

# Chapter 3

## Numerical methods to solve the Liouville–von Neumann equation

### 3.1 Integration of the Liouville–von Neumann equation: Direct and indirect approaches

How to approximate the time evolution determined by the dissipative Liouville–von Neumann equation and how to represent the operators necessary to portray the phenomena under study (including the density matrix itself) are essential points to treat realistic systems.

Details on the representation problem will be discussed in the application sections, because it is directly related to the representation of state vectors in the Schrödinger formalism. A vast literature exists on this subject (Ref.[13] is a recent and detailed review with exhaustive references to the original works).

For the time evolution of a density matrix, there are two main approaches:

- (i) A *direct* strategy, where (2.9) is solved by propagating a density operator; or
- (ii) *indirect* schemes, where (many) different wave packets must be propagated.

The observables are computed via the incoherent summation (2.2).

The direct approaches are usually general, being equally applicable to Redfield or Lindblad forms of dissipation<sup>1</sup>. Direct approaches have been extensively used in recent years. Different choices of representations for the operators have been made: discretized spatial grids [70, 71, 72], possibly with more than one electronic surface [73, 74, 75]; energy or zeroth order or other state representations [76, 77, 78, 79] or mixed representations [80]. Many different approximations for the time evolution have been introduced, ranging from Runge–Kutta general purpose integrators [72, 76] to split operator schemes [70, 73, 75], direct diagonalization of the Liouvillian matrix [77], polynomial methods [81, 74, 82, 80], and Krylov methods [78]. The most promising methods seem to be the split schemes, the polynomial interpolations and approximations, and the Krylov methods. At the present state of development, it is not clear yet which integrators are more suited for a specific problem. They have a slightly different memory occupation, needing respectively at least one, three, and some ten copies of the matrix representing the density operator. They are supposed to have different performances when used for time dependent or time independent problems, as it is the case for their counterparts in the state vector dynamics with the Schrödinger equation [83]. In addition they have much different levels of complication for their practical use. A disadvantage of the direct methods is their unfavorable memory requirements which scale at least as  $N^2$  (the size of a density matrix itself) where  $N$  is the dimension of the Hilbert space of the system.

The wave packet based indirect methods usually offer considerable savings because only  $N$  component wave functions have to be propagated. However, this has to be done  $n$  times to compute observables via incoherent summation. When  $n < N$ , not only memory can be saved, but even a speedup in computation time relative to the direct methods is possible.

There are many examples of wave packet approaches to open quantum systems. Among them, the Monte Carlo Wave Packet method (MCWP) is a relatively common technique [84, 85, 73, 86, 87, 88, 89, 90]. In its most straightforward imple-

---

<sup>1</sup>This is often true also for other less known schemes like the Caldeira–Leggett model [65], because the equation is integrated “as it is”.

mentation, a Liouville–von Neumann equation of Lindblad form (2.11) is replaced with an ensemble of wave functions, whose dynamics follows the non–Hermitean time–dependent Schrödinger equation

$$|\dot{\psi}(t)\rangle = \left( -i\hat{H}_s - \frac{1}{2} \sum_i \hat{C}_i^\dagger \hat{C}_i \right) |\psi(t)\rangle. \quad (3.1)$$

Here, the  $\hat{C}_i$  are the Lindblad dissipation operators, as defined in section 2.2. The term  $-\frac{1}{2} \sum_i \hat{C}_i^\dagger \hat{C}_i$  is the source of the non–Hermitecity, and being negative it reduces the norm of the wave function  $|\psi(t)\rangle$  in time. Each wave function is propagated for a timestep, and its norm loss computed; if it is larger than a random number, chosen between zero and one, the function undergoes a “quantum jump” according to the form of the dissipation operators [85]. There are other implementations of the method, see for example Ref.[87]. The stochastic quantum state diffusion method (QSD) [91, 92, 93, 94, 73, 72] is similar to the MCWP.

An interesting alternative is the “jumping wave packet & incoherent, weighted averaging scheme”, or Gadzuk method [41]. For very simple forms of dissipation, the distribution of the MCWP jump probabilities can be determined analytically [41] or semianalytically [90]. The wavefunctions are then propagated with the time–dependent Schrödinger equation and undergo jumps out of a time distribution. The observables are computed via incoherent summation of the values obtained for each function via the distribution function. For the cases where it can be applied, the method is very efficient.

The Variational Wave Packet (VWP) method is a recently introduced alternative [95], which will be discussed in the forthcoming sections.

The wave packet methods are not always general. MCWP, QSD and related schemes are known only for dissipation models of Lindblad form [85]. The Gadzuk method and its generalized versions can be used for special varieties of Lindblad operators [90]. Only the VWP is in principle completely general in its applicability. Some of the wave packet methods are stochastic approaches and sometimes converge only poorly to the exact results, in particular when these are dominated by statistically rare events (see Refs.[89, 96], section 3.4 and chapter 4).

In connection with direct methods, we have been working with polynomial expansions of the Liouvillian. As described in Section 3.2, these polynomial expansions can only be used if the dynamics is Markovian and linear. As indirect methods, we used the VWP method, described in section 3.4. A comparison of VWP with the MCWP and the Gadzuk method is given in chapter 4.

## 3.2 Polynomial expansions

### 3.2.1 Overview on the polynomial integrators

The representation of operators in a computer simulation can always be expressed in matrix formalism. In particular, the polynomial methods are based on the theory of the approximation of functions of matrices [97]. In this section, we will always consider Eqn.(2.9), written in matrix form as

$$\dot{\rho} = \mathcal{L}\rho, \quad (3.2)$$

With the initial condition  $\rho(0) = \rho_0$ . Here  $\mathcal{L}$  is a  $D \times D$  matrix representation of the Liouvillian and  $\rho_0$  as a vector of size  $D = N \times N$ , where  $N$  is the size of the Hilbert space of the system. Analytically, the solution of (3.2) for a time  $\tau \geq 0$  is

$$\rho(\tau) = \exp(\tau\mathcal{L})\rho_0. \quad (3.3)$$

In practice, the exponential of a large matrix has to be approximated. If we choose a polynomial approximation, we are interested in the polynomial  $P_n^\tau(\mathcal{L})\rho_0$  which minimizes among all polynomials  $P$  of degree  $\leq n$  the local error

$$\epsilon_{\text{loc}}(n) = \|\exp(\tau\mathcal{L})\rho_0 - P_n^\tau(\mathcal{L})\rho_0\|. \quad (3.4)$$

The application of functional calculus of analytic functions [98] gives an insightful framework to our approximation problem. Let  $\Gamma \subset \mathbb{C}$  be any Jordan curve<sup>2</sup>, enclosing the complex eigenvalue spectrum of the dissipative Liouvillian  $\mathcal{L} = \mathcal{L}_H + \mathcal{L}_D$ ;

---

<sup>2</sup>A closed curve that does not intersect itself, *e.g.*, the boundary of a rectangle or an ellipse.

then

$$\begin{aligned}\epsilon_{\text{loc}}(n) &= \|\exp(\tau \mathcal{L}) \rho_0 - P_n^\tau(\mathcal{L}) \rho_0\| \\ &= \left\| \frac{1}{2\pi i} \int_{\Gamma} \{\exp(\tau z) - P_n^\tau(z)\} (z \cdot \mathbf{I} - \mathcal{L})^{-1} \rho_0 dz \right\| \\ &\leq \min_{\Gamma} \left\{ C_{\Gamma} \max_{z \in \Gamma} |\exp(\tau z) - P_n^\tau(z)| \right\}\end{aligned}\quad (3.5)$$

for a constant  $C_{\Gamma} > 0$ , depending on  $\mathcal{L}$  and  $\Gamma$ , but independent of  $n$ .  $\mathbf{I}$  is the identity  $D \times D$  matrix. From the above inequality, we see that the local error is related to the problem of approximating a scalar analytic function, *i.e.*, finding among all polynomials  $P_n^\tau$  at most of degree  $n$  the one that minimizes for fixed  $\Gamma$

$$\max_{z \in \Gamma} |\exp(\tau z) - P_n^\tau(z)|. \quad (3.6)$$

Following the maximum principle [99], one can substitute in (3.6)  $\Gamma$  by the domain  $G = G(\Gamma)$  defined as the set of all points enclosed by  $\Gamma$ .

The exact solution of the min–max problem requires the calculation of the spectrum of  $\mathcal{L}$ , that is equivalent to diagonalize it. This already corresponds to having solved problem (3.3) [77], rendering superfluous the subsequent polynomial approximation. Anyway, direct matrix diagonalization is ineffective compared to iterative multiplication with a sparse matrix<sup>3</sup> [78] because of unfavorable scaling [102].

Therefore, it is a common practice to fix a curve  $\Gamma$  or respectively a domain  $G$  and consider the scalar valued approximation problem (3.6). The choice of the domain  $G$  is important for the numerical aspects as we are going to see below.

There is a rich literature about this approximation problem (3.6) both in complex analysis [103, 104, 105, 106] and in theoretical chemistry [12, 83, 107] to cite a few. These different methods can be considered as special realizations of polynomial approximations.

In the next section, the notion of conformal mapping associated with a domain is introduced to be used in the definition of the “quasioptimal” polynomial approximation on a domain that includes the eigenvalues of the matrix  $\mathcal{L}$ .

---

<sup>3</sup>The Liouvillian can be a very large matrix, but it is often very sparse [100, 101, 102].

### 3.2.2 Conformal mapping

For a given domain  $G$ , it is advantageous to consider separately its geometry and its size. The shape of  $G$  determines the form of the polynomial approximation  $P_n^\tau$ , while the size influences its numerical stability.

