

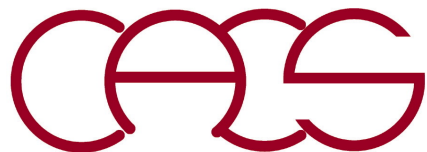
Lanczos Method for Eigensystems

Aiichiro Nakano

*Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Quantitative & Computational Biology
University of Southern California*

Email: anakano@usc.edu

- 1. $O(N)$ (vs. conventional $O(N^3)$) eigensolver**
- 2. Krylov subspace**



B. N. Parlett
The Symmetric Eigenvalue Problem
(Prentice-Hall, '80) Secs. 11-13



Rayleigh Quotient

Theorem

Let \mathbf{A} be an $n \times n$ real symmetric matrix, $\lambda_1[\mathbf{A}] \leq \dots \leq \lambda_n[\mathbf{A}]$ its eigenvalues in ascending order, $\mathbf{x} \in \mathbb{R}^n$, & the Rayleigh quotient

$$\rho(\mathbf{x}; \mathbf{A}) = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad \text{then} \quad \begin{cases} \lambda_1[\mathbf{A}] = \min_{\mathbf{x} \in \mathbb{R}^n} \rho(\mathbf{x}; \mathbf{A}) \\ \lambda_n[\mathbf{A}] = \max_{\mathbf{x} \in \mathbb{R}^n} \rho(\mathbf{x}; \mathbf{A}) \end{cases}$$

Proof

Let $\mathbf{q}^{(k)}$ be the k -th orthonormalized eigenvector of \mathbf{A} , $\mathbf{A} \mathbf{q}_k = \lambda_k \mathbf{q}_k$, & orthogonal transformation matrix, $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$, then

$$\mathbf{Q}^T \mathbf{A} \mathbf{Q} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Let $\mathbf{x} = \mathbf{Q} \mathbf{z}$ (note $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$), then

$$\rho(\mathbf{x}; \mathbf{A}) = \frac{\mathbf{z}^T \mathbf{Q}^T \mathbf{A} \mathbf{Q} \mathbf{z}}{\mathbf{z}^T \mathbf{Q}^T \mathbf{Q} \mathbf{z}} = \frac{z_1^2 \lambda_1 + \dots + z_n^2 \lambda_n}{z_1^2 + \dots + z_n^2}$$

which is a weighted average of $\lambda_1, \dots, \lambda_n$, & the minimum is when $\mathbf{z}^T = (1, 0, \dots, 0) = \mathbf{e}_1$ & $\mathbf{x} = \mathbf{Q} \mathbf{e}_1 = \mathbf{q}_1$.

Rayleigh-Ritz Procedure

Theorem

Let $\{\mathbf{q}_1, \dots, \mathbf{q}_m\}$ ($\mathbf{q}_j \in \mathbb{R}^n; j = 1, \dots, m; m < n$) be an orthonormal set, so that any vector $\mathbf{x} \in \mathbb{R}^n$ in the range is expressed as a linear combination of $\mathbf{q}_1, \dots, \mathbf{q}_m$:

$$\mathbf{x} = z_1 \mathbf{q}_1 + \dots + z_m \mathbf{q}_m \quad \text{or} \quad \begin{matrix} 1 & & m & & 1 \\ n & \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} & = & n & \begin{bmatrix} \mathbf{q}_1 & \dots & \mathbf{q}_m \end{bmatrix} & \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix} & m \end{matrix} = \mathbf{Q} \mathbf{z}$$

$\mathbf{x} \in \text{rank-}m \text{ subspace} \subset \mathbb{R}^n$

then the best approximations for $\lambda_1[\mathbf{A}]$ & $\lambda_n[\mathbf{A}]$ are obtained by diagonalizing

$$\begin{matrix} m \times m & m \times n & n \times n & n \times m \\ \mathbf{H} & = & \mathbf{Q}^T & \mathbf{A} & \mathbf{Q} \end{matrix}$$

as $\lambda_1[\mathbf{H}]$ & $\lambda_m[\mathbf{H}]$.

Proof

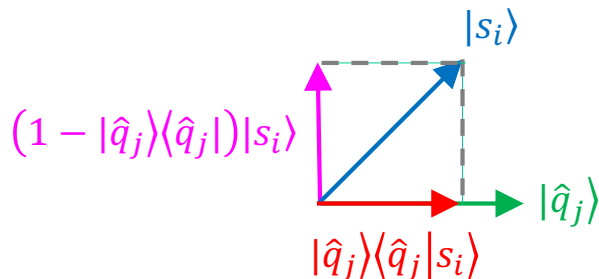
Note $(\mathbf{Q}^T \mathbf{Q})_{ij} = \sum_{k=1}^n Q_{ki} Q_{kj} = \mathbf{q}_i \cdot \mathbf{q}_j = \delta_{ij}$

then
$$\rho(\mathbf{x}; \mathbf{A}) = \frac{\mathbf{z}^T \mathbf{Q}^T \mathbf{A} \mathbf{Q} \mathbf{z}}{\mathbf{z}^T \mathbf{Q}^T \mathbf{Q} \mathbf{z}} = \frac{\mathbf{z}^T \mathbf{H} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} = \frac{z_1^2 \lambda_1(\mathbf{H}) + \dots + z_m^2 \lambda_m(\mathbf{H})}{z_1^2 + \dots + z_m^2}$$

the minimum of which is $\lambda_1[\mathbf{H}]$ (cf. proof in the previous page).

