Science & Technology Facilities Council
Rutherford Appleton Laboratory

An Introduction to HSL: Design of reliable software

M. Arioli

http://www.numerical.rl.ac.uk/people/marioli/marioli.html

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Outline

- HSL packages
- Multifrontal
- Static pivoting for Multifrontal
- Mixed precision
- Iterative refinement, GMRES and Flexible GMRES
- Numerical experiments

Linear system

We wish to solve large sparse systems

$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad \mathbf{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$$

Linear system

We wish to solve large sparse systems

$$\mathbf{Ax} = \mathbf{b} \qquad \mathbf{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$$

A particular and important case arises in saddle-point problems where

$$A = \begin{bmatrix} H & B \\ B^T & 0 \end{bmatrix} \qquad \left(A = L^T D L\right)$$

Background to HSL Library

- HSL began as Harwell Subroutine Library in 1963. Internal library for those at AERE Harwell. Early contributors included: Mike Powell, Mike Hopper and Alan Curtis.
- Distributed externally for the ïňĄrst time in 1964 and has been constantly developed and available under licence ever since.
- In 2000 older codes were made freely available for research in HSL Archive.
- HSL 2007: **free** academic access for personal research and teaching.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Background to HSL Library

- HSL began as Harwell Subroutine Library in 1963. Internal library for those at AERE Harwell. Early contributors included: Mike Powell, Mike Hopper and Alan Curtis.
- Distributed externally for the ïňĄrst time in 1964 and has been constantly developed and available under licence ever since.
- In 2000 older codes were made freely available for research in HSL Archive.
- HSL 2007: **free** academic access for personal research and teaching.

Since 1970s, one of the main strengths of HSL has been its sparse direct solvers. **First HSL sparse direct solver**: 1971 Curtis and Reid `MA18`
Many well-known packages, including `MA27, MA28, MA42, MA48, MA57 ...`
See www.cse.scitech.ac.uk/nag/hsl

Direct solvers

Advantages of direct methods:

- High accuracy
- Robust. Can be used as black box solvers
- Can cheaply solve for multiple right-hand sides
- Savings can be made if solving a series of problems

Direct solvers

Advantages of direct methods:

- High accuracy
- Robust. Can be used as black box solvers
- Can cheaply solve for multiple right-hand sides
- Savings can be made if solving a series of problems

Disadvantages:

- Memory required grows more rapidly than problem size
- Difficult to code efficently. (Massive) parallelism very hard

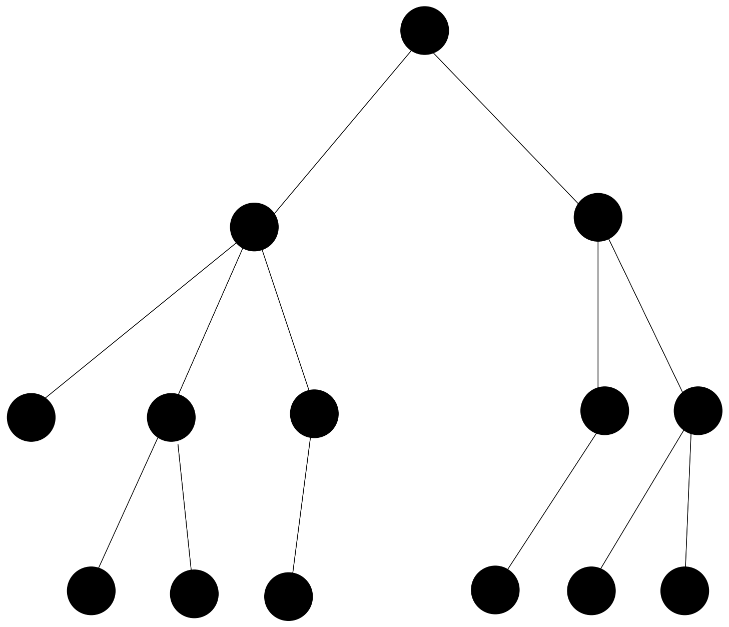
Direct solvers

- Techniques to extend the application of direct methods include:
- Parallel codes (eg `MUMPS, PARDISO, WSMP, SuperLU, ...`).
- Use as a preconditioner eg
  - incomplete factorization
  - use a block form and factorize diagonal blocks with direct solver (in parallel)
  - **static pivoting**
- **Out-of-core storage** (hold matrix data, matrix factors and some work arrays in files). Advantage: disk storage is **cheap**.
- **New**: **mixed precision approach** (single precision factorization then refinement in double precision).

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal method

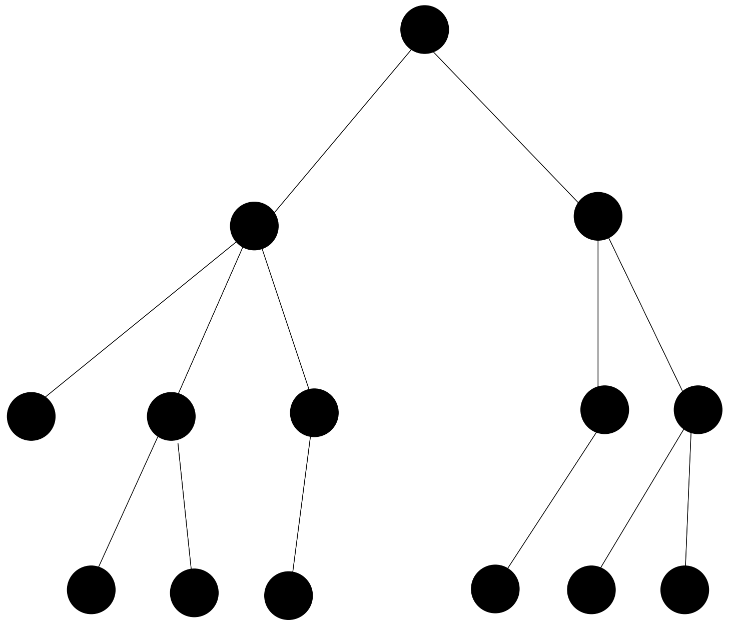
ASSEMBLY TREE



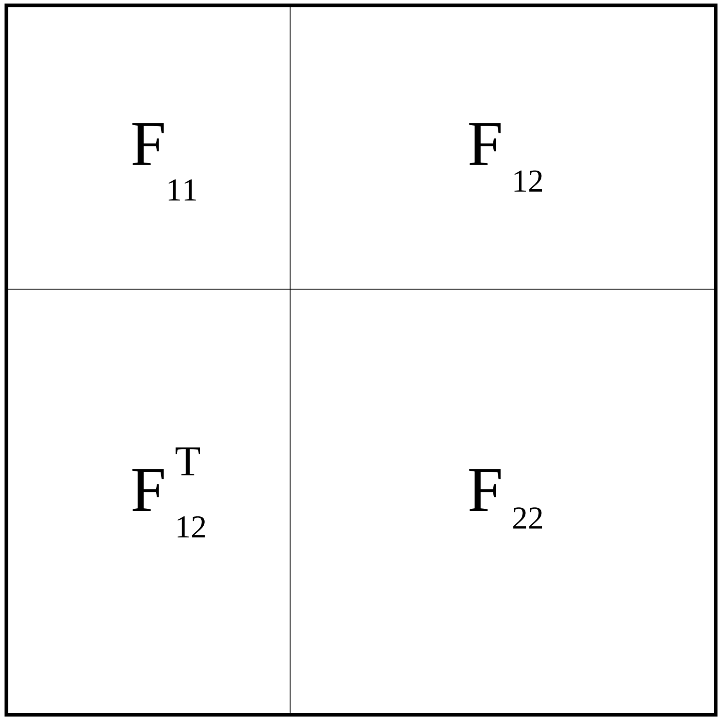
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal method

ASSEMBLY TREE



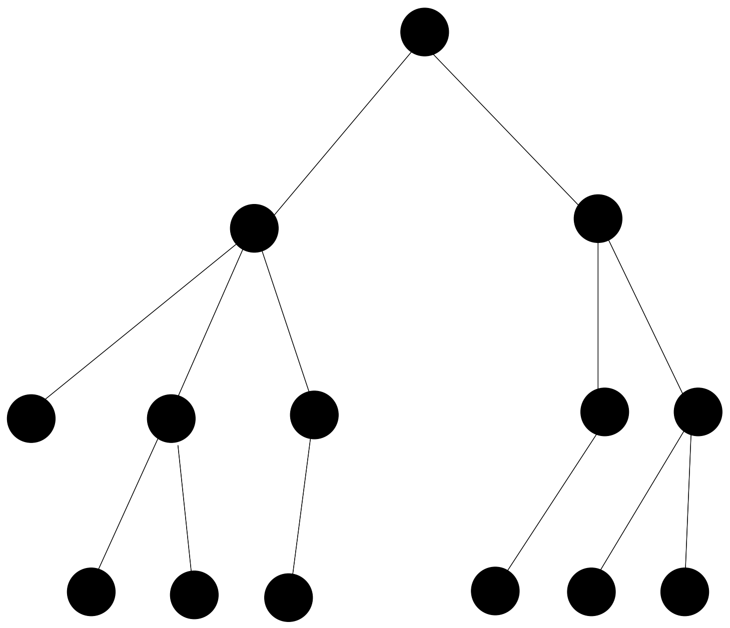
AT EACH NODE



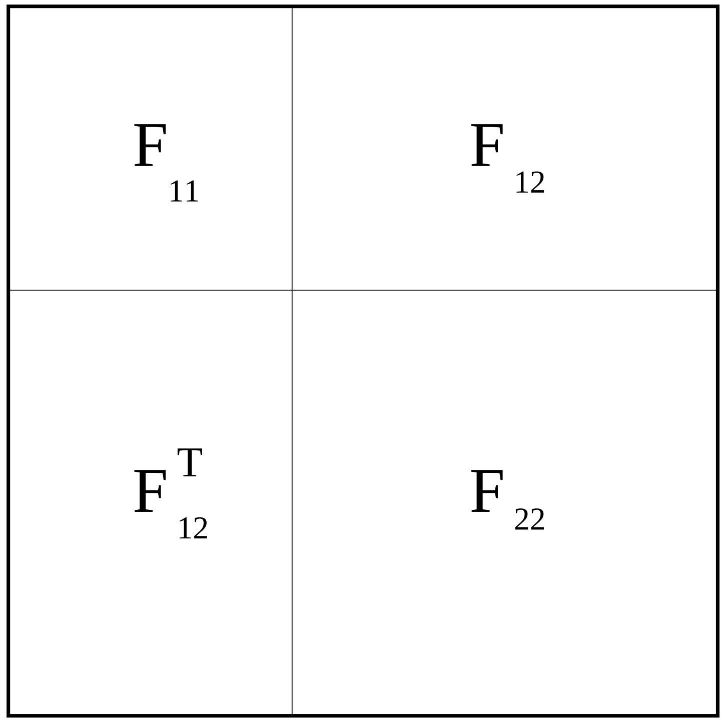
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal method

