A new implementation of the dqds algorithm

Kinji Kimura

Kyoto Univ.

We propose a new implementation of the dqds algorithm for the bidiagonal SVD.

Our aim(1)

```
srand(1); n=70000;
for(i=0;i<n;i++){
  b[i]=rand()/(double)RAND_MAX;
  if (rand() % 2==0) b[i]=-b[i];
}
for(i=0;i<n-1;i++){
  c[i]=rand()/(double)RAND_MAX;
  if (rand() % 2==0) c[i]=-c[i];
}
```

When we use the bisection method for computing singular values, Golub and Kahan tridiagonal matrices are adopted.
The smallest singular value is $1.028E-0214$ (bisection float128, bisection float64)
However, LAPACK DLASQ computes 0 as the smallest singular value.

In $n = 150000$, we found the same result.

Our aim(2)

To overcome this difficulty, we propose new shift strategies.

dqds algorithm

pqds algorithm

- $s^{(i)}$ is a shift term, The diagonal elements of $B^{(i)}$ converge singular values of $B^{(0)}$.

$$\left(B^{(i+1)}\right)^\top B^{(i+1)} = B^{(i)}\left(B^{(i)}\right)^\top - s^{(i)} I_n$$

- $s^{(i)} \leq \lambda_{\min}\left(B^{(i)}\left(B^{(i)}\right)^\top\right)$. When we choose the value of $s^{(i)}$ as the good approximate value of the smallest eigen value of $B^{(i)}\left(B^{(i)}\right)^\top$, the convergence speed of $(B^{(i)})_{n-1,n} \to 0$ is accelerated.
- When we choose $s^{(i)} > \lambda_{\min}\left(B^{(i)}\left(B^{(i)}\right)^\top\right)$, the "shift reconstruction" is adopted.

Relationship between the orthogonal QD algorithm and the pqds algorithm

From,

$$P^{(i)} \begin{bmatrix} L^{(i)} \\ t^{(i)} I_n \end{bmatrix} = \begin{bmatrix} U^{(i)} \\ t^{(i+1)} I_n \end{bmatrix},$$

we obtain,

$$\left(L^{(i)}\right)^\top L^{(i)} + \left(t^{(i)}\right)^2 I_n = \left(U^{(i)}\right)^\top U^{(i)} + \left(t^{(i+1)}\right)^2 I_n.$$

From,

$$t^{(i+1)} = \sqrt{(t^{(i)})^2 + (u^{(i)})^2}, \left(t^{(i+1)}\right)^2 = \left(t^{(i)}\right)^2 + \left(u^{(i)}\right)^2,$$

we obtain,

$$\left(L^{(i)}\right)^\top L^{(i)} - \left(u^{(i)}\right)^2 I_n = \left(U^{(i)}\right)^\top U^{(i)}.$$

The orthogonal QD algorithm

$$P\begin{bmatrix} L \\ tI_n \end{bmatrix} = \begin{bmatrix} U \\ \hat{t}I_n \end{bmatrix}, \hat{t} = \sqrt{t^2+u^2},\ L = \mathrm{bidiag}\begin{bmatrix} \alpha_1 & \cdots & \alpha_{n-1} & & \alpha_n \\ & \beta_1 & \cdots & \beta_{n-1} & \end{bmatrix}$$

LU step presented in the matrix elements

$$\eta_1 = \alpha_1, \rho_1 = \sqrt{\eta_1 - u}\sqrt{\eta_1 + u}$$

for $j = 1, \ldots, n-1$ **do**

$$\gamma_j = \sqrt{(\rho_j)^2 + (\beta_j)^2}, \zeta_j = \frac{\beta_j}{\gamma_j}\alpha_{j+1},$$

$$\eta_{j+1} = \frac{\rho_j}{\gamma_j}\alpha_{j+1}, \rho_{j+1} = \sqrt{\eta_{j+1} - u}\sqrt{\eta_{j+1} + u}$$

end for

$$\gamma_n = \rho_n.$$

The dqds algorithm

The dqds algorithm

From the orthgonal QD algorithm, using the variable transformation,

$$q_j = \alpha_j^2, e_j = \beta_j^2, \theta = u^2, \overline{q}_j = \gamma_j^2, \overline{e}_j = \zeta_j^2, d_j = \rho_j^2$$

we obtain the dqds algorithm,

$d_1 = q_1 - \theta$

for $j = 1, \ldots, n-1$ **do**

$$\overline{q}_j = d_j + e_j, \ \overline{e}_j = e_j \frac{q_{j+1}}{\overline{q}_j},$$

$$d_{j+1} = q_{j+1} \frac{d_j}{\overline{q}_j} - \theta = d_j \frac{q_{j+1}}{\overline{q}_j} - \theta$$

end for

$\overline{q}_n = d_n$

Idea for computing the correct values as the smallest singular values of the matrices in $n = 70000, 150000$

Key point

We catch the smallest singular value as the summation of shift values.

Tactics

For all matrices with any condition numbers, the lower bound($s^{(i)}$) which can outputs the positive value for the smallest eigen value and **has the sharpness** is required;

$$0 < s^{(i)} \leq \lambda_{\min}\left(B^{(i)}\left(B^{(i)}\right)^{\top}\right).$$

Shift strategies of dqds algorithm

Our proposed shift strategies

1. **Newton bound**, **generalized Newton bound**, **Laguerre bound**, Johnson bound, generalized Rutishauser shift, and Collatz shift are combined.
2. When $s^{(i)}$ is computed from these bounds, it is guaranteed that $0 < s^{(i)} \leq \lambda_{\min}\left(B^{(i)}\left(B^{(i)}\right)^{\top}\right)$ mathematically.
3. On computers, it can not be guaranteed.
4. However, the "shift reconstruction" solve the problem generated by rounding error.

