MENCK, J.

Fast Analysis of the Model in Approximate Newton-Like Coupling

The present paper fills a gap that we felt our description of the Approximate Tangential Block Newton method in [3], [2] and [4] left: The method involved solving a two–dimensional discrete minimization problem at each iteration step. Our previous papers did not address the details of this task but since the model problem has a specific structure, it seems sensible to think about specific ways of solving it, and that is what we do here.

Address. *Jürgen Menck*, Technische Universität Hamburg-Harburg, Arbeitsbereich Mathematik, Schwarzenbergstraße 95, D-21073 Hamburg, Federal Republic of Germany, `http://www.tu-harburg.de/mat/, menck@tu-harburg.de`

Keywords. Newton's method, approximate Newton's method, block-structured Newton's method, coupled systems, stationary process simulation, work control.

1 Introduction

In [3], [2] and [4], we proposed a method for solving large coupled systems of equations,

$$\begin{aligned} f &= 0 \,, \\ g &= 0 \,. \end{aligned}$$

For our influences and a list of related work (as well as more details on the method), please refer to the above papers. The aim was to use a given iterative solver for $f = 0$ and everything else that was related to f and still give the algorithm a flavor of Newton's method. Since it would be too expensive to force quadratic convergence under these circumstances, we proposed to settle for a linearly convergent approximation of Newton steps. We called the method ATBN or the Approximate Tangential Block–Newton method.

Owing to a block elimination inherent in the method, each step would be composed of two partial steps that involved a number of parameters, and we proposed to use a model of the step to choose approximate values for them, thus trying to optimize the efficiency of the method.

According to said model, the effect of the first half step on f and g can be described as

$$\|f^+\| \approx ((1-\alpha) + \alpha q^{\kappa_1})\|f_n\| \,, \tag{1}$$

$$\|g^+\| \approx \|g_n\| + \alpha\mu\frac{1-q^{\kappa_1}}{1-q}\|f_n\| \,, \tag{2}$$

that of the second half step as

$$\|f_{n+1}\| \approx \|f^+\| + \beta(1+\varepsilon_1)q^{\kappa_2+1}\lambda\|g^+\| \,, \tag{3}$$

$$\|g_{n+1}\| \approx ((1-\beta) + \beta\varepsilon_1)\|g^+\| \,. \tag{4}$$

The quality of the current approximate solution is judged by the maximum of both residua,

$$m_n := \max(\|f_n\|, \|g_n\|) \,. \tag{5}$$

Basically, the first half step is designed to reduce the norm of f, while the second half step will reduce the norm of g. But there is an important difference: While the first half step can damage the norm of g freely, κ_2 provides a handle for diminishing the negative effects of the second half step on f.

q, λ and μ are not method parameters. Instead, q represents the convergence rate of the f related iteration while λ and μ describe the strength of the coupling between f and g; they are all provided by earlier ATBN steps. ε_1 is a small positive parameter that controls the accuracy of an iterative linear solver used inside the second half step.

We prefer to keep this number fixed during the course of the iteration because the relationship between it and the number L introduced below is largely unpredictable. The mentioned linear solver is not to be mixed up with the f related iterative solver. For clarity and brevity, we will call it GMRES in the sequel.

Now four method parameters remain to be chosen for each upcoming ATBN step, $\kappa_1 \in \mathbb{N}$, $\kappa_2 \in \mathbb{N}$, $0 < \alpha \leq 1$ and $0 < \beta \leq 1$. α seems to be the natural parameter for damping the composite step. In a suitable neighbourhood of the solution, it can be chosen to be 1, and we will assume in this paper that it is at least close to 1. β however, although it is technically a damping parameter for the second half step, is best chosen to be slightly smaller than 1; in fact, it can be used to balance the values of $\|f\|$ and $\|g\|$ such that they will almost be identical at the end of the step, which is a reasonable aim considering our choice of m_n as an error estimate (cf. (5)). Assuming that β is chosen accordingly, the result of the composite step can be written as

$$m_{n+1} \approx \max(\varepsilon_1 \|g^+\|, m^*_{n+1}) \ , \quad \text{where} \tag{6}$$

$$m^*_{n+1} := \frac{(1-\varepsilon_1)\|f^+\| + (1+\varepsilon_1)\lambda q^{\kappa_2+1}\|g^+\|}{(1-\varepsilon_1) + (1+\varepsilon_1)\lambda q^{\kappa_2+1}} \ . \tag{7}$$

(See [3] for details.) This leaves us with only two parameters to adjust, namely the κ_i. κ_1 is the number of f related iterations used to treat $f = 0$ in the first half step, κ_2 is the number of iterations used to approximate a certain matrix multiplication associated with the second half step. We proposed to measure the efficiency of the step by computing the average reduction of m_n achieved by one f related iteration. (We assume that the f iterations constitute the dominant costs of the composite step; of course, there are lots of possible modifications.)

