

# Subspace Projected Approximate Matrix The SPAM Method

M. Minkoff, R. Shepard, A. Wagner

- ◆ Outline
- ◆ Background on SPAM Approach
- ◆ Results
- ◆ Cumulative Reaction Probabilities
- ◆ Initial Results

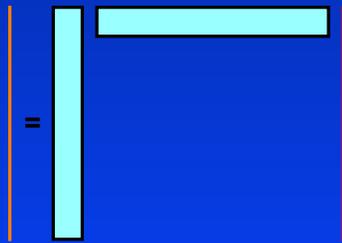
# Subspace Projected Approximate Matrix: SPAM Method

- Subspace iterative solution to eigenvalue, linear, and nonlinear problems  
e.g. eigenvalue problem  $(H - \lambda_j) v_j = 0$
- Subspace iterative solutions have form  $v_j = X^n c_j$   
where:  $X^n = x^1, x^2, \dots, x^n$
- $c_j$  is solved in a subspace:  $H^n c_j = \lambda_j^n c_j$   
where:  $H^n = (W^n)^T X^n$   
where:  $W^n = H X^n$  <---for  $N \gg n$ , where all the work is
- New  $x^{n+1}$  vector from residual:  $r^{n+1} = (W^n - \lambda_j^n X^n) c_j$
- SPAM gives more accurate or faster converging way to get  $r^{n+1}$  in 3 steps

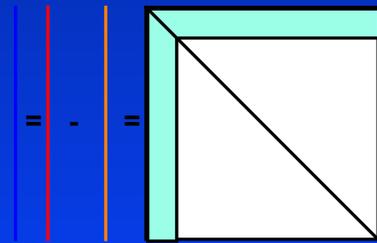
# Subspace Projected Approximate Matrix: SPAM Method

Step 1: Assemble and apply Projections Operators on current vector

$$P^n r^{n+1} (=X^n[X^n]^T r^{n+1})$$



$$Q^n r^{n+1} (= r^{n+1} - P^n r^{n+1})$$



where

n trial vectors  
(already processed)  
 $(X^n)$



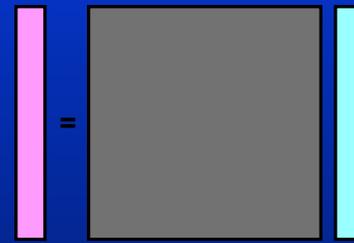
n+1 trial vector  
 $r^{n+1}$



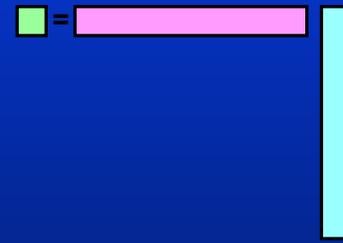
exact  
matrix



matrix-vector  
product



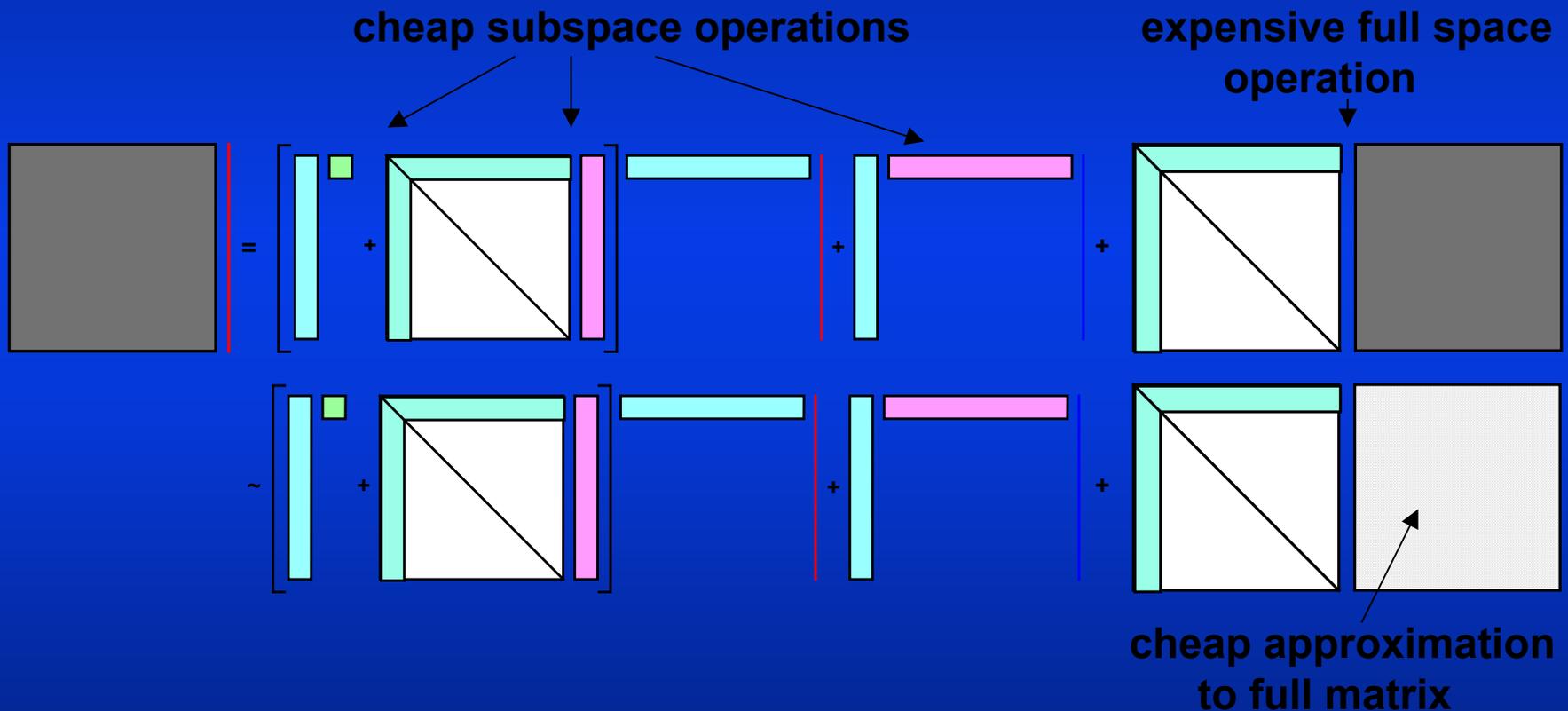
subspace  
matrix



# Subspace Projected Approximate Matrix: SPAM Method

Step 2: Decomposition and Approximation of  $H \mathbf{r}^{n+1}$

$$H\mathbf{v} = (P^n+Q^n) H (P^n+Q^n) \mathbf{r}^{n+1}$$



Step 3: Solve approximate subspace problem  
 - solution is  $\mathbf{r}^{n+1}$

CMCIM02

# Subspace Projected Approximate Matrix: SPAM Method

## SPAM properties (broad view):

- ◆ **Relation to other subspace methods (e.g., Davidson)**
  - More flexible (sequence of approx. matrices - no sequence => SPAM= Davidson)
  - If iteration tolerances are correct, always no worse than Davidson
- ◆ **Relation to multigrid methods**
  - SPAM sequences of approx. matrices ~ multigrid sequences of approx. grids
  - Subspace method => solution vector composed of multiple vectors  
Multigrid methods => single solution vector that is updated
- ◆ **Relation to preconditioned conjugate gradient (PCG) methods**
  - SPAM has multiple approximations always improving by projection  
PCG has a fixed single preconditioner
- ◆ **Deep injection of physical insight into numerics**
  - Application experts can design physical approximation sequences
  - SPAM maps approximation sequences onto numerics sequences
  - Projection operators continually improve the approximations  
=> coarse approximations can still be numerically useful

