

Modification of the *Regula Falsi* method to accelerate system convergence in the prediction of trace quantities of atmospheric pollutants

J.B. Phillips*, A.S. Menawat¹, S.R. Carden

Engineering Development Institute, 7925-A N. Oracle 155, Tucson, AZ 85704, USA

Received 16 January 1995; accepted in revised form 6 March 1995

Abstract

The *Regula Falsi* method is applied to prediction of trace quantities of air pollutants produced by combustion reactions such as those found in industrial point sources. Equilibrium quantities of uncombusted fuel generally are quite small, due to the exothermic nature of combustion reactions. Accordingly, calculating the maximum theoretical efficiency which may be achieved in a combustion process is difficult since the equations describing the equilibrium state are stiff.

After reducing the system to that of one equation and one unknown, the *Regula Falsi* method may be applied in its modified form. The *Regula Falsi* method is modified through the introduction of an under-interpolation factor, which accelerates convergence of the numerical procedure in a manner similar to the Marquardt method of successive over-relaxation (SOR). Analogously, such a modification of the *Regula Falsi* method is referred to in this work as successive under-interpolation (SUI).

Results indicate an excellent correlation between the under-interpolation factor and the number of iterations required to converge upon a solution.

Keywords: Numerical methods; *Regula Falsi* method; Combustion; Equilibrium; Interpolation

1. Introduction

Emissions of uncombusted substances from industrial point sources containing a combustion reaction have received increased attention recently as air pollution has been brought to the forefront of environmental concerns. This is in large part due to the Clean Air Act Amendments of 1990 and regulations such as the Benzene NESHAP (National Emission Standards for Hazardous Air Pollutants) Regulation

* Corresponding author. E-mail: drjbond@aol.com. Tel.: (414) 365-9351.

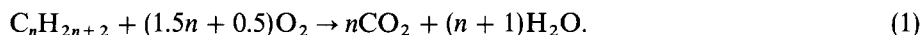
¹ Current address: Aeroquip Corporation, 2323 Green Rd, Ann Arbor, MI 98105, USA.

[1] and the HON (Hazardous Organic NESHAP) Rule [2] which have been promulgated by the Environmental Protection Agency (EPA) as a result of the Clean Air Act. Specific point sources of interest include boilers, furnaces, incinerators, and internal combustion engines, to name a few.

In 1985, the EPA, working in conjunction with the State of California, published the AP-42 emission factors [3], which allowed prediction of uncombusted substances from point sources such as those mentioned previously (*vide supra*), as well as compressors, storage tanks, rail tank cars, tank trucks, service stations, motor vehicle tanks, barges, FCC units, cokers, fugitive emissions, and other industrial equipment items. The AP-42 emission factors allow facile computation of emissions by multiplying the appropriate emission factor by the throughput. For example, the emission factors for combustion of natural gas are reported in units of kilograms of emissions per million cubic meters of fired material. Emissions are broken down into particulate matter, sulfur dioxide, sulfur trioxide, nitrogen oxides, and volatile organic compounds (speciated into methane emissions and nonmethane emissions).

The AP-42 emission factors were determined by stack tests and equivalent measurements. Accordingly, their use represents an accurate method of estimating emissions from equipment which was tested in determining said emission factors. However, the measurements used in determining the AP-42 emission factors were made over a decade ago. Advances in combustion equipment efficiency have led many operating companies to abandon the practice of using the AP-42 data to measure emissions from their equipment, since use of the AP-42 emission factors can result in over-reporting emissions from more modernized equipment with improved combustion efficiencies.

This leads to the question of the maximum efficiency which might be obtained, and the minimum VOC emissions which might be realized from an industrial point source containing a combustion process. The combustion reaction of a saturated aliphatic hydrocarbon is as follows:



The standard enthalpy of reaction may be found by summing the products of the standard enthalpies of formation and the stoichiometric coefficients, i.e.,

$$\Delta H_{R_x}^0 = \sum (v_i \Delta H_{f,i}^0) \quad (2)$$

The standard enthalpies of formation for reactants and products may be found in the JANAF (Joint Army, Navy, Air Force) tables [4]. For the combustion of methane, the relevant data are as given in Table 1.