Let  $G$  be a bounded, closed continuum in the complex plane, such that the complement of  $G$  is simply connected in the extended plane and contains the point at  $\infty$ , *e.g.*, a rectangle or an ellipse. By the Riemann mapping theorem [99], there exists a conformal mapping  $\psi : \mathbb{C} \rightarrow \mathbb{C}$  which maps the complement of a closed disc with center at the origin and radius  $\rho$  onto the complement of  $G$ , satisfying the normalization condition<sup>4</sup>  $\lim_{|w| \rightarrow \infty} \psi(w)/w = 1$ , where  $w$  is a complex number. Then, its Laurent expansion at  $\infty$  is given by

$$\psi(w) = w + \gamma_0 + \gamma_1 w^{-1} + \gamma_2 w^{-2} + \dots \quad (3.7)$$

with coefficients  $\gamma_i \in \mathbb{C}$ . The logarithmic capacity of  $G$  is defined as the radius  $\rho$  of the above disc. We call a domain *scaled*, if  $\rho = 1$ .

#### Examples:

- (i) The conformal mapping

$$\psi(w) = w + m + d/w \quad (3.8)$$

with parameters  $m, d \in \mathbb{C}$  is the one that we are going to use throughout to integrate the Liouville–von Neumann equation, so it is of central importance. The left picture of Fig.3.1 shows that for a given logarithmic capacity  $\rho$ ,  $\psi(w)$  describes a family of ellipses with center at  $m$  and minor and major axis  $a = (\rho - d/\rho)$  and  $b = (\rho + d/\rho)$ , respectively.

- (ii) The mapping

$$\psi(w) = w + m - 1/(2w)^3, \quad m \in \mathbb{C},$$

specifies a family of “rounded rectangles”, centered at  $m$  (see the right panel of Fig.3.1).

---

<sup>4</sup>The normalization condition is equivalent to the requirement that the coefficient corresponding to  $w$  is equal to 1.

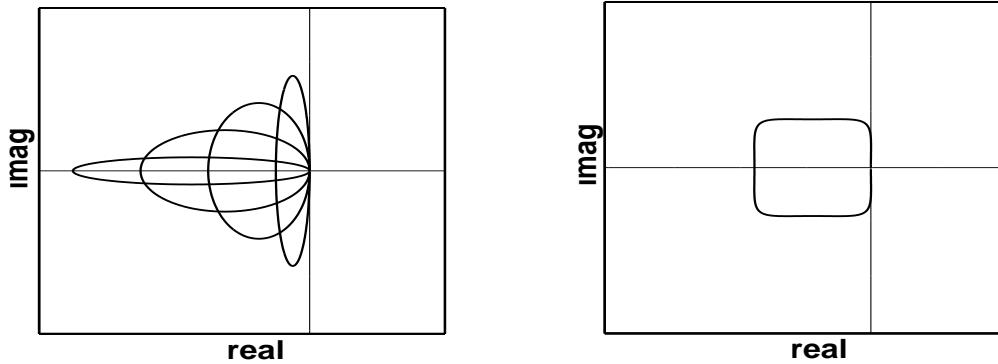


Figure 3.1: Elliptical and rectangle like *scaled* domains as in the example below. The parameters are  $m = -0.25, -0.75, -1.25, -1.75$  and  $d = -(m + 1)$  (from the left to the right ellipse), and  $m = -0.875$  for the “rectangle”. The real and imaginary axes cross at  $(0,0)$  in the complex plane, real part goes from  $w = -2$  to  $w = +1$ , the imaginary from  $-2i$  to  $+2i$ .

### 3.2.3 The Faber approximation

In this section, a brief introduction to Faber polynomials is provided. It is interesting to notice that the well known Chebyshev polynomials [108] are a special family of Faber polynomials constructed to approximate continuous functions of real variables. When functions of matrices have to be computed, the Chebyshev approximation is consequently suited to matrices with real (or purely imaginary) eigenvalues, like the Hamiltonian (or the Hamiltonian multiplied by  $i$ ). While Faber polynomials are generally appropriate when the eigenvalues are defined in the complex plane, as necessary for the Liouvillian, with an imaginary (Hamiltonian times  $i$ ) and real (dissipative part).

The family of *Faber polynomials*  $\{F_k\}_{k \in \mathbb{N}}$  associated with a conformal mapping  $\psi$  is defined via the recursion relation

$$F_{k+1}(z) = z \cdot F_k(z) - \sum_{j=0}^k \gamma_j \cdot F_{k-j}(z) - k \cdot \gamma_k \quad (3.9)$$

for  $k \geq 1$  and  $F_0(z) \equiv 1$  [109, 110, 111, 107]. The corresponding relations for matrix

operations are obtained substituting  $z$  by  $\mathcal{L}$  and multiplying the equations by  $\boldsymbol{\rho}_0$ , as is exemplified in Eqn.(3.11) below.

The recursion relation is stable, if  $z$  or the spectrum of  $\mathcal{L}$  respectively are contained in the *scaled* domain [112]. It can be seen from (3.9) that Faber polynomials, defined by their recursion relation, depend upon the coefficients  $\gamma_j$  of the conformal mapping  $\psi$  and thus on the shape<sup>5</sup> of  $G$ , but they are generated independently of the size of the domain  $G$  (there is no  $\rho$  in (3.9)).

From a numerical point of view, *e.g.*, for memory occupation, we are interested in the families of Faber polynomials which allow short term recursions. Thus, we are interested in domains  $G$  whose associated conformal mappings have only a few nonzero terms in their Laurent expansion at  $\infty$  (see (3.7)). Among them, we have been working mainly with the family of Faber polynomials corresponding to the conformal mapping  $\psi(w) = w + m + d/w$  (see Example (i) above). The parameters  $m$  and  $d$  depend upon the relative “strength” of the Hamiltonian and dissipative dynamics of the physical problem studied.

For our conformal mapping  $\psi(w) = w + m + d/w$ , the associated Faber polynomials are defined by the three term recursion

$$F_{k+1}(z) = (z - m)F_k(z) - d \cdot F_{k-1}(z), \quad k \geq 1 \quad (3.10)$$

with initial values  $F_0(z) \equiv 1$ ,  $F_1(z) \equiv z - m$  and  $F_2(z) \equiv (z - m)^2 - 2d$ . The matrix equivalents of these relations are

$$F_{k+1}(\mathcal{L})\boldsymbol{\rho}_0 = (\mathcal{L} - m \cdot \mathbf{I})F_k(\mathcal{L})\boldsymbol{\rho}_0 - d \cdot F_{k-1}(\mathcal{L})\boldsymbol{\rho}_0, \quad k \geq 1 \quad (3.11)$$

with initial values  $F_0(\mathcal{L})\boldsymbol{\rho}_0 \equiv \boldsymbol{\rho}_0$ ,  $F_1(\mathcal{L})\boldsymbol{\rho}_0 = (\mathcal{L} - m \cdot \mathbf{I})\boldsymbol{\rho}_0$  and  $F_2(\mathcal{L})\boldsymbol{\rho}_0 = (\mathcal{L} - m \cdot \mathbf{I})F_1(\mathcal{L})\boldsymbol{\rho}_0 - 2d \cdot \boldsymbol{\rho}_0$ .

Setting the parameters  $m$  to 0 and  $d$  to  $1/4$ , the associated Faber polynomials  $F_k$  are equal to the normalized Chebyshev polynomials  $T_k$  via  $F_k(z) = 2^{1-k}T_k$ , for  $k \geq 1$ , while  $F_0 = T_0$  for  $k = 0$ . For our purposes the Chebyshev polynomials are too restrictive since they cannot be adapted to the strength of dissipation.

---

<sup>5</sup>This is the reason why the conformal mapping has to be introduced.

Any function that is analytic inside  $G$  can be expanded in terms of the Faber polynomials associated with  $\psi$  [109, 113, 110]. In application to  $\exp(\tau z)$ , this yields:

$$\exp(\tau z) = \sum_{k=0}^{\infty} \underbrace{\frac{1}{2\pi i} \int_{|w|=1} \frac{\exp(\tau\psi(w))}{w^{k+1}} dw}_{c_k(\tau)} F_k(z) \quad (3.12)$$

for all  $z \in G$ . Now we define the *Faber approximation* of order  $n$  to be the truncated series

$$P_n^\tau(z) = \sum_{k=0}^n c_k(\tau) F_k(z)$$

with expansion coefficients as defined above. Substituting  $F_k(z)$  by  $F_k(\mathcal{L})\rho_0$ , one gets the “matrix valued version”  $P_n^\tau(\mathcal{L})\rho_0$

$$\rho(\tau) = \exp(\tau\mathcal{L})\rho_0 \approx P_n^\tau(\mathcal{L})\rho_0 = \sum_{k=0}^n c_k(\tau) F_k(\mathcal{L})\rho_0. \quad (3.13)$$

That is what we are interested in (notice that choosing another polynomial series with terms  $S_k(t)$ , the relation would be formally the same, being only necessary to substitute  $F_k(\mathcal{L})\rho_0$  with  $S_k(t)(\mathcal{L})\rho_0$  and use the coefficients belonging to the new series).