## Outline of the Davidson Method

Generate an initial vector  $\mathbf{x}_1$

MAINLOOP: DO  $n = 1$

Compute and save  $\mathbf{w}_n = \mathbf{H}\mathbf{x}_n$

Compute the  $n$ -th row and column of  $\langle \mathbf{H} \rangle$ :

$$\langle \mathbf{H} \rangle_{1:n,n} = \mathbf{w}_n^T \mathbf{X}^{[n]}$$

Compute the subspace eigenvector and value:  $(\langle \mathbf{H} \rangle - \rho)\mathbf{c} = \mathbf{0}$

Compute the residual:  $\mathbf{r} = \mathbf{W}_{1:n} \mathbf{c}_{1:n} - \rho \mathbf{X}_{1:n} \mathbf{c}_{1:n}$

Check for convergence using  $|\mathbf{r}|$ ,  $\mathbf{c}$ ,  $\rho$ , etc.

IF (*converged*) THEN

EXIT MAINLOOP

ELSE

Generate a new expansion vector  $\mathbf{x}_{n+1}$  from  $\mathbf{r}$ ,  $\rho$ ,  $\mathbf{v} = \mathbf{X}\mathbf{c}$ , etc.

ENDIF

ENDDO MAINLOOP

## Outline of the SPAM Method

Generate an initial vector  $\mathbf{x}_1$

Set  $wtype_1 = 1$  ! *Start the iterations with approximate products*

Set  $n_0 = 0$ ;  $n = 1$

MAINLOOP: DO

Compute and save  $\mathbf{w}_n = \mathbf{H}(wtype_n, n_0) \mathbf{x}_n$

Compute the  $n$ -th row and column of  $\langle \mathbf{H} \rangle$ :  $\langle \mathbf{H} \rangle_{1:n,n} = \mathbf{w}_n^T \mathbf{X}^{[n]}$

Compute the subspace eigenvector and value:  $(\langle \mathbf{H} \rangle - \rho)\mathbf{c} = \mathbf{0}$

Compute the residual:  $\mathbf{r} = \mathbf{W}_{1:n} \mathbf{c}_{1:n} - \rho \mathbf{X}_{1:n} \mathbf{c}_{1:n}$

Check for convergence using  $|\mathbf{r}|$ ,  $\mathbf{c}$ ,  $\rho$ , etc.

IF (*converged* .AND.  $wtype_n = 0$ ) then

EXIT MAINLOOP ! *Final convergence is achieved*

ELSEIF (*converged* .AND.  $wtype_n \neq 0$ ) then

Contract

Set  $n = n_0 + 1$ ;  $n_0 = n_1$

Set  $wtype_n = 0$  ! *The next product will be exact*

ELSE

Set  $n \leftarrow n + 1$

Generate a new expansion vector  $\mathbf{x}_n$  from  $\mathbf{r}$ ,  $\rho$ ,  $\mathbf{v} = \mathbf{X}\mathbf{c}$ , etc.

