# Accelerating the solution of slowly varying linear systems - Jemima M. Tabeart

In many applications users need to solve a large number of linear systems with similar matrix structures e.g. in engineering design. In this setting, users would like to test a large number of parameters to find either an optimal solution (e.g. minimum energy requirement) or to explore the range of possible behaviours for a realistic system. For each set of parameters, a linear system arising from a PDE is solved. Although individually some of these linear systems can be small, solving a large number of them via iterative methods may be computationally expensive and time consuming. Mathematically we want to solve a number of problems of the form

$$\begin{aligned} A_1 x_1 &= b_1 \\ A_2 x_2 &= b_2 \\ &\cdots\cdots \\ A_n x_n &= b_n \end{aligned}$$

where the $A_i$ matrices change slowly between subsequent problems. Other examples of problems of this nature include:

- In optimisation: the $A_i$ matrices come from linearisations of a non-linear problem e.g. a PDE operator at different timesteps, within different outer loops of an optimisation problem.
- In model order reduction where $A_i(\rho)$ might be realisations of a problem with a different parameter value, $\rho$ e.g. viscocity.

In this project we will consider methods to 'reuse' information when solving a new, but similar, problem. Depending on the interest of the student there are multiple research directions which are described below. A Bachelor or Master project could focus on one or multiple of these topics, and each project would include some numerical investigation (with code and examples provided by the supervisor) and theoretical work e.g. developing bounds on eigenvalues, convergence theory. The precise balance of these two aspects will depend on the project and student interest.

## Preconditioners

Preconditioners can be used to accelerate the solution of linear systems of the form $Ax = b$ by solving a transformed problem with better spectral properties. Designing good preconditioners is not easy - we want the preconditioner to be cheap to apply, while ensuring that using it reduces the number of iterations taken to reach convergence.
In this task we will look at how preconditioners for the original problem $A_0 x_0 = b_0$ can be adapted to the subsequent problems $A_i x_i = b_x$. We will investigate whether different preconditioners/problems are more amenable to this 'adaptive' approach, and metrics for determining ahead of time when we are better off just developing a new preconditioner for a given problem.

## The role of initial conditions

For many numerical studies the initial condition for an iterative method is taken to be random or the zero vector. However, if $A_2$ is close to $A_1$ and the right hand sides $b_1$ and $b_2$ are close, it might make sense to use the previously computed choice of $x_1$ as an initial condition for solving the second problem $A_2 x_2 = b_2$. In this project we will start by perturbing just the matrix $A_1$ and investigating to what extent an alternative choice of initial condition affects convergence of two related linear systems. We can then extend the numerical and theoretical investigations to the case of a perturbed right hand side, $b$. Part of this project will involve carefully designing numerical experiments, and appropriate quality metrics to properly compare reusing versus the current naive approach.

## Recycling

Krylov methods are some of the most popular approaches to solve linear systems for large dimensional problems. In these methods each iteration adds a new vector (search direction) to a subspace. Recycling approaches propose to reuse some of these search directions when solving a new problem, rather than starting with an empty search space.
In this project we will investigate whether recycling can lead to improvements in convergence compared to standard Krylov approaches, and for what types of problems. We will begin with using information from a previous basis to intialise a Krylov basis for the next problem, and investigate how to choose the best vectors, and best number of vectors to recycle. A second step could also consider using the Krylov vectors to develop preconditioners, providing a natural link with the first subproject.

*Requirements:* Experience in Python or Matlab. Strong familiarity with linear algebra is required, having studied numerical linear algebra would also be beneficial but not strictly required.

*Information:* Contact Jemima Tabeart at j.m.tabeart@tue.nl for further information.