Orthogonalization by QR Decomposition

- **Gram-Schmidt orthonormalization:** The orthonormal set Q required for the Rayleigh-Ritz procedure is obtained starting from an arbitrary set of m vectors, $S = [s_1 \dots s_m]$ ($s_j \in \mathbb{R}^n$) as (see [supplementary note](#)):



$$q_1 = s_1 / |s_1|$$

for $i = 2$ to m

$$q'_i = s_i - \sum_{j=1}^{i-1} q_j (q_j \cdot s_i)$$

$$q_i = q'_i / |q'_i|$$

endfor

Project out!

$$\left(1 - \sum_{j=1}^{i-1} |\hat{q}_j\rangle\langle\hat{q}_j| \right) |s_i\rangle$$

- The Gram-Schmidt procedure amounts to QR decomposition, $S = QR$, where R is an $m \times m$ right-triangle matrix:

$$\begin{matrix} n & m & & m & & m & \\ \left[\begin{array}{cccc} s_1 & s_2 & s_3 & s_4 \end{array} \right] & = & n & \left[\begin{array}{cccc} q_1 & q_2 & q_3 & q_4 \end{array} \right] & \left[\begin{array}{cccc} |q'_1| & q_1 \cdot s_2 & q_1 \cdot s_3 & q_1 \cdot s_4 \\ 0 & |q'_2| & q_2 \cdot s_3 & q_2 \cdot s_4 \\ 0 & 0 & |q'_3| & q_3 \cdot s_4 \\ 0 & 0 & 0 & |q'_4| \end{array} \right] & m \end{matrix}$$

$$\therefore s_i = |q'_i| q_i + \sum_{j=1}^{i-1} q_j (q_j \cdot s_i)$$

cf. QR decomposition
 $A = Q \begin{bmatrix} \blacksquare & & \\ & \blacksquare & \\ & & \blacksquare \end{bmatrix}$

Rayleigh-Ritz Algorithm

1. Start from $\mathbf{S} = [\mathbf{s}_1 \dots \mathbf{s}_m]$ ($\mathbf{s}_j \in \mathbf{R}^n$) & do Gram-Schmidt orthonormalization, $\mathbf{S} = \mathbf{Q}\mathbf{R}$, to obtain an orthonormal set $\mathbf{Q} = [\mathbf{q}_1 \dots \mathbf{q}_m]$
2. Form $\mathbf{H} = \mathbf{Q}^T \mathbf{A} \mathbf{Q}$
3. Diagonalize \mathbf{H} to get $\lambda_1[\mathbf{H}], \dots, \lambda_m[\mathbf{H}]$: $\mathbf{H} \mathbf{g}_k = \lambda_k[\mathbf{H}] \mathbf{g}_k$ ($k = 1, \dots, m$)
4. Approximations of $\lambda_1[\mathbf{A}]$ & $\lambda_n[\mathbf{A}]$ are given by $\lambda_1[\mathbf{H}]$ & $\lambda_m[\mathbf{H}]$ with the corresponding eigenvectors, $\mathbf{y}_k = \mathbf{Q} \mathbf{g}_k$ ($k = 1$ & m).

$$\begin{array}{c}
 \mathbf{H} \\
 \overbrace{\mathbf{Q}^T \mathbf{A} \mathbf{Q}} \\
 \mathbf{g}_k = \lambda_k(\mathbf{H}) \mathbf{g}_k \\
 * \quad \downarrow \quad \mathbf{Q} \times \\
 \mathbf{A} \overbrace{\mathbf{Q} \mathbf{g}_k}^{\mathbf{y}_k} \cong \lambda_k(\mathbf{H}) \overbrace{\mathbf{Q} \mathbf{g}_k}^{\mathbf{y}_k}
 \end{array}$$

* $\mathbf{Q}\mathbf{Q}^T \neq \mathbf{I}^{n \times n}$ but spans a subspace of the n -dimensional space
 cf. $\mathbf{Q}^T \mathbf{Q} \neq \mathbf{I}^{m \times m}$ — orthonormal in \mathbf{R}^m subspace but not complete in total \mathbf{R}^n space

Krylov Subspace

- Krylov subspace S_m is spanned by a Krylov matrix, $K^m(\mathbf{f}) = [\mathbf{f} \ A\mathbf{f} \ \dots \ A^{m-1}\mathbf{f}]$ ($\mathbf{f} \in \mathbb{R}^n$)

Theorem

Let Q_m be the orthonormal basis obtained by QR factorization, $K_m(\mathbf{f}) = Q_m R$, then $T_m = Q_m^T A Q_m$ is a tridiagonal matrix

Proof (see [supplementary note](#))

For $i > j+1$, $\mathbf{q}_i^T(A\mathbf{q}_j) = 0$, since $A\mathbf{q}_j \in S_{j+1}$ by construction & $\mathbf{q}_i \perp S_{j+1}$ by Gram-Schmidt orthonormalization for $i > j+1$. By the symmetry of A , $\mathbf{q}_i^T(A\mathbf{q}_j) = \mathbf{q}_j^T(A^T\mathbf{q}_i) = \mathbf{q}_j^T(A\mathbf{q}_i) = 0$ for $j > i+1$ or $i < j-1$.

