_{k+1} = \mathbf{r}_k^+ - \alpha_k\,\mathbf{c}_k^+, \quad \text{compute } \mathbf{A}\mathbf{r}_{k+1} \\ &\mathbf{c}_{k+1}^- = \mathbf{A}\mathbf{r}_{k+1}^- - \beta_{k+1}\,\mathbf{c}_k \perp \widetilde{\mathbf{r}}_0 \\ &\mathbf{c}_{k+1} = \mathbf{A}\mathbf{r}_{k+1} - \beta_{k+1}\,\mathbf{c}_k^+, \quad \text{compute } \mathbf{A}\mathbf{c}_{k+1} \end{aligned}$$

(b) Derive formulae to initialise the iteration (to compute $\mathbf{r}_1, \ldots$). Complete the scheme with update formulae for α_k, β_k, $\mathbf{x}_k$ and $\mathbf{u}_k$

Selecting ν_k and ω_k to minimise the 2-norm of the residual $\mathbf{r}_k^+$ leads to a variant of GPBiCG, called BiCGsafe.

E Effects of rounding errors

To avoid confusion with update vector $\mathbf{u}$, we denote the relative machine precision in this section by $\bar{\xi}$ instead of $\mathbf{u}$. Further, we use the conventions and notations of Section E in Lecture 1: if α is the result of an algorithm, then $\widehat{\alpha}$ or $(\alpha)\widehat{\ }$ is the actual result as obtained by the computer using the same algorithm. The ξ are such that $|\xi| \leq \bar{\xi}$, but ξ at different locations may have different values. And we neglect $\mathcal{O}(\bar{\xi}^2)$-terms.

DOTs, AXPYs and MVs are fundamental operations for Krylov subspace methods. The following results follow from Exercise 1.20. They give error estimates for these operations.

DOT. For $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$,

$$(\mathbf{y}^*\mathbf{x})\widehat{\ } = \mathbf{y}^*\mathbf{x}\left(1 + n\,\xi\,\frac{\|\mathbf{x}\|_2\,\|\mathbf{y}\|_2}{|\mathbf{y}^*\mathbf{x}|}\right) = \mathbf{y}^*\mathbf{x}\left(1 + \frac{n\,\xi}{\cos\angle(\mathbf{x},\mathbf{y})}\right). \tag{8.17}$$

AXPY. For $\mathbf{r}, \mathbf{c} \in \mathbb{C}^n$, $\alpha \in \mathbb{C}$, with $\mathbf{s} \equiv \mathbf{r} + \alpha\,\mathbf{c}$,

$$(\mathbf{r} + \alpha\,\mathbf{c})\widehat{\ } = \mathbf{r} + \alpha\,\mathbf{c} + \delta_s \quad \text{with} \quad |\delta_s| \leq \bar{\xi}\,(|\mathbf{s}| + |\alpha|\,|\mathbf{c}|), \qquad \|\delta_s\|_2 \leq 3\,\bar{\xi}\max(\|\mathbf{s}\|_2, \|\mathbf{r}\|_2). \tag{8.18}$$

MV. For an $n \times n$ matrix $\mathbf{A}$ with at most m non-zero entries in each row, and an n-vector $\mathbf{u}$,

$$(\mathbf{A}\mathbf{u})\widehat{\ } = \mathbf{A}\mathbf{u} + \delta_c \quad \text{with} \quad |\delta_c| \leq m\,\bar{\xi}\,|\mathbf{A}|\,|\mathbf{u}|, \quad \|\delta_c\|_2 \leq m\,\bar{\xi}\,\|\,|\mathbf{A}|\,\|_2\,\|\mathbf{u}\|_2. \tag{8.19}$$

We put

$$\mathcal{C} \equiv m\|\,|\mathbf{A}|\,\|_2\,\|\mathbf{A}^{-1}\|_2. \tag{8.20}$$

Exercise 8.15.

(a) Prove (8.17), (8.18), and (8.19). Discuss the sharpness of these results.

(b) Use Exercise 1.7 to relate $\mathcal{C}$ to the condition number of $\mathbf{A}$.

(c) Assume the exact solution $\mathbf{x}$ is available of the equation $\mathbf{A}\mathbf{x}=\mathbf{b}$. To check whether $\mathbf{x}$ is indeed the exact solution, we compute the residual $\mathbf{b}-\mathbf{A}\mathbf{x}$. Show that

$$(\mathbf{b}-\mathbf{A}\mathbf{x})\widehat{\ }=\delta \quad \text{with} \quad \|\delta\|_2\leq\bar{\xi}\,\mathcal{C}\|\mathbf{b}\|_2. \tag{8.21}$$

Discuss the sharpness of the result.