Set  $wtype_n = 1$  ! *The next product will be approximate*

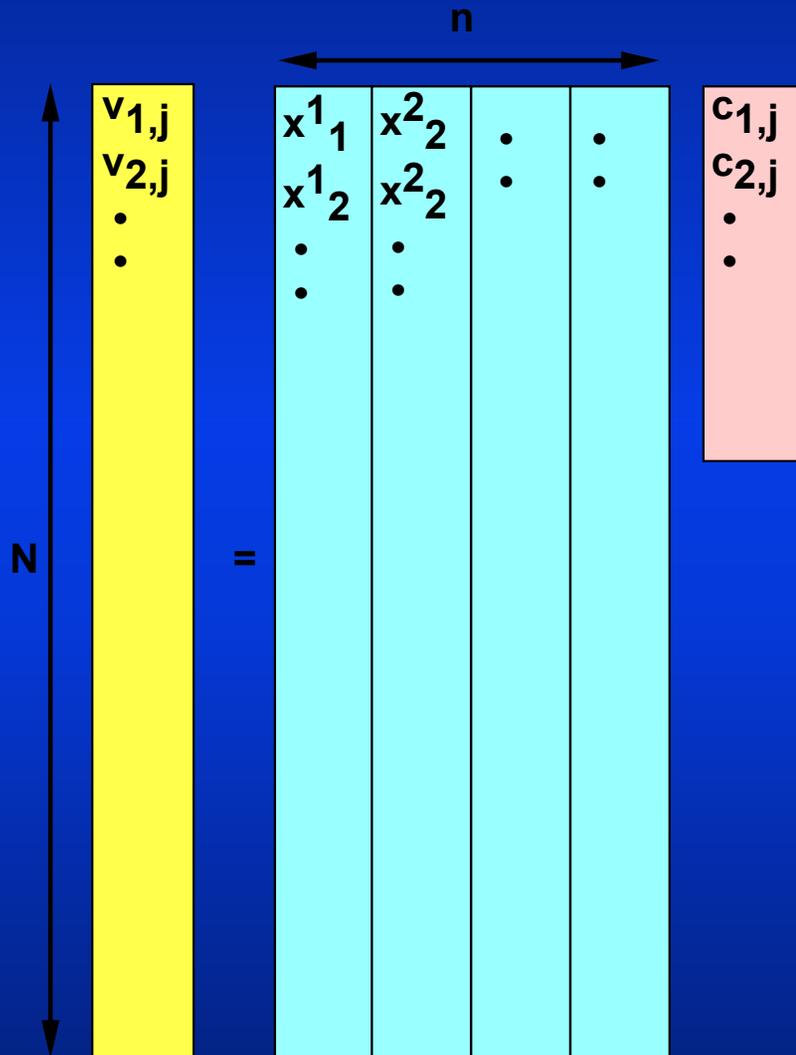
ENDIF

ENDDO MAINLOOP

# Subspace iterative solution to eigenvalue problem:

$$(H - \lambda_j) v_j = 0$$

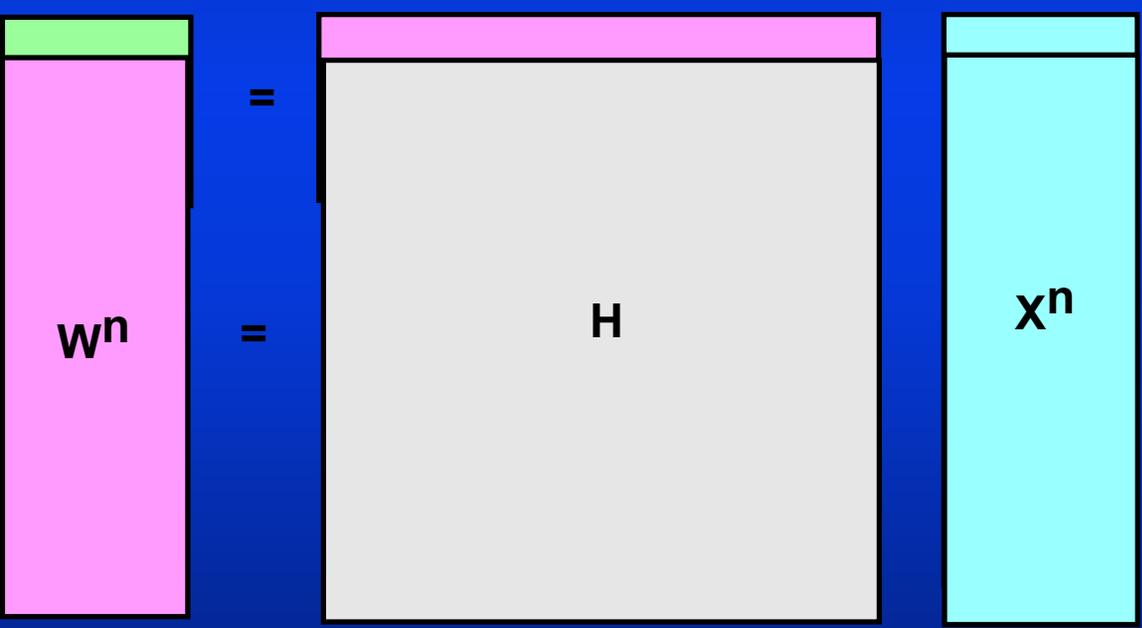
subspace iterative solutions have form  $v_j = X_n c_j$



where:  $X_n = x_1, x_2, \dots, x_n$   
 $c_j$  is solved in a subspace:  $H_n c_j = \lambda_{nj} c_j$

That subspace  $H_n$  is of the form:

where:



For  $N \gg n$ , work/iteration =  $H \times n$

New  $x$  vector from residual:  $x_{n+1} = (W_n - \lambda_n j X_n) c_j$

SPAM gives better way to get  $x_{n+1}$  via 4 steps:

- Define Projection Operators P and Q:

$$P_n = X_n ((X_n)^T X_n)^{-1} (X_n)^T$$

=  $X_n (X_n)^T$  with orthonormal basis vectors,

$$Q_n = (I - P_n)$$

$P v$  = the  $X_n$  components of  $v$

$Q v$  = the part of  $v$  that is not spanned by  $X_n$

- Decomposition of H with P and Q:

$$H = (P_n + Q_n) H (P_n + Q_n)$$
$$= P_n H P_n + P_n H Q_n + Q_n H P_n + Q_n H Q_n$$
$$= \underbrace{X_n H_n (X_n)^T}_{\text{blue}} + \underbrace{X_n (W_n)^T Q_n + Q_n W_n (X_n)^T}_{\text{red}} + \underbrace{Q_n H Q_n}_{\text{red}}$$

$\Rightarrow$  cheap subspace operations

$\Rightarrow$  expensive matrix vector multiply

- Approximate  $Q_n H Q_n$  with cheap approximate  $H(1)$ :

$$H_1 = X_n H_n (X_n)^T + X_n (W_n)^T Q_n + Q_n W_n (X_n)^T + Q_n H(1) Q_n$$

- Iteratively solve for  $x_{n+1}$ :

$$x_{n+1} = w_j \quad \text{where } (H_1 - \epsilon_j) w_j = 0$$

CMCIM02

## What might $H(1)$ be?

Anything to make **matrix vector multiplies cheaper**

- Sparser
- Smaller underlying basis
- Lower-order expansion of matrix elements
- Coarser underlying grid
- Lower-order difference equation
- Tensor-product approximation
- Operator approximation

## SPAM has great properties:

- projection operators  $\Rightarrow$  convergence from any  $H(1)$
- multi-level SPAM with dynamic tolerances
  - $Q_n H Q_n$  approximated by  $Q_n H(1) Q_n$
  - $Q_n H(1) Q_n$  approximated by  $Q_n H(2) Q_n$
  - $Q_n H(2) Q_n$  approximated by  $Q_n H(3) Q_n \dots$
- tens of lines of additional code to existing iterative subspace eigensolver
- applicable to any subspace problem (linearsolves?)

