ECHTZEITOPTIMIERUNG GROSSER SYSTEME

T. Binder, L. Blank, W. Dahmen, W. Marquardt

Iterative Multiscale Methods for Process Monitoring

Preprint 00-16

SCHWERPUNKTPROGRAMM DER DEUTSCHEN FORSCHUNGSGEMEINSCHAFT

Iterative Multiscale Methods for Process Monitoring*

T. Binder[1], L. Blank[2], W. Dahmen[2], W. Marquardt[1]

[1] Lehrstuhl für Prozesstechnik,
[2] Institut für Geometrie und Praktische Mathematik,
RWTH Aachen, Germany.

Abstract

Process monitoring via state estimation is accomplished by the solution of an appropriately posed optimization problem on a receding horizon. The need of a fast solution of the optimization problem is additionally restricted by the a priorily unknown available computation time. Therefore, we propose a multiscale approach with the following features. It provides a coarse approximation after a very short time and upgrades iteratively approximations from coarser levels through further refinement steps. At each refinement step the error is to be reduced by a fixed fraction according to the discretization error while the corresponding computational cost is to remain proportional to the current number of degrees of freedom. Our multiscale discretization is based on suitable wavelet bases. It will be presented along with numerical results as well as with open questions.
Keywords: dynamic real time optimization, adaptive wavelet discretization, nested iteration, inverse problem

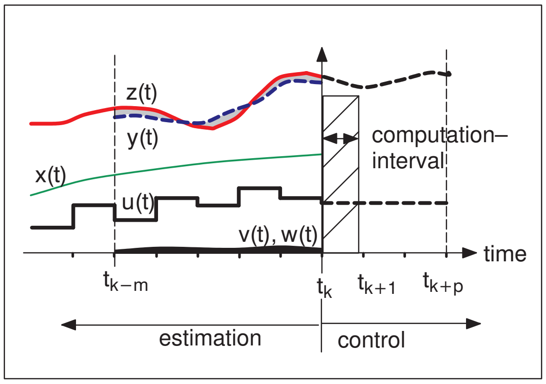
1 Problem formulation

Monitoring chemical processes requires real time estimates of the states, which are typically not accessible by measurement, and of the modeled output functions from a few given, possibly noisy data. In case of nonlinear dynamic modeling state estimation can be formulated as a *nonlinear constrained optimization problem*, where the difference between modeled output and measured data on a receding, moving horizon is to be minimized [14, 16, 17]. The dynamic behaviour of the plant is usually modeled by a system of differential–algebraic equations (DAEs) which together with bounds on selected variables form the constraints. Figure 1 illustrates the situation for a simple one–dimensional setting. The present time is denoted by t_k, and we assume that the noisy discrete data, which are known for the receding time, are processed into denoised signal functions $\boldsymbol{z}$. The control functions are denoted by $\boldsymbol{u}$ and the parameters by $\boldsymbol{p}$. For state estimation both, $\boldsymbol{u}$ and $\boldsymbol{p}$ are given. The goal is to estimate on the receding fixed time interval $[t_{k-m}, t_k]$ the state functions $\boldsymbol{x}$ and the output functions $\boldsymbol{y}$, which correspond to the signals. The solution has to be determined in a computation interval, which is given by the dynamic of the process. Then, the time

*This work has been supported by the “Deutsche Forschungsgemeinschaft” under grant no. MA1188/6

horizon will be moved to $[t_{k-m+1}, t_{k+1}]$ and the estimation process will start again with new additional data. Moreover, additive model correction terms are introduced into the DAEs as functions $\boldsymbol{v}$ and $\boldsymbol{w}$ which have to be estimated as well. The introduction of model correction terms actually may enhance ill–posedness. Therefore, at any rate, regularization should include the correction terms $\boldsymbol{v}$ and $\boldsymbol{w}$ as quadratic terms in the cost functional [2].

FIGURE 1. Receding horizon



For given $\boldsymbol{z} \in (L_2)^{n_y}$, $\boldsymbol{p}$ and $\boldsymbol{u} \in (L_2)^{n_u}$ and unknown functions $\boldsymbol{x} \in (H^1)^{n_x}$, $\boldsymbol{y} \in (L_2)^{n_y}$, $\boldsymbol{v} \in (L_2)^{n_v}$ and $\boldsymbol{w} \in (L_2)^{n_w}$ with $n_w \leq n_x$, $n_v \leq n_y$, the resulting optimization problem has the form

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{w}} \int_{t_{k-m}}^{t_k} \left\{ (\boldsymbol{y}-\boldsymbol{z})^T \boldsymbol{Q} (\boldsymbol{y}-\boldsymbol{z}) + \boldsymbol{v}^T \boldsymbol{R_v} \boldsymbol{v} + \boldsymbol{w}^T \boldsymbol{R_w} \boldsymbol{w} \right\} \, d\tau \tag{1}$$

with constraints

$$\begin{aligned} \dot{\boldsymbol{x}} - \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}) - \boldsymbol{W}\boldsymbol{w} &= \boldsymbol{0} && (2)\\ \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}) - \boldsymbol{V}\boldsymbol{v} &= \boldsymbol{0} && (3)\\ \boldsymbol{l_x}(t) \leq \boldsymbol{x}(t) &\leq \boldsymbol{s_x}(t) && (4)\\ \boldsymbol{l_y}(t) \leq \boldsymbol{y}(t) &\leq \boldsymbol{s_y}(t)\ . && (5) \end{aligned}$$

