Parallelization of the Resolution of the Secular Equation in OpenMP

Students :
BAMBA Ibrahim
NITCHEU Paul

Supervisor :
ABBAS TURKI Lokmane

Sorbonne University/Science

June 2019

Summary

Introduction

The symmetric eigenvalue problem is one of the most fundamental problems of computational mathematics.

Many algorithms available for the symmetric eigenproblem :

- `Bisection algorithm`
- `QR iteration`
- **`Tridiagonalization of Householder`**

We're using here the *Tridiagonalization of Householder* : one of the fastest methods, find all eigenvalues and eigenvectors, use the Divide-and-conquer method

Introduction

Divide-and-Conquer

- partition the tridiagonal eigenvalue problem into two (or more) smaller tridiagonal eigenvalue problems
- solve the two (or more) smaller problems
- combine the solutions of the smaller problems to get the desired solution of the overall problem

At the heart of solving problems is the resolution of an equation so-called *Secular Equation*

Introduction

Goal of our study

Adapt and compare in OpenMP two algorithms for solving the Secular Equation namely the Gragg Method and the Hybrid Method

Motivation

Tridiagonalization of Householder

The problem we consider is the following : given a real $n * n$ symmetric matrix T, find all of the eigenvalues and corresponding eigenvectors of T.

$$
T = \left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
 & \ddots & a_{m-1} & b_{m-1} & & & & \\
 & & b_{m-1} & a_m & b_m & & & \\
\hline
 & & & b_m & a_{m+1} & b_{m+1} & & \\
 & & & & b_{m+1} & \ddots & & \\
 & & & & & & \ddots & b_{n-1} \\
 & & & & & & b_{n-1} & a_n
\end{array}\right]
$$

A tridiagonal real symmetric matrix T can be decomposed into the sum of two matrices,

$$
= \left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
& \ddots & a_{m-1} & b_{m-1} & & & & \\
& & b_{m-1} & a_m - b_m & & & & \\
\hline
& & & & a_{m+1} - b_m & b_{m+1} & & \\
& & & & b_{m+1} & \ddots & & \\
& & & & & & \ddots & b_{n-1} \\
& & & & & & b_{n-1} & a_n
\end{array}\right]
$$

$$
+ \left[\begin{array}{cc|cc}
& & & \\
& b_m & b_m & \\
\hline
& b_m & b_m & \\
& & &
\end{array}\right]
$$

$$
= \left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array}\right] + b_m \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [0, \ldots, 0, 1, 1, 0, \ldots, 0] \equiv \left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array}\right] + b_m v v^T .
$$

$$
\begin{aligned}
T &= \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + b_m vv^T \\
&= \begin{bmatrix} Q_1 D_1 Q_1^T & 0 \\ 0 & Q_2 D_2 Q_2^T \end{bmatrix} + b_m vv^T \\
&= \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} (D + \rho vv^T) \begin{bmatrix} Q_1^T & 0 \\ 0 & Q_2^T \end{bmatrix}
\end{aligned}
$$

where

$$
b_m = \rho, u = \begin{bmatrix} Q_1^T & 0 \\ 0 & Q_2^T \end{bmatrix}, D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}
$$

To find the eigenvalues of $D + \rho uu^T$, assume first that $D - \lambda I$ is nonsingular, and compute the characteristic polynomial as follows :

$$
det(D + \rho uu^T - \lambda I) = det((D - \lambda I)(I + \rho (D - \lambda I)^{-1} uu^T) \tag{1}
$$

SORBONNE UNIVERSITÉ

Lemma : If x and y are vectors, $det(I + xy^T) = 1 + y^T x$
Therefore equation (1) becomes :

$$\begin{aligned} det(I + \rho(D - \lambda I)^{-1}uu^T) &= 1 + \rho u^T (D - \lambda I)^{-1} u \\ &= 1 + \rho \sum_{i=1}^{k} \frac{u_i^2}{d_i - \lambda} \\ &= f(\lambda) \end{aligned}$$