Denoting by L the number of approximate matrix multiplications used in GMRES, our implementation of the composite step used

$$N = (3+\kappa_1) + (L+1)(\kappa_2+1) \tag{8}$$

f iterations. (We have replaced ℓ by L in this paper to make the formula independent of the particular type of linear solver used.) Now, the effective reduction per f iteration step is

$$Q = \left(\frac{m_{n+1}}{m_n}\right)^{1/N} \ ,$$

and κ_i would have to be chosen such that they minimize this term. In the former papers, we have treated this part of the algorithm as a black box, but it should be clear that it deserves some thought: If handled carelessly, the two dimensional minimization can become quite costly in itself.

In the present paper, we will point out how to keep the effort low by eliminating yet another variable from the problem beforehand and translating it into a minimization problem on a finite interval that can be solved rather cheaply.

2 A New Formulation of the Model Problem

We can not expect to be able to solve the model problem explicitly. What we can do, however, is eliminate one more degree of freedom. Still, in order to do this, we have to simplify the model slightly. The modification should not be crucial in most situations, and we will compensate for it by re–introducing the omitted terms into the resulting one dimensional model in the final stage.

The present paper is again concerned with the standard situation, which means that $\|f_n\|$ and $\|g_n\|$ are supposed to be close to each other (which they ought to be after a certain startup phase of the algorithm), and in particular neither of them should be zero. Furthermore, we assume that the equations are really coupled in the sense that $\lambda \neq 0$, and of course we demand that the f related iteration be contracting, i.e. $q < 1$, and that we do not have to use severe damping, i.e. $\alpha \approx 1$. A multi purpose software would naturally have to deal with a lot of exceptions, but this is the setting for which the algorithm has been constructed, and algorithmically it is also the most rewarding situation.

We start our analysis by omitting the highest order terms of q in (7), namely the terms of order $q^{\kappa_1+\kappa_2+1}$ in the denominator. This will allow us to eliminate one of the two degrees of freedom further below. The natural way to

achieve this modification is to replace (2) by the modified equation

$$\|g^+\| \approx \|g_n\| + \alpha\mu\frac{1}{1-q}\|f_n\| , \tag{9}$$

and leave everything else as it is. If this does not seem to be apparent, it can easily be verified by tracking back the derivation of the formulas in [3]. Hence we basically replace our estimate of the side effect that the first half step has on g by a slightly more pessimistic variant.

Next we substitute new real variables for the natural numbers κ_i and work out the details of Q both for the original and the simplified models: Let x and y be defined by

$$\begin{aligned} x &:= q^{\kappa_2} , \\ y &:= q^N . \end{aligned}$$

(It should be obvious that these variables bear no direct relation to the state variables that we used to call x and y in the previous papers.) Note that (8) implies

$$q^{\kappa_1} = q^{-4-L} \cdot \frac{y}{x^{1+L}} .$$

In the case of $m_{n+1} = m^*_{n+1}$ the effective reduction Q can be written as

$$Q(x,y) = \phi(x,y)^{\log(q)/\log(y)}$$

where ϕ has the form

$$\phi(x,y) = \frac{ac_1 y + c_1 b_1 x^{1+L} + c_2 b_2 x^{2+L}}{m_n c_1 x^{1+L} + m_n c_2 x^{2+L}} . \tag{10}$$

This is the simplified expression that results from using (9) instead of (2). The original model yields a similar result which we will distinguish by using a tilde:

$$\tilde{Q}(x,y) = \tilde{\phi}(x,y)^{\log(q)/\log(y)}$$

where

$$\tilde{\phi}(x,y) = \frac{ac_1 y - ab_3 xy + c_1 b_1 x^{1+L} + c_2 b_2 x^{2+L}}{m_n c_1 x^{1+L} + m_n c_2 x^{2+L}} . \tag{11}$$

