



9th September 2008

Multifrequential Harmonic Balance Technique

Philipp Rauschenberger

Projet de fin d'études

Tutor at CERFACS : Guillaume Puigt
Tutor at ISAE SUPAERO : Jérémie Gressier
Tutor at Universität Stuttgart : Ulrich Rist

CERFACS - CFD - 42, avenue Gaspard Coriolis - 31057 Toulouse Cedex 1
Référence CERFACS : WN/CFD/08/83

Abstract

Current unsteady methods are very time consuming and remain difficult to use in industry. The Harmonic Balance Technique is a method that benefits from taking into account the periodicity of periodic unsteady flow in order to reduce computational efforts considerably.

Based upon the work that has been done to implement the Harmonic Balance Technique in the *elsA* code, this report treats the generalisations of this method to several frequencies in a flow (multifrequential approach). On the one hand, it will be possible to take into account several frequencies that are not multiples of each other, on the other hand non-uniformly distributed instants may be imposed on the problem.

This report presents the realised literature research and the recapitulation of the theory of the Harmonic Balance Technique for one frequency and its harmonics, the theory of the multifrequential approach and a short comparison of the two approaches. Some remarks about oscillations in the neighbourhood of shocks are made. It also covers the analytic work that has been done on the determination of the instants within the considered period. The implementation in *elsA* as well as the code validation will be discussed. Attention is also turned towards the effect of non-uniformly distributed instants on the results of unsteady computation.

Acknowledgement

First and foremost, I thank my tutor at CERFACS, Guillaume Puigt, for the interesting topic he proposed. I would also like to express my gratefulness for the excellent mentoring and support over the whole period of the project. Furthermore, I would want to thank him for revising this report. I would like to express my gratitude towards everyone at CERFACS, who supported me during my final year project. In particular, Frédéric Sicot and Guillaume Dufour for giving explanations concerning the code of the monofrequential approach and Marc Montagnac for his help in various matters of the *elsA* code and revising this report.

I also thank Jérémie Gressier (ISAE SUPAERO) and Ulrich Rist (Universität Stuttgart) for their mentoring throughout this final year project.

Last but not least, I really liked working at CERFACS and enjoyed the very pleasant and friendly working atmosphere in the CFD team.

Contents

Nomenclature	vii
1 Introduction	1
2 Development of the methods	3
2.1 Monofrequential Harmonic Balance Technique	3
2.1.1 Decomposition in discrete Fourier series	3
2.1.2 Principle of the Monofrequential HBT	4
2.1.3 Starting point of the resolution	5
2.1.4 Implementation in the Navier-Stokes equations	5
2.1.5 Numeric treatment	8
2.1.6 Algorithm of <i>McMullen</i>	8
2.1.7 Source term	9
2.2 Multifrequential Harmonic Balance Technique	9
2.2.1 Transformation into the frequency domain	9
2.2.2 Principle of the Multifrequential HBT	10
2.2.3 Starting point of the resolution	11
2.2.4 Implementation in the Navier-Stokes equations	11

2.2.5	Numeric treatment	13
2.2.6	Source term	13
2.2.7	Real formulation	14
2.2.8	Summary	15
2.3	Differences between the Mono- and the Multifrequential HBT	16
3	Simulation of an unsteady shock with Multifrequential HBT	17
3.1	Gibbs' effect	17
3.2	Oscillations due to non-uniformly distributed instants	17
4	Methods to find restrictions for the choice of the instants	21
4.1	Analytic solutions	21
4.1.1	Choice of the instants based on the determinant of A_1^N	22
4.1.2	Choice of the instants based on a decomposition of A_1^N into a lower and upper triangular matrix	23
4.1.3	Choice of the instants based on the eigenvalues of A_1^N	23
4.1.4	Choice of the instants based on the singular values of A_1^N	24
4.1.5	Summary and proceeding	24
4.2	Numeric solutions	24
4.2.1	Choice of the instants in a way that the lines of A_1 are forming an orthogonal basis	24
4.2.2	Evolutionary determination	27
4.3	Determination of the last instant t_{2N}	28
5	Implementation	29
5.1	Interface	29

5.2	Main code	30
5.2.1	Computation of matrix D	31
5.2.2	Computation of the source term	32
5.2.3	Stability	33
5.2.4	Implicit method	33
6	Validation	35
6.1	Forced motion of a 2D airfoil	35
6.1.1	Pitching at low angle of attack in subsonic flow: CT2 (DI29)	36
6.1.2	Pitching at low angle of attack in transonic flow: CT6 (DI55)	40
6.1.3	Pitching at high angle of attack: Shock Stall case (DI89)	43
6.2	Forced motion of a 3D airfoil	46
6.2.1	Pitching at low angle of attack: CT5	47
6.2.2	Pitching at high angle of attack: CT9	55
6.3	Precision discrepancy of D-coefficients	62
6.4	Summary	63
7	Non-uniformly distributed instants	65
7.1	2D NACA airfoil	67
7.1.1	Test case CT2	67
7.1.2	Test case CT6	70
7.1.3	Test case Shock Stall	70
7.1.4	Summary	71
7.2	3D LANN wing	72
7.2.1	Test case CT5	73

7.2.2 Test case CT9	84
7.3 Summary	94
8 Conclusion and perspectives	95
Bibliography	97
Appendix: Structograms	99

Nomenclature

Latin letters

A_1	[-]	Discrete transformation matrix
A	[-]	Matrix with A_1 on the diagonal
A_k, A'_k	[-]	Coefficients of transformation (in real formulation)
B_k, B'_k	[-]	Coefficients of transformation (in real formulation)
B_1	[-]	Discrete Fourier transformation (DFT) matrix
B	[-]	Matrix with B_1 on the diagonal
c	[m]	Chord length
c_0	[m/s]	Velocity of sound
C_p	[-]	Pressure coefficient
D	[-]	Coefficient matrix D
E	[J]	Energy
E_k	[-]	Coefficient of transformation of ρE
f	[Hz]	Frequency
F	[-]	Vector of discrete conservative variables
i	[-]	Imaginary unit ($i^2 = -1$)
I	[-]	Identity matrix
k_p	[rad/s]	$k_p = \omega_{max} = 2\pi f_{max}$
L	[-]	Lower triangular matrix
M	[-]	Mach number
N	[-]	Order of the HBT computation/number of frequencies
p	[Pa]	Pressure
Q	[-]	Discrete residual vector (convection, diffusion and source terms)
R	[-]	Residual vector (convection, diffusion and source terms)
Re	[-]	Reynolds number
S	[-]	Source term matrix
SD	[-]	Sub-determinant
t	[s]	Time

Latin letters

T	[s]	Period
u	[m/s]	Velocity in x-direction
U_k	[-]	Coefficient of transformation of ρu
U	[-]	Upper triangular matrix
v	[m/s]	Velocity in y-direction
V	[m^3]	Volume
V_k	[-]	Coefficient of transformation of ρv
$\mathcal{V}(\Omega)$	[m]	Characteristic dimension of a cell
w	[m/s]	Velocity in z-direction
W	[-]	Vector of conservative variables
W_k	[-]	Coefficient of transformation of ρw

Greek letters

α_0	[$^\circ$]	Mean angle of attack
α_m	[$^\circ$]	Amplitude of the angle of attack
$\kappa(D)$	[-]	Condition number of coefficient matrix D
λ	[-]	Eigenvalue
$\tilde{\nu}, \nu$	[m^2/s]	Kinematic viscosity of the turbulence model
ν_k	[-]	Coefficient of transformation of $\rho \tilde{\nu}$
ρ	[m^3/s]	Density
$\rho(A)$	[m/s]	Spectral radius of the Jacobian (here: $\rho(A) = \ u\ + c_0$)
σ	[-]	Singular value
τ	[s]	Time step
Ψ_k	[-]	Coefficient of transformation of ρ
ω	[rad/s]	Angular frequency
ω_{2N+1}	[-]	$\omega_{2N+1} = e^{-2i\pi\frac{1}{2N+1}}$

Chapter 1

Introduction

Even if steady Reynolds Averaged Navier-Stokes (RANS) simulations are done classically during the process of design of planes or aero-engines, design needs more and more unsteady simulations to capture unsteady effects. This is done by the resolution of unsteady RANS equations (URANS). To do so, there are two ways to proceed. In the first way, temporal coherence must be assured among all grid cells, leading to the choice of the minimum local time as the time step and to an explicit time integration algorithm. In this case, a large number of steps is needed to describe unsteady effects during one period. Another way to proceed consists in choosing the time step of the computation. In this case, temporal coherence is lost and a Dual Time Stepping algorithm (DTS) is used to converge inner loops with an implicit time integration. Therefore, each inner loop converges to a solution which can be seen as a snapshot of the flow at the corresponding time.

In an industrial context, the number of URANS computations during the design process is increasing every day and most of the considered URANS simulations are periodic in time. As an example, pitching airfoil and turbo machinery are periodic. Actually, the URANS technique does not account for periodicity and this leads to a loss in CPU time, since a transient regime must be by-passed before the flow is periodic. Moreover, the length of this transition is unknown preliminarily and depends on the case at hand, but it is equivalent to several periods of computation (typically four or more). Actually, no information about the periodic nature of the flow is used and most effort is needed only to reach the periodic state and not to understand the flow's physics. This is related to the parabolic nature in time of the URANS equations that implies that the present instant only influences the future instants. However, in a periodic flow, the present instant influences the past, because the future will replicate a scheme of the past. In summary, the poor adaption of URANS towards periodic flow simulation leads to a waste of computing time.

Here, a novel approach for the computation of periodic unsteady flows is considered. The *Harmonic Balance Method* has been developed primarily at Duke University [10, 15, 17] and at Stanford University under the name of *Time Spectral Method* [8, 9, 18]. The principle of this approach is to assume that the flow is periodic and the conservative variables can be represented by Fourier coefficients. Introducing Fourier expressions in the Navier-Stokes equations and some algebraic simplifications, one transforms the periodic URANS computation on several steady coupled simulations. This approach has been implemented in the *elsA* code (cf. Chap. 5) and promises higher

efficiency than classic methods (faster than URANS and better capture of physics than linearization methods). The main interest of the technique is its compatibility with all techniques for acceleration in steady flows convergence such as local time stepping and multigrid algorithms. Moreover, due to the intrinsic periodicity, the transition phase of convergence of the steady computations is attaining the periodic state much faster.

The method in its present form disposes of two major disadvantages. Firstly, only one frequency and its harmonics (multiples of the base frequency) can be taken in consideration. Secondly, the instants, when the steady problems are solved, are always uniformly distributed over the greatest period (i.e. the period of the base frequency). However, multifrequential flows are encountered in current multi-stage turbo machinery applications as well as in classical aerodynamic simulations around planes (e.g. pitching airfoil, ...). The Multifrequential Harmonic Balance Technique is a very recent approach (cf. *Ekici and Hall* [5, 6]). Several modifications in the theoretical and numerical approach and of the present implementation in the *elsA* code will have to be made, but generally speaking an approach similar to the monofrequential technique is intended. Particularly, some work concerning the chosen instants will have to be effected. The instants might not necessarily be uniformly distributed over the time period that is observed.

This report is structured into a theoretical part, treating the mono- and multifrequential approaches and the differences between them. The transformation into the frequency domain gives rise to the Gibbs effect and non-uniformly distributed instants can also lead to oscillations. This problem is treated in a separate chapter. One chapter is dedicated on how to properly choose the instants in advance. This is necessary due to the need to invert the matrix that transforms the Navier-Stokes equations into the frequency domain. The report is furthermore structured into the following chapters: Implementation in the *elsA* code; validation of the modifications; and the impact of non-uniformly distributed instants on the solution, if only one base frequency and its harmonics are considered.

Chapter 2

Development of the methods

2.1 Monofrequential Harmonic Balance Technique

The **H**armonic **B**alance **T**echnique (HBT) is based on supposing the unsteady periodic variations of angular frequency ω of the conservative fields over time being representable by discrete Fourier series at the order of N (cf. Hall [10]). The following section presents a short recall of the Fourier transformation. Succeeding sections demonstrate the principle of the HBT and the implementation in the Navier-Stokes equations.

2.1.1 Decomposition in discrete Fourier series

Let g be a function of period T . The discrete Fourier series of g at the order of N is a function of complex value (in \mathbb{C}):

$$g_N(t) = \sum_{n=-N}^{n=N} c_n \exp\left(2i\pi n \frac{t}{T}\right),$$

where

$$c_n = \frac{1}{2N+1} \sum_{k=0}^{k=2N} g_k \exp\left(-2i\pi \frac{nk}{2N+1}\right) \quad \text{with} \quad g_k = g\left(\frac{kT}{2N+1}\right).$$

Two important properties of the Fourier series are pointed out which shall be referred to below. First of all, one has to pay attention to the discontinuity points of g . Although the Fourier series of a function g tends to g for an infinite number of points, at the discontinuity points we find (cf. [7]):

$$\lim_{N \rightarrow \infty} g_N(x) = \frac{g(x^+) + g(x^-)}{2}.$$

Secondly, consider a given equation $g = 0$. The decomposition in Fourier series yields (\hat{g}_k are the Fourier coefficients):

$$g = \sum_{k=-N}^N \hat{g}_k \exp(i\omega kt) = 0.$$

The orthogonality of the complex exponential functions implies, that the individual contribution of each mode is zero: $\hat{g}_k = 0$ for $-N \leq k \leq N$. This is proved by multiplying the preceding equation by $\exp(i\omega lt)$ and integrating over one period $T = \frac{2\pi}{\omega}$.

$$\sum_{k=-N}^N \hat{g}_k \int_0^T \exp(i\omega kt) \exp(i\omega lt) dt = 0.$$

If $k = -l$:

$$\int_0^T \exp(i\omega kt) \exp(-i\omega kt) dt = T,$$

if $k \neq -l$:

$$\begin{aligned} \int_0^T \exp(i\omega kt) \exp(i\omega lt) dt &= \int_0^T \exp(i\omega(k+l)t) dt \\ &= \frac{1}{i\omega(k+l)} [\exp(i\omega(k+l)t)]_0^T \\ &= \frac{1}{i\omega(k+l)} (\exp(i\omega(k+l)T) - 1) \\ &= \frac{1}{i\omega(k+l)} (\exp(2i\pi(k+l)) - 1) \\ &= 0. \end{aligned}$$

2.1.2 Principle of the Monofrequential HBT

To begin with, one can write the Navier-Stokes equations in the following semi-discrete form:

$$V \frac{\partial W}{\partial t} + R(W) = 0, \quad (2.1)$$

- V : Control volume,
- W : Vector of conservative variables,
- $R(W)$: Residual (convection, diffusion and source terms).

Under the assumption of temporal periodicity with an angular frequency ω , the conservative variables are developed in Fourier series (with the Fourier coefficients \hat{W}_k and \hat{R}_k):

$$W_{eq} = \sum_{k=-N}^N \hat{W}_{eq,k} \exp(i\omega kt), \quad (2.2)$$

$$R(W)_{eq} = \sum_{k=-N}^N \hat{R}_{eq,k} \exp(i\omega kt). \quad (2.3)$$

Thus, the Navier-Stokes equations can be noted in the following manner (continuity, momentum and energy equation, as well as the equations of the turbulence model respectively: $1 \leq eq \leq$ number of equations):

$$\sum_{k=-N}^N (i\omega V \hat{W}_{eq,k} + \hat{R}_{eq,k}) \exp(i\omega kt) = 0. \quad (2.4)$$

2.1.3 Starting point of the resolution

As it has been shown in section 2.1.1, the orthogonality of the complex exponential functions implies that the individual contributions of each mode are zero. Therefore $(2N + 1)$ equations are obtained linking the $(2N + 1)$ Fourier coefficients of W and $R(W)$:

$$ik\omega V \hat{W}_{eq,k} + \hat{R}_{eq,k} = 0. \quad (2.5)$$

2.1.4 Implementation in the Navier-Stokes equations

Noting down the conservative variables (with a one-equation turbulence model)

$$\begin{aligned} \rho &= \sum_k \psi_k(x, y, z) \exp(ik\omega t), \\ \rho u &= \sum_k U_k(x, y, z) \exp(ik\omega t), \\ \rho v &= \sum_k V_k(x, y, z) \exp(ik\omega t), \\ \rho w &= \sum_k W_k(x, y, z) \exp(ik\omega t), \\ \rho E &= \sum_k E_k(x, y, z) \exp(ik\omega t), \\ \rho \tilde{\nu} &= \sum_k \nu_k(x, y, z) \exp(ik\omega t), \end{aligned}$$

one may write the Navier-Stokes equations in matrix form:

$$\hat{R} + Vi\omega P_1 \hat{W} = 0, \quad (2.6)$$

with

$$\begin{aligned} \hat{W} &= [\psi_{-N}, \dots, \psi_N, U_{-N}, \dots, U_N, V_{-N}, \dots, V_N, \\ &\quad W_{-N}, \dots, W_N, E_{-N}, \dots, E_N, \nu_{-N}, \dots, \nu_N]^T, \\ \hat{R} &= [R_{1,-N}, \dots, R_{1,N}, R_{2,-N}, \dots, R_{2,N}, R_{3,-N}, \dots, R_{3,N}, \\ &\quad R_{4,-N}, \dots, R_{4,N}, R_{5,-N}, \dots, R_{5,N}, R_{6,-N}, \dots, R_{6,N}]^T, \\ P_1 &= \text{diag}(-N, \dots, N, -N, \dots, N, -N, \dots, N, -N, \dots, N, -N, \dots, N). \end{aligned}$$

The discrete Fourier Transformation operator (DFT) is introduced, where $\omega_{2N+1} = \exp(-2i\pi \frac{1}{2N+1})$:

$$B_1 = \begin{bmatrix} 1 & \omega_{2N+1}^{-N} & \omega_{2N+1}^{2(-N)} & \cdots & \omega_{2N+1}^{2N(-N)} \\ 1 & \omega_{2N+1}^{-N+1} & \omega_{2N+1}^{2(-N+1)} & \cdots & \omega_{2N+1}^{2N(-N+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{2N+1}^{-1} & \omega_{2N+1}^{-2} & \cdots & \omega_{2N+1}^{-2N} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_{2N+1}^1 & \omega_{2N+1}^2 & \cdots & \omega_{2N+1}^{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{2N+1}^N & \omega_{2N+1}^{2(N)} & \cdots & \omega_{2N+1}^{2N(N)} \end{bmatrix}. \quad (2.7)$$

By applying the following relations

$$\omega_{2N+1}^k = \omega_{2N+1}^{k+2N+1} \quad \text{since} \quad \omega_{2N+1}^{2N+1} = \exp\left(-2i\pi \frac{2N+1}{2N+1}\right) = 1$$

and by interchanging some lines of the matrix, matrix B_1 is now found in the following form:

$$B_1 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_{2N+1}^1 & \omega_{2N+1}^2 & \cdots & \omega_{2N+1}^{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{2N+1}^{2N} & \omega_{2N+1}^{2(2N)} & \cdots & \omega_{2N+1}^{2N(2N)} \end{bmatrix} \quad (2.8)$$

The interest of casting the matrix into this form is to make it easily invertible. The elements of the family ω_{2N+1}^i form an orthogonal basis. B_1 is an orthogonal matrix and the inverse of matrix B_1 can be easily determined:

$$B_1^{-1} = \frac{1}{2N+1} \bar{B}_1 = \frac{1}{2N+1} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_{2N+1}^{-1} & \omega_{2N+1}^{-2} & \cdots & \omega_{2N+1}^{-2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{2N+1}^{-2N} & \omega_{2N+1}^{-2(2N)} & \cdots & \omega_{2N+1}^{-2N(2N)} \end{bmatrix}, \quad (2.9)$$

where \bar{B}_1 is the complex conjugate transpose of B_1 . For the Fourier transformation, one needs the conservative variables at the $(2N+1)$ instants uniformly distributed over one period T and therefore, the conservative and turbulent fields are now noted at these discrete points (with a

one-equation turbulence model):

$$F = \begin{bmatrix} \rho^i & \text{if } 0 \leq i \leq (2N+1) - 1 \\ U^{i-(2N+1)} & \text{if } 2N+1 \leq i \leq 2(2N+1) - 1 \\ V^{i-2(2N+1)} & \text{if } 2(2N+1) \leq i \leq 3(2N+1) - 1 \\ W^{i-3(2N+1)} & \text{if } 3(2N+1) \leq i \leq 4(2N+1) - 1 \\ E^{i-4(2N+1)} & \text{if } 4(2N+1) \leq i \leq 5(2N+1) - 1 \\ \nu^{i-5(2N+1)} & \text{if } 5(2N+1) \leq i \leq 6(2N+1) - 1 \end{bmatrix},$$

$$Q = \begin{bmatrix} R^i & \text{if } 0 \leq i \leq (2N+1) - 1 \\ R^{i-(2N+1)} & \text{if } 2N+1 \leq i \leq 2(2N+1) - 1 \\ R^{i-2(2N+1)} & \text{if } 2(2N+1) \leq i \leq 3(2N+1) - 1 \\ R^{i-3(2N+1)} & \text{if } 3(2N+1) \leq i \leq 4(2N+1) - 1 \\ R^{i-4(2N+1)} & \text{if } 4(2N+1) \leq i \leq 5(2N+1) - 1 \\ R^{i-5(2N+1)} & \text{if } 5(2N+1) \leq i \leq 6(2N+1) - 1 \end{bmatrix},$$

$$B = \text{diag}(B_1, B_1, B_1, B_1, B_1, B_1)$$

and hence, \hat{W} and \hat{R} can be defined:

$$\hat{W} = \frac{1}{2N+1}BF,$$

$$\hat{R} = \frac{1}{2N+1}BQ.$$

The Navier-Stokes equations may be written then in the following matrix form:

$$\hat{R} + Vi\omega P \hat{W} = 0$$

or

$$\frac{1}{2N+1}BQ + \frac{1}{2N+1}Vi\omega PBF = 0. \quad (2.10)$$

$$\hat{W} = [\psi_0 \dots \psi_N, \psi_{-N} \dots \psi_{-1}, U_0 \dots U_N, U_{-N}, \dots, U_{-1}, V_0, \dots, V_N, V_{-N}, \dots, V_{-1}, W_0, \dots, W_N, W_{-N}, \dots, W_{-1}, E_0, \dots, E_N, E_{-N}, \dots, E_{-1}, \nu_0, \dots, \nu_N, \nu_{-N}, \dots, \nu_{-1}]^T,$$

$$\hat{R} = [R_{1,0}, \dots, R_{1,N}, R_{1,-N}, \dots, R_{1,-1}, R_{2,0}, \dots, R_{2,N}, R_{2,-N}, \dots, R_{2,-1}, R_{3,0}, \dots, R_{3,N}, R_{3,-N}, \dots, R_{3,-1}, R_{4,0}, \dots, R_{4,N}, R_{4,-N}, \dots, R_{4,-1}, R_{5,0}, \dots, R_{5,N}, R_{5,-N}, \dots, R_{5,-1}, R_{6,0}, \dots, R_{6,N}, R_{6,-N}, \dots, R_{6,-1}]^T,$$

$$P = \text{diag}(C, C, C, C, C, C) \quad \text{with} \quad C = \text{diag}(0, \dots, N, -N, \dots, -1).$$

It was proved that matrix B_1 is invertible and by applying the same reasoning, B is invertible, too:

$$B^{-1} = \text{diag}(B_1^{-1}, B_1^{-1}, B_1^{-1}, B_1^{-1}, B_1^{-1}, B_1^{-1}) = \frac{1}{2N+1} \text{diag}(\bar{B}_1, \bar{B}_1, \bar{B}_1, \bar{B}_1, \bar{B}_1, \bar{B}_1).$$

In this vein, the components of the Fourier transform are converted back to conservative variables (at the different instants):

$$F^* = (2N+1)B^{-1}\hat{W},$$

$$Q^* = (2N+1)B^{-1}\hat{R}.$$

Nevertheless, special attention should be paid to the points of discontinuity. The remark of 2.1.1 regarding the points of discontinuity is repeated here, detecting the following property:

$$\lim_{N \rightarrow \infty} g_N(x) = \frac{g(x^+) + g(x^-)}{2}.$$

However, the assumption $Q^* = Q$ can be made. This assumption is justified by the addition of artificial viscosity which smooths the shocks (cf. Fig. 2.1).

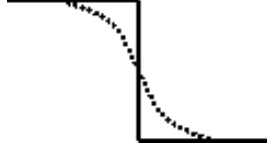


Figure 2.1: *Smoothing of shocks by added artificial viscosity.*

Transformed back into the time domain, the equation reads:

$$Q + i\omega B^{-1} PBF = 0. \quad (2.11)$$

2.1.5 Numeric treatment

The term $S = i\omega VB^{-1} PBF$, called source term, is a spectral discretisation of the time marching operator (coupling of all the terms of the discrete Fourier transformation). This term can be interpreted as a discretisation of the time derivative centered in time at $\frac{kT}{2N+1} \quad \forall \quad 0 \leq k \leq 2N$. Eq. (2.11) is a steady equation relying among each other the solutions taken at $(2N + 1)$ instants of the discretisation of the phenomenon's period. For numeric treatment, a virtual time τ (cf. *Hall et al.*), that allows to iterate in virtual time in order to obtain a steady solution for each physical instant, is added:

$$V \frac{\partial F}{\partial \tau} + Q + S = 0, \quad (2.12)$$

$$S = i\omega VB^{-1} PBF.$$

2.1.6 Algorithm of *McMullen*

On the other hand, the algorithm of *McMullen et al.* [11] of Stanford University consists in resolving multiple steady systems in the frequency domain:

$$V \frac{d \hat{W}_{eq,k}}{d\tau} + ikV\omega \hat{W}_{eq,k} + \hat{R}_{eq,k} = 0 \quad \text{with} \quad -N \leq k \leq N, \quad 1 \leq eq \leq \text{number of equations} \quad (2.13)$$

Since \hat{R}_k is not linear, it cannot be calculated directly from \hat{W}_k . This is why one has to go back into the time domain and calculate $R(W)$ there.

2.1.7 Source term

A purely real source term can be built by exploiting all the matrix products (cf. Jameson et al. [12]). Let F_j with $0 \leq j \leq 2N$ be the $(2N + 1)$ values of the conservative field F and be aware that the source term of time level l does not depend on the conservative field at the instant l :

$$S_l = -V\omega \sum_{j=1, j \neq l}^{2N+1} d_{jl} F_{j-1} \quad \text{with} \quad d_{jl} = \frac{(-1)^{j-l}}{2 \sin\left(\frac{\pi(j-l)}{2N+1}\right)}. \quad (2.14)$$

2.2 Multifrequential Harmonic Balance Technique

When problems are tackled, where several frequencies appear that are not multiples of a base frequency or if instants must be chosen that are not uniformly distributed over the considered period, a different approach is necessary. *Ekici and Hall* [5, 6] have proposed a method that consists in resolving the Navier-Stokes equations using at about 50% more instants $(3N + 1)$, which leads to a DFT operator that is not square anymore and for which a pseudoinverse must be found. A different approach is proposed here, which is indeed similar to the monofrequential one, already implemented in *elsA*.

2.2.1 Transformation into the frequency domain

Again, a function g is considered. The discrete transformation of g into the frequency domain is a function of complex values in \mathbb{C} :

$$g_N(t) = \sum_{n=-N}^{n=N} \hat{g}_n \exp(i\omega_n t) \quad \text{with} \quad \omega_n = 2\pi f_n.$$

For the monofrequential approach, the following convergence property is found at the points of discontinuity of g (cf. section 2.1.1):

$$\lim_{N \rightarrow \infty} g_N(x) = \frac{g(x^+) + g(x^-)}{2} \quad (\text{monofrequential}).$$

This property is not valid anymore, if several frequencies appear that are not multiples of the smallest frequency, due to the fact that the derivation of this property is based upon the assumption of one frequency and its harmonics (cf. [7]) and cannot be transposed on the multifrequential approach.

Another difference appears when accounting for an equation $g = 0$. The transformation into the frequency domain yields (\hat{g}_k are the coefficients of transformation for an angular frequency ω_k):

$$\sum_{k=-N}^N \hat{g}_k \exp(i\omega_k t) = 0.$$

By multiplying the preceding equation by $\exp(i\omega_l t)$ and by integrating over one period $T = \frac{2\pi}{\omega}$, the previous equation reads:

$$\sum_{k=-N}^N \hat{g}_k \int_0^T \exp(i\omega_k t) \exp(i\omega_l t) dt = 0.$$

If $k = -l$:

$$\int_0^T \exp(i\omega_k t) \cdot \exp(i\omega_{-k} t) dt = \int_0^T \exp(i\omega_k t) \cdot \exp(-i\omega_k t) dt = T,$$

if $k \neq -l$:

$$\begin{aligned} \int_0^T \exp(i\omega_k t) \exp(i\omega_l t) dt &= \int_0^T \exp(i(\omega_k + \omega_l)t) dt \\ &= \frac{1}{i(\omega_k + \omega_l)} (\exp(i(\omega_k + \omega_l)T) - 1) \\ &\neq 0 \quad \text{in general.} \end{aligned}$$

It is not possible to state the individual contributions of each mode to be equal to zero anymore.

2.2.2 Principle of the Multifrequential HBT

Again, the origin is the Navier-Stokes equations in semi-discrete form, Eq. (2.1):

$$V \frac{\partial W}{\partial t} + R(W) = 0. \quad (2.1)$$

The N frequencies are sorted in ascending order. When carrying through the transformation into the frequency domain, the negative values of the N frequencies and $\omega_0 = 0$ appear in addition. It holds $\omega_{-k} = -\omega_k$ and basically it must be assumed that ω_k is not necessarily equal to $k\omega_1$. That way, the conservative variables may be developed with a discrete transformation (with \hat{W}_k and \hat{R}_k being the coefficients of transformation for a frequency ω_k):

$$W_{eq} = \sum_{k=-N}^N \hat{W}_{eq,k} \exp(i\omega_k t), \quad (2.15)$$

$$R(W)_{eq} = \sum_{k=-N}^N \hat{R}_{eq,k} \exp(i\omega_k t). \quad (2.16)$$

The Navier-Stokes equations may be written in the following way then (for each of the following equations: continuity, momentum and energy equations, as well as the equations of the turbulence model: $1 \leq eq \leq$ number of equations):

$$\sum_{k=-N}^N (i\omega_k V \hat{W}_{eq,k} + \hat{R}_{eq,k}) \exp(i\omega_k t) = 0. \quad (2.17)$$

2.2.3 Starting point of the resolution

As it was shown in section 2.2.1, $ik\omega V \hat{W}_k + \hat{R}_k = 0$ cannot be asserted anymore. The approach described below consists nevertheless in pursuing the exact same procedure as for the monofrequency approach. In order to make sure that the individual contribution of each angular frequency ω_k equals zero, the complex exponential functions $\exp(i\omega_k t_n)$ must form a basis. Since the angular frequencies ω_k are prescribed by the examined flow and are therefore constants, the idea is to choose the instants preliminarily so, that the complex exponential functions form a base. Then, the instants are not necessarily uniformly distributed over a period T anymore. The effort to find such a distribution of the instants is treated in section 4. Matrix A_1 is defined, in order to be able to describe the Navier-Stokes equations in the matrix form $A_1 \cdot X = 0$:

$$A_1 = \begin{bmatrix} \exp(i\omega_{-N} t_0) & \dots & \exp(i\omega_0 t_0) & \dots & \exp(i\omega_N t_0) \\ \vdots & & \vdots & & \vdots \\ \exp(i\omega_{-N} t_k) & \dots & \exp(i\omega_0 t_k) & \dots & \exp(i\omega_N t_k) \\ \vdots & & \vdots & & \vdots \\ \exp(i\omega_{-N} t_{2N}) & \dots & \exp(i\omega_0 t_{2N}) & \dots & \exp(i\omega_N t_{2N}) \end{bmatrix}. \quad (2.18)$$

Be aware of the fact that matrix A_1 is not equal to matrix B_1 and is rather the pendant to B_1^{-1} . This proceeding results in $A_1 \cdot X = A_1 \cdot [\hat{R}_{eq} + ViP_1 \hat{W}_{eq}] = 0$ (the definitions of the vectors are given below). If the complex exponential functions are forming a basis, A_1 is invertible. From now on, it is supposed that the instants are properly chosen and therefore A_1 is invertible. Then, Eq. (2.1) reads:

$$i\omega_k V \hat{W}_{eq,k} + \hat{R}_{eq,k} = 0. \quad (2.19)$$

2.2.4 Implementation in the Navier-Stokes equations

By defining again (with a one-equation turbulence model)

$$\begin{aligned} \rho &= \sum_k \psi_k(x, y, z) \exp(i\omega_k t), \\ \rho u &= \sum_k U_k(x, y, z) \exp(i\omega_k t), \\ \rho v &= \sum_k V_k(x, y, z) \exp(i\omega_k t), \\ \rho w &= \sum_k W_k(x, y, z) \exp(i\omega_k t), \\ \rho E &= \sum_k E_k(x, y, z) \exp(i\omega_k t), \\ \rho \tilde{\nu} &= \sum_k \nu_k(x, y, z) \exp(i\omega_k t), \end{aligned}$$

the Navier-Stokes equations in matrix form read:

$$\hat{R} + ViP \hat{W} = 0, \quad (2.20)$$

with

$$\begin{aligned}
\hat{W} &= [\psi_{-N}, \dots, \psi_N, U_{-N}, \dots, U_N, V_{-N}, \dots, V_N, W_{-N}, \dots, W_N, E_{-N}, \dots, E_N, \nu_{-N}, \dots, \nu_N]^T, \\
\hat{R} &= [R_{1,-N}, \dots, R_{1,N}, R_{2,-N}, \dots, R_{2,N}, R_{3,-N}, \dots, R_{3,N}, \\
&\quad R_{4,-N}, \dots, R_{4,N}, R_{5,-N}, \dots, R_{5,N}, R_{6,-N}, \dots, R_{6,N}]^T, \\
A &= \text{diag}(A_1, A_1, A_1, A_1, A_1, A_1), \\
P &= \text{diag}(P_1, P_1, P_1, P_1, P_1, P_1) \quad \text{with} \quad P_1 = \text{diag}(\omega_{-N}, \dots, \omega_N)
\end{aligned}$$

Similarly to the monofrequential approach, one needs the conservative and turbulent fields at the discrete points i to effect the transformation into the frequency domain.

$$\begin{aligned}
F &= \begin{bmatrix} \rho^i & \text{if} & 0 & \leq i \leq & (2N+1) - 1 \\ U^{i-(2N+1)} & \text{if} & 2N+1 & \leq i \leq & 2(2N+1) - 1 \\ V^{i-2(2N+1)} & \text{if} & 2(2N+1) & \leq i \leq & 3(2N+1) - 1 \\ W^{i-3(2N+1)} & \text{if} & 3(2N+1) & \leq i \leq & 4(2N+1) - 1 \\ E^{i-4(2N+1)} & \text{if} & 4(2N+1) & \leq i \leq & 5(2N+1) - 1 \\ \nu^{i-5(2N+1)} & \text{if} & 5(2N+1) & \leq i \leq & 6(2N+1) - 1 \end{bmatrix}, \\
Q &= \begin{bmatrix} R^i & \text{if} & 0 & \leq i \leq & (2N+1) - 1 \\ R^{i-(2N+1)} & \text{if} & 2N+1 & \leq i \leq & 2(2N+1) - 1 \\ R^{i-2(2N+1)} & \text{if} & 2(2N+1) & \leq i \leq & 3(2N+1) - 1 \\ R^{i-3(2N+1)} & \text{if} & 3(2N+1) & \leq i \leq & 4(2N+1) - 1 \\ R^{i-4(2N+1)} & \text{if} & 4(2N+1) & \leq i \leq & 5(2N+1) - 1 \\ R^{i-5(2N+1)} & \text{if} & 5(2N+1) & \leq i \leq & 6(2N+1) - 1 \end{bmatrix}.
\end{aligned}$$

The coefficients of transformation can be calculated in matrix form with the help of matrix A_1 . As a matter of fact, it is the inverse of A_1 that needs to be calculated. This can easily be shown by examining the arbitrary functions $g_i(t)$ unified in a vector G :

$$\begin{aligned}
f_i(t) &= \sum_{k=-N}^{k=N} \hat{f}_k \exp(i\omega_k t), \\
G &= A \hat{G}, \\
A^{-1}G &= \underbrace{A^{-1}A}_{=I} \hat{G} = \hat{G}.
\end{aligned}$$

Note the coherence among A_1^{-1} and A^{-1} :

$$A^{-1} = \text{diag}(A_1^{-1}, A_1^{-1}, A_1^{-1}, A_1^{-1}, A_1^{-1}, A_1^{-1}).$$

This obviously results immediately in:

$$\begin{aligned}
\hat{W} &= A^{-1}F, \\
\hat{R} &= A^{-1}Q.
\end{aligned}$$

The Navier-Stokes equations are noted in matrix form in the following way:

$$\hat{R} + ViP \hat{W} = 0$$

or

$$A^{-1}R + \frac{1}{2N+1} ViPA^{-1}F = 0.$$

(2.21)

In order to transform back to the time domain, Eq. (2.21) must be multiplied from the left side by A . That way, the coefficients of transformation are transformed into the conservative variables (at each distinct instant):

$$\begin{aligned} F^* &= A \hat{W} = AA^{-1}F, \\ R^* &= A \hat{R} = AA^{-1}Q. \end{aligned}$$

Taking into account the remark concerning the convergence at the points of discontinuity in section 2.2.1, the following assumption is made: $Q^* = Q$ similarly to the monofrequential case, which is justified by the added artificial viscosity, that smooths the shocks. The transformed equation in the time domain presents itself therefore in the following form:

$$Q + ViAPA^{-1}F = 0. \quad (2.22)$$

2.2.5 Numeric treatment

A new source term $S = ViAPA^{-1}F$ appears. It couples all the terms of the discrete transformation. Eq. (2.22) is a steady equation, connecting among themselves all the solutions taken at $(2N + 1)$ instants of discretisation over the considered period. A virtual time step is added for the numeric treatment. That way the solution can be iterated until a steady solution of the physical instant is obtained:

$$\begin{aligned} V \frac{\partial F}{\partial \tau} + Q + S &= 0, \\ S &= ViAPA^{-1}F. \end{aligned} \quad (2.23)$$

2.2.6 Source term

The source term of the multifrequential approach cannot be simplified like the one of the monofrequential approach, because the frequencies are not necessarily multiples of a base frequency and the instants are not uniformly distributed in general. Hence, the arising matrix products must be evaluated in the *elsA* code in the following formulation:

$$\begin{aligned} S &= VDF, \\ D &= iAPA^{-1}. \end{aligned} \quad (2.24)$$

Admittedly, the following minor problem will be encountered: D is not necessarily real which cannot be tolerated since the initial values are real. The way out of this problem, chosen for this approach, is to simply pass to a real formulation.

2.2.7 Real formulation

Eq. (2.15) and (2.16) are noted then ($\omega_0 = 0$):

$$W_{eq} = \sum_{k=-N}^N \hat{W}_{eq,k} \exp(i\omega_k t) = A_{eq,0} + \sum_{k=1}^N (A_{eq,k} \cos(\omega_k t) + B_{eq,k} \sin(\omega_k t)), \quad (2.25)$$

$$R(W)_{eq} = \sum_{k=-N}^N \hat{R}_{eq,k} \exp(i\omega_k t) = A'_{eq,0} + \sum_{k=1}^N (A'_{eq,k} \cos(\omega_k t) + B'_{eq,k} \sin(\omega_k t)). \quad (2.26)$$

Matrix A_1 becomes:

$$A_1 = \begin{bmatrix} 1 & \cos(\omega_1 t_0) & \dots & \cos(\omega_N t_0) & \sin(\omega_1 t_0) & \dots & \sin(\omega_N t_0) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(\omega_1 t_k) & \dots & \cos(\omega_N t_k) & \sin(\omega_1 t_k) & \dots & \sin(\omega_N t_k) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(\omega_1 t_{2N}) & \dots & \cos(\omega_N t_{2N}) & \sin(\omega_1 t_{2N}) & \dots & \sin(\omega_N t_{2N}) \end{bmatrix}. \quad (2.27)$$

It is useful for the following sections, particularly section 4, to mention here, that matrix A_1 in real formulation can be obtained by elementary transformations of the matrix in complex formulation. Actually, this is only about the addition of two corresponding columns (i.e. two columns of the same frequency but different algebraic sign of the exponent) and the division by 2 or $-2i$ (complex number: $i = \sqrt{-1}$) respectively.

The coefficients of transformation are:

$$\hat{W}_{eq} = \begin{bmatrix} A_{eq,0} \\ A_{eq,1} \\ \vdots \\ A_{eq,N} \\ B_{eq,1} \\ \vdots \\ B_{eq,N} \end{bmatrix}, \quad \hat{R}_{eq} = \begin{bmatrix} A'_{eq,0} \\ A'_{eq,1} \\ \vdots \\ A'_{eq,N} \\ B'_{eq,1} \\ \vdots \\ B'_{eq,N} \end{bmatrix}. \quad (2.28)$$

The term $\frac{\partial W}{\partial t}$ changes slightly:

$$\frac{\partial W_{eq}}{\partial t} = \frac{\partial A_1}{\partial t} \hat{W}_{eq}. \quad (2.29)$$

Matrix $\frac{\partial A_1}{\partial t}$ is obtained by temporal derivation of matrix A_1 ($\omega_0 = 0$):

$$\frac{\partial A_1}{\partial t} = \begin{bmatrix} 0 & -\omega_1 \sin(\omega_1 t_0) & \dots & -\omega_N \sin(\omega_N t_0) & \omega_1 \cos(\omega_1 t_0) & \dots & \omega_N \cos(\omega_N t_0) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & -\omega_1 \sin(\omega_1 t_k) & \dots & -\omega_N \sin(\omega_N t_k) & \omega_1 \cos(\omega_1 t_k) & \dots & \omega_N \cos(\omega_N t_k) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & -\omega_1 \sin(\omega_1 t_{2N}) & \dots & -\omega_N \sin(\omega_N t_{2N}) & \omega_1 \cos(\omega_1 t_{2N}) & \dots & \omega_N \cos(\omega_N t_{2N}) \end{bmatrix} \quad (2.30)$$

Eq. (2.11) finally reads:

$$Q + V \frac{\partial A}{\partial t} A^{-1} F = 0 \quad (2.31)$$

with

$$\frac{\partial A}{\partial t} = \text{diag} \left(\frac{\partial A_1}{\partial t}, \frac{\partial A_1}{\partial t}, \frac{\partial A_1}{\partial t}, \frac{\partial A_1}{\partial t}, \frac{\partial A_1}{\partial t}, \frac{\partial A_1}{\partial t} \right).$$

Numeric treatment in real formulation

The numeric treatment remains essentially the same:

$$V \frac{\partial F}{\partial \tau} + Q + S = 0, \tag{2.32}$$
$$S = V \frac{\partial A}{\partial t} A^{-1} F.$$

Source term in real formulation

The source term in real formulation is missing one of the matrix products and is noted therefore in the following way:

$$S = VDF \tag{2.33}$$
$$D = \frac{\partial A}{\partial t} A^{-1}$$

2.2.8 Summary

Thus, it is possible to define a scheme similar to the monofrequential approach. The essential point is to define in advance the instants distributed over a period T in such a way that matrix A is invertible. It would be particularly desirable to find analytic formulations that allow to delimit the intervals in which the instants must be chosen with the angular frequencies being known constants.

2.3 Differences between the Mono- and the Multifrequential HBT

The following table (Tab. 2.1) displays the most important differences between the Harmonic Balance Technique for one single frequency and its harmonics (monofrequential) and the HBT for several frequencies and/or non-uniformly distributed instants (multifrequential).