(d) Suppose at step k, the residual $\mathbf{r}_{k-1}$ is updates as $\mathbf{r}_k=\mathbf{r}_{k-1}-\alpha_{k-1}\mathbf{A}\mathbf{u}_{k-1}$. Assume $\mathbf{r}_{k-1}$, α_{k-1}, and $\mathbf{u}_{k-1}$ are as obtained by the computer (for ease of notation, we drop the $\widehat{\cdot}$ here). Show that $(\mathbf{r}_k)\widehat{\ }=\mathbf{r}_k+\delta_k$ with $\|\delta_k\|_2\leq\bar{\xi}\,(\mathcal{C}+2)\max(\|\mathbf{r}_k\|_2+\|\mathbf{r}_{k-1}\|_2)$, whence

$$\mathbf{r}_k=\mathbf{r}_{k-1}-\alpha_{k-1}\mathbf{A}\mathbf{u}_{k-1} \quad\Rightarrow\quad \frac{\|(\mathbf{r}_k)\widehat{\ }-\mathbf{r}_k\|_2}{\|\mathbf{r}_k\|_2}\lesssim\bar{\xi}\,\mathcal{C}\left(1+\frac{\|\mathbf{r}_{k-1}\|_2}{\|\mathbf{r}_k\|_2}\right). \tag{8.22}$$

Convergence and smooth convergence. The perturbation of the kth residual from computations in the kth step is relatively large if the convergence history exhibits a large **peak** at step $k-1$, i.e., $\|\mathbf{r}_{k-2}\|_2\ll\|\mathbf{r}_{k-1}\|_2$ and $\|\mathbf{r}_k\|_2\ll\|\mathbf{r}_{k-1}\|_2$. Large relative errors may affect the speed of convergence. For this reason it is wise to use methods that tend to avoid large peaks. Words as "stabilising" have been used in the naming of (variants of) methods to indicate that the (variant of the) method avoids peaks. However, not all intermediate results will be plotted. Therefore, a smooth convergence history does not necessarily imply that large relative errors did not effect the convergence.

Accuracy and the residual gap. Iterative methods compute at step k a scalar ρ_k that, in exact arithmetic, is equal to the residual norm $\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2$. In GMRES, ρ_k is computed as $\rho_k=\min\{\|\rho_0\,e_1-\underline{H}_k y\|_2\,\big|\,y\in\mathbb{C}^k\}$, in methods as CG, Bi-CGSTAB, etc., $\rho_k=\|\mathbf{r}_k\|_2$, where the residual $\mathbf{r}_k$ is obtained by recursive updating. In rounded arithmetic, the norm of the **true residual** $\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k$ need not be equal to the computed quantity $\widehat{\rho}_k$. It is important to control, or to be able to estimate, the **residual gap** γ_k

$$\gamma_k\equiv|\,\widehat{\rho}_k-\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k\|_2\,|. \tag{8.23}$$

If the residual gap is small (much smaller than the required residual accuracy *tol*), then the computed quantity $\widehat{\rho}_k$ can safely be used in a stopping criterion that checks the residual size: if $\widehat{\rho}_k\leq tol$, then $\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2\leq\gamma_k+\widehat{\rho}_k\lesssim tol$.

Note that in the expression for the true residual in (8.23), we consider $\widehat{\mathbf{x}}_k$ as computed by the computer according to the algorithm, but we assumed the true residual to be the exact residual for the computed solution, i.e., the multiplication by $\mathbf{A}$ and the subtraction from $\mathbf{b}$ is assumed to be exact. Of course, this is in practice is not possible either, cf., (8.21). In particular, it does not make sense to have a tolerance *tol* that is less than $\bar{\xi}\,\mathcal{C}\,\|\mathbf{b}\|_2$. Therefore, we consider a method to be **accurate** (for a problem $\mathbf{A}\mathbf{x}=\mathbf{b}$) if its residual gaps γ_k are less than (a modest multiple of) $\bar{\xi}\,\mathcal{C}\|\mathbf{b}\|_2$.

In practice it turns out that (γ_k) is an increasing sequence, while for many methods $\widehat{\rho}_k\to0$ for $k\to\infty$: the quantity $\widehat{\rho}_k$ decreases far below the value $\bar{\xi}\,\mathcal{C}\,\|\mathbf{b}\|_2$.

Note that, except for a factor $\bar{\xi}\,\mathcal{C}\,\|\mathbf{b}\|_2$, the true residual norm $\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k\|_2$ could be computed from the computed approximate solution $\widehat{\mathbf{x}}_k$. However, computing the true residual in order to use it in a stopping criterion, has two disadvantages

- it increases (doubles?) the number of MVs dramatically,
- the iteration may fail to stop: if the optimal accuracy ($\min_k\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k\|_2$) is larger than the required tolerance (or, equivalently, $\gamma_k>tol$), then testing $\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k\|_2<tol$ will never be successful.

Since the sequence $(\widehat{\rho}_k)$ (usually) decreases towards 0, $\widehat{\rho}_k<tol$ will occur. Whether this guarantees sufficient accuracy, i.e., $\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}\|_2\leq tol$, depends on the size of the residual gap γ_k. Upon termination the size of the true residual can be computed (requiring only one extra

MV). If it turns out that $\|\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k\|_2 \not\lesssim \mathit{tol}$, then the method could be restarted for the residual shifted problem $\mathbf{A}\mathbf{y}=\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k$.

