16th European Symposium on Computer Aided Process Engineering
and 9th International Symposium on Process Systems Engineering
W. Marquardt, C. Pantelides (Editors)

Simulation of the population balance for droplet breakage in a liquid-liquid stirred tank reactor using H-matrix methods

Jürgen Koch,[a] Wolfgang Hackbusch,[b] Kai Sundmacher[a,c]

[a]*Max-Planck-Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, Magdeburg D-39106, Germany*
[b]*Max-Planck-Institute for Mathematics in the Sciences, Inselstr. 22, Leipzig D-04103, Germany*
[c]*Otto-von-Guericke-University Magdeburg, Process Systems Engingeering, Universtiätsplatz 2, Magdeburg D-39106, Germany*

Abstract

In population balance equations particle breakage is often described by a Volterra integral operator. Naive discretisation of this operator leads to quadratic complexity. In this paper an efficient numerical treatment for Galerkin discretisation of the integral operator is suggested, based on the idea of H-matrices, which leads to linear complexity.

Keywords: H-matrices, fast methods.

1. Introduction

Emulsions are mixtures of two immiscible liquids (e.g. water and oil) and a surfactant, needed to stabilise the liquid-liquid interface. One of the liquids, i.e. the disperse phase usually establishes a population of droplets and so the emulsion can be considered as a disperse system. Such systems are often applied directly in consumer designed products or they appear as intermediates in chemical processes.

Furthermore, well characterised emulsions can be used as a structured reaction medium to manufacture solid particles in the sub micrometer range [6]. This approach allows to control the particle properties by droplet size and other emulsion properties. For the particle synthesis two similar emulsions, each containing dissolved educts in water droplets, are mixed in a stirred tank. By droplet coalescence and breakage events the reaction is initialised leading subsequently to the formation of solid particles in the droplets.

The correct understanding of the droplet population behaviour is thus of practical importance. Commonly for the description of the system a population balance equation is used, where the coalescence and breakage events are characterised by Volterra integral operators of the first kind. After a naive discretisation these operators have usually a complexity of $O(n^2)$ for storage and matrix-vector-multiplication if n denotes the problem size. In this work, for the operator being responsible for the breakage of droplets, introduced in [1] from Coulaloglou and Tavalarides, a Galerkin discretisation will be presented using the ideas of the H-matrices, compare Hackbusch [4]. After some modifications this discretisation leads to a complexity of $O(n)$.

2. General model

In literature population balance equations can be found in arbitrary detailed versions. The balance equation according to [1] has a non-local character and can be written in the general form

$$\frac{\partial F(x,t)}{\partial t} = F_{\text{in}}(x,t) - F_{\text{out}}(x,t) + F_{\text{co}}^{+}(x,t) - F_{\text{co}}^{-}(x,t) + F_{\text{br}}^{+}(x,t) - F_{\text{br}}^{-}(x,t). \quad (1)$$

The individuals of the droplet population are described by the number density function F. This function specifies the number of droplets of the size (volume) x at the time t in the tank reactor. The range of F in the first variable is given by $[x_{\min}, x_{\max}] \subseteq [0, x_{\max}]$, where $x_{\min}$ stands for the smallest and $x_{\max}$ for the biggest possible droplet size. The terms F_{in} and F_{out} denote the inflow and the outflow of liquid and droplets, respectively. The operator F_{co}^{+} and F_{co}^{-} name the source and sink terms of the coalescence and the operator F_{br}^{+} and F_{br}^{-} the source and sink terms of breakage. The interest is directed towards the operators representing the breakage phenomenon. In detail they read as

$$F_{\text{br}}^{+}(x,t) = K_{\text{lin}}[F(.,t)](x) = \int_{x}^{x_{\max}} \nu(x)\varphi(x,y)\beta_{\text{br}}(y)F(y,t)dy \quad \text{and} \quad (2)$$

$$F_{\text{br}}^{-}(x,t) = K_{\text{diag}}[F(.,t)](x) = \beta_{\text{br}}(x)F(x,t). \quad (3)$$

Here β_{br} is the breakage frequency, $\nu(x)$ represents the number of daughter droplets formed from a droplet with size x, and $\varphi(x,y)$ is the distribution of daughter droplets generated from the breakage of a droplet of size y. The discretisation of the diagonal operator F_{br}^{-} is trivial henceforth we focus on the integral operator F_{br}^{+}.