Table 2.1: *Differences between the monofrequential and the multifrequential approach.*

monofrequential	multifrequential
<p>one frequency f and its harmonics: $f_k = kf$</p> <p style="text-align: center;">$0, 1 \cdot f, 2 \cdot f, \dots, N \cdot f$</p>	<p>N frequencies: f_k is not necessarily kf</p> <p style="text-align: center;">$0, f_1, f_2, \dots, f_N$</p>
in advance B_1 is invertible	in advance A_1 is not necessarily invertible
the instants t_n are uniformly distributed over the period T	the instants t_n are not necessarily uniformly distributed over the period T
<p>the source term $S = i\omega VB^{-1}PBF$</p> <p>can be simplified into a purely real form:</p> $S_l = -V\omega \sum_{j=1, j \neq l}^{2N+1} d_{j,l} F_{j-1}$ $d_{j,l} = \frac{(-1)^{j-l}}{2\sin\left(\frac{\pi(j-l)}{2N+1}\right)}$	<p>the source term $S = VDF$</p> <p>$D = iAPA^{-1}$ (complex formulation)</p> <p>$D = \frac{\partial A}{\partial t} A^{-1}$ (real formulation)</p> <p>$d_{j,l}$ must be calculated numerically</p>

Chapter 3

Simulation of an unsteady shock with Multifrequential HBT

3.1 Gibbs' effect

The Gibbs' effect is related to a raise in order of the Fourier transformation. If a function g disposes of a discontinuity, the discrete Fourier transformation of g has the tendency to oscillate around this point of discontinuity. This leads to a bad resolution of the initial function in the neighbourhood of the discontinuity.

3.2 Oscillations due to non-uniformly distributed instants

Oscillations may also appear when non-uniformly distributed instants are used, particularly oscillations are found around shocks (cf. Chap. 7). This effect is related to bad capturing of shocks in unsteady flow. The shock's position is time dependent and the choice of the instants has therefore an influence on the correct prediction of the shock. When a uniform distribution is chosen, the whole physical time range is equally covered, while this statement cannot be made for a non-uniform distribution.

To clarify the above statements and to visualize the effects, a discontinuous function g of second order with a variable x is defined. The position of the discontinuity is time dependent:

$$g(x) = \begin{cases} \left(\frac{x}{\frac{1}{40}\sin(2\pi ft) + 0.5} \right)^2 & \text{for } 0 \leq x \leq x_s, \\ 2 \left(\frac{x-1}{\frac{1}{40}\sin(2\pi ft) - 0.5} \right)^2 & \text{for } x_s \leq x \leq 1, \end{cases}$$

with $x_s = \frac{1}{40}\sin(2\pi ft) + 0.5, \quad f = 1.$

The graph of $g(x)$ is displayed in Fig. 3.1 for three different instants: $t_0 = 0$, the maximum displacement of the discontinuity to the right at $t = 0.25\frac{1}{f}$ and to the left at $t = 0.75\frac{1}{f}$. Hence, the

discontinuity moves in the range $0.475 \leq x_s \leq 0.525$. The period of the unsteady function is $T = \frac{1}{f}$.

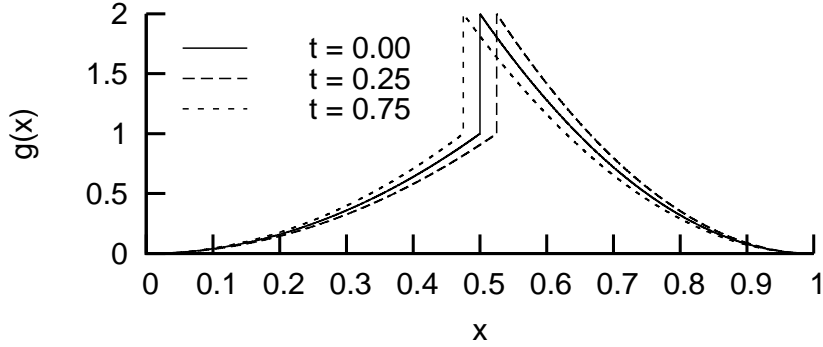


Figure 3.1: $g(x)$ at significant instants.

Several distributions of the instants have been tested with up to five harmonics (given all in Tab. 7.1). However, only some of them are discussed here. These are given in Tab. 3.1.

Table 3.1: *Distribution of the instants as ratio of period $T = \frac{1}{f}$ with $t_0 = 0$ for each order N and distribution i .*

N, i	$\frac{t_1}{T}$	$\frac{t_2}{T}$	$\frac{t_3}{T}$	$\frac{t_4}{T}$	$\frac{t_5}{T}$	$\frac{t_6}{T}$	$\frac{t_7}{T}$	$\frac{t_8}{T}$	$\frac{t_9}{T}$	$\frac{t_{10}}{T}$
3, 1	0.0416	0.0833	0.125	0.166	0.2083	0.25				
4, 0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8		
4, 3	0.05	0.2	0.25	0.4	0.5	0.72	0.8	0.95		
5, 3	0.1	0.2	0.225	0.25	0.64	0.68	0.7225	0.7575	0.79	0.9

Fig. 3.2 shows the mean, real and imaginary parts of the transformed function $g(x)$ for four frequencies and uniformly distributed instants. The oscillations at the shock can be seen clearly. Likewise, Fig. 3.3 shows the solution with distribution (4, 3) that is comparable to the one with the solution of a uniform distribution and does not necessarily show a worse result, because all parts of the transformed function must be taken in account in order to properly depict function $g(x)$. Furthermore, only the real and imaginary part of the first order are given. The real and imaginary part of higher orders are not displayed. However, distribution (4, 3) is quite close to a uniform one. An example on how the instants can affect the results in an extreme way can be seen in Fig. 3.4. The instants are distributed in steps of $\frac{1}{24}T$ over a quarter period T . The over prediction is of factor 400. It is not surprising that all computations effected with distribution (3, 1) did not pass due to errors, that indicate negative pressure or temperature values (cf. Chap. 7). In addition, the instants being distributed in $0 < t < 0.25\frac{1}{f}$ capture the shock's motion neither in $0.475 \leq x_s \leq 0.5$, nor do they see the backswing of the discontinuity from $x_s = 0.525$ to $x_s = 0.5$. Fig. 3.5 shows similar behaviour, although less distinct and is another example of bad results. This distribution disposes of no instants in $0.25 < t < 0.64$.

The conclusion to draw from this digression is that the instants, together with high orders, influence the results considerably and may not be easily comparable among each other. The experiences mentioned in this chapter should be kept in mind, when interpreting the results of Chap. 7.

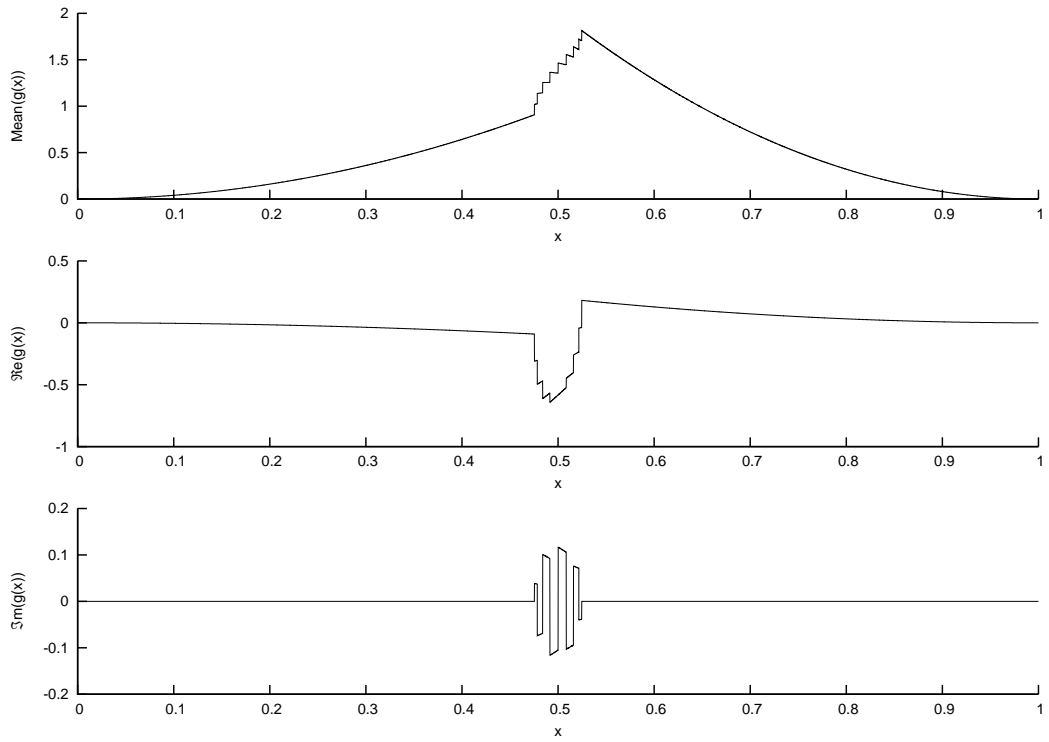


Figure 3.2: $g(x)$ with distribution $(4,0)$

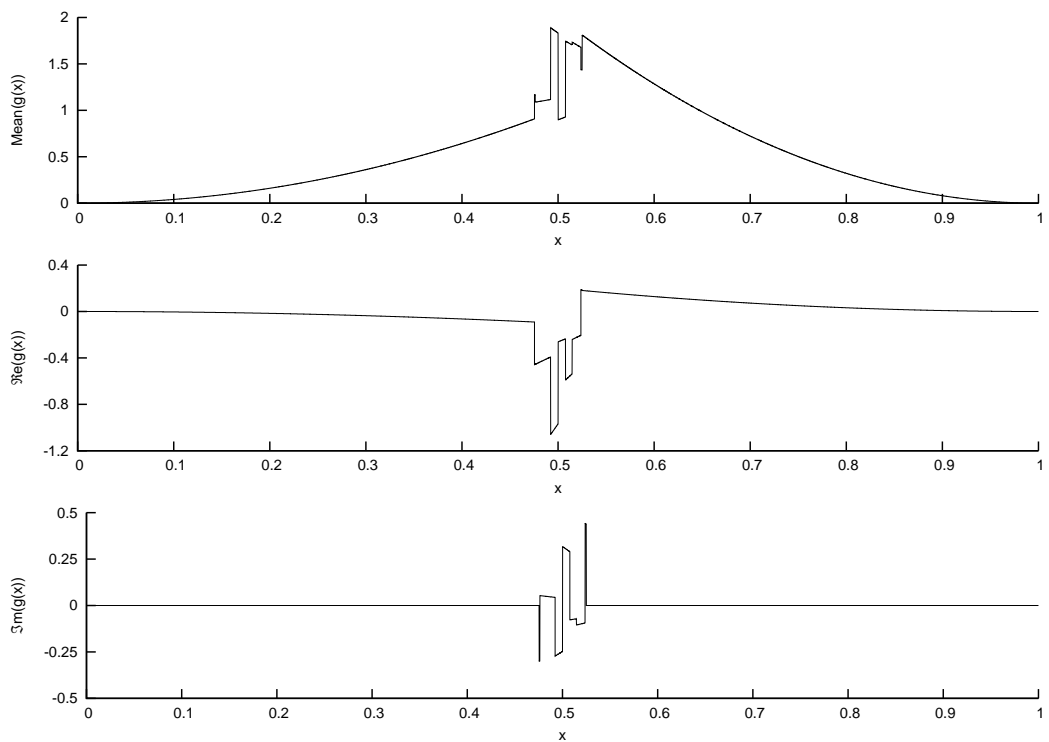


Figure 3.3: $g(x)$ with distribution $(4,3)$

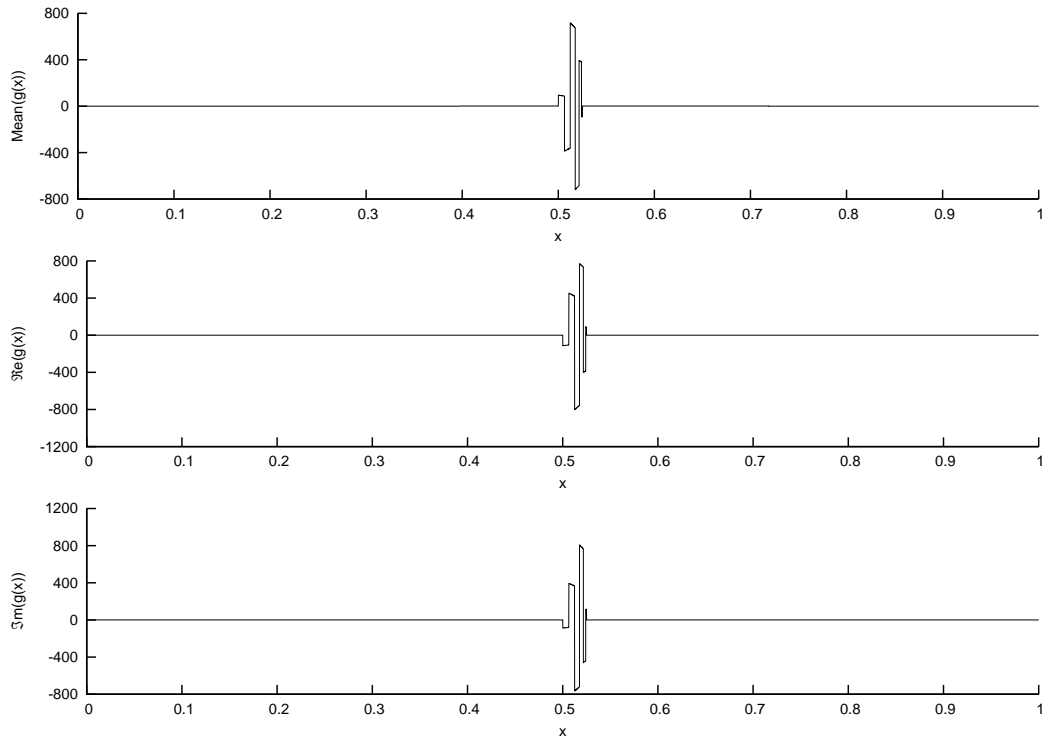


Figure 3.4: $g(x)$ with distribution $(3,1)$

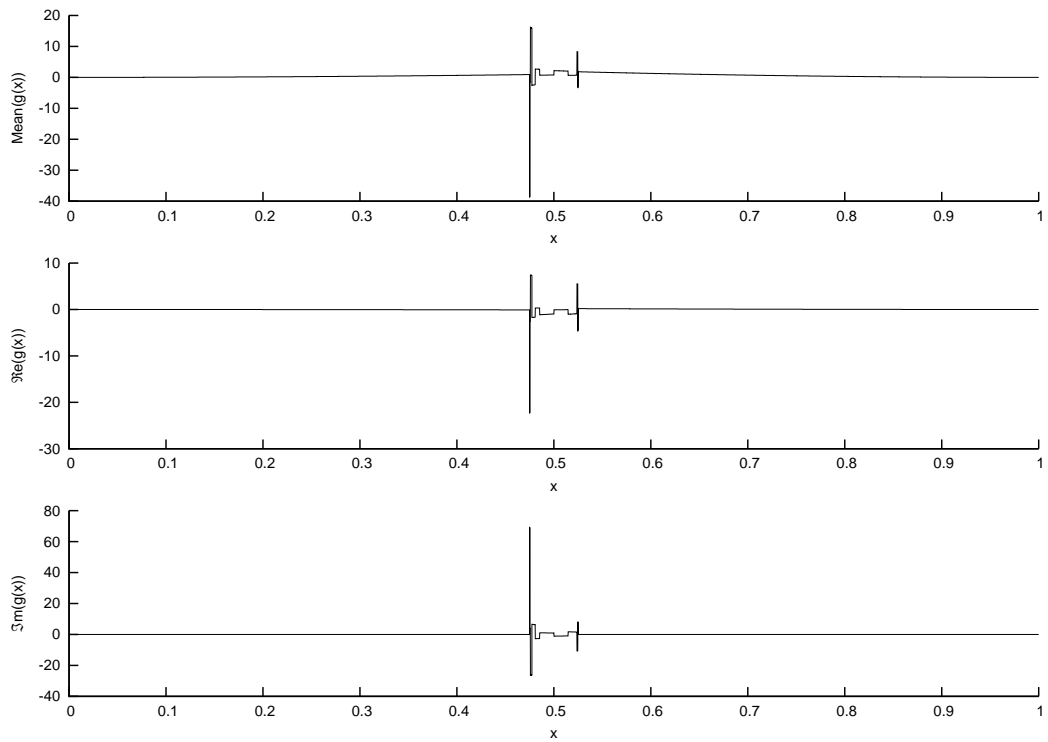


Figure 3.5: $g(x)$ with distribution $(5,3)$

Chapter 4

Methods to find restrictions for the choice of the instants

As a first approach, different ways are presented in order to find analytic expressions, that allow to determine the values the instants must not adopt in order to assure that A_1 is invertible.

4.1 Analytic solutions

Let K be a field of \mathbb{C} and let M be a square $n \times n$ matrix over this field K . Then M is invertible, if the following equivalent statements are satisfied:

$$\begin{aligned} M^{-1} \text{ exists} &\Leftrightarrow \text{rank}(M) = n \\ &\Leftrightarrow \text{Det}(M) \neq 0 \\ &\Leftrightarrow 0 \text{ is not an eigenvalue of } M \\ &\Leftrightarrow \text{the columns of } M \text{ span } K^n \text{ (i.e. the columns of } M \text{ form a basis of } K^n). \end{aligned}$$

Hence, a first approach would be to try to derive a condition for the instants by taking a look at the determinant. At first though, a small program has been conceived with *Maple* to build the matrix A_1 , that depends on the number of frequencies N taken in consideration. Since $\omega_0 = 0$, $t_0 = 0$ and $\omega_{-k} = -\omega_k$, (2.18) reads as follows:

$$A_1 = \begin{bmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \exp(-i\omega_N t_1) & \dots & \exp(-i\omega_1 t_1) & 1 & \exp(i\omega_1 t_1) & \dots & \exp(i\omega_N t_1) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \exp(-i\omega_N t_k) & \dots & \exp(-i\omega_1 t_k) & 1 & \exp(i\omega_1 t_k) & \dots & \exp(i\omega_N t_k) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \exp(-i\omega_N t_{2N}) & \dots & \exp(-i\omega_1 t_{2N}) & 1 & \exp(i\omega_1 t_{2N}) & \dots & \exp(i\omega_N t_{2N}) \end{bmatrix}.$$

4.1.1 Choice of the instants based on the determinant of A_1^N

Complex formulation To begin with, matrix A_1 and its determinant are taken in consideration with one single frequency $N = 1$:

$$A_1^{N=1} = \begin{bmatrix} 1 & 1 & 1 \\ \exp(-i\omega_1 t_1) & 1 & \exp(i\omega_1 t_1) \\ \exp(-i\omega_1 t_2) & 1 & \exp(i\omega_1 t_2) \end{bmatrix}.$$

$$\det(A_1^{N=1}) = \exp(i\omega_1 t_2) - \exp(i\omega_1 t_1) + \exp(-i\omega_1 t_1) - \exp(-i\omega_1 t_2) + \exp(i\omega_1(t_1 - t_2)) - \exp(i\omega_1(t_2 - t_1))$$

The angular frequency ω_1 is given. The unknowns are therefore t_1 and t_2 , two instants. By resolving $\det(A_1^{N=1}) = 0$, values for t_1 and t_2 are obtained, which they must not adopt for A_1 to be invertible. For $N = 1$, these values are easily obtained. The calculation can be performed manually:

$$\begin{aligned} \det(A_1^{N=1}) &= 2i [\sin(\omega_1 t_2) - \sin(\omega_1 t_1) + \sin(\omega_1(t_1 - t_2))] \\ &= 4i \left[\sin\left(\frac{\omega_1}{2}(t_1 - t_2)\right) \cdot \sin\left(\frac{\omega_1}{2}t_1\right) \cdot \sin\left(\frac{\omega_1}{2}t_2\right) \right], \\ \det(A_1^{N=1}) = 0 &\Leftrightarrow t_1 = t_2 \quad \text{or} \quad t_1 = \frac{m\pi}{\omega_1} + t_2 \quad \text{or} \\ &t_1 = 0 \quad \text{or} \quad t_1 = \frac{m\pi}{\omega_1} \quad \text{or} \\ &t_2 = 0 \quad \text{or} \quad t_2 = \frac{m\pi}{\omega_1}, \\ &(m \in \mathbb{N}). \end{aligned}$$

For two frequencies ($N = 2$), it is getting considerably worse. There are four unknown instants now: t_1, t_2, t_3 and t_4 . The matrix $A_1^{N=2}$ reads:

$$A_1^{N=2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \exp(-i\omega_2 t_1) & \exp(-i\omega_1 t_1) & 1 & \exp(i\omega_1 t_1) & \exp(i\omega_2 t_1) \\ \exp(-i\omega_2 t_2) & \exp(-i\omega_1 t_2) & 1 & \exp(i\omega_1 t_2) & \exp(i\omega_2 t_2) \\ \exp(-i\omega_2 t_3) & \exp(-i\omega_1 t_3) & 1 & \exp(i\omega_1 t_3) & \exp(i\omega_2 t_3) \\ \exp(-i\omega_2 t_4) & \exp(-i\omega_1 t_4) & 1 & \exp(i\omega_1 t_4) & \exp(i\omega_2 t_4) \end{bmatrix}.$$

The determinant $\det(A_1^{N=2})$ becomes noticeably more complex and already *Maple* does not manage to resolve $\det(A_1^{N=2}) = 0$ in a reasonable period of time on a regular computer. The computation had actually to be interrupted.

Real formulation Recall the remarks concerning matrix A_1 in section 2.2.7 and some properties of the determinant:

- The interchange of two lines (columns) changes the sign of the determinant
- The multiplication of a line (column) by a multiplies the determinant by a
- The addition of a multiple of one line (column) to another line (column) does not change the value of the determinant

Hence, for both formulations (complex and real approach), the same solutions of the equation $\det(A_1) = 0$ will be obtained and the two formulations will result in the same problems of resolution for higher orders.

4.1.2 Choice of the instants based on a decomposition of A_1^N into a lower and upper triangular matrix

The determinant of a matrix of upper triangular form is simply the product of its entries on the diagonal. We can therefore decompose A_1^N into LU (L : lower triangular matrix, U : upper triangular matrix). The definition of L states that L should only have "ones" on the diagonal. The decomposition furnishes:

$$\begin{aligned} A_1^N &= L \cdot U, \\ \det(A_1^N) &= \det(L \cdot U) = \underbrace{\det(L)}_{=1} \cdot \det(U) = \det(U). \end{aligned}$$

If each entry u_{ii} of U on the diagonal is non-zero, the determinant is non-zero and thus, A_1^N is invertible. That way, a solution may be found for the t_i by setting each $u_{ii} = 0$. Maybe *Maple* is able to resolve this more rapidly, because each term on the diagonal is less voluminous than the whole determinant. The remarks made above concerning the two formulations are valid here just as well. For $N = 1$, the same results, already known from the approach with the determinant, are obtained. For $N = 2$, the same problems as before are encountered.

Maple has enormous problems to solve the occurring equations for more than two unknown instants. For *Maple* the angular frequencies ω_k are unknowns, too. They cannot be set to their numeric values though, because a general analytic solution is wanted, which depends necessarily on the inflicted angular frequencies. Even if the numeric values of those frequencies would be given, *Maple* still could not solve these equations for more than two unknowns. The fact that the number of unknowns/instants is increased by two, when one supplementary frequency is considered, makes computations for higher orders quasi impossible.

4.1.3 Choice of the instants based on the eigenvalues of A_1^N

A_1^N is invertible, if zero is not an eigenvalue. Equation $\det(A_1^N - \lambda \cdot I) = 0$, that provides the characteristic polynomial, is therefore considered:

$$p_N \lambda^N + p_{N-1} \lambda^{N-1} + \dots + p_1 \lambda^1 + p_0 = 0. \quad (4.1)$$

If $p_0 \neq 0$, none of the eigenvalues $\lambda_i = 0$, but p_0 is simply $\det(A_1^N)$ and this leads back to the problem of section 4.1.1. Nevertheless, one may make *Maple* calculate the eigenvalues and see, if *Maple* can find solutions of $\lambda_i = 0$ for the instants t_n . The expressions of the three eigenvalues that are obtained are enormous, even for $N = 1$, for both the complex and real formulation. Hence, the expressions do not simplify with this approach.

4.1.4 Choice of the instants based on the singular values of A_1^N

One property of singular values of a matrix is: the smallest and the biggest one of them set limits to the absolute values of the eigenvalues:

$$\sigma_{min}(A_1) \leq |\lambda_i(A_1)| \leq \sigma_{max}(A_1)$$

Matrix A_1 is therefore invertible, if $\sigma_{min}(A_1) > 0$. The singular values are calculated by extracting the square root of the eigenvalues of the matrix product $A_1^T \cdot A_1$:

$$\sigma_i(A_1) = \sqrt{\lambda_i(A_1^T \cdot A_1)}$$

This might be a promising approach due to the very special structure of matrix A_1 . For $N = 1$, very long expressions are found for the three singular values that do not hold any complex terms or exponentials in complex formulation. The variables ω_k and t_n are anyway in the exponents of integers. One finds for each equation $\sigma_i(A_1^{N=1}) = 0$ the well known solutions. No analytic solution may be achieved with the matrix in real formulation and just like for the eigenvalues of the matrix $A_1^{N=2}$, the five singular values cannot be determined. Of course, trying to find a solution by treating the determinant of the matrix product $A_1^T \cdot A_1$ may be promising as well. If this determinant is not zero, the smallest singular value will not be either. It is not surprising that the same problems, that have already been described in section 4.1.1, are encountered.

4.1.5 Summary and proceeding

Obviously, an analytic solution cannot be obtained with the approaches mentioned above. This is mostly due to the intricate structure of the matrix and the incapacity to solve the upcoming equations for one variable, that depends on six more unknowns for $N = 2$ (i.e. $t_0 = 0$, $\omega_0 = 0 \rightarrow$ two unknown instants and four frequencies). If this was possible, instants could be chosen arbitrarily and then a quick check could have been performed to see, if this choice was a valid one. Furthermore a universal rule for any number of frequencies might have been derived by considering the restrictions for $N = 1$ to 5.

Nevertheless, in the following section, approaches to determine the instants step by step with a numeric code are proposed.

4.2 Numeric solutions

4.2.1 Choice of the instants in a way that the lines of A_1 are forming an orthogonal basis

The objective of this section is to add a restriction to the choice of instants in order to find simpler expressions. This restriction specifies that the lines of A_1 , considered as vectors, do not only form a basis, but even an orthogonal basis.

Complex formulation Matrix A_1 is considered in its most general form:

$$A_1 = \begin{bmatrix} \exp(-i\omega_N t_0) & \dots & \exp(i\omega_0 t_0) & \dots & \exp(i\omega_N t_0) \\ \vdots & & \vdots & & \vdots \\ \exp(-i\omega_N t_k) & \dots & \exp(i\omega_0 t_k) & \dots & \exp(i\omega_N t_k) \\ \vdots & & \vdots & & \vdots \\ \exp(-i\omega_N t_l) & \dots & \exp(i\omega_0 t_l) & \dots & \exp(i\omega_N t_l) \\ \vdots & & \vdots & & \vdots \\ \exp(-i\omega_N t_{2N}) & \dots & \exp(i\omega_0 t_{2N}) & \dots & \exp(i\omega_N t_{2N}) \end{bmatrix}.$$

If the lines of the matrix form an orthogonal basis, the dot product of no matter which two lines must be equal to zero: $\forall l, k \in \{0 \dots 2N\} : v_l \cdot v_k = 0$,

$$\begin{aligned} v_k &= [\exp(-i\omega_N t_k) \quad \dots \quad \exp(i\omega_0 t_k) \quad \dots \quad \exp(i\omega_N t_k)], \\ v_l &= [\exp(-i\omega_N t_l) \quad \dots \quad \exp(i\omega_0 t_l) \quad \dots \quad \exp(i\omega_N t_l)], \\ v_l \cdot v_k^T &= \exp(-i\omega_N(t_k + t_l)) + \dots + \exp(i\omega_0(t_k + t_l)) + \dots + \exp(i\omega_N(t_k + t_l)) = 0. \end{aligned}$$

This equation may be simplified by passing to trigonometric notation and by setting $\omega_0 = 0$:

$$1 + 2\cos(\omega_1(t_k + t_l)) + \dots + 2\cos(\omega_i(t_k + t_l)) + \dots + 2\cos(\omega_N(t_k + t_l)) = 0. \quad (4.2)$$

Before the analysis of this case for one single frequency, it is remarked again that this relation must satisfy each combination of t_k and t_l . For $N = 1$, we resolve the equation:

$$1 + 2\cos(\omega_1(t_k + t_l)) = 0.$$

Thus, we find the following relation among the instants t_k and t_l :

$$t_k = t_l + \frac{a}{\omega_1}; \quad \text{with} \quad a = \frac{2\pi}{3}, \frac{4\pi}{3}, \frac{8\pi}{3}, \frac{10\pi}{3}, \frac{14\pi}{3}, \frac{16\pi}{3}, \dots$$

Three instants have to be determined: t_0 , t_1 and t_2 . One may choose as usually $t_0 = 0$. According to the above relation, $a = \frac{2\pi}{3}$ is chosen and the second instant is determined to $t_1 = \frac{2\pi}{3\omega_1}$. Since this relation must also be satisfied for t_0 and t_2 , the natural choice for the third instant would be $t_2 = \frac{4\pi}{3\omega_1}$. It can be seen though, that the third relation is not satisfied:

$$1 + 2\cos(\omega_1(t_1 + t_2)) = 1 + 2\cos\left(\omega_1\left(\frac{2\pi}{3\omega_1} + \frac{4\pi}{3\omega_1}\right)\right) = 1 + 2\cos(2\pi) = 3 \neq 0.$$

The solution consists in choosing the coefficients a of each instant in a way that the sum of the two chosen coefficients is another value out of the possible coefficients. This would be true e.g. for $a = \frac{8\pi}{3}$, which furnishes $t_2 = \frac{8\pi}{3\omega_1}$ and hence:

$$1 + 2\cos(\omega_1(t_1 + t_2)) = 1 + 2\cos\left(\omega_1\left(\frac{2\pi}{3\omega_1} + \frac{8\pi}{3\omega_1}\right)\right) = 1 + 2\cos\left(\left(\frac{10\pi}{3}\right)\right) = 1 + 2\left(-\frac{1}{2}\right) = 0.$$

If more frequencies are being taken in account, the resolution of Eq. 4.2 does not provide such a pretty solution anymore. Actually, the zeros of the function on the left side of Eq. 4.2 have to be determined numerically e.g. with an algorithm of *Newton*. To do so, the period of this function is calculated first. This is the least common multiple (LCM) of the frequencies' periods:

$$T = LCM\{T_1, T_2, \dots, T_N\} \quad (4.3)$$

There exist J solutions in the period of the following form:

$$x = t_k + t_l \in S = \{s_1, s_2, s_3, \dots, s_J\}. \quad (4.4)$$

The solutions $s_i + T$ (and equally $s_i + 2T$, $s_i + 3T$, ...) may be added to S . These are valid as well, but they do not need to be calculated separately.

Real formulation Special attention needs to be paid to the fact that different solutions are obtained, if a real formulation is used. The solutions of the real formulation are therefore examined, because this is the formulation used for programming.

$$A_1 = \begin{bmatrix} 1 & \cos(\omega_1 t_0) & \dots & \cos(\omega_N t_0) & \sin(\omega_1 t_0) & \dots & \sin(\omega_N t_0) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(\omega_1 t_k) & \dots & \cos(\omega_N t_k) & \sin(\omega_1 t_k) & \dots & \sin(\omega_N t_k) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(\omega_1 t_l) & \dots & \cos(\omega_N t_l) & \sin(\omega_1 t_l) & \dots & \sin(\omega_N t_l) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(\omega_1 t_{2N}) & \dots & \cos(\omega_N t_{2N}) & \sin(\omega_1 t_{2N}) & \dots & \sin(\omega_N t_{2N}) \end{bmatrix},$$

$$v_l = [1 \quad \cos(\omega_1 t_k) \quad \dots \quad \cos(\omega_N t_k) \quad \sin(\omega_1 t_k) \quad \dots \quad \sin(\omega_N t_k)]$$

$$v_k = [1 \quad \cos(\omega_1 t_l) \quad \dots \quad \cos(\omega_N t_l) \quad \sin(\omega_1 t_l) \quad \dots \quad \sin(\omega_N t_l)],$$

$$v_l \cdot v_k^T = 1 + \cos(\omega_1 t_k)\cos(\omega_1 t_l) + \dots + \cos(\omega_N t_k)\cos(\omega_N t_l) \\ + \sin(\omega_1 t_k)\sin(\omega_1 t_l) + \dots + \sin(\omega_N t_k)\sin(\omega_N t_l).$$

Merging the terms of the same frequency leads to:

$$1 + \cos(\omega_1(t_l - t_k)) + \dots + \cos(\omega_N(t_l - t_k)) = 0. \quad (4.5)$$

For $N = 1$, $1 + \cos(\omega_1(t_l - t_k)) = 0$ is resolved and the following relation is found:

$$t_k = t_l + \frac{a}{\omega_1}; \quad \text{with } a = \pi, 3\pi, 5\pi, \dots$$

Solutions similar to the complex formulation cannot be found. Generally speaking, for $N > 1$ in both, complex and real formulation, solutions like $t_k, t_l \in S$ and $t_k + t_l \in S \forall l, k \in \{0, \dots, 2N\}$ cannot be found. However, a system of equations may be created for all combinations of t_k, t_l : $t_k + t_l = s_i$ with $i \in \{0, \dots, J\}$. According to the formula of *Gauss*, this system holds $\frac{1}{2} \cdot ((2N + 1) - 1) \cdot (((2N + 1) - 1) + 1) = N \cdot (2N + 1)$ equations for N frequencies ($(2N + 1)$ instants). Hence, this approach features three major disadvantages:

1. The system of equations is over-determined for $N > 1$ and hence, a solution for every combination of instants cannot be found in S .
2. $t_k = 0$ cannot always be a solution anymore. If it was defined e.g. $t_0 = 0$, one would obtain immediately $t_k \in S \forall k \in \{1, \dots, 2N\}$ because of Eq. (4.4) and it is mentioned above that these instants do not resolve all equations.
3. The calculated instants might also be negative, which could be accepted, though, due to the periodicity of the problem.

4.2.2 Evolutionary determination

The determinant of a matrix can be found with the formulae of *Laplace*:

$$\det A = \sum_{i=1}^n (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij} \quad (\text{development by column } j),$$

$$\det A = \sum_{j=1}^n (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij} \quad (\text{development by line } i).$$

If matrix A_1 is developed by column subsequently, an expression, in which the determinants of matrices of the order of 2×2 appear, is obtained. Actually, the determinants of matrices built with every possible combination of two columns of the last two lines will be found.

$$\det(A_1) = c_1(t_1, \dots, t_{2N-2}) \cdot |[2 \times 2]| + c_2(t_1, \dots, t_{2N-2}) \cdot |[2 \times 2]| + \dots \quad (4.6)$$

These determinants will be referred to as *sub-determinants* of sub-matrices with a certain dimension d (given in the superscript): SD^d . A matrix of dimension $(2N + 1) \times (2N + 1)$ leads to $\binom{2N+1}{\dim(SD)}$ different *sub-determinants* ($\dim(SD)$: dimension of sub-determinant). Obviously, there exist much more sub-determinants, but only $\binom{2N+1}{\dim(SD)}$ are different from each other.

E.g.: $N = 2$, $\dim(SD) = 2 \rightarrow \binom{5}{2} = 10$. Hence, there exist ten different determinants when the 2×2 sub-matrices of a 5×5 matrix are considered. The following equation illustrates the development in sub-determinants with $d = 2$ for 2 frequencies. The coefficients c_i are unknowns for now and for $N = 2$ they depend on t_1 and t_2 . They derive from the formula of *Laplace*.

$$\det(A_1^{N=2}) = c_1(t_1, t_2) \cdot SD_1^2 + \dots + c_{10}(t_1, t_2) \cdot SD_{10}^2 \quad (4.7)$$

The instant t_{2N} (the last instant) fixes the observed period. Thus, t_{2N} may be considered known. For the determination of the last instant see section 4.3. Furthermore, the number of frequencies, the frequencies themselves and therefore the angular frequencies ω_k are known. The *sub-determinants* of Eq. (4.7) must not be zero simultaneously, otherwise the determinant $\det(A_1)$ is equal to zero. An even stricter restriction is used here: None of the SD_i^2 may be zero. The only unknown of the SD_i^2 is the instant t_3 in the present example. Hence, $\binom{5}{2} = 10$ equations $SD_i^2 = 0$ must be resolved for the instant t_3 in the period $t_0 \leq t_3 \leq t_4$ with e.g. an algorithm of *Newton*. Having done that, all the values, that t_3 must not adopt, are known and a value for this instant can be chosen arbitrarily. The instants t_0 , t_3 and t_4 are then known numbers in the matrix $A_1^{N=2}$.

The next step is to determine the instant t_{2N-2} . In the present example: $t_{2N-2} = t_2$. To do this, all combinations of three columns of matrix A_1 of the last three lines are considered and the *sub-determinants* SD^3 of dimension 3 are calculated. By carrying through the same steps described above, one can determine the value of t_{2N-2} (here: $t_{2N-2} = t_2$):

$$\det(A_1^{N=2}) = c_1(t_1) \cdot SD_1^3 + \dots + c_{10}(t_1) \cdot SD_{10}^3 \quad (4.8)$$

This must be repeated until all instants except for instant t_1 are known. Then attention must be paid: The instant $t_0 = 0$ is known. This means, that if instant t_1 would be determined with the help of *sub-determinants* $SD^{2N-1} \neq 0$ (here: $SD^4 \neq 0$), the combination of them, that furnishes $\det(A_1)$ might still be zero. SD^{2N-1} is skipped therefore and $\det(A_1)$ is calculated directly, which is only a function of t_1 and the values, that t_1 must not adopt are found by resolving $\det(A_1) = 0$. After having given t_1 a numeric value, one knows all the instants and is sure of A_1 to be invertible.

It is possible to accept that one or more of the sub-determinants become zero, when choosing the next time step (particularly if they are of higher order, i.e. determinants of matrices of a small dimension compared to A_1). This does not affect the determinant considerably.

The procedure described above was realised with a *Maple* program. The structogram of this program is given in the appendix. The program in the present configuration is able to perform the necessary computations up to an order of $N = 5$. The extension to higher orders is easy, one only has to add some extra external procedures and call them in the main code. These external procedures build the sub-determinants of matrix A_1 . For each dimension of sub-matrices, a proper procedure is needed. Procedures for a maximum sub-matrix dimension of 9×9 are already developed (this means, that instants for orders up to $N = 5$ can be determined). Procedures for higher orders can be obtained by copy-and-paste of these existing procedures and some minor modifications. It has to be mentioned that computation does become quite expensive for high orders. For an order of $N = 5$ ($\dim(A) = 11$) and a sub-matrix of dimension 5×5 , there exist $\binom{11}{5} = 462$ different sub-matrices and therefore 462 determinants with one unknown instant, for which the values of the instant have to be found so that the respective sub-determinants equal zero.

4.3 Determination of the last instant t_{2N}

The last instant needs to be chosen carefully. In the monofrequential approach with uniformly distributed instants, the last instant is always $t_{2N} = \frac{2N}{2N+1} \frac{1}{f}$. This is not a necessary condition anymore, when the Multifrequential HBT is used. The last instant may be chosen arbitrarily, but one must act with caution, because convergence behaviour might change dramatically when the last instant is chosen too small (i.e. near t_0). This is also true for a distribution of the instants far away from a uniform distribution. Tests regarding this issue were effected and are described in Chap. 7.

Chapter 5

Implementation

The modified computation of the source term and the fact of taking in account several frequencies and instants that might not be uniformly distributed over the considered period must be implemented in the *elsA* code (ensemble logiciel pour la simulation en Aérodynamique [2]). *elsA* is an *Onera* software co-developed by *Cerfacs* that is able to treat internal and external flows as well as multidisciplinary aerodynamic applications, including:

- aerodynamics, aeroelastics
- airplanes, helicopters, turbo-engines, missiles, rockets, air inlets, nozzles
- research and industrial applications
- Euler, RANS, URANS, DES, LES simulations
- hypothesis of pure ideal gas with a given value for the specific heat coefficient
- computation of sensibility for ideal conception

The code is object-oriented and is written in *C++* with routines for low-level computation written in *FORTRAN*. The user interface is offered by a *Python* upper layer.

All modifications described in the following sections were realised in a way that the monofrequential approach still runs on this version of the *elsA* code. The structograms of all modified or newly created code are given in the appendix.

5.1 Interface

Some additional input is needed for the Multifrequential HBT. First of all, each frequency must be defined separately, because the frequencies are not multiples of a base frequency anymore. Furthermore, the instants not being necessarily uniformly distributed must also be declared by the user.

The necessary *Python*-keys are created for ten frequencies and consequently 21 instants. This is thought to be sufficient for all upcoming problems. Anyway, the number of frequencies/instants can easily be increased by copy-and-paste. The order of the computation (or the number of frequencies) remains an input just like for the Monofrequent HBT, because it is a frequently needed constant (e.g. for the computation of matrix A_1 , etc.).

The implementation of the HBT in the *elsA*-code must acknowledge the existence of several instants of the same computation. Besides, the HBT is meant to be used in RANS. Hence, two different systems are treated in *elsA*: a mean field and a turbulent field. There is no multigrid application for the turbulent field. The constructor (of the source term, but also of other operators like e.g. the flux) must therefore be called for each multigrid level of the mean field and for the turbulent field. This is done for each instant and was realised for the Monofrequent HBT by duplication of blocks. The identifier *brotherhood* is introduced here. Blocks of the same brotherhood derive from the same block of the original computation, but may be at a different instant, multigrid level or system. If the original computation disposes of n blocks, a multigrid level of m is employed and N frequencies are taken in account, the final number of blocks is $nm(2N + 1)$ for an Euler or laminar computation and $n(m + 1)(2N + 1)$ for a turbulent computation. It is chosen here to define all frequencies and instants for each block. This makes the predefinition in the *Python* layer a little more laborious, but the great advantage is obviously, that each block may be given different frequencies or instants. This will be needed for multistage turbo machinery applications.

The additional *Python* keys for the monofrequent approach (*harmonic_balance*, *harmonic_bal_steps* and *harmonic_frequency*), that are still in use here, are listed in Tab. 5.1. These were completed by *hbt_multifreq_freq_i* ($1 \leq i \leq 10$) and *hbt_multifreq_timelevel_j* ($0 \leq j \leq 20$).

Table 5.1: Python keys.

Key	Signification	Type	Values
<i>harmonic_balance</i>	HBT active ?	list	['inactive', 'HBT', 'HBT deform']
<i>harmonic_bal_steps</i>	number of frequ.	integer	[0]
<i>harmonic_frequency</i>	frequ. (monofrequ.)	real	[0.0]
<i>hbt_multifreq_freq_01</i>	frequ. (multifrequ.)	real	[0.0]
⋮			
<i>hbt_multifreq_freq_10</i>			
<i>hbt_multifreq_timelevel_00</i>	instants (multifrequ.)	real	[0.0]
⋮			
<i>hbt_multifreq_timelevel_20</i>			

5.2 Main code

A new class *SouHbtMultiFreq.C* was derived from class *SouHbt.C* which is derived from *OperSource.C*. The necessary attributes that are used both for the monofrequent and multifrequent approach are given in List. 5.1.