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-2} & \alpha_{m-1} & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix} \quad \begin{cases} \alpha_j = \mathbf{q}_j^T A \mathbf{q}_j & j = 1, \dots, m \\ \beta_j = \mathbf{q}_{j+1}^T A \mathbf{q}_j & j = 1, \dots, m-1 \end{cases}$$

- Tridiagonal matrix can be diagonalized in $O(N)$ time
*cf. `tqli()` in *Numerical Recipes**

Alexei Krylov with daughter Anna,
later Anna Kapitsa, wife of Pyotr
Kapitsa (1904)



Recursion Formula

- Due to the tridiagonality, $A\mathbf{q}_i$ is a linear combination of \mathbf{q}_{i-1} , \mathbf{q}_i & \mathbf{q}_{i+1} :

$$A\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i\mathbf{q}_{i+1} \quad 2 \leq i \leq m-1$$

If we define $\mathbf{q}_0 = 0$, the above equation is valid for $i = 1$ as well. Let $\mathbf{r}_i \equiv \beta_i\mathbf{q}_{i+1}$ (\mathbf{r}_i is a component of $A\mathbf{q}_i$ orthogonal to \mathbf{q}_j for $j \leq i$), then

$$\mathbf{r}_i = A\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1} - \alpha_i\mathbf{q}_i \quad 1 \leq i \leq m-1$$

$$A\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i\mathbf{q}_{i+1} \quad 2 \leq i \leq m-1$$

- **Lanczos algorithm:**

Given $\mathbf{r}_0, \beta_0 = \|\mathbf{r}_0\|$ ($\mathbf{q}_0 = 0$)

for $i = 1, \dots, m$

$$\mathbf{q}_i \leftarrow \mathbf{r}_{i-1} / \beta_{i-1}$$

$$\mathbf{r}_i \leftarrow A\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}$$

$$\alpha_i \leftarrow \mathbf{q}_i^T \mathbf{r}_i \quad \because \mathbf{q}_i^T (A\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}) = \mathbf{q}_i^T A\mathbf{q}_i = \alpha_i \text{ (orthogonality)}$$

$$\mathbf{r}_i \leftarrow \mathbf{r}_i - \alpha_i\mathbf{q}_i$$

$$\beta_i = \|\mathbf{r}_i\| \text{ (only when } i \leq m-1) \quad \beta_i = \mathbf{q}_{i+1}^T A\mathbf{q}_i$$

endfor

Keep increasing m until $\lambda_1[\mathbf{T}_m]$ converges

Application of Rayleigh-Ritz/Lanczos

- Search for transition states (with a negative eigenvalue of the Hessian matrix, $\partial^2 E / \partial r_i \partial r_j$) by following the eigenvector with the smallest eigenvalue
 - **Rayleigh-Ritz:** Kumeda, Wales & Munro, *Chem. Phys. Lett.* **341**, 185 ('01)
 - **Lanczos:** Mousseau *et al.*, *J. Mol. Graph. Model.* **19**, 78 ('01)

