LECTURE 30: IRREGULAR SPARSITY

1. Finite element discretization

For the steady-state heat equation $-\nabla\cdot(\alpha\nabla u)=f$ with spatially-varying diffusivity, symmetric discretizations on uniform grids lead to systems $\boldsymbol{A}\boldsymbol{u}=\boldsymbol{c}$ with $\boldsymbol{A}=\boldsymbol{A}^T$, and a regular sparsity pattern. Irregular domain discretization will lead to more complicated sparsity patterns that require different approaches to solving the linear system. It is important to link the changes in the structure of $\boldsymbol{A}$ to specific aspects of the approximation procedure. Consider the difficulties of applying finite difference discretization on a domain Ω of arbitrary shape with boundary $\Gamma=\partial\Omega$ (Fig. 1). At grid node (i,j) closer to the boundary than the uniform spacing h, centered finite difference formulas would refer to undefined values outside the domain. One-sided finite difference formulas would fail to take into account boundary values for the problem. Taylor series expansions could be used,

$$u_A=u(\xi h,jh)=u_{i,j}+\left(\frac{\partial u}{\partial x}\right)_{i,j}(\xi h)+\frac{1}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}(\xi h)^2+\cdots$$

$$u_{i+1,j}=u_{i,j}+\left(\frac{\partial u}{\partial x}\right)_{i,j}h+\frac{1}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j}h^2+\cdots$$

from which elimination of the second derivative leads to an approximation of the first derivative as

$$\left(\frac{\partial u}{\partial x}\right)_{i,j}=\frac{u_A-\xi^2 u_{i+1,j}-(1-\xi^2)u_{i,j}}{\xi h(1-\xi)}. \tag{1}$$

Note that setting $\xi=-1$ would place A at a grid node, $u_A=u_{i-1,j}$ and from (1) the familiar centered finite difference approximation of the first derivative

$$\left(\frac{\partial u}{\partial x}\right)_{i,j}=\frac{u_{i+1,j}-u_{i-1,j}}{2h},$$

is recovered. For an arbitrary domain the values of ξ,η would vary and the resulting linear system $\boldsymbol{A}\boldsymbol{u}=\boldsymbol{c}$ would no longer be symmetric. From a physical perspective this might be surprising at first since the operator $\mathscr{L}=-\nabla\cdot(\alpha\nabla)$ is isotropic, but this is true for an inifinitesimal domain. Upon irregular discretization the problem $\boldsymbol{A}\boldsymbol{u}=\boldsymbol{c}$ is only an approximation of the physical problem $\mathscr{L}u=f$, and can exhibit different behavior, in this case loss of isotropy near the boundaries.

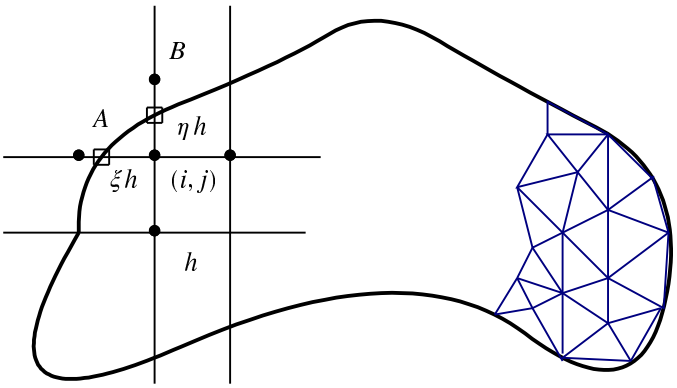

Figure 1. Left: Modified finite difference stencil near a boundary not aligned with the grid. Boundary points A,B are distances ξh, ηh from the nearest interior node, with $\xi,\eta\in(-1,1)$. Right: Triangles covering the domain.