Laguerre bound, (generalized) Newton bound

- Laguerre bound
  $n/\left(\mathrm{Tr}\left((BB^\top)^{-1}\right)+\sqrt{(n-1)\left(n\cdot\mathrm{Tr}\left((BB^\top)^{-2}\right)-(\mathrm{Tr}((BB^\top)^{-1}))^2\right)}\right)$
- (generalized) Newton bound: $\left(\mathrm{Tr}(BB^\top)^{-2}\right)^{-\frac{1}{2}}$, $\left(\mathrm{Tr}(BB^\top)^{-1}\right)^{-1}$

For,

$$B=\begin{bmatrix} b_1 & c_1 & & \\ & \ddots & \ddots & \\ & & \ddots & c_{n-1} \\ & & & b_n \end{bmatrix},$$

we can compute $\mathrm{Tr}\left((BB^\top)^{-1}\right)$ and $\mathrm{Tr}\left((BB^\top)^{-2}\right)$ using the following recursion relation.

$$f_1=\frac{1}{b_1^2}, f_j=\frac{1}{b_j^2}+\left(\frac{c_{i-1}}{b_j}\right)^2 f_{i-1}, j=2,\cdots,n, \mathrm{Tr}\left((BB^\top)^{-1}\right)=\sum_{i=1}^{n} f_j$$

$$g_1=f_1^2, g_j=f_j^2+\left(\frac{c_{i-1}}{b_j}\right)^2\left(g_{i-1}+f_{i-1}^2\right), j=2,\cdots,n, \mathrm{Tr}\left((BB^\top)^{-2}\right)=\sum_{i=1}^{n} g_j$$

Shift reconstruction

Shift reconstruction

$\tilde{u}$ is an arbitary constant. We use the following partial dqds algorithm;

$$\hat{d}_1 = q_1 - \tilde{u},\ \hat{d}_j = \hat{d}_{j-1} q_j / (\hat{d}_{j-1} + e_{j-1}) - \tilde{u}\ (j = 2, \ldots, n).$$

- If $\hat{d}_1 \leq 0$, after $\tilde{u} \leftarrow (1-\varepsilon) q_1$, we recompute.
- If $\hat{d}_j < 0 (j = 2, \ldots, n)$, after $\tilde{u} \leftarrow \max\left(\hat{d}_j + \tilde{u}, \tilde{u}/2\right)$, we recompute.
- If $\hat{d}_j = 0 (j = 2, \ldots, n-1)$, after $\tilde{u} \leftarrow (1-\varepsilon)\tilde{u}$, we recompute.
- We accept $\hat{d}_n = 0$.

Even if $\tilde{u}$ is the upper bound of the smallest eigen value, after the repetition of the above operations, we can get the lower bound of the smallest eigen value.

We can also use the thorem described above in the dqds algorithm.

Computational environment and test matrices

Computational environment

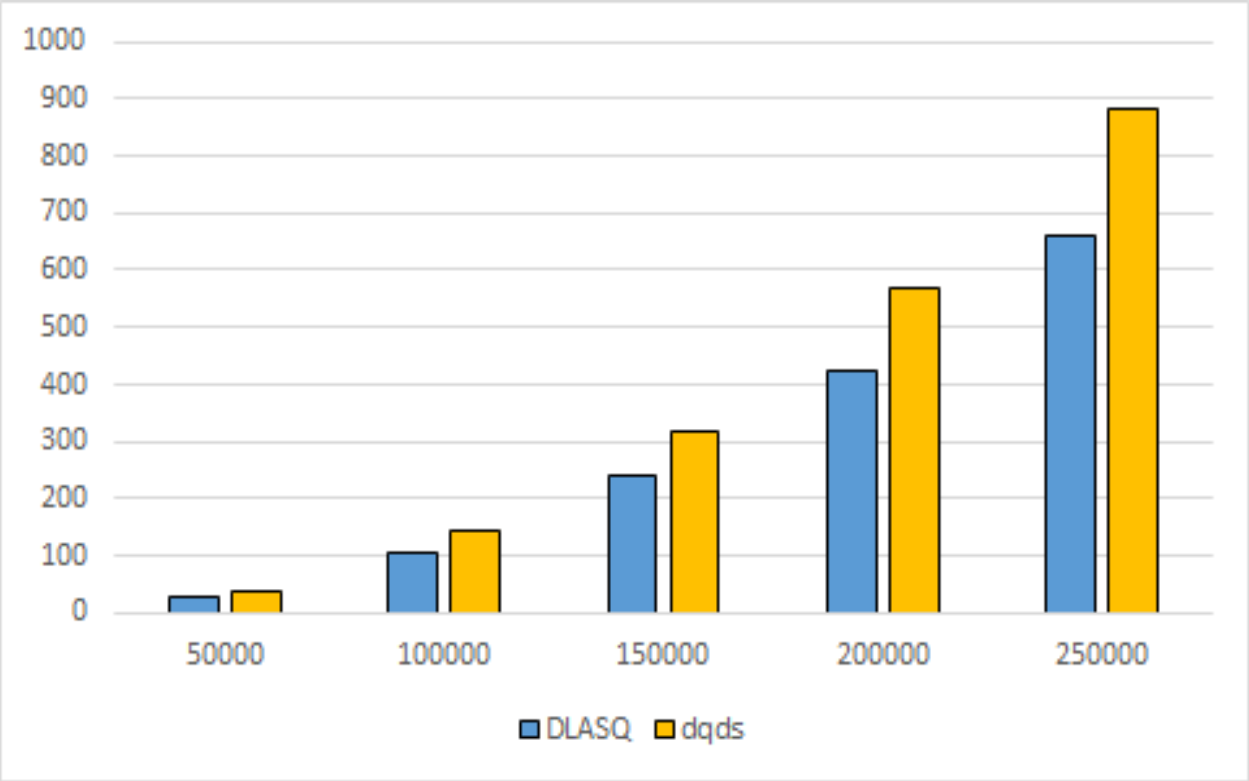
CPU: Intel Core i3-6100 CPU @ 3.70GHz, Memory: 16GB,
Compiler: icc & ifort 16.0.4, Compiler option:-O3 -ip -xHOST -fp-model precise -qopenmp -mkl, Library:Intel MKL 11.3.4

Test matrices

- All diagonal elements=1 and all off-diagonal elements=1:the correct singular value $\sigma_i = 2\cos\left(\frac{j}{2n+1}\pi\right), j = 1, \cdots, n$
- All diagonal elements and all off-diagonal elements are generated by uniformly distributed random numbers:the correct value computed by the parallel bisection method(float64)
- The glued Kimura matrices
- Using the inverse singular value problem, we cat get cluster matrices: $\sigma_i = \varepsilon^{(j-1)/(n-1)}, j = 1, \cdots, n$