ASSEMBLY TREE



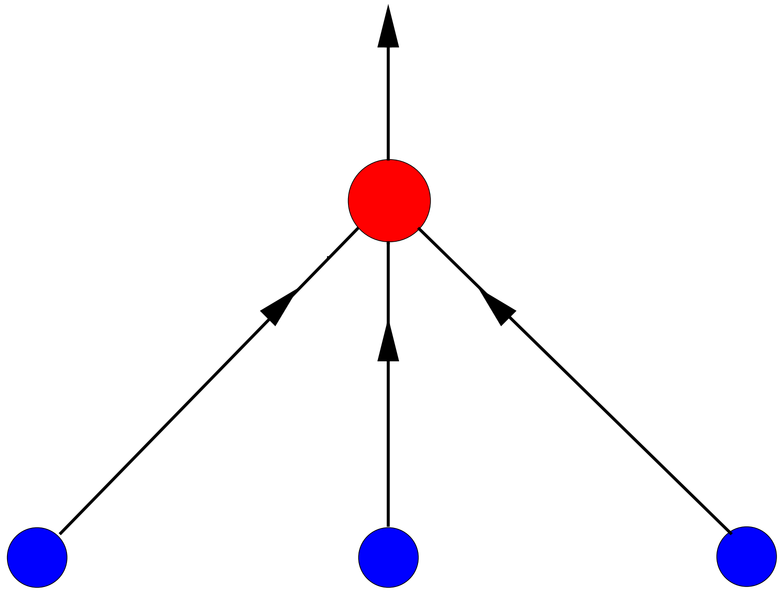
AT EACH NODE



$$F_{22} \leftarrow F_{22} - F_{12}^T F_{11}^{-1} F_{12}$$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

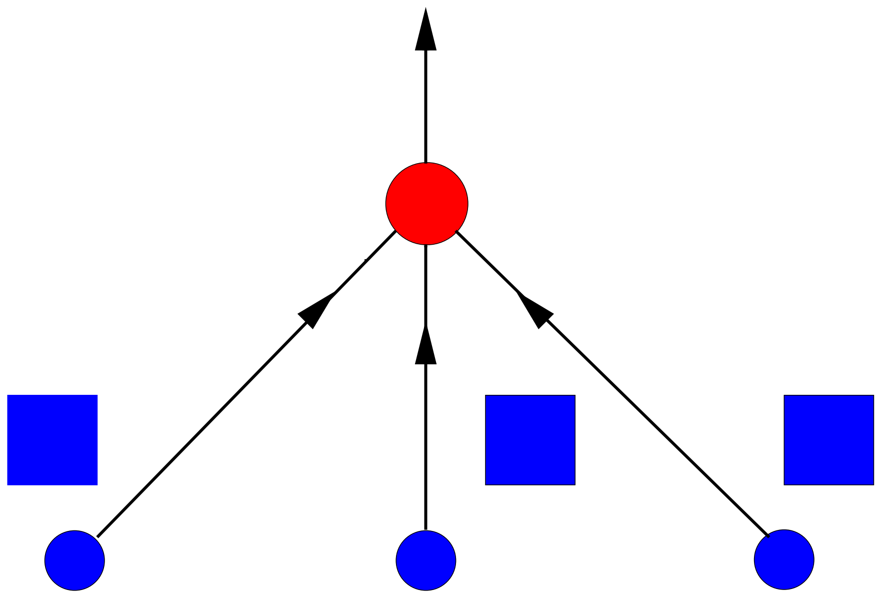
Multifrontal method



- From children to parent

Science & Technology Facilities Council
Rutherford Appleton Laboratory

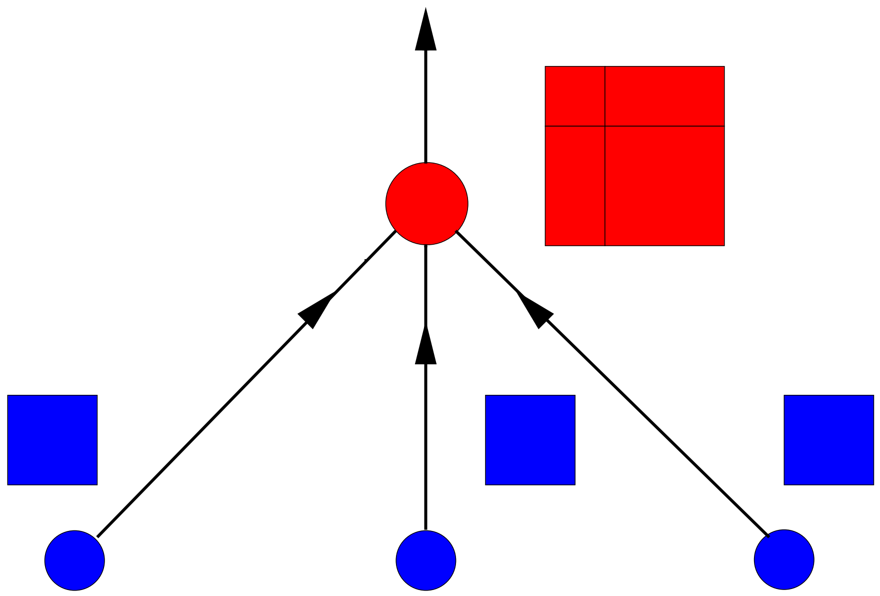
Multifrontal method



- From children to parent
- **ASSEMBLY** Gather/Scatter operations (indirect addressing)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

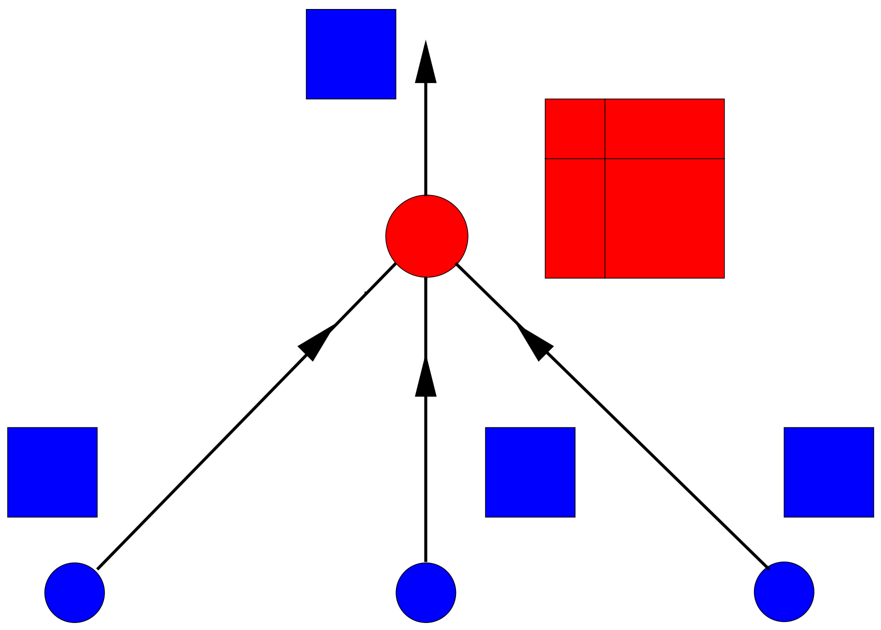
Multifrontal method



- From children to parent
- **ASSEMBLY** Gather/Scatter operations (indirect addressing)
- **ELIMINATION** Full Gaussian elimination, Level 3 BLAS (TRSM, GEMM)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

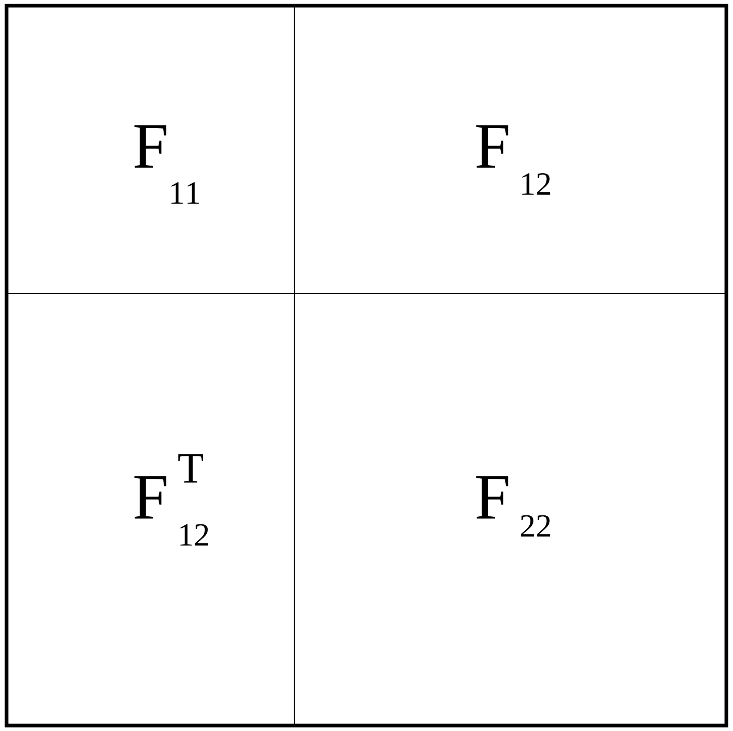
Multifrontal method



- From children to parent
- **ASSEMBLY** Gather/Scatter operations (indirect addressing)
- **ELIMINATION** Full Gaussian elimination, Level 3 BLAS (TRSM, GEMM)

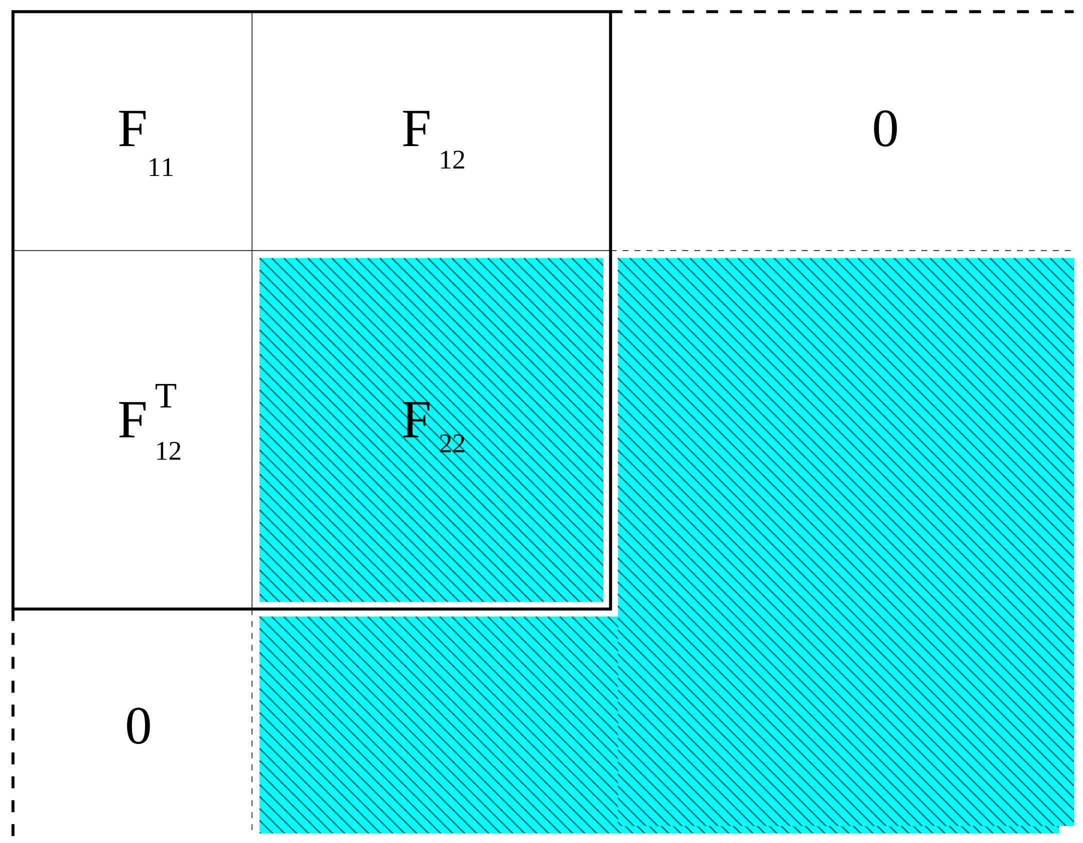
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal method