3. Kernel function for droplet breakage

The equation (2) can be rewritten with the help of the kernel function κ

$$K_{lin}[F](x) = \int_{x}^{x_{\max}} \nu(x)\varphi(x,y)\beta_{br}(y)F(y)dy = \int_{x}^{x_{\max}} \kappa(x,y)F(y)dy \quad (4)$$

where the time dependency is omitted. In the explicit construction of the kernel function it is assumed, that a breaking droplet forms always two daughter droplets, i.e. $\nu(x) \equiv 2$. The breakage frequency β_{br} represents as

$$\beta_{br}(x) = c_1 x^{-\frac{2}{9}} \exp\left[-c_2 x^{\frac{5}{9}}\right] \quad (5)$$

where the constants c_1 and c_2 depend on the stirrer diameter, its revolution speed, the density of the liquids and their surface tension, see [1], and for the a orders of magnitude holds $c_1 \sim 1$ and $c_2 \sim 10^{-6}$. Further on the daughter droplet distribution φ is based on a Gaussian distribution and can be written as

$$\varphi(x,y) = \frac{c_3}{y}\exp\left[-c_4 \frac{(2x-y)^2}{y^2}\right] \quad (6)$$

where the constants c_3 and c_4 are chosen in the way that 99.6% of the broken droplets are located in the interval $[0, y]$, compare again [1]. The order of magnitude is $c_3 \sim c_4 \sim 1$.

4. Numerical approach

For the discretisation of the integral operator (4) a Galerkin scheme is used. The Galerkin system matrix $\boldsymbol{K}$ has for $i, j = 1,...,n$ the entries

$$K_{ij} = \int_{\operatorname{supp} b_i} b_i(x) \int_{\operatorname{supp} b_j \cap [x, x_{\max}]} b_j(y) \kappa(x, y) dy dx, \tag{7}$$

where b_i and b_j denote the corresponding basis functions and n the degrees of freedom. Assuming piecewise constant basis functions with $\operatorname{supp}(b_i)=[x_i, x_{i+1}]$ for $i=1,...,n$. The entries of $\boldsymbol{K}$ can be written

$$K_{ij} = \begin{cases} 0 & \text{for } x_i \le x_{j+1} \\ \int_{x_i}^{x_{i+1}} \int_{x_j}^{x_{j+1}} \kappa(x, y) dy dx & \text{for } x_j > x_{i+1}, \quad i,j=1,...,n. \\ \int_{x_i}^{x_{i+1}} \int_{x}^{x_{j+1}} \kappa(x, y) dy dx & \text{for } x_i = x_j \end{cases} \tag{8}$$

Though the resulting matrix is an upper triangular with complexity $O(n^2)$.
First we assume the integral operator (4) has a fixed lower boundary $x_{\min}$ (Fredholm operator) and the kernel κ is separable, i.e. it can be written in the form

$$\kappa(x, y) = \sum_{l=1}^{k} \Phi_l(x) \cdot \Psi_l(y) \tag{9}$$

where k is called the *separation rank*. Then the entries of the corresponding system matrix $\boldsymbol{K}$ would have the following form

$$K_{ij} = \int_{x_i}^{x_{i+1}} \int_{x_j}^{x_{j+1}} \kappa(x, y) dy dx = \sum_{l=1}^{k} \int_{x_i}^{x_{i+1}} \Phi_l(x) dx \cdot \int_{x_j}^{x_{j+1}} \Psi_l(y) dy.$$

In this case the matrix $\boldsymbol{K}^{n \times n}$ can be written as a low rank matrix with help of two matrices $\boldsymbol{S}^{n \times k}$ and $\boldsymbol{T}^{k \times n}$ in the following way

$$S_{il} = \int_{x_i}^{x_{i+1}} \Phi_l(x) dx \quad \text{and} \quad T_{lj} = \int_{x_j}^{x_{j+1}} \Psi_l(y) dy.$$

In the format $\boldsymbol{K}=\boldsymbol{S}\boldsymbol{T}$ only $2kn$ entries have to be stored. A matrix-vector-multiplication of $\boldsymbol{K}$ with a vector $\boldsymbol{x}$ can be done in two steps. First $\boldsymbol{x}$ has to be multiplied with $\boldsymbol{T}$, to get the vector $\boldsymbol{Tx}$. That requires $\sim 2kn$ operations. Next vector $\boldsymbol{Tx}$ is multiplied with $\boldsymbol{S}$. That requires also $\sim 2kn$ operations, see Figure 1.