The enthalpies of formation are determined at standard conditions of 298 K and 1 atm. Therefore, the enthalpy of combustion which is calculated by the stoichiometrically weighted sum of the enthalpies of formation is also at standard conditions. This is significant because, of course, combustion reactions do not occur at standard conditions, but rather at or near the adiabatic flame temperature of the species being combusted.

Using the data from Table 1 in conjunction with heat capacity data, the adiabatic flame temperature for the combustion of methane, with some given percent excess

Table 1
Standard enthalpies of formation and combustion pertaining to combustion of methane

Species	Standard enthalpy of formation (kcal/mol)	Standard enthalpy of combustion (kcal/mol)
Methane	-17.889	-745.6
Carbon dioxide	-94.052	NA
Water	-57.7979	NA
Oxygen	0 (by definition)	NA

Table 2
Ideal gas heat capacity data used in the calculation of adiabatic flame temperature (C_p , J/mol; T , K)

	Oxygen	Water	Carbon dioxide	Nitrogen
a	29.8832	34.0471	19.0223	29.4119
b	-1.13842×10^{-2}	-9.65064×10^{-3}	7.96291×10^{-3}	-3.00681×10^{-3}
c	4.33779×10^{-5}	3.29983×10^{-5}	-7.37067×10^{-5}	5.45064×10^{-6}
d	-3.70082×10^{-8}	-2.04467×10^{-8}	3.74572×10^{-8}	5.13186×10^{-9}
e	1.01006×10^{-11}	4.30228×10^{-12}	-8.13304×10^{-12}	-4.30228×10^{-12}

theoretical air, can be calculated. Ideal gas heat capacity data used in the adiabatic flame temperature calculation were obtained [5] in the form

$$C_p = a + bT + cT^2 + dT^3 + eT^4. \quad (3)$$

Specifically, the ideal gas heat capacity data used in the calculation were those listed in Table 2.

It should be noted that the heat capacity data are not valid at temperatures above 1500 K. This bears on the problem at hand since attempting to integrate the heat capacity from an assumed inlet temperature of 311 K to the adiabatic flame temperature, which is well above 1500 K for methane and many other substances of interest, would lead to a physically unrealistic result due to the negative coefficient in the quartic term for nitrogen. Hence, the adiabatic flame temperature was calculated by evaluating the heat capacity at 1500 K and using this "average" heat capacity, i.e.,

$$T_{ad} = T_{in} - \frac{\Delta H_{Rx}^0}{C_{p,av}}. \quad (4)$$

The heat capacity pertains to the flue gas from the combustion process, i.e., the mole-weighted average of gases produced by 1 mol of methane.

Although combustion reactions usually are presumed to go to completion, the measurements on which the AP-42 emission factors are based demonstrated that some uncombusted fraction of fuel remains in the stack gas. While it has been demonstrated that combustion reactions do not proceed to the state of chemical equilibrium [6], this would represent the minimum theoretical emission level which could be expected from a combustion point source. Accordingly, if combustion equipment is found to produce emissions near the theoretical minimum, then

Table 3
Standard Gibbs free energies of formation and combustion pertaining to combustion of methane

Species	Standard Gibbs free energy of formation (kcal/mol)	Standard Gibbs free energy of combustion (kcal/mol)
Methane	– 12.14	– 199,234
Oxygen	0 (by definition)	NA
Carbon dioxide	– 94.26	NA
Water	– 54.6351	NA

Table 4
Calculated parameters pertaining to the combustion of methane

T_{ad} , K	K^0	K_{Tad}
2245	1.343×10^{146}	6.146×10^{32}

additional modifications to equipment design will yield little or no reduction in emissions.