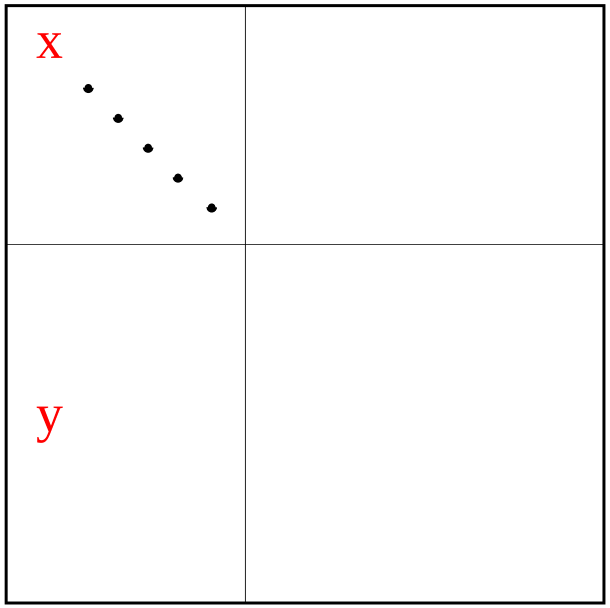
Pivot can only be chosen from F_{11} block since F_{22} is **NOT** fully summed.

Multifrontal method



Situation wrt rest of matrix

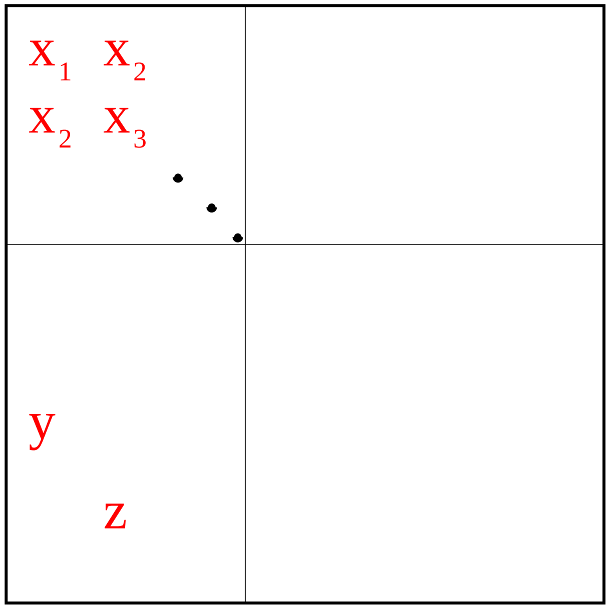
Pivoting (1×1)



Choose x as 1×1 **pivot** if $|x| > u|y|$
where $|y|$ is the largest in column.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Pivoting (2×2)



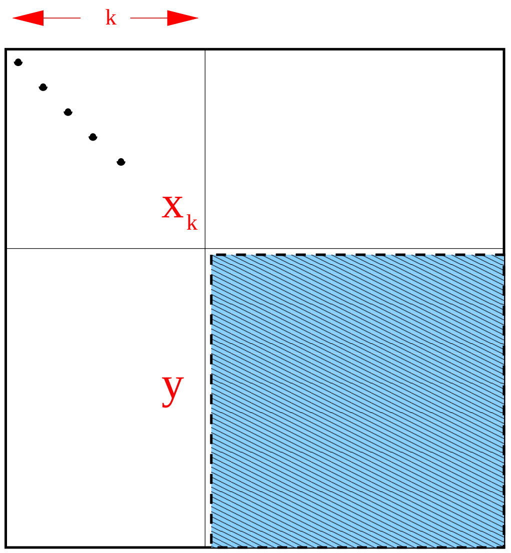
For the indefinite case, we can choose 2×2 **pivot** where we require

$$\left| \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix}^{-1} \right| \begin{bmatrix} |y| \\ |z| \end{bmatrix} \leq \begin{bmatrix} \frac{1}{u} \\ \frac{1}{u} \end{bmatrix}$$

where again $|y|$ and $|z|$ are the largest in their columns.

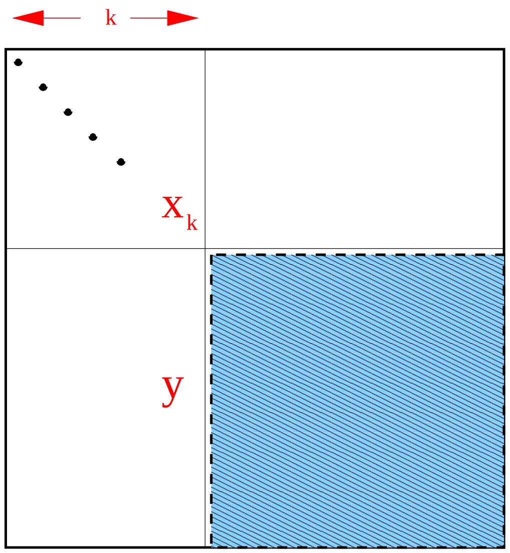
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Pivoting



If we assume that $k-1$ pivots are chosen but $|x_k| < u|y|$:

Pivoting

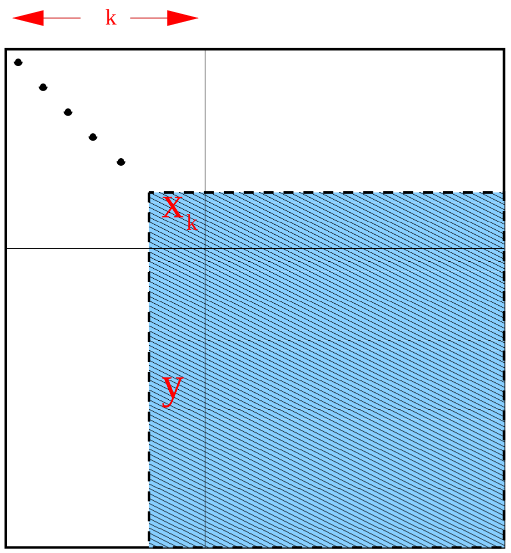


If we assume that $k-1$ pivots are chosen but $|x_k| < u|y|$:

- we can either take the **RISK** and use it or

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Pivoting

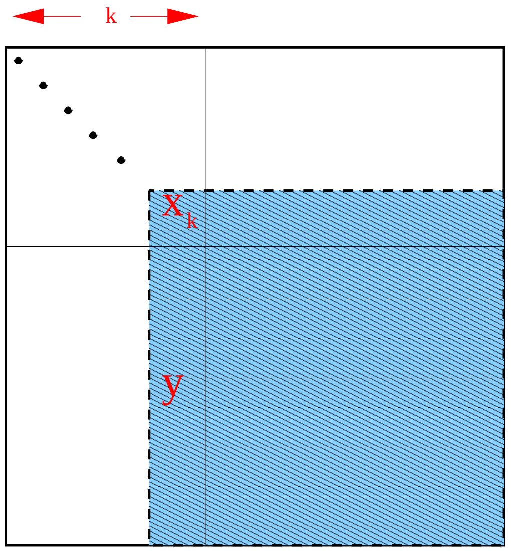


If we assume that $k-1$ pivots are chosen but $|x_k| < u|y|$:

- we can either take the **RISK** and use it or
- **DELAY** the pivot and then send to the parent a larger Schur complement.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Pivoting



If we assume that $k-1$ pivots are chosen but $|x_k| < u|y|$:

- we can either take the **RISK** and use it or
- **DELAY** the pivot and then send to the parent a larger Schur complement.

This can cause more work and storage

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Static Pivoting

An **ALTERNATIVE** is to use **Static Pivoting**, by replacing x_k by

$$x_k + \tau$$

and CONTINUE.

Static Pivoting

An **ALTERNATIVE** is to use **Static Pivoting**, by replacing x_k by

$$x_k + \tau$$

and CONTINUE.

This is even more important in the case of parallel implementation where static data structures are often preferred

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Static Pivoting

Several codes use (or have an option for) this device:

- SuperLU (Demmel and Li)
- PARDISO (Gärtner and Schenk)
- MA57 (Duff and Pralet)

Static Pivoting

We thus have factorized

$$A + E = LDL^T = M$$

where $|E| \leq \tau I$

Static Pivoting

We thus have factorized

$$A + E = LDL^T = M$$

where $|E| \leq \tau I$

The three codes then have an **Iterative Refinement** option.
IR will converge if $\rho(M^{-1}E) < 1$

Static Pivoting

Choosing τ

Static Pivoting

Choosing τ

Increase $\tau \Longrightarrow$ increase stability of decomposition

Static Pivoting

Choosing τ

Increase $\tau \Longrightarrow$ increase stability of decomposition

Decrease $\tau \Longrightarrow$ better approximation of the original matrix, reduces $||E||$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Static Pivoting

Choosing τ

Increase $\tau \Longrightarrow$ increase stability of decomposition

Decrease $\tau \Longrightarrow$ better approximation of the original matrix, reduces $||E||$

Trade-off

- $\approx \varepsilon \Longrightarrow$ big growth in preconditioning matrix M
- $\approx 1 \Longrightarrow$ huge error $||E||$.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Static Pivoting

Choosing τ

Increase $\tau \Longrightarrow$ increase stability of decomposition

Decrease $\tau \Longrightarrow$ better approximation of the original matrix, reduces $||E||$

Trade-off

- $\approx \varepsilon \Longrightarrow$ big growth in preconditioning matrix M
- $\approx 1 \Longrightarrow$ huge error $||E||$.

Conventional wisdom is to choose

$$\tau = \mathcal{O}(\sqrt{\varepsilon})$$

Static Pivoting

Choosing τ

Increase $\tau \implies$ increase stability of decomposition

Decrease $\tau \implies$ better approximation of the original matrix, reduces $||E||$

Trade-off

- $\approx \varepsilon \implies$ big growth in preconditioning matrix M
- $\approx 1 \implies$ huge error $||E||$.