Although the method stops if $\widehat{\rho}_k$ is used in a stopping criterion, many MVs can be 'wasted' if $\mathit{tol} \ll \gamma_k$. As stated before, it is important to be able to control the residual gap.

Recursively updated residual methods. Methods as Richardson, CG, Bi-CGSTAB, etc., use update formulae for computing approximate solutions and residuals:

$$\left\{\begin{array}{ll} \mathbf{x}_k=\mathbf{x}_{k-1}+\alpha_{k-1}\mathbf{u}_{k-1}, & \mathbf{c}_{k-1}=\mathbf{A}\mathbf{u}_{k-1}, \\ \mathbf{r}_k=\mathbf{r}_{k-1}-\alpha_{k-1}\mathbf{c}_{k-1}, & \rho_k\equiv\|\mathbf{r}_k\|_2 \end{array}\right. \tag{8.24}$$

Inexact MVs. Often the rounding errors in the MV dominate the rounding errors from the other arithmetic operations (as AXPYs and DOTs). Therefore, to simplify the analysis, we now assume that *except for the MV the arithmetic operations are exact.*

In Exercise 8.16, we will see that, under this assumption, the residual gap in these **recursively updated residual methods** (**RURal methods**) is determined by the largest intermediate residual:

$$\gamma_k \lesssim \bar{\xi}\, 2k\,\mathcal{C}\max_{j\leq k}\|\mathbf{r}_j\|_2, \tag{8.25}$$

with $\mathcal{C}$ as in (8.20). The update vector $\mathbf{u}_{k-1}$ and the scalar α_{k-1} are determined in the algorithms in lines preceding (8.24). Actually, the way the update vector is computed defines the method. Here, we assume that $\mathbf{c}_{k-1}$ is computed by explicit matrix-vector multiplication from $\mathbf{u}_{k-1}$ (which is the case in CG, Bi-CGSTAB, etc.). In some methods (as GCR), $\mathbf{c}_{k-1}$ is in exact arithmetic equal to $\mathbf{A}\mathbf{u}_{k-1}$, but the vectors have been obtained by some recursive update steps after the MV. Below, we use that $\|(\mathbf{c}_{k-1})\widehat{\ }-\mathbf{c}_{k-1}\|_2\leq\bar{\xi}\mathcal{C}\|\mathbf{c}_{k-1}\|_2$ (for given $\mathbf{u}_{k-1}$). This is estimate is not correct if vectors have been adapted by recursive steps after the MV.

Exercise 8.16. **Recursively updated residuals.** Consider the update formulae (8.24) with $\mathbf{x}_0=\mathbf{0}$.

(a) Prove that, in exact arithmetic, $\mathbf{r}_k=\mathbf{b}-\mathbf{A}\mathbf{x}_k$, $\rho_k=\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2$ if $\mathbf{r}_0=\mathbf{b}-\mathbf{A}\mathbf{x}_0$.

(b) Assume that the α_{k-1} and $\mathbf{u}_{k-1}$ are as obtained by the computer (that is, for ease of notation, we dropped the $\widehat{\cdot}$ in the notation of the α_js and $\mathbf{u}_j$). Prove that, in rounded arithmetic, the residual gap γ_k can be bounded by

$$\gamma_k\leq\|\widehat{\mathbf{r}}_k-(\mathbf{b}-\mathbf{A}\widehat{\mathbf{x}}_k)\|_2\leq\bar{\xi}\,(2\mathcal{C}+4)\sum_{j=0}^{k}\|\mathbf{r}_j\|_2\lesssim\bar{\xi}\,2k\,\mathcal{C}\max_{j\leq k}\|\mathbf{r}_j\|_2. \tag{8.26}$$

Here, you may assume that MVs only are inexact. Discuss the sharpness of these estimates.

(c) Prove (8.25) for RURal methods with $\mathbf{x}_0=\mathbf{0}$. Conclude that
such a method is accurate $\quad\Leftrightarrow\quad$ $\max\|\mathbf{r}_j\|_2$ is less than some modest multiple of $\|\mathbf{b}\|_2$.

If $\max_j\|\mathbf{r}_j\|_2\gg\|\mathbf{b}\|_2$, then, as we learn from (8.25), the result using (8.24) is not accurate. As an alternative for (8.24), consider

$$\left\{\begin{array}{ll} \mathbf{x}_k=\mathbf{x}_{k-1}+\alpha_{k-1}\mathbf{u}_{k-1}, & \\ \mathbf{r}_k=\mathbf{b}-\mathbf{A}\mathbf{x}_k, & \rho_k\equiv\|\mathbf{r}_k\|_2. \end{array}\right. \tag{8.27}$$

Then, clearly, the $\widehat{\rho}_k$ is equal to the norm of the true residual (except for some terms of order $\bar{\xi}$). However, this approach may pose two problems.