For our conformal mapping  $\psi(w) = w + m + d/w$ , the coefficients can be computed analytically

$$c_k(\tau) = \frac{1}{2\pi i} \int_{|w|=1} \frac{\exp(\tau(w + m + d/w))}{w^{k+1}} dw \quad (3.14)$$

$$= (-i/\sqrt{-d})^k \exp(\tau m) J_k(2\tau\sqrt{-d}), \quad (3.15)$$

where we used the identity  $\exp(x(t + 1/t)/2) = \sum_k (t/i)^k J_k(ix)$  [114]. Here,  $J_k$  is the Bessel function of the first kind. For  $d = 0$  the conformal mapping  $\psi$  is a simple translation. Thus, equation (3.12) is simply the power series expansion at point  $m$ . From now on, the term “Faber approximation” is always meant with respect to the conformal mapping (3.8).

### 3.2.4 The Newton interpolation at Lejá points

Another way to approximate functions of matrices with polynomials is to make use of the theory of the interpolation of analytic functions. The complex Newton interpolation based on Lejá points is an efficient implementation of this idea, and was introduced in density matrix calculations by Kosloff and coworkers [81, 115]. The method is shortly outlined in the following.

Let  $G$  be a domain as defined in Section 3.2.2. A sequence  $(z_m)_{m \in \mathbb{N}}$  of points on the boundary of  $G$ , *i.e.*,  $(z_m)_{m \in \mathbb{N}} \subset \Gamma$ , is called a *Lejá point sequence* [106], if  $|z_1| = \max_{z \in \Gamma} |z|$  and<sup>6</sup>

$$\prod_{k=1}^m |z_{m+1} - z_k| = \max_{z \in \Gamma} \prod_{k=1}^m |z - z_k|$$

for  $m > 1$ . In numerical applications, one substitutes the maximum of all  $z \in \Gamma$  by the maximum of all  $z \in \Gamma_L$ , where  $\Gamma_L = \{\tilde{z}_1, \tilde{z}_2, \tilde{z}_3, \dots, \tilde{z}_L\}$  is a set of equally spaced points on the boundary of  $G$  with  $L \gg n = \text{degree of } P_n^\tau$ . We call a Lejá point sequence *scaled* if the points lie on the boundary of a *scaled* domain.

A sequence of Lejá points defines the associated Newton polynomials  $\{\omega_k(z)\}_{k \in \mathbb{N}}$  by the two term recursion

$$\omega_{k+1}(z) = (z - z_{k+1}) \omega_k(z) \quad (3.16)$$

for  $k \geq 0$  and  $\omega_0(z) \equiv 1$ . The relations used for functions of matrices are defined by

$$\omega_{k+1}(\mathcal{L})\rho_0 = (\mathcal{L} - z_{k+1} \cdot \mathbf{I}) \omega_k(\mathcal{L})\rho_0 \quad (3.17)$$

with starting term  $\omega_0(\mathcal{L})\rho_0 \equiv \rho_0$ . The recursion relation is stable if the Lejá points are *scaled* and  $z$  (or the spectrum of  $\mathcal{L}$  respectively) is contained in the *scaled* domain [106, 115].

The Newton polynomials are related to the logarithmic capacity  $\rho$  of  $G$  by

$$\sqrt[m+1]{|\omega_m(z_{m+1})|} \longrightarrow \rho; \quad m \rightarrow \infty \quad (3.18)$$

---

<sup>6</sup>For the numerical purposes,  $z_1$  can be any point  $\in \Gamma$ .

thus

$$|\omega_m(z_{m+1})| \approx \rho^{m+1} \quad (3.19)$$

for “large”  $m$ . This shows why a rescaling of the iteration is necessary to avoid overflows (underflows) [116, 106].

The coefficients are the so called divided differences. For a function  $f$  on  $G$ , the divided differences can be defined recursively<sup>7</sup>:

$$[z_k, \dots, z_l]f = \frac{[z_{k+1}, \dots, z_l]f - [z_k, \dots, z_{l-1}]f}{z_l - z_k} \quad (3.20)$$

for  $1 \leq k < l$  and initial values  $[z_k]f = f(z_k)$ . To have a stable recursion, the Lejá points have to be *scaled* [106, 116].

Any function that is analytic inside the domain  $G$  can be expanded in terms of the Newton polynomials associated with the Lejá points  $(z_m)_{m \in \mathbb{N}}$  [109, 113, 106]. The application to  $f(z) = \exp(\tau z)$  yields

$$\exp(\tau z) = \sum_{k=0}^{\infty} [z_1, \dots, z_{k+1}] \exp \omega_k(z)$$

for all  $z \in G$ . Now we define the *Newton interpolation* of order  $n$  as the truncated series

$$\exp(\tau z) \approx P_n^{\tau}(z) = \sum_{k=0}^n [z_1, \dots, z_{k+1}] \exp \omega_k(z). \quad (3.21)$$

Substituting  $\omega_k(z)$  by  $\omega_k(\mathcal{L})\rho_0$  one gets the “matrix valued version”  $P_n^{\tau}(\mathcal{L})\rho_0$  that is formally equivalent to relation (3.13).

In contrast to the Faber approximation for the Newton interpolation, there are no restrictions on the shape of the domain or the function  $f(\mathcal{L})\rho_0$ .

---

<sup>7</sup>It is easy to see that every two elements of a Lejá point sequence are different and therefore the denominator is different from zero.

### 3.2.5 Numerical aspects

Analytically, the Faber approximation and the Newton interpolation are very similar [112]. Interesting is whether in numerical applications they would behave in a similar way or not. Before starting, we have to decide how to choose the domain that includes the eigenvalues of  $\mathcal{L}$ . This step is very similar for the two algorithms.

#### On scaling and domain

The effects of the choice of the domain on the numerical stability and efficiency are of general nature, so any system can be used to exemplify them. For this purpose we have chosen an abstract model as dissipative system. Here  $\mathcal{L}$  corresponds simply to a diagonal matrix with complex eigenvalues, which are plotted in the first panel of Fig.3.2. The function approximated is, of course, the propagator (3.3), and the results are reported for a single timestep, if many are used, the errors add together. The shape of the domain  $G$  is taken as elliptic. The results are shown only for the Faber approximation, being the effects of the shapes and sizes of domains absolutely general for any polynomial integrator.

The behavior can be illustrated with three exemplary cases, shown in Figs.3.2, 3.3, and 3.4.

In Fig.3.2, a typical case of numerical instability is presented. In the first panel of Fig.3.2, the dots represent the eigenvalues and the solid curve the elliptic domain used for the polynomial expansion. The recursion relation is unstable because the *nonscaled* eigenvalues of  $\mathcal{L}$  lie outside the domain. The local error (3.4) grows first exponentially, but then finally decreases to a constant (second panel). The Frobenius norm of the last term  $F_n(\mathcal{L})\rho_0$  in the series (3.13) explodes, as shown in the right panel of Fig.3.2, so the computation of its value allows to control this kind of numerical instability. Furthermore, it was shown in [117] that in the above case the remaining error grows exponentially with the timestep  $\tau$ , so for large values of  $\tau$  the accuracy is completely lost.

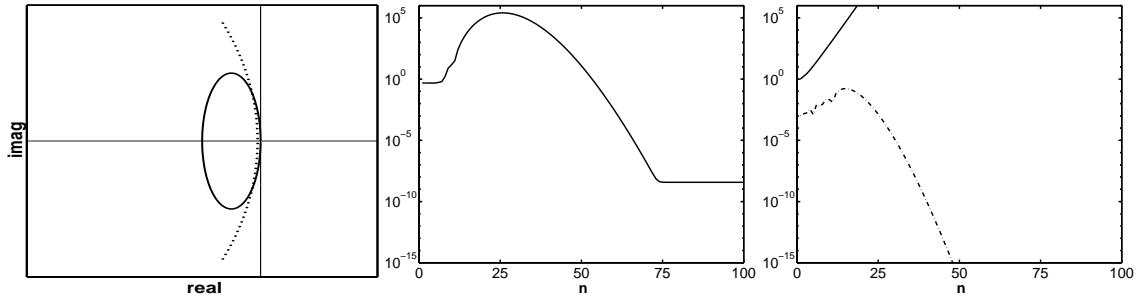


Figure 3.2: Behavior of the integration for a *nonscaled* Liouvillian. Starting from the left, the *scaled* domain and the *nonscaled* spectrum (' $\dots$ ') are shown in the first graph (the graph proportions are the same as in Fig.3.1). The local error (3.4) versus the order  $n$  of approximation is depicted in the center, while in the last picture the Frobenius norm of the Faber polynomials ('—') are presented together with the modulus of the coefficients (3.15) ('--').

**Scaling.** To avoid these numerical instabilities in the recursion relation, the Liouvillian has to be scaled:

$$\mathcal{L} \rightarrow \sigma^{-1} \mathcal{L}. \quad (3.22)$$

The scaling factor  $\sigma > 0$  should make the spectrum of  $\sigma^{-1} \mathcal{L}$  lie inside the *scaled* domain. As a consequence<sup>8</sup>, the step size has to change, too:  $\tau \rightarrow \sigma\tau$ .

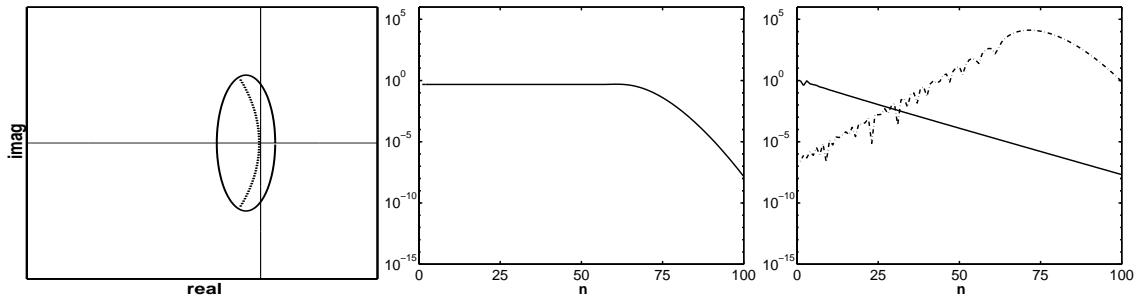


Figure 3.3: Behavior of the integration when the Liouvillian is properly scaled but the domain lies partially in the right part of the complex plane. The meaning and size of the graphs is the same as in Fig.3.2.