Computing the appropriate mesh size fractions $(\xi h,\eta h)$ for all grid points near a boundary is an onerous task, and suggests seeking a different approach. A frutiful idea is to separate the problem of geometric description from that of physics expressed by some operator $\mathscr{L}$. Domains within $\mathbb{R}^d$ of arbitrary complexity can be approximated to any

desired precision by a simplicial covering. Simplicia are the simplest geometric objects with non-zero measure μ in a space. For $d=1$ these are line segments that can approximate arbitrary curves. The corresponding simplicia for $d=2$ and $d=3$ are triangles and tetrahedra, respectively. Consider $d=2$ and specify a set of triangles $\{T_k \mid k=1,2,\ldots,n\}$ with vertices V_j, $j=1,\ldots,m$, that form a partition of precision $\varepsilon \geqslant 0$ of the domain Ω,

$$\forall k,l \in \{1,2,\ldots,n\}, \mu(T_k \cap T_l) = 0, \left| \mu\left(\bigcup_{k=1}^{n} T_k \right) - \mu(\Omega) \right| \leqslant \varepsilon.$$

The above state that intersections of triangles must have zero measure in $d=2$, i.e., triangles can share edges or vertices but cannot overlap over a non-zero area. The area of the union of triangles approximates the area of the overall domain Ω.

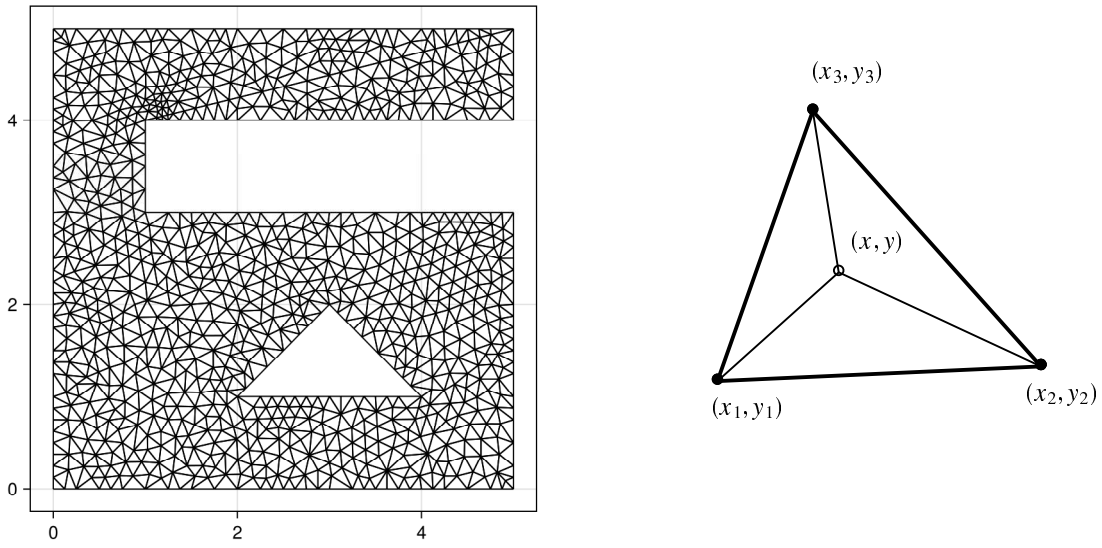

Figure 2. Left: Triangulation of a domain with a hole. Right: Triangle form function

In a finite difference discretization the function $u\colon \mathbb{R} \to \Omega$ is approximated by a set of values $\{u_{i,j}\}$, often referred to as a grid function. Similarly, a set of values $u_j \cong u(x_j, y_j)$ can be defined at the triangle vertices $V_j(x_j, y_j)$. Denote the vertex coordinates of triangle T by (x_j, y_j), $j=1,2,3$. Values of $u(x,y)$ within the triangle T are determined through piecewise interpolation, a generalization of one-dimensional B-splines, using the form functions

$$N_1(x,y) = \frac{1}{2A}\begin{vmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & y_3 \end{vmatrix}, N_2(x,y) = \frac{1}{2A}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x & x_3 \\ y_1 & y & y_3 \end{vmatrix}, N_3(x,y) = \frac{1}{2A}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x \\ y_1 & y_2 & y \end{vmatrix},$$

with A the triangle area

$$A = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}.$$

Note that for $(x,y) \in T$ the form functions give the fraction of the overall area occupied by the interior triangles such that $N_j(x,y) \in [0,1]$. The linear spline interpolation p_1 of u based upon the vertex values u_1, u_2, u_3 is

$$u(x,y) \cong p_1(x,y) = \sum_{j=1}^{3} u_j N_j(x,y), \tag{2}$$

the familiar form of a linear combination. It is customary to set $N_j(x,y) = 0$ if $(x,y) \notin T$, recovering the framework of B-splines. Since $u(x,y)$ thus approximated is non-zero only over the single triangle T, such an approach is commonly referred to as a finite element method (FEM).