Listing 5.1: *Attributes of SouHbt.C and SouHbtMultiFreq.C.*

```

/* List of Source Terms and associated Grids */
vector<const SouHbt*>      _listSouHbt;
vector<const GeoGridBase*> _listGrid;

/* Terms shared by all instances of SouHbt / SouHbtMultiFreq*/
vector<E_Float>           _f;           // base frequency of monofreq.
vector<FldArrayF*>       _dCoeff;      // D-coefficients
vector<FldArrayF*>       _fMulti;      // frequencies of multifreq.
vector<FldArrayF*>       _timeLevels;  // time levels of multifreq.
E_Int                    _currentMgLevel; // multigrid level
E_Int                    _N;           // number of frequ. / order

/* Parameters of the current instance of SouHbt / SouHbtMultiFreq */
const E_Int              _id;           // id of the block
const E_Int              _step;        // step in the time period
const E_Int              _brother;     // brotherhood
const E_Int              _nbGrids;     // number of multigrid levels
const E_Int              _mgLevel;     // multigrid level

```

In *createSou.C*, a query states whether a frequency for the Multifrequential HBT was declared or not. If so, the frequencies and instants defined by the user are passed to two field arrays and are now available in the software. In class *SouHbtMultiFreq.C*, one first passes by the constructor *SouHbt::SouHbt*, where the lists of blocks and source terms are created. In the case of a multifrequential application, the base frequency of the monofrequential approach is set to zero. This will provide a possibility to distinguish in the program whether the monofrequential or the multifrequential part of the code is to be used. In the constructor of *SouHbtMultiFreq.C*, the vectors *_fMulti* and *_timeLevels* are equipped with the values of the $(N + 1)$ frequencies ($f_0 = 0$) and the $(2N + 1)$ instants. Then, the method *computeDCoeff* is called to calculate the elements of the field array *dCoeff*.

5.2.1 Computation of matrix D

The basic modification is the computation of the source term. For the monofrequential approach, all source term elements are calculated by simply applying Eq. (2.14). As mentioned, for the Multifrequential HBT, these elements are obtained by exploiting the matrix products (cf. Eq. (2.33)). The elements of matrix D are referred to as D-coefficients, because they multiply the conservative variables. Four *FORTRAN* routines were added and are called in *computeDCoeff*:

- *souhbtmultifreqa* - Matrix A_1 is defined here, all matrix elements depending on the known frequencies and instants are calculated

- *souhbtmultifreqdadt* - Matrix $\frac{\partial A_1}{\partial t}$ is defined here, all matrix elements depending on the known frequencies and instants are calculated
- *souhbtmultifreqainv* - This routine gets Matrix A_1 as input and calculates its inverse using the *LAPACK*¹-subroutines *dgetri.f* and *dgetrf.f*
- *souhbtmultifreqdccoeff* - Performs the matrix product $\frac{\partial A_1}{\partial t} \cdot A_1^{-1}$ and furnishes D

5.2.2 Computation of the source term

Two methods in *SouHbtMultiFreq.C* had to be modified for the computation of the source term: *compute* and *updateSourceTerm*. These are actually applying Eq. (2.33) $S = VDF$ except that the multiplication by the volume V is done in *RhsTerm::compute()*.

The source term is identified by three attributes that do not change during the computation:

- An integer *_id* defining non-ambiguously the source term (it is equal to the corresponding block ID).
- The time step in the period: $0 \leq \textit{_step} \leq 2N$.
- The brotherhood *_brother*.

The attributes *_currentMgLevel*, *_mgLevel* and *_nbGrids* are auxiliary attributes to determine the multigrid level. It is iterated over all instances of the source list and the ones of same brotherhood and multigrid level are determined as the present term. If this is the case, the conservative field and the corresponding D-coefficient are loaded in order to update the source term.

Checking the brotherhood and multigrid level attributes is done in *compute*. In the monofrequential code, there is additionally a query whether the current time step is the same. If so, the source term is not updated because there are zeros on the diagonal of matrix D. This simplification must not be effected in the multifrequential approach, because the diagonal elements cannot be stated to be zero in general. The method *updateSourceTerm* is called then. In the monofrequential approach, only a small part of the matrix D has to be calculated, because of the matrix' inherent properties (zeros on the diagonal, $\omega_{-k} = -\omega_k$ and each line is a permutation of the first line of matrix D). Obviously, the whole matrix D must be gotten in *updateSourceTerm* to pick the right coefficient in the multifrequential approach. The current instant provides the information in which line this coefficient remains and the column is determined by the time step of the element of the source term list. The D-coefficient is then passed to *FORTTRAN* subroutine *souhbt_*, which updates the sum of the source term for all cells and all equations of the considered system.

¹LAPACK - Linear Algebra PACKage (<http://www.netlib.org/lapack/>): *dgetri.f* computes the inverse of a general matrix, using the LU factorization, computed by *dgetrf.f*. *dgetrf.f* computes an LU factorization of a general matrix, using partial pivoting with row interchanges.

5.2.3 Stability

When a resolution of the Navier-Stokes equations is effected, the local time step must be limited in order to guarantee stability of the equations. Hence, the convection time step of a cell is calculated with a CFL number (Courant, Friedrichs, Levy) imposed by the user:

$$\Delta\tau = CFL \frac{\mathcal{V}(\Omega)}{\rho(A)}. \quad (5.1)$$

$\mathcal{V}(\Omega)$ is a characteristic dimension of the cell and $\rho(A)$ is the spectral radius of the Jacobian that is equal to the maximum characteristic velocity: $\rho(A) = \|u\| + c_0$. $\|u\|$ is the module of the velocity and c_0 is the speed of sound.

Due to the addition of the source term in Eq. (2.32), the computation of the virtual time step $\Delta\tau$ must be modified in order to solve the steady computation Eq. (2.31) (cf. [18]). The virtual time step is estimated with a stability analysis in the frequency domain (cf. [13]). The modification consists in adding a term in the denominator:

$$\Delta\tau = CFL \frac{\mathcal{V}(\Omega)}{\rho(A) + k_p \mathcal{V}(\Omega)} \quad \text{where} \quad k_p = \omega_{max}. \quad (5.2)$$

Actually, the most restrictive definition has been chosen by taking the highest frequency or smallest wavelength, otherwise multiple matrix inversions would have been necessary (for a deeper insight cf. [13]). In the monofrequential approach $\omega_{max} = N\omega = 2\pi Nf$, because all frequencies are multiples of the base frequency. In the multifrequential approach $\omega_{max} = 2\pi f_{max}$.

The following method has to be modified: *TmoTimeStepLocalConvGridMotion*. The modifications consist in performing a query whether the monofrequential or the multifrequential approach is used. This is simply done by checking which constant has the greater value: the frequency constant (monofrequential) or the maximum frequency of the vector of frequencies (multifrequential). Either the frequency of the monofrequential approach is passed after being multiplied by the order (or number of frequencies) of the current computation or the maximum frequency of the multifrequential approach is passed to the *FORTRAN* subroutine, calculating the actual virtual time step. This is *compcellspecradius2_* for *TmoTimeStepLocalConvGridMotion*. Before, the frequency and the order were passed to these subroutines and multiplied in there. This is now done in the *C++* methods, because obviously the maximum frequency of the multifrequential approach is not to be multiplied by the order.

5.2.4 Implicit method

An implicit solver is used, in order to be able to apply high CFL numbers (at the order of 10^2). The theory of the implicit method is described here in short. The Navier-Stokes equations in semi-discrete form are noted:

$$V \frac{\partial W}{\partial t} + R(W) = 0. \quad (2.1)$$

In completely discrete form, the equation is noted:

$$V \frac{W^{t_0+\delta t} - W^{t_0}}{\delta t} + R(W^*) = 0,$$

where the * value remains unknown.

With an explicit approach, $W^{t_0+\delta t}$ is obtained by calculating R directly at t_0 :

$$W^{t_0+\delta t} = W^{t_0} + \frac{\delta t}{V}R(W^{t_0}).$$

However, with an implicit approach, it is not possible to calculate $W^{t_0+\delta t}$ directly from W^{t_0} :

$$W^{t_0+\delta t} = W^{t_0} + \frac{\delta t}{V}R(W^{t_0+\delta t}).$$

The value of $R(W^{t_0+\delta t})$ is approximated around the value of W at t_0 :

$$\begin{aligned} R(W^{t_0+\delta t}) &= R(W^{t_0}) + \frac{\partial R}{\partial t}(W^{t_0})\delta t \\ &= R(W^{t_0}) + \frac{\partial R}{\partial W}(W^{t_0})\frac{\partial W}{\partial t}(t_0)\delta t \\ &= R(W^{t_0}) + \frac{\partial R}{\partial W}(W^{t_0})(W^{t_0+\delta t} - W^{t_0}), \end{aligned}$$

leading to:

$$(I - \frac{\delta t}{V}\frac{\partial R}{\partial W}(W^{t_0}))(W^{t_0+\delta t} - W^{t_0}) = \frac{\delta t}{V}R(W^{t_0}).$$

elsA is a finite volume code treating structured multi-block grids. The implicit phase is also grid block oriented. This means, similarly to the fact that the field's increment is obtained sequentially block after block, the implicit phase follows suit. The Navier-Stokes equations in their compact form with the term pertaining to the virtual time step read:

$$V\frac{\partial F}{\partial \tau} + Q(F) = S(F). \quad (5.3)$$

The source term has a negative algebraic sign in *elsA*. This is why it appears on the right hand side of Eq. (5.3). With the exact same procedure stated above, Eq. (5.3) becomes:

$$V\frac{\partial F}{\partial \tau} + \frac{\partial Q}{\partial F}(F^{t_0}) + Q(F^{t_0}) = \frac{\partial S}{\partial F}(F^{t_0}) + S(F^{t_0}). \quad (5.4)$$

Hence, the implicit treatment of the source term makes use of the conservative fields at all time steps and the D-coefficients will therefore be needed in the implicit phase. This is why some modifications have to be effected in the method *computeLhs* of the class *TmoSetOfSolver_1.C*.

These modifications are actually similar to the ones described in section 5.2.2. Again, the terms on the diagonal of matrix D must be taken in account contrary to the monofrequential approach. A case differentiation is therefore implemented, based on the principle described in section 5.2.3 and instead of loading the repetitive coefficients of the monofrequential approach, matrix D is used to extract the D-coefficients for the source term computation.

Chapter 6

Validation

To validate the modified *elsA* code, tests were run that had also been effected for the monofrequential code. When the input values are equal (one frequency and its harmonics and uniformly distributed instants), the exact same results must be obtained. Convergence behaviour should be equal as well. In particular, the exact same residuals should be obtained. Three two-dimensional and two three-dimensional test cases were taken in consideration. For each computation the frequencies and instants are imposed that have been used for the monofrequential validation and are then compared to the monofrequential solution.

6.1 Forced motion of a 2D airfoil

This section describes a pitching airfoil in subsonic and transonic flow with harmonic excitation (cf. Sicot [13]). Given a mean angle of attack of α_0 , the airfoil is subjected to a sinusoidal variation of the angle of attack of amplitude α_m and pulsation ω :

$$\alpha(t) = \alpha_0 + \alpha_m \sin(\omega t) \quad (6.1)$$

The 2D test case NACA 64A010 is considered in three configurations given in Tab. 6.1. The reduced frequency of oscillation $\bar{\omega} = \omega c / (2 V_\infty)$ where c is the airfoil's chord and V_∞ the velocity at infinity. The rotation axis is at $x/c = 0.269$ for CT2 and at $x/c = 0.249$ for the two other test cases. CT2 is

Table 6.1: *Parameters of Test Case NACA 64A010.*

	$\alpha_0(deg)$	$\alpha_m(deg)$	$Re(\times 10^6)$	$f(Hz)$	$\bar{\omega}$	M_∞
CT2 (DI29)	-0.22	1.02	9.98	10.8	0.100	0,502
CT6 (DI55)	-0.21	1.01	12.56	34.4	0.202	0.796
Shock Stall Case (DI89)	4.0	1.01	11.88	35.1	0.204	0.789

completely subsonic and allows a first simple flow computation. CT6 is often used in literature to

test the precision of numeric methods. The correct prediction of the aerodynamic field requires a precise simulation of the interaction of shock and boundary layer. The last test case is at increased angle of attack and presents a boundary layer separation. The experimental data (emanating from Davis [3]) and a solution obtained from a URANS AEL¹ computation is also given in the presented figures for the sake of completeness, although they are not subject to the present discussion which only aims to prove equality of the two approaches. The unsteady AEL computations originate from *Delbove* [4] (DTS computation with 40 instants per period). The data for comparison with the Monofrequential HBT originates from *Sicot* [13].

The grid is fix referring to the airfoil and the variations of the angle of attack are accounted for by changing the boundary conditions of the far field with time.

When a flow shows boundary layer separation, the turbulence model has a major influence on the quality of the results (*Bohbot* [1]). However, the shock's dynamic in these applications depends rather on the airfoil's motion than on turbulence effects. The model of Spalart and Allmaras [14] is considered to be in good agreement with costs and physical precision by *Srinivasan* [16] and will therefore be applied here.

6.1.1 Pitching at low angle of attack in subsonic flow: CT2 (DI29)

This first test case shows the behaviour of the HBT in an entirely subsonic flow without shock. The distribution of the dimensionless pressure coefficient C_p on the airfoil is compared. The pressure coefficient is defined as the difference between the local static pressure and the static pressure at infinity normalised by the dynamic pressure at infinity:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty |V_\infty|^2}$$

In incompressible flow $C_p = 1$ at the stagnation point of the leading edge, for compressible flow $C_p \geq 1$. The pressure coefficient decreases then rapidly both on the suction side and on the pressure side and tends to a small positive value near the trailing edge. In the aeronautic community, C_p graphs are plotted with the axis of ordinates being inverted. The real and imaginary parts of the pressure coefficient in complex notation are displayed with no inversion of the axis of ordinates. The real and imaginary parts of the experimental data are normed by the amplitude of the pitching angle α_m (with α_m in radian): $\Re(C_p)/\alpha_m + i\Im(C_p)/\alpha_m$.

Convergence

The residuals of the computations for different numbers of harmonics are plotted in Fig. 6.1. For computations with up to four harmonics, a *CFL* number of 100 was sufficient. After 300 iterations, five orders of magnitude were lost. When a fifth harmonic is taken into consideration, the *CFL* number had to be decreased to 55 in order to achieve convergence. This is actually similar to the computations that have been made with the monofrequential code. The curve progression of each case is also akin to a monofrequential calculation (monofrequential residual progression over the number of iterations is not depicted here). As regards the convergence behaviour of the

¹The AEL module is the aero-elasticity module of *elsA*

multifrequential approach, it is similar to the monofrequential approach, but residuals are not strictly the same. The author believes that this discrepancy is due to the computation of the D-coefficients. Although those are calculated with double precision, they are not exactly equal to the last digit when being compared to the monofrequential analytically calculated D-coefficients. This will be discussed to a greater extent in section 6.3.

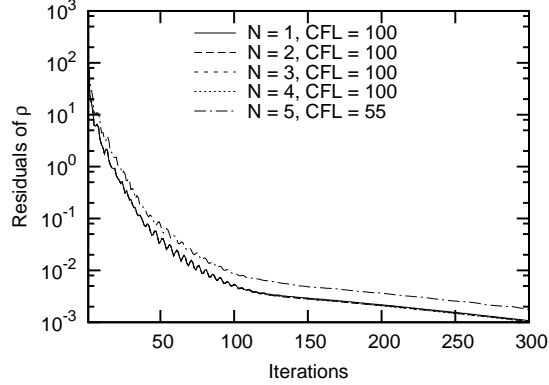
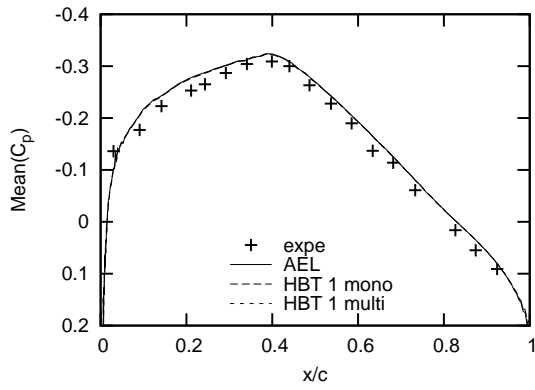


Figure 6.1: *NACA 64 CT2: Convergence.*

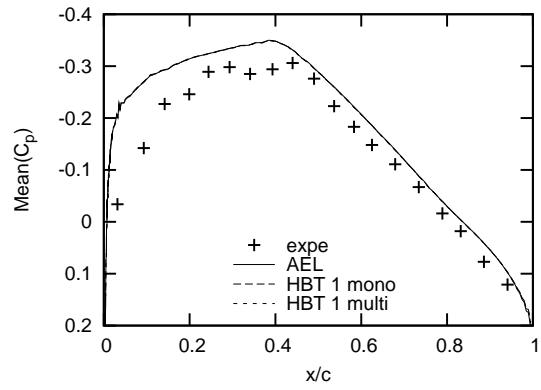
Results

Computations have been performed for 1, 2, 3, 4 and 5 frequencies. They all show perfect accordance to the monofrequential computations. The results for the orders $N = 1$ and 5 are shown in the following figures. For one frequency and three instants per period, results are shown in Fig. 6.2. The computations of the mean pressure coefficient with URANS AEL, the Monofrequential HBT and the new Multifrequential HBT are perfectly coincident (cf. Fig. 6.2(a) - 6.2(b)). This applies for the real part (cf. Fig. 6.2(c)) of the pressure coefficient as well. In Fig. 6.2(d) the imaginary part of C_p is displayed. A small shift to the URANS computations is observed, but the result of the Monofrequential and Multifrequential HBT are again coincident, which is to be shown here. Fig. 6.3 depicts the resulting graphs when five harmonics are taken in consideration. The results are essentially the same and even for the imaginary part of the pressure coefficient accordance to the URANS computation is achieved.

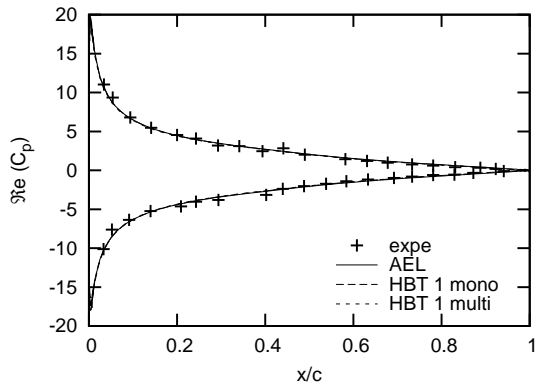
It was pointed out, that for this first simple test case the multifrequential approach shows no differences to the monofrequential one. Except for small differences in the residuals, the resulting graphs of the solution are the same and convergence behaviour is equal. Therefore a more demanding test case, actually provoking the emergence of a shock, may now be approached.



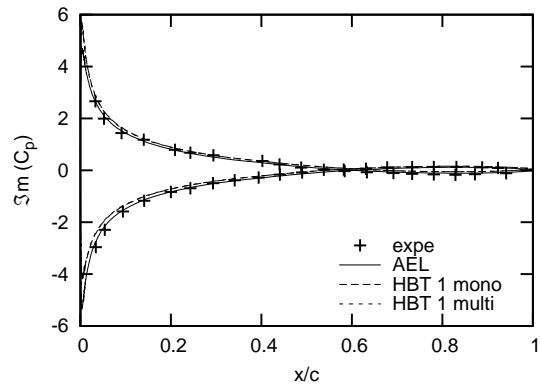
(a) Suction side



(b) Pressure side

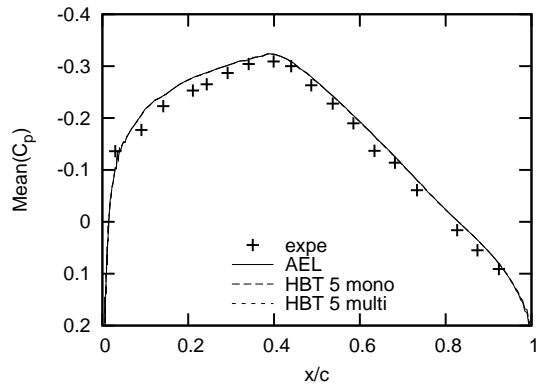


(c) Real part

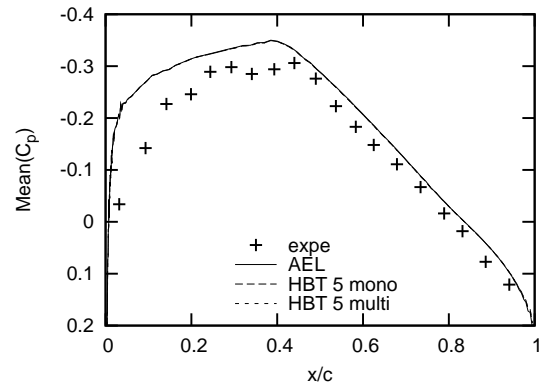


(d) Imaginary part

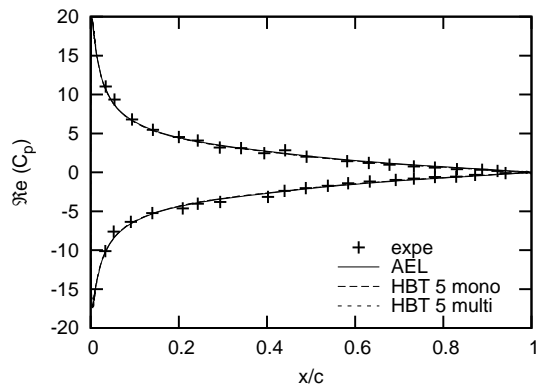
Figure 6.2: NACA 64 CT2: Pressure coefficient C_p for $N = 1$.



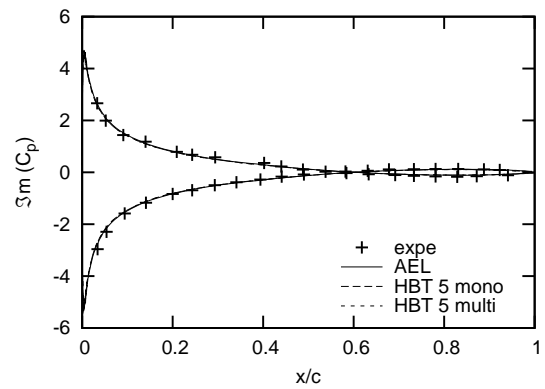
(a) Suction side



(b) Pressure side



(c) Real part



(d) Imaginary part

Figure 6.3: NACA 64 CT2: Pressure coefficient C_p for $N = 5$.

6.1.2 Pitching at low angle of attack in transonic flow: CT6 (DI55)

This test case considers a transonic regime at $M \approx 0.8$. Disposing of a low mean angle of attack, the case shows no boundary layer separation. Shocks appear on the suction and on the pressure side during the whole oscillation cycle. The shocks are slightly stronger on the pressure side due to a negative mean angle of attack.

Convergence

In order to achieve convergence, CFL numbers had to be decreased considerably. Convergence with $CFL = 100$ and 300 iterations could only be obtained for one frequency. For two frequencies 1000 iterations were performed and the CFL number had to be decreased to 15. For higher orders, CFL was again reduced by one order of magnitude. The graphs of the residuals are shown in Fig. 6.4. Convergence behaviour of the Multifrequential HBT is again equal to the one of the Monofrequential HBT.

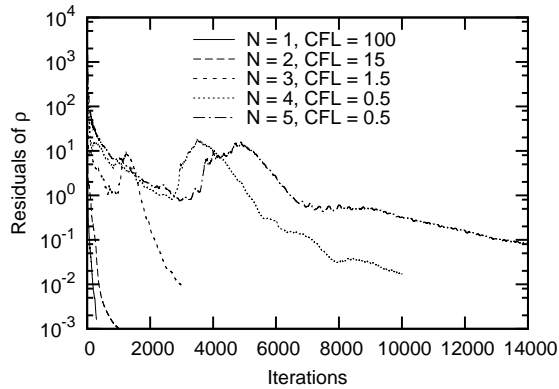
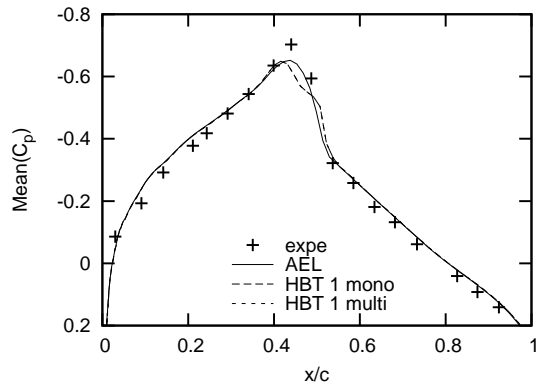


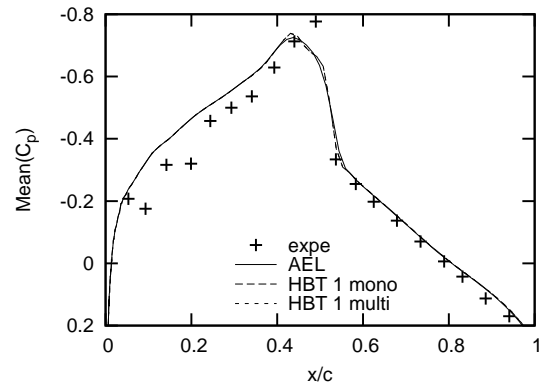
Figure 6.4: *NACA 64 CT6: Convergence.*

Results

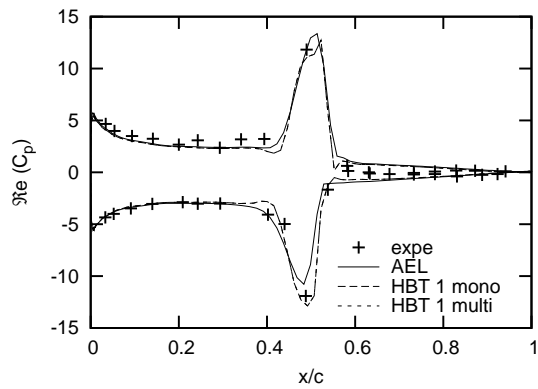
Again, only two out of the total of five computations are presented here. Fig. 6.5 depicts the obtained graph of the pressure coefficient for $N = 1$. Although, the two different HBT computations show a deviation to the AEL graph, the monofrequential and multifrequential graphs are perfectly superposed. Deviations are particularly noticeable on the real and imaginary parts of the pressure coefficient (cf. Fig. 6.5(c) - 6.5(d)). By elevating the order of the HBT computation, the oscillations (cf. Fig. 6.5(a): $0.4 \leq x/c \leq 0.55$) and overshoots (cf. Fig. 6.5(c) - 6.5(d)) diminish. This can be seen in Fig. 6.6 for $N = 4$. The graphs of the monofrequential and of the multifrequential approach are coincident.



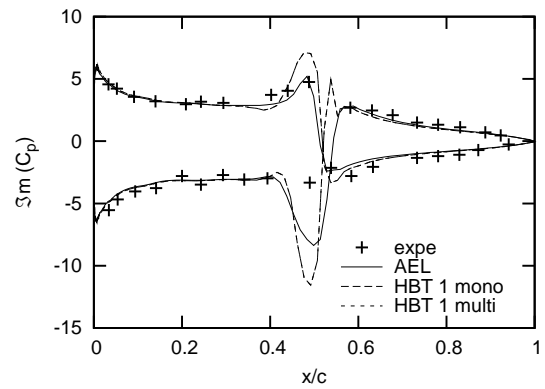
(a) Suction side



(b) Pressure side

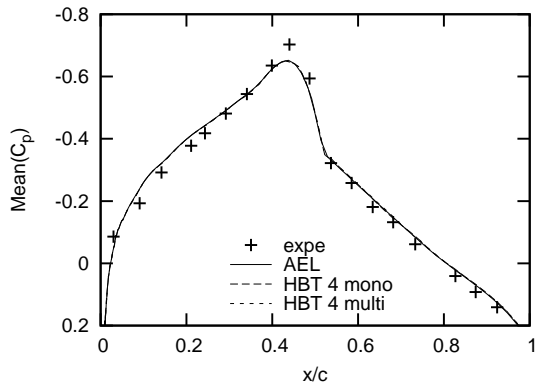


(c) Real part

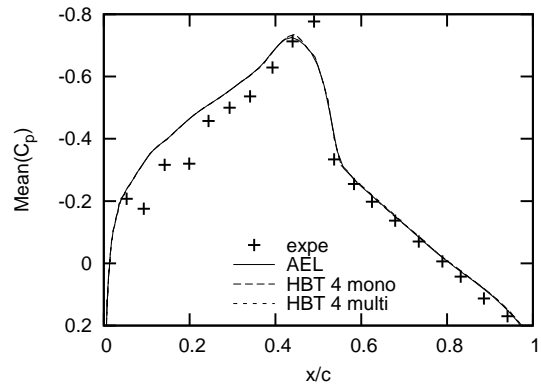


(d) Imaginary part

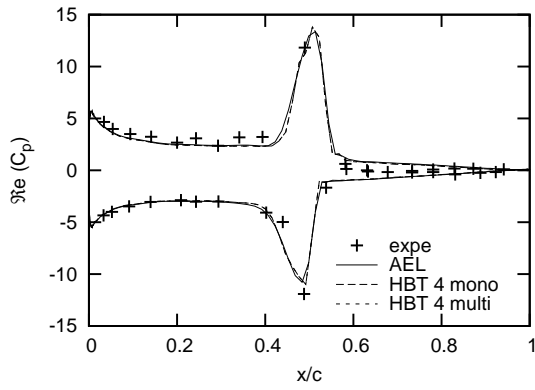
Figure 6.5: NACA 64 CT6: Pressure coefficient C_p for $N = 1$.



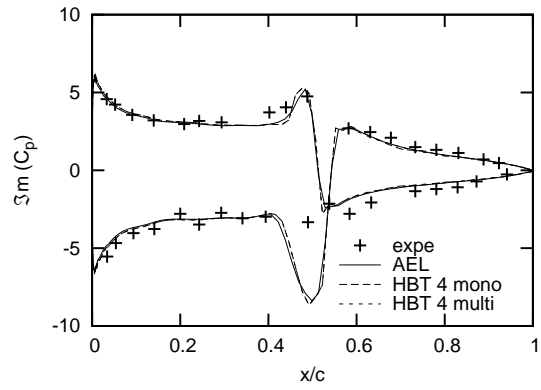
(a) Suction side



(b) Pressure side



(c) Real part



(d) Imaginary part

Figure 6.6: NACA 64 CT6: Pressure coefficient C_p for $N = 4$.

6.1.3 Pitching at high angle of attack: Shock Stall case (DI89)

This test case deals with an elevated mean angle of attack ($\alpha_0 = 4.0^\circ$). This is why a shock only appears on the suction side, while the flow on the pressure side remains subsonic. Originally, this case was thought to show the efficiency of the HBT in both subsonic and supersonic regime. It serves here as the last two-dimensional test case to prove equivalence of the mono- and multifrequential approaches. The relatively high incidence also provokes a separation of the boundary layer which will trouble the turbulence model.

Convergence

As one can state from Fig. 6.7, convergence becomes more difficult to obtain in the Shock Stall case. Considerably more iterations are needed and the *CFL* numbers are reduced to even lower values for the HBT computations. However, the convergence behaviour of the Multifrequential HBT is again essentially the same compared to the Monofrequential HBT.

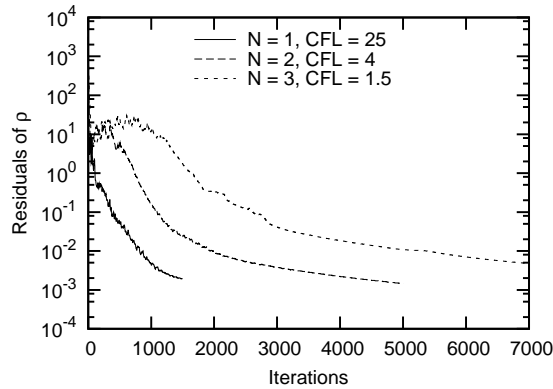
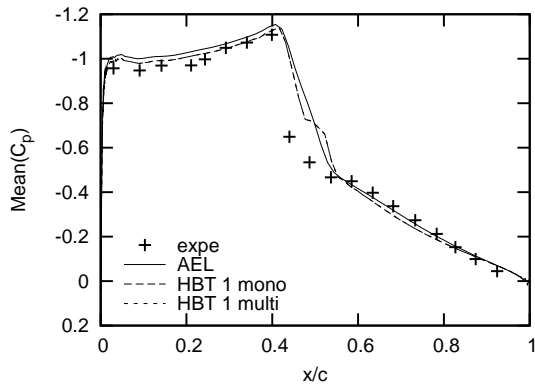


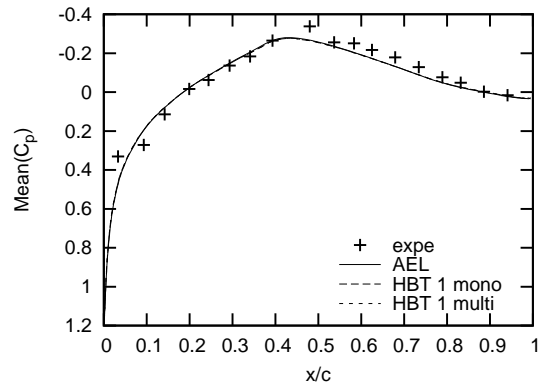
Figure 6.7: *NACA 64 Shock Stall case: Convergence.*

Results

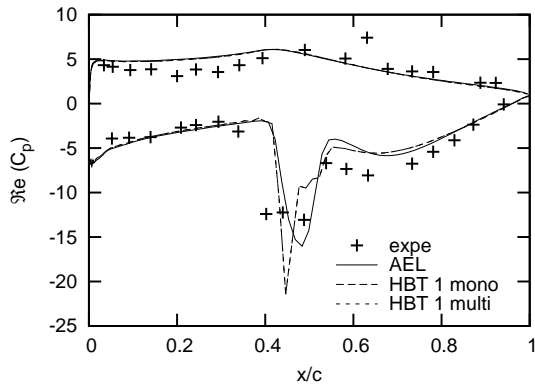
The results of the computations with one (cf. Fig. 6.8) and with three (cf. Fig. 6.9) frequencies are presented here. The step appearing on the suction side in Fig. 6.8(a) can be diminished, when a higher order is used (cf. Fig. 6.9(a)). The graph of the mean pressure coefficient shows very good accordance to the AEL graph on the pressure side (cf. Fig. 6.8(b)) for $N = 1$, but disposes of a small deviation for $N = 3$ (cf. Fig. 6.9(b)). For the real and imaginary parts, good results are obtained on the pressure side. Big differences appear though on the suction side at the location of the shock and higher order computations do not improve this situation (cf. Fig. 6.8(c) - 6.8(d) and Fig. 6.9(c) - 6.9(d)). However, the crucial observation to be made is the complete accordance between the Mono- and Multifrequential HBT.



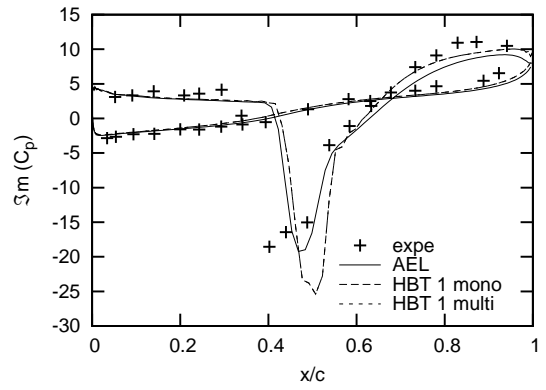
(a) Suction side



(b) Pressure side

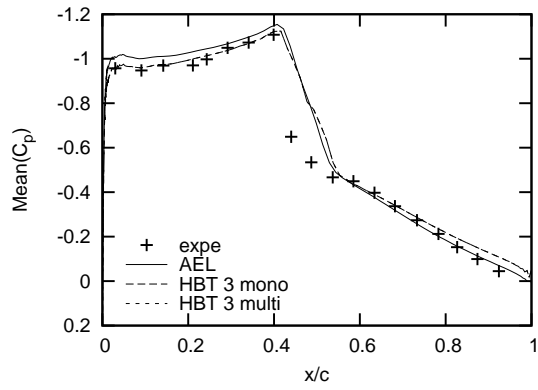


(c) Real part

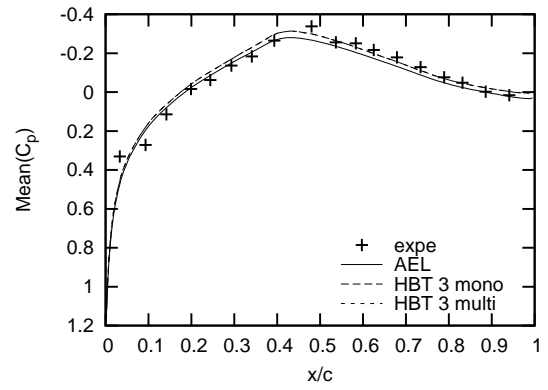


(d) Imaginary part

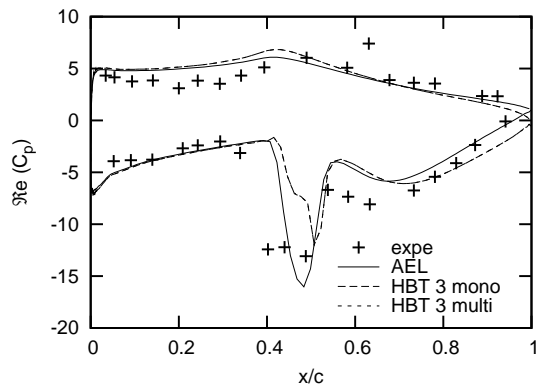
Figure 6.8: NACA 64 Shock Stall case: Pressure coefficient C_p for $N = 1$.



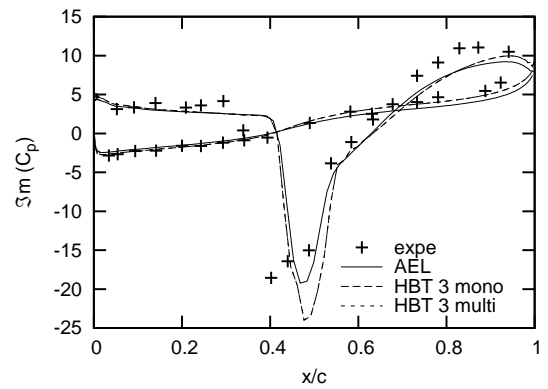
(a) Suction side



(b) Pressure side



(c) Real part



(d) Imaginary part

Figure 6.9: NACA 64 Shock Stall case: Pressure coefficient C_p for $N = 3$.

6.2 Forced motion of a 3D airfoil

The LANN wing is made of super-critical airfoils (cf. Fig. 6.10). Experimental data was furnished by a collaboration of several research centres (Lockheed-Georgia, Air Force Flight Dynamic Laboratory, NASA Langley and NLR: LANN, cf. [19]) in order to evaluate CFD software. The geometry data is given in Tab. 6.2. Six wing sections had been instrumented to allow pressure measurement (cf. Fig. 6.11).

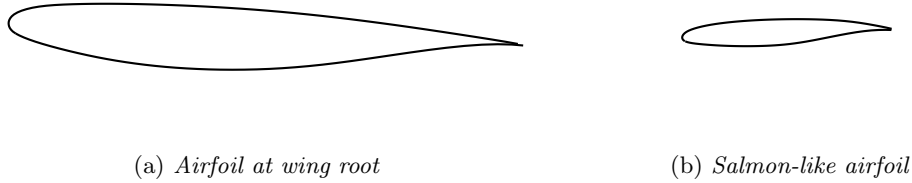


Figure 6.10: *Different super-critic airfoils of the LANN wing at the same scale.*

Table 6.2: *Geometry of the LANN wing.*

aspect ratio	7.92
sweep	25.0°
sweep at leading edge	27.493°
sweep at trailing edge	16.908°
wing torsion	-4.8°
chord at wing root	360.8mm
wing span	1000.0mm
wing surface	0.2526m ²

As mentioned before, two test cases were considered (cf. Tab. 6.3). The same variation of the angle of attack described by Eq. (6.1) is applied here. CT5 is at low angle of attack and there is no boundary layer separation. CT9 induces a separation due to the high mean angle of attack.

Table 6.3: *Parameters of the LANN test cases.*

	$\alpha_0(deg)$	$\alpha_m(deg)$	$Re(\times 10^6)$	$f(Hz)$	$\bar{\omega}$	M_∞
CT5	0.6	0.25	5.43	24	0.102	0,822
CT9	2.6	0.25	5.32	24	0.103	0.822

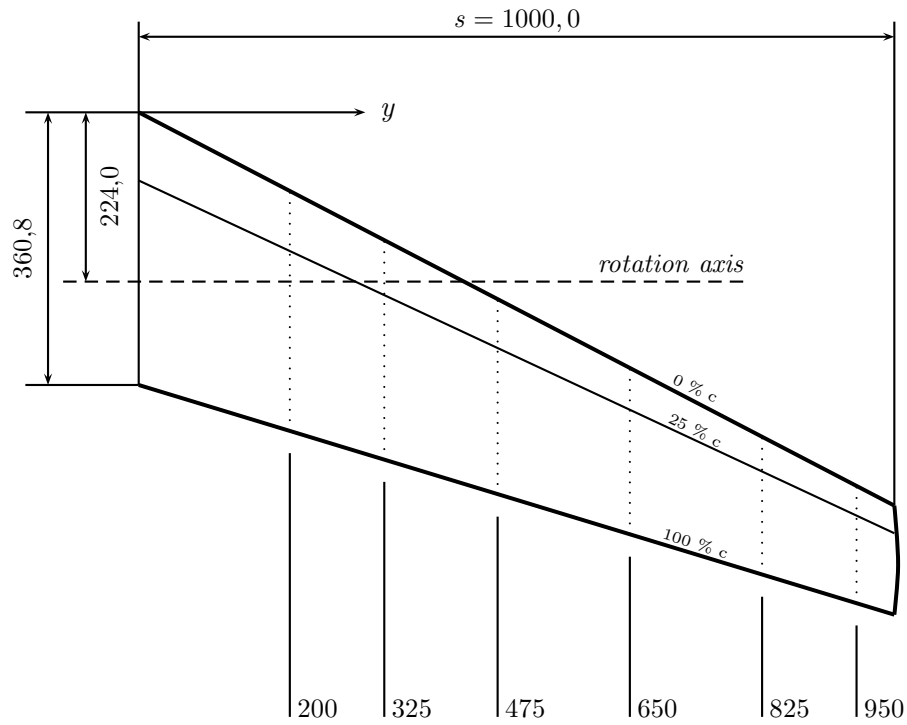


Figure 6.11: *Top view of the LANN wing (dimensions in mm).*

6.2.1 Pitching at low angle of attack: CT5

Five tests were run for this first three dimensional test case at low angle of attack including up to five frequencies.

Convergence

Fig. 6.12 illustrates convergence behaviour of the performed computations:

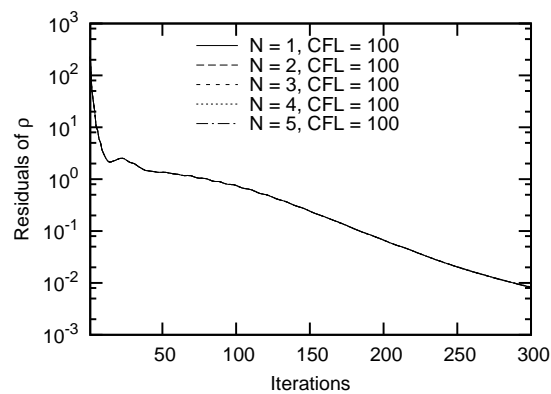


Figure 6.12: *LANN CT5: Convergence.*

The LANN wing does not encounter the same problems of convergence as the NACA airfoil does. All computations were carried out with $CFL = 100$ up to five harmonics with only 300 iterations. Convergence behaviour is the same for each test run, regardless of the number of harmonics taken in account. It is also equal to the monofrequential HBT convergence graph.