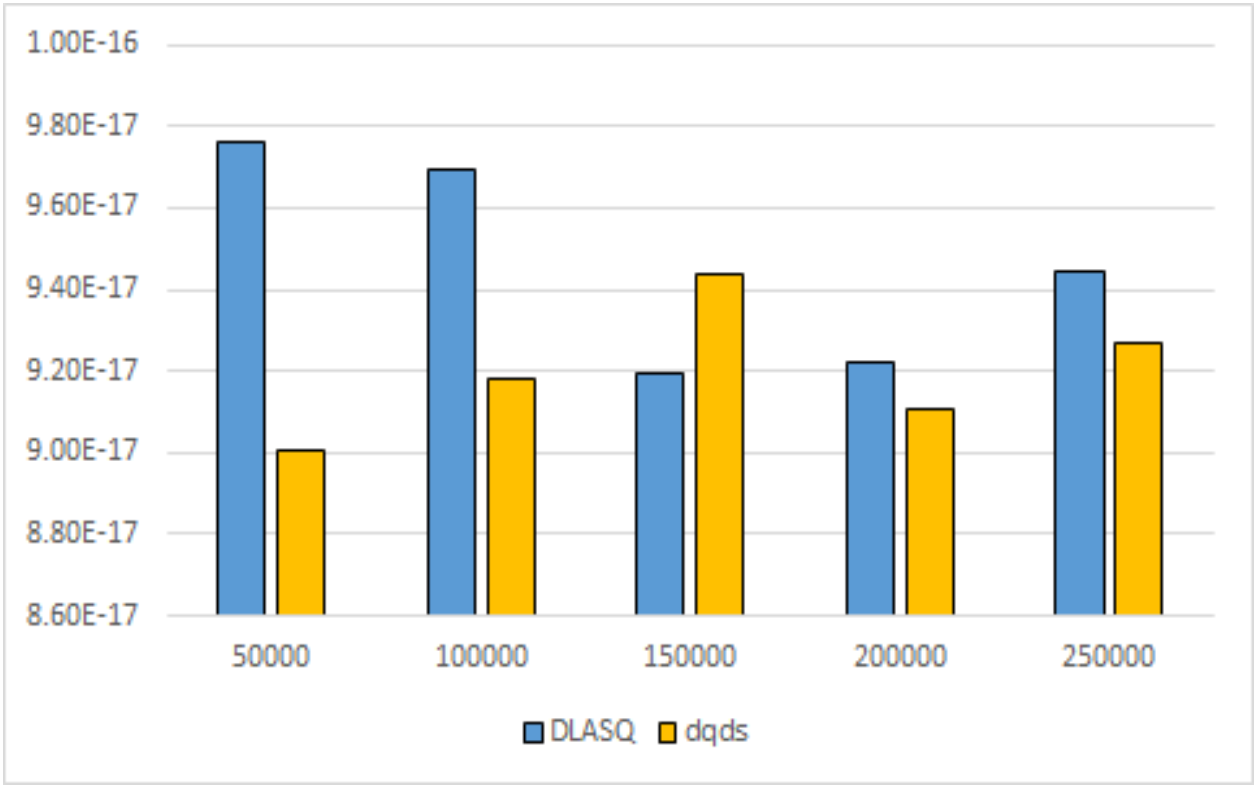
Numerical experiment: All diagonal elements=1 and all off-diagonal elements=1(1)

- Execution time[second]



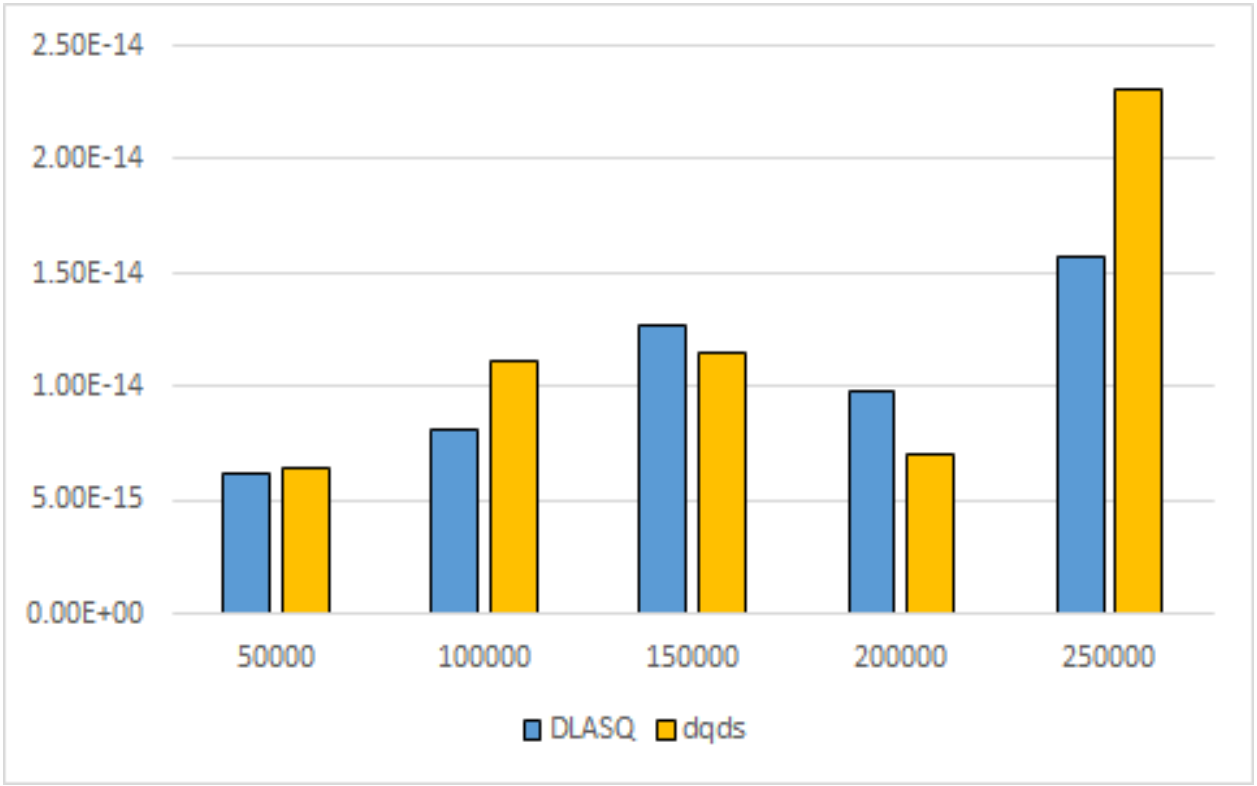
Numerical experiment: All diagonal elements=1 and all off-diagonal elements=1(2)