The coefficients in these terms are

$$\begin{aligned} a &= \alpha q^{-4-L}\|f_n\| , \\ c_1 &= 1-\varepsilon_1 , \\ c_2 &= (1+\varepsilon_1)\lambda q , \\ b_1 &= (1-\alpha)\|f_n\| , \\ b_2 &= \|g_n\| + \alpha\mu\frac{1}{1-q}\|f_n\| , \\ b_3 &= \frac{\mu}{1-q}\cdot c_2 . \end{aligned}$$

Due to our assumptions all of these numbers except maybe b_3 are positive. Furthermore, we will assume that

$$c_2(b_2 - b_1) > 0 , \tag{12}$$

which is a more explicit version of our assumption that α be close to 1.

For the case $m_{n+1} = \varepsilon_1\|g^+\|$, Q has the rather simple form

$$\hat{Q}(x,y) = \hat{\phi}(x,y)^{\log(q)/\log(y)}$$

with

$$\hat{\phi}(x,y) = \frac{\varepsilon_1 b_2}{m_n} = \frac{(1-c_1)b_2}{m_n} .$$

This expression is again based on the simplified model. For the correct model, the right hand side would have to be replaced by

$$\frac{(1-c_1)}{m_n}\left(b_2 - a\frac{\mu y}{(1-q)x^{1+L}}\right) \ .$$

In the following section, we will analyze Q for arbitrary but fixed y. This approach has the obvious advantage that we only have to deal with a rational function, since the exponent $1/N$ remains constant. Last but not least, the case $m_{n+1} = \varepsilon_1\|g^+\|$ can be "eliminated" as well: We will show how to choose the domain for the reduced minimization problem such that the reduced Q will contain no trace of $\hat{Q}$.

3 Reducing the problem

We will now attempt to find the minimum of ϕ for arbitrary but fixed y: A straightforward calculation yields

$$\frac{\partial\phi}{\partial x}(x,y) \quad = \quad \left((c_2(b_2-b_1)x^{2+L} - ay((1+L)c_1 + (2+L)c_2x)\right)\frac{c_1}{m_n(c_1+c_2x)^2x^{2+L}} \ .$$

If $x > 0$, the fraction will not influence the sign of the derivative, and we can thus conclude:

T h e o r e m For fixed $y > 0$, $\phi(*, y)$ has a global minimum $\hat{x}$ in $\mathbb{R}_+$ which is characterized by

$$c_2(b_2-b_1)\hat{x}^{2+L} - ay((1+L)c_1 + (2+L)c_2\hat{x}) \quad = \quad 0 \ . \tag{13}$$

P r o o f. If we can show that

$$\psi(x) \quad = \quad c_2(b_2-b_1)x^{2+L} - ay((1+L)c_1 + (2+L)c_2x)$$

has exactly one zero $\hat{x}$ in $\mathbb{R}_+$ and is negative for $x < \hat{x}$ and positive for $x > \hat{x}$, the theorem will follow. To show this property, we first focus on the derivative of ψ,

$$\psi'(x) \quad = \quad (2+L)c_2(b_2-b_1)x^{1+L} - a(2+L)c_2y \, .$$

ψ' has exactly one zero $\bar{x}$ in $\mathbb{R}_+$ and is negative for $x < \bar{x}$ and positive for $x > \bar{x}$. (Remember (12)!) ψ is thus monotonously decreasing for $x < \bar{x}$ and monotonously increasing for $x > \bar{x}$. But $\psi(0) < 0$, and hence ψ can at most have one zero in $\mathbb{R}_+$. On the other hand, $\psi(x) > 0$ for $x \gg 0$, and thus ψ has to have at least one zero. Hence, ψ has exactly one zero.

Since (13) has to be solved for all admissible y, we may as well look at it as a two dimensional problem. Luckily, the equation can be solved for y explicitly, yielding:

T h e o r e m Let y_* denote

$$y_*(x) \quad = \quad \frac{\gamma}{\delta_1 x + \delta_0}x^{2+L}$$

where

$$\begin{aligned} \gamma &= c_2(b_2-b_1) \ , \\ \delta_0 &= (1+L)c_1a \ , \\ \delta_1 &= (2+L)c_2a \ . \end{aligned}$$

Then for every $x > 0$, x minimizes $\tilde{\phi}(*, y)$ for $y = y_*(x)$. In the case $m^*_{n+1} \geq \varepsilon_1\|g^+\|$, it is thus sufficient to minimize Q for all pairs $(x, y_*(x))$ with $x \in (0, 1]$.