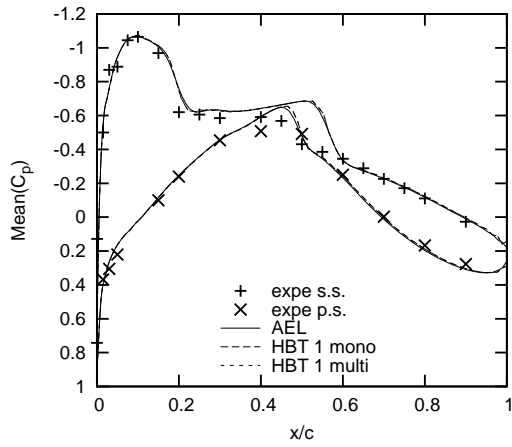
Results

Results of the mean C_p , the real and the imaginary part of the pressure coefficient are given for one frequency in Fig. 6.13, 6.14 and 6.15 for each section of the wing. Fig. 6.16, 6.17 and 6.18 display the results for five harmonics. The labeling 'expe s.s.' and 'expe p.s.' denote the experimental results on the suction side and on the pressure side respectively.

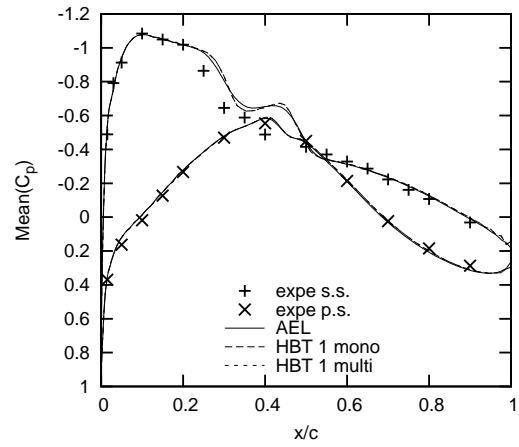
The mean pressure coefficient is in good agreement with the experiments and particularly with the AEL computations. Due to the appearing shocks on the suction side, the results of the HBT differ slightly from AEL results. Compare Fig. 6.13(c), 6.13(d) and 6.13(e) to Fig. 6.16(c), 6.16(d) and 6.16(e) respectively. Deviations can be reduced by elevating in order (actually already two frequencies are enough to do so). The small overshoot on the pressure side of section 6 (6.13(f) and 6.16(f)) cannot be reduced, though.

The real part of the coefficient of pressure is already less well approximated, although position and amplitude of the peaks of the HBT are quite good. Using higher orders, a better approach towards the AEL results is possible (see e.g. section 3 Fig. 6.14(c), 6.17(c) and section 5 Fig. 6.14(e), 6.17(e) and compare Fig. 6.14(b) and Fig. 6.17(b)). Similar conclusion are drawn from the imaginary part of C_p . Again, a better approximation of the peaks' position and amplitude is obtained for higher order computation.

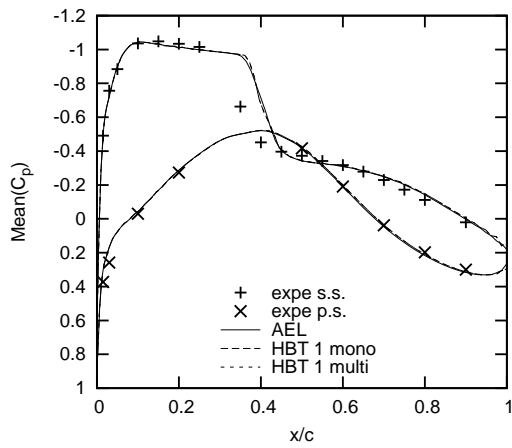
Nevertheless, the key conclusion to draw is, that monofrequential and multifrequential results are superposed for all frequencies tested, on all sections of the wing for the mean pressure coefficient, its real and its imaginary part.



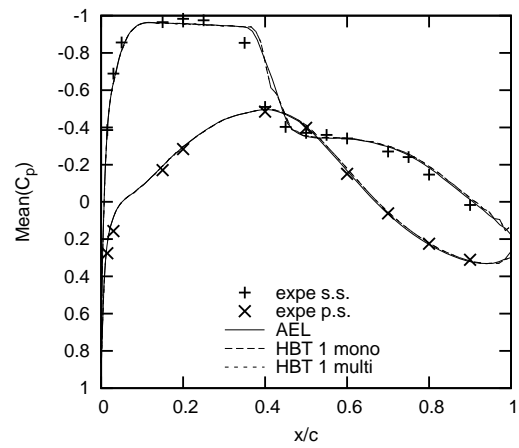
(a) Section 1 - 20%



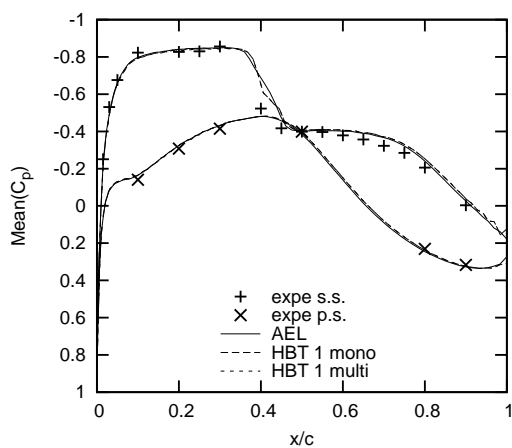
(b) Section 2 - 32.5%



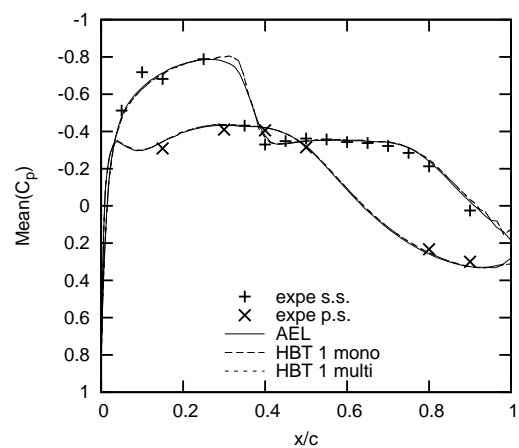
(c) Section 3 - 47.5%



(d) Section 4 - 65%

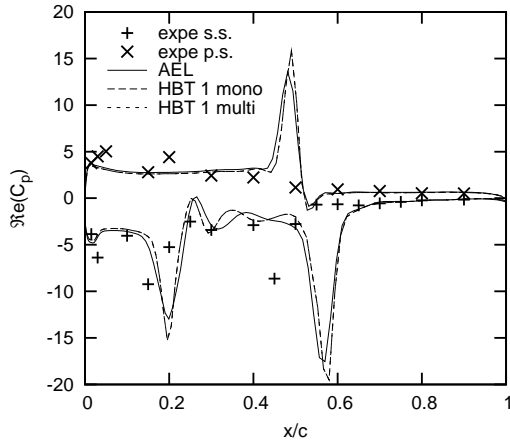


(e) Section 5 - 82.5%

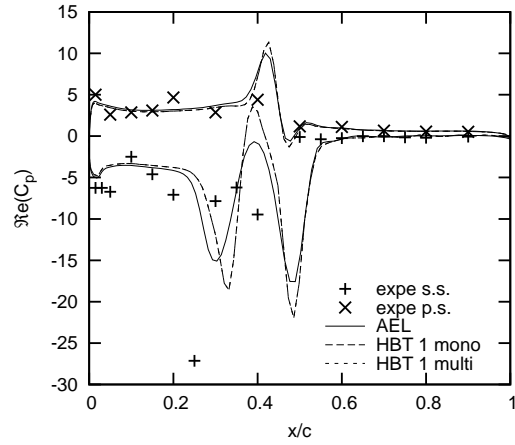


(f) Section 6 - 95%

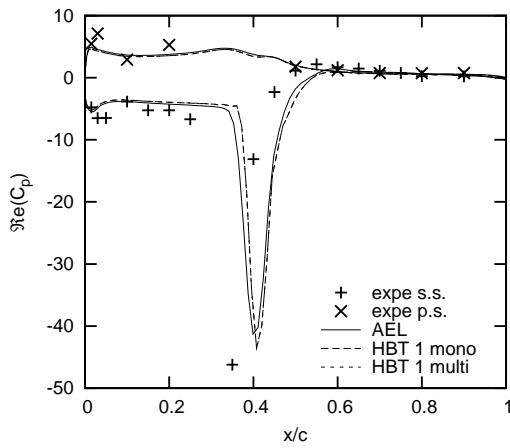
Figure 6.13: LANN CT5: Mean pressure coefficient C_p for $N = 1$.



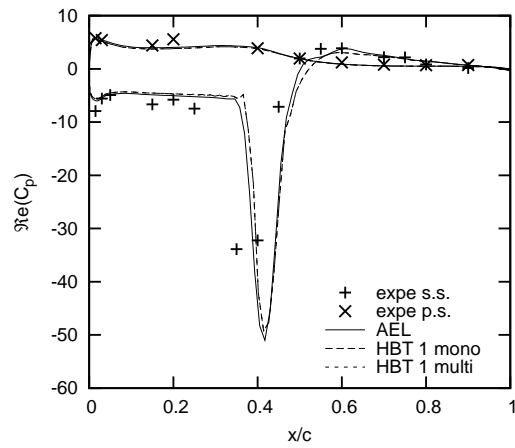
(a) Section 1 - 20%



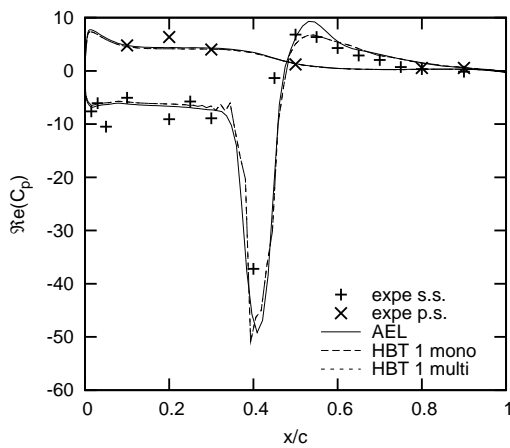
(b) Section 2 - 32.5%



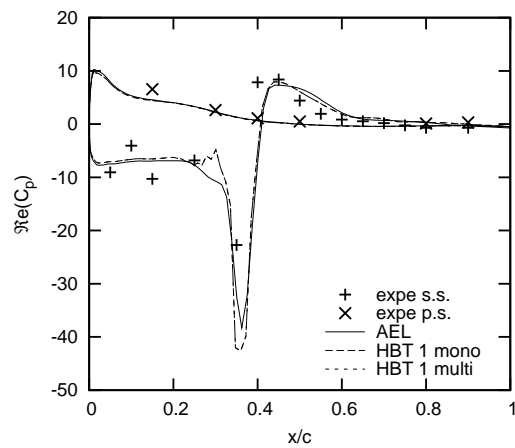
(c) Section 3 - 47.5%



(d) Section 4 - 65%

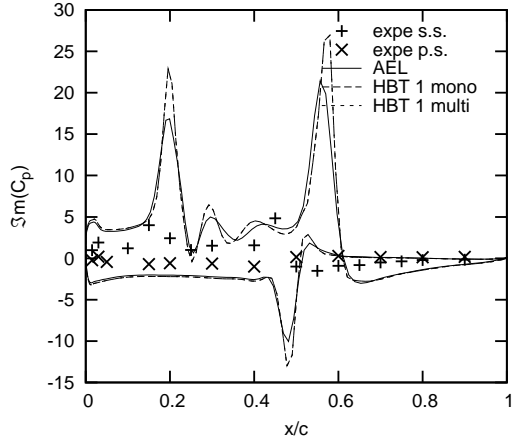


(e) Section 5 - 82.5%

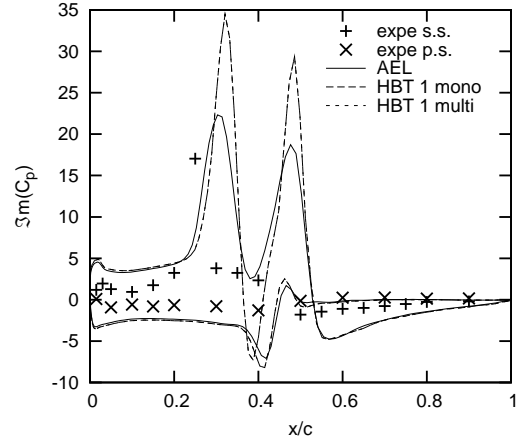


(f) Section 6 - 95%

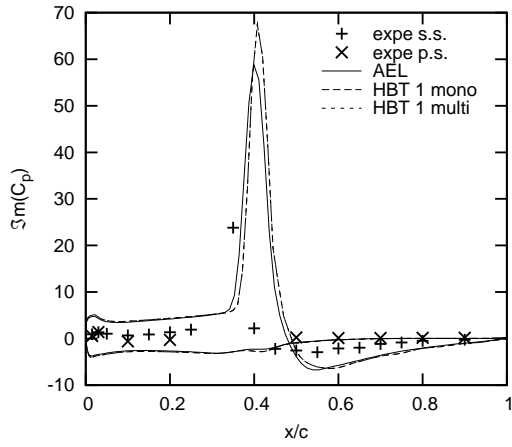
Figure 6.14: LANN CT5: Real part of C_p for $N = 1$.



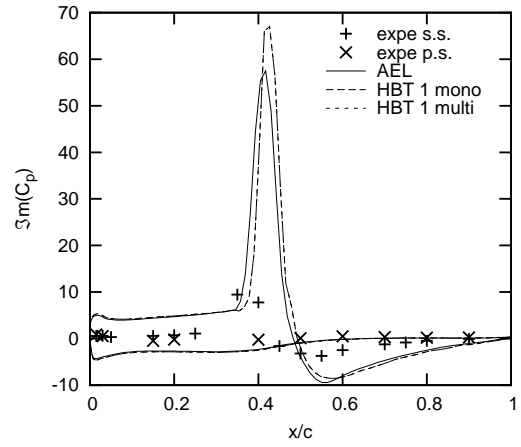
(a) Section 1 - 20%



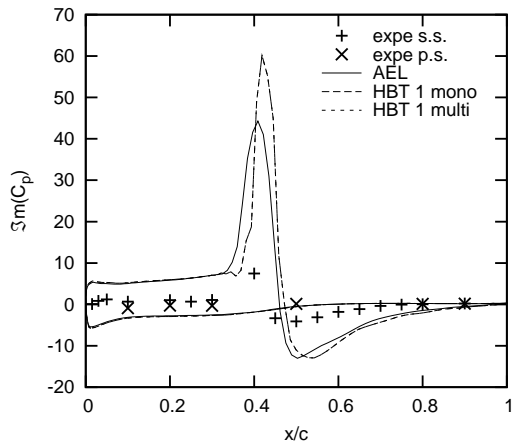
(b) Section 2 - 32.5%



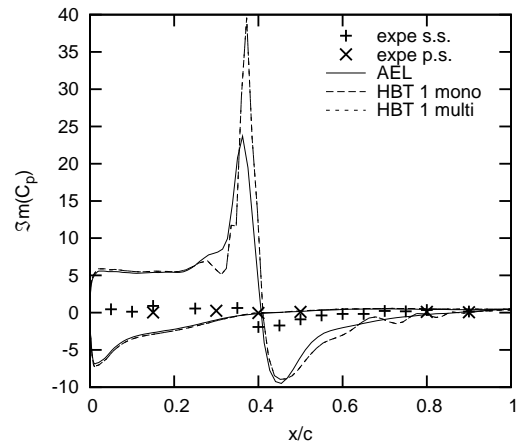
(c) Section 3 - 47.5%



(d) Section 4 - 65%

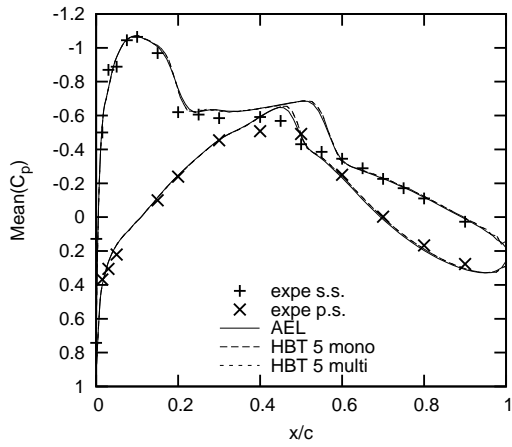


(e) Section 5 - 82.5%

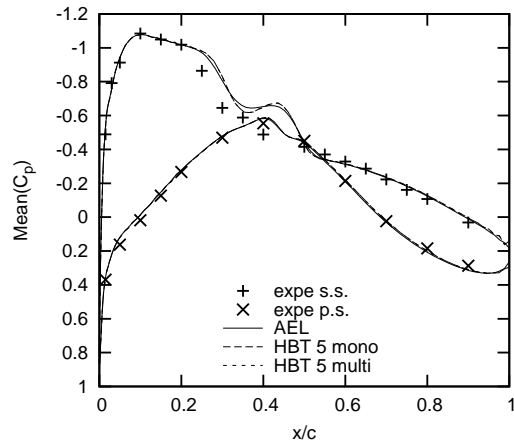


(f) Section 6 - 95%

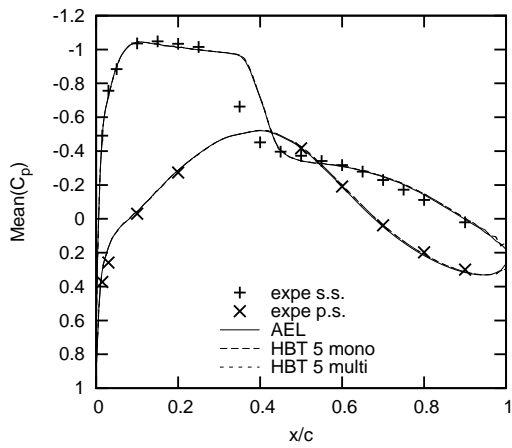
Figure 6.15: *LANN CT5*: Imaginary part of C_p for $N = 1$.



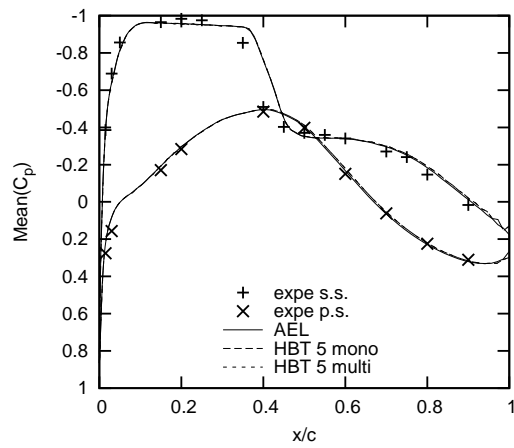
(a) Section 1 - 20%



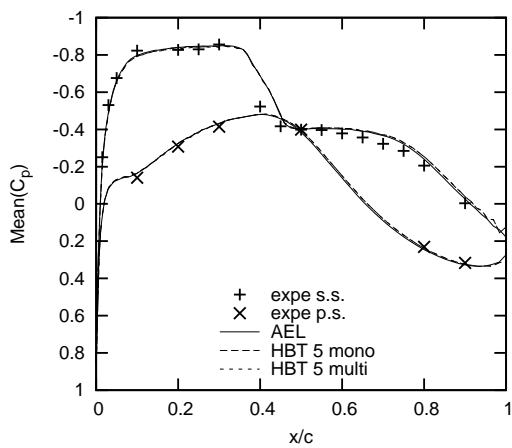
(b) Section 2 - 32.5%



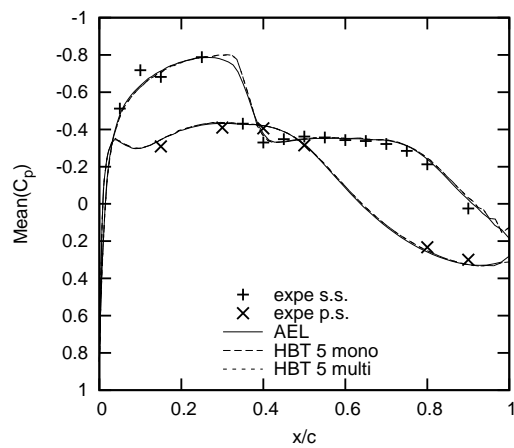
(c) Section 3 - 47.5%



(d) Section 4 - 65%

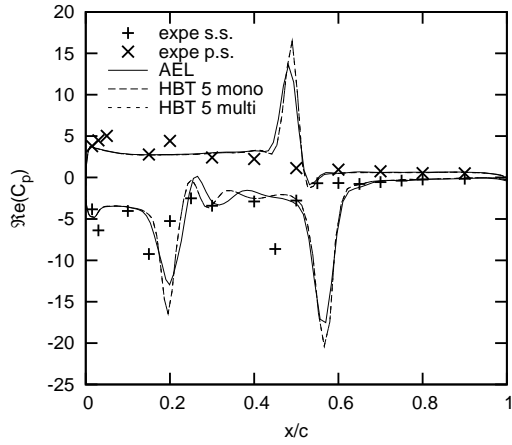


(e) Section 5 - 82.5%

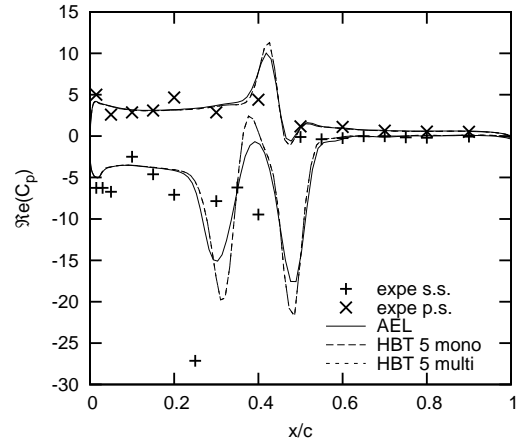


(f) Section 6 - 95%

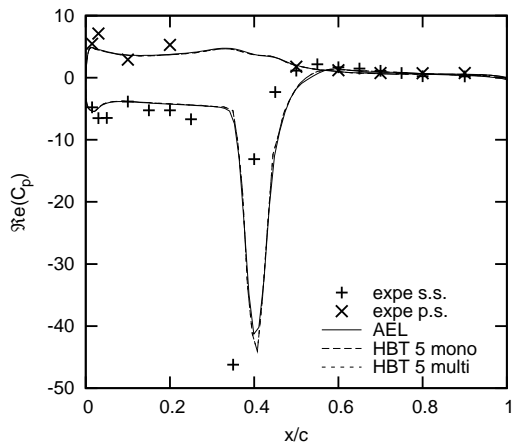
Figure 6.16: *LANN CT5*: Mean pressure coefficient C_p for $N = 5$.



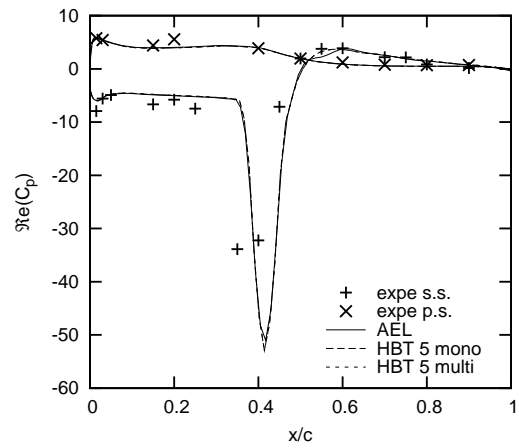
(a) Section 1 - 20%



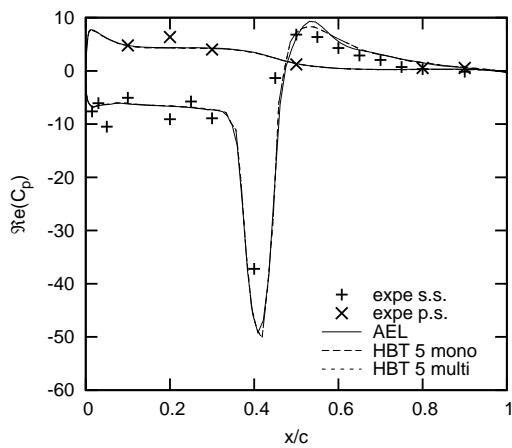
(b) Section 2 - 32.5%



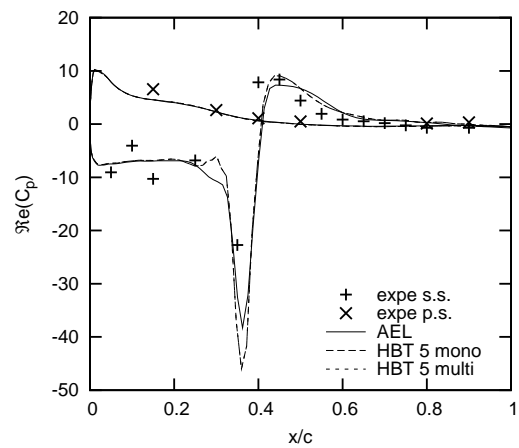
(c) Section 3 - 47.5%



(d) Section 4 - 65%

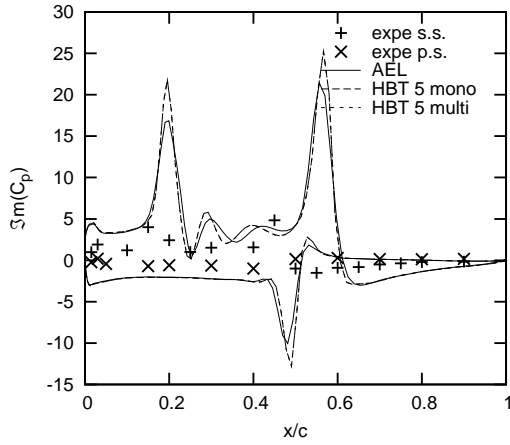


(e) Section 5 - 82.5%

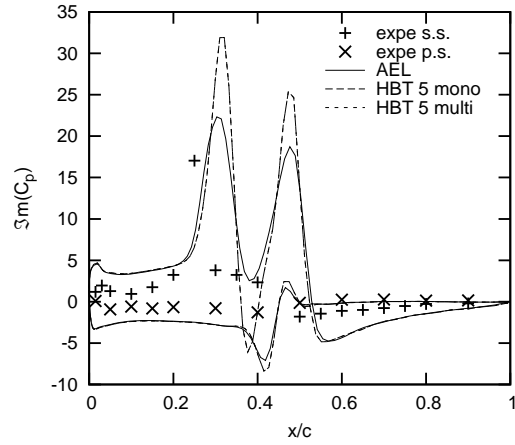


(f) Section 6 - 95%

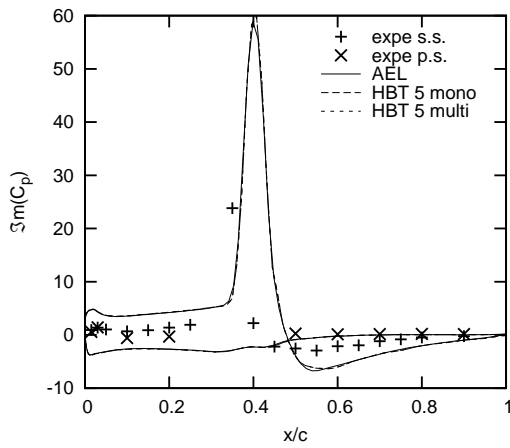
Figure 6.17: LANN CT5: Real part of C_p for $N = 5$.



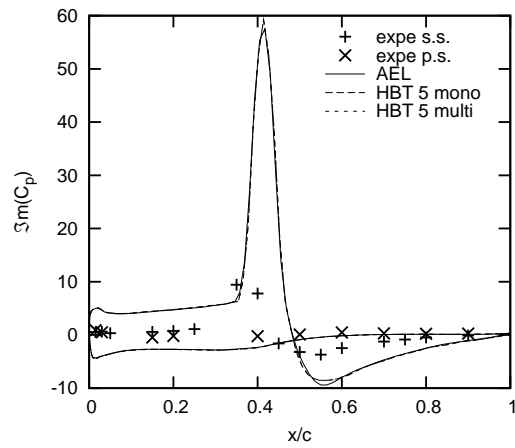
(a) Section 1 - 20%



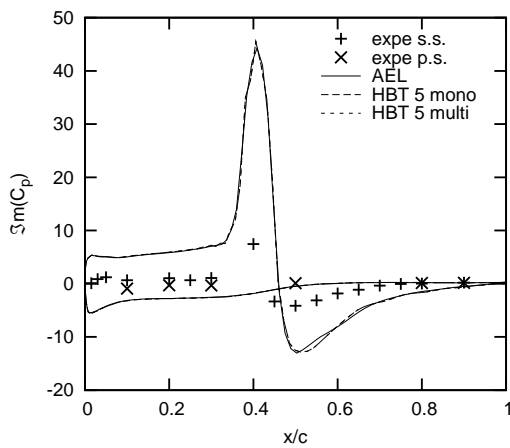
(b) Section 2 - 32.5%



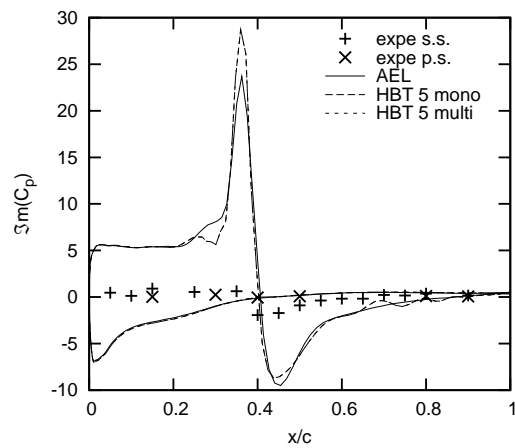
(c) Section 3 - 47.5%



(d) Section 4 - 65%



(e) Section 5 - 82.5%



(f) Section 6 - 95%

Figure 6.18: LANN CT5: Imaginary part of C_p for $N = 5$.

6.2.2 Pitching at high angle of attack: CT9

The present test case will be the last to validate the Multifrequential HBT. The frequency of the periodic oscillation of the wing remains the same as for the previous test case. It deals however with a high mean angle of attack.

Convergence

Just like test case CT5, convergence behaviour is alike for all five computations. Again a *CFL* number of 100 was sufficient. The number of iterations is raised to 600 in order to achieve orders of magnitude of the residuals similar to test case CT5.

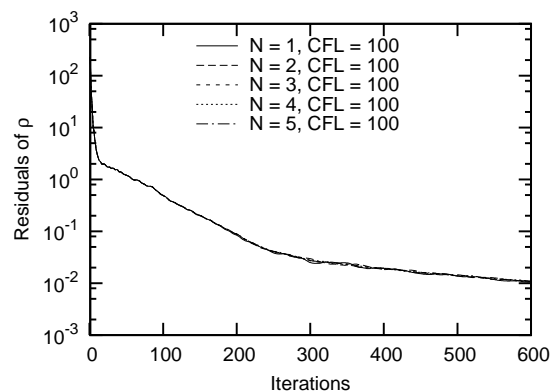
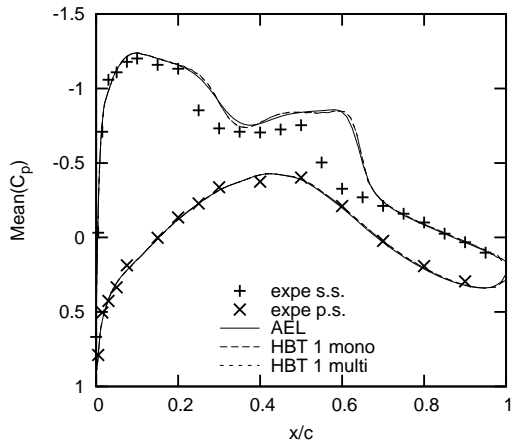


Figure 6.19: *LANN CT9: Convergence.*

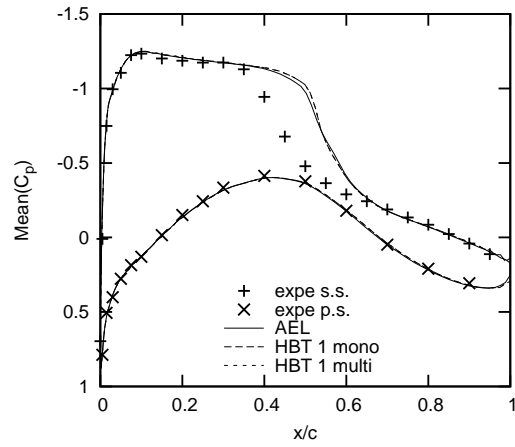
Results

The boundary layer separation is predicted quite badly by the turbulence model as it can be seen in Fig. 6.20(a) - 6.20(c). On these three sections, the shock is dislocated towards the trailing edge. There is good accordance of the mean pressure coefficient between the AEL and HBT computations, though. A step appears on section 4 at the location of the shock (cf. Fig. 6.20(d)). However, this step is smoothed and finally disappears, the more harmonics are taken in account (cf. Fig. 6.23(d)). As for the real part of the pressure coefficient (cf. Fig. 6.21 and 6.24), there are strong differences on section 1 between the AEL and HBT computation. On section 3 - 5, the amplitudes of the peaks of the HBT results are considerably higher. This over prediction is not reduced noticeably for higher orders, but these graphs are closer to the experimental data than the AEL results. This becomes apparent also, when Fig. 6.21(f) is examined: On this section the peak's orientation is well predicted by the HBT. The AEL result predicts the inverse orientation. The imaginary part shows good accordance to the AEL results, while section 1 is the furthest away from the AEL computation, but closer to the experimental data.

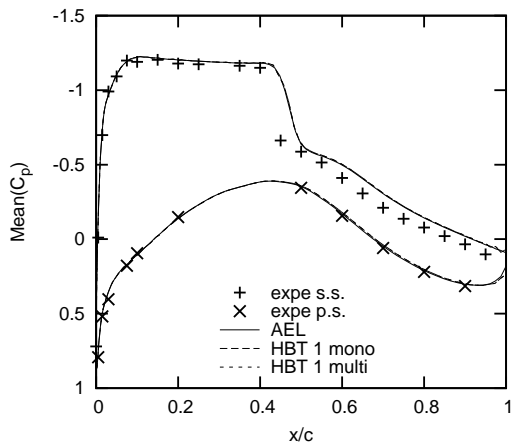
Mono- and multifrequential results are superposed for all frequencies tested. No differences are perceptible on any section of the wing.



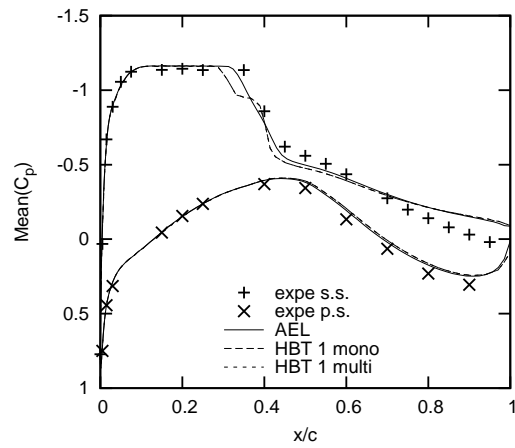
(a) Section 1 - 20%



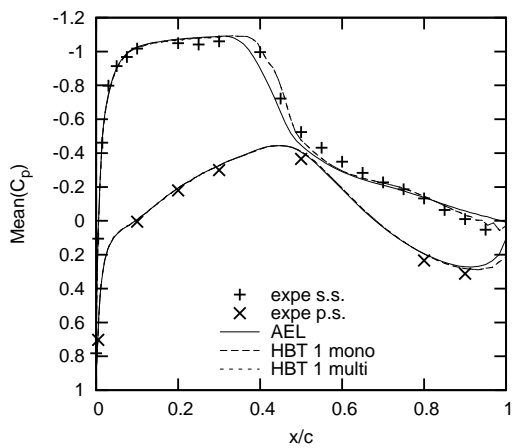
(b) Section 2 - 32.5%



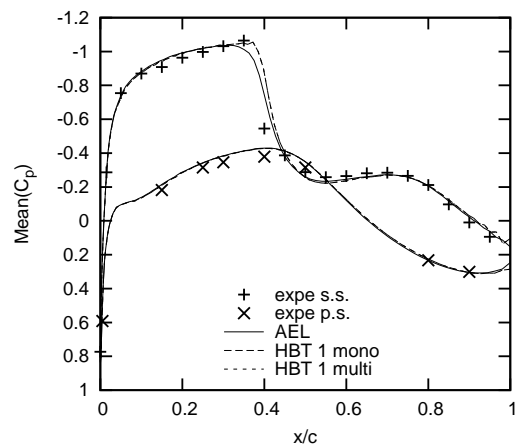
(c) Section 3 - 47.5%



(d) Section 4 - 65%

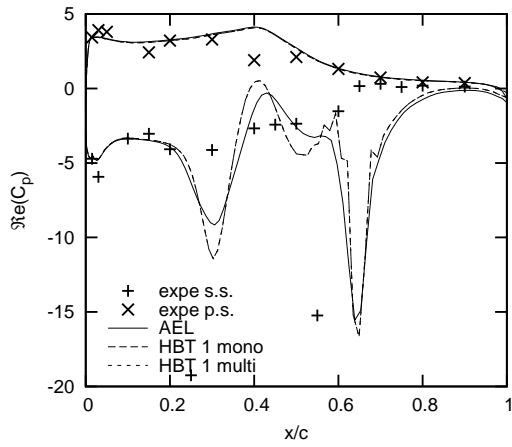


(e) Section 5 - 82.5%

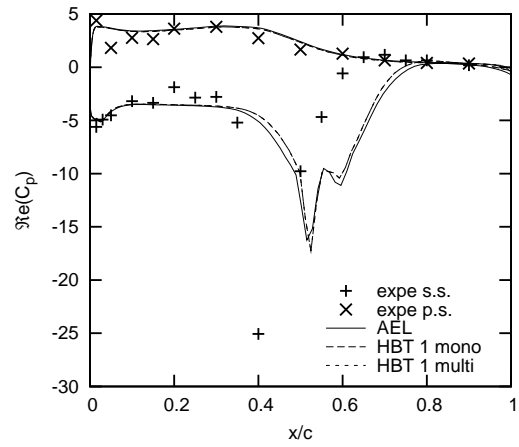


(f) Section 6 - 95%

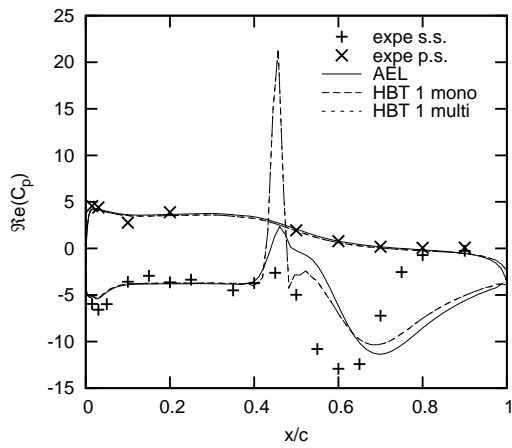
Figure 6.20: LANN CT9: Mean pressure coefficient C_p for $N = 1$.



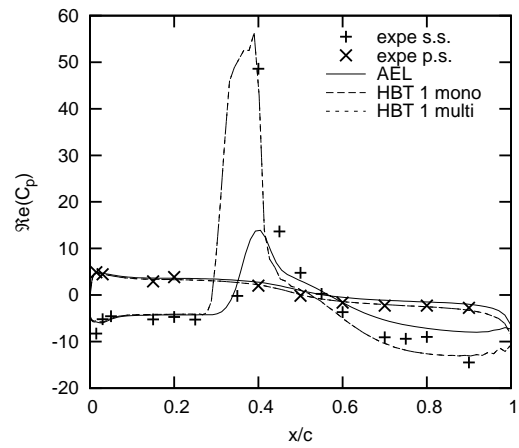
(a) Section 1 - 20%



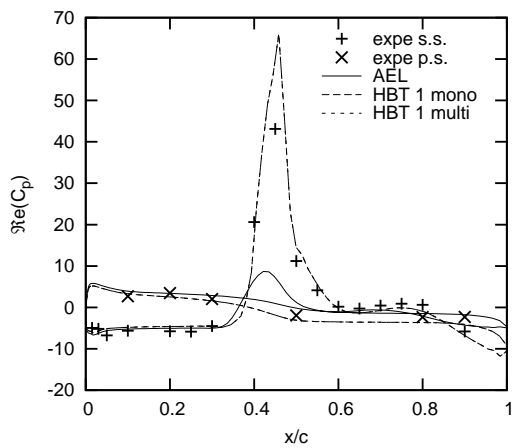
(b) Section 2 - 32.5%



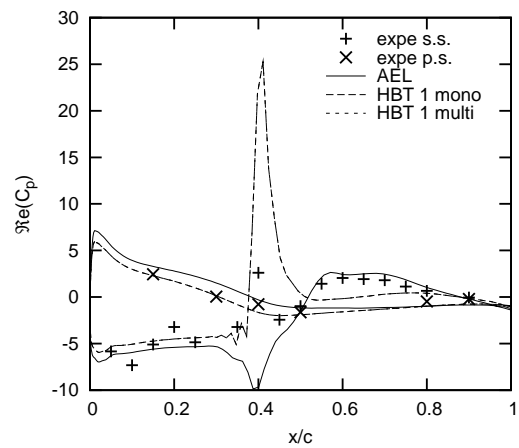
(c) Section 3 - 47.5%



(d) Section 4 - 65%

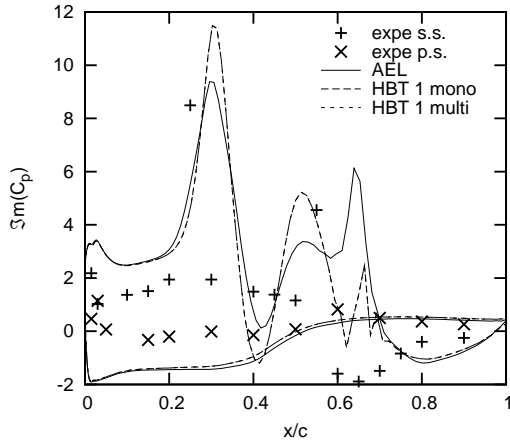


(e) Section 5 - 82.5%

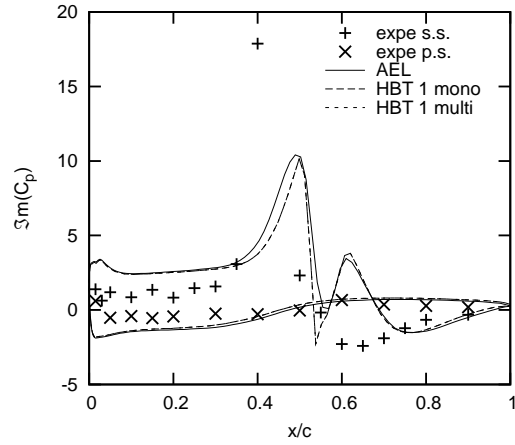


(f) Section 6 - 95%

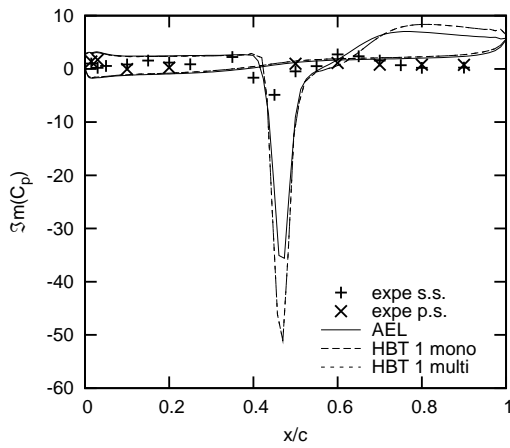
Figure 6.21: LANN CT9: Real part of C_p for $N = 1$.