- The average of the relative errors for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices: $\frac{1}{n}\sum_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



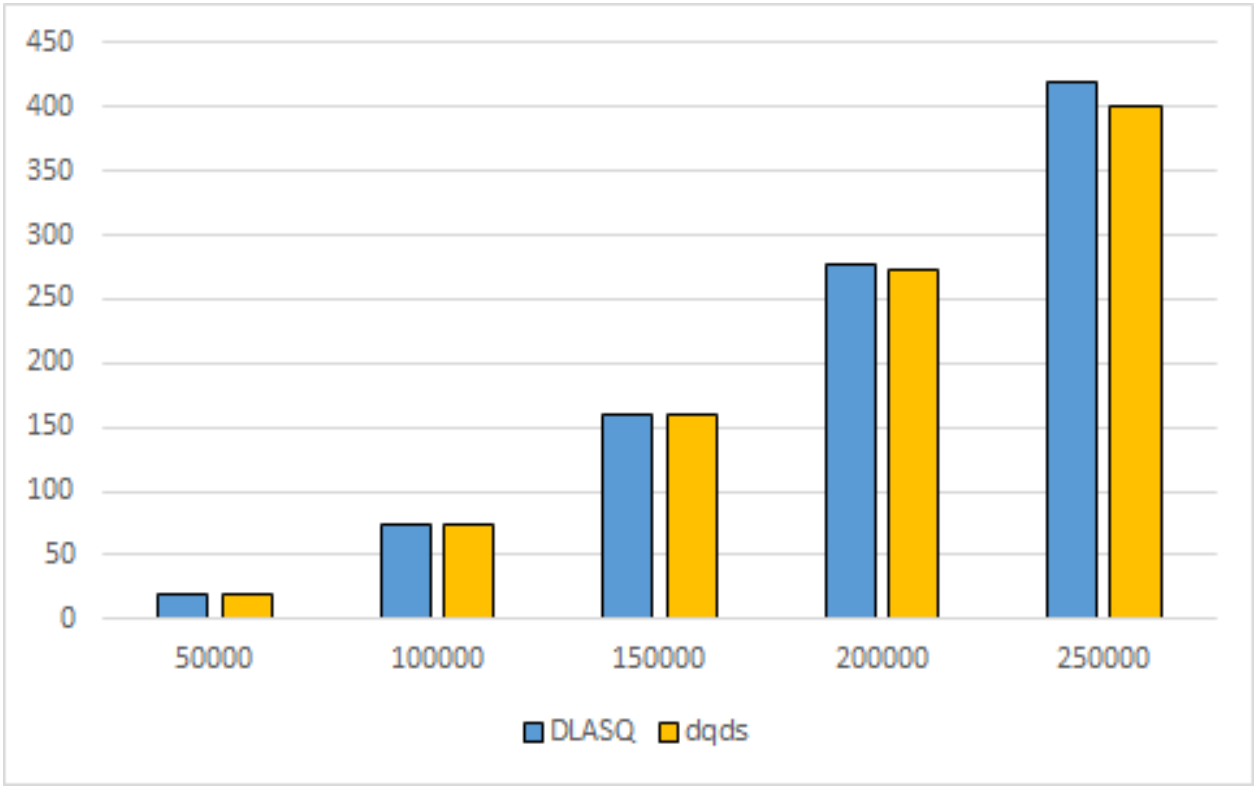
Numerical experiment: All diagonal elements=1 and all off-diagonal elements=1(3)

- The maximum relative error for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices:$\max_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



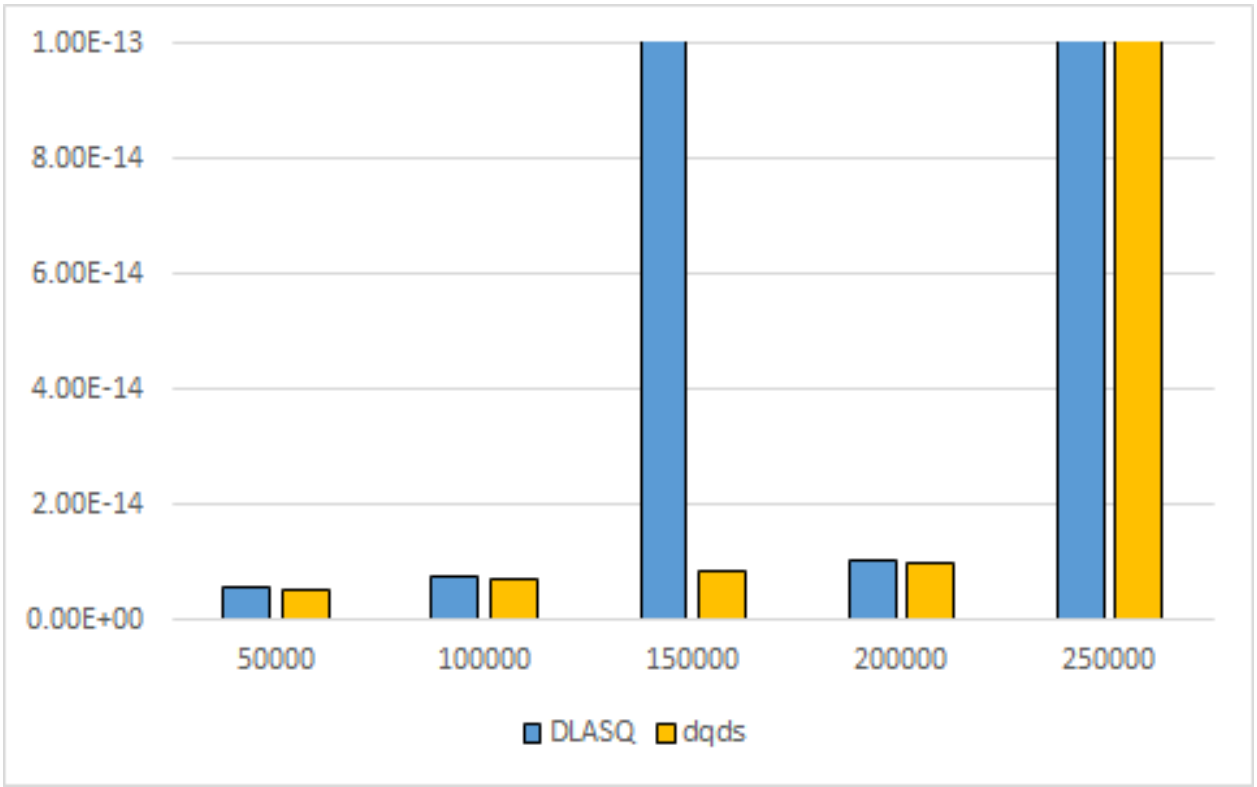
Numerical experiment: All diagonal elements and all off-diagonal elements are generated by uniformly distributed random numbers(1)