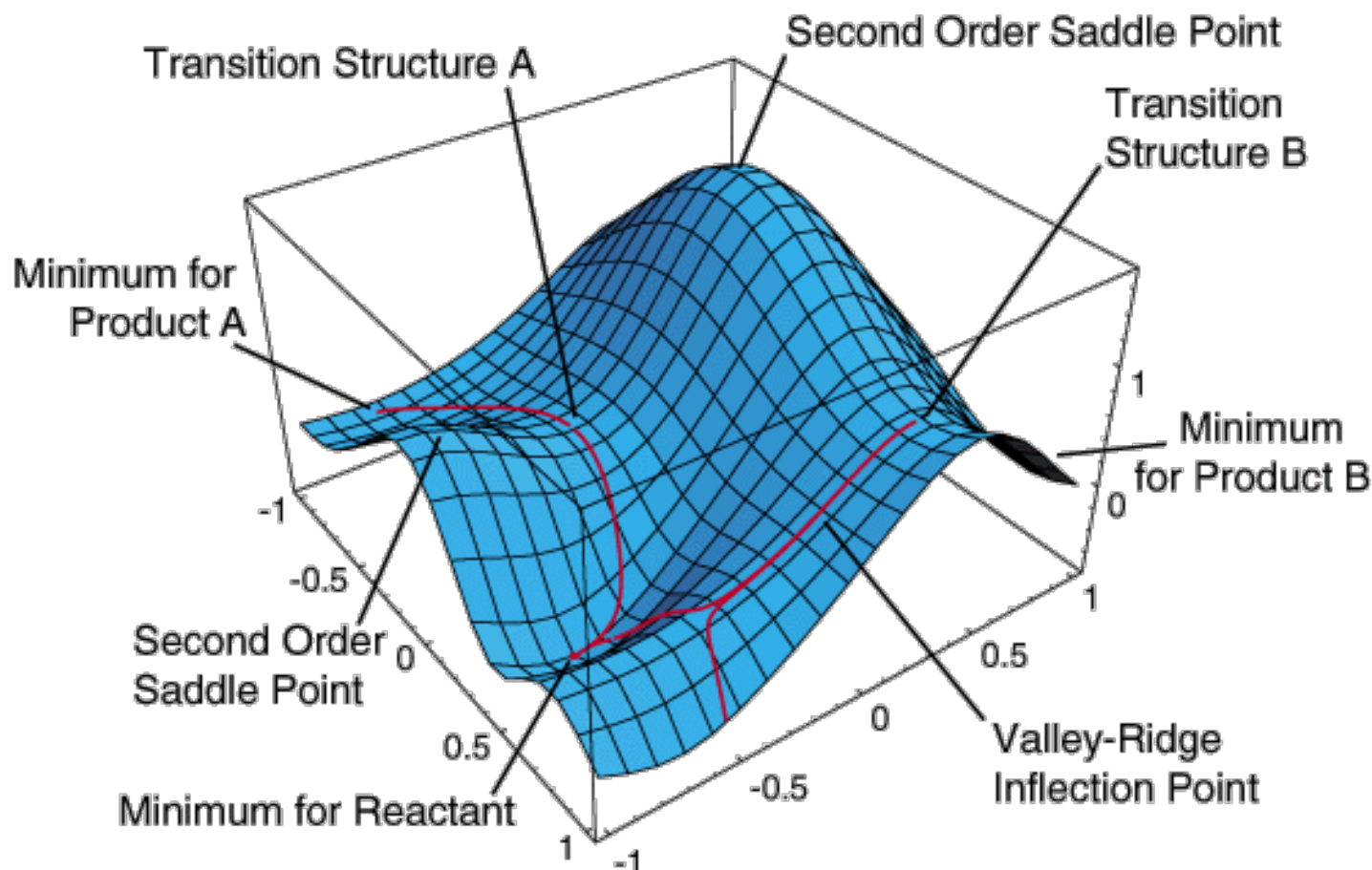


Figure from Prof. H. B. Schlegel; <http://chem.wayne.edu/schlegel>

Lanczos Algorithm for Hessian Calculation

A. Nakano / Computer Physics Communications 176 (2007) 292–299

Algorithm Lanczos

Input:

$\mathbf{R} \in \mathbb{R}^{3N}$: a state

logical *initialize*: TRUE for the first call in each event generation; FALSE otherwise

Output:

λ_1 : the minimum eigenvalue of the Hessian matrix, $\mathbf{H}(\mathbf{R}) = \partial^2 V / \partial \mathbf{R}^2$

$\mathbf{V}^1 \in \mathbb{R}^{3N}$: the Hessian eigenvector corresponding to λ_1

Steps:

if *initialize*

randomize $\Delta \in \mathbb{R}^{3N}$, such that it contains no translational motion

$s \leftarrow 0$

$\beta^s \leftarrow \|\Delta\|$

$\mathbf{Q}^s (\in \mathbb{R}^{3N}) \leftarrow 0$

do

$s \leftarrow s + 1$

$\mathbf{Q}^s \leftarrow \Delta / \beta^{s-1}$

$c_{fd} \leftarrow \max_{i\alpha} \{|q_{i\alpha}^s| \mid i = 1, \dots, N; \alpha = x, y, z\} / \delta_{fd}$

$\Delta \leftarrow c_{fd} [-\mathbf{F}(\mathbf{R} + \mathbf{Q}^s / c_{fd}) + \mathbf{F}(\mathbf{R})] - \beta^{s-1} \mathbf{Q}^{s-1}$

$\alpha^s \leftarrow \mathbf{Q}^{sT} \Delta$

$\Delta \leftarrow \Delta - \alpha^s \mathbf{Q}^s$

$\beta^s \leftarrow \|\Delta\|$

$$\text{diagonalize } \mathbf{T}_s = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{s-2} & \alpha_{s-1} & \beta_{s-1} \\ & & & \beta_{s-1} & \alpha_s \end{bmatrix}, \quad \text{so that } \tilde{\mathbf{Q}}_s^T \mathbf{T}_s \tilde{\mathbf{Q}}_s = \text{diag}(\tilde{\lambda}_1^s, \dots, \tilde{\lambda}_s^s)^*$$

while $|(\tilde{\lambda}_1^s - \tilde{\lambda}_1^{s-1}) / \tilde{\lambda}_1^{s-1}| > \Delta_{\text{eigen}}$