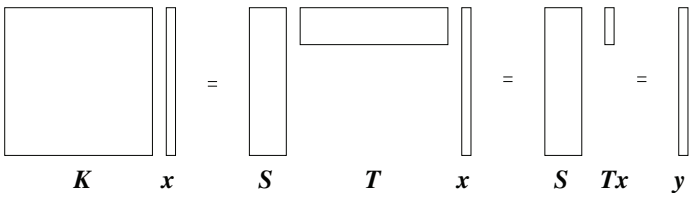

Figure 1. Multiplication scheme of a low rank matrix.

For a Volterra operator and separable kernel, the matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ are created as in the Fredholm case. For multiplication the following algorithm is used, where the entries of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are denoted by x_j and y_i.

Algorithm 1:

```
for (l = 1 to k) begin
   sum = 0.0; j = n;
   for (i=n-1 down to 1) begin
      while (j>i) do begin
         sum = sum + T_lj x_j;
      end;
      y_i = y_i + S_il * sum;
   end;
end
```

Note, that the diagonal elements of $\boldsymbol{K}$ have to be added separately. We call this kind of low rank matrix *Volterra low rank matrix*. The overall complexity is also $O(kn)$.

5. Construction of special H-matrices

The kernel function κ from (9) is not separable, but it could be approximated by a separable kernel expansion, e.g. by Lagrange interpolation in the first variable

$$\kappa(x,y) = \sum_{l=0}^{k} \kappa(x_l, y) L_l(x)$$

where the x_l denote the interpolation knots and k denotes the polynomial degree.
Usually this will not deliver satisfying results over the whole integration interval. In the concept of H-matrices the matrix $\boldsymbol{K}$ is split up blockwise with the help of a so-called *block cluster tree* for a class of kernel functions, see [4,3]. Each matrix block is connected to an integration area $[a,b]\times[c,d]$, where $[a,b],[c,d] \subseteq [x_{\min}, x_{\max}]$, and on each area a different kernel approximation is used. This procedure leads to low rank matrix blocks, with the advantages mentioned above. However not on every block this procedure is reasonable, thus some (small) blocks are represented by a full matrix, for details see again [4,3]. In [5] a modification of H-matrices is introduced taking into account the properties of the kernel (9) and of the Volterra character of the integral operator (4). The resulting modified H-matrix is sketched out in Fig. 2. On each matrix block, with corresponding integration area $[a,b]\times[c,d]$, the interpolation error (in x or y) can be estimated by

$$\left\| \kappa - \Pi_k[\kappa] \right\|_{\infty,[a,b]} \le (1 - \Lambda_k) \left\| \kappa \right\|_{\infty,C} \frac{1}{2^{k-1}} \tag{10}$$

independent of the interval $[c,d]$. Here Π_k denotes the Lagrange interpolation operator of degree k and Λ_k the so-called *Lebesgue constant*, see [2]. The definition of the number $\|\kappa\|_{\infty,C}$ can be found in [5]. Important in (10) is the factor $1/2^{k-1}$, it allows to decrease the error with increasing degree k. The relative error made by using the H-matrix instead of the full matrix for the system matrix $\boldsymbol{K}$ (refer to system (8)) is given in Table 1 in the spectral norm. The error decreases by the factor ~0.25 every time k is increased by 1.
The complexity of the H-matrix sketched out in Fig. 2 is of $O(kn)$, i.e. $O(n)$ for fixed k, instead of formerly $O(n^2)$. The amount of storage for k=12 is pointed out in Table 2.

Due to the quadratic behaviour of the full system matrix e.g. for n=8192 it needs ~140 times more memory than the H-matrix with linear behaviour.

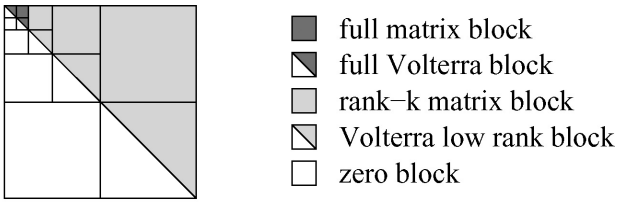

Figure 2. Block structure of a modified H-matrix.

In the Tables 3 and 4 the times for matrix creation and matrix-vector-multiplication are compared. Here the H-matrix method is ~100 times faster for n = 8192.