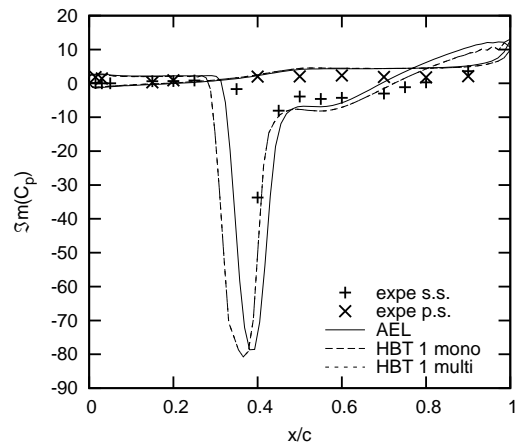
(a) Section 1 - 20%



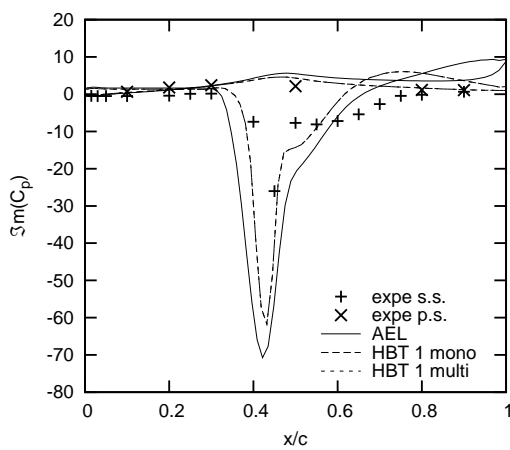
(b) Section 2 - 32.5%



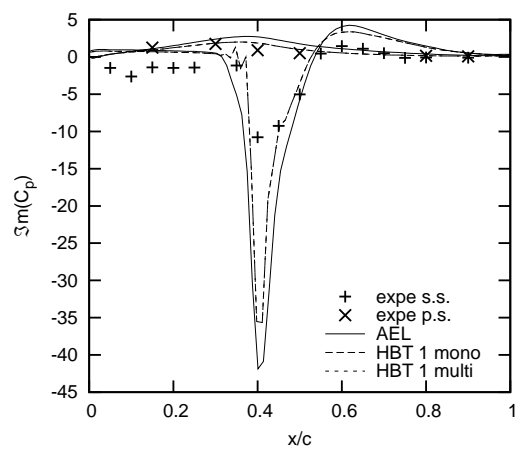
(c) Section 3 - 47.5%



(d) Section 4 - 65%

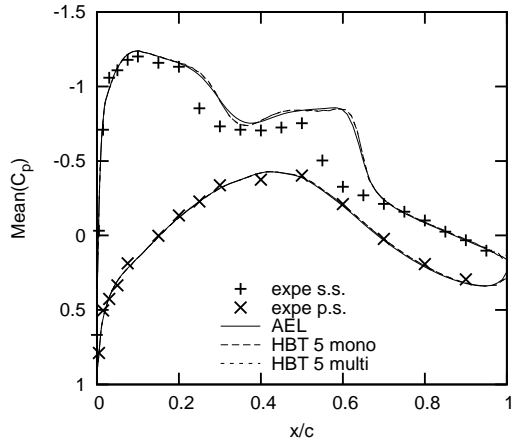


(e) Section 5 - 82.5%

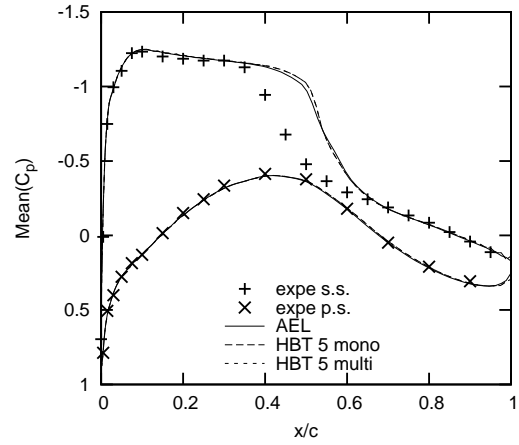


(f) Section 6 - 95%

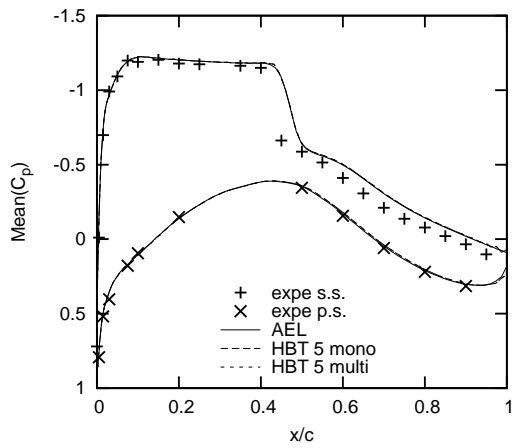
Figure 6.22: LANN CT9: Imaginary part of C_p for $N = 1$.



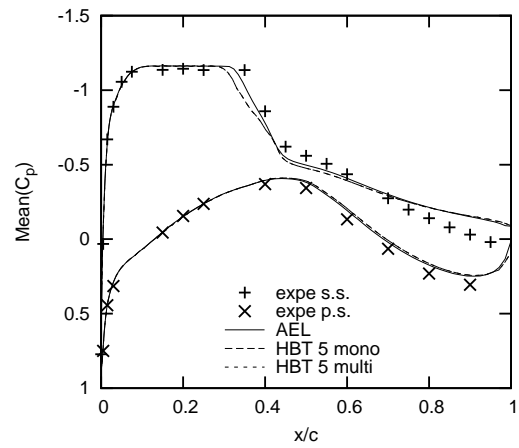
(a) Section 1 - 20%



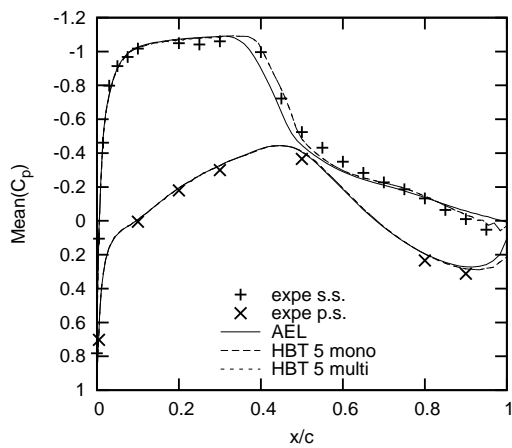
(b) Section 2 - 32.5%



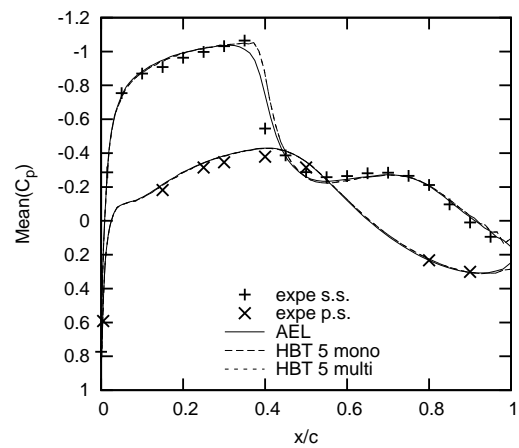
(c) Section 3 - 47.5%



(d) Section 4 - 65%

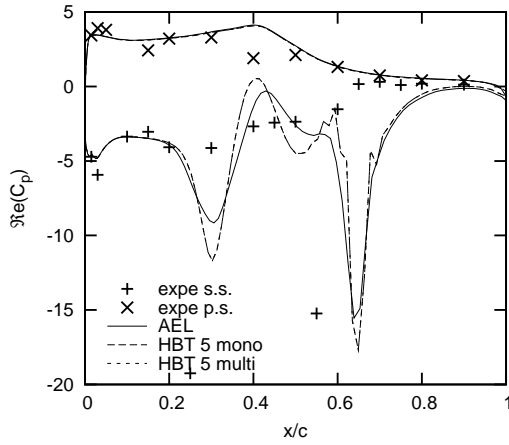


(e) Section 5 - 82.5%

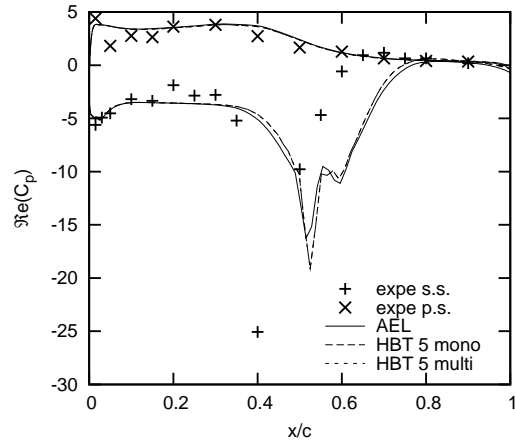


(f) Section 6 - 95%

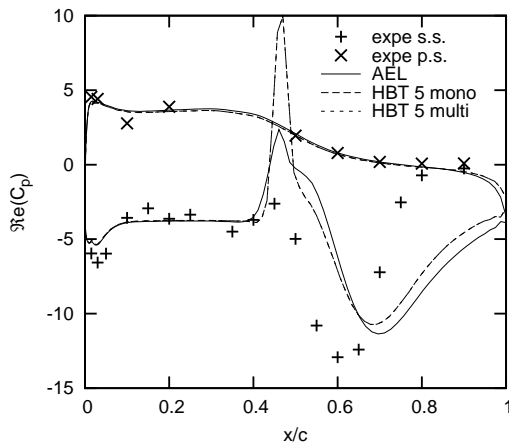
Figure 6.23: LANN CT9: Mean pressure coefficient C_p for $N = 5$.



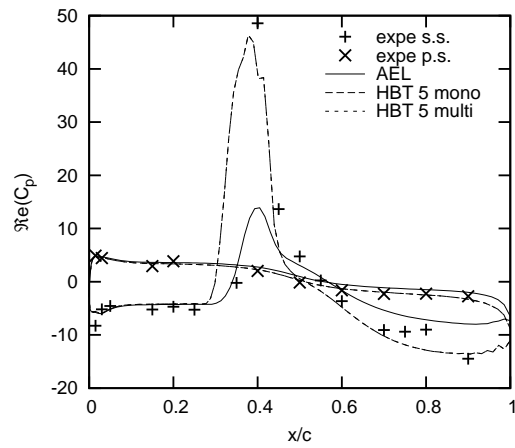
(a) Section 1 - 20%



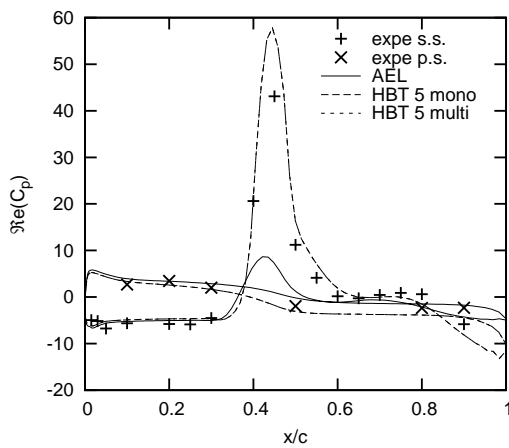
(b) Section 2 - 32.5%



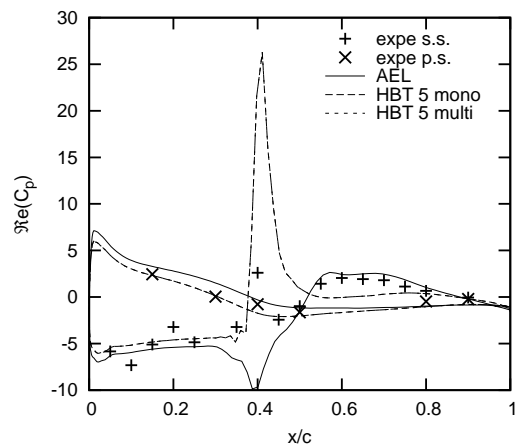
(c) Section 3 - 47.5%



(d) Section 4 - 65%

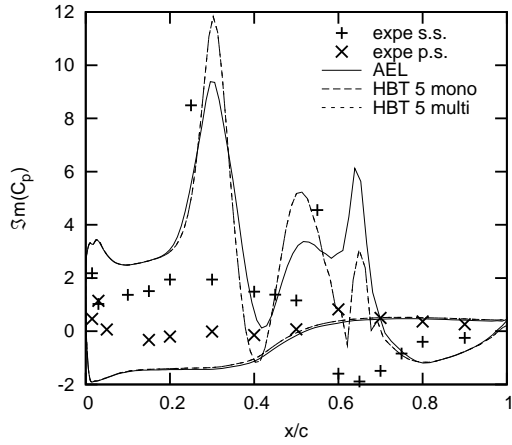


(e) Section 5 - 82.5%

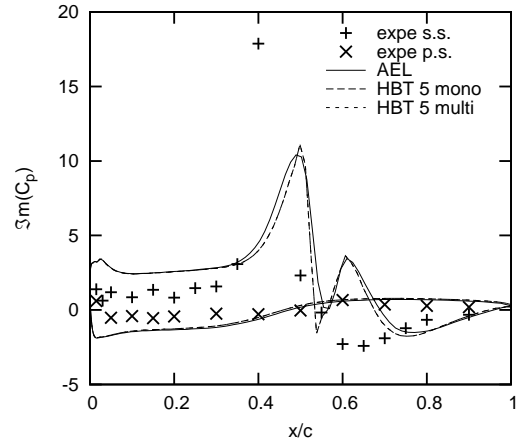


(f) Section 6 - 95%

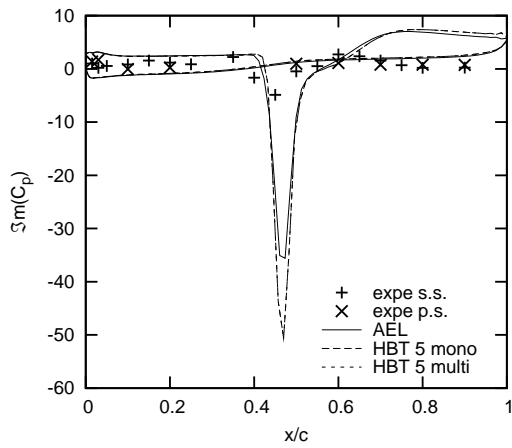
Figure 6.24: LANN CT9: Real part of C_p for $N = 5$.



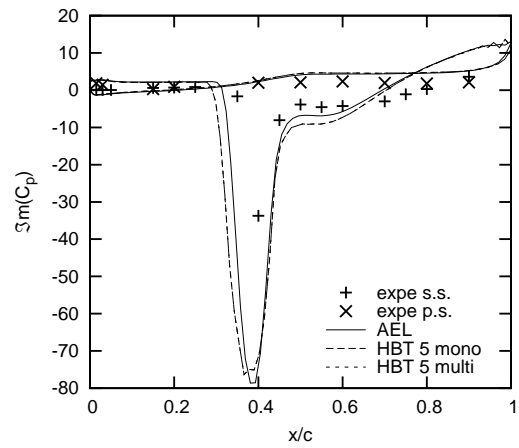
(a) Section 1 - 20%



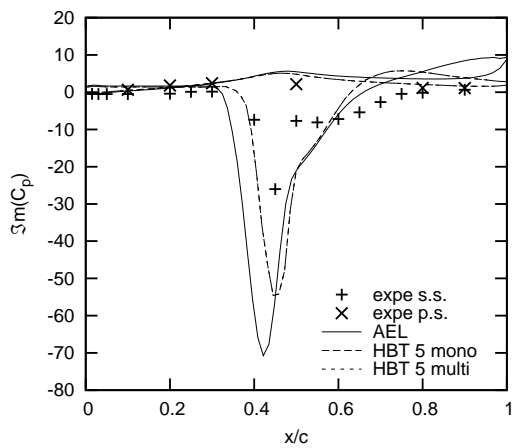
(b) Section 2 - 32.5%



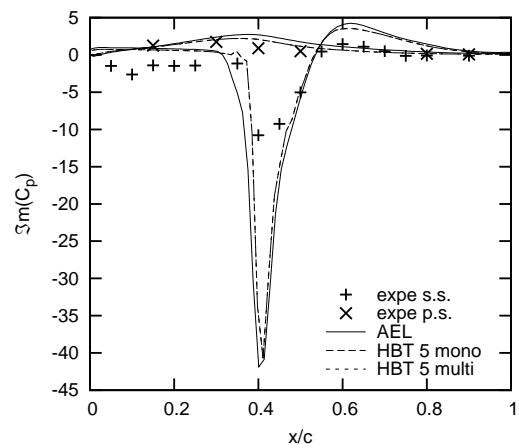
(c) Section 3 - 47.5%



(d) Section 4 - 65%



(e) Section 5 - 82.5%



(f) Section 6 - 95%

Figure 6.25: LANN CT9: Imaginary part of C_p for $N = 5$.

6.3 Precision discrepancy of D-coefficients

There is a discrepancy between the D-coefficients of the monofrequential approach calculated by Eq. (2.14) and those calculated for the multifrequential approach, although they should theoretically be the same, when the same frequencies and instants are used. One must take into consideration though: For the multifrequential approach the inverse of matrix A_1 must be found (and is calculated numerically) and has to be multiplied by the derivative of matrix A_1 (cf. Eq. (2.33)). There is certainly a loss in precision, when inverting the matrix even though all computation is done with double precision in *elsA*.

A first additional computation was performed for test case CT9 with two harmonics and 600 iterations using the multifrequential approach, but imposing in FORTRAN subroutine *souhbtmultifreqd* the precise coefficients of the monofrequential approach. A comparison to the Monofrequential HBT code without any modifications by the author is given in Tab. 6.4. The percent of deviation is referred to the monofrequential residuals.

Table 6.4: *CT9, N = 2: Comparison of residuals of the Mono- and Multifrequential HBT, when the D-coefficients are imposed in the multifrequential approach.*

L2	monofrequ.	multifrequ.	deviation [%]
ρ	1.0719779E-02	1.0719779E-02	0.00E+00
ρu	8.0258603E-03	8.0258604E-03	1.25E-06
ρv	5.1383217E-03	5.1383218E-03	1.95E-06
ρw	1.8547960E-03	1.8547962E-03	1.08E-05
ρE	4.0618512E-02	4.0618512E-02	0.00E+00
$\rho \nu$	2.8551927E-05	2.8551927E-05	0.00E+00

A very small deviation in the residual of ρu ($\Delta L2 = 1E-10$), ρv ($\Delta L2 = 1E-10$) and ρw ($\Delta L2 = 2E-10$) is detected. This effect appears not only after 600 iterations, but similar deviations appear throughout the course of iteration, vanish from time to time and reappear. It is interesting to see that this effect does not occur when monofrequential computations effected with the modified code and the original code are compared. In this case, all residuals for each and every iteration are exactly the same. These deviations are probably due to round-off errors and/or a difference in the computation of the sine- and cosine-function between FORTRAN and C++, because the monofrequential D-coefficients are computed in C++, while the multifrequential routines for the D-coefficients are included in FORTRAN. Hence, this might be a compiler problem. Given the extremely small order of magnitude of this deviation (seven orders of magnitude smaller than the residual when convergence is obtained) and considering in addition that it does not increase, but is rather damped in the course of iteration, the code is considered to be deployable. Furthermore, the two approaches furnish the same solutions, if the exact same D-coefficients are used for the computation.

The results shown in the previous sections do not show any deviation. Anyway, due to the deviation in the D-coefficients calculated by the matrix product mentioned above, a difference in residuals arises. The effect of the D-coefficients on the residuals is therefore examined. Ten computations of test case CT9 are taken in account to do so. For $N = 1, \dots, 5$ frequencies both a monofrequential and a multifrequential computation are performed. In Tab. 6.5 the maximum relative deviation

of the D-Coefficients and the resulting maximum relative deviation of the residuals are given in percent. The conservative variable that suffers this maximum deviation is also given. Tab. 6.6 contains the absolute deviation from the monofrequential computation.

Table 6.5: *CT9: Comparison of D-coefficients and resulting residuals of the mono- and multifrequential approach (percent deviation).*

Order	iterations	max. dev. d-Coeff. [%]	max. dev. residual [%]	Variable
$N = 1$	600	-6.15E-06	+1.66E-05	ρw
$N = 2$	600	-1.65E-05	+7.39E-06	ρE
$N = 3$	600	-2.90E-05	-9.26E-06	ρ
$N = 4$	600	-4.31E-05	-4.87E-06	ρE
$N = 5$	600	-5.84E-05	+9.51E-06	ρ

Table 6.6: *CT9: Comparison of D-coefficients and resulting residuals of the mono- and multifrequential approach (absolute deviation).*

Order	iterations	max. dev. d-Coeff.	max. dev. residual	Variable
$N = 1$	600	-1.1713E-08	+4.0E-09	ρE
$N = 2$	600	-3.51E-08	+3.0E-09	ρE
$N = 3$	600	+7.03E-08	-1.0E-09	ρ
$N = 4$	600	-1.17E-07	-2.0E-09	ρE
$N = 5$	600	+1.76E-07	+3.0E-09	ρE

Tab. 6.5 clearly shows that the deviations of the D-coefficients are rather small, i.e. about an order of magnitude of $-5\text{E-}05\%$. For the smallest matrix ($N = 1$, $\dim(A) = 3$), the least deviation occurs and rises constantly when the order N is elevated. The resulting deviation in the residuals is also very small, but does not seem to have a linear correlation with the deviation of the D-coefficients: The residual's relative deviation is actually quite arbitrary. The maximum deviation is $+1.66\text{E-}05\%$. The maximum absolute deviations of the residuals are of an order of magnitude of 10^{-09} after 600 iterations. These numbers show clearly that the multifrequential approach can be thought to be equivalent to the monofrequential one. Deviations do appear, but are negligible. This is particularly true, when the results of the test cases shown in the previous sections are considered. They are perfectly coincident with the monofrequential results.

6.4 Summary

The validation of the Multifrequential HBT code with the previous test cases shows very good accordance of the solutions of the mono- and multifrequential approaches. Convergence behaviour is essentially the same. The slight deviation of the residuals due to a difference in the D-coefficients does not pose a problem. The raise in computing time between a multifrequential and a monofrequential computation for the test cases described above is less than three percent ($\Delta = \frac{t_{\text{multi}} - t_{\text{mono}}}{t_{\text{mono}}} < 3\%$).

Chapter 7

Non-uniformly distributed instants

This chapter is dedicated to the effect of non-uniformly distributed instants on the results of the test cases. Since the considered test cases only dispose of one frequency, the frequencies taken in account, when higher orders are used, will still be multiples of this one base frequency. Tab. 7.1 shows the distribution i of the chosen instants for each order N . T denotes the period of the base frequency of each test case: $T = \frac{1}{f_1}$. In Tab. 7.1, the instants are given as the ratio of T and this means that the instants are not the same for two test cases. The product of frequency and instant $f_1 \cdot t_i$ will be equal however. The uniform distribution is given for the purpose of comparison (distribution $i = 0$). The determinant of matrix A_1 and the condition number of matrix D are listed in Tab. 7.2. The author recalls the definition of the condition number for regular matrices (using the L_2 -norm):

$$\kappa(D) = \|D\| \|D^{-1}\| \quad \text{with} \quad \|D\|_2 = \sqrt{\sum_{i=1}^{2N+1} \sum_{j=1}^{2N+1} |D_{i,j}|^2} \quad (7.1)$$

As far as one can see, the condition number κ of matrix D is extremely high. Orders of magnitude are mostly greater than 10^{10} (cf. Tab. 7.2). However, this is true for uniformly distributed instants as well. The condition describes the dependency of a problem's solution on the input data. The condition number is a quantity to measure this condition and specifies the factor by which an error in the input data is amplified in the worst case. Since the very high condition number does not seem to pose a problem for the monofrequential approach, no further attention will be paid towards it.

Nevertheless, the effect of a matrix A_1 with a small determinant (like distributions (3, 1), (4, 1) and (5, 1)) will be discussed in the following sections. For all other tests, the instants were chosen with the help of the program described in section 4.2.2 in a way that matrix A_1 is a regular matrix with a determinant greater than 0.1 and therefore invertible without any trouble. Unless otherwise stated, the same CFL and number of iterations are used as for the computations with uniformly distributed instants. All distributions $i = 2$ are chosen in a way that the maximum difference to the corresponding instant of the uniform distribution is less than $0.05T$. Hence, it is a distribution that should lead to convergence and similar results in any case.

The first section will treat 2D test cases with the NACA airfoil and a short proof that frequencies may be chosen that are not a harmonic of the base frequency. The subsequent section discusses the solutions of 3D test cases using the LANN wing.

Table 7.1: *Distribution of the instants as ratio of period $T = \frac{1}{f_1}$ with $t_0 = 0$ for each order N and distribution i .*

N,i	$\frac{t_1}{T}$	$\frac{t_2}{T}$	$\frac{t_3}{T}$	$\frac{t_4}{T}$	$\frac{t_5}{T}$	$\frac{t_6}{T}$	$\frac{t_7}{T}$	$\frac{t_8}{T}$	$\frac{t_9}{T}$	$\frac{t_{10}}{T}$
1,0	0.3	0.6								
1,1	0.25	0.75								
1,2	0.3	0.62								
1,3	0.1	0.2								
1,4	0.1	0.9								
1,5	0.49	0.51								
2,0	0.2	0.4	0.6	0.8						
2,1	0.125	0.5	0.625	0.75						
2,2	0.175	0.375	0.575	0.775						
2,3	0.1	0.2	0.3	0.4						
2,4	0.4	0.45	0.55	0.6						
3,0	0.142857	0.285714	0.428571	0.571428	0.714285	0.857142				
3,1	0.0416	0.083	0.125	0.16	0.2083	0.25				
3,2	0.16	0.32	0.45	0.605	0.73	0.88				
3,3	0.082	0.17	0.25	0.33	0.42	0.5				
3,4	0.1	0.2	0.3	0.5	0.6	0.7				
3,5	0.08	0.16	0.24	0.32	0.4	0.48				
4,0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8		
4,1	0.0625	0.125	0.1875	0.25	0.3125	0.375	0.4375	0.5		
4,2	0.1	0.21	0.31	0.425	0.53	0.65	0.755	0.875		
4,3	0.05	0.2	0.25	0.4	0.5	0.72	0.8	0.95		
5,0	0.09	0.18	0.27	0.36	0.45	0.54	0.63	0.72	0.81	0.90
5,1	0.15	0.2	0.225	0.25	0.65	0.7	0.725	0.75	0.8	0.9
5,2	0.08	0.17	0.2625	0.3525	0.45	0.536	0.625	0.715	0.82	0.925
5,3	0.1	0.2	0.225	0.25	0.64	0.68	0.7225	0.7575	0.79	0.9

Table 7.2: *Determinant of matrix A_1 : $|A_1|$, Condition number of coefficient matrix D : $\kappa(D)$ for each order N and distribution i .*

N,i	$ A_1 $	$\kappa(D)$	N,i	$ A_1 $	$\kappa(D)$	N,i	$ A_1 $	$\kappa(D)$
1,0	2.6	0.18E18	2,0	-14.0	0.41E16	3,0	-113.4	0.81E16
1,1	2.0	∞	2,1	-5.7	0.64E17	3,1	-0.000011	0.26E18
1,2	2.5	0.37E17	2,2	-13.6	0.44E18	3,2	-105.4	0.70E17
1,3	0.30	0.22E11	2,3	-0.30	0.18E11	3,3	-0.38	0.66E17
1,4	0.23	0.21E10	2,4	-0.21	0.35E12	3,4	-29.4	0.34E17
1,5	0.25	0.93E11				3,5	-0.19	0.62E17
4,0	1230.2	0.35E17	5,0	16692.1	0.12E17			
4,1	0.020	0.21E18	5,1	0.051	0.81E17			
4,2	1162.5	0.99E17	5,2	127.7	0.80E18			
4,3	98.8	0.73E18	5,3	0.33	0.26E18			

7.1 2D NACA airfoil

7.1.1 Test case CT2

Fig. 7.1 shows the solution for one frequency: The reference with uniformly distributed instants (1,0) and four different distributions of instants to be found in Tab. 7.1.

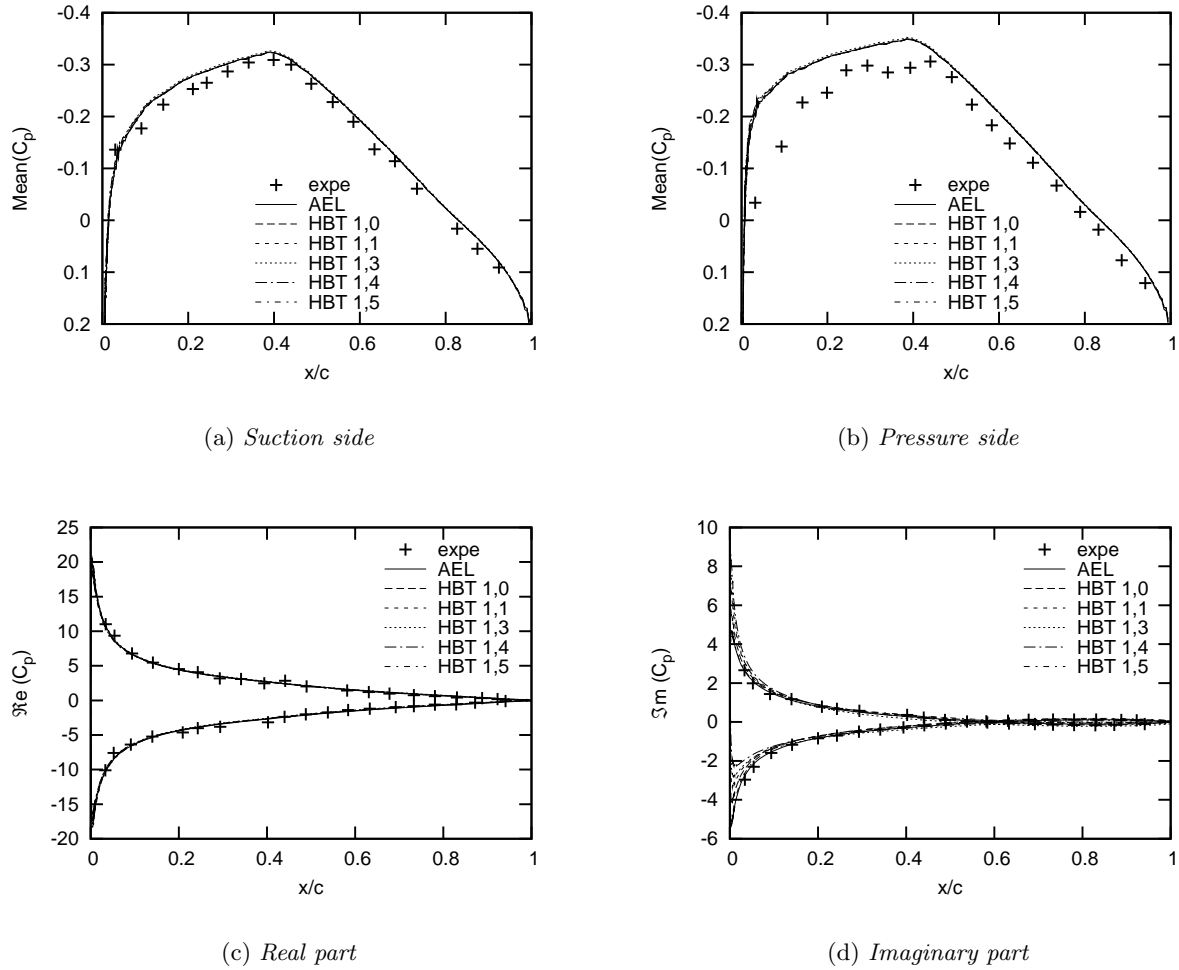


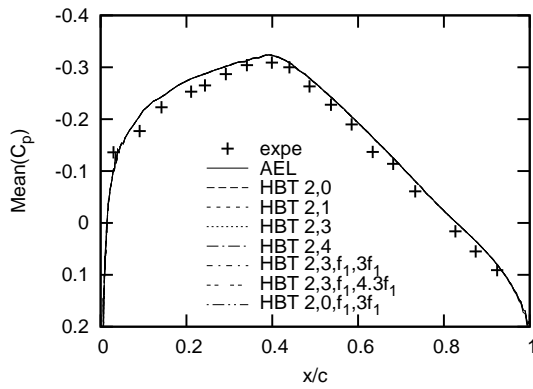
Figure 7.1: *NACA 64 CT2: Pressure coefficient C_p for $N = 1$ with non-uniformly distributed instants.*

Results are very good for the mean pressure coefficient and its real part. The graphs are virtually superposed with the solution disposing of uniformly distributed instants. Taking in account, that distributions (1,3), (1,4) and (1,5) are really far away from a uniform distribution and have a determinant $|A_1| \leq 0.3$, this is a very pleasant and encouraging result. However, there is an evident difference on the imaginary part between the five computations at the leading edge. Distributions (1,1), (1,3) and (1,4) are the ones, who are the furthest away from the AEL solution. Considering the whole graph, it is still a quite good solution. Convergence behaviour is the same as for the uniform distribution with $CFL = 100$ and 300 iterations (five orders of magnitude were lost on the

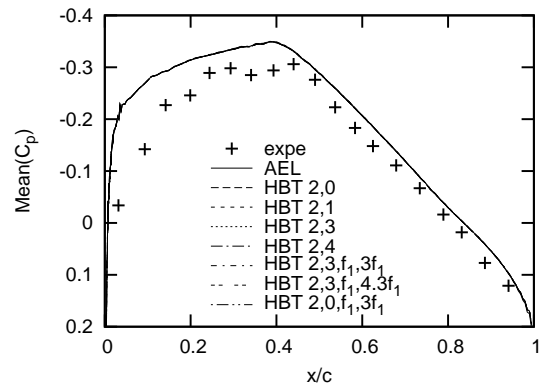
residuals).

Results for two frequencies are displayed in Fig. 7.2. Again, the graphs of the residuals are not given, they are equal to the ones obtained for the uniform distribution of instants. The solution for distributions (2, 1), (2, 3) and (2, 4) coincide perfectly with (2, 0), even though (2, 3) and (2, 4) have a small determinant $-0.4 < |A_1| < -0.1$ and are remote of the uniform distribution.

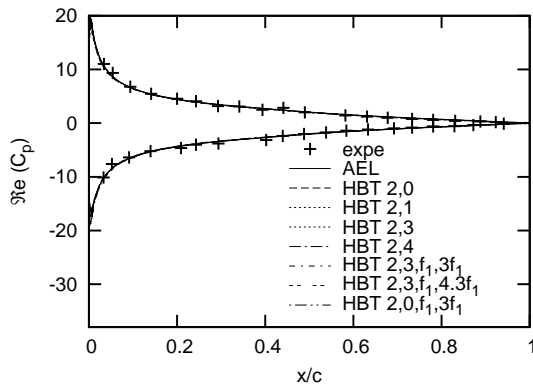
Three more graphs are given in Fig. 7.2 for which the second frequency is modified. For the first one: $f_2 = 3f_1$ and for the second one: $f_2 = 4.3f_1$. The last graph also uses $f_2 = 3f_1$, but with uniformly distributed instants (distribution (2, 0)). It is perfectly coincident with (2, 0), which is not too surprising, because the results for this test case are already very good for one frequency and do not vary when higher frequencies are taken in account. Nevertheless, using a different frequency f_2 in combination with non-uniformly distributed instants does lead to slight deviations in the graphs of the real and imaginary part of the pressure coefficient. An undershoot is stated at the leading edge of about 10% on the real part of C_p and an overshoot of the same magnitude on its imaginary part. However, using different frequencies does also lead to good results in this test case, even in combination with non-uniformly distributed instants.



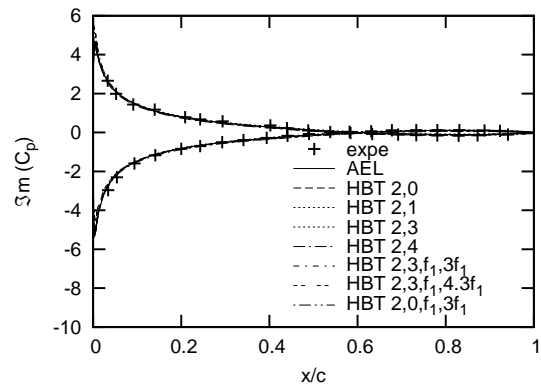
(a) Suction side



(b) Pressure side



(c) Real part



(d) Imaginary part

Figure 7.2: NACA 64 CT2: Pressure coefficient C_p for $N = 2$ with non-uniformly distributed instants and/or varying frequencies.

Looking at the solution of computations of order three (cf. Fig. 7.3), it is evident that very good accordance with the original results is obtained. All graphs are coincident and the convergence graphs are again identical to the one with a uniform distribution of the instants.

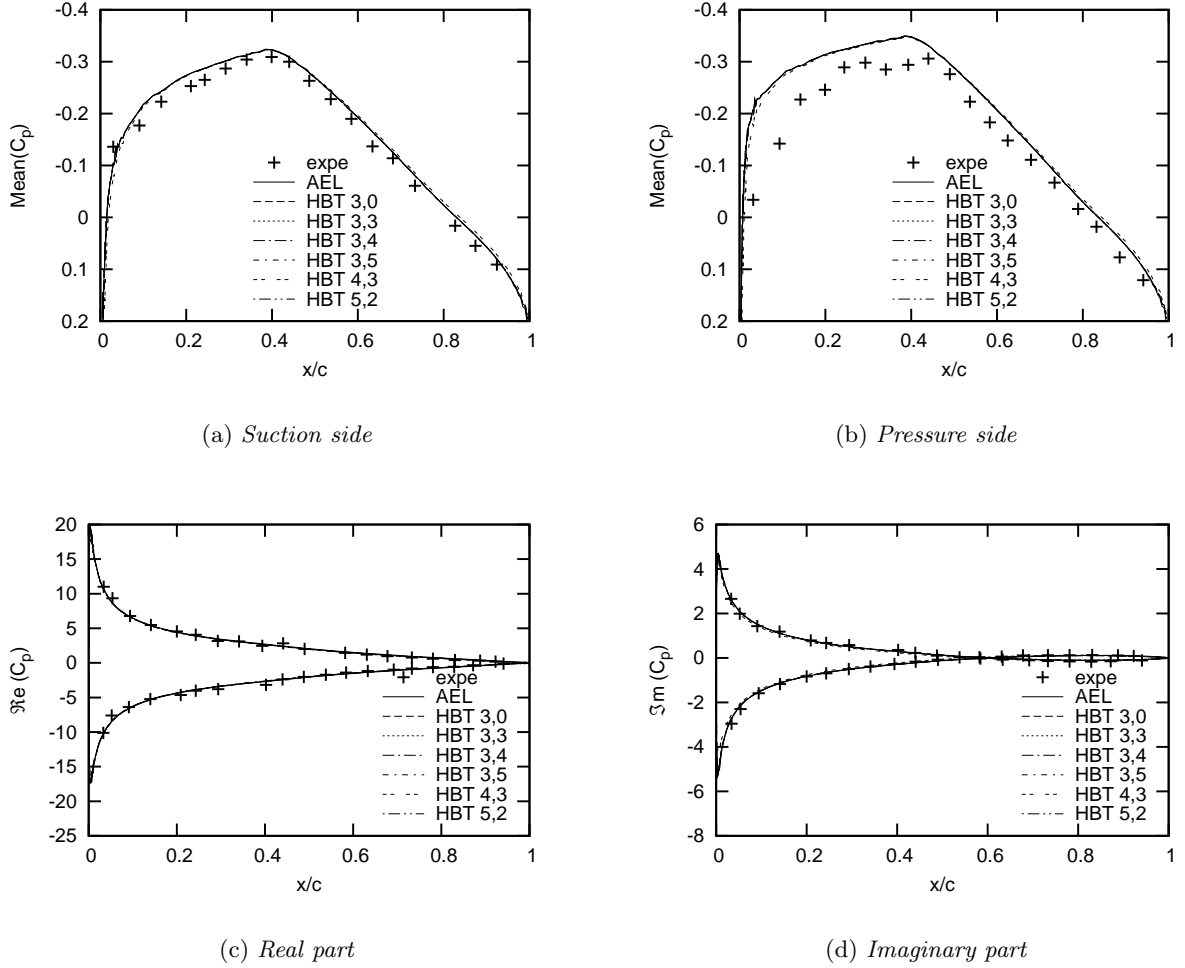


Figure 7.3: *NACA 64 CT2: Pressure coefficient C_p for $N = 3, 4$ and 5 with non-uniformly distributed instants.*

However, distribution (3,1) was not computable and either negative temperature or pressure values aborted the computation. Rigorous reductions of CFL (CFL was reduced from 100 down to 0.01) and an increase of cycles in the implicit phase did not lead to convergence. These modifications make computation much more costly and should be circumvented. It is obviously the very low determinant of matrix A_1 ($|A_1| \approx 10^{-5}$, cf. Tab. 7.2) that messes up the computation. Distribution (3,3) and (3,5) dispose of a low determinant ($0.1 < |A_1| < 0.5$), but pass very well. It is the author's experience that determinants at least greater than 0.1 should be chosen in order to obtain good convergence behaviour and results, although this does not necessarily assure good results and greater determinants may have to be obtained. This restriction becomes more and more difficult to obey though, when a higher order is needed, which is not the case for test case CT2, because results are already good with only one frequency taken in account. For other test cases more frequencies might have to be used and there are certainly applications, where an elevated number of frequen-

cies appear. This means immediately that the number of instants rises as well (number of instants = $(2N + 1)$). If it is in addition necessary to distribute these instants over a very small period, $|A_1|$ will decrease automatically. This is due to the nature of matrix A_1 and inevitable. This effect can be seen very nicely for distribution (3, 1), where seven instants are distributed between $t_0 = 0$ and $t_{2N} = \frac{1}{4}T = \frac{1}{4f_1}$.

The statements concerning the determinant made for distribution (3, 1) are equally true for distributions (4, 1) and (5, 1). Both computations did not converge. The solution of distribution (4, 3) shows a slight deviation on the mean C_p and the imaginary part compared to the other graphs. Distribution (5, 3) did not converge, the determinant of $|A_1| \approx 0.3$ being apparently too small. However, (5, 2) shows very good accordance, but is also very close to a uniform distribution.

7.1.2 Test case CT6

For test case CT6, results are presented in Fig. 7.4. Distributions (1, 1), (2, 1), (3, 2) and (4, 2) are compared to the computation (4, 0) (cf. Fig. 6.6).