<sup>8</sup>We have  $\rho(\tau) = \exp(\tau\mathcal{L})\rho_0 = \exp(\sigma\tau \sigma^{-1}\mathcal{L})\rho_0 \approx P_n^{\sigma\tau}(\sigma^{-1}\mathcal{L})\rho_0$ . Note that in general  $P_n^{\sigma\tau}(\sigma^{-1}\mathcal{L})\rho_0 \neq P_n^\tau(\mathcal{L})\rho_0$ , although the identity holds for the exponential function.

In Fig.3.3, it is shown what happens if the domain is properly scaled but improperly set and enters in the right part of the complex plane. This depends on the position of the points for the Newton interpolation and on the parameter  $m$  for the Faber approximation. The norm of the Faber polynomials decreases exponentially in this case, because the recursion relation is stable, the modulus of the coefficients (3.15) increases exponentially until  $n \approx 70$ , where it reaches a values far larger than 1, then it starts to decrease. The situation is far from “optimal”, and worsens while the domain is further moved to the right on the complex plane (the growth of the coefficients can easily overflow). If the center  $m$  of the ellipse lies on the imaginary axis, the local error looks similar to the one shown in Fig.3.2. The local error is constant until  $n \approx 70$  and then starts to decrease, so the calculation finally converges, but this represents a very inefficient propagation, and can be avoided checking the growth of the modulus of the coefficients (the coefficients are computed together at the beginning, so this check is easy to perform).

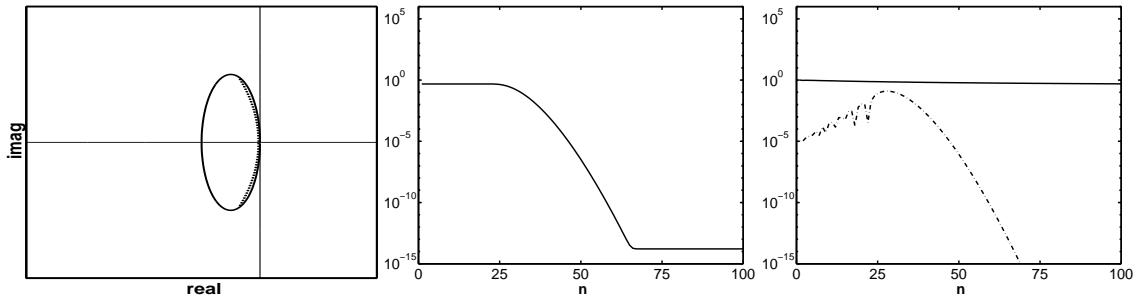


Figure 3.4: Behavior of the integration when the Liouvillian is properly scaled and the domain is correctly set. The meaning and size of the graphs is the same as in Fig.3.2.

In Fig.3.4, the spectrum of the *scaled* Liouvillian lies entirely in the *scaled* domain, which is completely contained in the left part of the complex plane. The norm of the Faber polynomials is almost constant, showing that the scaling was correct [97]. The modulus of the coefficients (3.15) is bounded by 1 and decays exponentially from  $n \approx 30$  on. The effort to reach a given local tolerance is much less than in the two examples before. The optimal ellipse has the smallest scaling factor  $\sigma$  compatible with a stable propagation.

The results for the Newton interpolation are reasonably equivalent, if the points

are set on the boundary of the *scaled* domain shown in Figs.3.2-3.4. The local error explodes in the first example, it is a factor  $10^5$  bigger than the error of the Faber series in the second example and it is nearly the same of the other series in the third example. In contrast to the smooth error curves of the Faber approximation, the corresponding curves for the Newton interpolation are “shaky” due to the use of divided difference coefficients.

In practice, for the choice of the domain, an algorithm is necessary to determine a rough estimate of the region of the complex plane where the eigenvalues of the Liouvillian matrix  $\mathcal{L}$  lie. This part is again identical both for the Faber and for the Newton series. We are not interested in direct diagonalization, being the matrix often too big to be held in the central memory of any present day computer, and the algorithm too expensive to be used just to determine the shape of the eigenvalue spectrum. To estimate an eigenvalue spectrum, we decided to use the most simple iterative method, the power method [118], because with the reasonable assumption that the eigenvalues lie in a domain between the zero and the eigenvalue with largest modulus, only this last one has to be evaluated<sup>9</sup>. The method is not very efficient, but the time required for this step is negligible compared to the complete propagation (typically fifty iterations compared to tens of thousands). We are aware of the fact that the vector iteration is more or less ill conditioned, if the dimension  $D$  (see Eqn.(3.2)) is very large, but we need only a rough estimate of the maximum eigenvalue.

To have a reasonably stable iteration, one should consider that:

- 1) The eigenvalues are distributed symmetrically with respect to the real axis, because their imaginary part derives from the commutator in Eqn.(2.9), *i.e.*,  $-i \cdot \hat{H}_s \hat{\rho} + i \cdot \hat{\rho} \hat{H}_s$ . This term corresponds to all the possible differences between the eigenvalues of  $\hat{H}_s$ , that is the imaginary frequencies  $i\omega_{ij} = i(E_i - E_j)$ . Accordingly, there are always two eigenvalues with maximum modulus. Adding  $i\tilde{E}_{max} * \hat{I}$  to the Liouvillian (a known trick in eigenvalues calculations) where  $\tilde{E}_{max}$  is an estimate of the maximal energy (following for example [12]), the eigenvalue spectrum is translated along the

---

<sup>9</sup>Except for special cases where the shape of the dissipation domain is expected to be “strange”, and in these cases the need arises to determine more eigenvalues accurately. In these cases it would be more reasonable to use a Krylov iterative method for the complete dynamics, as is done in [102].

imaginary axis and one of the eigenvalues with maximum modulus becomes larger than the other one.

2) The Frobenius norm (0.2) and relative scalar product has to be used to compute the eigenvalue as

$$\lambda = \frac{\text{tr}(\rho_n^\dagger \rho_{n-1})}{\sqrt{\text{tr}(\rho_{n-1}^\dagger \rho_{n-1})}}.$$

3) The number of iterations must be larger than a minimum value in order to let the system relax close enough to the desired eigenvalue. Then set an error criterion between the different iterations to chose when to stop the process.

### 3.2.6 Spectral estimates, scaling and coefficients

**Newton interpolation.** In order to stabilize the computation of the Newton polynomials and the divided difference coefficients, the Lejá points have to be scaled (see Eqn.(3.20)). For this purpose a precise evaluation of the logarithmic capacity of the domain is essential. We used an iterative process because to generate a precise logarithmic capacity many uniformly distributed points are necessary; but the computation of the uniform Lejá points needs scaled starting points (with the logarithmic capacity that we do not have) before, for numerical stability. Thus we first use a few points in the Lejá algorithm to derive a rough estimate of  $\rho$  (see (3.18)), then all of the points are rescaled with it. More points are used to generate a more precise capacity, until the logarithmic capacity values obtained for a group of random points inside the domain differ by less than the required accuracy. Usually no more than three or four iterations are needed.

The generation of the Lejá points requires the repetition of many operations [115]. If we call  $z_i$  the  $i$ th point chosen, and  $d_n(z_i)$  the product of the distances of point  $i$  with the  $n$  points already chosen, the Lejá algorithm starts as follows:

- The first point  $z_1$  is chosen.

- $d_1(\tilde{z}_i) = \tilde{z}_i - z_1$  for every other point  $\tilde{z}_i$  is computed and the  $\tilde{z}_i$  with maximal  $|d_1|$  is chosen as the second point  $z_2$ .
- The product  $d_2(\tilde{z}_i) = (\tilde{z}_i - z_1)(\tilde{z}_i - z_2) = d_1(\tilde{z}_i)(\tilde{z}_i - z_2)$  is computed, and the maximum value of its modulus determines  $z_3$ .
- At every step  $k$  the  $d_k(\tilde{z}_i) = \prod_{j=1}^k (\tilde{z}_i - z_j) = d_{k-1}(\tilde{z}_i)(\tilde{z}_i - z_k)$  for every  $\tilde{z}_i$  is computed and its maximum value in modulus determines  $z_k$ , until the number of required points,  $n$ , is generated.

If the last  $d_{k-1}(\tilde{z}_i)$  for all the remaining points in  $\Gamma_L$  is kept in the memory, the algorithm scales as  $n(L - n - 1)$ , where  $n$  is the number of points to be chosen for the interpolation and  $L$  is the total number of points used by the Lejá algorithm.

**Faber approximation.** If an eigenvalue  $\lambda$  of maximal modulus is known, it is possible to determine an “optimal” ellipse: Its parameter  $m \in [-2, 0]$  solves the third order equation  $(1+r^2)m^3 + (6r^2-2)m^2 + 12r^2m + 8r^2 = 0$  with  $r = \text{imag}(\lambda)/\text{real}(\lambda)$ , which can be solved by Newton methods [118]. Since the ellipse should not penetrate the right part of the complex plane, we choose  $d = -(m+1)$ . Now, the scaling factor is fixed, too:  $\sigma = |\lambda/q|$  for  $q = \sqrt{1+r^2}2rm(2+m)^2/(m^2+r^2(2+m)^2)$ .