Before proceeding with the computational details, we will now study the case $m_{n+1} = \varepsilon_1\|g^+\|$. In an ideal second half step, we expect $\|g\|$ to sink to the level of $\|f\|$, with the damping parameter β preventing $\|f\|$ from dominating the end result. In this situation, m_{n+1} will be the m^*_{n+1} discussed above. But if $\|f^+\|$ was too small in the first place, the second half step may not be able to bring $\|g\|$ down sufficiently, and the resulting m_{n+1} can not be smaller than the value of $\|g\|$ that an undamped second half step can achieve, which is $m_{n+1} = \varepsilon_1\|g^+\|$. Said value of m_{n+1}

will hence occur if and only if for $\beta = 1$, the value of $\|f\|$ after the second half step is smaller than the value of $\|g\|$, i.e.

$$(1 - c_1)b_2 \quad \geq \quad b_1 + a\frac{y}{x^{1+L}} + c_2 b_2 x$$

(according to the simplified model) or, equivalently,

$$y \quad \leq \quad y_1(x) \quad := \quad (\rho_0 - \rho_1 x)x^{1+L} \ ,$$

where

$$\begin{aligned} \rho_0 &= ((1 - c_1)b_2 - b_1)/a \ , \\ \rho_1 &= c_2 b_2 / a \ . \end{aligned}$$

To sum up our results, $y = \max(y_*(x), y_1(x))$ is the ideal choice of y for any given $x \in (0, 1]$. Now what can we say a priori about the x for which y_1 will be active and the ones for which y_* will be active? Both are identical iff

$$(\rho_0 - \rho_1 x)x^{1+L} \quad = \quad \frac{\gamma x^{2+L}}{\delta_1 x + \delta_0} \ ,$$

that is

$$\delta_1\rho_1 x^{3+L} + (\gamma + \delta_0\rho_1 - \delta_1\rho_0)x^{2+L} - \delta_0\rho_0 x^{1+L} \quad = \quad 0 \ .$$

A straightforward computation confirms that this equation can be factorized as in

$$x^{1+L} \cdot ((2 + L)b_2c_2x - (1 + L)(b_2 - b_1 - c_1b_2)) \cdot (c_2x + c_1) \quad = \quad 0 \ .$$

Hence, its solutions are

$$\begin{aligned} x_0 &= 0 \ , \\ x_+ &= \frac{(1 + L)(b_2 - b_1 - c_1b_2)}{(2 + L)b_2c_2} \ , \\ x_- &= -\frac{c_1}{c_2} \ . \end{aligned}$$

x_0 and x_- are irrelevant because they do not belong in $\mathbb{R}_+$, and consequently

$$\max(y_*, y_1) \quad = \quad \begin{cases} y_1 & , \quad x < x_+ \ , \\ y_* & , \quad x \geq x_+ \ . \end{cases}$$

If $x_+ \leq 0$, the former case will not occur at all. But there is a good chance that it will, since x_+ is positive in the undamped case $\alpha = 1$.

We shall now insert our y into the respective ϕ and Q variants to get an explicit description of the function we have to minimize: In the case of $y = y_*(x)$ we have

$$\phi(x, y_*(x)) \quad = \quad \frac{\alpha_2 x^2 + \alpha_1 x + \alpha_0}{\beta_2 x^2 + \beta_1 x + \beta_0}$$

or

$$\tilde{\phi}(x, y_*(x)) \quad = \quad \frac{\tilde{\alpha}_2 x^2 + \alpha_1 x + \alpha_0}{\beta_2 x^2 + \beta_1 x + \beta_0}$$

where

$$\begin{aligned} \alpha_0 &= (1 + L)c_1^2 b_1 \ , \\ \alpha_1 &= ((2 + L)b_2 + (1 + L)b_1)c_1c_2 \ , \\ \alpha_2 &= (2 + L)c_2^2 b_2 \ , \\ \tilde{\alpha}_2 &= \alpha_2 - b_3\gamma \ , \\ \beta_0 &= m_n(1 + L)c_1^2 \ , \\ \beta_1 &= m_n(3 + 2L)c_1c_2 \ , \end{aligned}$$