It is illustrative to consider the maximum possible conversion of fuel that can be achieved, which would be at the state of chemical equilibrium. The adiabatic flame temperature, combined with the Gibbs free energy of combustion and the enthalpy of combustion, can be used to determine the equilibrium constant at the adiabatic flame temperature. The equilibrium constant at standard conditions is found by the relation

$$K^0 = \exp(-\Delta G_{Rx}^0/RT). \quad (5)$$

The standard Gibbs free energy of combustion can be calculated in a manner similar to that used to determine the enthalpy of combustion, i.e.,

$$\Delta G_{Rx}^0 = \sum(v_i \Delta G_{f,i}^0) \quad (6)$$

(see Table 3).

The Gibbs-Helmholtz relation can be applied to the problem of calculating the equilibrium constant at the adiabatic flame temperature by employing the assumption that enthalpies of reaction are not nearly as dependent on temperature as Gibbs free energies of reaction [7]. This gives the following relation for the equilibrium constant at the adiabatic flame temperature:

$$K_{Tad} = \exp[(-\Delta G_{Rx}^0/RT^0) + (\Delta H_{Rx}^0 [(T_{ad})^{-1} - (T^0)^{-1}]/R)]. \quad (7)$$

The results of the calculations represented by Eqs. (4), (5), and (7) are shown in Table 4 for the combustion of methane.

Knowledge of the inlet composition allows the computation of the final (i.e., equilibrium) composition through the use of the equilibrium constant. Since the combustion reaction typically takes place in a boiler or similar combustion device at a total pressure of 1 atm, the expression for the equilibrium constant may be written in

the following form:

$$K_{\text{Tad}} = \prod C_i^{v_i} = \prod P_i^{v_i}. \quad (8)$$

In the case of methane combustion, the equilibrium relation is as follows:

$$K_{\text{Tad}} = \frac{[\text{CO}_2][\text{H}_2\text{O}]^2}{[\text{CH}_4][\text{O}_2]^2} = \frac{P_{\text{CO}_2} P_{\text{H}_2\text{O}}^2}{P_{\text{CH}_4} P_{\text{O}_2}^2}. \quad (9)$$

All of the partial pressures in Eq. (9) can be expressed in terms of the partial pressure of methane, the species of greatest interest in this work, i.e.,

$$P_{\text{CO}_2} = P_{\text{CO}_2, \text{max}}(P_{\text{CH}_4, 0} - P_{\text{CH}_4}) \quad (10a)$$

$$P_{\text{H}_2\text{O}} = 2 P_{\text{CO}_2, \text{max}}(P_{\text{CH}_4, 0} - P_{\text{CH}_4}) \quad (10b)$$

$$P_{\text{O}_2} = P_{\text{O}_2, 0} - 2(P_{\text{CH}_4, 0} - P_{\text{CH}_4}) \quad (10c)$$

where $P_{\text{CO}_2, \text{max}}$ is the partial pressure of carbon dioxide which would result if the combustion reaction went to completion. Substituting into Eq. (9) gives the following result:

$$K_{\text{Tad}} = \frac{(a - bP_{\text{CH}_4})(2a - 2bP_{\text{CH}_4})^2}{P_{\text{CH}_4}(c + 2P_{\text{CH}_4})^2}, \quad (11)$$

where a , b , and c are constants based on the inlet composition and reaction stoichiometry. Thus, Eq. (11) represents a system of one equation with one unknown, the equilibrium pressure of methane. Eq. (11) can be rearranged so that it is of the form $f(P) = 0$, i.e.,

$$K_{\text{Tad}} - \frac{(a - bP_{\text{CH}_4})(2a - 2bP_{\text{CH}_4})^2}{P_{\text{CH}_4}(c + 2P_{\text{CH}_4})^2} = 0. \quad (12)$$