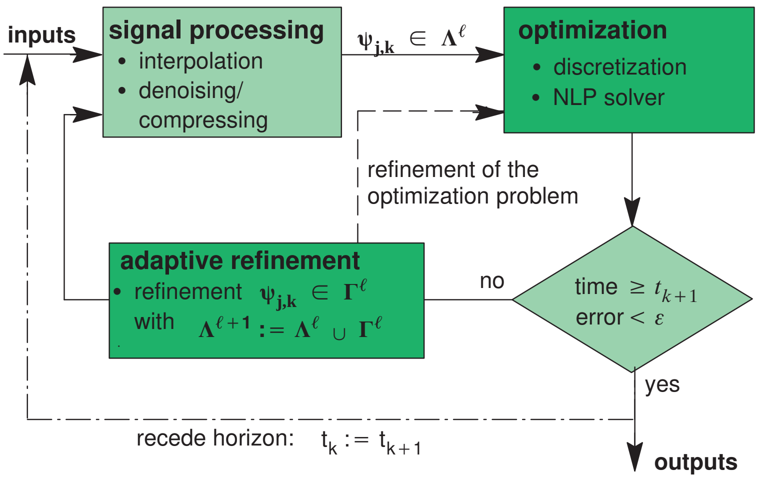
The dimension of the problem is dominated by the number of state functions n_x. We want to emphasize that state estimation is an ill–posed inverse problem. In the above regularized problem the initial data are still not given but have to be estimated. Additionally, the observability measure of the model, the stiffness of the DE's, the incorporated model error functions and the sensible choice of their regularization terms $\boldsymbol{R_v}$ and $\boldsymbol{R_w}$ have a major influence on the conditioning of the formulated optimization problem. In Section 8 these influences are shortly sketched. For more details we refer the reader to [3, 5].

2 Real time requirement and conceptional framework

In practice, typically several software processes are running simultaneously on one work station. As a consequence the available computation time is a priorily unknown and the process dynamic only gives an upper limit. This suggests a new view on **real time requirement**: we have to *provide an approximate solution at any time* in $[t_k, t_{k+1}]$. Typically, available solvers are based on a conservative estimation of computation costs and a fixed complexity and are therefore not designed to fulfill this restriction. The conceptional framework we want to present now hinges on this real–time requirement.

Firstly, a *unified framework* for signal analysis and the optimization problem (1)–(5) facilitates a very fast data exchange between these two blocks. Secondly, we start with a very coarse approximation and then *successively refine* it by an *adaptive strategy*, where the refinement costs should remain proportional to the current number of freedom, and the acquired informations should be exploited as much as possible for the refinement step. Figure 2 sketches our approach. The data input is processed by denoising, compressing and interpolation schemes to produce signal functions $\boldsymbol{z}$. This step provides also an initial mesh, respectively a first set of basis functions Λ^0, for discretizing the optimization problem. An NLP–solver determines a coarse grid solution of the optimization problem. Then, if time permits, this solution is updated in an adaptive way. Therefore, the coarse grid solution of the optimization problem as well as the denoised signals function as indicator for the new, refined mesh (basis index set $\Lambda^{\ell+1}$). The optimization problem is resolved on this refined mesh by exploiting additional signal information as well as any available information from previous stages. As long as time permits this refinement procedure is repeated, until eventually the time horizon is moved according to the dynamics of the process. The subsequent calculations exploit again the information of the previous time horizon.

FIGURE 2. Concept



We propose to realize the above concept with the aid of wavelet methods for following reasons. Firstly, in signal processing wavelets are a very efficient tool for denoising and compression [8, 9]. Secondly, multiscale representation avoids the change of basis in the updating procedure. Moreover, wavelet properties can be exploited for example for preconditioning and adaptive refinement. In the next section, we shortly introduce biorthogonal wavelets and some of their basic properties and indicate their significance in the present context. For a more detailed treatment of wavelets we refer the reader to [9, 10, 11, 13].

3 Wavelets

Apart from boundary effects wavelet functions $\psi_{j,k}$ are obtained by dilation and translation of a suitable *mother wavelet* ψ, i.e. $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$, where the normalizing factor $2^{j/2}$ keeps the L_2–norm of the $\psi_{j,k}$ independent of j, k. Thus when ψ is chosen to have bounded support, as will always be the case in our applications, one has $\operatorname{diam} \operatorname{supp} \psi_{j,k} \sim 2^{-j}$, $j \to \infty$. The key feature is that for suitable ψ the collection $\boldsymbol{\Psi} = \{\psi_{j,k}\}$ forms a *Riesz* *basis* of L_2, i.e., every $f \in L_2$ has a unique expansion

$$f \;=\; \mathbf{\Psi}^T \boldsymbol{d}_f = \sum_{j,k} d_{j,k}\psi_{j,k} \tag{6}$$

and

$$c_1 \parallel \boldsymbol{d}_f \parallel_{\ell_2} \leq \parallel f \parallel_{L_2} \leq c_2 \parallel \boldsymbol{d}_f \parallel_{\ell_2} \tag{7}$$

holds for some positive constants c_1, c_2 independent of f. Obviously in the special case when the $\psi_{j,k}$ form an orthonormal basis, (7) holds with $c_1 = c_2 = 1$ and hence equality holds between coefficient and function norm. In general (7) implies the existence of a *dual basis* $\tilde{\mathbf{\Psi}}$ in L_2 which is *biorthogonal* to $\mathbf{\Psi}$, i.e.

$$\langle\langle \mathbf{\Psi}, \tilde{\mathbf{\Psi}} \rangle\rangle = \boldsymbol{I} \tag{8}$$

where we use the shorthand notation $\langle\langle \mathbf{\Psi}, \tilde{\mathbf{\Psi}} \rangle\rangle = (\langle \psi_{j,k}, \psi_{i,l} \rangle)_{(j,k),(i,l)}$ and $\langle \cdot, \cdot \rangle$ is the standard L_2 inner product. Thus the wavelet coefficients $d_{j,k}$ are given by $\langle f, \tilde{\psi}_{j,k} \rangle$.

Clearly, (7) means that small perturbations of the coefficients cause only small changes of f in the norm $||\cdot||_{L_2}$ and vice versa. In particular, retaining only the first N largest coefficients provides up to a uniform constant the best approximation to f that can be composed from any selection of N basis functions, which is the basis for nonlinear compression techniques. Accordingly the objective of *adaptive* approximation is to successively track the most significant coefficients of the searched function.