Various approaches can be applied to derive an algebraic system for the vertex values from the conservation law of interest. Consider the operator $\mathscr{L} = -\nabla \cdot (\alpha \nabla)$ and the static equilibrium equation $\mathscr{L} u = f$ in Ω with Dirichlet boundary conditions $u = g$ on $\Gamma = \partial\Omega$. When u denotes temperature, this is a statement of thermal equilibrium where heat fluxes $q = -\alpha \nabla u$ balance out external heating f and imposed temperature values on the boundary. One commonly used approach closely resembles the least squares approximation of $\boldsymbol{b} \in \mathbb{R}^n$,

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\| .$$

The approximant $\tilde{\boldsymbol{b}}$ of $\boldsymbol{b}$ in this case is its projection onto $C(\boldsymbol{A})$, $\tilde{\boldsymbol{b}} = \boldsymbol{Q}\boldsymbol{Q}^T \boldsymbol{b}$, with $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ the (incomplete) QR decomposition of $\boldsymbol{A}$. The error of this approximation is $\boldsymbol{e} = \tilde{\boldsymbol{b}} - \boldsymbol{b} \in N(\boldsymbol{A}^T)$ is orthogonal to $C(\boldsymbol{A})$

$$\boldsymbol{Q}^T \boldsymbol{e} = \boldsymbol{Q}^T (\boldsymbol{Q}\boldsymbol{Q}^T \boldsymbol{b} - \boldsymbol{b}) = \boldsymbol{0}. \tag{3}$$

The generalization of (3) in which the finite-dimensional vector $\boldsymbol{b} \in \mathbb{R}^n$ is replaced by the function $u \in C^{(2)}(\Omega)$ that satisfies $\mathscr{L} u = f$ is

$$\left(N_i, \mathscr{L} \sum_{j=1}^{3} u_j N_j(x,y) - f \right) = 0, i = 1,2,3, \tag{4}$$

for each triangle T_k with (u, v) denoting the scalar product

$$(u, v) = \int_\Omega u(x,y)\, v(x,y)\, \mathrm{d}\omega .$$

The analogy can be understood by recognizing that finite element approximants lie within the span of the form functions $\{N_i^k\}$ for all triangles T_k and their vertices $j = 1, 2, 3$. This known as a Galerkin method with (4) expressing orthogonality of the error $e = \mathscr{L}\tilde{u} - f$ and all form functions $\{N_i^k\}$, leading to

$$\left(N_i, \mathscr{L} \sum_{j=1}^{3} u_j N_j(x,y) - f \right) = 0 \Rightarrow \sum_{j=1}^{3} \left(\int_T N_i(x,y)\, \mathscr{L} N_j(x,y)\, \mathrm{d}\omega \right) u_j = \int_T N_i(x,y)\, f(x,y)\, \mathrm{d}\omega .$$

The null result of applying the second-order differential operator $\mathscr{L} = -\nabla \cdot (\alpha \nabla)$ onto a linear form function N_j is avoided through integration by parts (divergence theorem

$$\int_{T_k} N_i(x,y)\, \mathscr{L} N_j(x,y)\, \mathrm{d}\omega = -\int_{T_k} N_i(x,y) [\, \nabla \cdot (\alpha \nabla)\, N_j(x,y)\,]\, \mathrm{d}\omega = \int_{T_k} \alpha\, [\nabla N_i(x,y)\,] \cdot [\, \nabla N_j(x,y)\,]\, \mathrm{d}\omega = a_{ij}^{(k)} .$$

Assembling contributions from all triangles T_k results in a linear system $\boldsymbol{A}\,\boldsymbol{u} = \boldsymbol{c}$, expressing an approximation of the steady-state heat equation $\mathscr{L} u = f$.