Attempts to solve Eq. (12) using MathCAD, Lotus, and Qbasic were not successful, due to the very large equilibrium constant, and the exceedingly small partial pressure of methane which appears in the denominator. However, the reverse reaction may be considered, in which carbon dioxide and water form methane and oxygen. This essentially results in a rearrangement of Eq. (11), with the equilibrium constant for the reverse reaction being the reciprocal of the equilibrium constant for the forward reaction, i.e.,

$$K_{\text{rev}} = K_{\text{Tad}}^{-1}, \quad (13)$$

which gives

$$K_{\text{rev}} = \frac{P_{\text{CH}_4}(c + 2P_{\text{CH}_4})^2}{(a - bP_{\text{CH}_4})(2a - 2bP_{\text{CH}_4})^2} \quad (14)$$

and

$$K_{\text{rev}} - \frac{(a - bP_{\text{CH}_4})(2a - 2bP_{\text{CH}_4})^2}{P_{\text{CH}_4}(c + 2P_{\text{CH}_4})^2} = 0. \quad (15)$$

The solution of Eq. (15) could be carried out using a number of numerical methods. Successive substitution, Newton's method, the secant method, and the *Regula Falsi* method (otherwise known as false positioning) are the methods most commonly used for the solution of one equation with one unknown. Successive substitution, Newton's method, and the secant method all have significant potential to fail to converge, whereas the *Regula Falsi* method generally requires only that the solution be bounded in order to converge. In particular, the *Regula Falsi* method has a far more reliable convergence than either the secant method or Newton's method. Bounding a solution to Eq. (15), or obtaining one value of P_{CH_4} which gives a negative value to the left-handside of Eq. (15) and another value of P_{CH_4} which gives a positive value to the right-handside of Eq. (15), does not present a difficult problem, since intuitively it is known that the equilibrium value of P_{CH_4} is extremely small. For these reasons, the *Regula Falsi* method appears to be a good choice for the solution of Eq. (15). The *Regula Falsi* method has been shown to be capable of solving an equation of the form of Eq. (15) [8], but the number of iterations needed is large.

2. The *Regula Falsi* method

Successive substitution, Newton's method, the secant method, and the *Regula Falsi* method all can be used to generate a numerical solution to an equation of the form $f(x) = 0$, $f: \mathbb{R} \rightarrow \mathbb{R}$ [or, $f(P) = 0$, in the case at hand]. The *Regula Falsi* method is similar to the secant method in that it is based on Newton's method. In contrast to Newton's method, which uses a function value and a derivative at that point to generate the next estimate for the solution of a given equation, the *Regula Falsi* method uses only function values to improve upon succeeding estimates for a solution to the desired equation. Unlike the secant method, the *Regula Falsi* method requires that the solution be bounded between a positive value of f_0 {i.e., $f(P_0)$ } and a negative value of f_1 {i.e., $f(P_1)$ } (or vice versa). Subsequent values of P_{i+1} are found by the relation

$$P_{i+1} = P_i - f_i \frac{(P_i - P_0)}{(f_i - f_0)}. \quad (16)$$

The *Regula Falsi* method is shown graphically in Fig. 1. The graphical representation of the *Regula Falsi* method involves drawing a line from the point (P_0, f_0) to the point (P_1, f_1) . This line crosses the P axis at the value of P_2 . Subsequently, a line is drawn from the point (P_0, f_0) to the point (P_2, f_2) . This line crosses the P axis at P_3 . In this manner, the estimates for P are improved upon until the absolute value of f is less than the required tolerance.