Furthermore, properly scaled wavelet bases turn out to form Riesz bases for other function spaces such as Sobolev spaces for a certain regularity range depending on ψ.

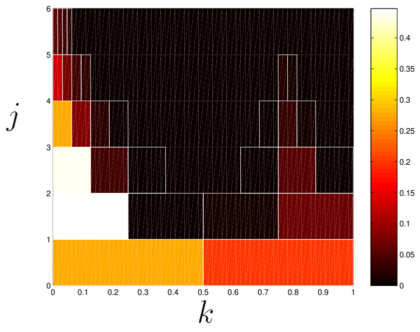
The starting point for the construction of wavelet bases is usually a so called *multiresolution sequences* of nested spaces $S_{j_0} \subset \ldots \subset S_j \subset S_{j+1} \subset \ldots$ whose union is dense in L_2. Here j_0 stands for some coarsest discretization level. The spaces S_j are spanned by scaling functions, $S_j = \text{span}\,\{\varphi_{j,k} : k \in \mathcal{I}_j\}$, which are obtained by a suitable, compactly supported function φ. Due to the compact support of φ we can view 2^{-j} as the uniform mesh size of S_j. Given a dual sequence $\tilde{S}_j$, a successive decomposition of S_j leads to the multiscale splitting $S_J = S_{j_0} \bigoplus_{l=j_0}^{J-1} W_l$ with $W_j = \text{span}\,\{\psi_{j,k} : k \in \mathcal{I}'_j\}$ fulfilling the biorthogonality condition (8).

All above properties can be realized when choosing φ as the cardinal B–Spline of order m. Moreover, one can construct for any $\tilde{m} \in I\!N, \tilde{m} \geq m, m + \tilde{m}$ even, a compactly supported dual scaling function $\tilde{\varphi}$ such that the dual multiresolution spaces $\tilde{S}_j$ contain all polynomials up to order $\tilde{m}$. As a consequence of biorthogonality, one has the *moment conditions*

$$\langle t^r, \psi \rangle = 0, \qquad r = 0, \ldots, \tilde{m} - 1 \; . \tag{9}$$

Likewise one has $\langle t^r, \tilde{\psi} \rangle = 0$ for $r = 0, \ldots, m - 1$. Therefore, smooth functions have wavelet expansions with rapidly decaying wavelet coefficients so that levelwise truncation of wavelet expansions provide good approximations. However, as soon as singularities occur different nonuniform selections of basis functions are expected to provide more economical approximations. In fact, combining the above statements with the locality of wavelets, large wavelet coefficients reflect a large local change of the function. The adaptive choice of Λ will be based on this fact. Figure 3 illustrates a typical way of space–scale representation of the wavelet coefficients for an adaptive set of coefficients Λ. Each block represents one coefficient $d_{j,k}$ to a basis function $\psi_{j,k}$ where j is the scale (frequency) and k is the locality.

FIGURE 3. Space–scale representation



4 Discretization: wavelets as trial– and test–functions

For the discretization of the optimization problem we scale the time horizon to $[0,1]$ and formulate the equality constraints in a weak sense. We obtain:

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{w}} \int_0^1 \left\{(\boldsymbol{y}-\boldsymbol{z})^T\boldsymbol{Q}(\boldsymbol{y}-\boldsymbol{z}) \;+\; \boldsymbol{v}^T\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{v} \;+\; \boldsymbol{w}^T\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{w}\right\}\, dt \tag{10}$$

$$\text{s.t.} \qquad \langle \boldsymbol{F}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{w}),\boldsymbol{\nu}\rangle \;=\; \mathbf{0} \qquad \text{for all } \boldsymbol{\nu}\in L_2^{n_x+n_y} \tag{11}$$

$$\boldsymbol{n}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{w}) \;\leq\; \mathbf{0} \tag{12}$$

Discretization is then given by the representation of each function with respect to an appropriately chosen wavelet basis $\boldsymbol{\Psi}$, i.e. we represent $\boldsymbol{x}$ as $\boldsymbol{x}=\boldsymbol{\Psi}_{\Lambda_x}^T\boldsymbol{d}_{\boldsymbol{x}}$, $\boldsymbol{y}$ as $\boldsymbol{y}=\boldsymbol{\Psi}_{\Lambda_y}^T\boldsymbol{d}_{\boldsymbol{y}}$ etc.. We obtain an equivalent, *infinite dimensional* but *discretized* optimization problem for the wavelet coefficients $\boldsymbol{\zeta}_\Lambda=(\boldsymbol{d}_{\boldsymbol{x}}^T,\boldsymbol{d}_{\boldsymbol{y}}^T,\boldsymbol{d}_{\boldsymbol{v}}^T,\boldsymbol{d}_{\boldsymbol{w}}^T)^T$:

$$\min_{\boldsymbol{\zeta}_\Lambda} \boldsymbol{\zeta}_\Lambda^T\boldsymbol{H}_\Lambda\boldsymbol{\zeta}_\Lambda \;+\; \boldsymbol{b}_\Lambda^T\boldsymbol{\zeta}_\Lambda \tag{13}$$

$$\text{s.t.} \qquad \boldsymbol{F}_\Lambda(\boldsymbol{\zeta}_\Lambda) \;=\; \mathbf{0} \tag{14}$$

$$\boldsymbol{n}_\Lambda(\boldsymbol{\zeta}_\Lambda) \;\leq\; \mathbf{0}\,. \tag{15}$$

In our particular case we use piecewise linear, continuous functions for $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{v}$ and the test–functions for the output equations (3), while we employ piecewise constant wavelets for $\boldsymbol{w}$ and the test–functions for the differential equations (2). This corresponds to the minimal regularity for a conforming discretization.

While the above formulation yields still the exact solution, for numerical treatment we have to choose finite dimensional approximations. Hence, for each involved function we have to choose a finite set of basis functions, described by their index set $\Lambda_{\boldsymbol{x}}$ etc.. As mentioned, we will start with some initial possibly small index sets $\Lambda_{\boldsymbol{x}}^0$ etc. and *adaptively* enlarge these index sets to $\Lambda_{\boldsymbol{x}}^{\ell+1}=\Lambda_{\boldsymbol{x}}^{\ell}+\Gamma_{\boldsymbol{x}}$ in each refinement step.