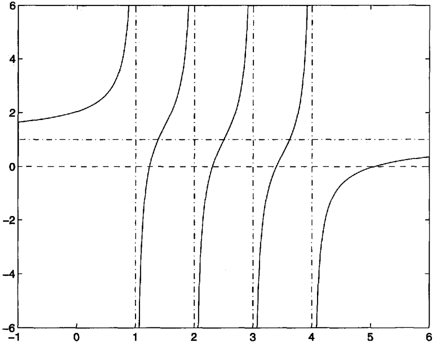
- $f(\lambda)$ is called Secular Equation
- we add one interval to the right of d_n with $d_{n+1} = d_n + \sum_{i=1}^{n} \frac{u_i^2}{\rho}$ if $\rho > 0$ or to the left of d_1 with $d_0 = d_1 + \sum_{i=1}^{n} \frac{u_i^2}{\rho}$ otherwise
- $f(x)$ has precisely n zeros λ_i, one in each open interval $(d_j; d_{j+1})$

SORBONNE UNIVERSITÉ

Motivation

Secular Equation - Example

Figure 0: Graph of secular function with 4 intervals



SORBONNE UNIVERSITÉ

Secular Equation Solvers

Let

$$f(x) = 1 + \psi(x) + \phi(x) \tag{2}$$

with

$$\psi(x) = \rho \sum_{j=1}^{k} \frac{u_j^2}{d_j - x}, \qquad \phi(x) = \rho \sum_{j=k+1}^{n} \frac{u_j^2}{d_j - x} \tag{3}$$

where :

- $\psi(x) \rightarrow$ sum of the negative terms of f
- $\phi(x) \rightarrow$ sum of the positive terms of f

Secular Equation Solvers

Gragg Method

* Cubic convergence and monotonous on normal intervals
* Based on a third-order approximation of the form

$$Q(x; c, s, S) = c + \frac{s}{d_k - x} + \frac{S}{d_{k+1} - x} \tag{4}$$

y fixed approximation to λ_k between $]d_k, d_{k+1}]$. (4) allow us to write

$$f(y) = c + \frac{s}{d_k - y} + \frac{S}{d_{k+1} - y} \tag{5}$$

The idea to compute a correction η to y for the next ("better") approximation $y + \eta$ to λ_k is to solve the equation $Q(y; c, s, S) = 0$.

SORBONNE UNIVERSITÉ

Gragg proposed to chose c, s, S so that $Q(y; c, s, S)$ matches $f(\lambda)$ at y up to the second derivate. So we have:

$$f'(y) = \frac{s}{(d_k - y)^2} + \frac{S}{(d_{k+1} - y)^2} \tag{6}$$

$$\frac{f''(y)}{2} = \frac{s}{(d_k - y)^3} + \frac{S}{(d_{k+1} - y)^3} \tag{7}$$

(5), (6), (7) yield:

$$s = \frac{{\Delta_k}^3 \Delta_{k+1}}{\Delta_k - \Delta_{k+1}} \left(\frac{f'(y)}{\Delta_{k+1}} - \frac{f''(y)}{2} \right) \quad = u_k^2 + \frac{(d_k - y)^3}{d_k - d_{k+1}} \sum_{i \neq k, k+1}^{n} \frac{d_i - d_{k+1}}{(d_k - y)^3} u_i^2 > u_k^2$$

$$S = \frac{{\Delta_k}^3 \Delta_{k+1}}{\Delta_{k+1} - \Delta_k} \left(\frac{f'(y)}{\Delta_k} - \frac{f''(y)}{2} \right) \quad = u_{k+1}^2 + \frac{(d_{k+1} - y)^3}{d_{k+1} - d_k} \sum_{i \neq k, k+1}^{n} \frac{d_i - d_k}{(d_i - y)^3} u_i^2 > u_{k+1}^2$$

$$c = f(y) - (\Delta_k - \Delta_{k+1}) f'(y) + \Delta_k \Delta_{k+1} \frac{f''(y)}{2}$$

with $\Delta_k = d_k - y$.

SORBONNE UNIVERSITÉ

Gragg Method

Gragg Algorithm

Algorithm 1 Calculate approximation λ_k of the root of f by Gragg Method

Require: k
Ensure: λ_k between d_k and d_{k+1}

```
j ← 0
λ_j ← initialguess on ]d_k, d_{k+1}[
repeat
   a ← compute a on ]d_k, d_{k+1}[
   b ← compute b on ]d_k, d_{k+1}[
   c ← compute c on ]d_k, d_{k+1}[
   {Computing the stop criteria}
   if (ρ > 0 and k ≠ n − 1) or (ρ < 0 and k ≠ 0) then
      stop ← (λ_{j−1} − λ_{j−2}) * (λ_{j−1} − λ_j) {intervals with monotonous}
   else
      stop ← compute stop criteria on non monotonous interval {interval without monotonous}
   end if
   λ_{j+1} ← λ_j + η {eta computed by 10
until stop is false
return λ_j
```