While it is unlikely that, in contrast to Aitken's Δ^2 method [12, 13], an exact solution will be attained through the use of the *Regula Falsi* method, nearly any requisite level of tolerance may be specified. The main limitation would be the precision of the software which is used. It is apparent from Fig. 1 that the *Regula Falsi* method is a simple and reliable method to solve a system consisting of one equation

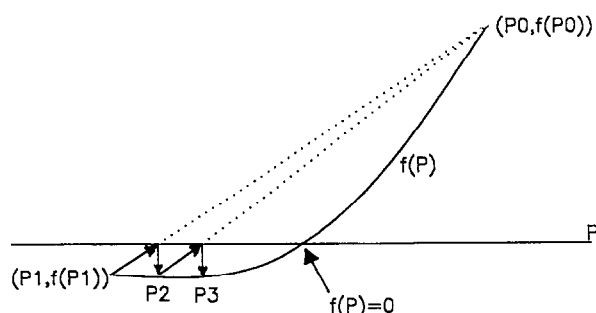


Fig. 1. The *Regula Falsi* method.

and one unknown. Moreover, extremely tight tolerances may be specified and achieved through the use of the *Regula Falsi* method. This bears significantly on the problem at hand because the value of K_{rev} is of the order of 10^{-32} , so a meaningful result is expected only if the tolerance is substantially smaller than K_{rev} . For the work at hand, a tolerance of 10^{-35} was chosen.

Unfortunately, the *Regula Falsi* method has some inherent disadvantages associated with it. Among these are the large number of iterations required for convergence. The fact that one of the initial estimates always is used to generate a new estimate of the solution bears significantly on this. The *Regula Falsi* method has an order of convergence of unity (linear convergence) [14]. The order of convergence is a parameter which describes how quickly the root is approached. In contrast, the Newton–Raphson method has an order of convergence of 2 (quadratic convergence). The disadvantage of using the Newton–Raphson method (otherwise known as Newton’s method) has been discussed in the Introduction (*vide supra*), but the question remains whether a method with cubic convergence or other higher-order convergence should be employed. Unfortunately, numerical methods displaying higher orders of convergence result in rapidly increasing complexity of iterative formulae [15]. Thus, there exists a need to modify the *Regula Falsi* method in such a manner as to preserve its feature of a near-certain convergence to very tight tolerances while reducing the number of iterations.

As mentioned previously (*vide supra*), the *Regula Falsi* method can solve an equation of the form of Eq. (15), but the number of iterations required is large. However, the numerical representation of the *Regula Falsi* method shown in Eq. (16) can be modified to the following form:

$$P_{i+1} = P_i - f_i \frac{(P_i - P_0)}{(f_i - f_0)} UF, \quad (17)$$

where the step size between P_i and P_{i+1} has been increased by a factor UF (the under-interpolation factor). This modification of the *Regula Falsi* method is similar to the Marquardt method of SOR for the solution of differential equations. Analogously, it appears appropriate to call this modification of the *Regula Falsi* method SUI.

An advantage of using the method of SUI is that it is more straightforward and easier to implement than the methods developed by Werner [9] and Della Dora [10]. It should be noted that other refinements to the *Regula Falsi* method have been proposed previously, including approaching the root from both sides of the domain. This is known as the modified false position method [11].

3. Results and discussion

A final methane pressure of 1.174×10^{-11} atm was calculated by the program. The QBasic code which was used to implement this modification of the *Regula Falsi* method is given in the appendix. The initial steps are to declare variables (Appendix, Line 10), and although this normally would not be significant, it merits attention that double precision was used due to the extremely tight tolerances required by the problem at hand. The *Regula Falsi* method requires that the solution be bounded, so initial estimates are made which bound the solution (Appendix, lines 170 and 220). Initial function evaluations (Appendix, Lines 210 and 250) are made based on the initial estimates. The most recent function value is tested to determine whether it is within the limits of tolerance (Appendix, Line 280). If so, the program prints the solution and stops (Appendix, Lines 950–999). Otherwise, the program applies the modification of the *Regula Falsi* method (SUI) in order to update the estimate of the solution (Appendix, Line 300). The line in the code which implements the modified *Regula Falsi* method (SUI) is