5 Treatment of the Optimization Problem

To realize the overall concept, which is designed for the particular real time requirement, we face many new problems on each level of the solver. The following sketch gives a short overview of the different levels and our current approach.

Optimization problem
adaptive refinement employing
– sensitivity of the cost functional with respect to $\boldsymbol{w}$ and $\boldsymbol{v}$ $\longrightarrow \Gamma_{\boldsymbol{w}}, \Gamma_{\boldsymbol{v}}$
– local error estimator for the differential equation $\longrightarrow \Gamma_{\boldsymbol{x}}$

> **NLP**
> – black–box algorithm using a SQP–method
> – efficient functional evaluation for $\langle f(\boldsymbol{\Psi}_\Lambda^T \boldsymbol{d_x}), \psi_{j,k}\rangle$ for given $\boldsymbol{d_x}$
>
> > **QP**
> > *treatment of the inequality constraints*
> > – inner–point methods for the first approximations
> > – active–set methods with initialization of the active sets for the refinement steps
> >
> > > **Linear system of equations**
> > > *iterative solver (simplified Uzawa)*
> > > – employing previous solution as initial value
> > > – using adapted Jacobi preconditioner motivated by wavelet theory

The outermost level correspond to the discretization of the whole optimization problem and the adaptive refinement. The goal is to spend minimal computational effort for realizing a fixed decay rate of the current error in each refinement step. The next subsection will shortly present our algorithm and some numerical results. Then follows the treatment of the nonlinearity. The optimization problem arising at each stage is currently solved by a black–box algorithm using an SQP–method. Due to the nature of the discretization we have to evaluate inner products of the form $\langle f(\boldsymbol{x}_\Lambda), \psi_{j,k}\rangle$ where $\boldsymbol{x}_\Lambda$ is a current wavelet expansion and f is a nonlinear function. This task arises in many other applications of wavelet concepts to variational problems. While straightforward quadrature would spoil the complexity gain by adaptive refinements more sophisticated efficient evaluation schemes have meanwhile been developed, see e.g. [12]. On the level of the linearized optimization problem the outer loop is concerned with the treatment of the inequality constraints. Here, we considered the question whether inner–point methods or active–set methods can exploit the information of the previous refinement step in a more efficient way. Results will be given in Subsection 5.2. The main focus of the remaining paper will be however on the solution process of the arising linear systems. The goal is that the refinement cost for a fixed error reduction should remain proportional to the current size of the problem. This immediately suggests the idea to use an iterative solver rather than a direct solver.

5.1 Adaptive Refinement

Briefly we sketch now the adaptive refinement step and refer the reader for a detailed discussion to [6]. We use the following *adaptive refinement algorithm*, where Γ^ℓ is the set of newly added indices:

1. solve the problem with index set Λ^ℓ
2. refine $\boldsymbol{z}$ and $\boldsymbol{u}$ by *thresholding* $\longrightarrow \Gamma^\ell_{z_i}, \Gamma^\ell_{u_i}$

3. apply *local error* estimator for $\boldsymbol{x}$, $\boldsymbol{y} \longrightarrow \Gamma^\ell_{x_i}, \Gamma^\ell_{y_i}$

4. use *sensitivity* of Lagrange function for $\boldsymbol{v}$, $\boldsymbol{w} \longrightarrow \Gamma^\ell_{v_i}, \Gamma^\ell_{w_i}$

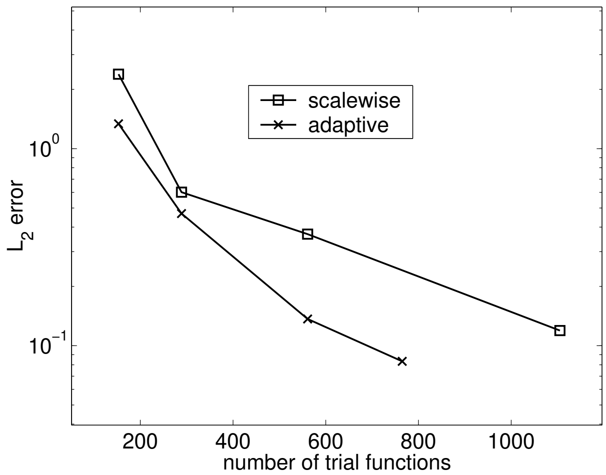
5. *compress* current approximation by thresholding $\longrightarrow \tilde{\Lambda}^\ell_{x_i}$ etc.

6. take *interactions* into account and use $\tilde{\Lambda}^\ell$, $\Gamma^\ell \longrightarrow \Lambda^{\ell+1}$
   then go to 1.

Step 2 is based on the norm equivalences of the form (7). They imply that using the largest wavelet coefficients gives nearly the best N–term approximation in the sense explained above. Hence, we sort the wavelet coefficients of $\boldsymbol{z}$ and $\boldsymbol{u}$ by their absolute value and increase the number of active coefficients in each step. In step 3 we use a local trapezoidal rule and evaluate the error at the midpoints. Where the largest error appears we increase the set of wavelet basis functions by adding to the current set the neighbouring functions in time and in frequence. In contrast to the local error estimator used for the states and output functions, the problem formulation allows us to employ the sensitivity of the Lagrange function with respect to the model error functions $\boldsymbol{v}$ and $\boldsymbol{w}$, since it facilitates easy–to–compute matrix–vector multiplications. While the first steps increase the index set, step 5 reduces the index set by compressing the current approximation. Basis functions with very small coefficients can be discarded thanks to the norm equivalence (7). Finally the different index sets for the various functions have to be combined to take their close interactions into account.

