# MAE 223, Spring 2009

**Homework 2**

Due Friday, May 15, before 3 PM.

**1.** The viscous Burgers equation,

$$u_t + uu_x - \alpha u_{xx} = 0 \tag{1}$$

with the initial data corresponding to a square wave, as follows:

$$\begin{aligned} u_0(x) &= 1 \quad , \quad 0 \leq x \leq 1 \\ u_0(x) &= 2 \quad , \quad 1 \leq x \leq 2 \\ u_0(x) &= 1 \quad , \quad x > 2 \end{aligned}$$

and $\alpha = 0.1$ is to be solved in the domain $0 \leq x \leq 10$ and $0 \leq t \leq 20$.

Assume inflow and outflow boundary conditions. Use a numerical method with central, 2nd-order spatial discretization and the low-storage RK scheme given below. Use $\Delta x = 0.1$. Explore the behavior of the solution when $\alpha$ is progressively reduced.

*Try to use Fortran as the programming language.*

### A low-storage RK3 scheme

Consider the o.d.e

$$\frac{dy}{dt} = f(y, t)\,. \tag{2}$$

A RK3 (Runge-Kutta, third order) algorithm involves three stages and approximates the local Taylor expansion of $y(t)$ with a truncation error of $O(\Delta t^4)$. The global error is $O(\Delta t^3)$.

A p-stage RK scheme is given by

$$y^{n+1} = y^n + \Delta t \sum_{j=1}^{p} w_j f_j \tag{3}$$

where $w_j$ are the the constant weighting coefficients and the successive function evaluations are given by

$$\begin{aligned} f_1 &= f(y^n, t^n) \\ f_i &= f(y^n + \beta_i \Delta t f_{i-1}, t^n + \gamma_i \Delta t) \end{aligned}$$

with $\beta_i$ and $\gamma_i$ given constants.

Now let the variable $\mathbf{y}$ be a m-dimensional vector so that Eq. (2) represents a *system* of o.d.e's. The integration scheme, Eq. (3), requires *two* additional m-dimensional vectors: $\mathbf{f}$ to store the function evaluations, $f_i$, and $\mathbf{q}$ to store the intermediate values of $y$ required for the function evaluation. Note that the new values at the end of stage $j$ replace the previous values at stage $j-1$ in $\mathbf{f}$ and $\mathbf{q}$ so that the storage requirement is independent of the number of stages.

A *low-storage* RK scheme is written as follows:

$$
\begin{aligned}
h_j &= a_j h_{j-1} + \Delta t f(y_{j-1}, \alpha_j \Delta t) \\
y_j &= y_{j-1} + b_j h_j
\end{aligned}
\tag{4}
$$

Here $j = 1, ...p$ with $p$ the number of stages, $j = 0$ corresponds to the old time level $t_n$, while $j = p$ corresponds to the required solution at the new time level $t_{n+1}$. The above scheme requires *one* additional array **h** to store $h_j$ instead of the two arrays required in the scheme given by Eq.(3). The scheme to be used in the HW is described by,

| | a | $\alpha$ | b |
|---|---|---|---|
| j =1 | 0 | 0 | $\frac{1}{3}$ |
| j = 2 | $-\frac{5}{9}$ | $\frac{1}{3}$ | $\frac{15}{16}$ |
| j = 3 | $-\frac{153}{128}$ | $\frac{3}{4}$ | $\frac{8}{15}$ |