$$300 P2 = P1 - (F1 * FAC * (P1 - P)/(F1 - F)). \quad (18)$$

The term FAC in line 300 refers to the under-interpolation factor. One potential problem with SUI is that any given under-interpolation step may overshoot the solution, i.e., give a solution to the right of the point where $F(P) = 0$ in Fig. 1. This would cause the numerical procedure to fail, because the *Regula Falsi* method depends on the most recent estimate of the solution to give a function evaluation of opposite sign of the initial estimate. In other words, at each step the solution must be bounded by P_0 and P_{i+1} . Accordingly, the new estimate is tested for overshoot (Appendix, Lines 306–308). If overshoot has occurred, then the under-interpolation factor is set to unity and the *Regula Falsi* method is applied again (Appendix, Lines 309–310). Thus, overshoot results in the program defaulting to the (unmodified) *Regula Falsi* method. The counter J tallies the number of overshoots. If the method overshoots too frequently, then the under-interpolation factor should be set to a lower value (Appendix, Line 200). While solution of Eq. (15) did not result in any overshoots regardless of the interpolation factor, there is the possibility that overshoot can occur in the solution of other non-linear equations.

The major disadvantage of the *Regula Falsi* method is that a large number of iterations are required to converge on a solution. The counter I tallies the number of interpolations required for convergence, and is printed out along with the solution. Fig. 2 shows the effect of the under-interpolation factor and the number of iterations required for the solution of Eq. (15). Since overshoot and cycling were not observed

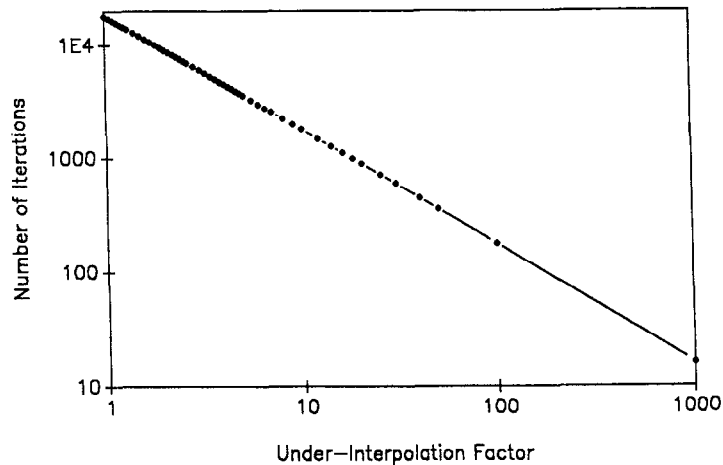


Fig. 2. Effect of under-interpolation factor on number of iterations required for convergence.

(vide supra), the number of iterations decreases monotonically with increasing values of the under-interpolation factor.

It should be noted that a refinement to the method of SUI would be to simply set the under-interpolation factor to a slightly lower value if overshoot occurs, instead of setting it at unity (Appendix, Line 309). However, this refinement was not needed in the present work since overshoot did not occur, so such a refinement remains the subject of future work.

4. Conclusions

The method of SUI has been shown to be an effective modification to the *Regula Falsi* method for the solution of non-linear equations. The use of an under-interpolation factor can dramatically reduce the number of iterations required for convergence, and can be especially helpful in achieving extremely tight tolerances.

Appendix

Complete QBasic Code

```

0      TYPE NUMBER
      P AS DOUBLE
      P1 AS DOUBLE
      P2 AS DOUBLE
      F AS DOUBLE
      F1 AS DOUBLE
      F2 AS DOUBLE