- Execution time[second]



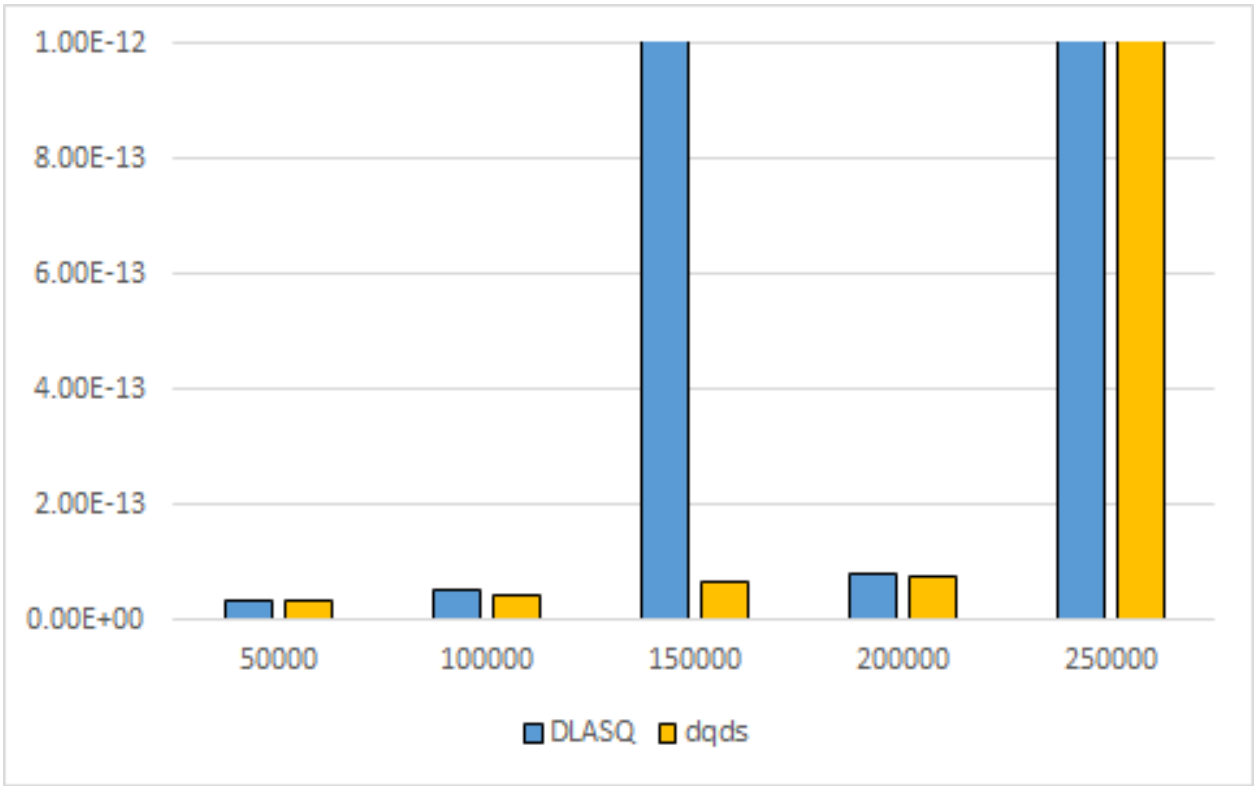
Numerical experiment: All diagonal elements and all off-diagonal elements are generated by uniformly distributed random numbers(2)

- The average of the relative errors for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices: $\frac{1}{n}\sum_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j|)}{DSTEBZ(\sigma_j)}$



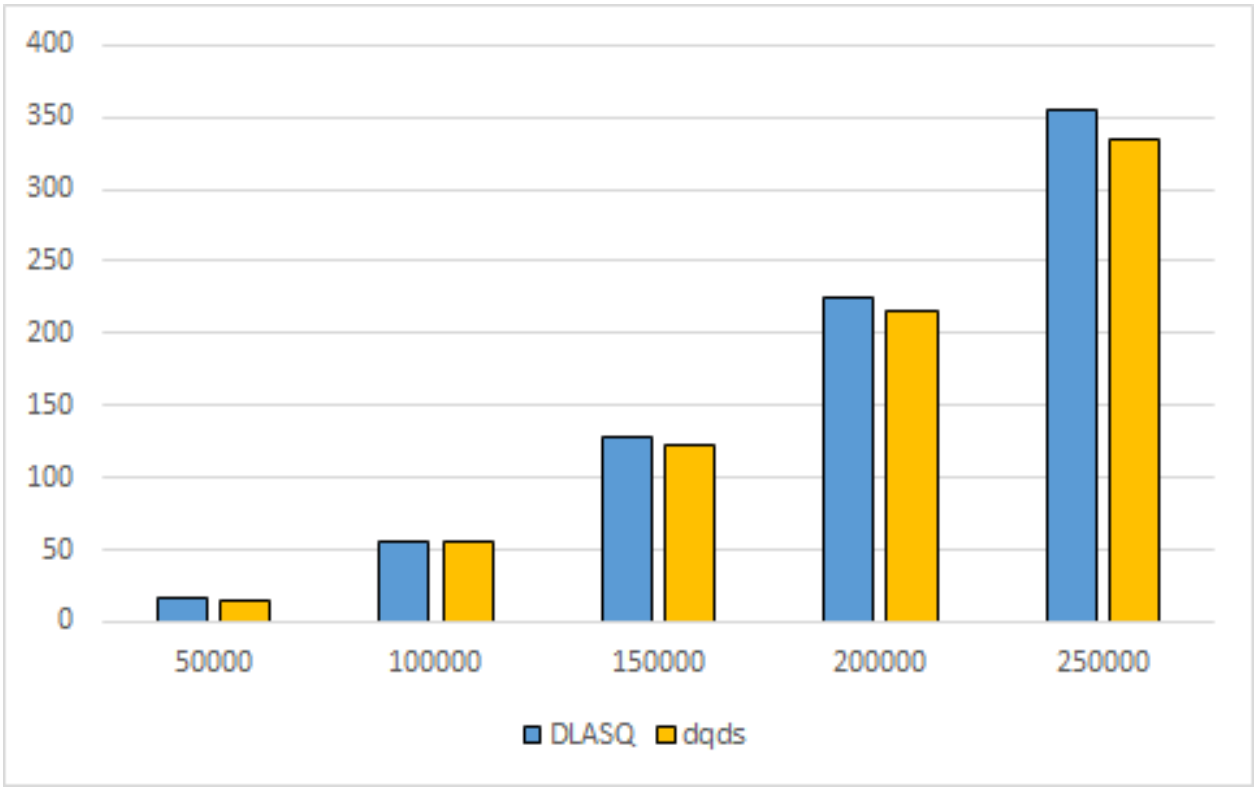
Numerical experiment: All diagonal elements and all off-diagonal elements are generated by uniformly distributed random numbers(3)