Conventional wisdom is to choose

$$\tau = \mathcal{O}(\sqrt{\varepsilon})$$

In real life $\rho(M^{-1}E) > 1$ but we have a plan **B** FGMRES

GMRES and FGMRES

Let $r_0 = b - Ax_0$ and $\mathcal{K}_k(A, r_0)$ be the usual Krylov space

GMRES

$$\min_{x \in x_0 + \mathcal{K}_k(A,r_0)} ||r_0 - Ax||_2 \qquad r_0 - Ax_k \perp A\mathcal{K}_k(A, r_0)$$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

GMRES and FGMRES

Let $r_0 = b - Ax_0$ and $\mathcal{K}_k(A, r_0)$ be the usual Krylov space
GMRES

$$\min_{x \in x_0 + \mathcal{K}_k(A, r_0)} ||r_0 - Ax||_2 \qquad r_0 - Ax_k \perp A\mathcal{K}_k(A, r_0)$$

GMRES Right preconditioning

$$AM^{-1}y = b \quad \begin{cases} (AM^{-1}, r_0) \longrightarrow (A, r_0) \\ \mathcal{K}_k(AM^{-1}, r_0) \longrightarrow \mathcal{K}_k(A, r_0) \\ x_k = M^{-1}y_k \\ AM^{-1}V_k = V_{k+1}H_k \end{cases}$$

GMRES and FGMRES

Let $r_0 = b - Ax_0$ and $\mathcal{K}_k(A, r_0)$ be the usual Krylov space
GMRES

$$\min_{x \in x_0 + \mathcal{K}_k(A, r_0)} ||r_0 - Ax||_2 \qquad r_0 - Ax_k \perp A\mathcal{K}_k(A, r_0)$$

GMRES Right preconditioning

$$AM^{-1}y = b \quad \left\{ \begin{array}{l} (AM^{-1}, r_0) \longrightarrow (A, r_0) \\ \mathcal{K}_k(AM^{-1}, r_0) \longrightarrow \mathcal{K}_k(A, r_0) \\ x_k = M^{-1}y_k \\ AM^{-1}V_k = V_{k+1}H_k \end{array} \right.$$

Flexible GMRES Right preconditioning

$$Z_k \longrightarrow \mathcal{K}_k(A, r_0) \; x_k = x_0 + Z_k y_k \quad AZ_k = V_{k+1}H_k$$

$$Z_k = span(r_0, AM_1^{-1}r_0, \ldots, \left(\prod_{j=0}^{k-1} AM_j^{-1} \right) r_0)$$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Right preconditioned GMRES and Flexible GMRES

```
procedure [x] = right_Prec_GMRES(A,M,b)
        $x_0 = M^{-1}b, r_0 = b - Ax_0$ and $\beta = ||r_0||$
        $v_1 = r_0/\beta$; k = 0;
        while $||r_k|| > \mu(||b|| + ||A||\ ||x_k||)$
            $k = k + 1;$
            $z_k = M^{-1}v_k; w = Az_k;$
            for $i = 1, \ldots, k$ do
                $h_{i,k} = v_i^T w;$
                $w = w - h_{i,k}v_i;$
            end for;
            $h_{k+1,k} = ||w||;$
            $v_{k+1} = w/h_{k+1,k};$
            $V_k = [v_1, \ldots, v_k];$
            $H_k = \{h_{i,j}\}_{1\le i\le j+1; 1\le j\le k};$
            $y_k = \arg\min_y ||\beta e_1 - H_k y||;$
            $x_k = x_0 + M^{-1}V_k y_k$ and $r_k = b - Ax_k;$
        end while ;
end procedure.
```

```
procedure [x] =FGMRES(A,$M_i$,b)
        $x_0 = M_0^{-1}b, r_0 = b - Ax_0$ and $\beta = ||r_0||$
        $v_1 = r_0/\beta$; k = 0;
        while $||r_k|| > \mu(||b|| + ||A||\ ||x_k||)$
            $k = k + 1;$
            $z_k = M_k^{-1}v_k; w = Az_k;$
            for $i = 1, \ldots, k$ do
                $h_{i,k} = v_i^T w;$
                $w = w - h_{i,k}v_i;$
            end for;
            $h_{k+1,k} = ||w||;$
            $v_{k+1} = w/h_{k+1,k};$
            $Z_k = [z_1, \ldots, z_k]; V_k = [v_1, \ldots, v_k];$
            $H_k = \{h_{i,j}\}_{1\le i\le j+1; 1\le j\le k};$
            $y_k = \arg\min_y ||\beta e_1 - H_k y||;$
            $x_k = x_0 + Z_k y_k$ and $r_k = b - Ax_k;$
        end while ;
end procedure.
```

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Roundoff error

The roundoff error analysis of both FGMRES and GMRES can be done in three stages:

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Roundoff error

The roundoff error analysis of both FGMRES and GMRES can be done in three stages:

1. Error analysis of the Arnoldi-Krylov process (Giraud and Langou, Björck and Paige, and Paige, Rozložník, and Strakoš).
   MGS applied to

$$z_1 = M_1^{-1} r_0 / ||r_0||, \quad z_j = M_j^{-1} v_j$$

$$C^{(k)} = (r_0, Az_1, Az_2, \ldots, Az_k) = V_{k+1} R_k$$

$$R_k = \begin{bmatrix} ||r_0|| & H_{1,1} & \ldots & H_{1,k} \\ 0 & H_{2,1} & \ldots & H_{2,k} \\ 0 & 0 & \ldots & H_{3,k} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & H_{k+1,k} \end{bmatrix}$$

Roundoff error

The roundoff error analysis of both FGMRES and GMRES can be done in three stages:

1. Error analysis of the Arnoldi-Krylov process (Giraud and Langou, Björck and Paige, and Paige, Rozložník, and Strakoš).
2. Error analysis of the Givens process used on the upper Hessenberg matrix H_k in order to reduce it to upper triangular form.

Roundoff error

The roundoff error analysis of both FGMRES and GMRES can be done in three stages:

1. Error analysis of the Arnoldi-Krylov process (Giraud and Langou, Björck and Paige, and Paige, Rozložník, and Strakoš).
2. Error analysis of the Givens process used on the upper Hessenberg matrix H_k in order to reduce it to upper triangular form.
3. Error analysis of the computation of x_k in FGMRES and GMRES.

Roundoff error

The roundoff error analysis of both FGMRES and GMRES can be done in three stages:

1. Error analysis of the Arnoldi-Krylov process (Giraud and Langou, Björck and Paige, and Paige, Rozložník, and Strakoš).
2. Error analysis of the Givens process used on the upper Hessenberg matrix H_k in order to reduce it to upper triangular form.
3. Error analysis of the computation of x_k in FGMRES and GMRES.

The first two stages of the roundoff error analysis are the same for both FGMRES and GMRES. The last stage is specific to each algorithm.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Roundoff error analysis of FGMRES

Theorem 1. If we apply FGMRES to solve $Ax = b$, using finite-precision arithmetic conforming to IEEE standard with relative precision ε and under the hypotheses:

$$2.12(n+1)\varepsilon < 0.01 \qquad \text{and} \qquad c_0(n)\varepsilon\kappa(C^{(k)}) < 0.1 \ \forall k$$

where

$$c_0(n) = 18.53 n^{\frac{3}{2}}$$

and

$$|\bar{s}_k| < 1 - \varepsilon, \ \forall k,$$

where $\bar{s}_k$ are the sines computed during the Givens algorithm applied to $\bar{H}_k$ in order to compute $\bar{y}_k$, then there exists $\hat{k}, \ \hat{k} \le n$ such that, $\forall k \ge \hat{k}$, we have

$$\begin{aligned} ||b - A\bar{x}_k|| \quad &\le \quad c_1(n,k)\varepsilon\Big(||b|| + ||A|| \, ||\bar{x}_0|| + \\ & ||A|| \, || \, |\bar{Z}_k| \, |\bar{y}_k| \, || + ||A\bar{Z}_k|| \, ||\bar{y}_k|| \Big) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

(Arioli, Duff, Gratton, and Pralet SISC 2007), (Arioli and Duff. ETNA 2008)

Multifrontal approach: `HSL_MA57`

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal approach: HSL_MA57

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal approach: `HSL_MA57`

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in

Multifrontal approach: `HSL_MA57`

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.

Multifrontal approach: `HSL_MA57`

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.
- We thus have factorized $A + E = LDL^T = M$ where $|E| \le \tau I$

Multifrontal approach: HSL_MA57

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.
- We thus have factorized $A + E = LDL^T = M$ where $|E| \leq \tau I$
- Several codes use (or have an option for) this device:

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal approach: HSL_MA57

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.
- We thus have factorized $A + E = LDL^T = M$ where $|E| \leq \tau I$
- Several codes use (or have an option for) this device:
  - SuperLU (Demmel and Li)

Multifrontal approach: HSL_MA57

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.
- We thus have factorized $A + E = LDL^T = M$ where $|E| \leq \tau I$
- Several codes use (or have an option for) this device:
  - SuperLU (Demmel and Li)
  - PARDISO (Gärtner and Schenk)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Multifrontal approach: `HSL_MA57`

In order to reduce the fill-in during the LDL^T factorization

- We scale and reorder the entries of A
- We weaken the numerical pivot strategy by using a threshold
- However, also this can be unsatisfactory: the numerical pivot strategy is still disrupting the ordering we have chosen and increases the fill-in
- An **ALTERNATIVE** is to use **Static Pivoting**, by replacing the pivot a_k failing the test by $a_k + \tau$ and CONTINUE.
- We thus have factorized $A + E = LDL^T = M$ where $|E| \leq \tau I$
- Several codes use (or have an option for) this device:
  - SuperLU (Demmel and Li)
  - PARDISO (Gärtner and Schenk)
  - MA57 (Duff and Pralet)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Why FGMRES for symmetric case?

- The computed values of Gaussian factorization $\hat{L}$ $\hat{D}$ are affected by roundoff: $M = \hat{L}\hat{D}\hat{L}^T$ and $||E|| = ||M - A|| \leq c(n)\varepsilon||A||$ with $E \neq E^T$

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Why FGMRES for symmetric case?

- The computed values of Gaussian factorization $\hat{L}$ $\hat{D}$ are affected by roundoff: $M = \hat{L}\hat{D}\hat{L}^T$ and $||E|| = ||M - A|| \leq c(n)\varepsilon||A||$ with $E \neq E^T$
- Thus $M^{-1}A \neq AM^{-1}$ and the preconditioned matrix is non symmetric

Why FGMRES for symmetric case?