(4, 0) has uniformly distributed instants and shows good accordance to the AEL computation. With one or two frequencies taken in account, distribution $i = 1$ converges very well. No convergence was obtained though for distributions further away from the uniform distribution. For $N = 3$ and 4, only distribution $i = 2$, which is very close to the uniform distribution, converges. Other distributions do not converge, even with rigorous reductions of CFL and a raise in implicit phase cycles. The graphs of the residuals are all four very similar to the ones showed in Fig. 6.4 (for each frequency respectively).

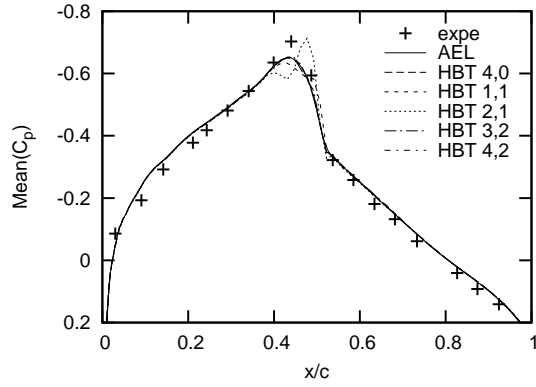
When Fig. 7.4(a) and 7.4(b) are considered (the mean pressure coefficient of the suction and pressure side), very good accordance for (3, 2) and (4, 2) is stated. However, (1, 1) and in particular (2, 1) have problems to resolve the shock at $0.4 \leq x/c \leq 0.6$. While (1, 1) shows an additional step on the pressure side, (2, 1) disposes of an overshoot on the suction side and one oscillation at the shock. Refer here to Chap. 3.

Distribution (4, 2) is almost coincident with the AEL computation for the real and imaginary part of C_p . (3, 2) is further away from it and oscillates around the shock on the pressure as well as on the suction side. The result of (1, 1) for the real part is acceptable although the shock is somewhat shifted towards the trailing edge on the pressure side. The imaginary part shows radical over- and undershoots though. (2, 1) is even worse. It computes pressure peaks way too high for both the real and imaginary part of the pressure coefficient. However, it does not show an overshoot at the end of the shock ($x/c = 0.55$) on the suction side of the imaginary part.

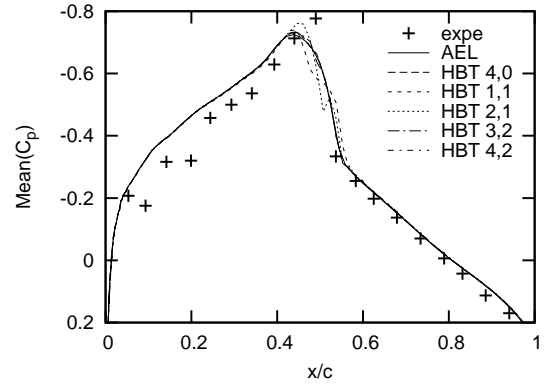
7.1.3 Test case Shock Stall

The last 2D test case is also the one which poses the biggest problem with non-uniformly distributed instants. For $N = 1, 2$ and 3, only distribution $i = 2$ converges, but shows some deviation. Only the results for three frequencies are shown in Fig. 7.5, because here, yet the best results were obtained.

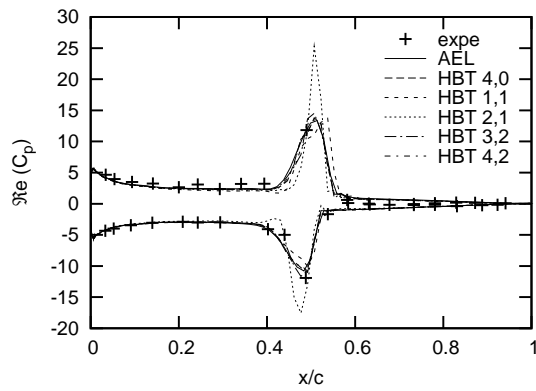
The graph of the mean pressure coefficient is quasi coincident with the one obtained with the uniform distribution of instants. The pressure side on which no shock appears also shows very good accordance to (3, 0). The pressure side of the imaginary part of C_p disposes of a solution that



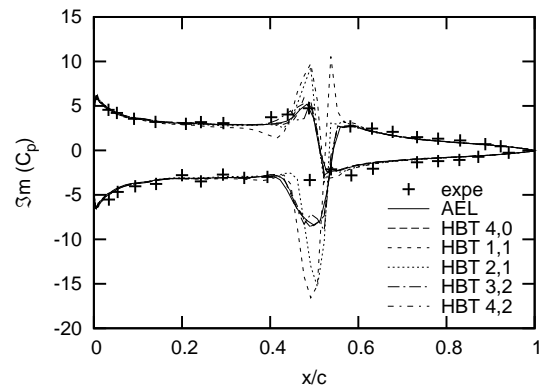
(a) Suction side



(b) Pressure side



(c) Real part



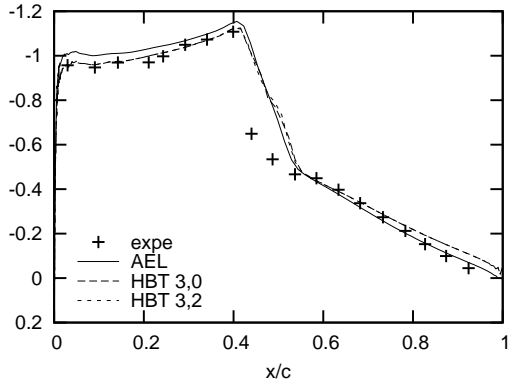
(d) Imaginary part

Figure 7.4: NACA 64 CT6: Pressure coefficient C_p for $N = 1, 2, 3$ and 4 with non-uniformly distributed instants.

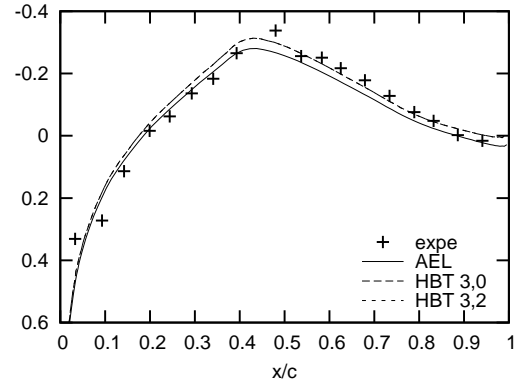
cannot be considered worse than the one with uniform distribution. Distribution (3,0) features a step at the location of the shock on the pressure side of the real part. Here, distribution (3,2) actually shows an oscillation with a quite high peak. Taking in account though, that distribution $i = 2$ is at most $\Delta T = 0.05T$ away from the uniform one, it cannot be considered to be a good solution, anymore.

7.1.4 Summary

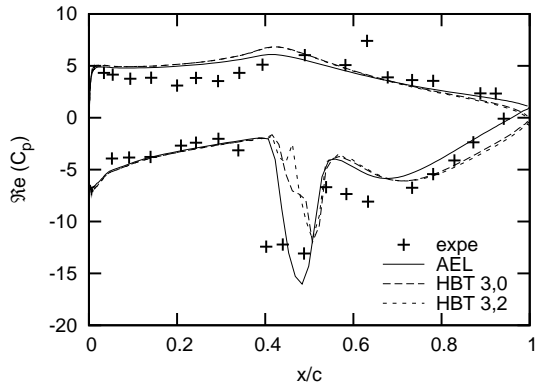
For now, the preceding results allow to draw the following conclusions: Using non-uniformly distributed instants does not pose a problem as long as the physics of the actual problem are not too extreme (e.g. shocks, boundary layer separation). The condition of the coefficient matrix D is irrelevant for both approaches. The determinant of matrix A_1 , giving an idea of the quality of the instant distribution, must not be too small, however. But again, this depends highly on the considered problem. For test case CT2, where no shock appears on the airfoil, this determinant



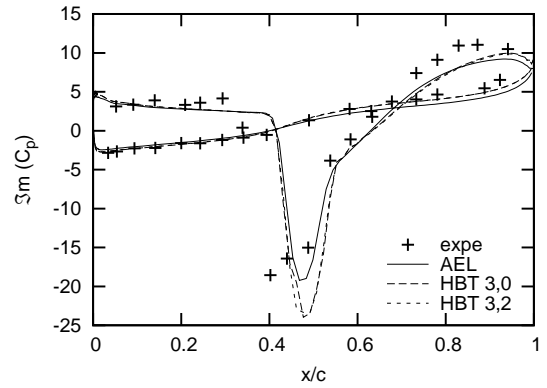
(a) Suction side



(b) Pressure side



(c) Real part



(d) Imaginary part

Figure 7.5: *NACA 64 Shock Stall case: Pressure coefficient C_p for $N = 3$ with non-uniformly distributed instants.*

may be less than one (although it should be chosen greater than 0.1). The other two test cases are much more sensitive towards the distribution of the instants. It was not possible here to obtain a solution with the instants being distributed only e.g. over a half or a quarter period. Section 7.2 will have to show, if the 3D test cases are more adequate for non-uniformly distributed instants.

7.2 3D LANN wing

Before discussing the test results, some general remarks are made. Not all the results gained are shown in this report. Only some results are displayed in order to emphasise the conclusions. The following distributions were not computable (abort of the computation due to negative pressure or temperature values) with test case CT5: (3, 1) and (5, 1) and with test case CT9: (3, 1), (4, 1) and (5, 1). Actually, these computations do not converge even with rigorous reductions of the CFL number and/or a raise of cycles in the implicit phase. This property was already discovered with

the NACA test cases and leads to the assumption that it is not really a problem of convergence, but rather inherent and due to the instant distribution.

Generally speaking, the C_p graphs of the pressure side, where no shock appears are in the majority of cases very close to or even superposed with the AEL or the HBT solution (with uniformly distributed instants). This is also true for the suction side except for the area in direct neighbourhood of the shock. Good results for section 4 and 5 seem to be especially hard to obtain. When oscillations emerge, it is usually first on these two sections.

Distribution $i = 2$ always furnishes results that are quasi superposed with the solution with uniformly distributed instants, except for distribution (2, 2) (CT5 and CT9). However, convergence behaviour is good and the determinant is almost equal to the one of (2, 0). Obviously, this instant distribution does not capture well the physics of the problem.

7.2.1 Test case CT5

As mentioned already, a reduction of the CFL number does not lead to results, if the computation aborts in the case of non uniformly distributed instants. Hence, $CFL = 100$ is used for all test runs. Convergence behaviour is similar to the one with uniformly distributed instants, but small differences arise. For $N = 1$ and $N = 3$, the residual curves are superposed only up to 150 – 200 iterations, but they converge all to the same order of magnitude after 300 iterations. However, more distinct deviations can be seen for order two and four (cf. Fig. 7.6). (2, 4) deviates the most from the other residual graphs for $N = 2$. Convergence behaviour of (5, 3) is similar to the one of (4, 3). Particularly, (4, 1) shows a quite different behaviour. It actually converges faster between 30 and 120 iterations, but shows oscillatory behaviour on the residuals afterwards. It is less well converged after 300 iterations than the other distributions. With test case CT9, this distribution is not computable at all.

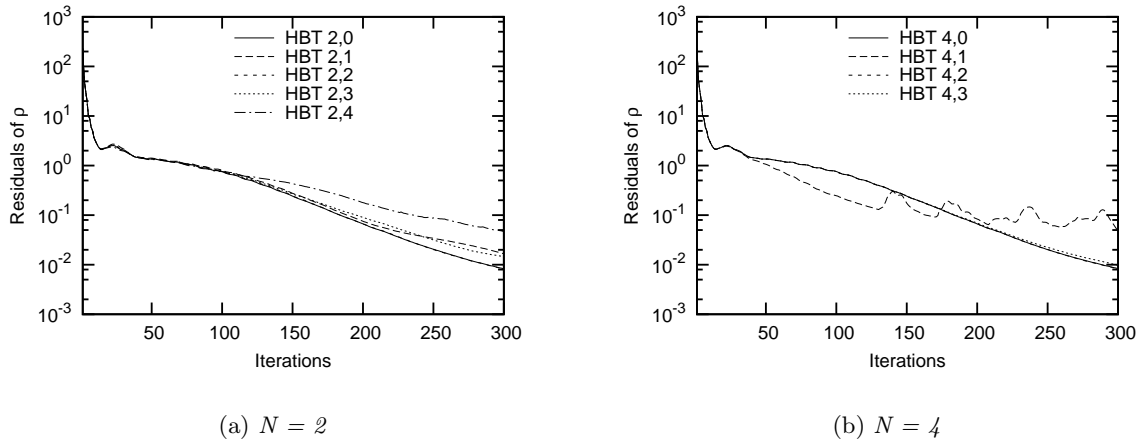
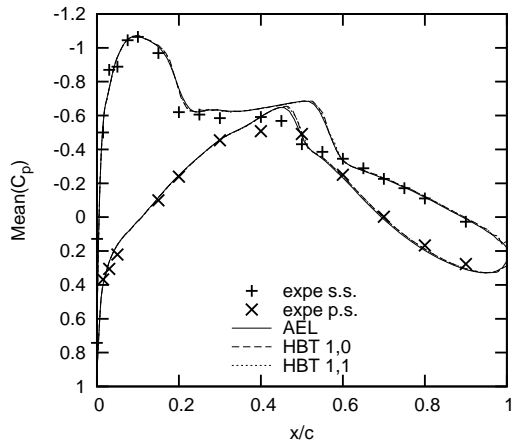


Figure 7.6: *LANN CT5: Convergence.*

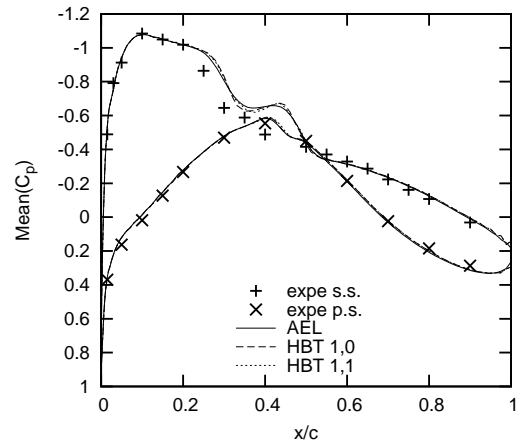
Fig. 7.7- 7.9 are an example of how a good solution with non-uniformly distributed instants (distribution (1, 1)) can be obtained. On sections 4 and 5 of C_p 's real part, the shock on the suction side is even shifted closer to the experimental data points. On the imaginary part, overshoots at the peaks of about 10% are observable.

Admittedly, distribution (1,4), that is further away from a uniform one, shows oscillations at the position of the shock already on the mean pressure coefficient (cf. Fig. 7.10). These oscillations aggravate on its real and imaginary part and significant overshoots appear on the peaks. On the real part of section 6 (cf. Fig. 7.11(f)), the oscillations even spread to the entourage of the shock. However, apart from the shocks, the solution is either very close or even superposed with HBT (1,0) and AEL.

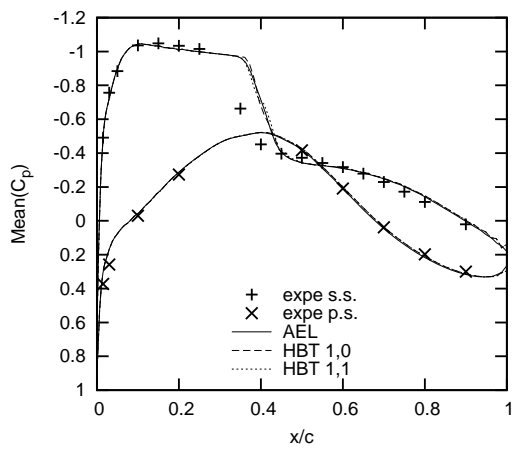
The last example of test case CT5 is of order 3 with a distribution of the instants over half the period. Fig. 7.13 - 7.15 depict the results. There are no problems with the approximation of the mean pressure coefficient on the pressure side and on section 1, the shock is still captured quite well. On all other sections, strong oscillations appear. These even spread to the trailing edge of wing sections 5 and 6 (i.e. the outer region of the wing). The real and imaginary part of C_p only show important deviation at the peaks and behind them. Aside from the oscillations, the peaks are partly extremely over predicted.



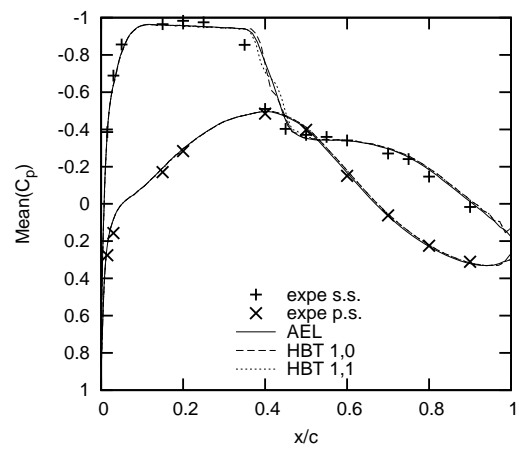
(a) Section 1 - 20%



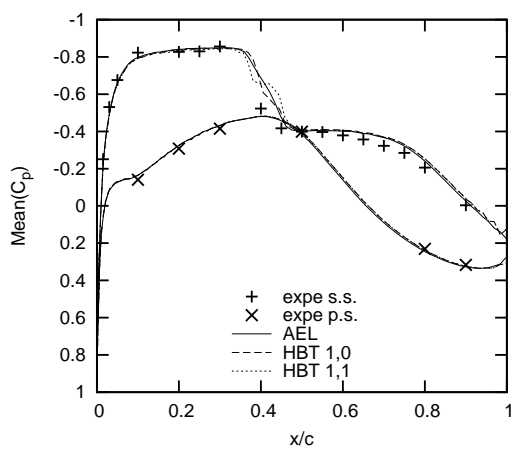
(b) Section 2 - 32.5%



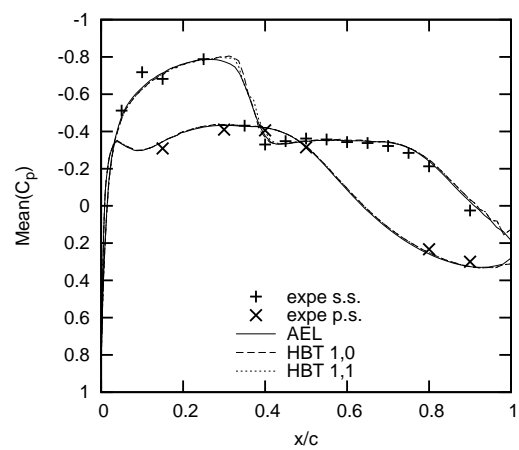
(c) Section 3 - 47.5%



(d) Section 4 - 65%

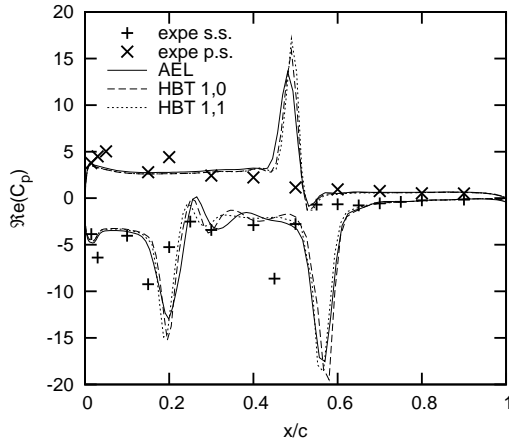


(e) Section 5 - 82.5%

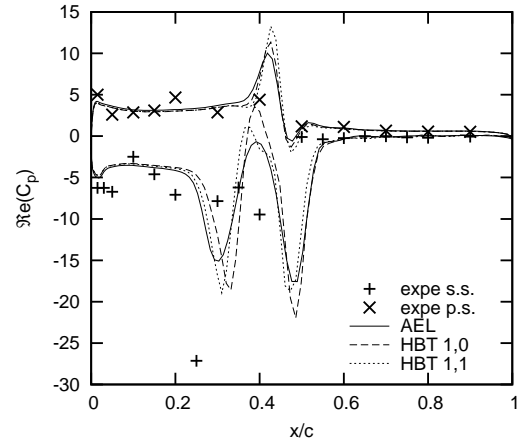


(f) Section 6 - 95%

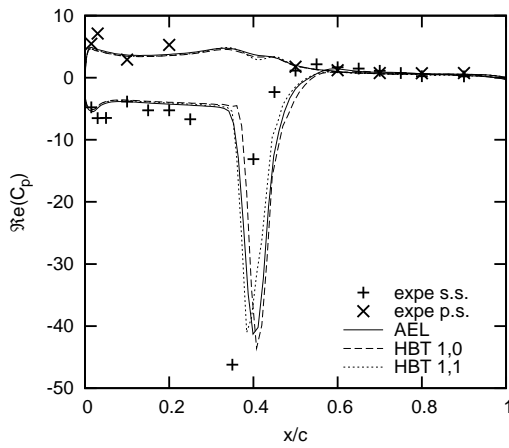
Figure 7.7: LANN CT5: Mean pressure coefficient C_p for (1,1).



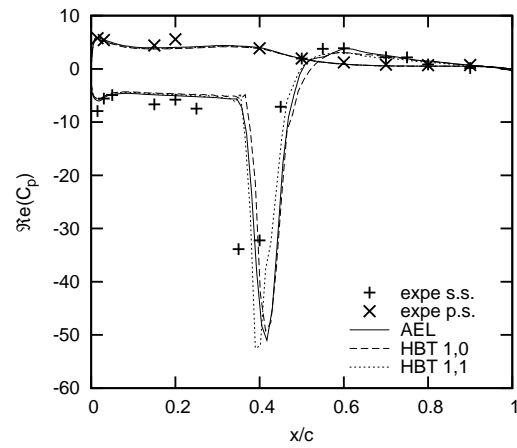
(a) Section 1 - 20%



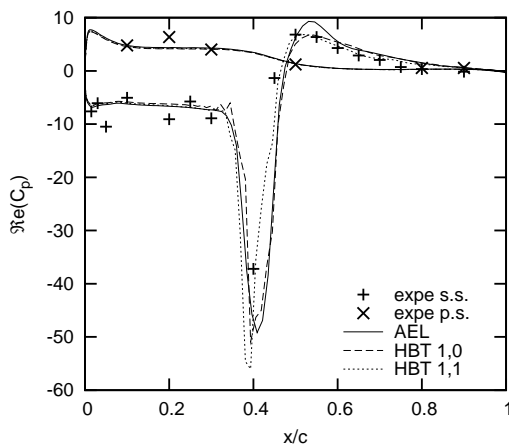
(b) Section 2 - 32.5%



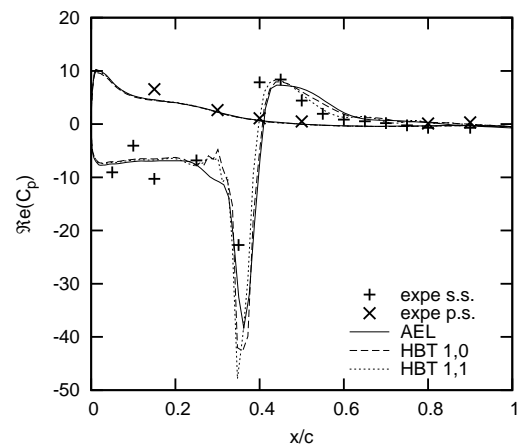
(c) Section 3 - 47.5%



(d) Section 4 - 65%

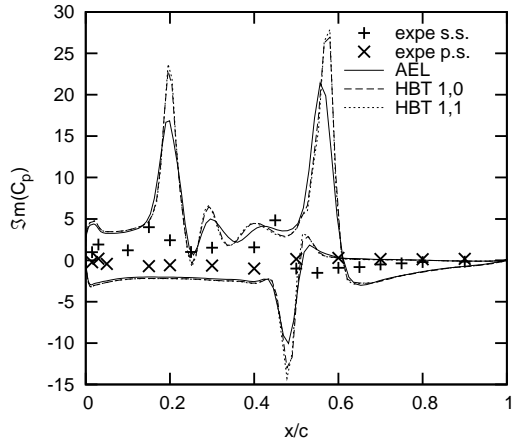


(e) Section 5 - 82.5%

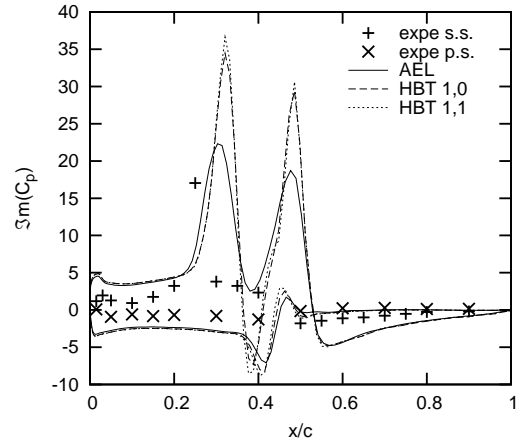


(f) Section 6 - 95%

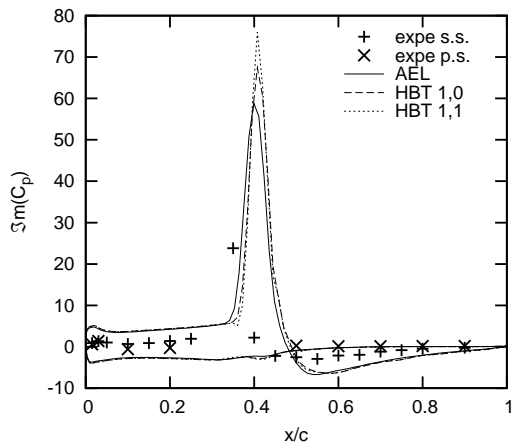
Figure 7.8: LANN CT5: Real part of C_p for (1,1).



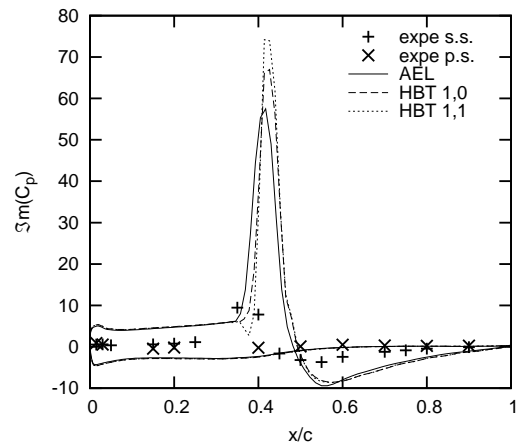
(a) Section 1 - 20%



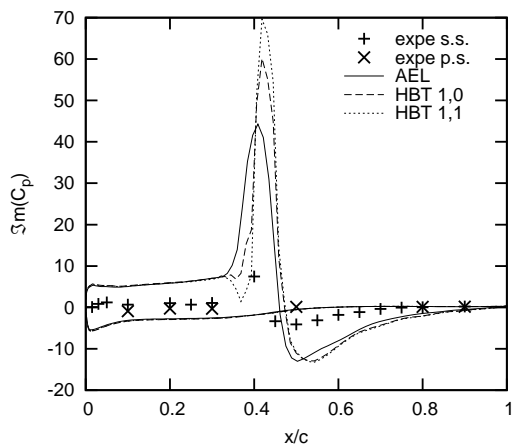
(b) Section 2 - 32.5%



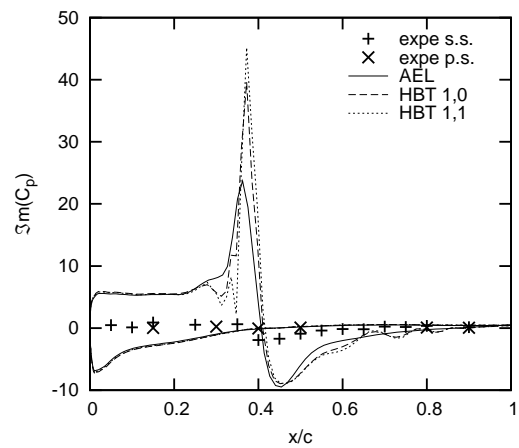
(c) Section 3 - 47.5%



(d) Section 4 - 65%

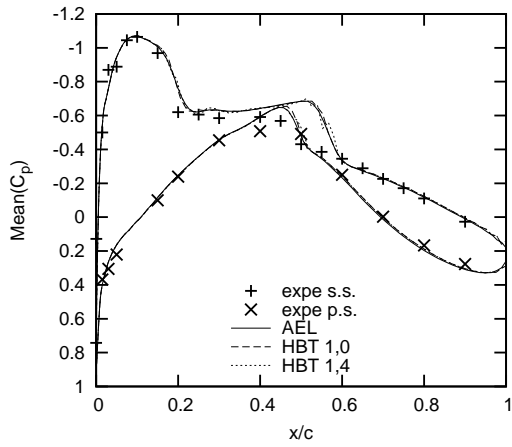


(e) Section 5 - 82.5%

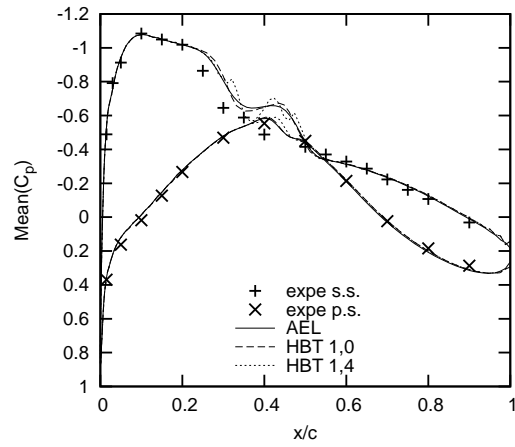


(f) Section 6 - 95%

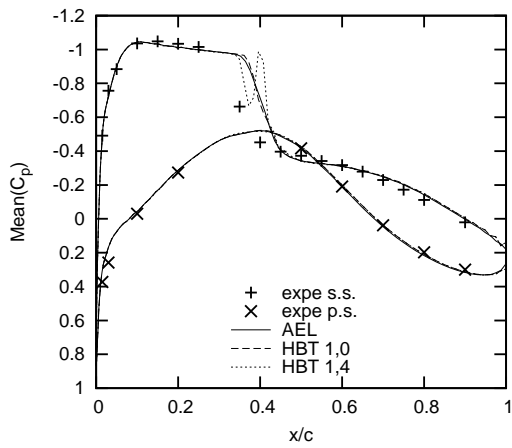
Figure 7.9: LANN CT5: Imaginary part of C_p for (1,1).



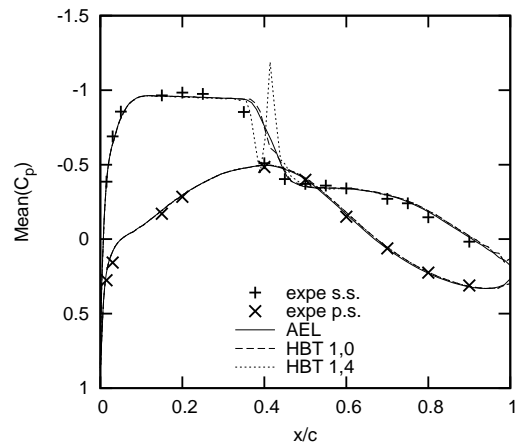
(a) Section 1 - 20%



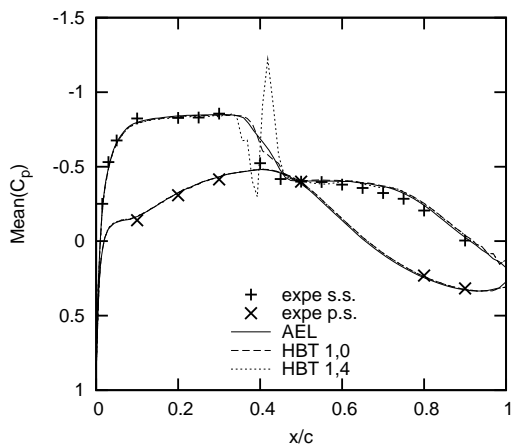
(b) Section 2 - 32.5%



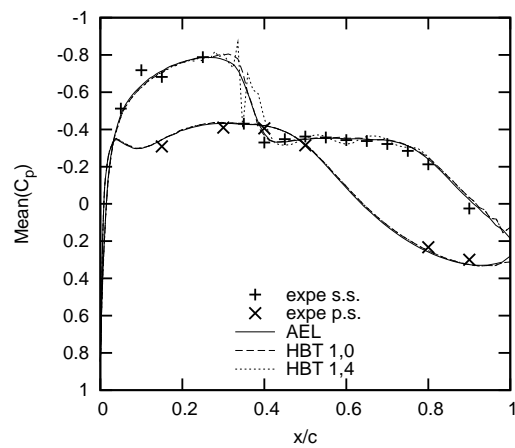
(c) Section 3 - 47.5%



(d) Section 4 - 65%

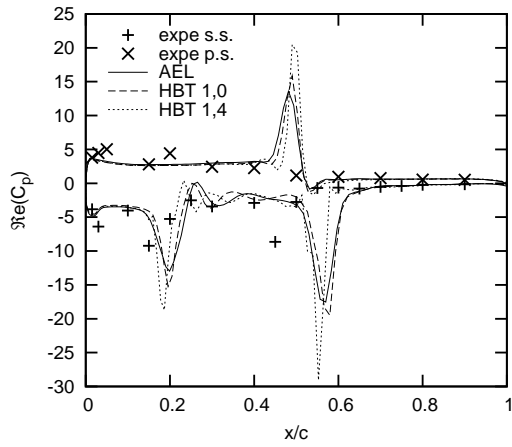


(e) Section 5 - 82.5%

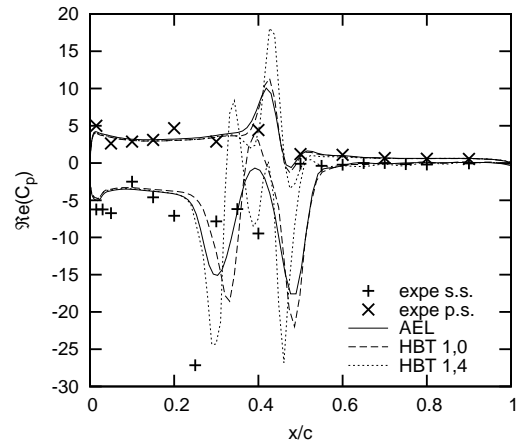


(f) Section 6 - 95%

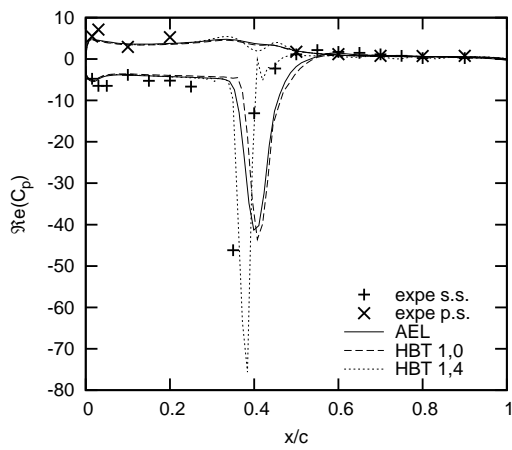
Figure 7.10: LANN CT5: Mean pressure coefficient C_p for (1,4).



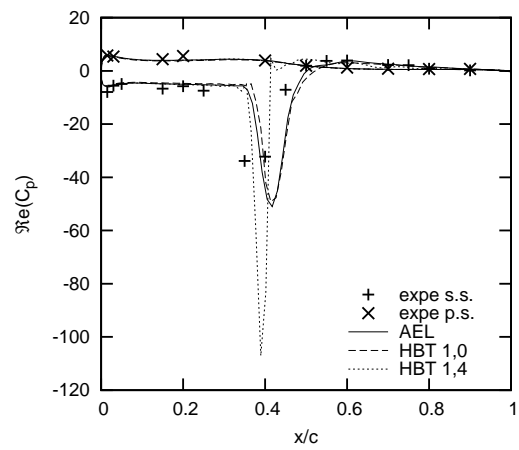
(a) Section 1 - 20%



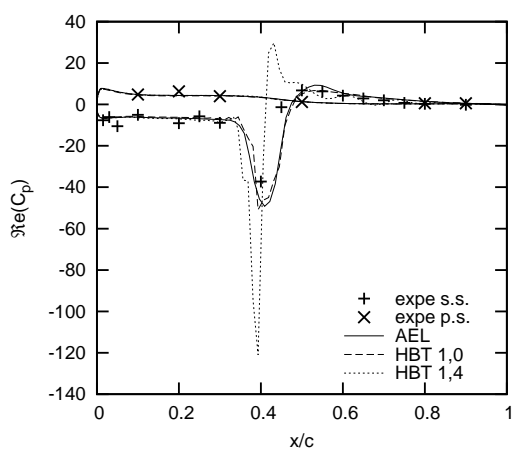
(b) Section 2 - 32.5%



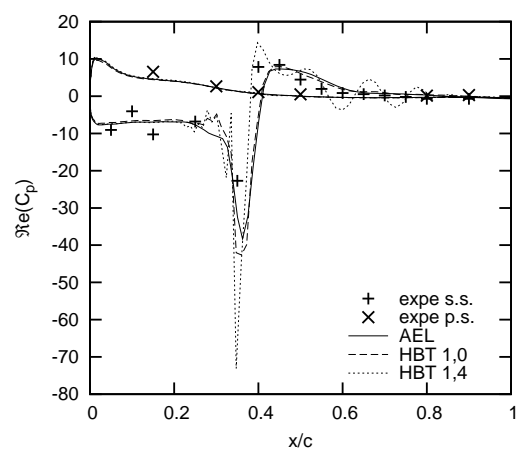
(c) Section 3 - 47.5%



(d) Section 4 - 65%

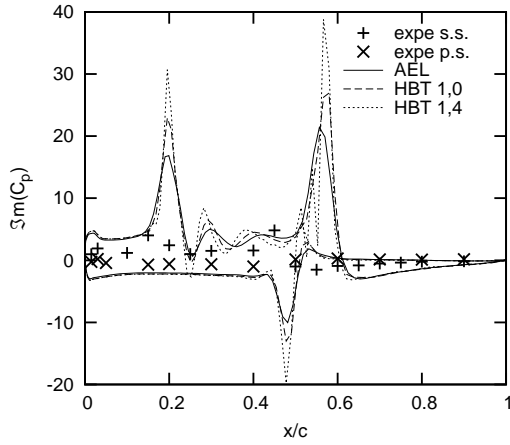


(e) Section 5 - 82.5%

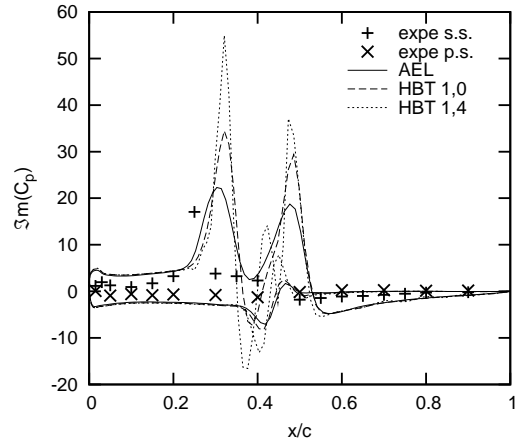


(f) Section 6 - 95%

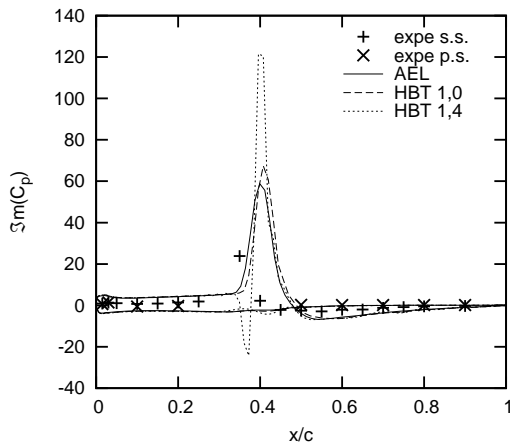
Figure 7.11: LANN CT5: Real part of C_p for (1,4).



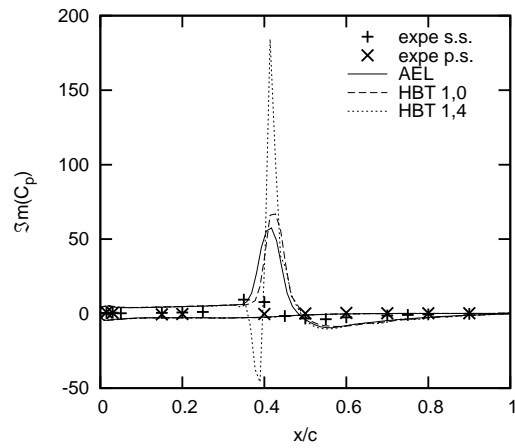
(a) Section 1 - 20%



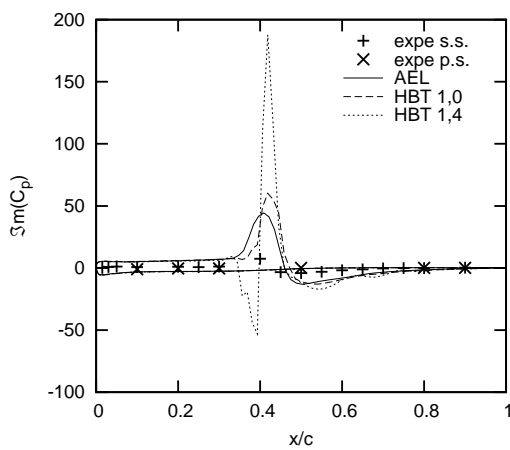
(b) Section 2 - 32.5%



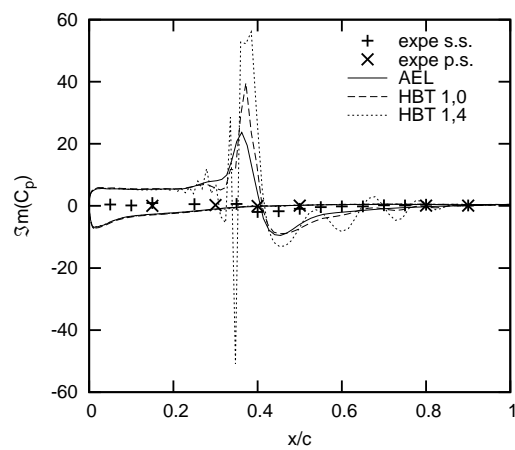
(c) Section 3 - 47.5%



(d) Section 4 - 65%

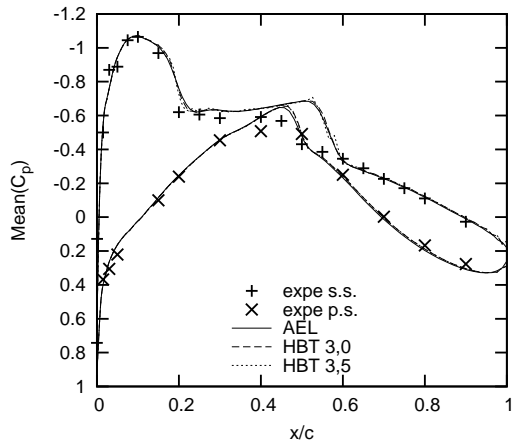


(e) Section 5 - 82.5%

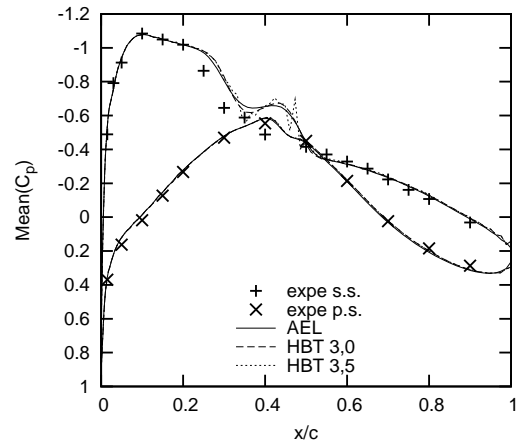


(f) Section 6 - 95%

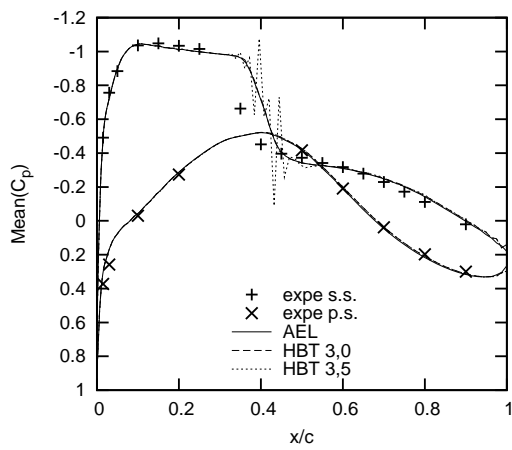
Figure 7.12: LANN CT5: Imaginary part of C_p for (1,4).