- In many methods, the computation of α_{k-1} involves $\mathbf{A}\mathbf{u}_{k-1}$. In such a case, (8.27) requires additional MVs.
- If $\|\mathbf{r}_k\|_2\ll\|\mathbf{b}\|_2$, then the perturbation of $\mathbf{r}_k$ by rounding errors when $\mathbf{r}_k$ is computed as in (8.27) and (even if) $\mathbf{x}_k$ is exact, may be much larger than when computed by (8.24) (cf., (8.22)):

$$\mathbf{r}_k=\mathbf{b}-\mathbf{A}\mathbf{x}_k\quad\Rightarrow\quad\|(\mathbf{r}_k)\widehat{\ }-\mathbf{r}_k\|_2\lesssim\bar{\xi}\,\mathcal{C}(\|\mathbf{b}\|_2+\|\mathbf{r}_k\|_2). \tag{8.28}$$

```
CGS [Neumaier]
Select $\mathbf{x}_0,\ \widetilde{\mathbf{r}} \in \mathbb{C}^n$
$\mathbf{x} = \mathbf{x}_0,\ \ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
$\mathbf{u} = \mathbf{w} = \mathbf{0},\ \ \rho = 1$
$\mathbf{x}' = \mathbf{0},\ \ \mathbf{b}' = \mathbf{r}$
While $\|\,\mathbf{r}\,\| > tol$ do
   $\sigma = -\rho,\ \ \rho = \widetilde{\mathbf{r}}^*\mathbf{r},\ \ \beta = \rho/\sigma$
   $\mathbf{w} \leftarrow \mathbf{u} - \beta\,\mathbf{w}$
   $\mathbf{v} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$
   $\mathbf{w} \leftarrow \mathbf{v} - \beta\,\mathbf{w},\ \ \mathbf{c} = \mathbf{A}\mathbf{w}$
   $\sigma = \widetilde{\mathbf{r}}^*\mathbf{c},\ \ \alpha = \rho/\sigma$
   $\mathbf{u} = \mathbf{v} - \alpha\,\mathbf{c}$
   $\mathbf{x}' \leftarrow \mathbf{x}' + \alpha\,(\mathbf{v} + \mathbf{u})$
   $\mathbf{r} \leftarrow \mathbf{b}' - \mathbf{A}\mathbf{x}'$
   if $\|\mathbf{r}\|_2 \leq \|\mathbf{b}'\|_2$ then
      $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{x}',\ \ \mathbf{x}' = \mathbf{0},\ \ \mathbf{b}' = \mathbf{r}$
end while
```

Algorithm 8.5. The Neumaier version of Conjugate Gradients Squared [Neumaier, '94] for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ with residual accuracy *tol* for a general non-singular matrix $\mathbf{A}$.

In CGS the computation of the α_{k-1} does not rely on $\mathbf{c}_{k-1} \equiv \mathbf{A}\mathbf{u}_{k-1}$. Hence, in CGS, we only have to address the second issue. To avoid relatively large perturbations in case $\|\mathbf{r}_k\|_2 \ll \|\mathbf{b}\|_2$ when using true residuals, we can apply a **residual shift** to the system of equations: from step k on, solve $\mathbf{A}\mathbf{x}' = \mathbf{b}' \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_k$. Then $\mathbf{x} = \mathbf{x}_k + \mathbf{x}'$.

Exercise 8.17. **Reliable updated residuals in CGS.**

(a) Prove (8.28).

(b) Prove that in exact arithmetic, the versions Alg. 8.5 and Alg. 8.3 of CGS are equivalent.

(c) Prove that the relative perturbation The relative perturbation of $\mathbf{r}_k$ by rounding errors in both versions of CGS are comparable.

(d) Argue that version Alg. 8.5 is accurate.

In methods as Bi-CGSTAB, $\mathbf{c}_{k-1}$ is needed in the computation of α_{k-1}. Therefore, unlike CGS, the computation of a true residual in Bi-CGSTAB costs an additional MV. To avoid many additional MVs, we could perform (8.24) only at a few selected steps, say at the steps $k_0 = 0, k_1, k_2, \ldots$. Then, at step $k = k_i$, $\mathbf{x}_k$ is not exact, and we have to combine the result in (8.28) with the one in (8.26):

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k \quad \Rightarrow \quad \|(\mathbf{r}_k)^\frown - \mathbf{r}_k\|_2 \lesssim \bar{\xi}\,\mathcal{C}(\|\mathbf{b}\|_2 + 2k \max \|\mathbf{r}_j\|_2). \tag{8.29}$$

Here, $k = k_{i+1}$ and we take the maximum over all $j = k_i + 1, k_i + 1, \ldots, k_{i+1}$.