- The computed values of Gaussian factorization $\hat{L}$ $\hat{D}$ are affected by roundoff: $M = \hat{L}\hat{D}\hat{L}^T$ and $||E|| = ||M - A|| \leq c(n)\varepsilon||A||$ with $E \neq E^T$
- Thus $M^{-1}A \neq AM^{-1}$ and the preconditioned matrix is non symmetric
- FGMRES is then the only way

GMRES vs FGMRES

- GMRES error bounds depend on $\| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, \|$. (Arioli, Duff, Gratton, and Pralet SISC 2007)

GMRES vs FGMRES

- GMRES error bounds depend on $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$. (Arioli, Duff, Gratton, and Pralet SISC 2007)
- For sparse matrices $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$ can be much larger than $||A||$.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

GMRES vs FGMRES

- GMRES error bounds depend on $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$. (Arioli, Duff, Gratton, and Pralet SISC 2007)
- For sparse matrices $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$ can be much larger than $||A||$.
- For the static pivot the growth can be dramatic.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

GMRES vs FGMRES

- GMRES error bounds depend on $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$. (Arioli, Duff, Gratton, and Pralet SISC 2007)
- For sparse matrices $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$ can be much larger than $||A||$.
- For the static pivot the growth can be dramatic.
- Theorem 1 shows that FGMRES does not depend on $|| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, ||$.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Test Problems

	n	nnz	Description
CONT_201	80595	239596	KKT matrix Convex QP (M2)
CONT_300	180895	562496	KKT matrix Convex QP (M2)
TUMA_1	22967	76199	Mixed-Hybrid finite-element

Test problems

Science & Technology Facilities Council
Rutherford Appleton Laboratory

MA57 tests

	n	nnz(L)+nnz(D)	Factorization time
CONT_201	80595	9106766	9.0 sec
CONT_300	180895	22535492	28.8 sec

MA57 without static pivot

Science & Technology Facilities Council
Rutherford Appleton Laboratory

MA57 tests

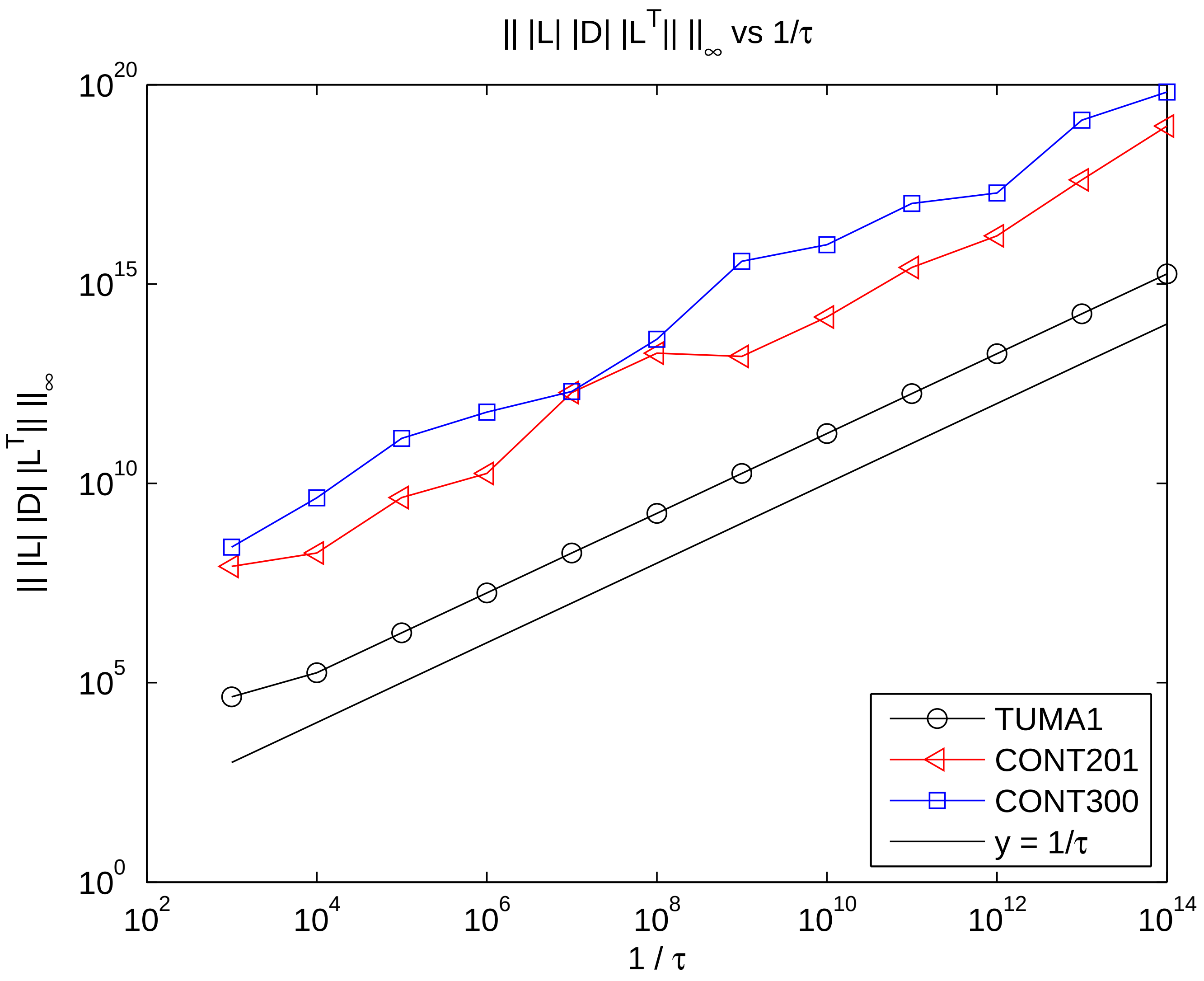
	n	nnz(L)+nnz(D)	Factorization time
CONT_201	80595	9106766	9.0 sec
CONT_300	180895	22535492	28.8 sec

MA57 without static pivot

	nnz(L)+nnz(D)+ FGMRES (#it)	Factorization time	# static pivots
CONT_201	5563735 (6)	3.1 sec	27867
CONT_300	12752337 (8)	8.9 sec	60585