It is illuminating to note that though the physical process itself is isotropic, the FEM approximation typically leads to a non-symmetric system matrix $\boldsymbol{A}$ due to the different sizes of the triangularization elements. The fact that the approximation depends on the domain discretization is not surprising; this also occurred for finite difference approximations as evidenced by the eigenvalue dependency on grid spacing h, e.g., $v_l = 4 \sin^2(l \pi h / 2)$. The particularity of FEM discretization is that the single parameter h has been replaced by the individual geometry of *all* triangles within the domain partition. It is to be expected that the resulting matrices will exhibit condition numbers that are monotonic with respect to $\max_k \mu(T_k) / \min_k \mu(T_k)$, the ratio of the area of the largest triangle area to the smallest. This is readily understood: when $\min_k \mu(T_k) \to 0$ the spanning set $\{N_i^k\}$ becomes linearly dependent since one of its members approaches the zero element. The same effect is obtained if the aspect ratio of a triangle becomes large (i.e., one of its angles is close to zero), since again the spanning set is close to linearly dependent. A finite element system matrix $\boldsymbol{A}$ will still exhibit sparsity since the form functions are non-zero on only one triangle. The sparsity pattern is however determined by the connectivity, i.e., the number of triangles at each shared vertex. A typical sparsity matrix is shown in Fig. 3. If the physical principle of action and reaction (Newton's third law) is respected by discretization the matrix will still be symmetric, a considerable advantage with respect to the use of Taylor series to extend finite difference methods to arbitrary domains.

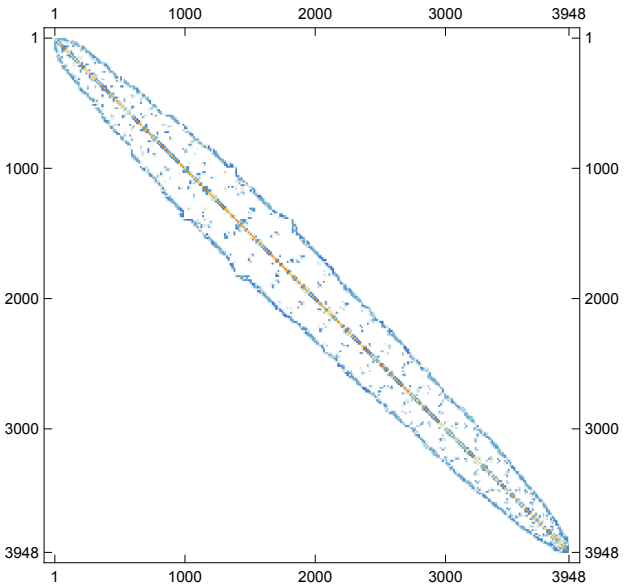

Figure 3. Non-zero elements with $\boldsymbol{A} \in \mathbb{R}^{m \times m}$, $m = 3948$ of a matrix from the Boeing-Harwell collection.

2. Krylov methods, Arnoldi iteration

From the above general observations it becomes apparent that solution techniques considered up to now are inadequate. Factorization methods such as LU or QR would lead to fill-in and loss of sparsity. Additive splitting is no longer trivially implemented since connectivity has be accounted for other than by simple loops. The already slow convergence rate of methods based upon additive splitting is likely to degrade further or perhaps diverge due to the influence the spatial discretization has upon eigenvalues of the iteration matrix $\boldsymbol{M} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}$. Similar considerations apply to gradient descent.