- The maximum relative error for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices:$\max_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



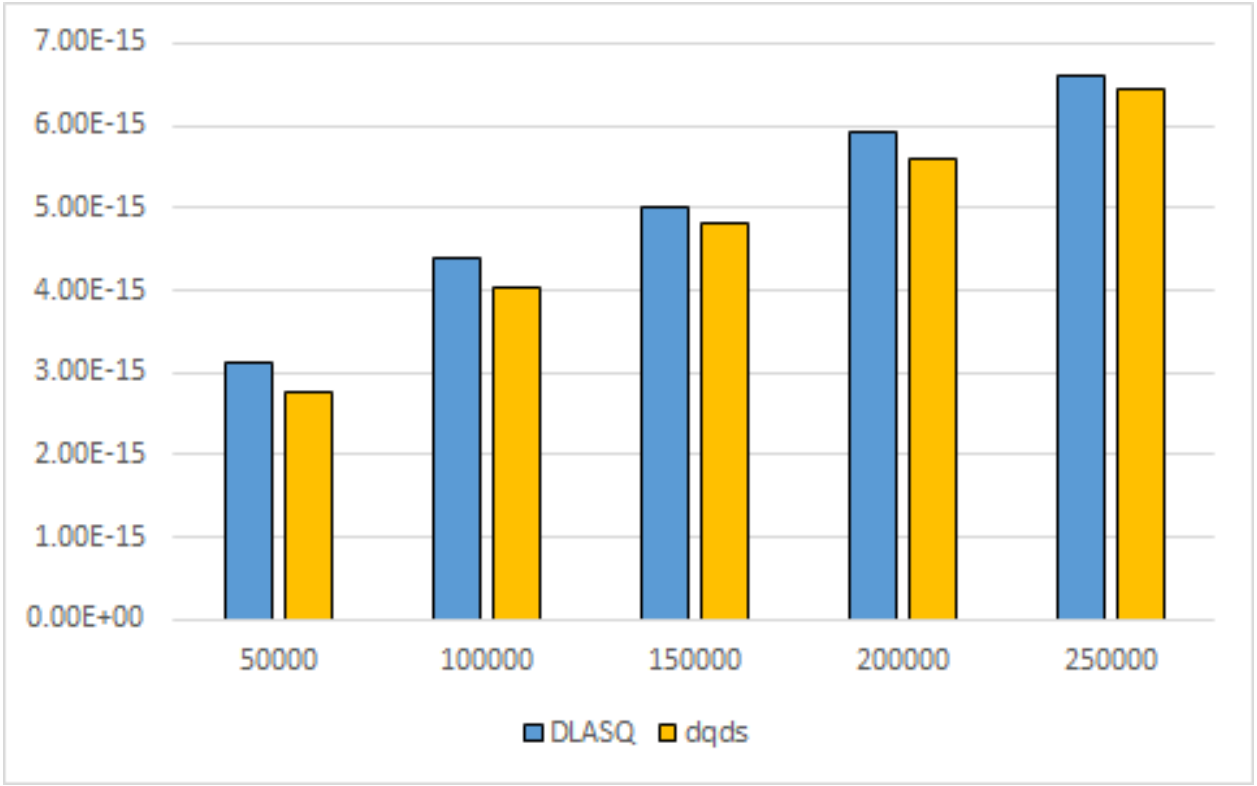
Numerical experiment: The glued Kimura matrices(1)

- Execution time[second]



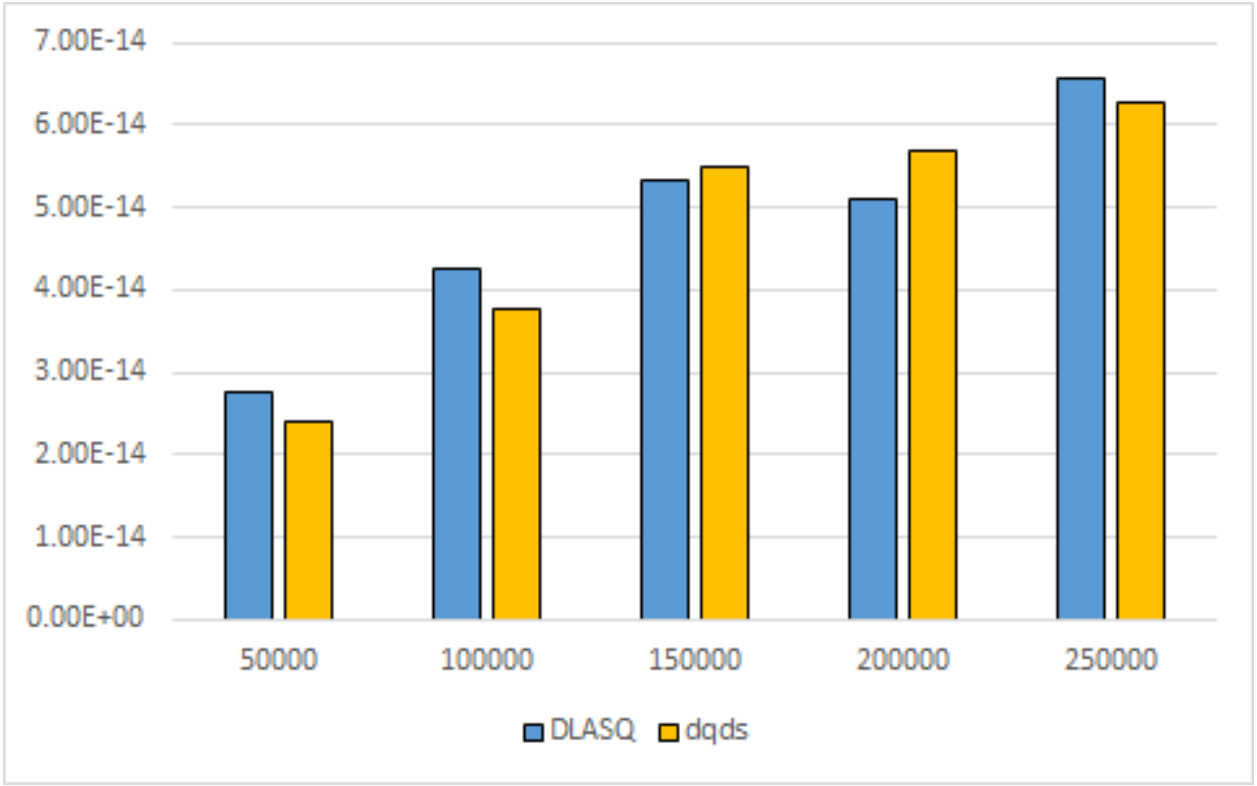
Numerical experiment: The glued Kimura matrices(2)