## SPAM Example 1:

**Banded Matrix:**

**H characterized by:**

**N = matrix dimension**

**$\Delta$  = off diagonal "increment"**

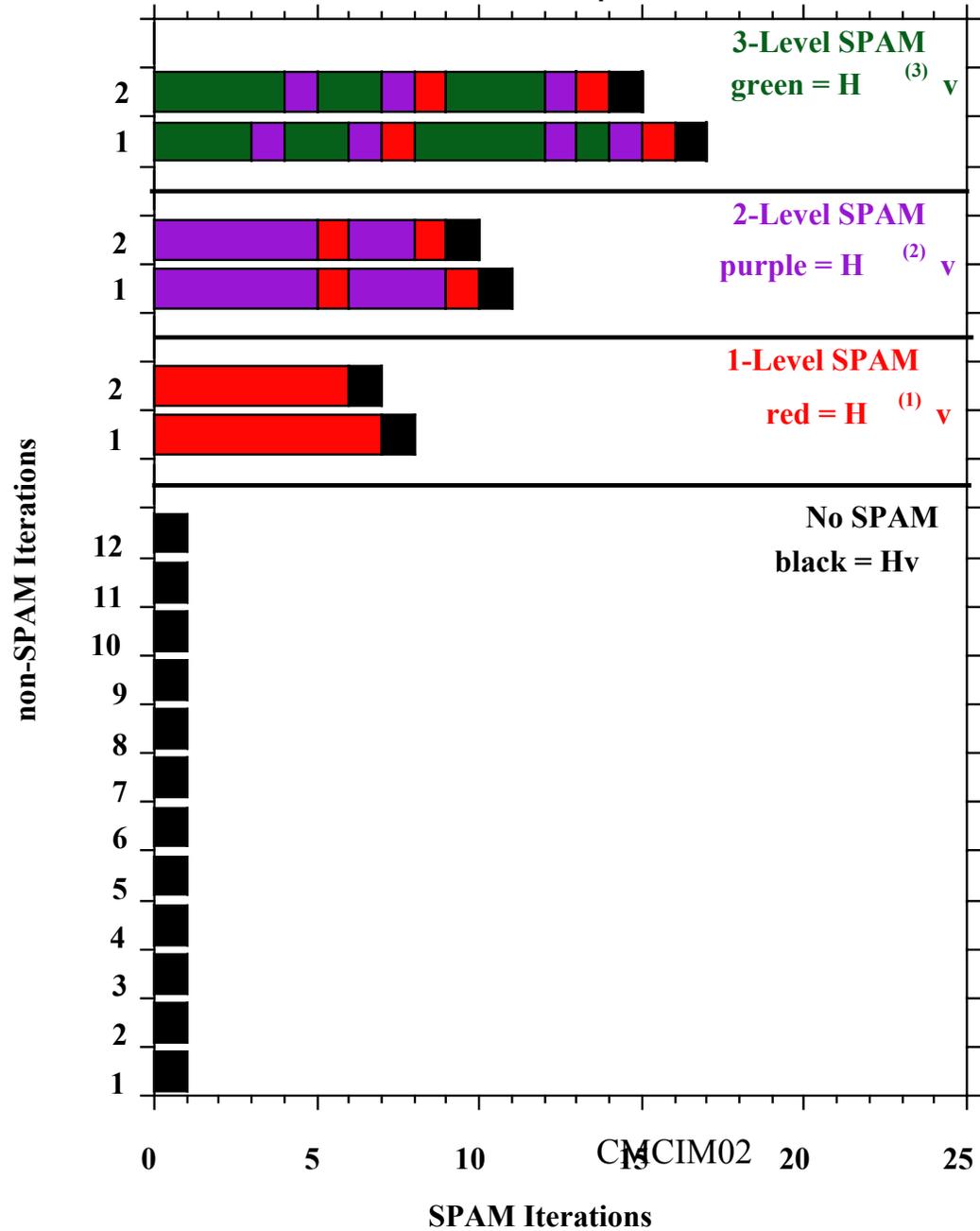
**$W_0$  = width of band**

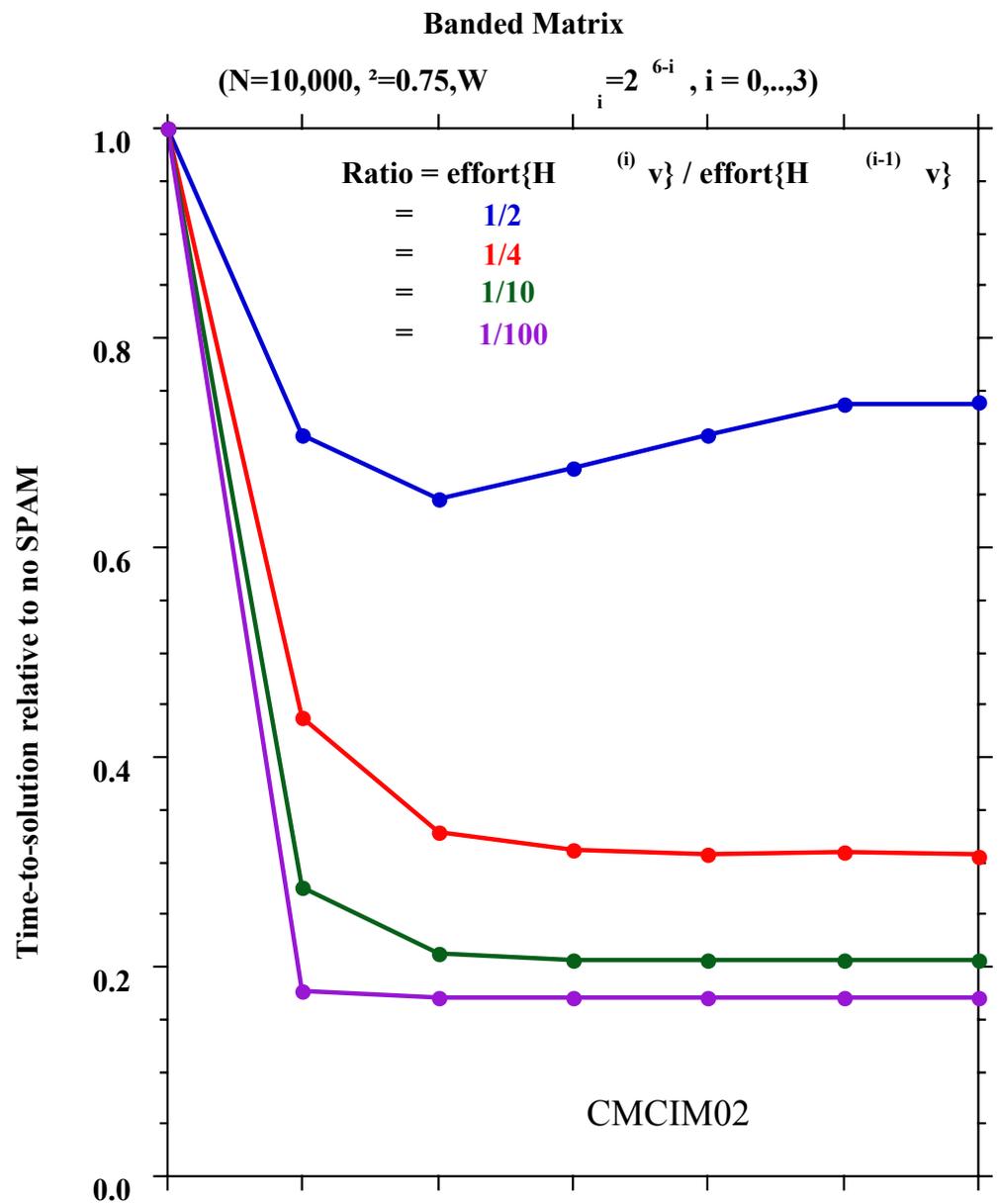
**H(i) has  $W_i < W_{i-1}$**

$$\begin{pmatrix} 1 & \Delta & \Delta^2 & 0 \\ \Delta & 2 & \Delta & 0 \\ \Delta^2 & \Delta & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

# Banded Matrix

( $N=10,000, \alpha=0.75, W_i=2^{6-i}, i=0,\dots,3$ )