The next step is to calculate the expansion coefficients  $c_k = \eta_k J_k(2\sigma\tau\sqrt{-d})$  with  $\eta_k = (-i/\sqrt{-d})^k \exp(\sigma m)$ . Since the density matrix theory in general is applicable in the weak or at least medium coupling limit (for Markovian equations), the spectrum of  $\mathcal{L}$  is “near” the imaginary axis and therefore  $-1 \leq m \leq 0$ . Accordingly,  $d \geq 0$  and we need only Bessel functions for purely real arguments. The factor  $\eta_k$  is prone to underflow for large  $\tau$  and to overflow for large  $k$ . For this purpose we calculated  $\eta_k$  in the following way:

$$\eta_k = \begin{cases} \{(\frac{1}{\sqrt{-d}})^{k/\tau} \exp(\sigma m)\}^\tau; & |\eta_{k-1}| < 10^{-3} \text{tol} \\ \frac{1}{\sqrt{-d}} \eta_{k-1} & \text{otherwise} \end{cases}$$

with  $\eta_0 = \exp(\sigma m)$  and tol denoting the local tolerance.

### 3.2.7 Local error estimator

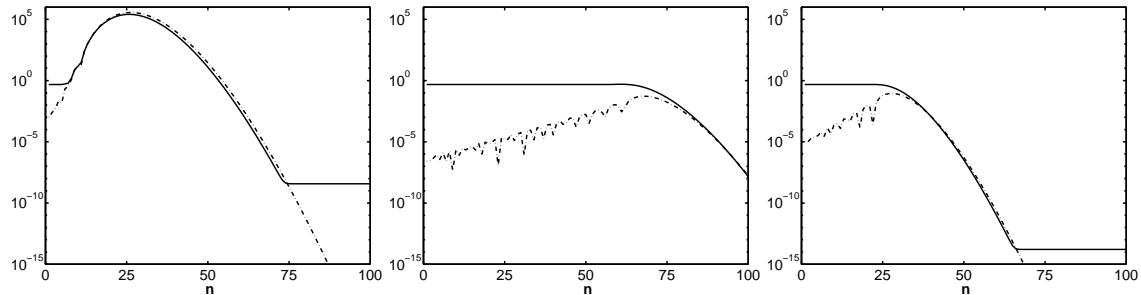


Figure 3.5: The local error of the Faber approximation (‘—’) and the local error estimator (3.23) (‘-··-’) for the three examples above. The two curves do behave very similar *in the interesting decaying phase of the local error*.

The operation

$$\epsilon_{\text{loc}}(n) = |c_n(\sigma\tau)| \operatorname{tr}(F_n^* F_n), \quad (3.23)$$

defined for the last term  $F_n$  included in the series (3.13) can be used to construct a local error estimator. The estimator is only valid in the decaying phase, which starts around  $n \approx \sigma\tau$  and ends at the constant remaining error level, see Fig.3.5. The norm of a *scaled*  $F_k$  being usually around 1, the order of the series is taken as high as necessary for the modulus of the coefficient of the last polynomial term,  $c_n(\sigma\tau)$ , to be smaller than the required tolerance. Then (3.23) is evaluated to see if the procedure was correct, so it is used only in the decaying phase, as it should be.

For simulation purposes, also the accuracy limit (the horizontal line) is perfectly fine, being the errors usually done far larger than the remaining error (see in Fig.3.5). Indeed the above error estimator worked very reliably for the Faber approximation [112, 82, 80], while some caution must be taken for the Newton interpolation, since the divided difference coefficients oscillate in the decaying phase. This may cause an underestimation of the real error. To eliminate this problem a check on a few coefficients before the last is usually sufficient.

### 3.2.8 Comparison between Faber approximation and Newton interpolation

Regarding the memory occupation, the two methods are equivalent: The Newton interpolation is based on a two term recursion while the Faber approximation is based on a three term recursion. Both need three copies of the matrix  $\rho$  to realize the recursive generation of  $\rho(t)$ . The preparation step (generation of points and/or coefficients) is negligible for the global computation time for both methods, if the Lejá algorithm is properly programmed. We found that the Faber method is definitely easier to implement: No points have to be generated and staggered and the coefficients are analytical.

To test the propagators a damped harmonic oscillator model was chosen, following the equation [46]:

$$\dot{\hat{\rho}}(t) = -i\omega [\hat{a}^\dagger \hat{a}, \hat{\rho}] + \gamma \left( \hat{a} \hat{\rho} \hat{a}^\dagger - \frac{1}{2} [\hat{a}^\dagger \hat{a}, \hat{\rho}]_+ \right).$$

Here,  $\hat{a}^\dagger$  and  $\hat{a}$  are the annihilation and creation operators, respectively,  $\omega = 0.02E_h$  is the oscillator frequency, and  $\gamma$  a damping constant. The dissipation was weak, because the values  $\gamma$  were  $\approx 10^{-2}$  times smaller than  $\omega$ . The system was represented on a grid of 128 points. The comparison is made for single step propagations.

As is clear from the example reported in Table 3.1, the two algorithms require about the same order to perform a time integration with the same accuracy. The two algorithms are very similar, so the resulting CPU times are only slightly different. The Faber algorithm was found to be more stable for higher orders. Many other calculations were performed, changing the accuracy of the integration, the length of the timestep, the size of the domain, and the strength of the dissipation, but all of them were consistent in showing the similarity of the two methods.

In conclusion, for dynamical problems the two methods are almost equally efficient, however, the Faber algorithm is easier to implement compared to the Lejá–Newton method.

The method was tested also for non–Lindblad forms of dissipation and explicitly

Table 3.1: Typical polynomial order necessary for a Newton and a Faber polynomial series to have a relative error in the energy smaller than  $10^{-4}$ .

time /[a.t.u.]	100	400	1000	2000	3000
Faber	277	1059	2509	5249	7795
Newton	273	1067	2545	5662	—

time dependent problems, using the model of Ref.[119] as test case. The method showed itself as reliable and efficient, thus confirming its qualities, and its general applicability to density matrix calculations.

### 3.2.9 Calculation of absorption spectra by polynomial expansions

#### Polynomial expansion of an absorption spectrum

Often, the propagated density matrix is used in other formulae to derive observables of interest. A typical case is the computation of a continuous wave (cw) absorption spectrum. For a weak, continuous wave field, there is a known expression for the absorption coefficient of a system embedded in a dissipative environment [120]

$$\alpha(\omega) = \frac{4\pi\omega n_{mol}}{nc} \mathcal{R} e \int_0^\infty dt e^{i\omega t} \text{tr} (\hat{\mu} e^{\mathcal{L}t} [\hat{\mu}, \hat{\rho}]) , \quad (3.24)$$

which can be viewed as a generalization of the so called Heller formula [17] to the dissipative case. Here,  $\omega$  is the light frequency,  $\hat{\mu}$  is the dipole operator,  $n_{mol}$  is the density of molecules,  $c$  the velocity of light, and  $n$  the refractive index. Setting  $\hat{\rho}' := [\hat{\mu}, \hat{\rho}]$ , the solution of (3.24) is equivalent to the propagation of a matrix

according to the dissipative Liouville-von Neumann equation

$$e^{\mathcal{L}t}\hat{\rho}' = e^{\mathcal{L}t} [\hat{\mu}, \hat{\rho}]. \quad (3.25)$$

The integration is done for a discrete number of timesteps; the trace is computed for each timestep and Fourier transformed. So it is a similar numerical problem as the one treated in the previous sections.

An interesting aspect of the polynomial integrators is that the time dependence is only in the coefficients and the representation dependence is left to the Faber or Newton recursion relations [121, 107, 122],

$$e^{\mathcal{L}t} [\hat{\mu}, \hat{\rho}] \approx \sum_{k=0}^n c_k(t) P_k(\mathcal{L}) \hat{\rho}'. \quad (3.26)$$

This implies that the integral in (3.24) acts only on the coefficients, leaving the recursion relations unchanged. The coefficients can be taken out of the trace operation.

Set  $K_s := \frac{4\pi n_{mol}}{nc}$ , then

$$\alpha(\omega) = K_s \mathcal{R} e \int_0^\infty dt e^{i\omega t} \text{tr} (\hat{\mu} e^{\mathcal{L}t} [\hat{\mu}, \hat{\rho}]) \approx K_s \mathcal{R} e \sum_{k=0}^n \underbrace{\int_0^\infty dt c_k(t) e^{i\omega t} \text{tr} (\hat{\mu} P_k(\mathcal{L}) \hat{\rho}')}_{s_k(\omega)}, \quad (3.27)$$

where the only approximation is in the polynomial expansion of the propagator. If the integration can be performed analytically a new series with coefficients  $s_k(\omega)$  will be obtained. The recursion relations will be unchanged, but it will refer directly to the spectrum. For the Newton interpolation the coefficients are not given analytically, so they can only be integrated numerically, thus loosing the exactness of the operation. This is not true for the new Faber approximation that will be considered now.