Secular Equation Solvers

Middle Way Method

Consist of interpolating $\psi(x)$ and $\phi(x)$ by

$$r + \frac{s}{d_k - x} \text{ an approximation of } \psi(x)\text{, and}$$
$$R + \frac{S}{d_{k+1} - x} \text{ an approximation of } \phi(x)$$

with $\begin{cases} s = \Delta_k^2 \psi_k'(y) > 0, \\ r = \psi_k(y) - \Delta_k \psi_k'(y) \leq 0, \end{cases}$ and $\begin{cases} S = \Delta_{k+1}^2 \phi_k'(y) > 0, \\ R = \phi_k(y) - \Delta_{k+1} \phi_k'(y) \geq 0, \end{cases}$

and $\Delta_k = d_k - y < 0 < \Delta_{k+1} = d_{k+1} - y$

SORBONNE UNIVERSITÉ

Middle Way Method (*suite*)

Next better iteration $y + \eta$:

$$\eta = \frac{2b}{a+(-1)^i\sqrt{a^2-4bc}}$$

with $i = 0$ if $k \neq 0$ and 1 otherwise and

$$\begin{aligned} a &= (\Delta_k + \Delta_{k+1})f(y) - \Delta_k\Delta_{k+1}f'(y), \\ b &= \Delta_k\Delta_{k+1}f(y), \\ c &= f(y) - \Delta_k\psi_k'(y) - \Delta_{k+1}\phi_k'(y) \end{aligned}$$

Secular Equation Solvers

Fixed Weight Method

Fixes one of the weights u_k and u_{k+1} while satisfying (5), (6)

The case λ_k closer to d_k, we fix s to u_k^2 :

$$s = u_k^2$$

$$S = \Delta_{k+1}^2 (f'(y) - \frac{u_k}{\Delta_k})$$

$$= u_{k+1}^2 + \sum_{j \neq k, k+1} \frac{\Delta_{k+1}^2}{\Delta_j^2} u_j^2 > u_{k+1}^2$$

$$c = f(y) - \frac{u_k^2}{\Delta_k} - \Delta_{k+1}(f'(y) - \frac{u_k^2}{\Delta_k^2})$$

$$= f(y) - \Delta_{k+1} f'(y) - \frac{u_k^2}{\Delta_k^2}(d_k - d_{k+1})$$

The case λ_{k+1} closer to d_k : we fix S to u_{k+1}^2 :

$$s = \Delta_k^2 (f'(y) - \frac{u_{k+1}}{\Delta_{k+1}})$$

$$= u_k^2 + \sum_{j \neq k, k+1} \frac{\Delta_k^2}{\Delta_j^2} u_j^2 > u_k^2$$

$$S = u_{k+1}^2$$

$$c = f(y) - f'(y)\Delta_k - \frac{u_{k+1}^2}{\Delta_{k+1}^2}(d_{k+1} - d_k)$$

SORBONNE UNIVERSITÉ

Secular Equation Solvers

Hybrid Scheme

Let

$$f_m(x) = \rho + \sum_{j=1, j\neq m}^{n} \frac{u_j^2}{d_j - x} \tag{8}$$

the secular function $f(x)$ with the *mth* term in the summation removed.

- uses the Fixed Weight Method to interpolate $f(x)$ or $f_k(x)$ for the first iteration
- combines the two previous methods by switching between both when necessary for the next
- uses in some cases three poles : d_{k-1}, d_k and d_{k+1} instead of d_k and d_{k+1}

Secular Equation Solvers

Hybrid Scheme

Interpolating $f_k(x)$

Let

$$\tilde{Q}(x; c, s, S) = c + \frac{s}{d_{k-1} - x} + \frac{u_k^2}{d_k - x} + \frac{S}{d_{k+1} - x} \tag{9}$$

- uses three poles by approximating $f(x)$ to $\tilde{Q}(x; c, s, S)$
- c, s and S are determined by interpolating $f_k(x)$ or $f_{k+1}(x)$
- then uses dichotomy to find next best approximation $y + \eta$