### Tensor-Product Convergence Results for Multiple Eigenvectors

| Method                              | $m=8$     |               |                         | $m=10$    |               |                         |
|-------------------------------------|-----------|---------------|-------------------------|-----------|---------------|-------------------------|
|                                     | $n_m$     | $N_{product}$ | $Effort$                | $n_m$     | $N_{product}$ | $Effort$                |
| <i>DPR</i>                          | <i>ax</i> |               |                         | <i>ax</i> |               |                         |
| One vector at a time                | 40        | [203]         | 1.000                   | 26        | [145]         | 1.000                   |
| Simultaneous/lowest                 | 82        | [82]          | 1.000                   | 99        | [99]          | 1.000                   |
| Simultaneous/cycle                  | 94        | [94]          | 1.000                   | 94        | [94]          | 1.000                   |
| Simultaneous/largest $ \mathbf{r} $ | 95        | [95]          | 1.000                   | 93        | [93]          | 1.000                   |
| <i>SPAM+DPR</i>                     |           |               | $\mu=4.9 \cdot 10^{-4}$ |           |               | $\mu=1.0 \cdot 10^{-5}$ |
| One vector at a time                | 42        | [20,164<br>]  | 0.099                   | 26        | [20,13<br>2]  | 0.138                   |
| Simultaneous/lowest                 | 83        | [19,87]       | 0.232                   | 99        | [20,10<br>4]  | 0.202                   |
| Simultaneous/cycle                  | 86        | [19,90]       | 0.203                   | 94        | [20,10<br>1]  | 0.213                   |
| Simultaneous/largest $ \mathbf{r} $ | 95        | [19,99]       | 0.201                   | 94        | [20,99]       | 0.215                   |
| <i>SPAM+IIGD</i>                    |           |               | $\mu=2.0 \cdot 10^{-3}$ |           |               | $\mu=4.0 \cdot 10^{-5}$ |
| One vector at a time                | 11        | [20,20]       | 0.099                   | 11        | [20,20]       | 0.138                   |
| Simultaneous/lowest                 | 20        | [19,21]       | 0.232                   | 20        | [20,21]       | 0.202                   |
| Simultaneous/cycle                  | 20        | [19,21]       | 0.203                   | 20        | [20,21]       | 0.213                   |
| Simultaneous/largest $ \mathbf{r} $ | 20        | [19,21]       | 0.200                   | 20        | [20,21]       | 0.215                   |

Convergence summaries of the lowest 10 roots of the  $m=8$  and  $m=10$  perturbed-tensor-product matrices described in the text. The initial vectors in all cases are the eigenvectors of the tensor-product matrices, which were computed as tensor-products of the eigenvectors of the  $4 \times 4$  component matrices. The matrix-vector product counts are the totals for all 10 roots. For the  $m=8$  calculations,  $N=65,536$ ,  $\beta=10$ , and  $|\mathbf{r}_j| < 10^{-1}$ . For the  $m=10$  calculations,  $N=1,048,576$ ,  $\beta=100$ , and  $|\mathbf{r}_j| < 10^0$ .

# How Fast Do Molecules React?

## MPP Simulation of Exact Cumulative Reaction Probabilities

- Combustion modeling requires accurate determination of rate constants which are approximated by statistical simulation techniques. Exact calculations of rate constants can be obtained from Cumulative Reaction Probabilities (CRP), however this can be done for only a few degrees of freedom (DOF) problems. Initial computational results suggest that the statistical methods may increasingly differ with exact calculations as DOF increases. Larger problems (up to 10 DOF) can be addressed only via MPP calculations. (This work is a collaborative project in chemical kinetics modeling (ANL/CHM) and massively parallel simulation (ANL/MCS).
  - CRP simulation involves calculation of a few eigenvalues of the in an outer iteration with an inner iteration evaluation of a Greens function dealing the solution of two Hamiltonian (linear) systems.
  - PETSc provides the capability to develop rapid-prototyping for the MPP software development of CRP simulations and an software environment for developing novel preconditioners for these and other linear systems and iterative eigenvalue solvers.

# *Advanced Software for the Calculation of Thermochemistry, Kinetics and Dynamics*

## Parallelization of **C**umulative **R**eaction **P**robabilities

- ◆ **Cumulative Reaction Probabilities (CRP)**
  - computational core of reaction rate constants
  - exact computation computational intensive
  - approximate computation underlies all major reaction rate theories in use
- ⇒ efficient exact CRP code will
  - give exact rates (if the computed forces are accurate)
  - calibrate ubiquitous approximate rate methods
  - Time Independent (Miller and Manthe, 1994, and others)
- ◆ Highly parallel approaches to both methods being pursued

# Parallelization of Cumulative Reaction Probabilities

## Time Independent Approach

- $N(E) = \sum_k p_k(E, J)$

where  $p_k(E, J)$  = eigenvalues of Probability Operator:

$$P(E) = 4 \varepsilon_r^{1/2} (H+i\varepsilon-E)^{-1} \varepsilon_p (H-i\varepsilon-E)^{-1} \varepsilon_r^{1/2}$$

where  $i\varepsilon_x$  = absorbing diagonal potentials  
= imaginary potentials

H = hamiltonian (differential operator)