An alternative approach is to seek a suitable basis $\mathcal{B} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_m\}$ in which to iteratively construct improved approximations $\boldsymbol{u}_k$ of the solution $\boldsymbol{u}$ of the discretized system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{c}$,

$$\boldsymbol{u} \cong \boldsymbol{u}_k = \boldsymbol{Q}_n \boldsymbol{x}, \boldsymbol{Q} = [\, \boldsymbol{q}_1 \;\; \boldsymbol{q}_2 \;\; \cdots \;\; \boldsymbol{q}_n \,] \in \mathbb{R}^{m \times n}.$$

Vectors within the basis set should be economical to compute and also lead to fast convergence in the sense that the coefficient vector $\boldsymbol{x}$ should have components that rapidly decrease in absolute value. One idea is to recognize that for a sparse system matrix $\boldsymbol{A}$ with an average of $p \ll m$ nonzero elements per row the cost to evaluate the matrix-vector product $\boldsymbol{A}\boldsymbol{u}$ is only $\mathcal{O}(mp)$ as opposed to $\mathcal{O}(m^2)$ for a full system with $p = m$. This suggests considering a vector set

$$\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots\},$$

starting from some arbitrary vector $\boldsymbol{b}$. The resulting sequence of vectors has been encountered already in the power iteration method for computing eigenvalues and eigenvectors of $\boldsymbol{A}$, and for large n, $\boldsymbol{A}^n\boldsymbol{b}$ will tend to belong to the null space associated with the largest eigenvalue, leading to the ill-conditioned matrices

$$\boldsymbol{V}_n = [\, \boldsymbol{b} \;\; \boldsymbol{A}\boldsymbol{b} \;\; \ldots \;\; \boldsymbol{A}^{n-1}\boldsymbol{b} \,] \in \mathbb{R}^{m \times n}.$$

As in the development of power iteration into the QR method for eigenvalue approximation, the ill-conditioning of $\boldsymbol{V}_n$ can be eliminating by orthogonalization of $\boldsymbol{V}_n$. In fact, the procedure can be organized so as to iteratively add one more vector $\boldsymbol{q}_{n+1}$ to the vectors $\boldsymbol{Q}_n = [\, \boldsymbol{q}_1 \;\; \boldsymbol{q}_2 \;\; \cdots \;\; \boldsymbol{q}_n \,]$ already obtained from orthogonalization of $\boldsymbol{V}_n$. Start in iteration $n = 1$ from $\boldsymbol{q}_1 = \boldsymbol{b} / \|\boldsymbol{b}\|$. A new direction is obtained through multiplication by $\boldsymbol{A}$, $\boldsymbol{v}_2 = \boldsymbol{A}\boldsymbol{q}_1$. Gram-Schmidt orthogonalization leads to

$$\boldsymbol{v}_2 = h_{11}\boldsymbol{q}_1 + h_{21}\boldsymbol{q}_2, h_{11} = \boldsymbol{q}_1^T \boldsymbol{v}_2, h_{21} = \|\boldsymbol{v}_2 - h_{11}\boldsymbol{q}_1\|, \boldsymbol{q}_2 = (\boldsymbol{v}_2 - h_{11}\boldsymbol{q}_1) / h_{21}.$$

The above can be written as.

$$\boldsymbol{A}\,[\boldsymbol{q}_1] = [\,\boldsymbol{q}_1\;\;\boldsymbol{q}_2\,]\begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix}, \boldsymbol{A}\boldsymbol{Q}_1 = \boldsymbol{Q}_2\tilde{\boldsymbol{H}}_1. \tag{5}$$

Note that

$$C(\boldsymbol{V}_n) = C(\boldsymbol{Q}_n) = \text{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{b}\},$$

thus constructing a sequence of vector spaces of increasing dimension $C(\boldsymbol{Q}_1) \subseteq C(\boldsymbol{Q}_2) \subseteq \ldots \subseteq C(\boldsymbol{Q}_n)$ when $\boldsymbol{b}$ is not an eigenvector of $\boldsymbol{A}$. These are known as Krylov spaces $\mathcal{K}_n = C(\boldsymbol{Q}_n)$. In the n^{th} iteration (5) generalizes to