The coefficients  $s_k(\omega)$  of the Faber approximation for absorption spectra eval-

ation are derived from (3.27) evaluating the integral in time using [123]:

$$\begin{aligned} s_k(\omega) &= K_s \omega \int_0^\infty e^{i\omega t} c_k(t) dt = \\ &K_s \omega \left( \frac{-i}{\sqrt{-d}} \right)^k \int_0^\infty e^{i\omega t} e^{m\sigma t} J_k(2\sigma t\sqrt{-d}) dt = \\ &K_s \frac{\omega}{\sqrt{4\sigma^2 d + (m\sigma + i\omega)^2}} \left( \frac{\sqrt{4\sigma^2 d + (m\sigma + i\omega)^2} + (m\sigma + i\omega)}{2\sigma di} \right)^k. \end{aligned} \quad (3.28)$$

They are very similar to the Chebyshev series for computing dissociative Raman spectra [121] or the Faber series for the Green operator [107]. This is to be expected because the underlying framework of the three series is equivalent.

The equations to compute the spectrum with the Faber series are then:

$$\alpha(\omega) \approx \mathcal{R}e \sum_{k=0}^n s_k(\omega) \text{tr} (\hat{\mu} F_k(\mathcal{L}) \hat{\rho}'). \quad (3.29)$$

Where  $\hat{\rho}' := [\hat{\mu}, \hat{\rho}]$  and  $F_k(\mathcal{L})\hat{\rho}'$  are the Faber polynomials acting on an operator defined in section 3.2.3.

### 3.3 Infrared absorption spectrum of benzoic acid dimers

#### 3.3.1 Preliminary numerical considerations

The coefficients (3.28) are algebraic, so there is no need for special functions like Bessel functions. For this reason, it is much more stable and series with orders up to millions can be used.

A different series is needed for every  $\omega$  point in the spectrum<sup>10</sup>. The memory of present day computers would be filled if all of the coefficients of the series should be generated at once, if “ill conditioned” spectra<sup>11</sup> have to be computed. The memory

---

<sup>10</sup>Simply because the coefficients are  $\omega$  dependent.

<sup>11</sup>For example, when very narrow lines appear in a spectrum.

required would scale as the number of  $\omega$  values times the polynomial order. The terms (3.28) can be generated via a simple one term recursion relation, which is easy to determine from the last line of Eqn.(3.28). For this series only two copies of the matrix have to be held in memory for each frequency: the  $k$ th coefficient and the multiplicative constant and then via the recursion relation, the  $k + 1$ th term can be generated at the same time as the corresponding polynomial.

Looking at the behavior of the coefficients in Fig.3.6 (shown for a wide range of energies), it is clear that the smallest  $\omega$  can be taken as a reference for the convergence, its coefficients being always the largest, as soon as the modulus of the coefficients is small enough for the spectrum to be reasonably converged. Indeed,

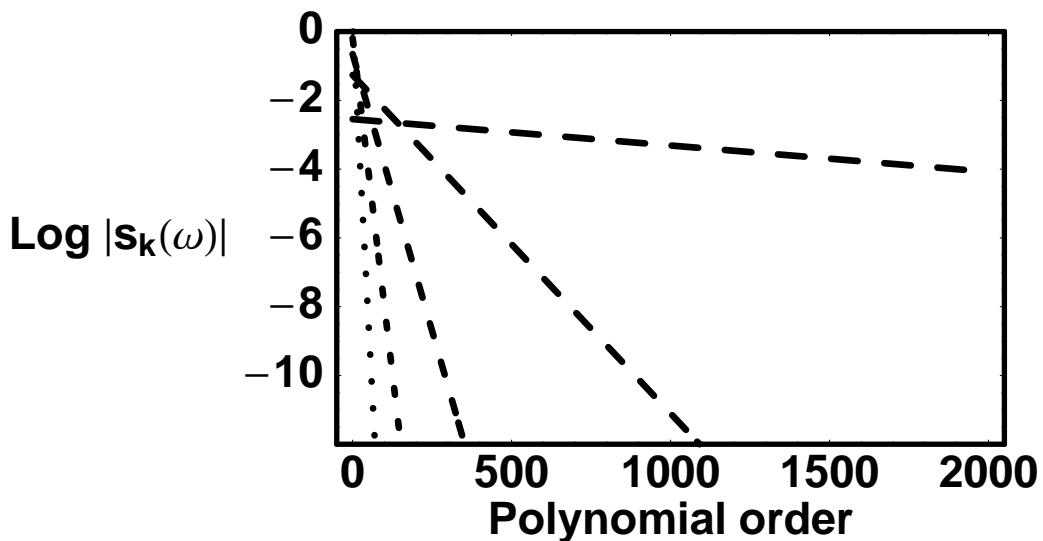


Figure 3.6: The logarithm of the modulus of the coefficient  $|s_k(\omega)|$  as a function of the polynomial order is plotted for  $\omega = 0.2, 0.7, 1.2, 1.7, 2.2 E_h$ ,  $m$  is set to  $0.2E_h$  and  $\sigma = 1.2$ .  $K_s$  is set to one. The larger the  $\omega$ , the smaller are the dashes in the dashed lines.

the limit for  $\omega \rightarrow \infty$  of the numerator of the  $n$ th-power term in (3.28) goes<sup>12</sup> to 0. The lower frequency for which the spectrum has to be evaluated should be very carefully chosen, because it influences the computation time by several orders of

---

<sup>12</sup>The standard (principal) branch must be used for the complex square root.

magnitude, the modulus of the coefficient for a certain  $n$  depending on  $\omega$  more than exponentially.

It is also possible to make an adaptive sampling of the spectral region, starting with a coarse grid of  $\omega$ s and refining where the spectrum has complex features, having only the traces of the polynomials to be held in the memory.

As already mentioned, the parameter  $m$  is related to the strength of the dissipation. In Fig.3.7 we can see which is the behavior of the coefficients for different  $m$ . It is interesting to notice that the series with coefficients (3.28) converges faster

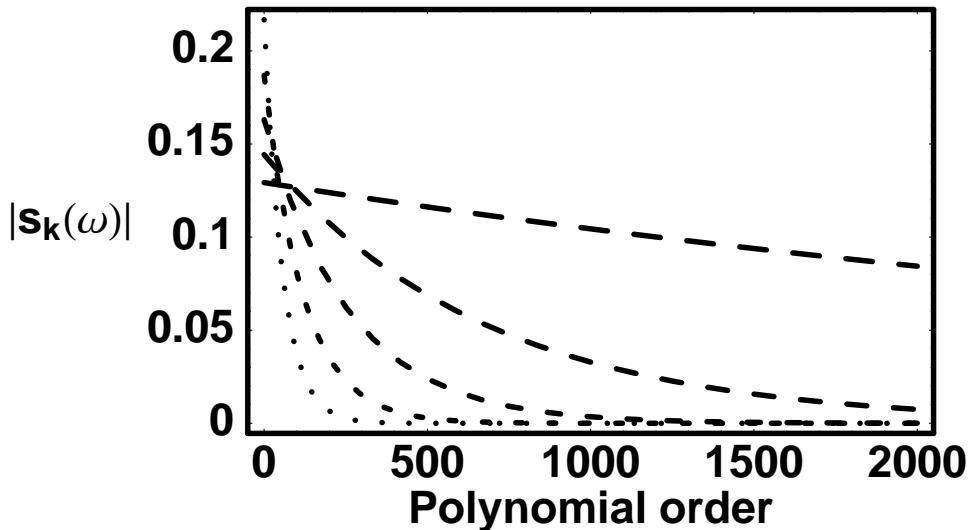


Figure 3.7: The modulus of the coefficients  $|s_k(\omega)|$  is plotted as a function of the polynomial order for  $m = -.05, -.25, -.45, -.65, -.85 E_h$ ,  $\omega$  is set to  $0.2 E_h$  and  $\sigma = 0.8$ .  $K_s$  is set to one. The bigger the modulus of  $m$  the smaller are the dashes in the dashed lines .

for a larger dissipation, as one would expect, since the dissipation will quench and decorrelate the system more efficiently. An overestimate of the magnitude of the dissipation can make the series decay too fast, and give un converged results, which show up as wild oscillations in the spectrum.

As common for all of the integrators, the bigger is the phase space volume of the dynamics, the slower is the convergence. This is reflected in the series with

coefficients (3.28), thus the factor  $\sigma$  has to be evaluated properly [12].

### 3.3.2 Absorption spectrum for the benzoic acid dimers

The method was applied to the evaluation of the IR cw-absorption spectrum of benzoic acid dimers embedded in benzoic acid crystals. For this purpose, a two dimensional, bound model (“hydrogen transfer mode” and “molecular frame mode”) was used. The operators were represented in the vibrational bound states of the model Hamiltonian. The individual levels  $|i\rangle$  relax due to vibrational energy dissipation. The relaxation operators  $\Gamma_{ij}$  were derived using a microscopic model. The relaxation is of Lindblad form. All details about the model can be found in Ref.[43]. There, the spectrum was obtained propagating a matrix (3.24) with a Newton polynomial integrator. In contrast to [43], we use the series (3.29).

For this model, the cw-spectrum can also be evaluated analytically, so information about the absolute error can be derived. The analytical spectrum is given by [43]:

$$\alpha(\omega) = \omega \sum_{i>j} \mu_{ij}^2 (g_j - g_i) \left( \frac{\Gamma_{ii} + \Gamma_{jj}}{(\Gamma_{ii} + \Gamma_{jj})^2 + (\omega + \omega_{ji})^2} - \frac{\Gamma_{ii} + \Gamma_{jj}}{(\Gamma_{ii} + \Gamma_{jj})^2 + (\omega - \omega_{ji})^2} \right), \quad (3.30)$$

where  $\Gamma_{ij}$  and  $\mu_{ij}$  refer to the matrix elements of dissipation and dipole moment operator respectively,  $\omega_{ij}$  to the frequency of the transition and  $g_i$  to the thermal populations for state  $i$ , i.e.  $\frac{e^{-\frac{E_i}{k_b T}}}{Q}$ . The  $\Gamma_{ii}$  are defined as  $\sum_{i \neq j} \Gamma_{ij}$ , as in Ref.[43].