To avoid hampering convergence by large perturbations if the recursively updated residual is replaced by a true residual, we have to avoid large intermediate residuals in between the computation of true residual. The problem of relative large perturbations on the residuals when working with true residuals in case $\|\mathbf{r}\|_2 \ll \|\mathbf{b}\|_2$ can simply be avoided by not working with true residuals in these cases or by 'decreasing $\|\mathbf{b}\|_2$' with a residual shift of the linear equation.

```
Reliable updating residuals
$\mathbf{x}' \leftarrow \mathbf{x}' + \alpha_{k-1}\mathbf{u}_{k-1}, \quad \mathbf{c}_{k-1} = \mathbf{A}\mathbf{u}_{k-1}$
$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\mathbf{c}_{k-1}$
if ‘compute_true_residual’ or ‘residual_shift’
    $\mathbf{r}_k = \mathbf{b}' - \mathbf{A}\mathbf{x}'$
    if ‘residual_shift’
        $\mathbf{x}_k = \mathbf{x}_k + \mathbf{x}', \quad \mathbf{x}' = \mathbf{0}, \quad \mathbf{b}' = \mathbf{r}_k$
```

Algorithm 8.6. Reliable updated residuals in RURal methods.
The method is assumed to be initiated with $\mathbf{x}' = \mathbf{0}$, $\mathbf{b}' = \mathbf{b} - \mathbf{A}\mathbf{x}_0$.

Alg. 8.6 gives the modification to the update formulae (8.24) that allow the computation of true residuals at selected steps and that allow residual shifts.

Exercise 8.18. **Reliable updated residuals in RURal methods.**

(a) Prove that in exact arithmetic, Alg. 8.6 is equivalent to the update formulae (8.24).

(b) Analyse the additional costs that are involved with the modifications of Alg. 8.6.

(c) Prove there is no need to compute a true residual if all $\|\mathbf{r}_j\|_2$ are less than $\|\mathbf{b}\|_2$ since the previous computation of the true residual.

We suggest to perform a residual shift only if $\|\mathbf{r}_k\|_2$ decreased significantly below $\|\mathbf{b}\|_2$ and there were large residuals before; `‘residual_shift’` is ‘`true`’ iff

$$\|\mathbf{r}_k\| \leq 10^{-2}\|\mathbf{b}\|_2 \quad \& \quad \|\mathbf{b}\|_2 \leq \max \|\mathbf{r}_j\|_2.$$

Here, we take the maximum over all residual norms since the previous residual shift. We compute true residuals if a preceding residual was large and larger than $\|\mathbf{b}\|_2$:
`‘compute_true_residual’` is ‘`true`’ iff

$$\|\mathbf{r}_k\|_2 \leq 10^{-2} \max \|\mathbf{r}_j\|_2 \quad \& \quad \|\mathbf{b}\|_2 \leq \max \|\mathbf{r}_j\|_2.$$

Here, we take the maximum over all residual norms since the previous computation of the true residual.

We obtain accurate approximate solutions at the costs of only a few extra MVs without hampering the convergence.

Inexact DOTs. In the following exercise, we address the problem of inaccurate inner products in the computation of the update scalars due to near orthogonality. To simplify the discussion, you may assume that the MVs and AXPYs are exact and only DOT products are polluted by rounding errors.

Exercise 8.19. **Maintaining convergence.** In exact arithmetic we know that

$$\rho_k = \widetilde{\mathbf{r}}_k^* \mathbf{r}_k^{\mathrm{BiCG}} = \widetilde{\mathbf{r}}_0^* \mathbf{r}_k, \quad \text{where} \quad \mathbf{Q}_k \equiv q_k(\mathbf{A}) \quad \text{and} \quad \widetilde{\mathbf{r}}_k \equiv \bar{\mathbf{Q}}_k \widetilde{\mathbf{r}}_0, \quad \mathbf{r}_k \equiv \mathbf{Q}_k \mathbf{r}_k^{\mathrm{BiCG}}$$

However, in rounded arithmetic, the scalars $\widetilde{\mathbf{r}}_k^* \mathbf{r}_k^{\mathrm{BiCG}}$ and $\widetilde{\mathbf{r}}_0^* \mathbf{r}_k$ may be different.

(a) Prove that

$$(\widetilde{\mathbf{r}}_0^* \mathbf{r}_k)\hat{} = \widetilde{\mathbf{r}}_0^* \mathbf{r}_k \left(1 + n\xi \frac{|\widetilde{\mathbf{r}}_0|^* |\mathbf{r}_k|}{|\widetilde{\mathbf{r}}_0^* \mathbf{r}_k|}\right) = \widetilde{\mathbf{r}}_0^* \mathbf{r}_k \left(1 + \frac{n\xi}{\cos \angle(\widetilde{\mathbf{r}}_0 \mathbf{r}_k}\right).$$

(b) Note that ρ_k does not change by replacing q_k by $q_k + p_{k-1}$ with p_{k-1} any polynomial of degree $< k$. Suppose $q_k(\zeta) = \Omega_k \zeta^k +$lower degree terms. We consider strategies of selecting the polynomial q_k such that $\frac{1}{|\Omega_k|}\|\mathbf{r}_k\|_2$ as small as possible (without affecting speed of convergence and efficiency too much). Why do we focus on the size of $\frac{1}{|\Omega_k|}\|\mathbf{r}_k\|_2$?