=> for realistic problems

size  $\sim 10^5 \times 10^5$  or much larger

number of eigenvalues < 100

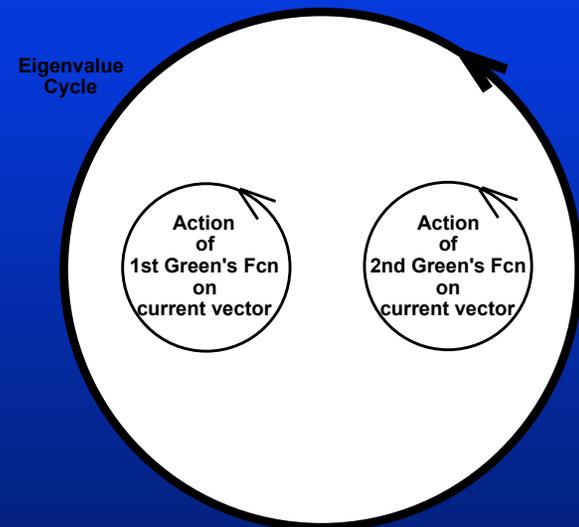
- iterative approach

macrocycle of iteration for eigenvalue

microcycle of iteration for

action of Green's function  $(H-i\varepsilon-E)^{-1}$

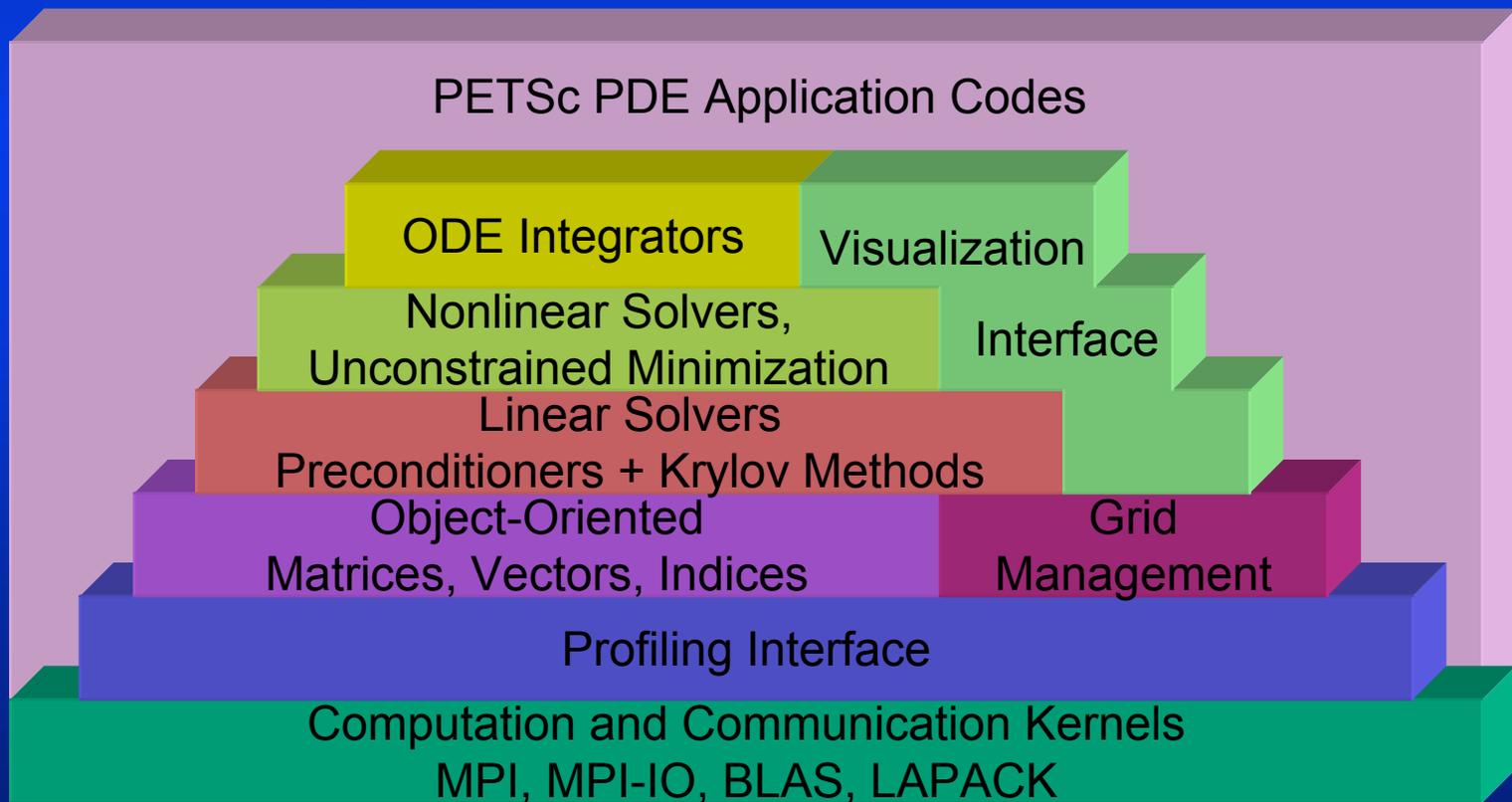
=> linear solve



# Parallelization of Cumulative Reaction Probabilities

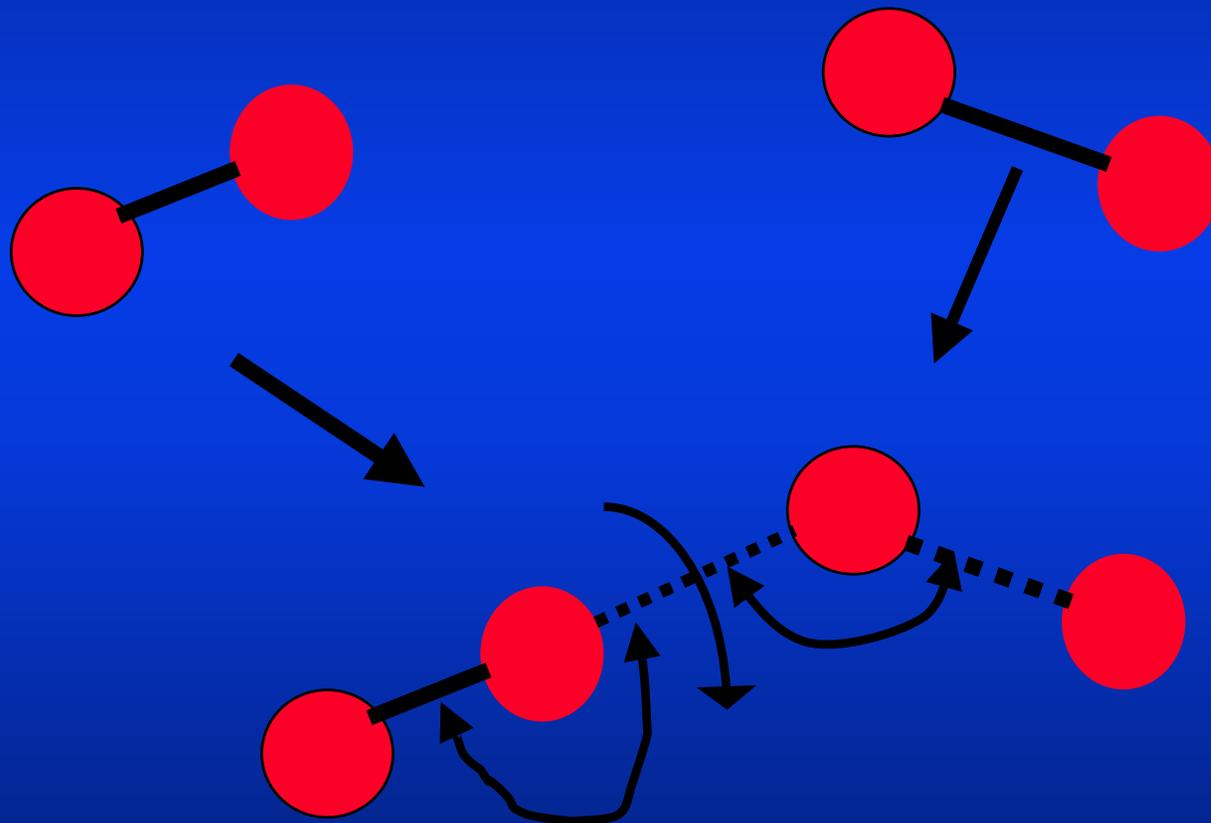
## Time Independent Approach

- Code built on PETSc (<http://www.mcs.anl.gov/petsc>) -> TOPS
- PETSc: data structure, GMRES linear solve, preconditioners  
USER: Lanczos method for eigensolve
- Future: user supplied preconditioners



# Chemical Dynamics Theory

3 angles, 3 stretches  
6 degrees of freedom



CMCIM02

# Chemical Dynamics Theory

## ◆ Probability Operator and It's Inverse

- Using probability method calculates a few large eigenvalues via iterative methods. The iterative evaluation involves the action of two Green's function.
- Using inverse probability method involves a direct calculation each iteration to obtain a few smallest eigenvalues. At each iteration the action of a vector by the Green's function is required. This leads to solving linear systems involving the Hamiltonian.