Hybrid Scheme - Hybrid algorithm

Algorithm 2 Calculate approximation λ_k of the root of f by Hybrid Scheme

Require: k
Ensure: λ_k between d_k and d_{k+1}

```
j ← 0
y_j ← initialguess on ]d_k, d_{k+1}[
compute f_k(y_j)
{Taking decision}
if f_k(y_j) > 0 then
  two poles d_k and d_{k+1} are used
  we interpolate f(x) with Fixed Weight
else
  three poles d_{k-1}, d_k and d_{k+1} are used
  if we are in the case 1 then
    we interpolate f_k(x) as in (20) with Fixed Weight
  end if
  if we are in the case 2 then
    we interpolate f_{k+1}(x) as in (20) with Fixed Weight
  end if
end if
j ← j + 1
if f(y_j) < 0 and |f(y_j)| > 0.1 * |f(y_{j-1})| then
  restart the first iteration and use Middle Way
else
  restart the first iteration and continue to use Fixed Weight
end if
j ← j + 1
while stop criteria is false do
  if f(y_j) * f(y_{j-1}) > 0 and |f(y_j| > 0.1 * |f(y_{j-1})| then
    switch method
  end if
  restart iteration with the method chosen
  j ← j + 1
end while
return y_j
```

Secular Equation Solvers

Initial Guess

We rewrite the secular function as $f(x) = g(x) + h(x)$ where,

$$g(x) = \rho + \sum_{j=1, j\neq k, k+1} \frac{u_j^2}{d_j - x} \quad \textit{and} \quad h(x) = \frac{u_k^2}{d_k - x} + \frac{u_{k+1}^2}{d_{k+1} - x} \tag{10}$$

We choose our initial guess y to be that one of the two roots of the equation

$$g(\frac{d_k + d_{k+1}}{2}) + h(y) = 0 \tag{11}$$

In case $f(\frac{d_k+d_{k+1}}{2}) \geq 0$, equation (11) should be solved for $\tau = y - d_k$ and while in case $f(\frac{d_k+d_{k+1}}{2}) < 0$, it should be solve for $\tau = y - d_{k+1}$. Define $\Delta = d_k - d_{k+1}$ and $c = g(\frac{d_k - d_{k+1}}{2})$.The τ formulas are:

$$\begin{aligned} \tau = y - d_k &= \frac{a - \sqrt{a^2 - 4bc}}{2c} \; \textit{if } a \leq 0, \\ &= \frac{2b}{a + \sqrt{a^2 - 4bc}} \text{ if } a > 0 \end{aligned} \tag{12}$$

where if $f(\frac{d_k+d_{k+1}}{2}) \geq 0$

$$K = k, \; a = c\Delta + (u_k^2 + u_{k+1}^2), \; b = u_k^2 \Delta, \tag{13}$$

and if $f(\frac{d_k+d_{k+1}}{2}) < 0$

$$K = k + 1, \; a = -c\Delta + (u_k^2 + u_{k+1}^2), \; b = -u_{k+1}^2 \Delta \tag{14}$$

Secular Equation Solvers

Stop Criteria

The Hybrid algorithm is not monotonous in each interval like Gragg algorithm in the added interval. So we need a specific stop criterion. In this case we have two criteria that have been defined for this algorithm:

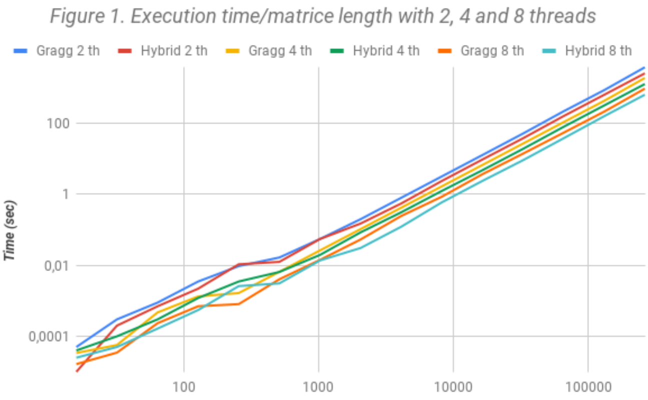
First stop criterion:

$$|\eta| \leq c\epsilon_m min(|d_k - x|, |d_{k+1} - x|)$$

Second stop criterion:

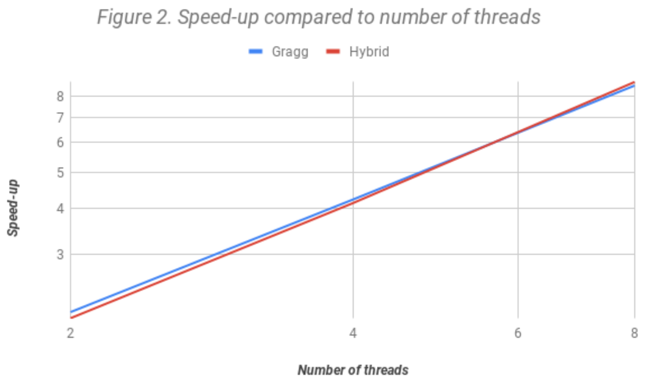
$$|f(d_k + \tau)| \leq e\epsilon_m + \epsilon_m|\tau||f'(d_k - \tau)|$$

Results and Analysis

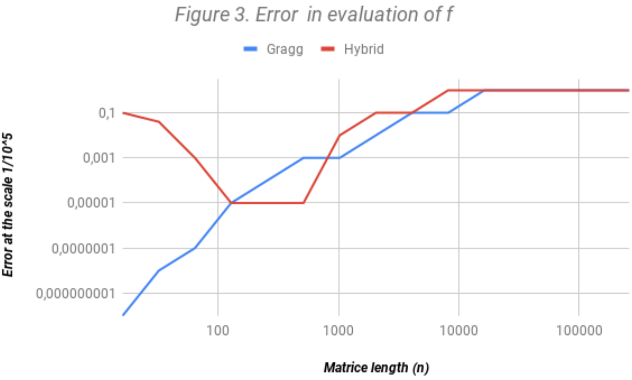


SORBONNE UNIVERSITÉ

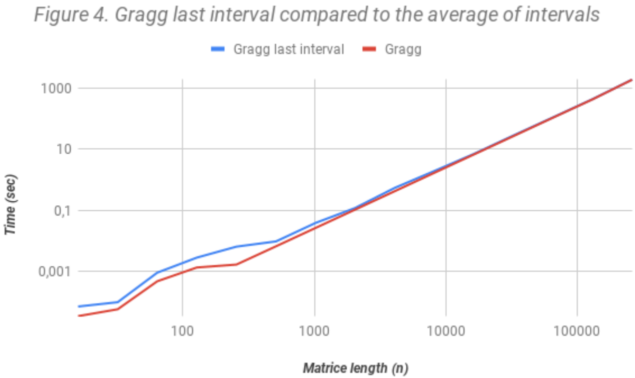
Results and Analysis



Results and Analysis



Results and Analysis



Conclusion

Our numerical results show that:

- The hybrid method has a much better execution time than Gragg (about 1.5 times less).
- In the Gragg algorithm the computation time on last interval slows down the overall calculation time of the algorithm
- The Hybrid method is better than Gragg when one is more concerned about the execution time. But when one is concerned about the accuracy of the eigenvalues the Gragg method seems to be better

References

- William B. Gragg, J. R. Thornton, and D. D. Warner (1992) : Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem and bidiagonal singular value problem, *Modeling and Simulation*, 23(1), 49-56, Clemson University
- James Demel (1997) : Apply numerical linear algebra, *SIAM*, 229-233
- J. J. M. Cuppen (1981) : A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numer. Math.*, 36, 177-195
- J. J. Dongarra and D. C. Sorensen (1987) : A fully parallel algorithm for the symmetric eigenvalue problem, *SIAM J. Sci. Stat. Comput.*, 8, s139-s154

References

- Ren-Cang Li (1993) : Solving Secular Equations Stably and Efficiently, *Computer Science Dept. Technical Report CS-94-260*, Department of Mathematics, University of California at Berkeley, Berkeley, California 94720
- G. H. Golub (1973) : Some modified matrix eigenvalue problems, *SIAM rev.*, 15, 318-334
- Gerard Meurand (2012) : Solving secular equations, 3-7
- Ming Gu and S. C. Eisenstadt (1992) : A stable and efficient algorithm for the rank one modification of the symmetric eigenproblems, *Research Report YaleU/DCS/RR 916*, Yale University

SORBONNE UNIVERSITÉ