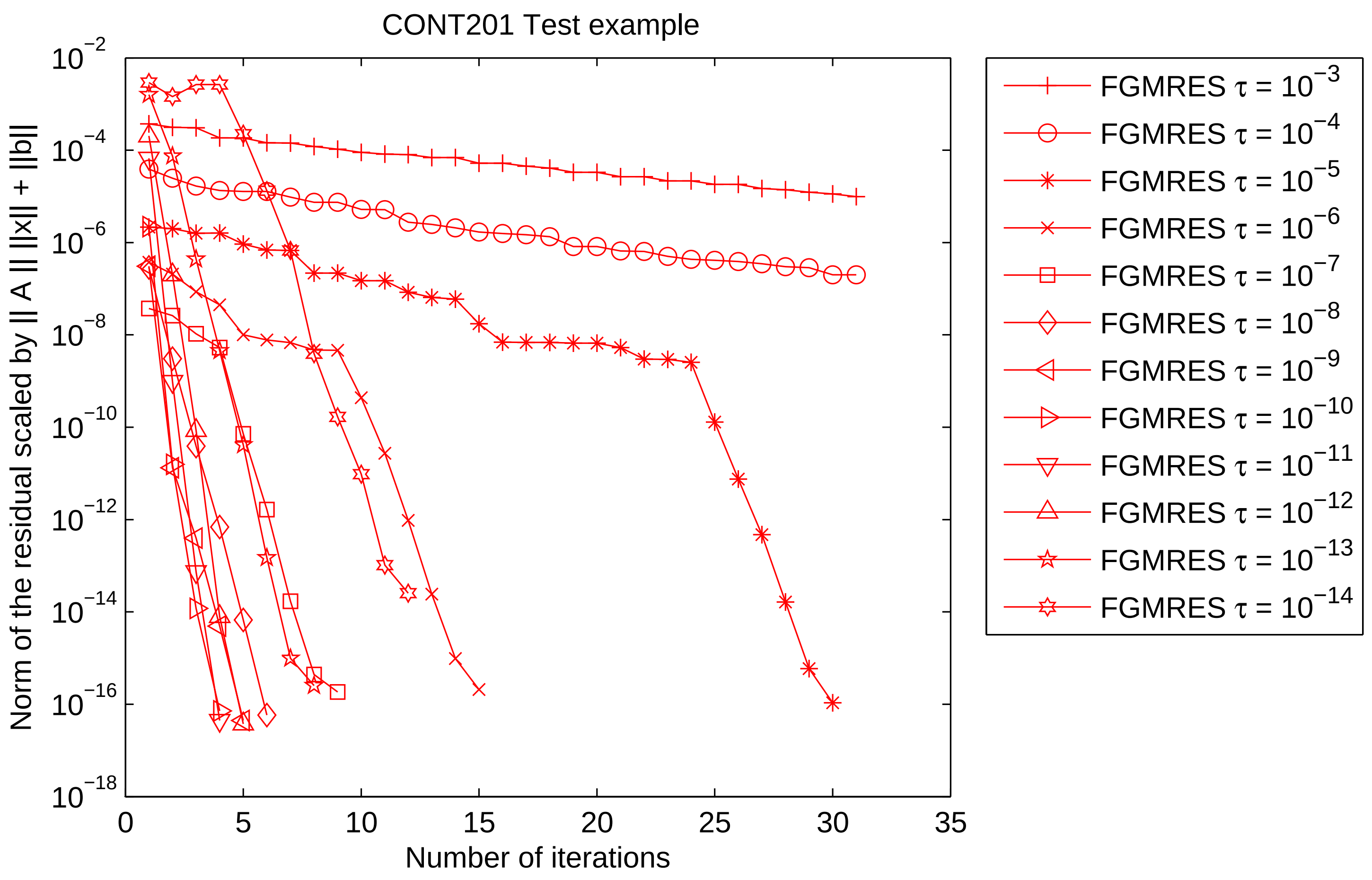
MA57 with static pivot $\tau = 10^{-8}$

$\| \, |\hat{L}| \, |\hat{D}| \, |\hat{L}^T| \, \|$ vs $1/\tau$



Science & Technology Facilities Council
Rutherford Appleton Laboratory

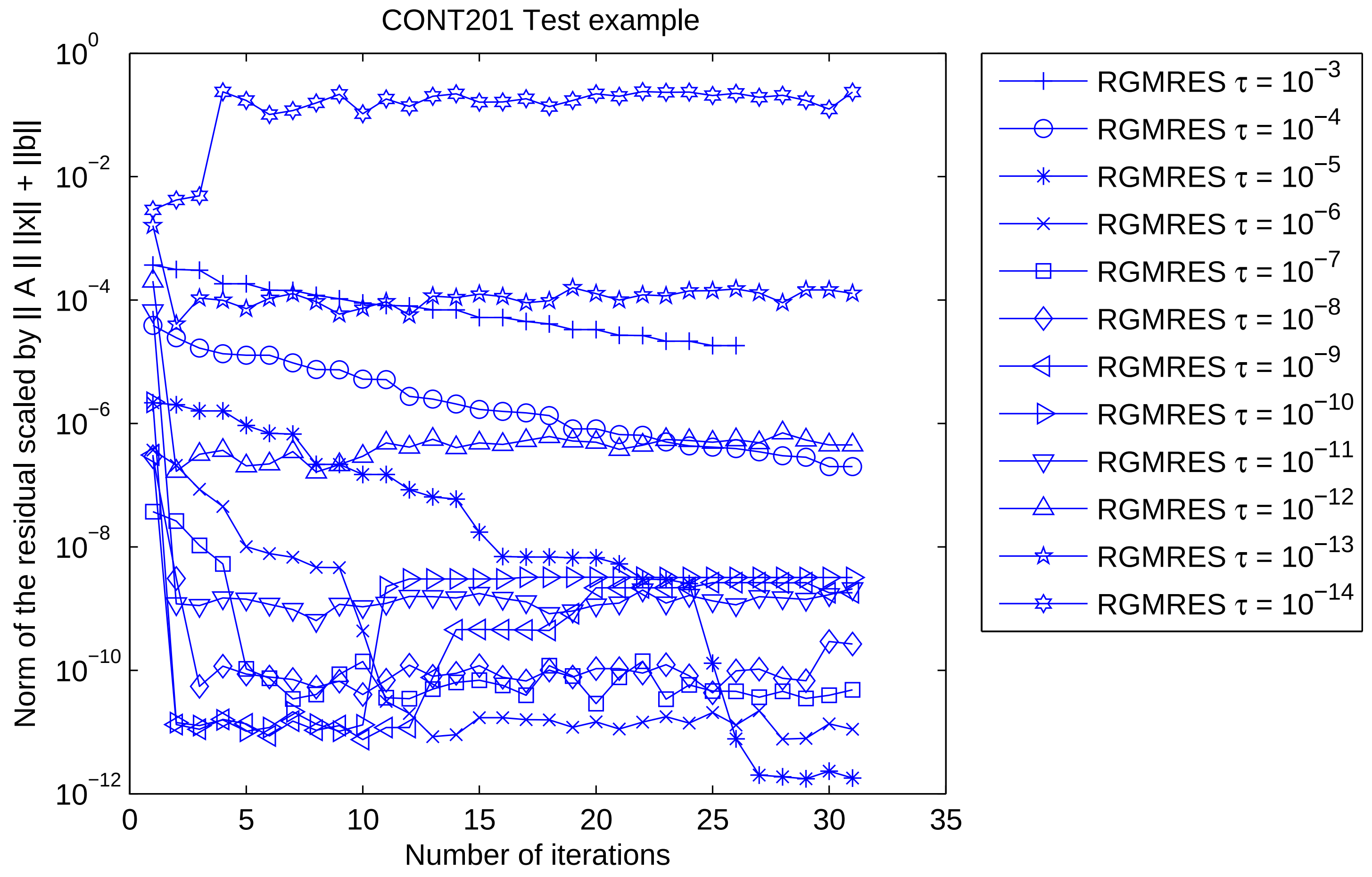
Numerical experiments



FGMRES on CONT-201 test example

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Numerical experiments



GMRES on CONT-201 test example

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Mixed precision arithmetic

- Very fast 32-bit arithmetic unit

(Arioli and Duff. ETNA 2008)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Mixed precision arithmetic

- Very fast 32-bit arithmetic unit
- We use 32-bit arithmetic for factorization and triangular solves
  M is the $fl(LU)$ of A and $||M - A|| \leq c(N)\sqrt{\varepsilon}||A||$
  $(\varepsilon = 2.2 \times 10^{-16})$

(Arioli and Duff. ETNA 2008)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Mixed precision arithmetic

- Very fast 32-bit arithmetic unit
- We use 32-bit arithmetic for factorization and triangular solves
  M is the $fl(LU)$ of A and $||M - A|| \leq c(N)\sqrt{\varepsilon}||A||$
  $(\varepsilon = 2.2 \times 10^{-16})$
- If $\kappa(A)\sqrt{\varepsilon} > 1$ then Iterative Refinement may not converge. FGMRES does

(Arioli and Duff. ETNA 2008)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Mixed precision arithmetic

- Very fast 32-bit arithmetic unit
- We use 32-bit arithmetic for factorization and triangular solves
  M is the $fl(LU)$ of A and $||M - A|| \leq c(N)\sqrt{\varepsilon}||A||$
  $(\varepsilon = 2.2 \times 10^{-16})$
- If $\kappa(A)\sqrt{\varepsilon} > 1$ then Iterative Refinement may not converge. FGMRES does
- FGMRES backward stable (Arioli and Duff. ETNA 2008)

Mixed precision arithmetic

- Very fast 32-bit arithmetic unit
- We use 32-bit arithmetic for factorization and triangular solves
  M is the $fl(LU)$ of A and $||M - A|| \leq c(N)\sqrt{\varepsilon}||A||$
  $(\varepsilon = 2.2 \times 10^{-16})$
- If $\kappa(A)\sqrt{\varepsilon} > 1$ then Iterative Refinement may not converge. FGMRES does
- FGMRES backward stable (Arioli and Duff. ETNA 2008)
- GMRES is not always backward stable

Test Environment using mixed precision

- Selected Sparse Matrices

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Test Environment using mixed precision

- Selected Sparse Matrices
- Forward and backward substitution

Test Environment using mixed precision

- Selected Sparse Matrices
- Forward and backward substitution
  - the vector $\bar{z}_k$ is computed using the forward and backward substitution algorithm in single precision on the single precision conversion of vector $\bar{v}_k$,
  - the vector $\bar{z}_k$ is computed using the forward and backward substitution algorithm in double precision on $\bar{v}_k$ after we converted the factors L and U to double precision.

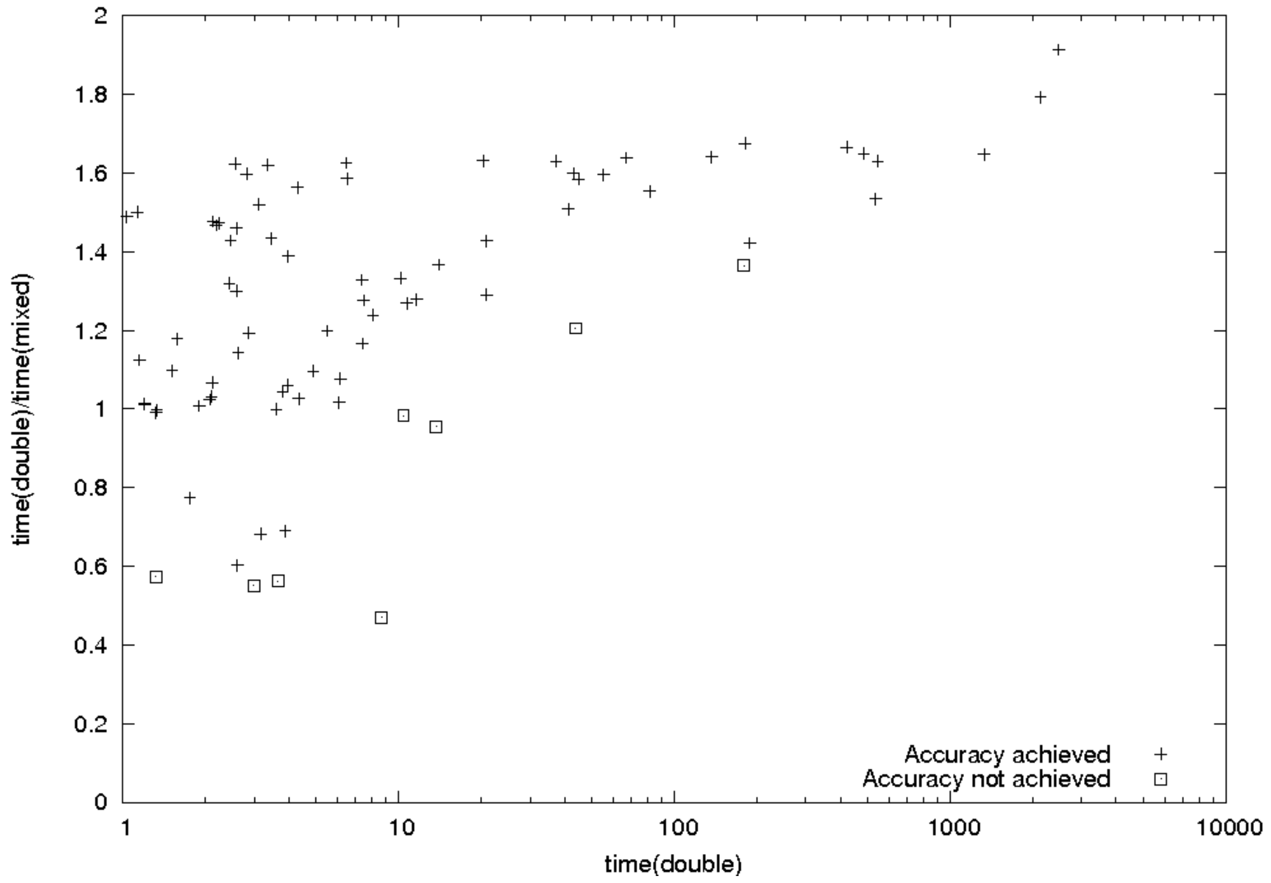
Science & Technology Facilities Council
Rutherford Appleton Laboratory

MA57 sparse tests using mixed precision

Matrix Id	n	Iterative refinement		FGMRES				
		Total It	SR	Total It	Inner it	SR	$\|\|A\bar{Z}_{\hat{k}}\|\|$	$\|\|\,\|\bar{Z}_{\hat{k}}\|\,\|\bar{y}_{\hat{k}}\|\,\|\|$
bcsstk20	485	30	2.1e-15	2	2	1.4e-11	1.7e+00	4.6e+02
$\kappa(A) = 5.\,10^9$				4	2	3.4e-14	1.6e+00	3.8e-01
				6	2	7.2e-17	1.6e+00	5.6e-04
bcsstm27	1224	22	1.6e-15	2	2	5.8e-11	1.7e+00	2.7e+01
$\kappa(A) = 5.\,10^9$				4	2	1.8e-11	6.3e-01	1.3e+00
				6	2	6.0e-13	2.0e+00	7.6e-02
				8	2	1.5e-13	1.7e+00	1.0e-02
				10	2	1.2e-14	1.7e+00	1.9e-03
				12	2	2.6e-15	1.8e+00	1.7e-04
				14	2	1.8e-16	1.6e+00	4.3e-05
s3rmq4m1	5489	16	2.2e-15	2	2	3.5e-11	1.0e+00	8.6e+01
$\kappa(A) = 4.\,10^9$				4	2	2.1e-13	1.1e+00	3.2e-01
				6	2	4.5e-15	1.7e+00	6.4e-03
				8	2	1.1e-16	1.6e+00	1.3e-04
s3dkq4m2	90449	53	1.1e-10	10	10	6.3e-17	1.2e+00	1.2e+03
$\kappa(A) = 7.\,10^{10}$								

Sparse matrices results ($SR = \dfrac{||b - A\overline{\mathbf{x}}_{\hat{k}}||}{(||A||\,||\overline{\mathbf{x}}_{\hat{k}}|| + ||b||)}$)

MA57 sparse tests using mixed precision



Ratio of times to solve linear system in mixed precision and double precision on a test set of 78 sparse problems with a scaled residual

$$\frac{||b - A\overline{\mathbf{x}}_{\hat{k}}||}{(||A||\,||\overline{\mathbf{x}}_{\hat{k}}|| + ||b||)} \leq 5 \times 10^{-15}.$$

Out-of-core solvers

Idea of out-of-core solvers **not** new: band and frontal solvers developed in 1970s and 1980s held matrix data and factors out-of-core eg `MA32` (later superseded by `MA42`).