Table 1. Relative errors for increasing k in spectral norm (n=1024).

k	7	8	9	10	11	12
rel. error	$4.97\cdot10^{-4}$	$1.10\cdot10^{-4}$	$2.41\cdot10^{-5}$	$5.12\cdot10^{-6}$	$1.19\cdot10^{-6}$	$2.75\cdot10^{-7}$

Table 2. Used memory in [MB] for k=12.

n	521	1024	2048	4096	8192
Full matrix	2.00	4.00	32.00	128.00	512.00
H-Matrix	0.28	0.51	0.95	1.85	3.64

Table 3. Time for matrix creation in [sec] for k=12.

n	521	1024	2048	4096	8192
Full matrix	$8.85\cdot10^{-1}$	$3.57\cdot10^{+0}$	$1.41\cdot10^{+1}$	$6.12\cdot10^{+1}$	$2.52\cdot10^{+2}$
H-Matrix	$6.67\cdot10^{-2}$	$1.32\cdot10^{-1}$	$2.64\cdot10^{-1}$	$5.41\cdot10^{-1}$	$1.06\cdot10^{+0}$

Table 4. Time for matrix-vector-multiplication in [sec] for k=12.

n	521	1024	2048	4096	8192
Full matrix	$4.81\cdot10^{-3}$	$1.91\cdot10^{-2}$	$8.02\cdot10^{-2}$	$3.51\cdot10^{-1}$	$1.43\cdot10^{+0}$
H-Matrix	$3.49\cdot10^{-4}$	$1.53\cdot10^{-3}$	$3.20\cdot10^{-3}$	$6.02\cdot10^{-3}$	$1.02\cdot10^{-2}$

As a first illustrated example the numerical solution of a reduced population balance equation

$$\frac{\partial F(x,t)}{\partial t} = F_{\mathrm{br}}^{+}(x,t) - F_{\mathrm{br}}^{-}(x,t), \tag{11}$$

was calculated, compare (2) and (3), using the full system matrix as well as a H-matrix with k=12. For the constants of the kernel (4) the following values were used $c_1 = 2.4$, $c_2 = 4.5$, $c_3 = 0.35$ and $c_4 = 4.7\cdot10^{-6}$. A normalised initial population $F(x,0)=F_0(x)$ was chosen and the development of the population for different times was evaluated with a simple Euler method, for n=1024.

Results can be found in the diagrams in Figure 3 and Figure 4 for different time steps t_1 and t_2. Here the graphs produced with the help of the full matrix and the H-matrix are nearly identical and therefore almost indistinguishable.

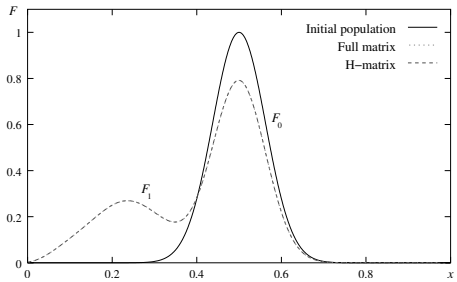

Figure 3. Population: $F(x,t_1) = F_1(x)$.

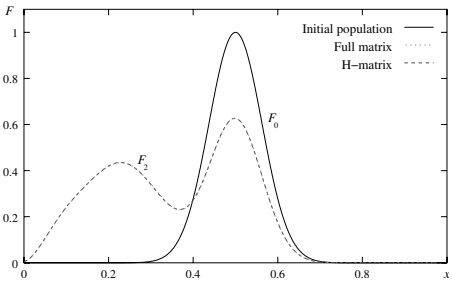

Figure 4. Population: $F(x,t_2) = F_2(x)$.

References

1. C. A. Coulaloglou and L. L. Tavalarides, 1977, Description of interaction processes in agitated liquid-liquid dispersions, Chem. Engng. Sci., 32, 1289-1297
2. P. J. Davis, 1963, Interpolation and Approximation, Blaisdell Publishing Co. Ginn and Co.
3. L. Grasedyck and W. Hackbusch, 2003, Construction and Arithmetics of H-matrices, Computing, 62, 89-108
4. W. Hackbusch, 1999, A sparse matrix arithmetic based on H-matrices, Part I: Introduction to H-matrices, Computing, 70, 295-334
5. J. Koch, 2005, Effiziente Behandlung von Integraloperatoren bei populationsdynamischen Modellen, Otto-von-Guericke-Universität Magdeburg
6. C. Y. Tai, M. H. Lee and Y. C. Wu, 2001, Control of zirconia particle size by using two-emulsion precipitation technique, Chem. Engng. Sci., 56, 2389-2398