At low temperature the spectrum shows a very sharp peak in the small frequency region (around  $60 \text{ cm}^{-1}$ ), as it will be illustrated below in Fig.3.10. This is a challenge for the convergence of the series, due to the very slow decay of the coefficients as a function of the polynomial order  $k$  for small  $\omega$ , and the oscillations in the polynomial terms generated by the delta like peak in the frequency domain.

Fig.3.8 shows the numerical effort for different choices of the smallest  $\omega$  to be

included in the spectrum. To converge the spectrum at a temperature of 40 K, a

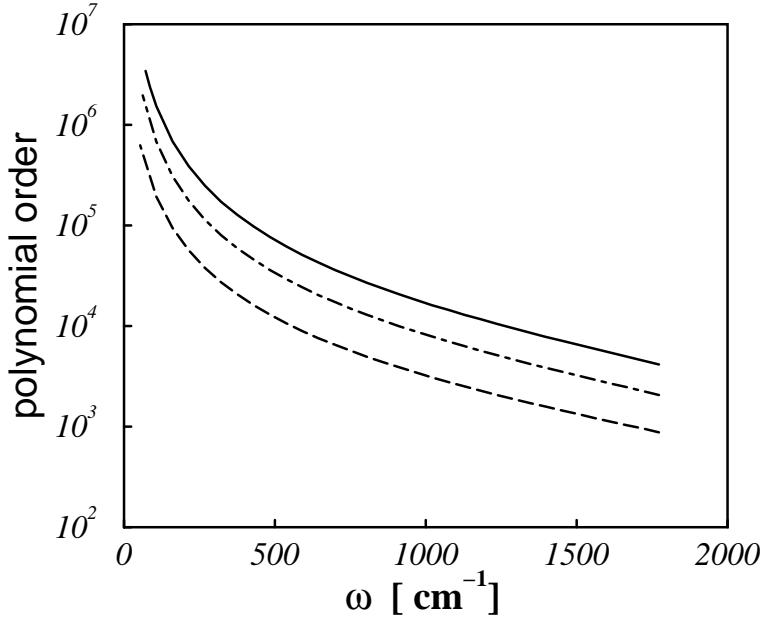


Figure 3.8: Order needed to have a converged spectral intensity. The polynomial order is plotted as a function of the frequency  $\omega$ , for a temperature of  $T = 40$  K. The three curves are obtained for a ratio between the modulus of the last coefficient and the first one equal to  $10^{-7}$  (solid line),  $10^{-5}$  (dot dashed),  $10^{-3}$  (long dashed), respectively.

series of order 4,000,000 was needed, which took about 40 min of CPU time (all the calculations were done on an SGI ORIGIN 2000 of ZEDAT, Freie Universität Berlin). From Fig.3.8, it is clear that to converge the complete spectrum except the sharp peak at  $\omega = 60$  cm<sup>-1</sup>, a time in the order of tens of seconds should be sufficient<sup>13</sup>. For the absorption spectrum at 40 K, the ratio between the modulus of the last coefficient and the first one was set to  $10^{-7}$ , while for the other calculations even a value of  $10^{-3}$  gives a spectrum almost indistinguishable from the analytical one. Fig.3.9 shows the sharp peak close to 60 cm<sup>-1</sup> in the 40 K spectrum, obtained with various accuracies in the Faber polynomial expansion, and compared to the analytical result. It is seen that with a ratio of the last to the first Faber polynomial expansion coefficient of  $10^{-7}$  even this very narrow peak ( 0.02 cm<sup>-1</sup> width) can be

<sup>13</sup>A globally coarser sampling of the spectrum and a lower accuracy would be sufficient.

reproduced exactly. For less accurate calculations, the computed peak is broader than the exact one and artificial oscillations become visible.

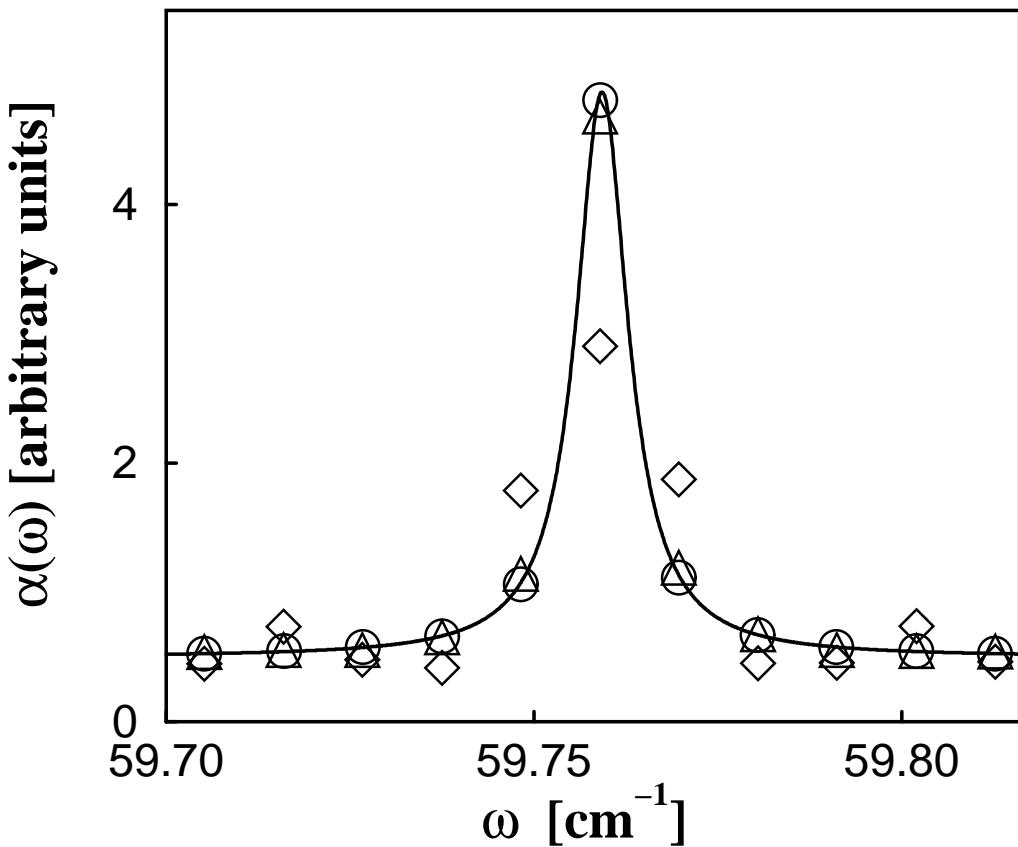


Figure 3.9: The absorption spectrum of benzoic acid dimers around the peak at  $59.76 \text{ cm}^{-1}$  is shown as computed for three different ratios between the modulus of the last coefficient and the first one:  $10^{-7}$  (circles),  $10^{-5}$  (triangles) and  $10^{-3}$  (diamonds). The analytical solution is depicted as a solid line.

In Fig.3.10, the spectra are also reported for different temperatures. In the lowest panel ( $T = 40 \text{ K}$ ) we recognize the very narrow peak around  $\omega = 60 \text{ cm}^{-1}$ , a bunch of broadened peaks is visible. The assignment of the individual peaks can be taken from [43]. For higher  $T$ , the well known trends are observed, that (1) temperature increases the magnitude of dissipation and hence makes the peaks broader, and (2) larger temperatures increase the importance of “hot bands” in the high  $\omega$  part of

the spectrum.

All spectra were in complete agreement with the analytical solutions, so that it is not possible to distinguish between computed and analytical curves in the figures. There is a certain quantitative disagreement with the spectra computed in [43], thus showing that this approach can improve accuracy in computing spectra in the presence of dissipation. In passing, we note that to have complete agreement between the analytical and propagated spectra, also the so called antiresonant terms have to be included in the analytic solution.