- The average of the relative errors for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices: $\frac{1}{n}\sum_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



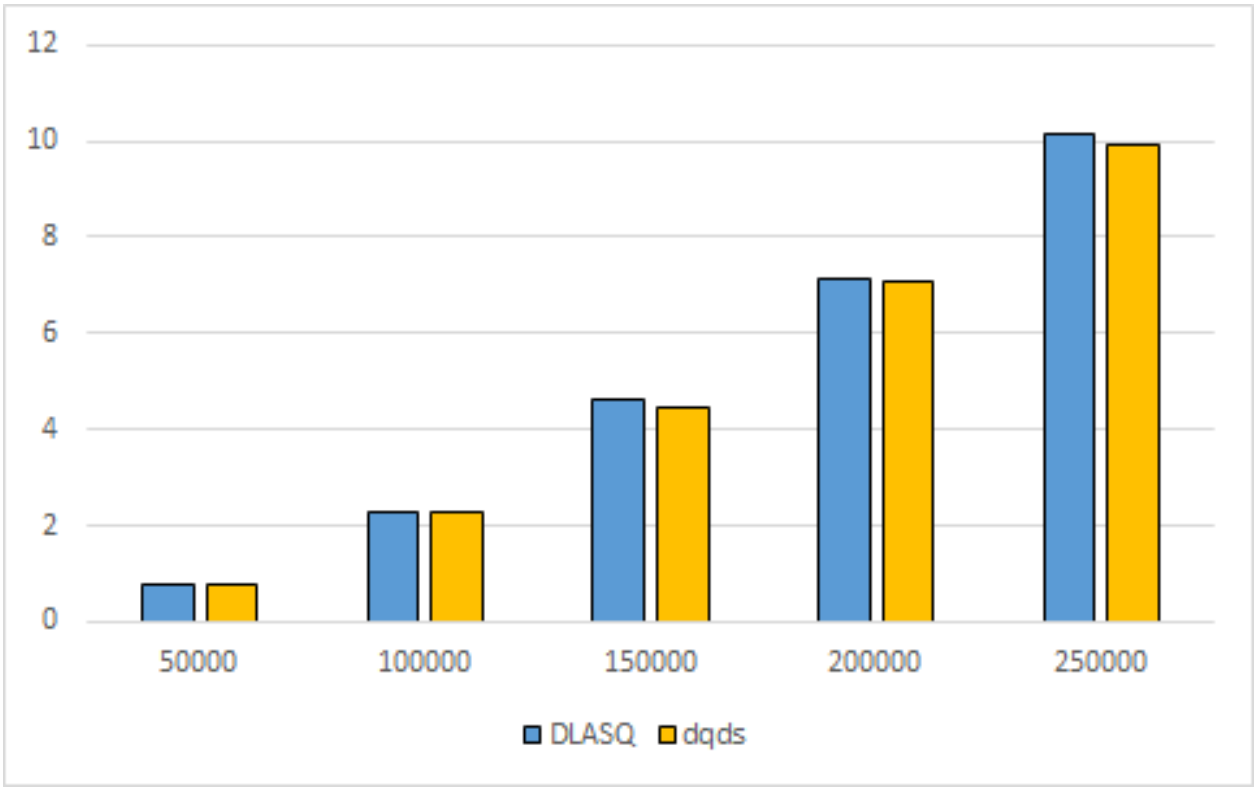
Numerical experiment: The glued Kimura matrices(3)

- The maximum relative error for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices:$\max_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