$\lambda_1 \leftarrow \tilde{\lambda}_1^s$

$\mathbf{V}^1 \leftarrow \sum_{k=1}^s \mathbf{Q}^k \tilde{\mathbf{q}}_k^1$

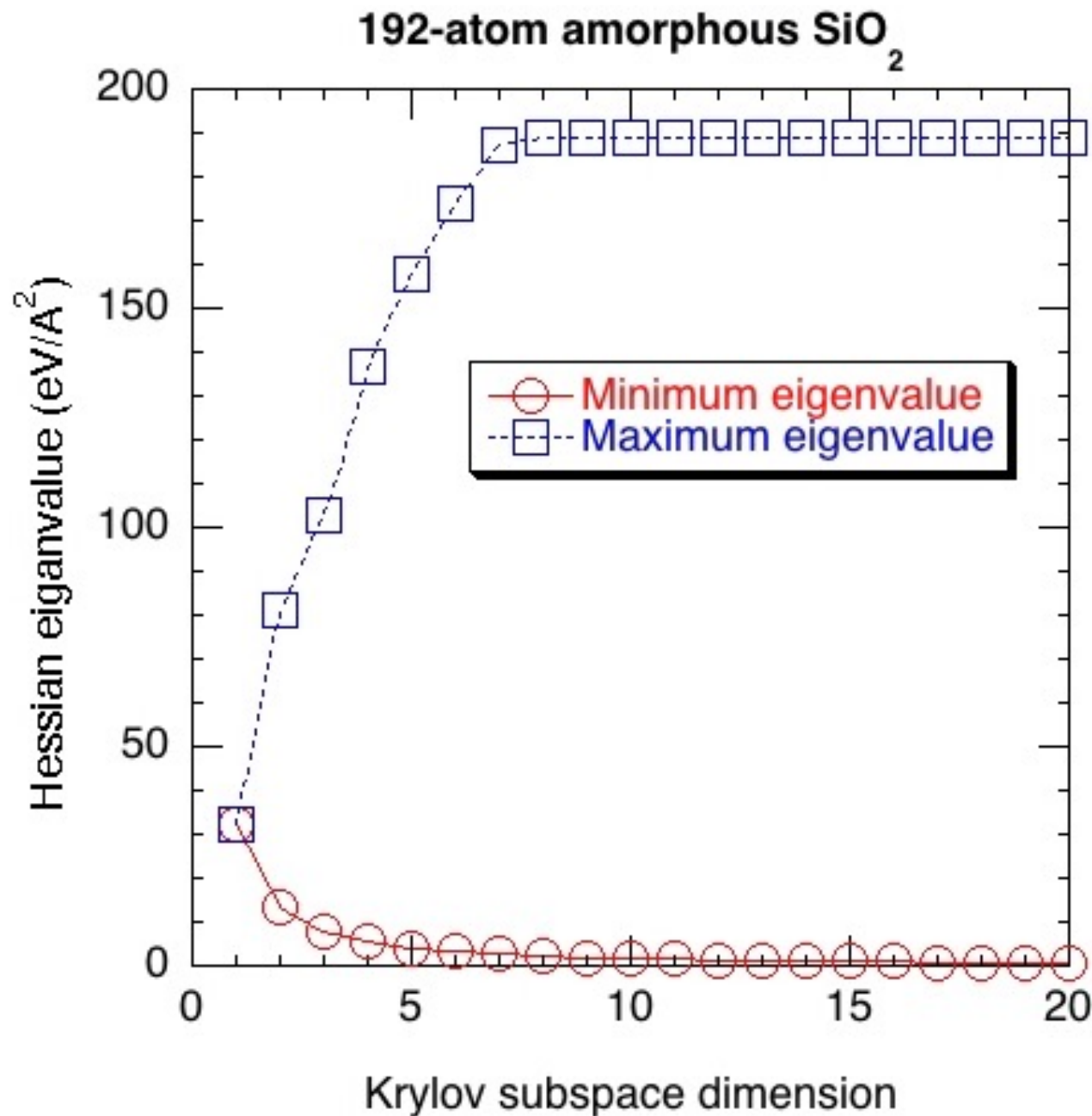
$\mathbf{V}^1 \leftarrow \mathbf{V}^1 / \|\mathbf{V}^1\|$

$$\begin{aligned} \vec{F}(\vec{R} + \vec{Q}) &= \vec{F}(\vec{R}) + \overbrace{\partial \vec{F} / \partial \vec{R}}^{-\vec{H}(\vec{R})} \cdot \vec{Q} \\ \therefore \vec{H}(\vec{R}) \cdot \vec{Q} &= -\vec{F}(\vec{R} + \vec{Q}) + \vec{F}(\vec{R}) \end{aligned}$$

tqli() — $O(N)$

* $\text{diag}(\tilde{\lambda}_1^s, \dots, \tilde{\lambda}_s^s)$ is an s by s diagonal matrix, with its diagonal elements given by $\tilde{\lambda}_1^s, \dots, \tilde{\lambda}_s^s$. $\tilde{\mathbf{Q}}^s = [\tilde{\mathbf{q}}^1, \dots, \tilde{\mathbf{q}}^s]$ is an s by s orthogonal matrix, with $\tilde{\mathbf{q}}^m \in \mathbb{R}^s$ is the m th eigenvector of \mathbf{T}_s .