(c) Suppose we update our polynomials q_k by degree one factors: $q_{k+1}(\zeta) = (1 - \omega_k\zeta)q_k(\zeta)$. Then, $\mathbf{r}_{k+1} = \mathbf{r}'_k - \omega_k\mathbf{A}\mathbf{r}'_k$. Show that the choice of ω_k such that $\mathbf{r}'_k - \omega_k\mathbf{A}\mathbf{r}'_k \perp \mathbf{r}'_k$ leads to the smallest $\frac{1}{|\Omega_{k+1}|}\|\mathbf{r}_{k+1}\|_2$ if we can only vary ω_k.

(d) Given $\mathbf{r}'_k$, the choice for ω_k such that

- $\mathbf{r}'_k - \omega_k\mathbf{A}\mathbf{r}'_k \perp \mathbf{A}\mathbf{r}'_k$ leads to smallest residuals norms $\|\mathbf{r}_{k+1}\|_2$, (as in Bi-CGSTAB) while
- $\mathbf{r}'_k - \omega_k\mathbf{A}\mathbf{r}'_k \perp \mathbf{r}'_k$ leads to smallest $\frac{1}{|\Omega_{k+1}|}\|\mathbf{r}_{k+1}\|_2$, i.e., highest accuracy of ρ_k.

For ease of discussion, put $\mathbf{s} \equiv \mathbf{r}'_k/\|\mathbf{r}'_k\|_2$, $\mathbf{t} \equiv \mathbf{A}\mathbf{r}'_k/\|\mathbf{A}\mathbf{r}'_k\|_2$. Let μ be such that $\mathbf{s} - \mu\mathbf{t} \perp \mathbf{t}$. Show that $\mathbf{s} - \frac{1}{\mu}\mathbf{t} \perp \mathbf{s}$. Express the optimal ω's in terms of μ, $\|\mathbf{r}'_k\|_2$ and $\|\mathbf{A}\mathbf{r}'_k\|_2$.

Note that small μ implies a large reduction of the residual norm (small $\|\mathbf{r}_{k+1}\|_2$ as compared to $\|\mathbf{r}'_k\|_2$). Unfortunately, this is precisely the case where the loss of accuracy is expected to be large as well (Why?). The two requirements, 'good reduction of the residual norm','maintain good accuracy', seem to be conflicting.

We do not need to have ρ_k in high accuracy. It turns out that as long as ρ_k has three or more significant digits (i.e., the relative error is less than 10^{-3}), then the convergence is not hampered. However, in some cases, the ρ_k of Bi-CGSTAB has less than one digit of accuracy. Then, the process will stagnate from k on. To maintain convergence, ρ_k has to be computed in higher accuracy. The following strategy seems to be effective

$$\mathbf{r}'_k - \omega_k\mathbf{A}\mathbf{r}'_k, \quad \text{with} \quad \omega_k = \operatorname{sign}(\mu)\max(0.7, |\mu|)\frac{\|\mathbf{r}'_k\|_2}{\|\mathbf{A}\mathbf{r}'_k\|_2}. \tag{8.30}$$

Check that this leads to the smallest residual norm if $|\mu| > 0.7$. It avoids significant loss of accuracy in case $|\mu|$ is small.

(e) With $\mathbf{R} \equiv [\mathbf{r}'_k, \mathbf{A}\mathbf{r}'_k]$, let $M \equiv \mathbf{R}^*\mathbf{R}$. Show that ω_k can be computed from only the matrix coefficients of the 2×2 matrix M.

Inexact AXPYs. Some short recurrence methods, as MINRES and QMR, rely on three-term vector-recursions, while in others, as CG and SYMMLQ, two-term vector-recurrences play a central role. The three-term vector-recurrence methods tend to be less accurate, that is, they lead to a larger residual gap (typically of order $\bar{\xi}\mathcal{C}_2^2(\mathbf{A})$), whereas two-term vector-recurrences tend to offer full accuracy. Below, we try to give some insight in this situation. We now are interested in the effects of errors in the AXPYs. To simplify analysis, we, therefore, assume that only AXPYs are polluted by rounding errors: the MVs and DOTs are exact.