$$\beta_2 \quad = \quad m_n(2+L)c_2^2 \ .$$

Rather than Q itself, we propose to minimize its logarithm h. Since $N = \log(y_*)/\log(q)$, it takes the form

$$h_*(x) \quad := \quad \log(Q(x, y_*(x))) \quad = \quad \frac{\log(\alpha_2 x^2 + \alpha_1 x + \alpha_0) - \log(\beta_2 x^2 + \beta_1 x + \beta_0)}{(2+L)\log(x) + \log(\gamma) - \log(\delta_1 x + \delta_0)} \cdot \log(q) \ ,$$

and $\tilde{h}_*(x) := \log(\tilde{Q}(x, y_*(x)))$ is the same except α_2 has to be replaced by $\tilde{\alpha}_2$. In the case of $y = y_1$ we have

$$h_1(x) \quad := \quad \log(\hat{Q}(x, y_1(x))) \quad = \quad \frac{\log(q)\log(\tau_0)}{(1+L)\log(x) + \log(\rho_0 - \rho_1 x)}$$

where

$$\tau_0 \quad = \quad \frac{(1-c_1)b_2}{m_n} \ .$$

In analogy to h_*, a corrected variant can be won by computing $\varepsilon_1 \|g_+\|$ in h_1 from (2) instead of (9). The result is

$$\tilde{h}_1(x) \quad = \quad \frac{\log(q)}{(1+L)\log(x) + \log(\rho_0 - \rho_1 x)} \cdot \log\left(\frac{(1-c_1)}{m_n}\left(b_2 - a(\rho_0 - \rho_1 x)\frac{\mu}{1-q}\right)\right) \ .$$

This term plays no role in our algorithm but we will use it to check the reliability of h_1 in our example further down.

It is intuitively clear that y_1 can actually be passed over in the final analysis. To prove this conjecture, we have a look at the derivative

$$h_1'(x) \quad = \quad \frac{\log(q)\log(\tau_0)}{((1+L)\log(x) + \log(\rho_0 - \rho_1 x))^2} \cdot \left(-\frac{1+L}{x} + \frac{\rho_1}{\rho_0 - \rho_1 x}\right) \ .$$

Since $q < 1$ and $\tau_0 < 1$, h_1' must have the same sign as

$$-\frac{1+L}{x} + \frac{\rho_1}{\rho_0 - \rho_1 x} \ .$$

Hence, h_1' is negative for $0 < x < x_+$ and positive for $x > x_+$, which implies that h_1 gets minimal at $x = x_+$, where it coincides with h_* anyway.

4 Conclusion: How to Treat the Model

By assembling the results of the previous sections, we can now formulate an algorithm for solving the model problem which is supposed to yield κ_1 and κ_2 for an upcoming step of the ATBN method. Basically, our analysis would suggest that h_* be minimized over $[x_+, 1]$, but at this point it is easy to improve the model by using $\tilde{h}_*$ instead, thus limiting the effect of the simplification. Practical tests have shown that this modification makes a recognizable difference for larger x while hardly changing the model in the neighbourhood of x_+, which indicates that it was indeed reasonable to analyze the relationship between y_* and y_1 on the basis of the simplified model.

One last topic we should address before formulating the algorithm is the accuracy needed in computing the optimal x_*. Since x represents q^{κ_2}, there is a natural limit to what is really useful: The minimization can be stopped once it can be decided which $\kappa_2 \in \mathbb{N}$ would fit best. Usually, this condition should translate into a very moderate stop criterion. If however x_* is close to 0 and q close to one, it might be useful to relax the condition further. On the bottom line, this is the algorithm we recommend:

Algorithm To find the optimal combination of κ_1 and κ_2, look for the minimum x_* of

$$\tilde{h}_*(x) \quad = \quad \frac{\log(\tilde{\alpha}_2 x^2 + \alpha_1 x + \alpha_0) - \log(\beta_2 x^2 + \beta_1 x + \beta_0)}{(2+L)\log(x) + \log(\gamma) - \log(\delta_1 x + \delta_0)} \cdot \log(q) \ .$$

in $[\max(0, x_+), 1]$. The algorithm can be stopped if $x_l \le x_* \le x_r$ is secured and $x_r/x_l \ge \sqrt{q}$. Replace x_* by the nearest power of q. (To avoid being too strict, one might provide an additional bound for $x_r - x_l$.) Then,

$$\kappa_2 \quad = \quad \frac{\log(x_*)}{\log(q)} \ ,$$

$$\kappa_1 \quad = \quad \frac{\log(\gamma x_*) - \log(\delta_1 x + \delta_0)}{\log(q)} - (4 + L) \ ,$$

and the expected reduction per f iteration step is $\exp(h_*(x_*))$. In a practical implementation, the resulting κ_i may have to be rounded, and both values should be replaced by 1 in case they round to 0.

Computational tests indicate that h_* tends to be rather well behaved, even monotonous, on $[\max(0, x_+), 1]$. Strictly speaking, we have no solid evidence, but it seems justified to settle with one of the simpler minimization algorithms. We used a modification of the Golden Section Search (cf. [1]) to good effect:

A l g o r i t h m A simple modification of the Golden Section Search that computes minima of univariate or monotonous functions h can be formulated as follows:

- Set $\gamma_1 = 0.382$ and $\gamma_2 = 0.618$. Choose interval $[x_L, x_R]$.
- Loop No. 1 (monotonous behaviour):
  - If $h(x_L) < h(x_R)$, set $x_M = \gamma_2 x_L + \gamma_1 x_R$, otherwise $x_M = \gamma_1 x_L + \gamma_2 x_R$.
  - If $h(x_M) < \min(h(x_L), h(x_R))$, leave this loop and proceed further down.
  - If $h(x_L) < h(x_R)$, set $x_R = x_M$, otherwise $x_L = x_M$.
  - If $x_R - x_L$ is small enough, stop.
- Set $x_{L_1} = \gamma_2 x_L + \gamma_1 x_R$ and $x_{R_1} = \gamma_1 x_L + \gamma_2 x_R$. (One of these will be identical with the last x_M.)
- Loop No. 2 (univariant behaviour):
  - If $h(x_{L_1}) < h(x_{R_1})$, set $x_R = x_{R_1}$, then $x_{R_1} = x_{L_1}$, then $x_{L_1} = \gamma_2 x_L + \gamma_1 x_R$. Otherwise set $x_L = x_{L_1}$, then $x_{L_1} = x_{R_1}$, then $x_{R_1} = \gamma_1 x_L + \gamma_2 x_R$.
  - If $x_R - x_L$ is small enough, stop.

The following figures visualize two typical situations in the Bratu test problem we introduced in the previous papers: In all the figures, the straight lines represent $\tilde{h}_*$ and the dashed lines represent $\tilde{h}_1$, whereas the neighboring dotted lines represent the simplified functions h_* and h_1, respectively. The vertical dotted line at their point of intersection marks x_+. While the left hand figures literally show the graphs of the h functions, the right hand figures use somewhat more suggestive scales: Their horizontal axes show the value of κ_2, and their vertical axes show the values of the efficient reduction factors Q. In the first two figures, $x = x_+$ is the best choice, in the last two figures, $x = 1$ is the best choice. Both pairs of figures show that the simplified model does a good job at describing the behaviour in the vicinity of $x = x_+$, and both show that h_* behaves rather nicely.