```

```

      A AS DOUBLE
      B AS DOUBLE
END TYPE
100  REM REGULA FALSI METHOD (REVERSE REACTION
      METHOD)
110  REM VARIABLE DIRECTORY
120  REM
130  REM P = PRESSURE OF METHANE (P1, P2 . .)
140  REM TOL = LIMITS OF TOLERANCE
150  REM K = EQUILIBRIUM CONSTANT
160  REM F = FUNCTION (F1, F2 . .)
170  P = 10-9
171  A = 1!
172  B = 1!
173  J = 0
175  I = 0
180  K = (6.146 * 1032)-1
190  TOL = 10-35
200  FAC = 1.05
210  F = K - ((P * (2 * P)2) / ((A - P) * (2 * A - 2 * A * P)2))
215  WRITE "F =", F
220  P1 = 10-12
240  REM EVALUATE FUNCTIONS
250  F1 = K - ((P1 * (2 * P1)2) / ((A - P1) * (2 * A - 2 * A * P1)2))
270  REM CHECK TOLERANCE
280  IF ABS(F1) < TOL THEN 950
290  REM (MODIFIED) REGULA FALSI ROUTINE
300  P2 = P1 - (F1 * FAC * (P1 - P)) / (F1 - F)
305  P1 = P2
306  REM CHECK FOR OVERSHOOT
307  F2 = K - ((P2 * (2 * P2)2) / ((A - P2) * (2 * A - 2 * A * P2)2))
308  IF (F2 * F) < 0 THEN 314
309  P2 = P1 - (F1 * (P1 - P)) / (F1 - F)
310  P1 = P2
311  J = J + 1
314  WRITE "P1 =", P1, "F1 =", F1
315  I = I + 1
320  GOTO 250
950  PRINT P1
960  PRINT P2
970  PRINT F1
980  PRINT F2
985  WRITE "I =", I
986  WRITE "FAC =", FAC
987  WRITE "J =", J
990  STOP
999  END

```

References

- [1] US Environmental Protection Agency, National emission standards for hazardous air pollutants; benzene waste operations final rule, Federal Register, Washington, DC, 1993, pp. 3072–3105.

- [2] US Environmental Protection Agency, Regulation of emissions of certain organic hazardous air pollutants and revision of test methods; proposed rules, Federal Register, Washington, DC, 1992, pp. 62608–62806.
- [3] US Environmental Protection Agency, Compilation of air pollutant emission factors, Vol. 1. Stationary point and area sources, Federal Register, Washington, DC, 1985, p. 1.4-3.
- [4] S.I. Sandler, Chemical and Engineering Thermodynamics, 2nd edn., Wiley, New York, 1989, p. 494ff.
- [5] G.V. Reklaitis, Introduction to Material and Energy Balances, Wiley, New York, 1983, pp. 641–648.
- [6] J.B. Phillips, Prediction of air emission factors for combustion point sources, *Ind. Eng. Chem.*, submitted.
- [7] J.M. Smith and H.C. Van Ness, Chemical and Engineering Thermodynamics, 2nd edn., Wiley, New York, 1975, p. 345ff.
- [8] J.B. Phillips and T.J. Leslie, Computerized predictions of trace emissions, *Sci. Comput. Automat.*, 10(10) (1994) 19–25.
- [9] W. Werner, Some efficient algorithms for the solution of a single nonlinear equation, *J. Comput. Math.*, 9(B) (1981) 141–149.
- [10] J. Della Doro and C. Di Crescenzo, Approximants de Pade-Hermite. 1ère partie: theorie, *Numer. Math.*, 43 (1984) 23–39.
- [11] R.W. Hamming, Numerical Methods for Scientists and Engineers, 2nd edn., McGraw-Hill, New York, 1973, p. 59ff.
- [12] A.C. Aitken, On Bernoulli's numerical solution of algebraic equations, *Proc. Roy. Soc. Edinburgh*, 46 (1926) 289–305.
- [13] C. Brezinski, Survey on convergence acceleration methods in numerical analysis, *Math. Student*, 46(1) (1978) 28–41.
- [14] G. Dahlquist and A. Bjorck, Numerical Methods, Prentice Hall, Englewood Cliffs, 1974, p. 218ff.
- [15] P. Jarratt, A review of methods for solving nonlinear algebraic equations in one variable, in: P. Rabinowitz (Ed.), Numerical Methods for Nonlinear Algebraic Equations, Gordon and Breach, London, 1970, p. 1ff.