Exercises part.2

27th March 2017

From now on, we focus on Conjugate Gradient only

Beyond machine precision

Choose at least one of the two features and implement them, discussing your findings in the report

1) Explicit residue

The residue is computed implicitly in the program through the recursive equation

$$r_k^{impl} = r_{k-1}^{impl} - \alpha A p_{k-1} \, .$$

As an effect of the numerical rounding, this actually differs from the explicit residue, computed from its definition:

$$r_k^{expl} = b - A x_k \, .$$

1. Compute the explicit residue and compare it with the implicit one (relative to $|b|$), at various step of the solution.
2. Ask the solver to find a solution requesting a small target relative precision $\hat{r}_{targ} \ll 10^{-16}$. Is the solver able to reach this target? Produce a graph which compares the implicit and explicit residues as a function of the number of iterations, showing where the algorithm breaks.

2) Initial guess

Splitting the solution in two parts, $x = x_{guess} + \Delta x$ one can rewrite the system as:

$$A \Delta x = b - A x_{guess} = c \, ,$$

and solve the system for Δx on the source c. At the end the solution Δx must be summed to x_0 to obtain x

1. Implement the possibility of taking an external guess as parameter of the solver.
   This is the basic ingredient for the development of a *restarted* solver able to achieve $\hat{r}_{targ} < \hat{r}_{machine}$.
2. Try solving the system:
   $$(M + \delta \mathrm{Id}) \, y = b$$
   using as a first guess y_0 the solution of the system:
   $$M x = b$$
   for various values of δ, for a certain M of choice.
   (a) Does the number of iterations needed to solve the shifted system decrease using a guess?
   (b) How is it related to δ? Produce a plot of the number of the relative number of iterations saved:
   $$gain = \frac{n_{iter}^{without\,guess} - n_{iter}^{with\,guess}}{n_{iter}^{without\,guess}} = 1 - \frac{n_{iter}^{with\,guess}}{n_{iter}^{without\,guess}} \, ,$$
   as a function of σ, for a fixed relative residue 10^{-8}.

Laplace Equation

The discrete Laplace equation is

$$M_{ij} f_j = b_i \,,$$

with:

$$M_{ij} = (\sigma + D)\, \delta_{ij} - \frac{1}{2} \sum_{\mu} \left(\delta_{i,j+\hat{\mu}} + \delta_{i,j-\hat{\mu}}\right) \,.$$

Here we represent M for a $D = 1$ space of size $L = 6$ with periodic boundary conditions ($d = \sigma + D$, $s = -1/2$):

$$\begin{pmatrix} d & s & 0 & 0 & 0 & s \\ s & d & s & 0 & 0 & 0 \\ 0 & s & d & s & 0 & 0 \\ 0 & 0 & s & d & s & 0 \\ 0 & 0 & 0 & s & d & s \\ s & 0 & 0 & 0 & s & d \end{pmatrix}$$

1) Implement the solution of the problem using Conjugate Gradient

1. Fill explicitly the matrix M (with periodic boundary conditions) for an arbitrary space of size L and σ in the case of a $1D$ space, and solve the system for a randomly generated b
2. Compare it with the 1D exact solution obtained with the provided routine

2) Implement efficiently Mp

We can avoid creating explicitly the unnecessary matrix M. In facts the computation of $t = Mp$ needed in the conjugate gradient solver can be done via

$$t_i = M_{ij} p_j = (\sigma + D)\, p_i - \frac{1}{2} \left(p_{[i-1]_N} + p_{[i+1]_N}\right) , \qquad [i]_N \equiv (i + N)\%N$$

1. Implement in 1D the implicit computation of Mp
2. Compare again the solution with the exact one
3. Compare time required to achieve a fixed precision using an explicit or implicit application of M

3) Check condition number [optional]

The eigenvalues of M can be computed analytically:

$$\lambda\,(k) = \sigma + D - \sum_{\mu} \cos k_{\mu} \qquad \kappa_{\mu} = \frac{2\pi}{L} i_{\mu}, \quad i_{\mu} \in [0,\, 1, \ldots L - 1]$$

such that in 1D the minimal eigenvalues is given by $\lambda_{min} = \sigma$ and the maximum by $\lambda_{max} = \sigma + 2$.

1. Check that the number of iterations needed to obtain a solution on a fixed size depends from σ as expected from the fact that $\kappa = \frac{\lambda_{max}}{\lambda_{min}} = \frac{2}{\sigma} + 1$
2. Check that when $-2 \leq \sigma \leq 0$ the algorithm does not converge (in facts no solution exists for $\sigma = 0$)