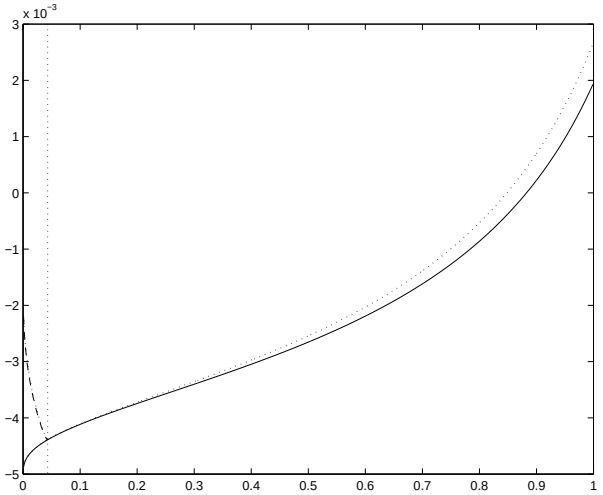

Figure 1: first example, h vs. x

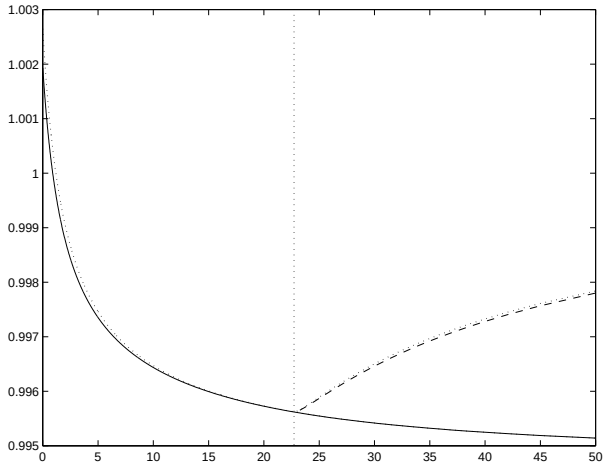

Figure 2: first example, Q vs. κ_2

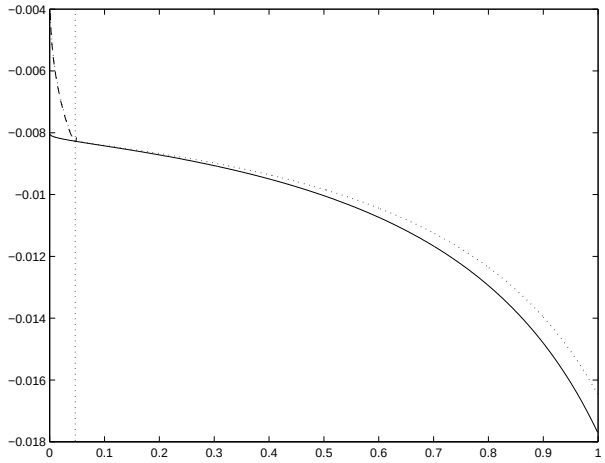

Figure 3: second example, h vs. x

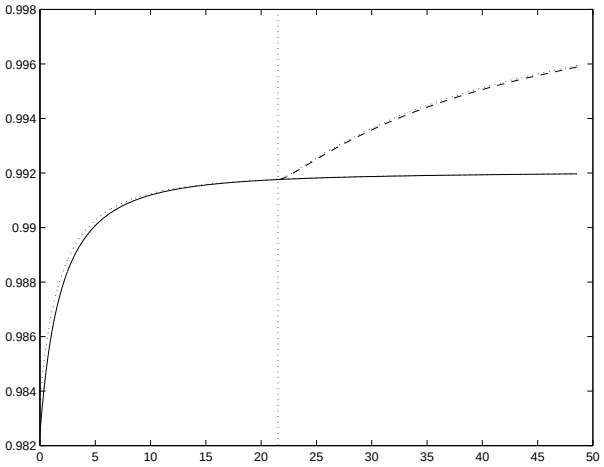

Figure 4: second example, Q vs. κ_2

References

[1] Gill, Philip E., Murray, Walter, and Wright, Margaret H.: *Practical Optimization*, Academic Press: London, San Diego (1981).

[2] Mackens, Wolfgang, Menck, Jürgen, and Voß, Heinrich: Coupling Iterative Subsystem Solvers, in *Scientific Computing in Chemical Engineering II – Simulation. Image Processing, Optimization, and Control*, Keil, F., Mackens, W., Voß, H., and Werther, J. (eds.) Springer–Verlag: Berlin, Heidelberg (1999), 183 – 191.

[3] Menck, Jürgen: *An Approximate Newton–Like Coupling of Subsystems*, Report 21, Arbeitsbereich Mathematik, Technische Universität Hamburg–Harburg (1998), to appear in ZAMM.

[4] Menck, Jürgen: Work Control for Newton Type Coupling, in *Scientific Computing in Chemical Engineering II – Simulation. Image Processing, Optimization, and Control*, Keil, F., Mackens, W., Voß, H., and Werther, J. (eds.) Springer–Verlag: Berlin, Heidelberg (1999), 192 – 199.