Figure 5.1 illustrates for an example process the comparison between the scalewise choice, i.e. uniform meshes, and an adaptive choice of index sets. The L_2–error is plotted versus the number of trial functions. We see, for example, that for an error 0.105 we need only roughly 550 basis functions with an adaptive approach while around 1000 are necessary with a uniform mesh (mesh size 2^{-10}). Obviously adaptivity outperforms uniform discretization with regard to work/accuracy balance.

FIGURE 5.1. Scalewise versus adaptive refinement



5.2 Inequality constraints

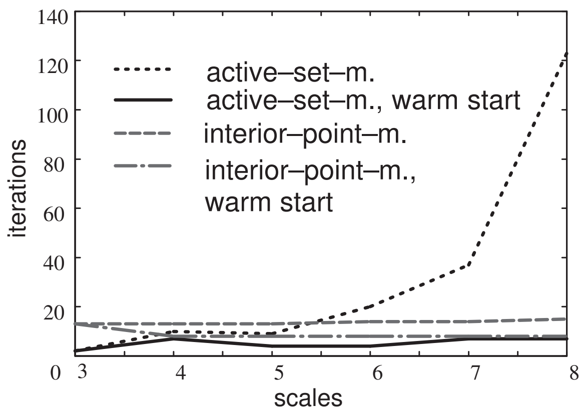
As on all levels of the solver we would like to exploit the information from the previous refinement step for treating the inequality constraints . We compare two common approaches.

On one hand, interior–point methods are often applied to very large systems. They usually need a fixed number of iterations independently of the dimension of the system. Hence they suggest themselves in the present context. On the other hand, the active–set methods need to solve smaller linear systems in each iteration step. Nevertheless, the number of iterations typically increases with the size of the optimization problem and the number of inequality constraints. This typically happens if one solves the optimization problem in a single run. In our refinement algorithm, though, this will be the case only in the first loop. For the next refinement we can employ the information from the previous step. In case of the interior–point method we can use the solution as initial guess for the refined optimization problem. Nevertheless, the current solution has most likely an active set and the iterates of the interior–point method tend to stay away from the boundary. So, it could happen, that the current solution tends to diverge during the iteration first away from the actual solution on this mesh, even if they are initially very close.

In case of the active–set method we can initialize the active–set with the set for which the current solution is active on the refined mesh. Since the current solution should be a good approximation of the refined solution, we do not expect significant changes of the active set and, therefore, we do not expect a large increase of iteration numbers.

Numerical examples show that this approach is indeed very effective. In contrast, for interior–point methods the initial value does not effect the iteration number much. The following Figure 5.2 illustrates a typical behaviour of the iteration numbers for the interior–point methods with and without setting the initial value, as well as the iteration numbers for the active–set method with and without initializing the active set. In this study we employ scalewise refinement.

FIGURE 5.2. Treatment of the inequalities



Of course, the dimensions of the systems to solve in each iteration are different for the active–set method, while they are constant for the interior–point method. In the latter one (much) larger systems arise though. Initialization drops the iteration number for the interior–point method only by two, while the numbers for the active–set method turn to be independent of the size of the system and are even less than for the interior–point method. Therefore, our strategy is to use an interior–point method for the first approximate solution of the optimization problem and for all following refinements we use the active–set method with initialization based on the previous solution. Details for our approach can be found in [4].

6 Linear algebra solver

The saddle point problem arising from the discretized optimization problem for active set methods takes the form:

$$\boldsymbol{\mathcal{L}}_\Lambda \begin{pmatrix} \boldsymbol{\Upsilon}_\Lambda \\ \boldsymbol{\lambda}_\Lambda \end{pmatrix} := \begin{pmatrix} \boldsymbol{\mathcal{A}}_\Lambda & \boldsymbol{\mathcal{B}}_\Lambda{}^T \\ \boldsymbol{\mathcal{B}}_\Lambda & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Upsilon}_\Lambda \\ \boldsymbol{\lambda}_\Lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{h}_\Lambda \\ \mathrm{g}_\Lambda \end{pmatrix} . \tag{16}$$

For convenience we consider an optimization problem without inequality constraints and nonlinearity, although the ideas can be extended.

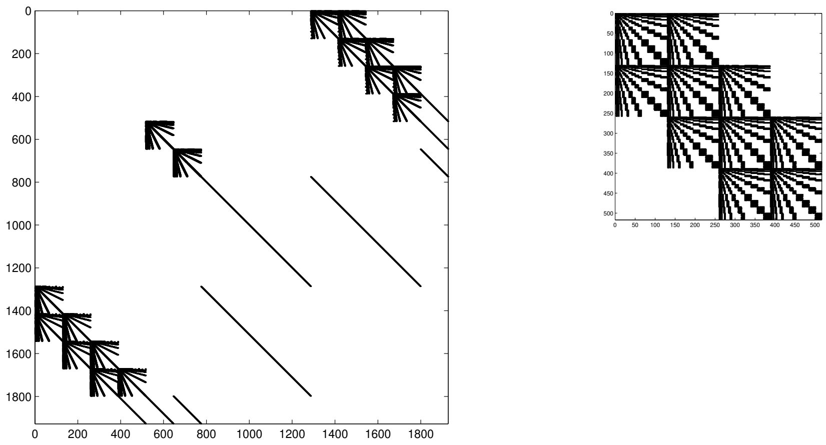
$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{w}} \int_0^1 \left\{(\boldsymbol{y}-\boldsymbol{z})^T \boldsymbol{Q}(\boldsymbol{y}-\boldsymbol{z}) \quad + \quad \boldsymbol{v}^T \boldsymbol{R}_v \boldsymbol{v} + \boldsymbol{w}^T \boldsymbol{R}_w \boldsymbol{w}\right\} \, d\tau \tag{17}$$

$$\text{s.t.} \qquad \dot{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{w} \;=\; \boldsymbol{B}\boldsymbol{u} \tag{18}$$

$$\boldsymbol{y} - \boldsymbol{C}\boldsymbol{x} - \boldsymbol{V}\boldsymbol{v} \;=\; \mathbf{0}\,. \tag{19}$$

Figure 6 shows the typical structure of the matrix $\boldsymbol{\mathcal{L}}_\Lambda$. It exhibits the finger block structure and diagonal block matrices arising from wavelet scalar products. Each block corresponds to entries of the regularization and model matrices.