Sample Run of Lanczos Program



Electronic Energy Bands of GaAs

• 8-band $k \cdot p$ model

$$H_k = \begin{pmatrix} A & 0 & V^* & 0 & \sqrt{3}V & -\sqrt{2}U & -U & \sqrt{2}V^* \\ 0 & A & -\sqrt{2}U & -\sqrt{3}V^* & 0 & -V & \sqrt{2}V & U \\ V & -\sqrt{2}U & -P+Q & -S^* & R & 0 & \sqrt{\frac{3}{2}}S & -\sqrt{2}Q \\ 0 & -\sqrt{3}V & -S & -P-Q & 0 & R & -\sqrt{2}R & \frac{1}{\sqrt{2}}S \\ \sqrt{3}V^* & 0 & R^* & 0 & -P-Q & S^* & \frac{1}{\sqrt{2}}S^* & \sqrt{2}R^* \\ -\sqrt{2}U & -V^* & 0 & R^* & S & -P+Q & \sqrt{2}Q & \sqrt{\frac{3}{2}}S^* \\ -U & \sqrt{2}V^* & \sqrt{\frac{3}{2}}S^* & -\sqrt{2}R^* & \frac{1}{\sqrt{2}}S & \sqrt{2}Q & -P-\Delta & 0 \\ \sqrt{2}V & U & -\sqrt{2}Q & \frac{1}{\sqrt{2}}S^* & \sqrt{2}R & \sqrt{\frac{3}{2}}S & 0 & -P-\Delta \end{pmatrix}$$

$$A = E_c - \frac{\hbar^2}{2m_0}(\partial_x^2 + \partial_y^2 + \partial_z^2),$$

$$P = -E_v - \gamma_1 \frac{\hbar^2}{2m_0}(\partial_x^2 + \partial_y^2 + \partial_z^2),$$

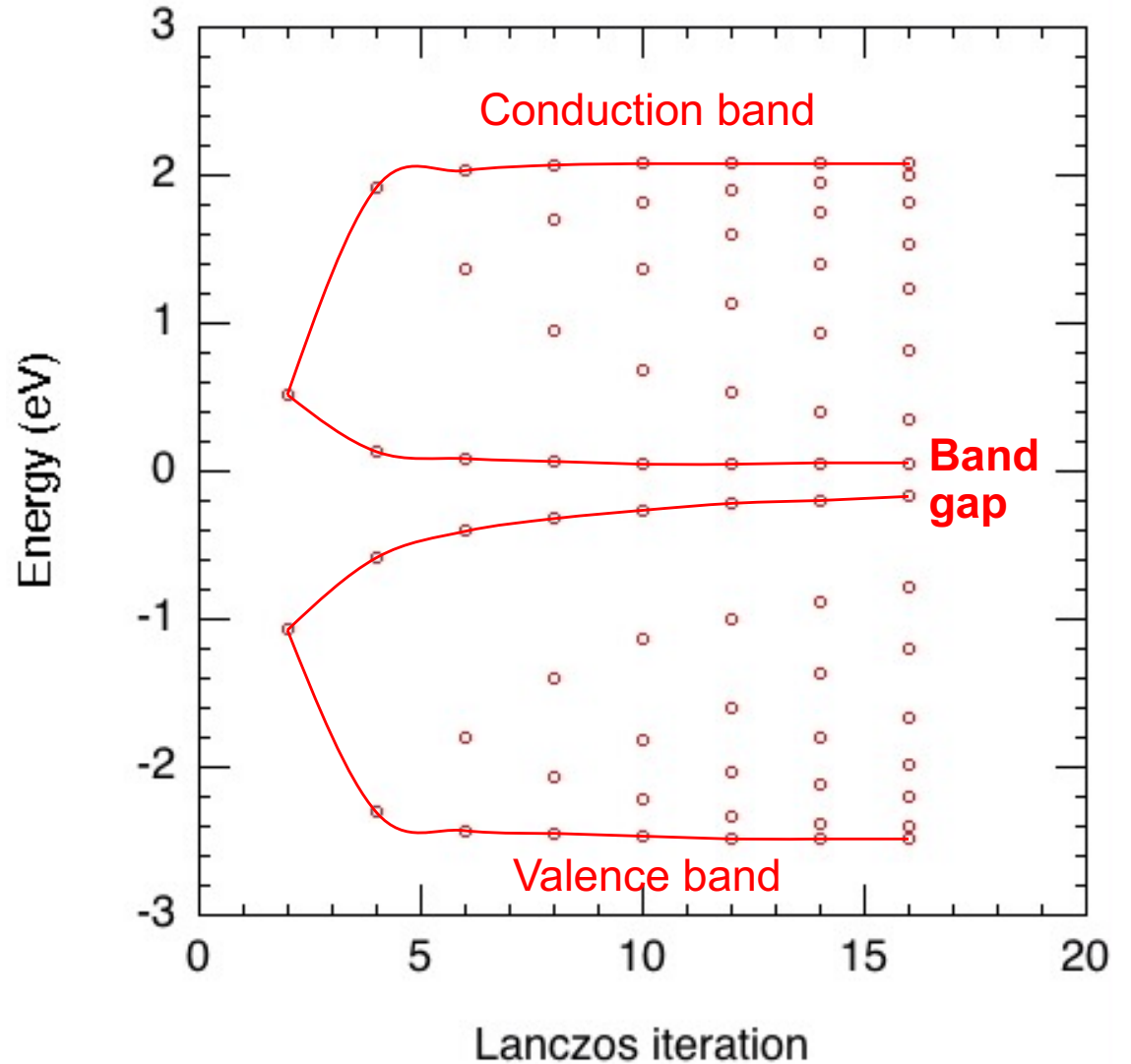
$$Q = -\gamma_2 \frac{\hbar^2}{2m_0}(\partial_x^2 + \partial_y^2 - 2\partial_z^2),$$