# Parallelization of Cumulative Reaction Probabilities

## Time Independent Approach

### Performance:

#### -model problem

- Optional number of dimensions
  - Eckhart potential along rxn coord.
  - parabolic potential perpendicular to reaction coord.
- DVR representation of H

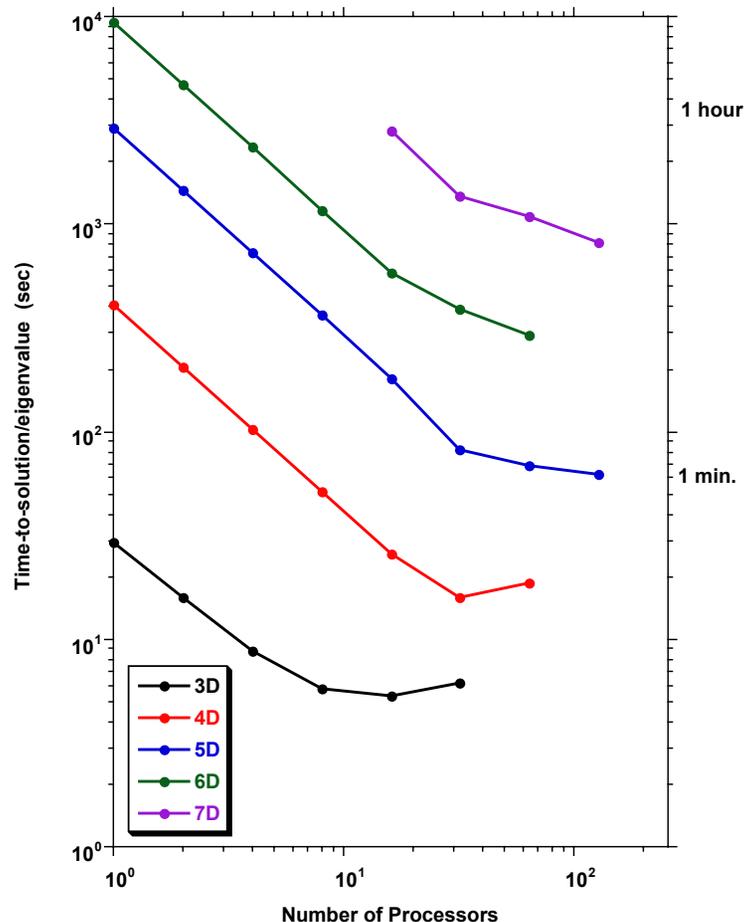
#### -Algorithm options

- Diagonal preconditioner
- other PETSc preconditioners slower **for this problem**

#### -Computers

- NERSC SP
- others include SGI Power Challenge, Cray T3E

Performance vs. processors for model with increasing dimensions

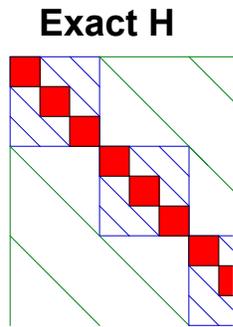


# Parallelization of Cumulative Reaction Probabilities

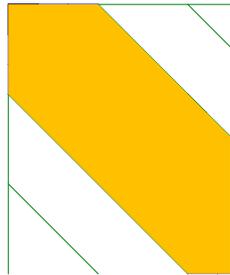
## Time Independent Approach

### Future Preconditioners:

- SPAM (see next poster)

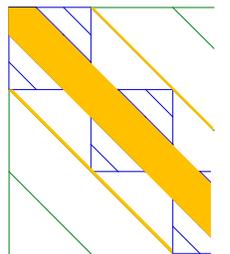


fat band  
precond.

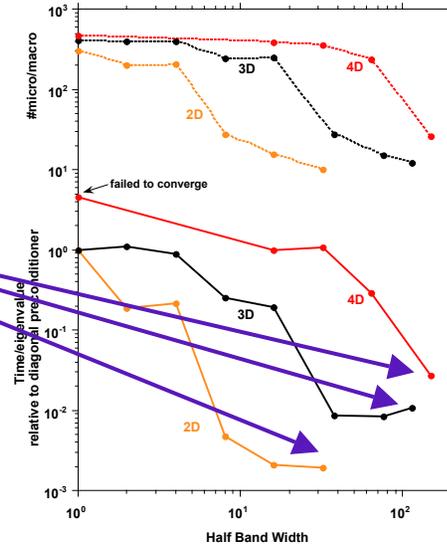


great  
performance

SPAM  
precond.

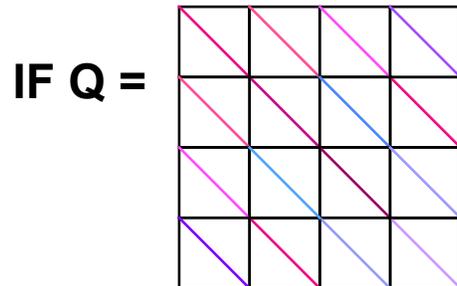


can SPAM get scalable performance?