FIGURE 6. $\boldsymbol{\mathcal{L}}_\Lambda$ $\qquad\qquad$ $\boldsymbol{\mathcal{S}}_\Lambda$



The discretization error is bounded from below by the approximation error of the functions in the adaptively chosen wavelet space. Hence, it is sufficient to solve the above system of equations only up to an error of this order. The goal is now to use an efficient solver up to an *accuracy* which *corresponds* to the *discretization error on each refinement step.* For the efficiency it is important to exploit the previous approximation. Furthermore, successive error reduction is in the spirit of the overall concept. Hence, while most optimization solvers use direct methods for the linear algebra problem, we focus on *iterative solvers.* Due to the indefiniteness of the system one could think of using GMRES, or, exploiting the symmetry, of using SYMMLQ or MINRES [1, 15]. Nevertheless, to exploit the arising structure as much as possible, we reduce the system and obtain a matrix $\tilde{\boldsymbol{\mathcal{L}}}_\Lambda$, where we can apply the Uzawa algorithm (see [7]). This algorithm is particularly designed for saddle point problems with positive definite $\boldsymbol{\mathcal{A}}_\Lambda$. It basically applies the PCG–method to the Schur complement and, to avoid inversion of $\boldsymbol{\mathcal{A}}_\Lambda$, also applies the PCG–method to $\boldsymbol{\mathcal{A}}_\Lambda$. In our case the last mentioned, inner application of the PCG–method is not necessary, since it is very efficient to actually set up $\boldsymbol{\mathcal{A}}_\Lambda^{-1}$. This means, that we apply essentially the PCG–method to the Schur complement $\boldsymbol{\mathcal{S}}_\Lambda$ of the reduced system. For a detailed discussion of this method we refer to [3]. In Figure 6 we can see the reduction of $\boldsymbol{\mathcal{L}}_\Lambda$ to the Schur complement $\boldsymbol{\mathcal{S}}_\Lambda$. The problem size is reduced from a size corresponding to all functions $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and the Lagrange parameter functions to the size of the discretization of $\boldsymbol{x}$. Furthermore, $\boldsymbol{\mathcal{S}}_\Lambda$ is still very sparse and has finger structure. Nevertheless, it will never be set up explicitly to avoid matrix multiplications.

In connection with the applications of iterative methods there arise, of course, two essential questions: the stopping criteria and the preconditioning. These two questions will be discussed in the next section in the context of nested iterations, which is applied to the sequence of refined optimization problems.

7 Nested Iteration

The refinement concept leads to a sequence of equations: for $j = j_0, j_0+1, \ldots$

$$\tilde{\boldsymbol{\mathcal{L}}}_{\Lambda_j} \begin{pmatrix} \tilde{\boldsymbol{\lambda}}_{\Lambda_j} \\ \tilde{\boldsymbol{\Upsilon}}_{\Lambda_j} \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{h}}_{\Lambda_j} \\ \tilde{\mathbf{g}}_{\Lambda_j} \end{pmatrix} \tag{20}$$

with increasing complexity.

The *stopping criteria* for the iterative solver hinges on three properties: firstly, the maximal reduction rate α of the approximation error in the energy norm; secondly, the convergence order of the PCG–method, which can be estimated in terms of $\kappa = \mathrm{cond}_2(\boldsymbol{M}^{1/2}\boldsymbol{\mathcal{S}}_\Lambda\boldsymbol{M}^{1/2})$ where $\boldsymbol{M}$ is the preconditioner; and finally on the norm equivalences induced by the wavelets leading to a constant c. These properties can be used to determine a maximal number of iteration

$$max_{it} \approx log\,(2c/\alpha)\,/\,\left(-log\left((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)\right)\right) \tag{21}$$

and a maximal relative error

$$\alpha_{rel} \approx \alpha/(c\sqrt{\kappa}) \tag{22}$$

to obtain the best possible convergence order during the refinement process (see [3]). In general the exact error reduction factor α is not computable. For our particular discretization we expect at best a first order convergence in the energy norm. Hence, in terms of degrees of freedom we take $\alpha = \#\Lambda_{j-1}/\#\Lambda_j$, which is $1/2$ for uniform mesh refinement. Moreover, to determine α_{rel} and max_{it} an upper bound for κ has to be estimated. Choosing $\alpha_{rel} = 10^{-3}$ for all j would cover condition numbers κ up to 1700. Nevertheless, to play safe we set $\alpha_{rel} = 10^{-4}$. Moreover, since max_{it} predicts the worst case we mostly use $max_{it} = 20$ as a pragmatic choice, which rigorously covers only condition numbers up to around 100.

The minimization problem has differential equations as constraints. This implies condition numbers increasing with the refinement level. Norm equivalences based on wavelets suggest a *level depending scaling* 2^{-j}, where j denotes the level, to obtain an upper bound for the eigenvalues. However, since the discretization is a Galerkin discretization of a saddle point problem, we require that the LBB–condition is fulfilled, to prove the existence of a lower bound. Additionally, we need the well posedness of the optimization problem with respect to the H^1–norm. Then, in [3] it is shown, that scale dependent preconditioning leads to eigenvalues bounded independently of the used index set Λ

$$0 < c_1 \leq \lambda(\boldsymbol{M}_{\Lambda_j}^{1/2}\boldsymbol{\mathcal{S}}_{\Lambda_j}\boldsymbol{M}_{\Lambda_j}^{1/2}) \leq c_2\ , \tag{23}$$

and therefore to bounded condition numbers. Instead of 2^{-j} we use an *approximate Jacobi preconditioner*, i.e. an approximation of the diagonal of the $\boldsymbol{\mathcal{S}}_{\Lambda_j}$ which can be set up very efficiently. Their entries reflect the necessary level dependent scaling. Furthermore, numerical experiments show, that it is more efficient than the scaling by powers of two, which corresponds to normalizing the wavelet basis in the energy norm and usually gives better results (see [3]).