Other codes with out-of-core options include `BCSEXT-LIB`, `Oblio`, `TAUCS`.

`MUMPS` team currently developing out-of-core version.

Out-of-core solvers

Idea of out-of-core solvers **not** new: band and frontal solvers developed in 1970s and 1980s held matrix data and factors out-of-core eg `MA32` (later superseded by `MA42`).

Other codes with out-of-core options include `BCSEXT-LIB`, `Oblio`, `TAUCS`.

`MUMPS` team currently developing out-of-core version.

Our new out-of-core solver for **LARGE** sparse symmetric systems, both positive definite and indefinite, is `HSL_MA77` (Reid and Scott).

Key features of `HSL_MA77`

- Multifrontal code written in Fortran 95
- Matrix data, matrix factor, and the multifrontal stack are optionally held in files

Key features of `HSL_MA77`

- Multifrontal code written in Fortran 95
- Matrix data, matrix factor, and the multifrontal stack are optionally held in files
- Matrix $\mathbf{A}$ may be either in assembled form or a sum of element matrices

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Key features of HSL_MA77

- Multifrontal code written in Fortran 95
- Matrix data, matrix factor, and the multifrontal stack are optionally held in files
- Matrix $\mathbf{A}$ may be either in assembled form or a sum of element matrices
- **Reverse communication interface** with input by rows or by elements

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Key features of HSL_MA77

- Multifrontal code written in Fortran 95
- Matrix data, matrix factor, and the multifrontal stack are optionally held in files
- Matrix **A** may be either in assembled form or a sum of element matrices
- **Reverse communication interface** with input by rows or by elements
- Separate calls for each phase (input data, analyse, factorize, solve, residual, restart).

Key features of `HSL_MA77`

- Multifrontal code written in Fortran 95
- Matrix data, matrix factor, and the multifrontal stack are optionally held in files
- Matrix $\mathbf{A}$ may be either in assembled form or a sum of element matrices
- **Reverse communication interface** with input by rows or by elements
- Separate calls for each phase (input data, analyse, factorize, solve, residual, restart).
- Additional flexibility through control parameters (default settings minimize decisions user must make)

Dense linear algebra kernels

- At the heart of the multifrontal method is the partial factorization of dense frontal matrices
- We have developed separate packages to perform these factorizations (and partial solves)
  - `HSL_MA54` for positive definite problems
  - `HSL_MA64` for indefinite problems (by default uses threshold partial pivoting with 1×1 and 2×2 pivots)

Dense linear algebra kernels

- At the heart of the multifrontal method is the partial factorization of dense frontal matrices
- We have developed separate packages to perform these factorizations (and partial solves)
  - `HSL_MA54` for positive definite problems
  - `HSL_MA64` for indefinite problems (by default uses threshold partial pivoting with 1×1 and 2×2 pivots)

Advantages:

- Modular design helps with readability, testing, maintenance etc
- Kernels can also be reused in other solvers
- Kernels use blocking and exploit Level 3 BLAS. **Highly efficient**
- Performance can be tuned for computing environment (`OpenMP` version currently being tested)

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Virtual memory management

For `HSL_MA77` to perform well, the i/o **must** be efficient.

We have developed a separate Fortran 95 package `HSL_OF01` to handle all i/o

- `HSL_OF01` provides read/write facilities for one or more direct access files through a single **in-core buffer** (work array)
- The buffer is divided into fixed length pages ... a page is the same length as a record in the direct access file
- Efficiency acheived by careful handling of the buffer within `HSL_OF01` to avoid actual i/o operations whenever possible eg.
  - All wanted pages that are in buffer are accessed before those that are not
  - When a page is freed, only written to file if it has changed

Virtual memory management

Each set of data (such as the reals in the matrix and its factor) is accessed as a **virtual array** i.e. as if it were a very long array

- Most active pages of the virtual array are held in the buffer
- Any contiguous section of the virtual array may be read or written
- Each virtual array is associated with a **primary file**
- For very large problems, the virtual array may be too large for a single file so **secondary files** are used: this is all handled **automatically.**

  Note: primary and secondary files can be held on different devices

Science & Technology Facilities Council
Rutherford Appleton Laboratory

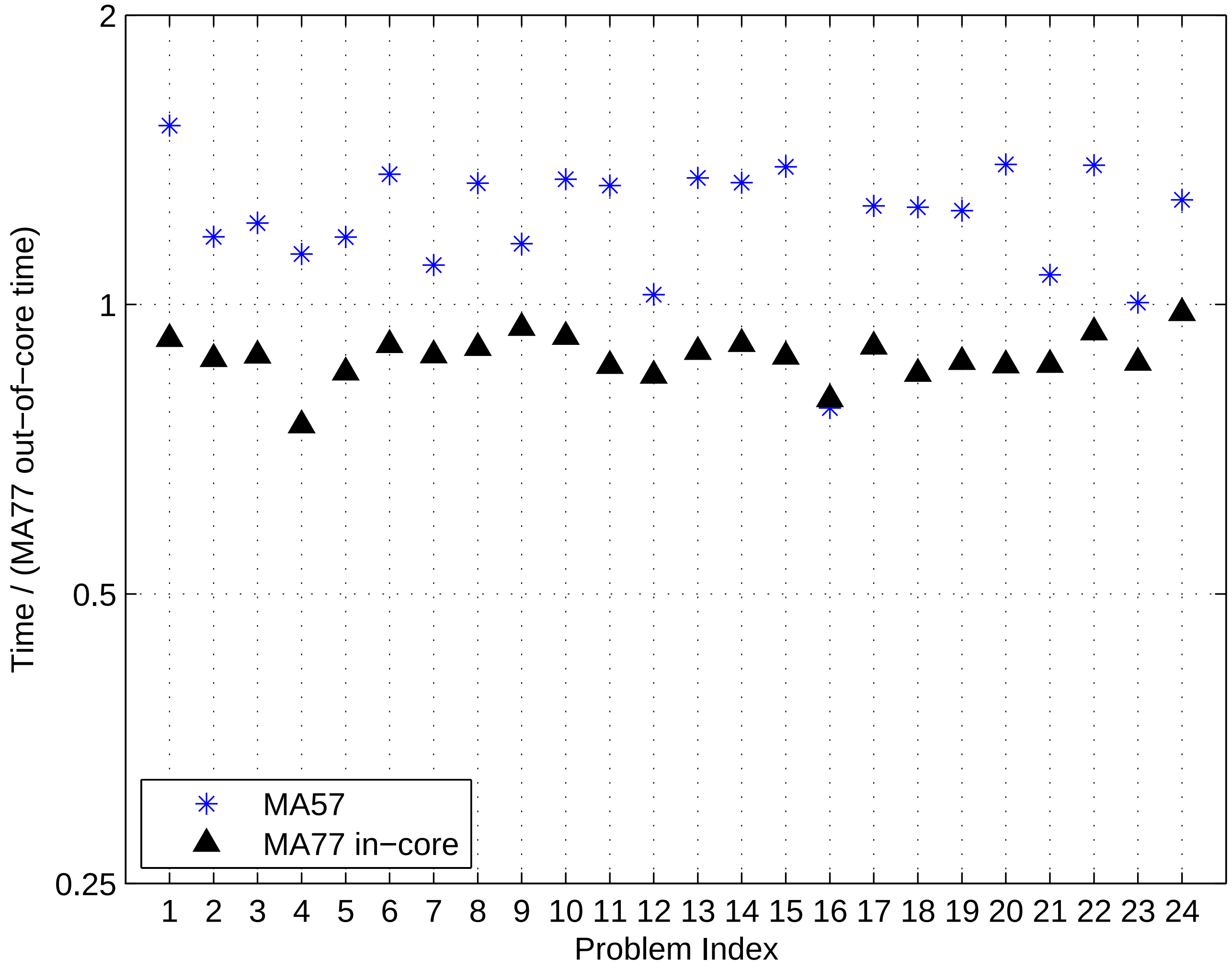
Use of long integers

HSL_MA77 make selective use of long integers (64-bit).

- 64-bit integers are needed for addresses in the virtual array
- If 32-bit integers are used to address the frontal matrix within HSL_MA77, its size is limited to about 2^{14} $(\sim 16,000)$
- We are now wanting to solve **LARGE** problems where this may be exceeded
- Do **not** want all integers used to be long integers (more storage and more data movement than necessary, and BLAS do not use long integers)
- Long integers are used selectively within HSL_MA77 (and within the dense linear algebra kernels)
- On 64-bit architecture, frontal size not limited to 2^{14} (user must specify if running on 64-bit machine but **no** other action needed)

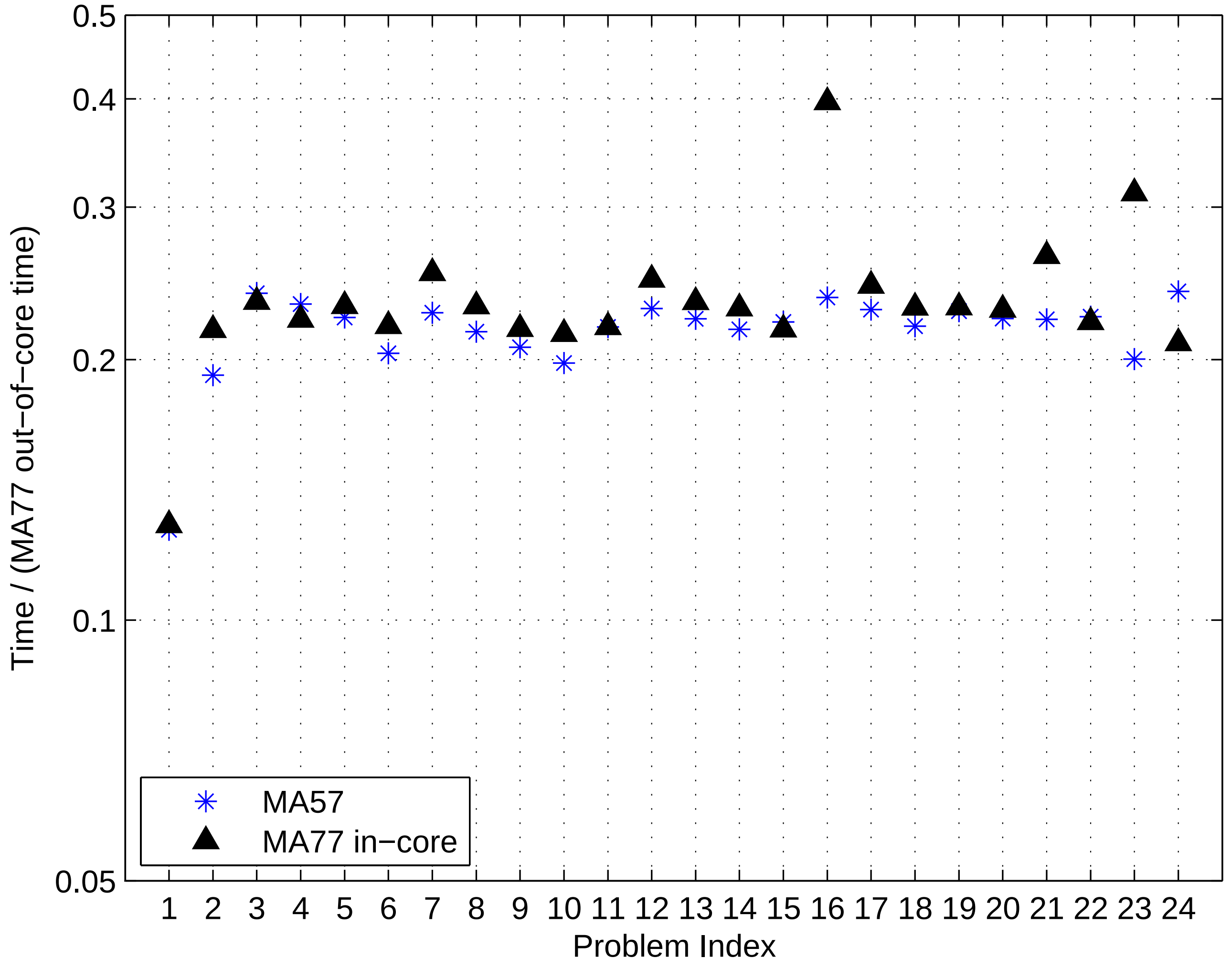
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Factor time compared with `MA57`