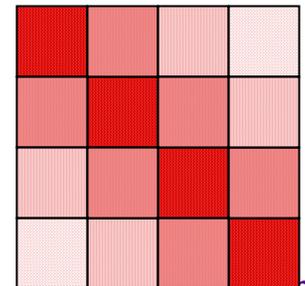


scales  
poorly  
- storage  
- banded  
matrix  
inversion

- Sparse optimal similarity transforms

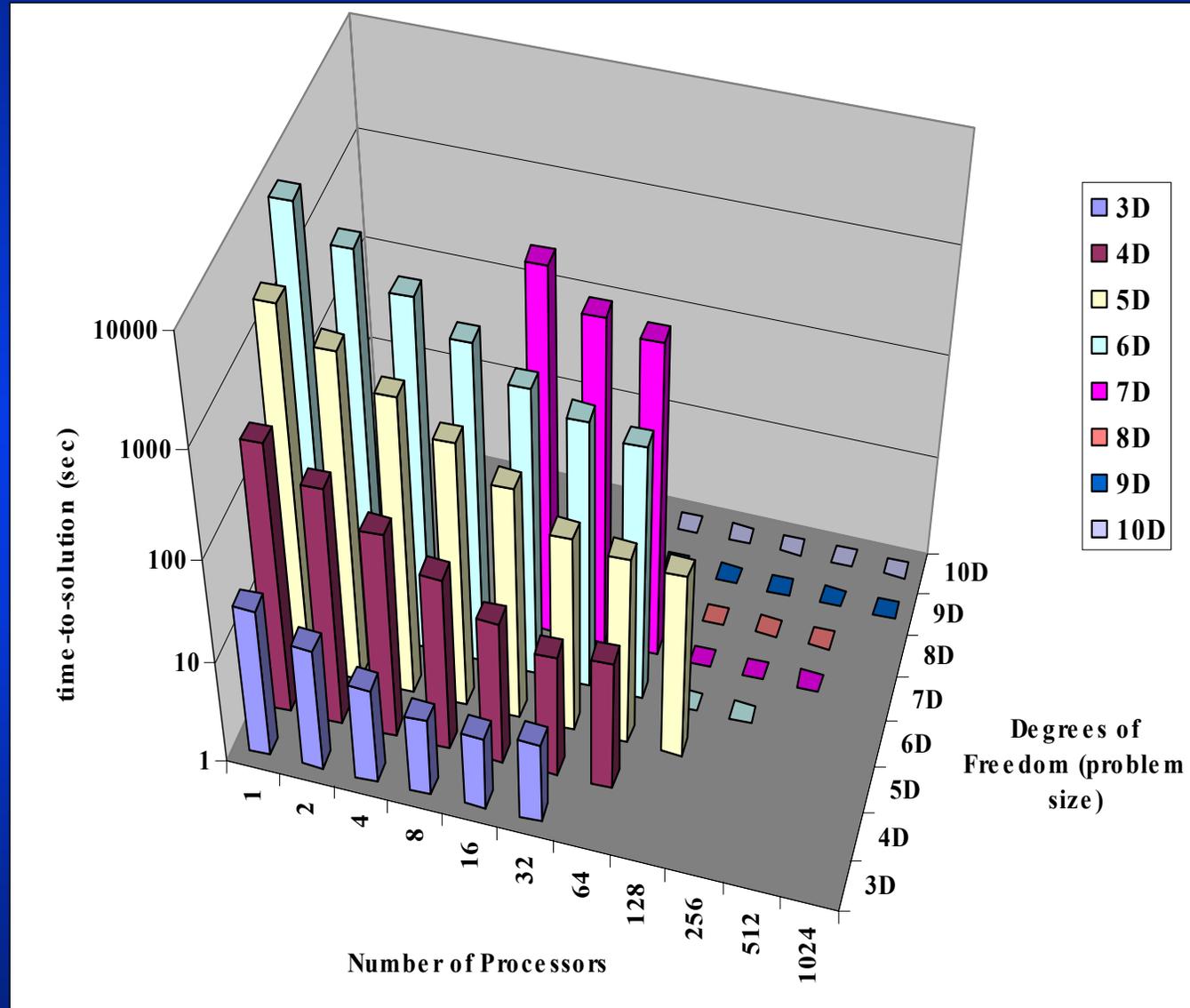


THEN find optimal  $Q$  such that  $QHQT =$   
*optimal block diagonal*  $\rightarrow$



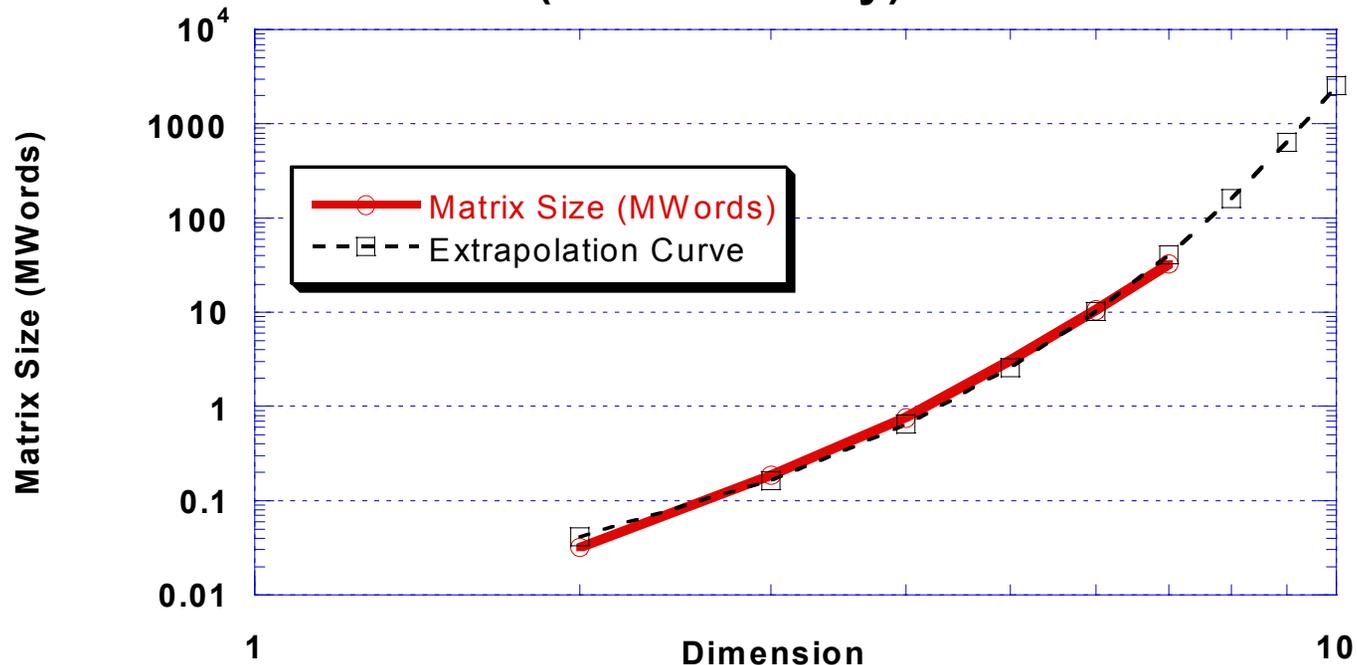
# Combustion Kinetics

- ◆ Parallel computing dramatically lowers time-to-solution for rate constants
- ◆ ANL software libraries (PETSc) important
- ◆ Parallel computing is only way to solve large problem: more processors  $\Rightarrow$  larger problems
- ◆ Results obtained on NERSC IBM/SP.



# Extrapolated Storage Requirements

Storage Required for F=3.5 Cutoff, .32ev  
(3% Accuracy)



CMCIM02