The algorithm of *nested iteration* reads then as follows:
set $j := j_0$ and solve (20) up to machine precision;

- *update* Λ_j and the representation of $\boldsymbol{z}$ and $\boldsymbol{u}$
- *extend* the discretization *matrices*
- set the initial guess $\tilde{\boldsymbol{\Upsilon}}^{(0)}_{\Lambda_j} \longleftarrow \tilde{\boldsymbol{\Upsilon}}^{(k)}_{\Lambda_{j-1}}$
- apply the *approximate Jacobi* preconditioner $\boldsymbol{M}_{\Lambda_j}$
- iterate the *simplified Uzawa* algorithm until
  a.) relative residual in ℓ_2–norm is below α_{rel} or
  b.) maximal number of iteration max_{it} is reached
- go to the first step

We illustrate the effects on condition number (Figure 7) and the results of the nested iteration process (Table 1) with a simple example. We consider the minimization problem $min \int_0^1 \{(\boldsymbol{y}-\boldsymbol{z})^2 + \boldsymbol{v}^2 + \boldsymbol{w}^2\}\ d\tau$ with the constraints $\dot{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{u} - \boldsymbol{w} = \boldsymbol{0}$, $\boldsymbol{y} - \boldsymbol{x} = \boldsymbol{0}$ and four state and output functions as well as one control function, and solve it by uniform refinement. In Figure 7 we can clearly see the level dependent increase of $cond(\boldsymbol{\mathcal{S}}_\Lambda)$, while preconditioning with approximate Jacobi preconditioner leads to nearly constant behaviour. The corresponding max_{it} determined by the exact condition number is between 15 and 17.

FIGURE 7. Condition numbers

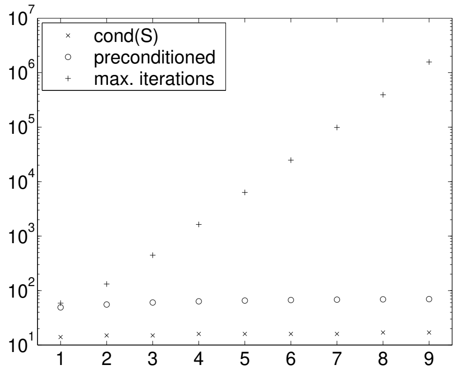


Table 1 lists the used refinement levels, the corresponding size of the system of equations, the number of iterations needed for the nested iteration process stated above, the error e between the wavelet coefficients of the current approximation to the exact solution, the ratio of error reduction e_j/e_{j-1} and the actual residuum of the equation system on each level. The last line gives the results for applying the simplified Uzawa algorithm on level 9 without using any previous information.

Due to the geometric progression of the problem sizes we see that the computational cost for nested iteration up to scale 9 corresponds to 20 or 21 iterations of the simplified Uzawa algorithm applied to the system on level 9 only. Perhaps more important than the expected higher efficiency of nested iteration is the fact it provides successively improved

Table 1: Results corresponding to nested iteration

Level j	dim	k	e_j	e_j/e_{j-1}	$\|res\|_{\ell_2}$
1	12	16	9.4525e−01		6.0166e−15
2	20	17	8.8248e−01	1.07	4.0636e−04
3	36	17	2.4626e−01	3.58	4.9402e−03
4	68	17	6.6437e−02	3.70	3.4785e−03
5	132	17	1.7124e−02	3.87	2.6048e−03
6	260	17	4.3262e−03	3.95	2.3966e−03
7	516	15	1.0811e−03	4.00	2.2025e−03
8	1028	14	2.6419e−04	4.09	2.5533e−03
9	2052	10	6.3861e−05	4.13	2.1355e−03
9	2052	34	5.8380e−05		2.4164e−03

approximations at a much earlier stage. We can see also that it is not necessary to determine on each scale the exact solution. The best approximation error is closely reached, although the residua are still quite large and in full agreement with the size of the condition numbers in this example. The quotient e_{j-1}/e_j is as good as an exact solver can provide to produce meaningful results associated to discretization accuracy.

Nevertheless, we have also examples with condition numbers up to 10^{20} where equilibration of the states, preconditioning, and choosing more appropriate regularization parameters $\boldsymbol{R_w}$ lead to condition numbers of 10^6. These are still too large to make predictions for the iteration process. The eigenvalue distribution is much more in favour for superconvergence of the method, though, this issue has still to be investigated more thoroughly. In the next section we will briefly outline several sources of the ill conditioning.

8 Inverse Problem – Observability

As mentioned in the beginning, state estimation is an ill–posed inverse problem. Even with our regularized problem formulation (1) – (5), we typically face three different problems:

1. the signal functions $\boldsymbol{z}$ are corrupted by noise and do *not* lie in H^1,
2. the model equations (2), (3) (without $\boldsymbol{v}$ and $\boldsymbol{w}$) have often a (very) low *observability measure* (see [3]),
3. the amount of regularization of $\boldsymbol{v}$ and $\boldsymbol{w}$ may be inadequate.

In particular 2. and 3. play an important role for the constants involved in the estimation of the well posedness with respect to H^1, and hence in the bounds c_1 and c_2 of the eigenvalues (23) of the preconditioned Schur complement. For example, the interrelation of the model matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ for linear constraints is essential in the estimations of the LBB–condition (see [7]), which –ignoring $\boldsymbol{v}$ and $\boldsymbol{w}$– leads to an estimation

$$\| \begin{pmatrix} -\dot{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{x} \\ \boldsymbol{C}\boldsymbol{x} \end{pmatrix} \|_{(L_2)^{n_x}\times(L_2)^{n_y}} \geq \beta||\boldsymbol{x}||_{(H^1)^{n_x}}$$

where β could be very small. Moreover, we immediately see in case of a one dimensional problem with $\boldsymbol{C} = \delta$ and $\boldsymbol{A} = \alpha$ that large eigenvalues of $\boldsymbol{A}$ are unfavourable, since for

$\boldsymbol{x} = e^{\alpha t}$ we have

$$\| \, (-\dot{\boldsymbol{x}} + \boldsymbol{A}\boldsymbol{x}, \boldsymbol{C}\boldsymbol{x})^T \, \|_{L_2 \times L_2} = \frac{\delta}{1+|\alpha|} \, ||\boldsymbol{x}||_{H^1} \; .$$

This reflects the fact, that no initial condition is included in the constraints, since for state estimation the initial data have to be estimated.