$$R = \sqrt{3} \frac{\hbar^2}{2m_0}[\gamma_2(\partial_x^2 - \partial_y^2) - 2i\gamma_3\partial_x\partial_y],$$

$$S = -\sqrt{3}\gamma_3 \frac{\hbar^2}{m_0} \partial_z(\partial_x - i\partial_y),$$

$$U = \frac{-i}{\sqrt{3}} P_0 \partial_z,$$

$$V = \frac{-i}{\sqrt{6}} P_0(\partial_x - i\partial_y).$$



C. Pryor, *Phys. Rev. B* **57**, 7190 ('98)

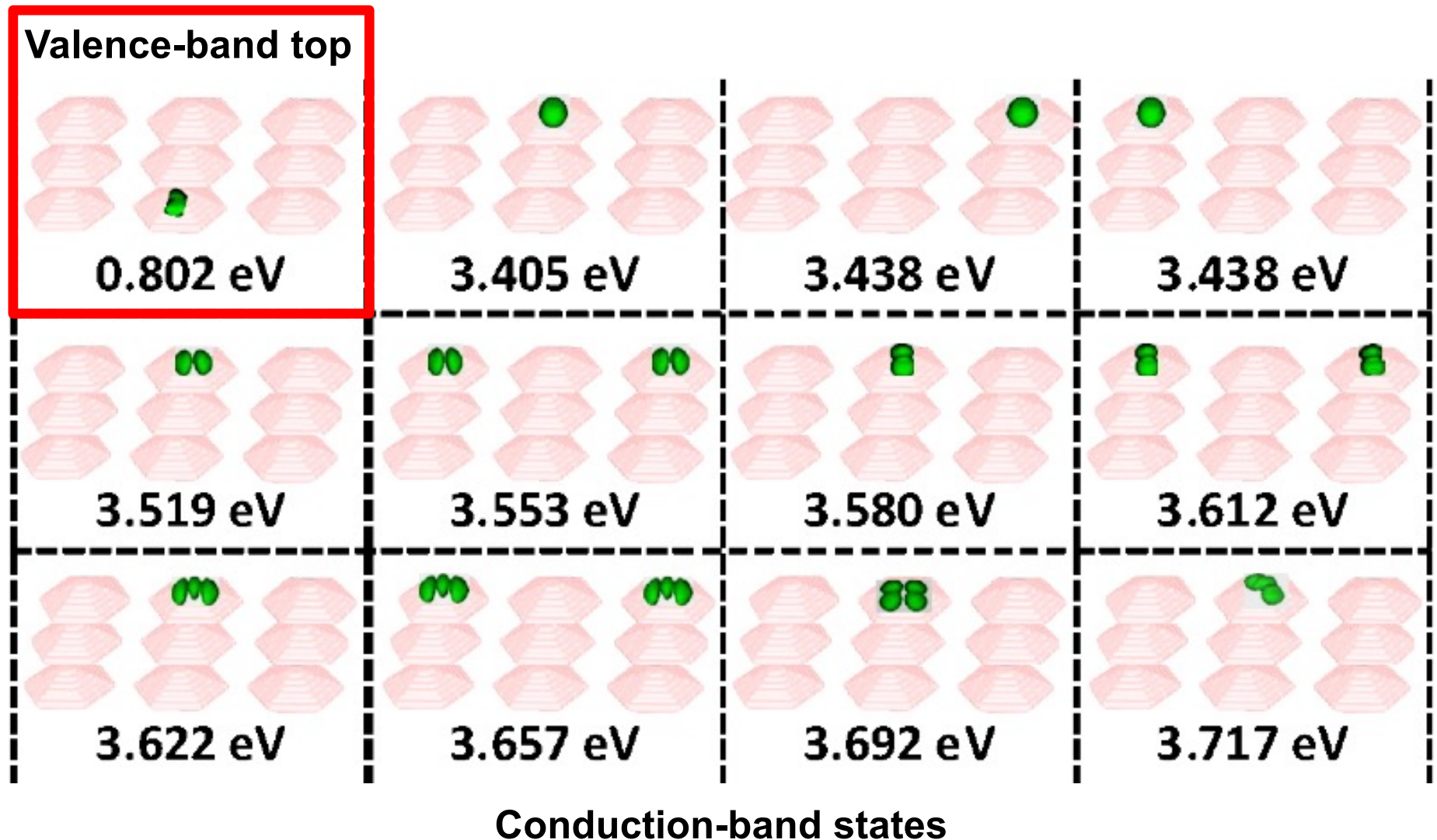
Lanczos Program in Fortran

```
do s = 1,NWF
  q(:, :, :, s) = v/bet(s-1)
  call hamiltonian_op(q(:, :, :, s), hv)  ! Operates Hamiltonian H on Q(S)
  v = hv - bet(s-1)*q(:, :, :, s-1)
  alp(s) = inner_product(q(:, :, :, s), v)
  v = v - alp(s)*q(:, :, :, s)
  bet(s) = sqrt(inner_product(v, v))
  call tridiag(eval, s)  ! Diagonalize the S by S tridiagonal matrix
end do  ! Lanczos iteration over s
```