If $\mathbf{W}$ is the matrix with columns from the vector-recursions, then, as we will see in Exercise 8.20, a back-ward error in $\widehat{\mathbf{W}}$, the computed $\mathbf{W}$, can be described typically by perturbation terms as (cf., (8.32) and (8.34))

$$\Delta_1 \text{ or } \Delta_2 R \ \text{ such that } \ |\Delta_1| \leq \kappa_1\,\bar{\xi}\,|\mathbf{W}|\,|R| \ \text{ and } \ |\Delta_2| \leq \kappa_2\,\bar{\xi}\,|\mathbf{W}|.$$

Here, κ_1 and κ_2 are modest constants and R is the matrix that describes the recursion (see Exercise 8.20 for details). Note that, except for the modest factor κ_1/κ_2, estimating $\|\Delta_1\|_2$ and $\|\Delta_2 R\|_2$ lead to the same (sharp) upper bound (why?). Nevertheless, there is a huge difference between these terms. The perturbation term $\Delta_2 R$ is *structured and may lead to cancellation of errors* in, e.g., the computed solution of the linear system: for instance, if $\mathbf{x} = \mathbf{W}R^{-1}z$ then, in contrast to $\|\Delta_2 R R^{-1}z\|_2$, a bound on $\|\Delta_1 R^{-1}z\|_2$ will contain a factor as $\mathcal{C}_2(R)$ (why?).

Exercise 8.20. Consider a three-term vector-recurrence

$$\mathbf{w}_k = (\mathbf{v}_k - (\beta_k\mathbf{w}_{k-1} + \gamma_k\mathbf{w}_{k-2}))/\alpha_k \quad (k = 1, 2, \ldots), \tag{8.31}$$

where $(\mathbf{v}_k)$ is a given sequence of n-vectors, (α_k), (β_k) and (γ_k) are given sequences of scalars, $\gamma_0 \equiv 0$ and, with $\mathbf{w}_{-1} \equiv \mathbf{w}_0 \equiv \mathbf{0}$, the n vectors $\mathbf{w}_k$ are to be computed.

Put $\mathbf{W}_k \equiv [\mathbf{w}_1, \ldots, \mathbf{w}_k]$, $\mathbf{V}_k \equiv [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, and let $R_k = (R_{ij})$ be the $k \times k$ upper tri-diagonal triangular matrix with entries $R_{jj} = \alpha_j$, $R_{j-1\,j} = \beta_j$ and $R_{j-2\,j} = \gamma_j$.

(a) Show that, $\mathbf{W}_k$ solves

$$\mathbf{W}_k R_k = \mathbf{V}_k.$$

(b) If $n = 1$, i.e., we are considering a three-term scalar-recurrence relation, then (why?)

$$\alpha_k(1+2\xi)\widehat{\mathbf{w}}_k = \mathbf{v}_k - \beta_k(1+2\xi)\widehat{\mathbf{w}}_{k-1} - \gamma_k(1+2\xi)\widehat{\mathbf{w}}_{k-2} \quad (k = 1, 2, \ldots).$$

Hence, in this case, for some $k \times k$ matrix Δ_k we have that (why?)

$$\widehat{\mathbf{W}}_k(R_k + \Delta_k) = \mathbf{V}_k \quad \text{with } \Delta_k \text{ such that } \quad |\Delta_k| \leq 2\bar{\xi}|R_k|. \tag{8.32}$$

(c) If $n > 1$, then the above result can be applied row-wise (why?). However, this does not lead to (8.32). Why not? However, for some $n \times k$ matrix Δ_k, we do have that (why?)

$$\widehat{\mathbf{W}}_k R_k + \Delta_k = \mathbf{V}_k \quad \text{with } \Delta_k \text{ such that } \quad |\Delta_k| \leq 2\bar{\xi}|\mathbf{W}_k|\,|R_k|. \tag{8.33}$$

(d) Discuss the sharpness of (8.32) and (8.33) and the consequences for the forward error, i.e., $\|\widehat{\mathbf{W}}_k - \mathbf{W}_k\|_2$.

Now, assume that $\gamma_k = 0$ for all k, that is, the three-term recurrences collapse to two-term recurrences. For $n \times n$ diagonal matrices δ_k, to be specified below, let $\mathbf{w}'_k$ be such that $\widehat{\mathbf{w}}_k = (\mathbf{I} + \delta_k)\mathbf{w}'_k$.

(e) Show that the δ_k can be selected such that

$$\mathbf{W}'_k R_k = \mathbf{V}_k + \Delta^{(1)}_k \quad \text{with } \Delta^{(1)}_k \text{ such that } \quad |\Delta^{(1)}_k| \leq 3k\,\bar{\xi}\,|\mathbf{V}_k|$$

and $|\delta_k| \leq 3k\,\bar{\xi}\,\mathbf{I}$, whence

$$\widehat{\mathbf{W}}_k R_k = \mathbf{V}_k + \Delta^{(1)}_k + \Delta^{(2)}_k R_k \quad \text{with } \Delta^{(2)}_k \text{ such that } \quad |\Delta^{(2)}_k| \leq 3k\bar{\xi}\,|\mathbf{W}_k|. \tag{8.34}$$