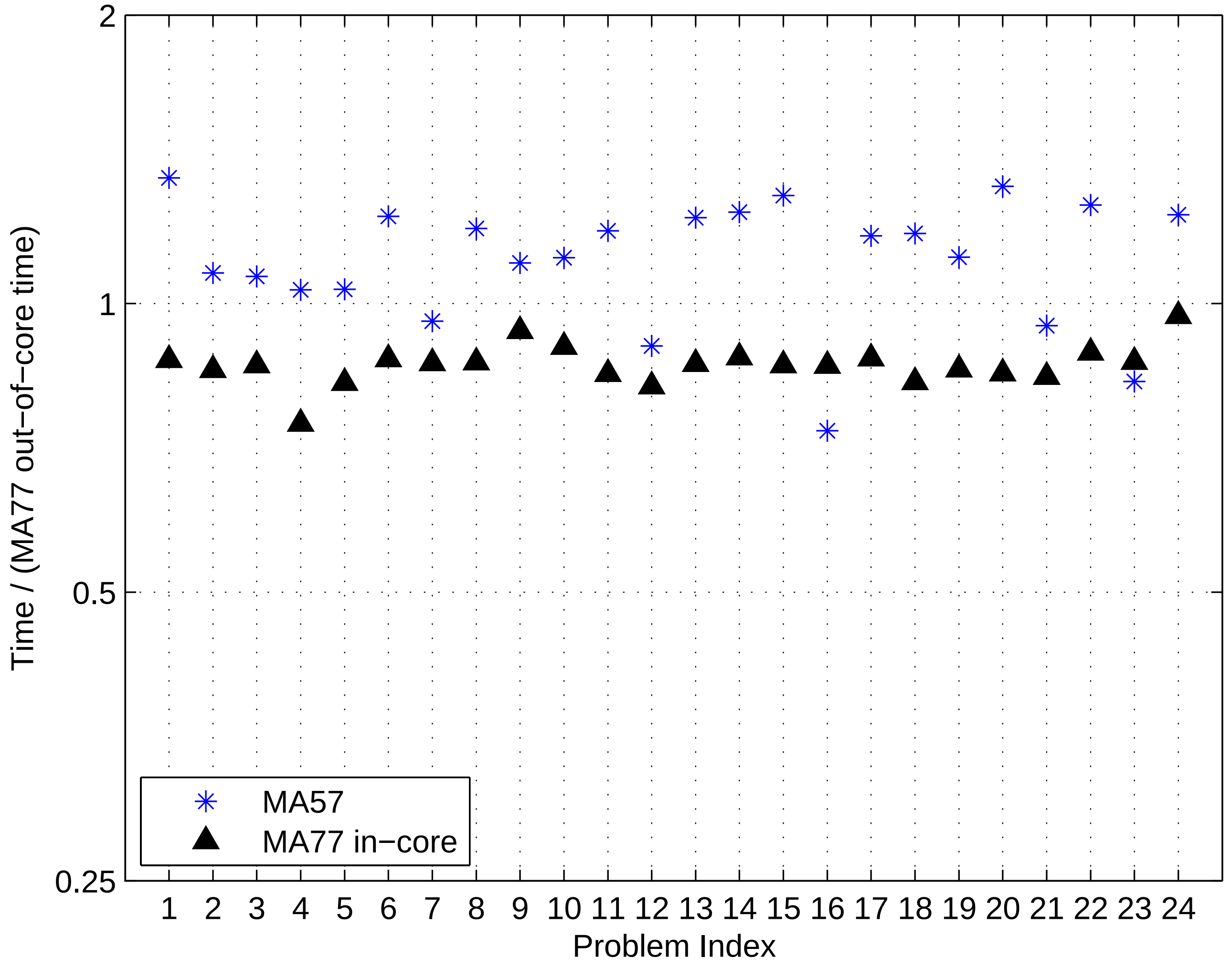
Science & Technology Facilities Council
Rutherford Appleton Laboratory

Solve time compared with `MA57`



Science & Technology Facilities Council
Rutherford Appleton Laboratory

Total time compared with `MA57`



Times (seconds) for larger problems

Phase	`inline_1` ($n = 503,712$)	`bones10` ($n = 914,898$)	`nd24k` ($n = 72,000$)	`bone010` ($n = 986,703$)
Input	4.87	6.25	2.86	8.00
Ordering	14.2	22.8	16.4	34.7
analyse	4.20	6.70	22.1	26.7
factorize	90.6	174	1284	1491
solve(1)	5.30	36.0	10.4	311
solve(8)	10.6	41.3	20.7	331
solve(64)	60.5	141	90.2	499

- `MA57` not able to solve these on our Dell Precision 670 (4 GB memory insufficient).

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Mixed precision approach

Advantages of single precision include:

- Reduces amount of data moved within direct solver (memory bandwidth bottleneck).
- Uses less storage (potentially solve LARGER problems).
- On a number of modern architectures, currently more highly optimised.

BUT potential loss of accuracy.

Solution: use double precision iterative method preconditioned by single precision factorization. This is currently a hot topic, particularly with regard to multicore machines.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Basic mixed precision algorithm

```
Input required accuracy $\epsilon$
Select initial factorization precision prec
do
    Factorize $\mathbf{A} = \mathbf{LDL^T}$ using precision prec
    Solve $\mathbf{Ax} = \mathbf{b}$ using double prec
    Compute scaled residual $res = \|\mathbf{b} - \mathbf{Ax}\|_\infty / (\|\mathbf{A}\|_\infty \|\mathbf{x}\|_\infty + \|\mathbf{b}\|_\infty)$
    if $res \le \epsilon$ then exit
    Perform iterative refinement using double prec
    if $res \le \epsilon$ then exit
    Perform FGMRES using double prec
    if $res \le \epsilon$ then exit
    if prec = single prec
        set prec = double prec and cycle
    else
    Set error flag and exit
end do
```

Science & Technology Facilities Council
Rutherford Appleton Laboratory

HSL mixed precision solver

New mixed precision solver will be `HSL_MA79` (Hogg and Scott).

Key features include:

- User inputs $\mathbf{A}$, required accuracy ϵ, and the right hand side(s), and `HSL_MA79` does the rest. **Simple to use and designed to be robust.**
- Code employs `MA57` and/or `HSL_MA77`.
- Control parameters allow the user to make choices (including solver, precision, number of refinement steps, ...)
- Multiple solves can follow single factorization.
- Option to factor and solve a system with the same pattern but different values using experience from previous system.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Example use of HSL_MA79

Application: 3D mine design and ground control, using 3D elasto-plastic FEM.

Problem data: $n = 3,633,677$, $nz = 145,626,418$

Test machine: Dell Precision T5400 with two 64-bit Quad-Core E5420 processors, 8 GB memory, two 146 GB SAS Hard Drives.

Analyse phase of HSL_MA77 predicts:
$nnz(L) = 1.47 * 10^{10}$, flops = $3.72 * 10^{14}$, max frontsize = $60,121$.

Projected memory usage for largest frontal matrix:

Number of entries	$1.8 * 10^9$
Single Precision:	6.7 GB
Double Precision:	13.5 GB

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Example use of HSL_MA79

Application: 3D mine design and ground control, using 3D elasto-plastic FEM.

Problem data: $n = 3,633,677$, $nz = 145,626,418$

Test machine: Dell Precision T5400 with two 64-bit Quad-Core E5420 processors, 8 GB memory, two 146 GB SAS Hard Drives.

Analyse phase of HSL_MA77 predicts:
$nnz(L) = 1.47 * 10^{10}$, flops = $3.72 * 10^{14}$, max frontsize = 60, 121.

Projected memory usage for largest frontal matrix:

Number of entries	$1.8 * 10^9$
Single Precision:	6.7 GB
Double Precision:	13.5 GB

HSL_MA79 results:

- approximately 250 minutes to factorize and solve system (using OpenMP version of dense kernel).
- **Scaled residual:** $\mathcal{O}(10^{-13})$.
- **Storage used:** approximately 70 GB.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Summary

- IR (PLAN A) does not always work with mixed precision

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting
- PLAN B is working

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting
- PLAN B is working
- Out-of-core solver `HSL_MA77` developed for large symmetric systems. Also version `HSL_MA78` for unsymmetric element problems.

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting
- PLAN B is working
- Out-of-core solver `HSL_MA77` developed for large symmetric systems. Also version `HSL_MA78` for unsymmetric element problems.
- Codes are performing well, solving larger problems than previously possible on desktop machines.

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting
- PLAN B is working
- Out-of-core solver `HSL_MA77` developed for large symmetric systems. Also version `HSL_MA78` for unsymmetric element problems.
- Codes are performing well, solving larger problems than previously possible on desktop machines.
- Out-of-core working adds an overhead but our memory management system `HSL_OF01` attempts to minimise this (single rhs solve expensive).

Summary

- IR (PLAN A) does not always work with mixed precision
- GMRES is also sensitive and not robust
- FGMRES is robust and less sensitive
- Gains from restarting
- PLAN B is working
- Out-of-core solver `HSL_MA77` developed for large symmetric systems. Also version `HSL_MA78` for unsymmetric element problems.
- Codes are performing well, solving larger problems than previously possible on desktop machines.
- Out-of-core working adds an overhead but our memory management system `HSL_OF01` attempts to minimise this (single rhs solve expensive).
- Mixed precision solver aimed at modern multicore architectures currently being developed ... so far, some encouraging results.

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Summary

Reminder: HSL 2007 packages are available for use **worldwide without charge** for individual academic research and teaching.

See www.cse.scitech.ac.uk/nag/hsl