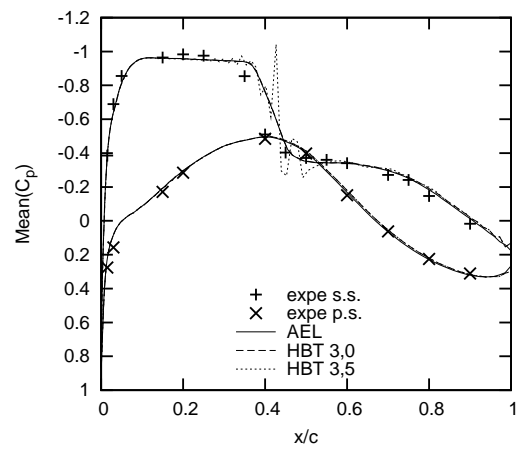
(a) Section 1 - 20%



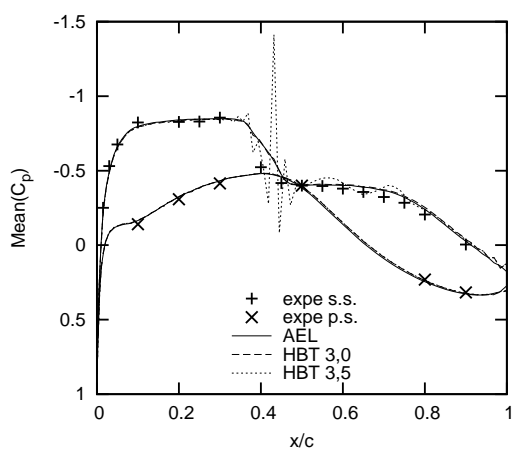
(b) Section 2 - 32.5%



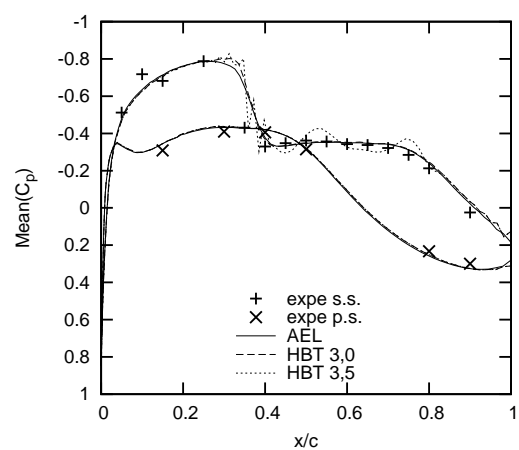
(c) Section 3 - 47.5%



(d) Section 4 - 65%

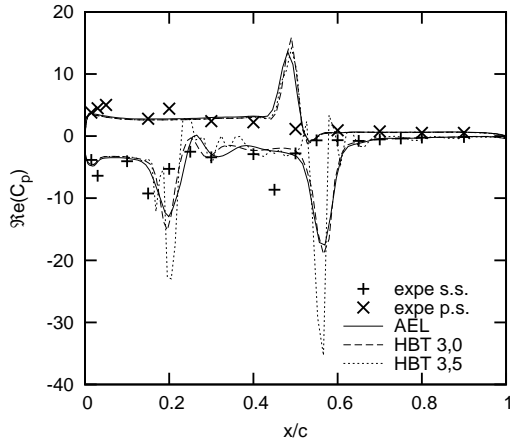


(e) Section 5 - 82.5%

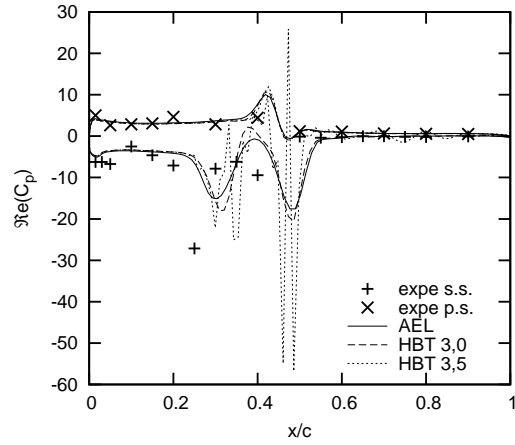


(f) Section 6 - 95%

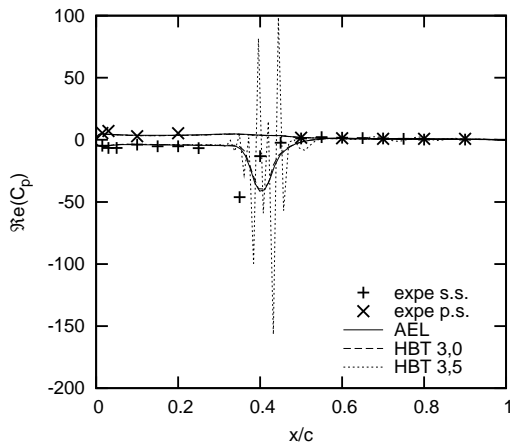
Figure 7.13: *LANN CT5*: Mean pressure coefficient C_p for (3,5).



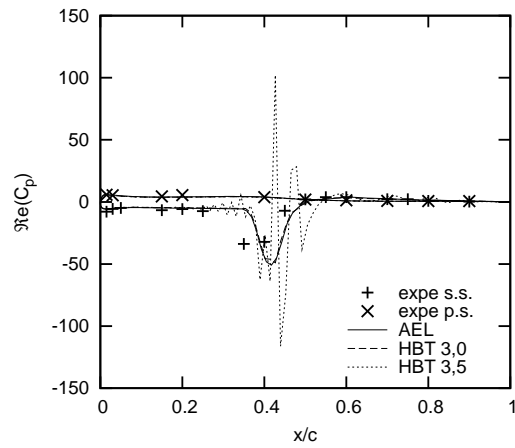
(a) Section 1 - 20%



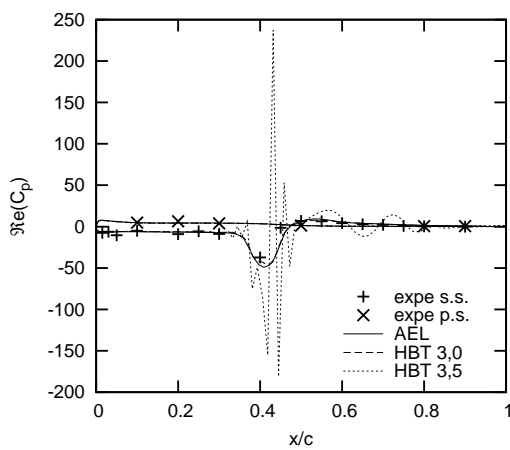
(b) Section 2 - 32.5%



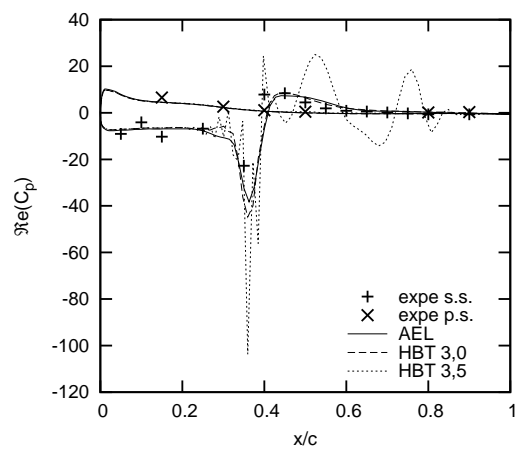
(c) Section 3 - 47.5%



(d) Section 4 - 65%

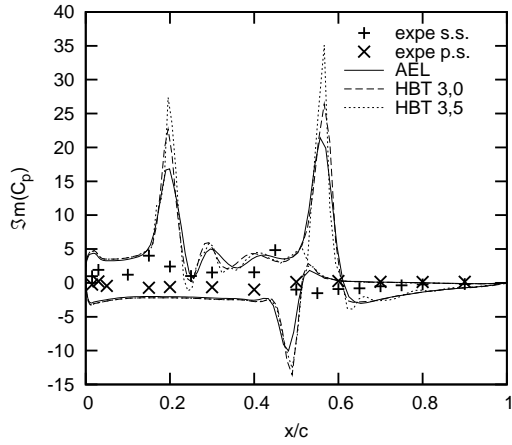


(e) Section 5 - 82.5%

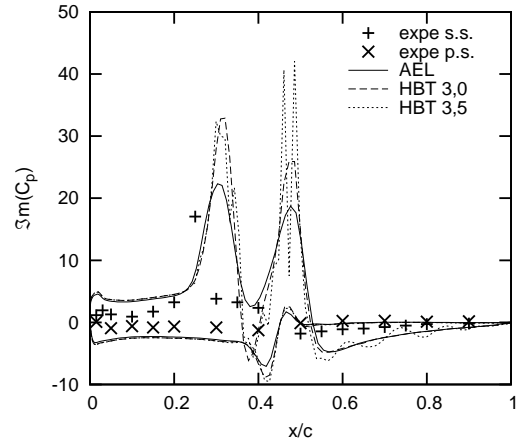


(f) Section 6 - 95%

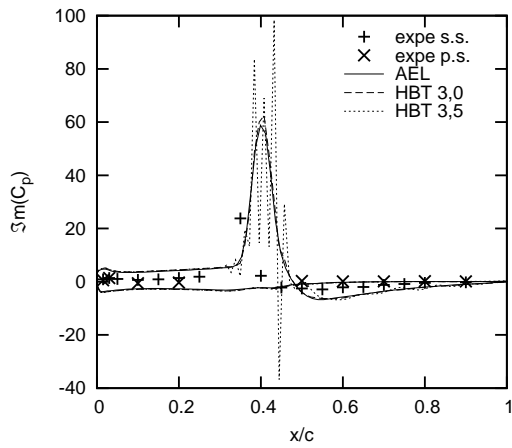
Figure 7.14: LANN CT5: Real part of C_p for (3,5).



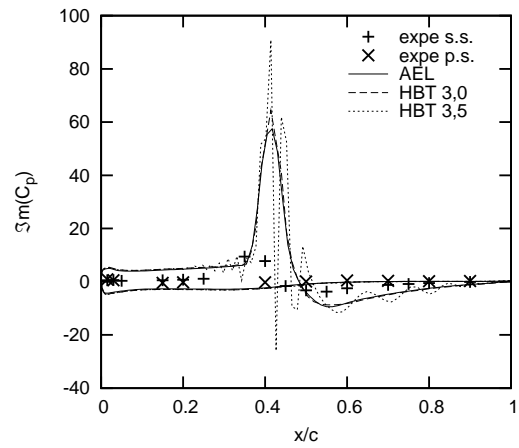
(a) Section 1 - 20%



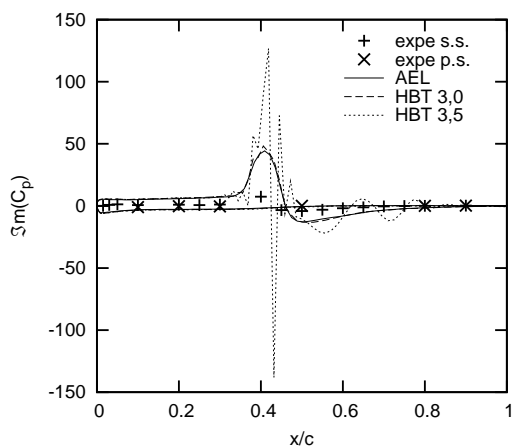
(b) Section 2 - 32.5%



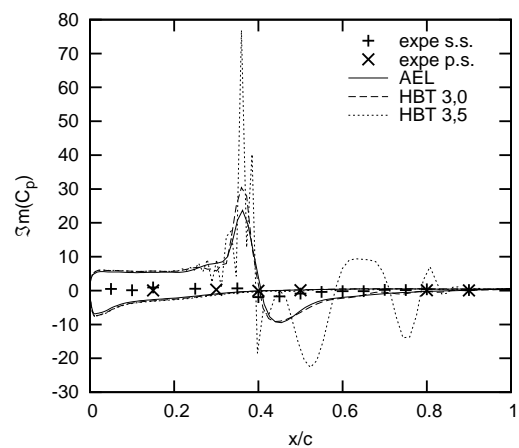
(c) Section 3 - 47.5%



(d) Section 4 - 65%



(e) Section 5 - 82.5%



(f) Section 6 - 95%

Figure 7.15: LANN CT5: Imaginary part of C_p for (3,5).

7.2.2 Test case CT9

$CFL = 100$ is used for all test runs. Only small differences in convergence behaviour compared to the uniform distributions are observed for $N = 1, 4$ and 5 for the computations that did not abort (except for $(5, 3)$, whose convergence behaviour is similar to the one of CT5: $(4, 3)$). The residual over the number of iterations is depicted in Fig. 7.16 for $N = 2$ and 3 . Distribution $(2, 4)$ converges less well than the other distributions and oscillates around 10^0 . $(3, 3)$ and $(3, 5)$ also show worse convergence behaviour, but end at the same order of magnitude.

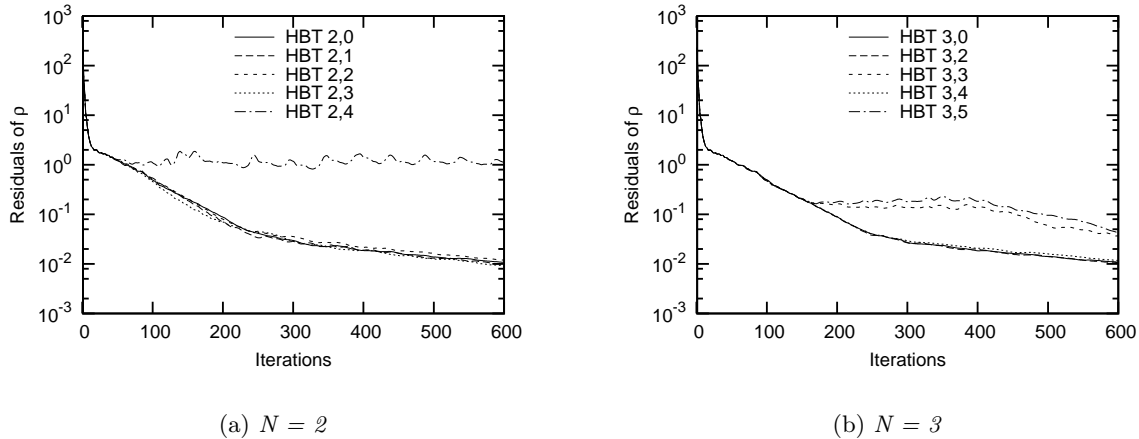


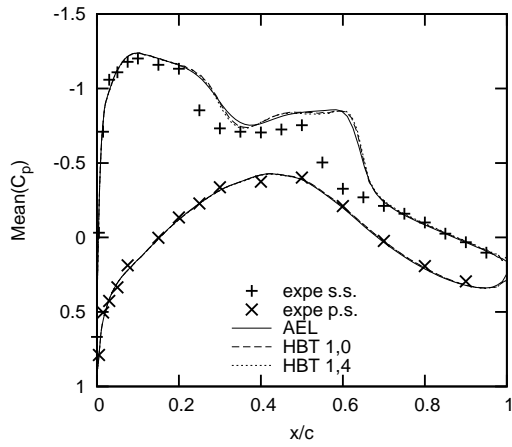
Figure 7.16: *LANN CT9: Convergence.*

Distribution $(1, 4)$ is used here again, in order to emphasise the fact that the same distribution may show different behaviour in two test cases, that are basically equal, but where CT9 deals with a higher mean angle of attack and a boundary layer separation. The mean pressure coefficient is actually very well approximated (cf. Fig. 7.17). On section 4, the double step appearing at the shock with $(1, 0)$ is even less distinct with $(1, 4)$. The real part is however shifted away from the AEL and HBT $(1, 0)$ graphs on both the pressure and suction side (cf. Fig. 7.18). Furthermore, it is interesting to see, that the shock on section 4 and 5 is very close to the experimental data. These statements are equally true for the imaginary part (cf. Fig. 7.19). Here, $(1, 4)$ is closer to the experimental data. Likewise, the real part is approximated quite well.

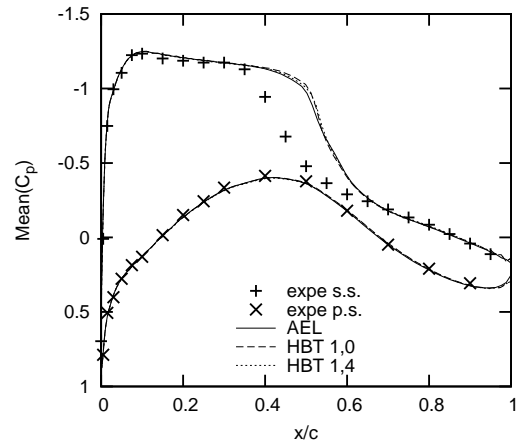
The previous experience demonstrates that a solution different from the AEL and Monofrequential HBT solution is not necessarily a bad solution. And sometimes, better results can be obtained, when a certain instant distribution is chosen.

Fig. 7.20- 7.22 depict the solution with distribution $(4, 3)$ that is also discussed in Chap. 3. The prediction that this distribution is quite good-natured, can be affirmed here. Only relatively small oscillations appear and for the most part the graphs are superposed. This can be an interesting additional test for the choice of a certain distribution. The instants may be first tested on an arbitrary, relatively simple function, like function $g(x)$ in chapter 3. Together with the program described in section 4.2.2, a good distribution may be found. Admittedly, this is a proceeding of trial and error.

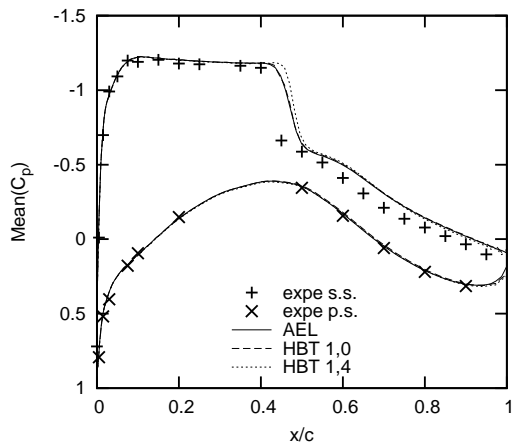
A very bad solution is depicted in Fig. 7.23- 7.25. This highlights, what can happen, when non-uniformly distributed instants are used. Again, this kind of behaviour was somewhat predicted by the graphs obtained in chapter 3 for distribution $(5, 3)$.



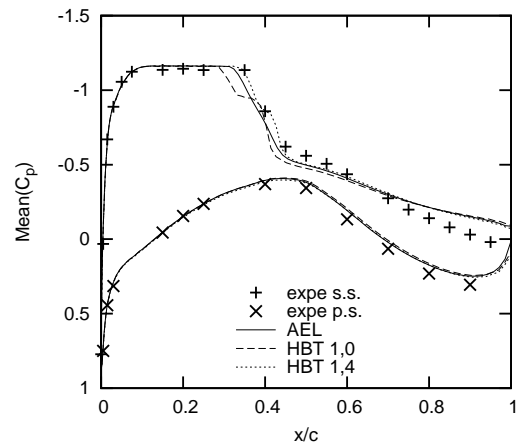
(a) Section 1 - 20%



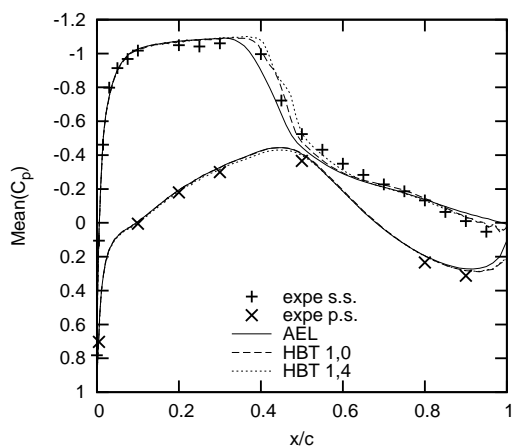
(b) Section 2 - 32.5%



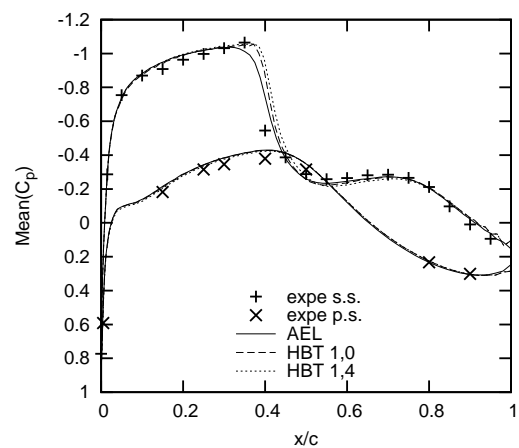
(c) Section 3 - 47.5%



(d) Section 4 - 65%

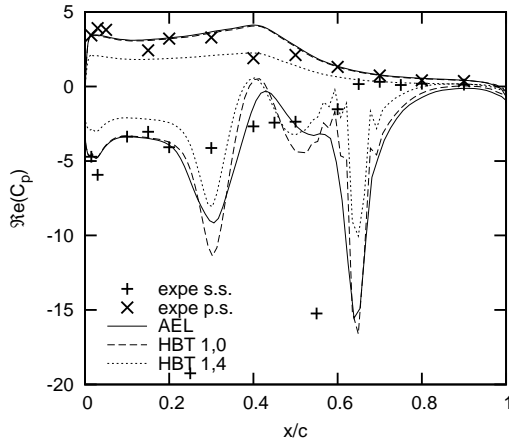


(e) Section 5 - 82.5%

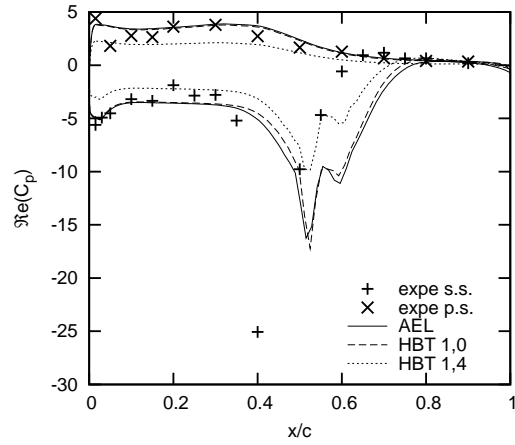


(f) Section 6 - 95%

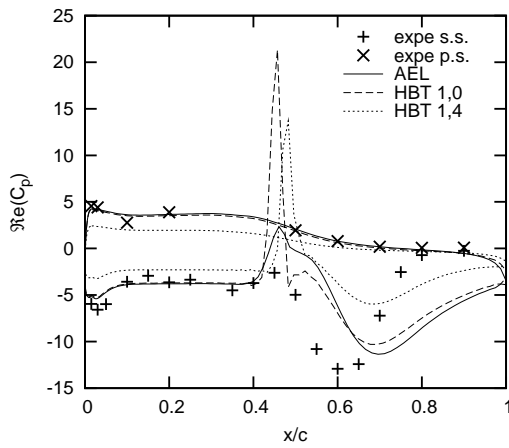
Figure 7.17: LANN CT9: Mean pressure coefficient C_p for (1,4).



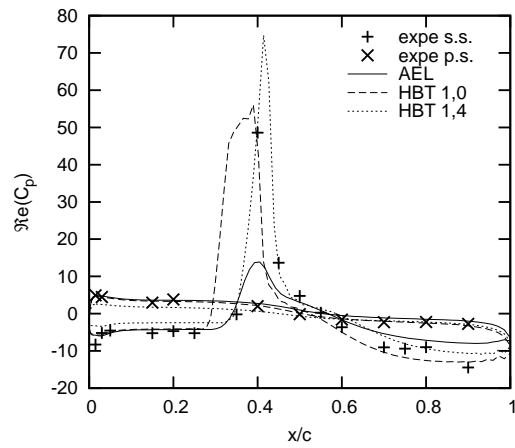
(a) Section 1 - 20%



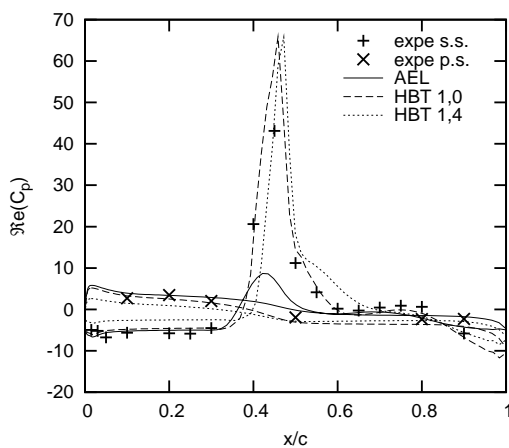
(b) Section 2 - 32.5%



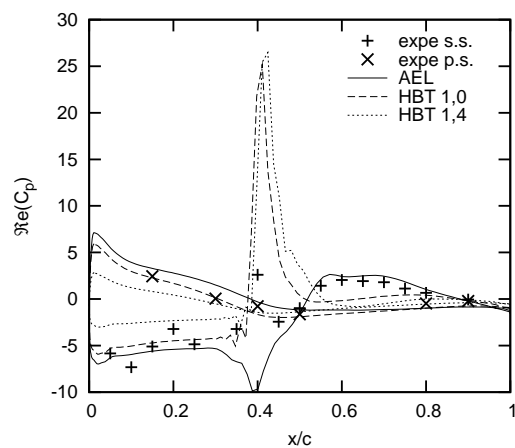
(c) Section 3 - 47.5%



(d) Section 4 - 65%

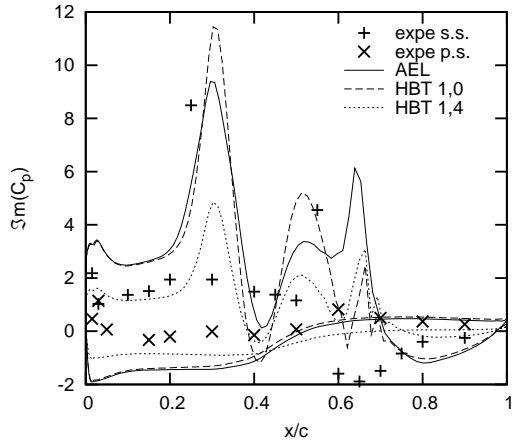


(e) Section 5 - 82.5%

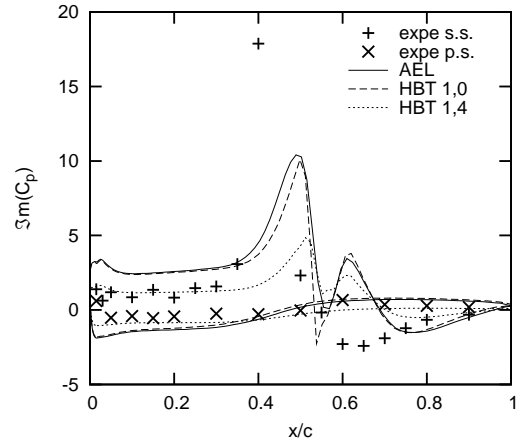


(f) Section 6 - 95%

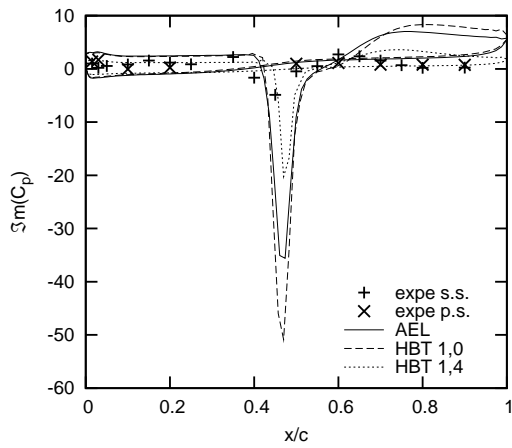
Figure 7.18: LANN ct9: Real part of C_p for (1,4).



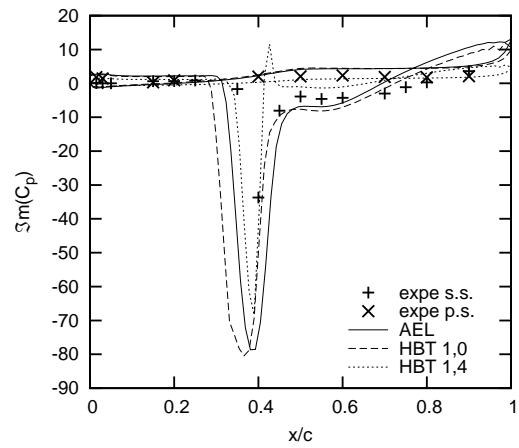
(a) Section 1 - 20%



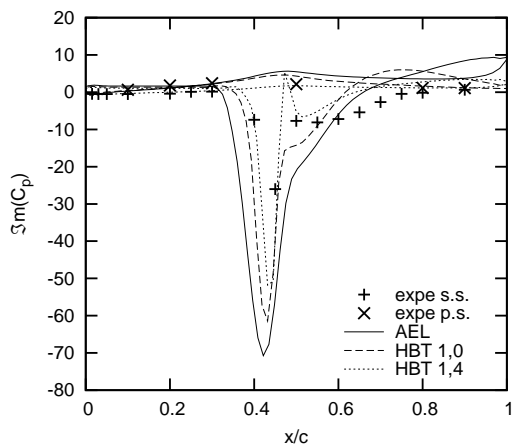
(b) Section 2 - 32.5%



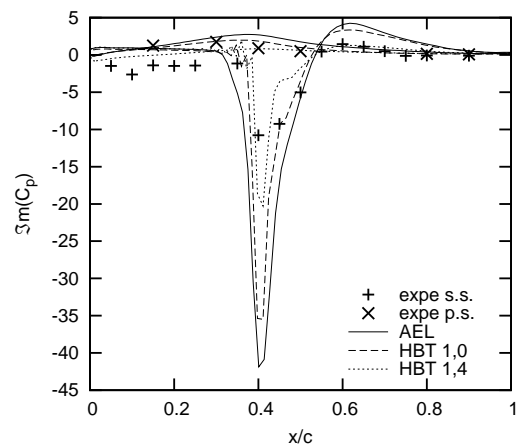
(c) Section 3 - 47.5%



(d) Section 4 - 65%

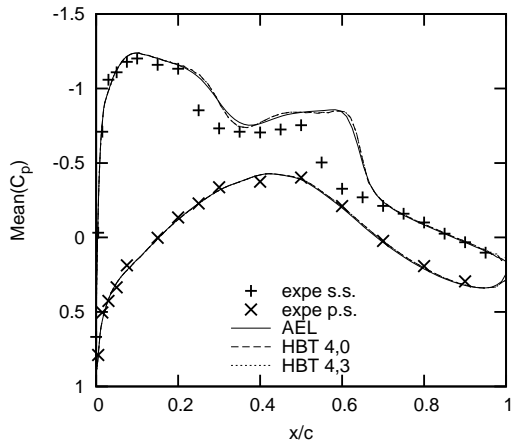


(e) Section 5 - 82.5%

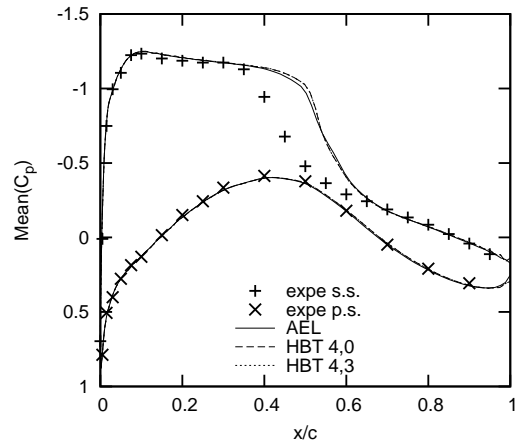


(f) Section 6 - 95%

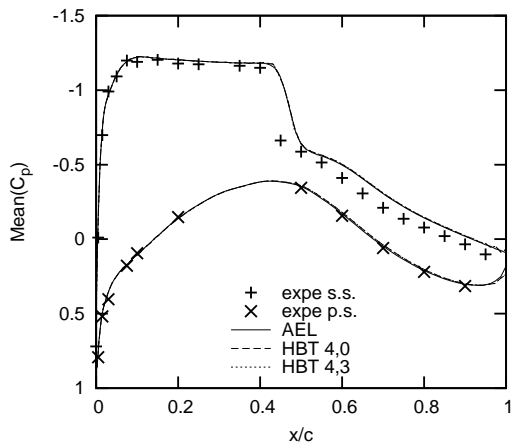
Figure 7.19: LANN CT9: Imaginary part of C_p for (1,4).



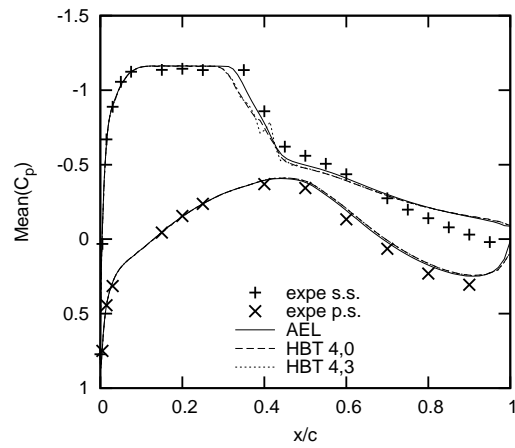
(a) Section 1 - 20%



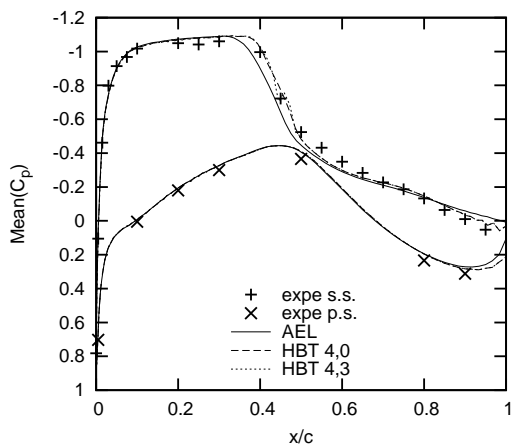
(b) Section 2 - 32.5%



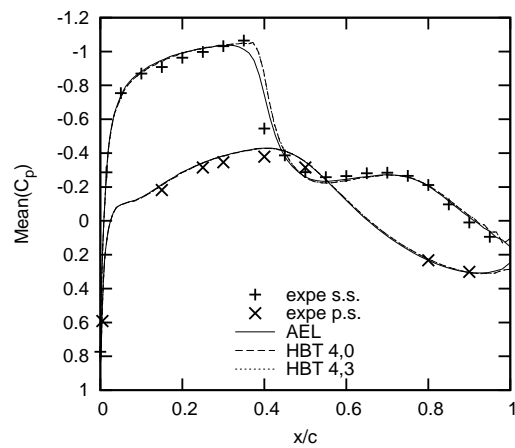
(c) Section 3 - 47.5%



(d) Section 4 - 65%

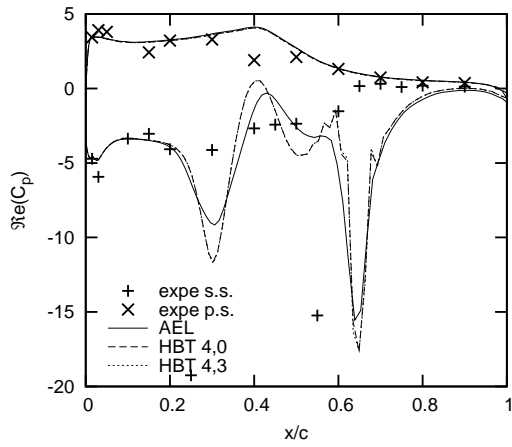


(e) Section 5 - 82.5%

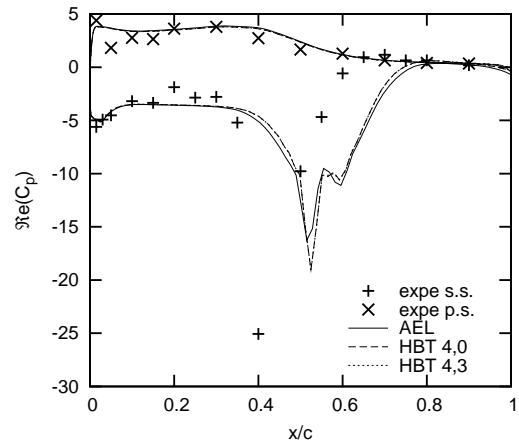


(f) Section 6 - 95%

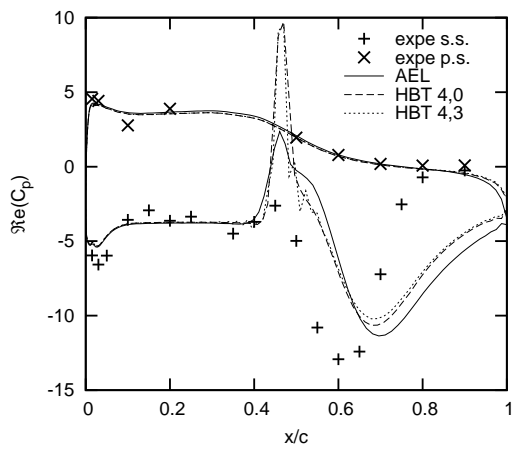
Figure 7.20: LANN CT9: Mean pressure coefficient C_p for (4,3).



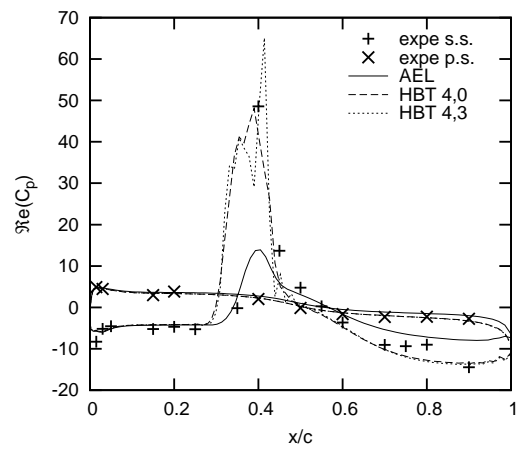
(a) Section 1 - 20%



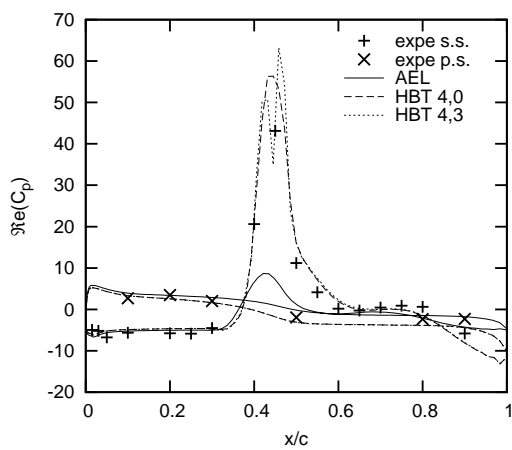
(b) Section 2 - 32.5%



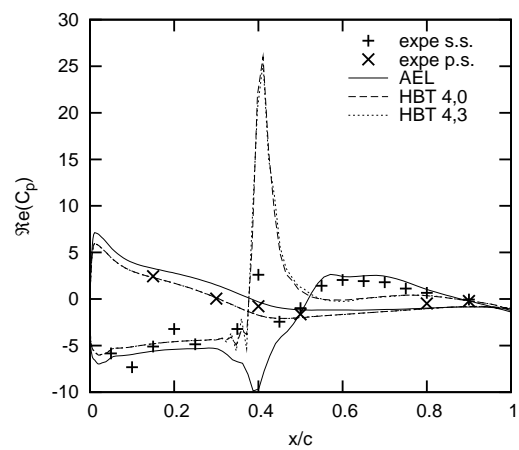
(c) Section 3 - 47.5%



(d) Section 4 - 65%

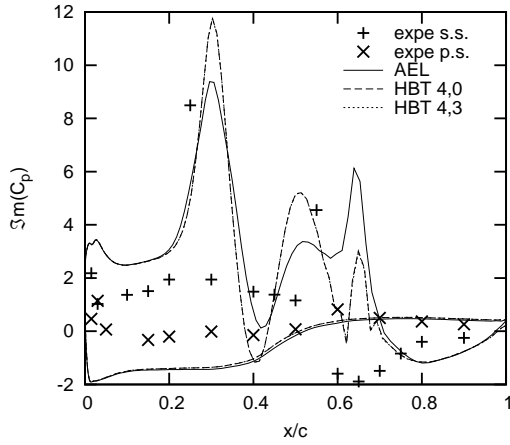


(e) Section 5 - 82.5%

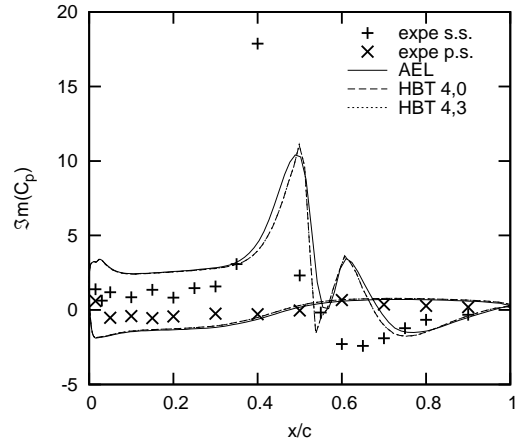


(f) Section 6 - 95%

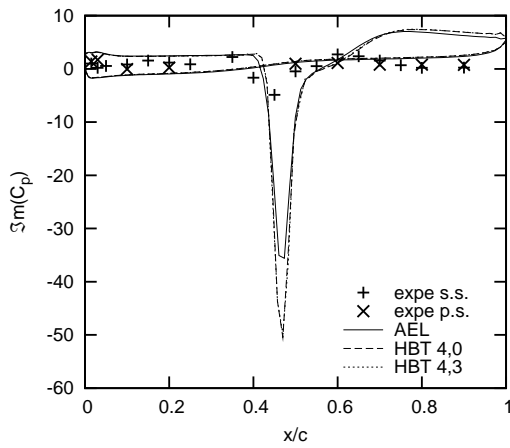
Figure 7.21: LANN ct9: Real part of C_p for (4,3).