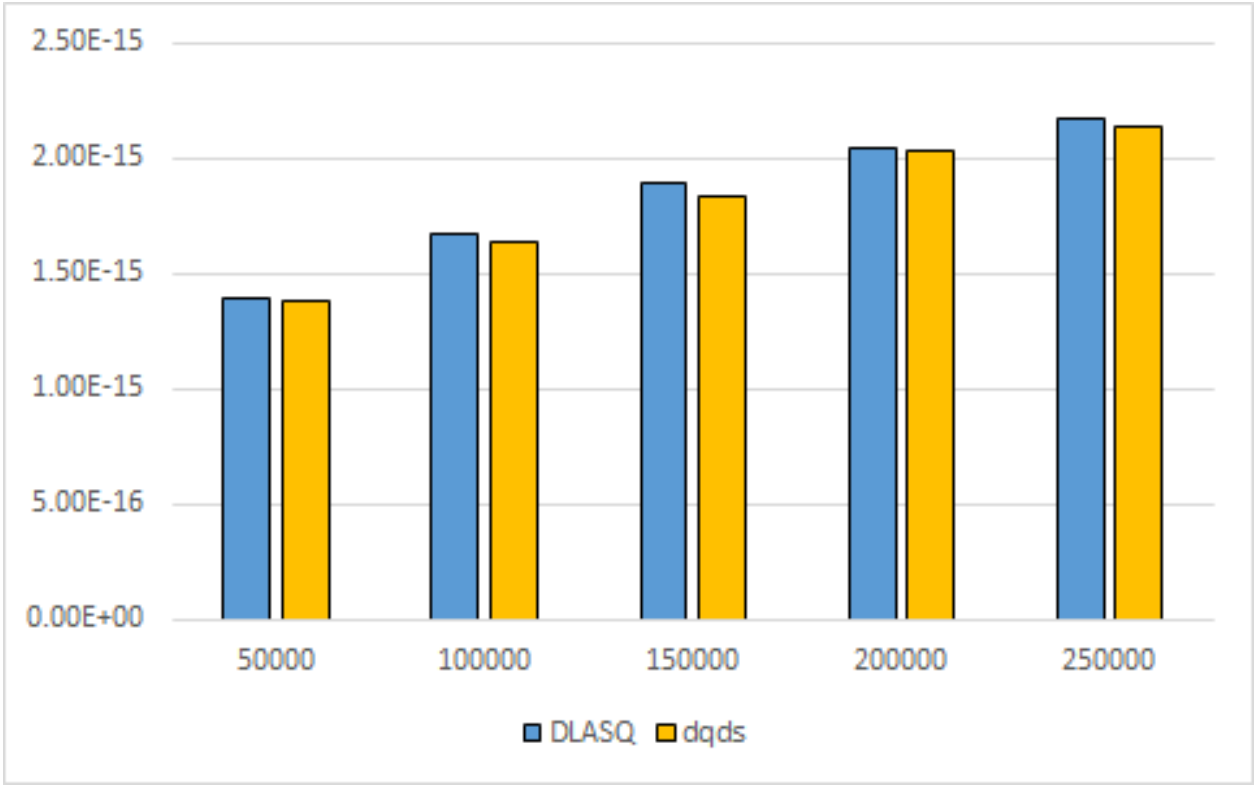
Numerical experiment: Cluster matrices(1)

- Execution time[second]



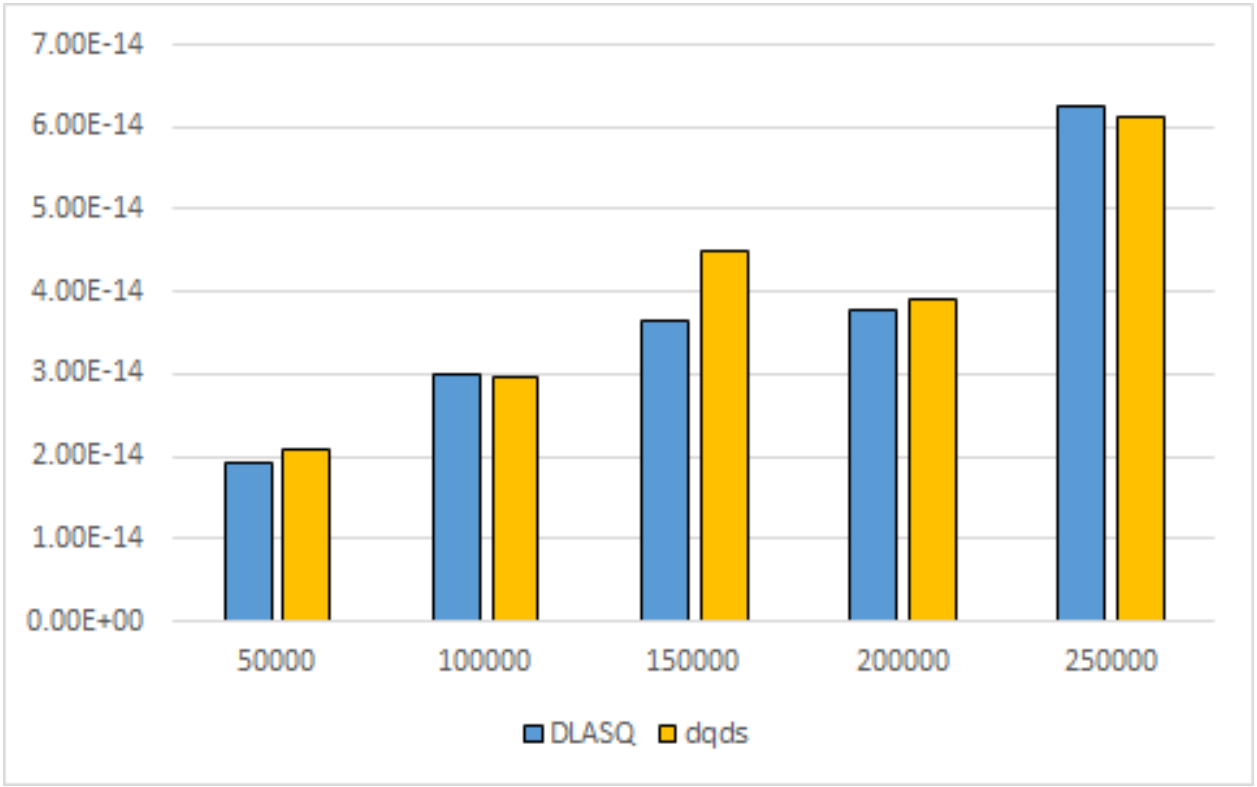
Numerical experiment: Cluster matrices(2)

- The average of the relative errors for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices: $\frac{1}{n}\sum_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



Numerical experiment: Cluster matrices(3)

- The maximum relative error for the result computed in the bisection method with float64 for tridiagonal Golub-Kahan matrices:$\max_j \frac{|\hat{\sigma}_j - DSTEBZ(\sigma_j)|}{DSTEBZ(\sigma_j)}$



Conclusion

Conclusion

- We proposed new shift strategies.
- We checked that the our implementation of the dqds algorithm can solve the problems in $n = 70000, 150000$.
- It is very difficult to solve the problem in $n = 250000$. Because the condition number is $8.01E+0307$. In the case, our proposed shift strategies can not compute the positive value. However, it is not a serious problem. **Because the QR algorithm and the orthogonal QD algorithm also can not compute the correct singular value of the problem in $n = 250000$.**

We can download our implementation of the dqds algorithm in http://www-is.amp.i.kyoto-u.ac.jp/kkimur/LAPROGNC/LAPROGNC.html