Given $\mathbf{r}_0, \beta_0 = \|\mathbf{r}_0\|$ ($\mathbf{q}_0 = 0$)
for $i = 1, \dots, m$
 $\mathbf{q}_i \leftarrow \mathbf{r}_{i-1} / \beta_{i-1}$
 $\mathbf{r}_i \leftarrow \mathbf{A}\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}$
 $\alpha_i \leftarrow \mathbf{q}_i^T \mathbf{r}_i$
 $\mathbf{r}_i \leftarrow \mathbf{r}_i - \alpha_i \mathbf{q}_i$
 $\beta_i = \|\mathbf{r}_i\|$ (only when $i \leq m - 1$)
endfor

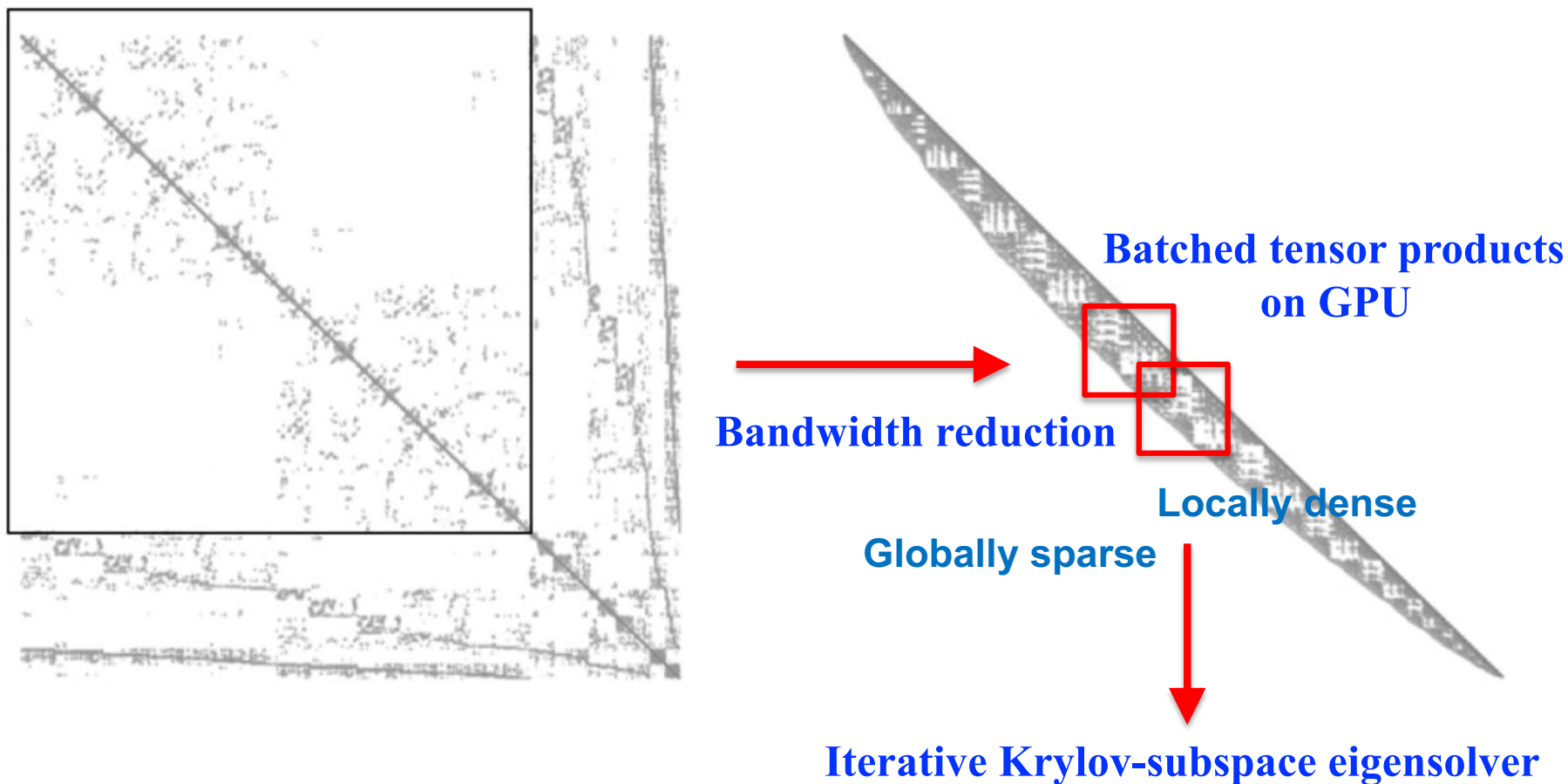
Band-edge Wave Functions

- Band-edge states in an array of GaN quantum dots in AlN matrix



S. Sburlan, Ph.D. dissertation, USC ('13)

Globally-Sparse Yet Locally-Dense Eigensolver



- 250-fold speed-up over state-of-the-art for 2.4M atom molecular vibrational modes

Top 10 Algorithms in History

In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

PHYS 516
CSCI 596
CSCI 653

IEEE CiSE, Jan/Feb (2000)