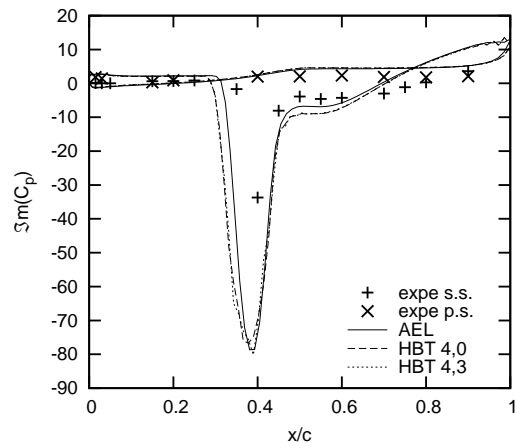
(a) Section 1 - 20%



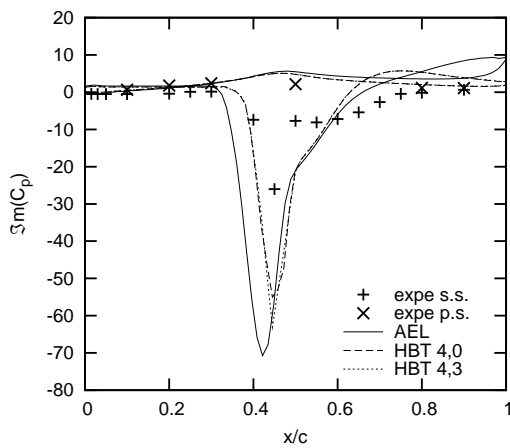
(b) Section 2 - 32.5%



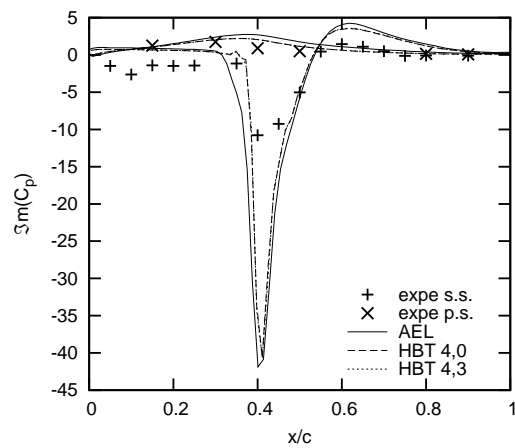
(c) Section 3 - 47.5%



(d) Section 4 - 65%

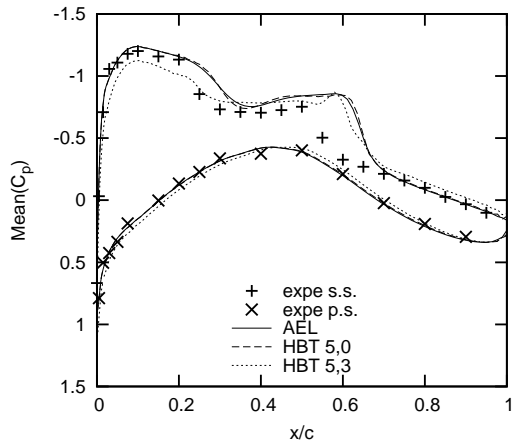


(e) Section 5 - 82.5%

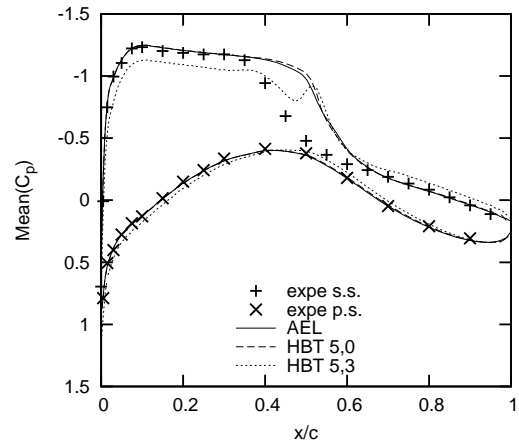


(f) Section 6 - 95%

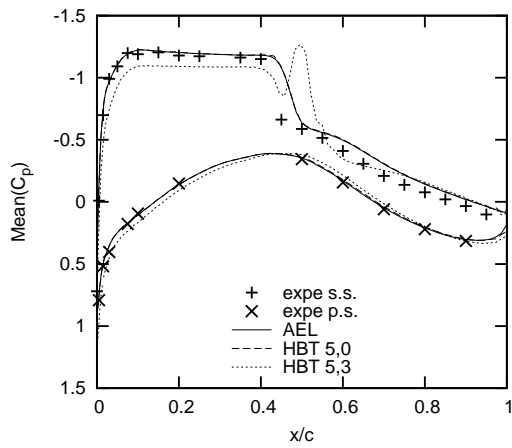
Figure 7.22: LANN CT9: Imaginary part of C_p for (4,3).



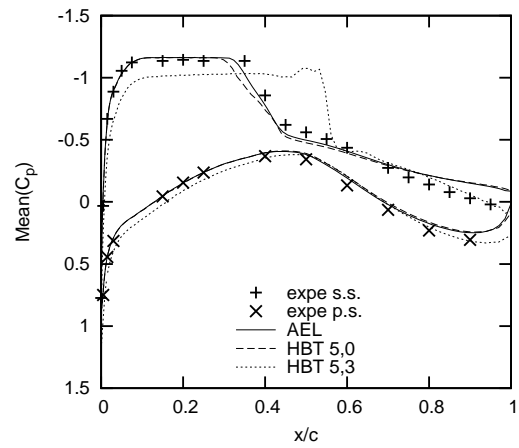
(a) Section 1 - 20%



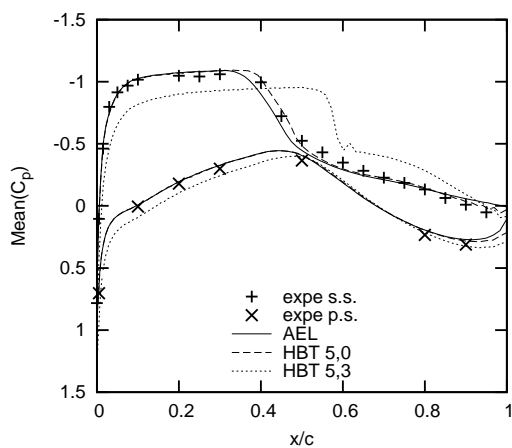
(b) Section 2 - 32.5%



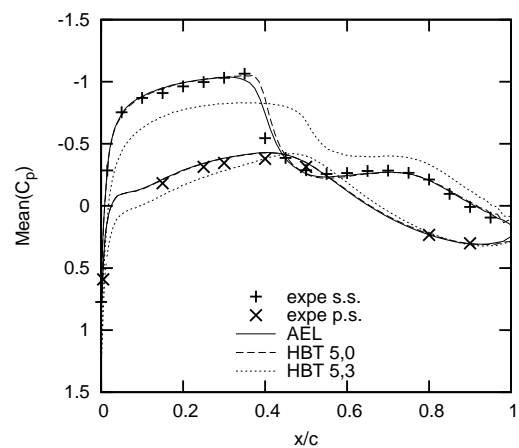
(c) Section 3 - 47.5%



(d) Section 4 - 65%

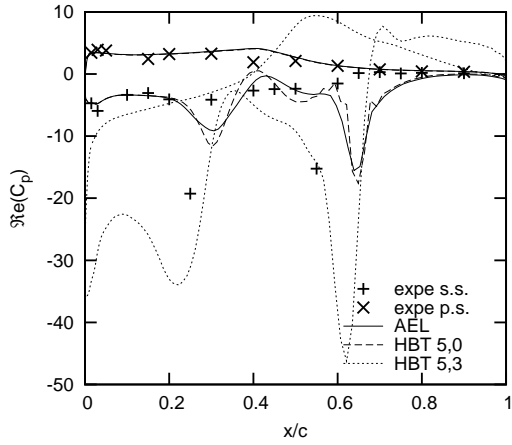


(e) Section 5 - 82.5%

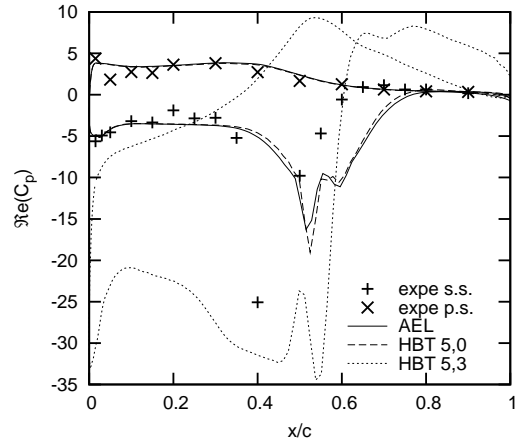


(f) Section 6 - 95%

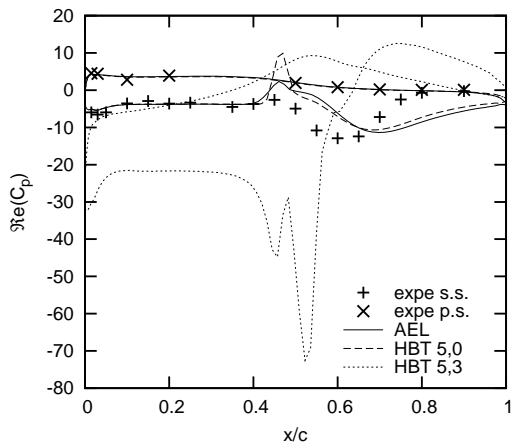
Figure 7.23: LANN CT9: Mean pressure coefficient C_p for (5,3).



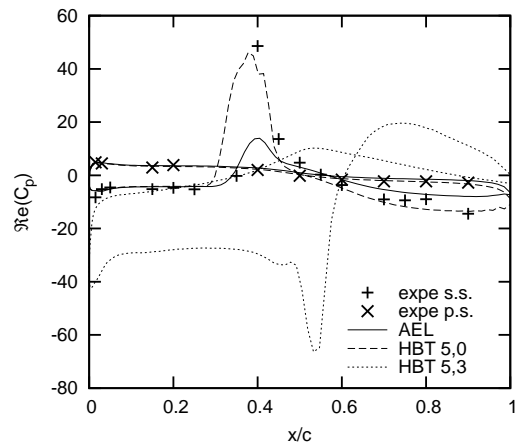
(a) Section 1 - 20%



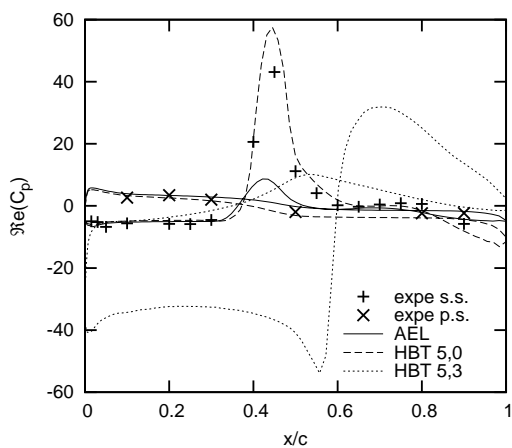
(b) Section 2 - 32.5%



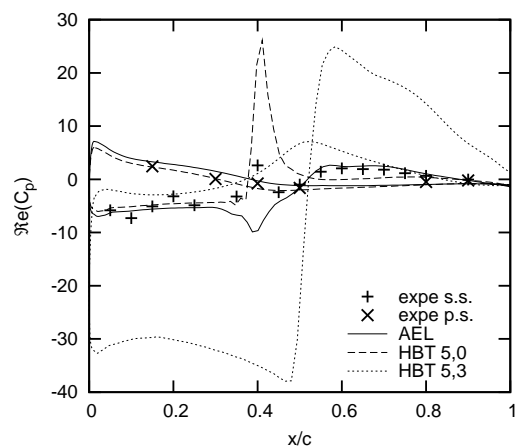
(c) Section 3 - 47.5%



(d) Section 4 - 65%

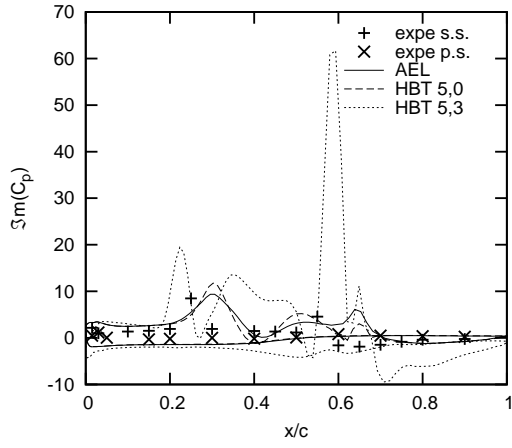


(e) Section 5 - 82.5%

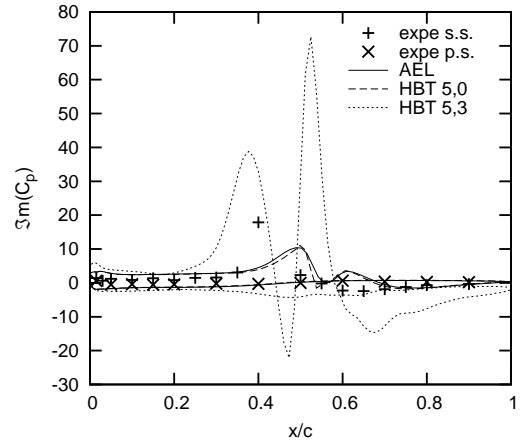


(f) Section 6 - 95%

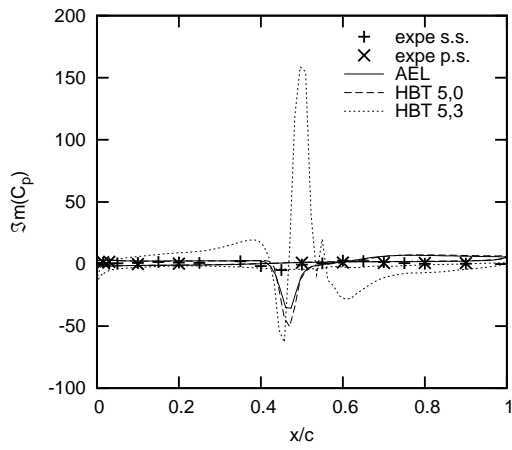
Figure 7.24: LANN ct9: Real part of C_p for (5,3).



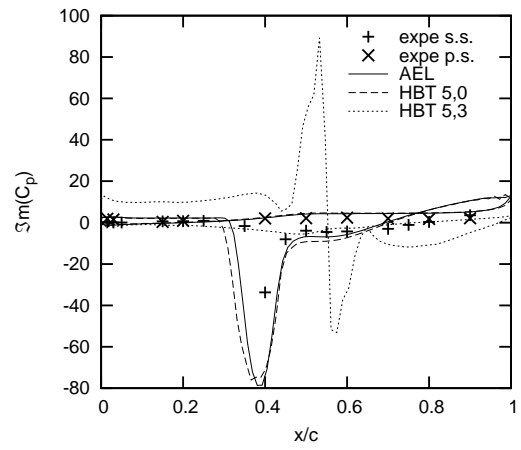
(a) Section 1 - 20%



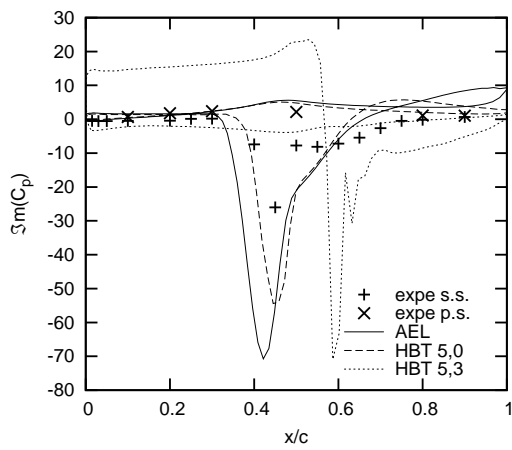
(b) Section 2 - 32.5%



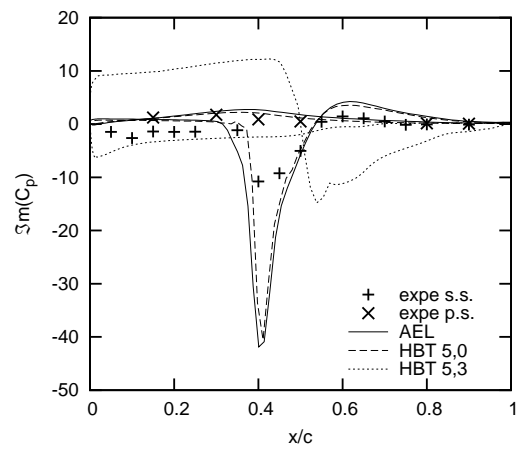
(c) Section 3 - 47.5%



(d) Section 4 - 65%



(e) Section 5 - 82.5%



(f) Section 6 - 95%

Figure 7.25: LANN CT9: Imaginary part of C_p for (5,3).

7.3 Summary

This chapter gives an insight on what is possible with the Multifrequential Harmonic Balance Technique and what cannot be done. Clearly, not just any distribution of instants may be chosen. Better results can be obtained, but for many instant distributions, they are rather worse.

The physical aspects of the problem must be taken in account. This means, by way of example, that if a shock is in the flow field, the chosen instants must be able to actually capture the shock. Otherwise, one cannot trust the results. Distributions that are close to a uniform one, usually (but not necessarily) lead to good results and may suffice for most industrial needs.

Chapter 8

Conclusion and perspectives

The Multifrequential Harmonic Balance Technique is an expansion of the Monofrequential HBT that is designed to compute periodic flows with several frequencies that are not multiples of a base frequency. Moreover, this technique also permits to choose instants that are not uniformly distributed over the period. The initially unsteady problem is transformed into several steady problems coupled among each other at the observed instants. The steady problems are then iterated with the help of a virtual time step until convergence is reached. This convergence is obtained quite quickly, since all techniques for steady simulations can be used to accelerate convergence (local time stepping, multigrid).

The validation of the newly added code showed that the monofrequential and the multifrequential approach may be considered to be equal for cases with a single frequency (and its harmonics) and uniformly distributed time steps over the period. This is proved by comparison of the pressure coefficient distribution for various two and three dimensional test cases. The deviation observed on the residuals is negligibly small and does not increase throughout the course of iteration. This deviation is due to a difference in the source term computation, that originates from the varying computation of the coefficient matrix. The monofrequential approach allows to compute analytically the coefficients, while the multifrequential approach must perform a matrix inversion, a matrix linearization and a matrix product to obtain the coefficients. All operations reduce the precision with respect to the analytic computation. The performance of both approaches is also similar: the raise in computing time, compared to a monofrequential computation, is less than 3% and therefore acceptable.

Using non-uniformly distributed instants is possible, but may influence the results to a lesser or greater extent depending on the flow properties. Hence, the choice of the distribution must be based on several issues. First of all, matrix A_1 must be invertible. A *Maple* program is provided and described in this report (cf. Chap. 4), that allows to define the instants in advance. However, if matrix A_1 still disposes of a small determinant ($|A_1| < 1.0$), problems do occur. The determinant mathematically describes the quality of the physical distribution of the instants. Choosing a very small period, in which the instants must be defined, or picking instants, that are very far away from a uniform distribution over the greatest period, may lead to oscillations and bad results. This is particularly true for high order HBT computations (i.e. an elevated number of frequencies taken

in account). Therefore, one cannot use just any distribution of instants: the physical aspect of the problem, i.e. shocks or boundary layer separation and particularly their time depending position in unsteady problems, must be considered.

Actually, the main conclusion from these experiences is, that a uniform distribution should be chosen, unless

- the problem imposes a different one due to external or internal physical boundaries (this kind of problems appear in current turbo machine research) or,
- the constraint of matrix A_1 to be invertible cannot be complied due to the flow's inherent frequencies.

The work that is presented in this report mainly consists in the implementation of the Multifrequential HBT in the *elsA* code and proves that the approach does work. Also, some examples are given on what happens, if a non-uniform distribution is chosen. Given these results, the next natural step is now to try the multifrequential code on an actual problem, where the flow is subjected to several frequencies that are not multiples of each other (e.g. current multi-stage turbo machinery applications). Furthermore, a test case, that does not allow uniformly distributed instants should be computed and compared to URANS computation in matters of the results' quality and computing time. That way, the advantages of the Multifrequential HBT are emphasized with regard to an industrial application. Furthermore, the approach could eventually be extended and results might be improved by using more than $(2N + 1)$ instants (N : number of frequencies). *Ekici and Hall* propose this approach in their articles (cf. [5, 6]). 50% more instants ($(3N + 1)$ instants) are used and the transformation matrix is not square anymore. Hence, its pseudoinverse must be found, in order to transform back to the time domain. According to *Ekici and Hall*, aperiodic flows may be simulated with this approach as well.

Bibliography

- [1] Julien Bohbot. *Simulation numérique d'écoulements transsoniques autour de voilures mobiles et analyse aéroélastique du flottement de sections d'ailes*. PhD thesis, ENSAM, 2002.
- [2] L. Cambier and M. Gazaix. elsA: an efficient object-oriented solution to CFD complexity. In *40th AIAA Aerospace Science Meeting and Exhibit, Reno*, pages 14–17. AIAA 2002-0108, January 2002.
- [3] S. S. Davis. NACA 64A010 (NASA Ames Model) Oscillatory Pitching. Technical Report 702, AGARD, 1982. Data Set 2.
- [4] Julien Delbove. *Contribution aux outils de simulation aéroélastique des aéronefs : prédiction du flottement et déformation statique des voilures*. PhD thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace, 2005.
- [5] K. Ekici and K. C. Hall. Nonlinear Analysis of Unsteady Flows in Multistage Turbomachines Using the Harmonic Balance Technique. In *44th AIAA Aerospace Sciences Meeting*, Reno, Nevada, January 2006. AIAA Paper 2006-422.
- [6] K. Ekici and K. C. Hall. Nonlinear Frequency-Domain Analysis of Unsteady Flows in Turbomachinery with Multiple Excitation Frequencies. In *24th AIAA Applied Aerodynamics Conference*, San Francisco, California, June 2006. AIAA Paper 2006-2995.
- [7] C. Gasquet and P. Wittomski, editors. *Analyse de Fourier et Applications: filtrage, calcul numérique, ondelettes*. MASSON, Paris, Milan, Barcelone, 1995.
- [8] A. Gopinath and A. Jameson. Time Spectral Method for Periodic Unsteady Computations over Two- and Three- Dimensional Bodies. In *43rd Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2005. AIAA Paper 2005-1220.
- [9] A. Gopinath and A. Jameson. Application of the Time Spectral Method to Periodic Unsteady Vortex Shedding. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2006. AIAA Paper 2006-0449.
- [10] K. C. Hall, J. P. Thomas, and W. S. Clark. Computation of Unsteady Nonlinear Flows in Cascades using a Harmonic Balance Technique. *AIAA Journal*, 40(5):879–886, 2002.
- [11] M. McMullen, A. Jameson, and J. Alonso. Acceleration of Convergence to a Periodic Steady State in Turbomachinery Flows. In *39th Aerospace Sciences Meeting*, Reno, Nevada, January 2001. AIAA Paper 2001-0152.

- [12] S. Nadarajah, M. McMullen, and A. Jameson. Optimum Shape Design for Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods. In *16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida, June 2003. AIAA Paper 2003-3875.
- [13] F. Sicot. *Simulation efficace des problèmes de génération de bruit de raies rencontrés dans les turbomachines - Rapport première année*. PhD thesis, CERFACS CFD/AAM Toulouse, January 2008.
- [14] P. R. Spalart and S. R. Allmaras. A One-Equation Turbulence Transport Model for aerodynamic flows. In *30th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1992. AIAA Paper 92-0439.
- [15] M. A. Spiker, J. P. Thomas, R. E. Kielb, K. C. Hall, and E. H. Dowell. Modeling Cylinder Flow Vortex Shedding with Enforced Motion Using a Harmonic Balance Approach. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials (SDM) Conference*, Newport, RI, May 2006. AIAA Paper 2006-1965.
- [16] G.R. Srinivasan, J.A. Ekaterinaris, and W.J. McCroskey. Evaluation of turbulence models for unsteady flows of an oscillating airfoil. *Computers and Fluids*, 24(7):833–861, September 1995.
- [17] J. P. Thomas, E. H. Dowell, and K. C. Hall. Modeling Viscous Transonic Limit Cycle Oscillation Behavior Using a Harmonic Balance Approach. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, Colorado, April 2002. AIAA Paper 2002-1414.
- [18] E. van der Weide, A. Gopinath, and A. Jameson. Turbomachinery Applications with the Time Spectral Method. In *35th AIAA Fluid Dynamics Conference and Exhibit*, Toronto, Ontario, Canada, June 2005. AIAA Paper 2005-4905.
- [19] Ir R. J. Zwann. LANN Wing. Pitching Oscillation. Technical Report 702, AGARD, 1982. Data Set 9.

Appendix: Structograms

Maple program

The main structogram shown in this section describes the code of the *Maple* program to determine in advance non-uniformly distributed instants for the Multifrequential HBT. Below the main program, the reader finds a selection of the external procedures called in this program. These procedures build the sub-matrices with a certain dimension out of matrix A_1 and compute their determinant. Not all of the procedures are displayed in this report, because a procedure for a sub-matrix of elevated dimension can easily be obtained by modification of the scheme applied for lower dimensions. This is why only two procedures for matrix dimensions 2×2 and 3×3 are given in this report.

Main Program

Restart (clear all variables)
Change to current directory in order to be able to load external procedures
Load <i>LinearAlgebra</i> - and <i>codegen</i> -package
$pi := evalf(Pi, 16)$ π with 16 decimals
$N := 2$ Number of frequencies (order)
$ni := 20$ Set number of intervals in which a solution is searched
$smd := 20$ Set dimension of current sub-matrix
$LastInst := false$ Last instant (t_2) to determine? -no: false, -yes: true
$w := Vector(N + 1)$ Define vector of frequencies $\omega[i] := 2 \cdot \pi \cdot f_i$ Define all pulsations with the known frequencies
... is continued on the following page

Main Program (continued)

$t := \text{Vector}(2 \cdot N + 1)$ Define the vector of instants $\tau[i] := \dots$ Define the instants already known (starting configuration: $\tau_0 := 0, \tau_{2 \cdot N}$)	
$A := \text{Matrix}(2 \cdot N + 1, 2 \cdot N + 1)$ Allocate Matrix A	
for $i = 1$ to $(2 \cdot N + 1)$	
for $j = 1$ to $(N + 1)$	
$A[i, j] := \cos(w[j, 1] \cdot t[i, 1])$	
for $j = N + 2$ to $(2 \cdot N + 1)$	
$A[i, j] := \sin(w[j - N, 1] \cdot t[i, 1])$	
$N\text{SubMat} := \binom{2 \cdot N + 1}{\text{smd}}$ Number of sub-matrices to build	
$(\text{smd} = 2) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure2x2"; proc2x2($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 3) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure3x3"; proc3x3($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 4) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure4x4"; proc4x4($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 5) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure5x5"; proc5x5($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 6) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure6x6"; proc6x6($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 7) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure7x7"; proc7x7($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
$(\text{smd} = 8) \text{ AND } (\text{LastInst} = \text{false})$	
yes	no
read "procedure8x8"; proc8x8($N, \text{smd}, A, N\text{SubMat}$)	\emptyset
<i>... is continued on the following page</i>	

Main Program (continued)

(smd = 9) AND (LastInst = false)	
yes	no
read "procedure9x9"; proc9x9(N, smd, A, NSubMat)	∅
LastInst = true	
yes	no
DetVect := Vector(1)	∅
DetVect[1] := Determinant(A)	
LastInst = true	
yes	no
NDet := 1	NDet := NSubMat
SolveVar := τ[1]	SolveVar := τ[((2 · N + 1) - smd)]
SolMat := Matrix(NDet, ni); Allocate matrix into which solutions are written	
for i = 1 to NDet	
for p = 1 to ni	
SolMat[i, p] := fsolve(DetVect[i] = 0, SolveVar, (p - 1)/ni · τ _{2.N} · ((p)/ni · τ _{2.N} + 0.001))	
for i = 1 to NDet	
for j = 1 to ni	
type(SolMat[i, j], float) ≠ true	
yes	no
SolMat[i, j] := 0	∅
len := 1000	
k := 1	
SolVec := Vector(len) Allocate solution vector	
for i = 1 to NDet	
for j = 1 to ni	
check := true	
SolMat[i, j] ≠ 0	
yes	no
for l = 1 to len	check := false
SolMat[i, j] = SolVec[l]	∅
yes	
check := false	∅
check = true	
yes	no
SolVec[k] := SolMat[i, j]	∅
k := k + 1	
... is continued on the following page	

Main Program (continued)

$SolVec := sort(SolVec)$	
$counter := 0$	
for $i = 1$ to len	
$SolVec[i] \neq 0$	
yes	no
$counter := counter + 1$	\emptyset
$solution[counter] := SolVec[i]$	
for $i = 1$ to $counter$	
$solution[i]$	
Print the values the current instant must not adopt	

Procedure proc2x2

Input
integer $\{N, smd, NSubMat\}$
real $\{A\}$
Global (output)
real $\{DetVect\}$
Local
integer $\{h, count, a, j, k\}$
real $\{H\}$
$h := (2 \cdot N + 1) - (smd - 1)$ Indicates the row of Matrix A , where the sub-matrices start
$H := Matrix(smd, smd)$ Initialises sub-matrix H of dimension $(smd \times smd)$ (here: 2×2)
$DetVect := Vector(NSubMat)$ Initialises the vector of sub-determinants of dimension $NSubmat$
$count := 0, \quad a := 0, \quad j := 0, \quad k := 0$ Auxiliary variables
for $j = 1$ to $(2 \cdot N)$
for $k = j + 1$ to $(2 \cdot N + 1)$
for $a = 1$ to smd
$H[a, 1] := A[h + a - 1, j]$
$H[a, 2] := A[h + a - 1, k]$
$DetVect[count] := Determinant(H)$
$count := count + 1$

Procedure proc3x3

Input
integer $\{N, smd, NSubMat\}$ real $\{A\}$
Global (output)
real $\{DetVect\}$
Local
integer $\{h, count, a, j, k, l\}$ real $\{H\}$
$h := (2 \cdot N + 1) - (smd - 1)$ Indicates the row of Matrix A , where the sub-matrices start
$H := Matrix(smd, smd)$ Initialises sub-matrix H of dimension $(smd \times smd)$ (here: 3×3)
$DetVect := Vector(NSubMat)$ Initialises the vector of sub-determinants of dimension $NSubmat$
$count := 0, \quad a := 0, \quad j := 0, \quad k := 0, \quad l := 0$ Auxiliary variables
for $j = 1$ to $(2 \cdot N - 1)$
for $k = j + 1$ to $(2 \cdot N)$
for $l = k + 1$ to $(2 \cdot N + 1)$
for $a = 1$ to smd
$H[a, 1] := A[h + a - 1, j]$
$H[a, 2] := A[h + a - 1, k]$
$H[a, 3] := A[h + a - 1, l]$
$DetVect[count] := Determinant(H)$
$count := count + 1$

elsA programs

The programs' struktograms depicted in this section are described in detail in chapter 5 and account for the programming realised in the *elsA* code.

createSou::createSouHbt

multifrequ. frequency_01 defined	
no	yes
Monofrequential assign to <i>freq</i> the monofrequential base frequency; return new SouHBT(<i>order</i> , <i>freq</i> , <i>step</i> , <i>brother</i> , <i>cBlock</i>)	Multifrequential Field <i>freqMulti</i> (<i>order</i> + 1); Field <i>instMulti</i> (2 * <i>order</i> + 1); <i>freqMulti</i> [0] = 0.0 for <i>i</i> = 1 to <i>order</i> assign to each entry of <i>freqMulti</i> (<i>i</i>) the corresponding frequency from the Python initialising files for <i>i</i> = 0 to (2 * <i>order</i> + 1) assign to each entry of <i>instMulti</i> (<i>i</i>) the corresponding instant from the Python initialising files return new SouHbtMultiFreq(<i>order</i> , <i>freqMulti</i> , <i>instMulti</i> , <i>step</i> , <i>brother</i> , <i>cBlock</i>)
∅	

SouHbtMultiFreq::SouHbtMultiFreq

allocate vectors <i>_fMulti</i> ; <i>_timeLevels</i> ; allocate arrays <i>_fMulti</i> [_brother] = <i>newFldArrayF</i> (<i>order</i> + 1); <i>_timeLevels</i> [_brother] = <i>newFldArrayF</i> (2 * <i>order</i> + 1)
for <i>i</i> = 0 to (<i>order</i>) (* <i>_fMulti</i> [_brother])[<i>i</i>] = <i>freqmulti</i> [<i>i</i>] assign instants of each brotherhood
for <i>i</i> = 0 to (2 * <i>order</i> + 1) (* <i>_timeLevels</i> [_brother])[<i>i</i>] = <i>instmulti</i> [<i>i</i>] assign instants of each brotherhood
<i>_dCoeff</i> [_brother] = <i>computeDCoeff</i> (<i>order</i>)

SouHbtMultiFreq::compute

Objects	{listBnd, listJoin, fldout}		
get number of grid cells: cellNbTot get grid dimensions: im, jm, km get number of equations: eqNb, eqNbTot			
firstVar = first Variable of mean of turbulent field (ρ)			
loop over all source terms to compute the sum of the source term			
same system (mean or turbulent)	yes	no	
same brotherhood	yes	no	\emptyset
same multigrid level	yes	no	
call updateSourceTerm(cellNbTot, im, jm, km, eqNbTot, eqNb, firstVar, loopSouHbt, loopGrid, fldOut)	\emptyset	\emptyset	

SouHbtMultiFreq::updateSourceTerm

Objects	{cellNbTot, im, jm, km, eqNbTot, eqNb, firstVar, loopSouHbt, loopGrid, fldout}
get dCoeff (D-coefficient matrix)	
l = current instant (determines the line in the dCoeff) m = time step of the element in the source term list (determines column in dCoeff) d = dCoeff(l,m)	
call souhbt_(cellNbTot, im, jm, km, eqNbTot, eqNb, firstVar, -d, solution, fldOut) (performs the multiplication of d and the conservative variables)	

SouHbtMultiFreq::computeDCoeff

Objects	{order}
allocate arrays $MatrixA(2 * order + 1, 2 * order + 1);$ $MatrixAInv(2 * order + 1, 2 * order + 1);$ $MatrixdAdt(2 * order + 1, 2 * order + 1);$ $dCoeff(2 * order + 1, 2 * order + 1)$	
call souhbtmultifreqa_ (order, _fMulti[_brother], _timeLevels[_brother], MatrixA)	
call souhbtmultifreqdadt_ (order, _fMulti[_brother], _timeLevels[_brother], MatrixdAdt)	
call souhbtmultifreqainv_ (order, MatrixA, MatrixAInv)	
call souhbtmultifreqd_ (order, MatrixA, MatrixAInv, dCoeff)	
return dCoeff	

Function SouHbtMultiFreq::getMaxFrequency

Objects	{brotherhood}
monofrequ. frequency $\neq 0$	
yes	no
0	\emptyset
(size of _fMulti < brotherhood	
yes	no
0	\emptyset
_fMulti[brotherhood] is nullpointer	
yes	no
0	\emptyset
currentFreq = _fMulti[brotherhood] (vector of frequencies) index = index of the maximum frequency in _fMulti[brotherhood]	
return currentFreq[index] (maximum frequency)	

Subroutine souhbtmultifreqa

C_IN	integer	{ <i>order</i> }
	real	{ <i>frequency</i> (1 : (<i>order</i> + 1))}
	real	{ <i>instant</i> (1 : (2 · <i>order</i> + 1))}
C_OUT	real	{ <i>A</i> ((2 · <i>order</i> + 1) : (2 · <i>order</i> + 1))}
C_LOC	integer	{ <i>i, j, n</i> }
$n = 2 \cdot \textit{order} + 1$		
for <i>i</i> = 1, <i>n</i>		
for <i>j</i> = 1, (<i>order</i> + 1)		
$A(i, j) = \cos(2 \cdot \pi \cdot \textit{frequency}(j) \cdot \textit{instant}(i))$		
for <i>j</i> = (<i>order</i> + 2), <i>n</i>		
$A(i, j) = \sin(2 \cdot \pi \cdot \textit{frequency}(j - \textit{order}) \cdot \textit{instant}(i))$		

Subroutine souhbtmultifreqdadt

C_IN	integer	{ <i>order</i> }
	real	{ <i>frequency</i> (1 : (<i>order</i> + 1))}
	real	{ <i>instant</i> (1 : (2 · <i>order</i> + 1))}
C_OUT	real	{ <i>dAdt</i> (1 : (2 · <i>order</i> + 1), 1 : (2 · <i>order</i> + 1))}
C_LOC	integer	{ <i>i, j, n</i> }
$n = 2 * \textit{order} + 1$		
for <i>i</i> = 1, <i>n</i>		
for <i>j</i> = 1, (<i>order</i> + 1)		
$dAdt(i, j) = -2 \cdot \pi \cdot \textit{frequency}(j) \cdot \sin(2 \cdot \pi \cdot \textit{frequency}(j) \cdot \textit{instant}(i))$		
for <i>j</i> = (<i>order</i> + 2), <i>n</i>		
$dAdt(i, j) = 2 \cdot \pi \cdot \textit{frequency}(j - \textit{order}) \cdot \cos(2 \cdot \pi \cdot \textit{frequency}(j - \textit{order}) \cdot \textit{instant}(i))$		

Subroutine souhbtmultifreqainv

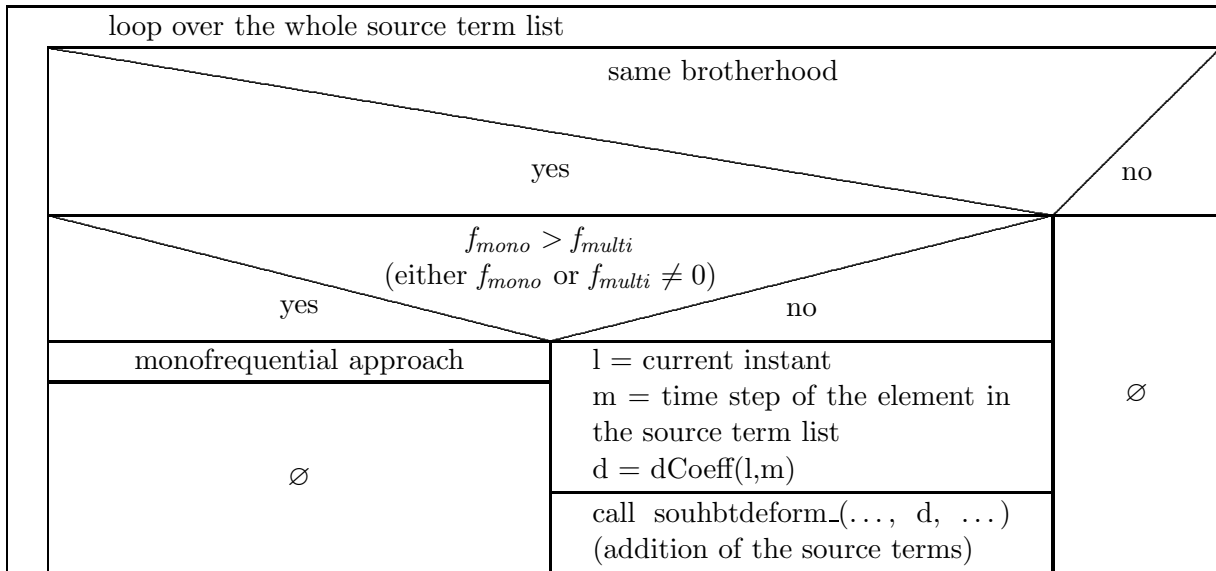
C_IN	integer	{ <i>order</i> }
	real	{ $B(1 : (2 \cdot \textit{order} + 1), 1 : (2 \cdot \textit{order} + 1))$ }
C_OUT	real	{ $AInv(1 : (2 \cdot \textit{order} + 1), 1 : (2 \cdot \textit{order} + 1))$ }
C_LOC	integer	{ <i>i, j, n, INFO, LWORK</i> }
	integer	{ <i>IPIV</i> (1 : 2 · <i>order</i> + 1)
	real	{ <i>WORK</i> (1 : 2 · <i>order</i> + 1)
$n = 2 \cdot \textit{order} + 1$		
$LWORK = n$		
CALL DGETF2(<i>n, n, B, n, IPIV, INFO</i>)		
CALL DGETRI(<i>n, B, n, IPIV, WORK, LWORK, INFO</i>)		
for <i>i</i> = 1, <i>n</i>		
for <i>j</i> = 1, <i>n</i>		
$A(i, j) = B(i, j)$		

The subroutines DGETF2 and DGETRI are included in dgetrf.f and dgetri.f respectively of the LAPACK - Linear Algebra PACKage (<http://www.netlib.org/lapack/>): dgetri.f computes the inverse of a general matrix, using the LU factorisation, computed by dgetrf.f. dgetrf.f computes an LU factorisation of a general matrix, using partial pivoting with row interchanges.

Subroutine souhbtmultifreqd

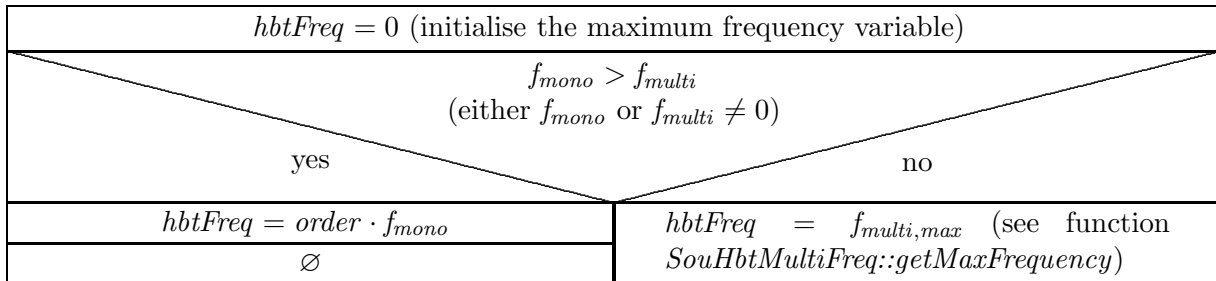
C_IN	integer	{ <i>order</i> }
	real	{ $dAdt(1 : (2 \cdot \textit{order} + 1), 1 : (2 \cdot \textit{order} + 1))$ }
	real	{ $AInv(1 : (2 \cdot \textit{order} + 1), 1 : (2 \cdot \textit{order} + 1))$ }
C_OUT	real	{ $dCoeff(1 : (2 \cdot \textit{order} + 1), 1 : (2 \cdot \textit{order} + 1))$ }
C_LOC	integer	{ <i>i, j, k, n</i> }
$n = 2 \cdot \textit{order} + 1$		
for <i>i</i> = 1, <i>n</i>		
for <i>j</i> = 1, <i>n</i>		
$dcoeff(i, j) = 0$		
for <i>k</i> = 1, <i>n</i>		
$dcoeff(i, j) = dcoeff(i, j) + dAdt(i, k) \cdot AInv(k, j)$		

TmoSetOfSolver::computeLhs



The D-coefficients are needed in the implicit phase. This is why the above differentiation between the monofrequential and the multifrequential approach has to be inserted.

TmoTimeStepLocalConvGridMotion::computeLocalTimeStep



In *computeLocalTimeStep* the subroutine *compcellspecradius2* is called and expects as an input the maximum occurring frequency, this is why the above query has to be inserted.

TmoCpCellSpecRadius2F.for

The only change that has to be effected in *TmoCpCellSpecRadius2F.for* is the calculation of the correction term k_p . It changes from $k_p = 2\pi \cdot N \cdot f$ to $k_p = 2\pi \cdot f_{max}$, because the input for the FORTRAN subroutine *compcellspecradius2* is not anymore the frequency, but the maximum occurring frequency now.