With $\Delta_k \equiv \Delta^{(1)}_k + \Delta^{(2)}_k R_k$, we clearly have that $|\Delta_k| \leq 6k\bar{\xi}\,|\mathbf{W}_k|\,|R_k|$, which is a modest factor $3k$ larger than the estimate in (8.33). However, in applications (see, for instance, Exercise 8.21), one is interested in estimates for

$$\frac{\|\Delta_k R_k^{-1} z\|_2}{\|\mathbf{V}_k\|_2}$$

rather than for $\|\Delta_k\|_2$. Show that,

$$\frac{\|\Delta_k R_k^{-1} z\|_2}{\|\mathbf{V}_k\|_2} \leq 6k\,\bar{\xi}\,\mathcal{C}_2(R_k)^2 \frac{\|z\|_2}{\|R_k\|_2} \quad \text{and} \quad \frac{\|\Delta_k R_k^{-1} z\|_2}{\|\mathbf{V}_k\|_2} \leq 6k^2\,\bar{\xi}\,\mathcal{C}_2(R_k)\frac{\|z\|_2}{\|R_k\|_2}$$

for Δ_k of (8.33) and Δ_k of (e), respectively. Typically $\mathcal{C}_2(R_k) \approx \mathcal{C}_2(\mathbf{A})$ for large k and for ill-conditioned matrices $\mathbf{A}$ is the second estimate is more favourable than the first one.

MINRES in following exercise forms an example to which the situation of Exercise 8.20 applies.

Exercise 8.21. **MINRES.** Assume that $\mathbf{A}$ is Hermitian. Consider the Lanczos relation $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{T}_k$, i.e., $\mathbf{V}_k$ is orthonormal and $\underline{T}_k$ is tri-diagonal.

(a) Let $\underline{T}_k = \underline{Q}_k R_k$ be the QR-decomposition of the tri-diagonal matrix $\underline{T}_k$. If $\mathbf{x}_k = \mathbf{V}_k\,y_k$ is the MINRES approximate solution (with $\mathbf{x}_0 = \mathbf{0}$), then, with $z_k \equiv \underline{Q}_k^*\,\vec{\beta}$, and $\mathbf{W}_k \equiv \mathbf{V}_k R_k^{-1}$, we have that $\mathbf{x}_k = \mathbf{W}_k\,z_k$. The vector z_k can be obtained by a repeated application of Givens rotations: the coordinates of z_k can be computed by two-term scalar-recurrences, whereas the computation of $\mathbf{W}_k$ relies on a three-term vector-recursion.

Assume $\widehat{\mathbf{W}}_k$ is computed from $\mathbf{W}_k R_k = \mathbf{V}_k$. Let $\widehat{\mathbf{x}}_k \equiv \widehat{\mathbf{W}}_k\, z_k$ and let $\widehat{\mathbf{r}}_k$ be the true residual (for $\widehat{\mathbf{x}}_k$). We assume update errors in the computation of $\mathbf{W}_k$ as analysed in Exercise 8.20, and we assume the other operations to be exact. The perturbation matrix Δ_k is as in (8.33).

(b) Show that $\widehat{\mathbf{x}}_k - \mathbf{x}_k = \Delta_k R_k^{-1} z_k = \Delta_k\, y_k$ and $\mathbf{r}_k - \widehat{\mathbf{r}}_k = \mathbf{A}\Delta_k R_k^{-1} z_k = \mathbf{A}\Delta_k\, y_k$.

(c) This leads to the following bound on the residual gap

$$\gamma_k \leq \|\widehat{\mathbf{r}}_k - \mathbf{r}_k\|_2 \leq 2\bar{\xi}\, \||\mathbf{A}|\, |\mathbf{V}_k R_k^{-1}|\, |R_k|\, |R_k^{-1} z_k|\|_2$$

(d) Prove that, for $k = n$, we have that $\mathcal{C}_2(R_k) = \mathcal{C}_2(\mathbf{A})$.

(e) Show that the residual gap in MINRES is bounded by

$$\gamma_k = \|\widehat{\mathbf{r}}_k - \mathbf{r}_k\|_2 \leq 2k\,\bar{\xi}\, \|\mathbf{A}\|_2\, \mathcal{C}_2(\mathbf{A})\, \|\mathbf{x}\|_2 \leq 2k\,\bar{\xi}\, \mathcal{C}_2(\mathbf{A})^2\, \|\mathbf{b}\|_2.$$

(f) If $\mathbf{r}_k$ converges towards 0, then for large k, we have that

$$\frac{\|\widehat{\mathbf{x}}_k - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq 2k\,\bar{\xi}\, \mathcal{C}_2(\mathbf{A}) \quad \text{and} \quad \frac{\|\widehat{\mathbf{r}}_k\|_2}{\|\mathbf{b}\|_2} \leq 2k\,\bar{\xi}\, \mathcal{C}_2(\mathbf{A})^2 \;:$$

The error is more favourable than may be anticipated from the true residual.

Note that SYMMLQ exploits a three-term scalar-recurrence and a two-term vector-recurrence, whereas in MINRES the three-term recurrence is on vectors while the two-term recurrence is on scalars. This explains why SYMMLQ is more accurate for ill-conditioned systems.

Note that QMR is the MINRES variant for non-symmetric systems. In particular, QMR also exploits three-term vector-recurrences.