Numerical considerations confirm, that large in modulus eigenvalues of $\boldsymbol{A}$ and low observability measures lead to very small lower bounds of the eigenvalues of the preconditioned and unpreconditioned Schur complement. This means that the high condition numbers are mainly problem inherent and will occur for any discretization.

Possible improvements of this situation may result from a more sophisticated choice of $\boldsymbol{R_v}$, $\boldsymbol{R_w}$ and/or an additional *scale depending regularization* of $\boldsymbol{x}$ with

$$\int_{\Omega_R} \left\{ (\boldsymbol{x} - \bar{\boldsymbol{x}})^T \boldsymbol{R_x} (\boldsymbol{x} - \bar{\boldsymbol{x}}) \right\} \, d\tau$$

in the cost functional (1), which should fade out for progressing levels. On the other hand, currently we develop and investigate a *preconditioner*, which eliminates the smallest eigenvalues based on the *adaptive refinement algorithm.*

9 Conclusions

We have developed and investigated several components of an iterative multiscale concept, that aims at realizing the real–time requirement encountered in state estimation. The relevant main ingredients for the adaptive treatment of the linear problem formulation with inequality constraints are now essentially available. We had to face in the development of these components a diversity of problems, some of which do only occur due to this particular real–time requirement whereas others are problem inherent to process monitoring. Nevertheless, there are still many open tasks and questions: the above mentioned choice of regularization and the improvement of the preconditioner, the extension to DAE's with appropriate wavelets as test functions, the development and extension of the current method to a nonlinear model, which is in the spirit of our conceptional framework, the improvement and thorough analysis of the adaptive refinement, as well as moving the horizon, while exploiting efficiently the solution on the previous horizon. Moreover, due to problem 1., we have to investigate the length of the horizon and its influence on the estimation quality given noisy data. In this context we discussed in [5] the regularization effect of the discretization. Here, the above adaptive refinement algorithm generates a hierarchy of discretization grids until a best compromise between data and regularization error is achieved.

References

[1] O. Axelsson, *Iterative Solution Methods,* Cambridge University Press, New York, **1994**

[2] T. Binder and L. Blank and W. Dahmen and W. Marquardt, *Towards Multiscale Dynamic and Data Reconciliation,* Nonlinear Model Based Process Control, NATO ASI, in R. Berber and C. Kravaris, Kluwer Academic Publishers, **1998**, 623–665

[3] T. Binder and L. Blank and W. Dahmen and W. Marquardt, *Iterative Algorithms for Multiscale Dynamic Data Reconciliation,* RWTH Aachen, IGPM–Report, **186**, 2000

[4] T. Binder and L. Blank and W. Dahmen and W. Marquardt, *Grid Refinement in Multiscale Dynamic Optimization,* Proceedings: ESCAPE-10, Florence, Italy, 7-10.5.2000, **2000**

[5] T. Binder and L. Blank and W. Dahmen and W. Marquardt, *Regularization of Dynamic Data Reconciliation Problems by Projection,* Proceedings: ADCHEM 2000. International Symposium on Advanced Control of Chemical Processes, Pisa, Italy, 14-16.6.2000, **2000**

[6] T. Binder and L. Blank and W. Dahmen and W. Marquardt, *An Adaptive Multiscale Method for Real-Time Moving Horizon Optimization,* Proceedings: American Control Conference 2000, Chicago, USA, 28-30.6.2000, **2000**

[7] D. Braess *Finite Elemente,* Springer–Verlag, Berlin, **1992**,

[8] A. Chambolle and R. DeVore and N.Y. Lee and J. Lucier, *Nonlinear Wavelet Image Processing: Varational Problems, Compression, and Noise Removal through Wavelet Shrinkage*, IEEE Image Processing **7**, (1998), 319-335

[9] C.K. Chui, *An Introduction to Wavelets,* Academic Press, Boston **1992**

[10] A. Cohen and I. Daubechies and J.C. Feauveau, *Biorthogonal Bases of Compactly Supported Wavelets,* Comm. Pure and Appl. Math., **45** (1992), 485–560

[11] W. Dahmen, *Wavelet and Multiscale Methods for Operator Equations,* Acta Numerica, **1997**, 55–228

[12] W. Dahmen and R. Schneider and Y. Xu, *Nonlinear Functionals of Wavelet Expansions – Adaptive Reconstruction and Fast Evaluation,* Numer. Math. **86**, No.1 (2000), 49-101

[13] I. Daubechies, *Ten Lectures on Wavelets* Society for Industrial and Applied Math., Philadelphia **1992**

[14] H. Michalska and D.Q. Mayne, *Moving Horizon Observers,* IFAC Symposium NOLCOS'92, Bordeaux, France **1992**

[15] G. Meurant, *Computer Solution of Large Linear Systems,* Elsevier Science B.V., Amsterdam, The Netherlands, **1999**

[16] K.R. Muske and J.B. Rawlings, *Nonlinear Receding Horizon State Estimation* in R. Berber, Methods of Model–Based Control, NATO–ASI Series, Kluwer Press, Dordrecht, The Netherlands, **1995**, 489–504

[17] D. Robertson and J.H. Lee and J.B. Rawlings, *A Moving Horizon–Based Approach for Least–Squares Estimation,* AIChE J., **42**, No. 8,(1996), 2209–2223