$$\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{A}\,[\,\boldsymbol{q}_1\;\;\boldsymbol{q}_2\;\;\cdots\;\;\boldsymbol{q}_n\,] = [\,\boldsymbol{A}\boldsymbol{q}_1\;\;\boldsymbol{A}\boldsymbol{q}_2\;\;\cdots\;\;\boldsymbol{A}\boldsymbol{q}_n\,] = [\,\boldsymbol{q}_1\;\;\boldsymbol{q}_2\;\;\cdots\;\;\boldsymbol{q}_n\;\;\boldsymbol{q}_{n+1}\,]\begin{bmatrix} h_{11} & h_{21} & \cdots & h_{1,n} \\ h_{21} & h_{22} & \cdots & h_{2,n} \\ 0 & h_{32} & \ddots & \vdots \\ \vdots & \vdots & \ddots & h_{n,n} \\ 0 & 0 & \cdots & h_{n+1,n} \end{bmatrix} = \boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n. \tag{6}$$

The resulting algorithm is known as the Arnoldi iteration.

Algorithm (Arnoldi)

$\boldsymbol{b}$, $\boldsymbol{q}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$
for $n = 1, 2, \ldots$
  $\boldsymbol{v} = \boldsymbol{A}\boldsymbol{q}_n$
  for $j = 1$ to n
    $h_{jn} = \boldsymbol{q}_j^T\,\boldsymbol{v}$
    $\boldsymbol{v} = \boldsymbol{v} - h_{jn}\boldsymbol{q}_j$
  end
  $h_{n+1,n} = \|\boldsymbol{v}\|$
  $\boldsymbol{q}_{n+1} = \boldsymbol{v}/h_{n+1,n}$
end

3. GMRES

Approximate solutions $\boldsymbol{u}_n \in C(\boldsymbol{Q}_n)$ to the system $\boldsymbol{A}\,\boldsymbol{u} = \boldsymbol{c}$ can now be obtained by choosing the starting vector of the embedded Krylov spaces as $\boldsymbol{b} = \boldsymbol{c}$ and solving the least squares problem

$$\min_{\boldsymbol{u}_n} \|\boldsymbol{A}\boldsymbol{Q}_n\boldsymbol{u}_n - \boldsymbol{c}\|. \tag{7}$$

Problem (7) is reformulated using (6) as

$$\min_{\boldsymbol{u}_n} \|\,\boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n\boldsymbol{u}_n - \boldsymbol{c}\| \Leftrightarrow \min_{\boldsymbol{u}_n} \|\,\tilde{\boldsymbol{H}}_n\boldsymbol{u}_n - \boldsymbol{w}\|,$$

with $\boldsymbol{w} = \|\boldsymbol{c}\|\,\boldsymbol{e}_1$ since $\|\,\boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n\boldsymbol{u}_n - \boldsymbol{c}\| = \|\,\boldsymbol{Q}_{n+1}^T(\boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n\boldsymbol{u}_n - \boldsymbol{c})\|$. This is known as the generalized minimal residual algorithm (GMRES).

Algorithm (GMRES)

$\boldsymbol{c}$, $s = \|\boldsymbol{c}\|$, $\boldsymbol{q}_1 = \boldsymbol{c}/s$
for $n = 1, 2, \ldots$
  $\boldsymbol{v} = \boldsymbol{A}\boldsymbol{q}_n$
  for $j = 1$ to n
    $h_{jn} = \boldsymbol{q}_j^T\,\boldsymbol{v}$
    $\boldsymbol{v} = \boldsymbol{v} - h_{jn}\boldsymbol{q}_j$
  end
  $h_{n+1,n} = \|\boldsymbol{v}\|$
  # Least squares approximation $\tilde{\boldsymbol{H}}_n\boldsymbol{u}_n \cong s\boldsymbol{e}_1$
  $\boldsymbol{P}\boldsymbol{R} = \text{qr}(\tilde{\boldsymbol{H}}_n)$; $\boldsymbol{w} = s\boldsymbol{P}^T\boldsymbol{e}_1$; $\boldsymbol{u}_n = \boldsymbol{R}\backslash\boldsymbol{w}$
  if $\|\boldsymbol{u}_n - \boldsymbol{u}_{n-1}\| \leqslant \varepsilon$ return$(\boldsymbol{u}_n)$
  $\boldsymbol{q}_{n+1} = \boldsymbol{v}/h_{n+1,n}$
end