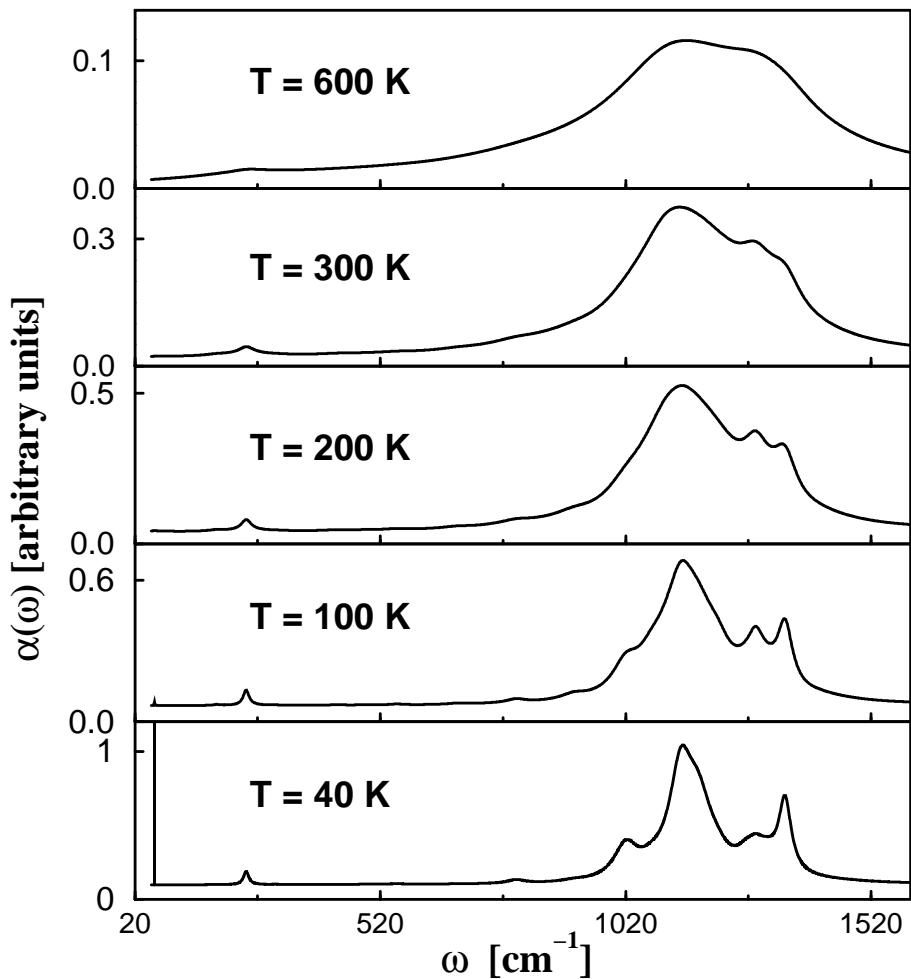


Figure 3.10: Spectra computed for  $T = 40, 100, 200, 300, 600 \text{ K}$  (from lowest to highest frame). The spectra are scaled in order to keep the peak at  $\approx 1140 \text{ cm}^{-1}$  at 80 % of the height of the frame. For the lowest temperature the higher peak would be the line at  $59.76 \text{ cm}^{-1}$ , whose features are plotted in Fig.3.9, in the same units. The analytical spectra are indistinguishable from the computed ones.

## 3.4 The Variational Wave Packet (VWP) method

### 3.4.1 Introduction

This method was recently introduced to integrate the dissipative Liouville-von Neumann equation for open quantum systems, in Ref.[95]. Its interesting features are that any quantities of interest can be converged with arbitrary accuracy, and density matrices can be represented as an ensemble of wave vectors, thus avoiding the at minimum quadratic scaling with the Hilbert space dimensions of the density matrix representation. Moreover, in comparison to the MCWP [Monte Carlo Wave Packet] method it can be applied to more general forms of dissipation, even to non-Markovian and non-positive generators. A reasonably accurate calculation of rare, improbable events is possible, even if the method is more suitable for frequent properties. The method is also efficient to compute observables depending dominantly on coherence, like spectra, where its variational behavior allows one to determine the proper dynamics with a small ensemble of functions.

### 3.4.2 Derivation of the VWP equations of motion

In this section, the equations of motion described in [95] are derived.

The dissipative Liouville-von Neumann equation, being first order in time, has to be solved subject to some initial condition, always expressible as

$$\hat{\rho}(t=0) = \sum_{i=0}^{\infty} \rho_i |\psi_i\rangle\langle\psi_i|,$$

where the  $\rho_i$  are real positive and the state vectors are orthonormal (see section 2.1).

For analogy, in the VWP method, the following ansatz is made for the open system (reduced) density matrix for any time  $t > 0$ :

$$\hat{\rho}(t) = \sum_{u=1}^n \sum_{v=1}^n \rho_{uv}(t) |\psi_u(t)\rangle\langle\psi_v(t)| \quad , \quad (3.31)$$

where  $|\psi_u(t)\rangle$  are time dependent expansion wave functions, and  $\rho_{uv}(t)$  time dependent coefficients defined as  $\rho_{uv}(t) = \rho_{vu}^*(t)$ , to keep the density matrix Hermitean. The approximative character of the ansatz lies in  $n$  being smaller than  $\infty$ . It is important to stress the similarities between this expression and the statistical definition of the density matrix (2.5). In principle, it is equivalent to the Multi Configurational Time Dependent Hartree (MCTDH) method or Multi Configurational Time Dependent Self Consistent Field (MCTDSCF) method, but the functions that have to be approximated (the density matrices) are exactly a sum of Hartree products, this being not true for general wavefunctions.

We need to derive the equations of motion for the  $|\psi_u(t)\rangle$  and the  $\rho_{uv}(t)$ . In (3.31), state vectors and scalar coefficients are arbitrary, and need only to approximate  $\hat{\rho}(t)$  when plugged in (3.31), but there are many possible sets that would satisfy this. For the  $|\psi_u(t)\rangle$ , we can formally write operator equations of the form

$$|\dot{\psi}_u(t)\rangle = -i\hat{h}|\psi_u(t)\rangle \quad (3.32)$$

to describe their time evolution. The only property required to the operator  $\hat{h}$  at this point is to reproduce correctly the global dynamics and to define unequivocally which vectors from the space spanned by the  $\{|\psi_u(t)\rangle\}$  would be used for the dynamics. In particular  $\hat{h}$  does not have to be linear; it is usually time dependent. We can include the boundary conditions for the dynamics directly in the evolution operators.

To derive the equations of motion, we use the Dirac–Frenkel time dependent variational principle [18], and follow the analogous derivation of McLachlan [19]. We want to find the solution of the Liouville–von Neumann equation with a restriction on the possible forms of  $\hat{\rho}(t)$ , but in this space, we want to approximate the exact dynamics as close as possible. Considering again the fact that the Liouville–von Neumann equation (2.9)

$$\dot{\hat{\rho}}(t) = \mathcal{L}\hat{\rho}(t) = -i [\hat{H}_s, \hat{\rho}(t)] + \mathcal{L}_D(\hat{\rho}(t)),$$

is a first order differential equation in time,  $\hat{\rho}(t)$  and the effect of all the operators on it are known at every time step  $t$ . Unknown is the differential, *i.e.*, the time derivative of the trial density operator (3.31), subject to the restriction of having the ansatz form. We will call this time derivative  $\hat{\Theta}(t)$ :

$$\hat{\rho}(t + \delta t) = \hat{\rho}(t) + \hat{\Theta}(t)\delta t.$$

A measure for the difference between this approximate time evolution and the exact one is given by

$$\text{tr} \left( \left( \mathcal{L}\hat{\rho}(t) - \hat{\Theta}(t) \right)^2 \right),$$

*i.e.*, by the square of the Frobenius norm.[ The density matrix norm as defined in (0.3) would not work, because it refers only to the diagonal elements of the density matrix]. To find the optimal evolution, we apply variational calculus by varying  $\hat{\Theta}(t)$  and computing

$$\text{tr} \left( \left( \mathcal{L}\hat{\rho}(t) - \hat{\Theta}(t) + \delta\hat{\Theta}(t) \right)^2 \right).$$

We take the functional derivative with respect to  $\Theta(t)$  (the part linear in  $\delta\hat{\Theta}(t)$ ) and set it to zero, in order to get a stationary value. All these operators being Hermitian and the trace operations commutative, one gets after some algebra the equation:

$$\text{tr} \left( \delta\hat{\Theta}(t) \left( \hat{\Theta}(t) - \mathcal{L}\hat{\rho}(t) \right) \right) = 0 \quad (3.33)$$

Since (3.33) holds for arbitrary  $\delta\hat{\Theta}(t)$ , we have:

$$\hat{\Theta}(t) - \mathcal{L}\hat{\rho}(t) = 0. \quad (3.34)$$

The time evolution in the restricted space must be equal to the Liouvillian applied to the restricted  $\hat{\rho}(t)$ .  $\hat{\Theta}(t)$  can be evaluated directly, being the derivative of the restricted density matrix, *i.e.*, we have to take the time derivative of the ansatz (3.31):

$$\begin{aligned} \hat{\Theta}(t) = \sum_{j,k=1}^n & \left[ \left( |\dot{\psi}_j(t)\rangle\langle\psi_k(t)| + |\psi_j(t)\rangle\langle\dot{\psi}_k(t)| \right) \rho_{jk}(t) + \right. \\ & \left. \dot{\rho}_{jk}(t) |\psi_j(t)\rangle\langle\psi_k(t)| \right]. \end{aligned} \quad (3.35)$$

We can apply the operator (3.34) to each  $|\psi_u(t)\rangle$

$$\left( \hat{\Theta}(t) - \mathcal{L}\hat{\rho}(t) \right) |\psi_u(t)\rangle = 0 \quad (3.36)$$

and compute its matrix elements between the functions chosen for the ansatz:

$$\langle\psi_v(t)| \left( \hat{\Theta}(t) - \mathcal{L}\hat{\rho}(t) \right) |\psi_u(t)\rangle = 0. \quad (3.37)$$

Substituting (3.35) in (3.37), yields

$$\sum_{j,k=1}^n \left[ (\langle \psi_v(t) | \dot{\psi}_j(t) \rangle \langle \psi_k(t) | \psi_u(t) \rangle + \langle \psi_v(t) | \psi_j(t) \rangle \langle \dot{\psi}_k(t) | \psi_u(t) \rangle) \rho_{jk}(t) + \dot{\rho}_{jk}(t) \langle \psi_v(t) | \psi_j(t) \rangle \langle \psi_k(t) | \psi_u(t) \rangle \right] = \langle \psi_v(t) | \mathcal{L}\hat{\rho}(t) | \psi_u(t) \rangle. \quad (3.38)$$

Since the starting functions can be arbitrarily chosen, we choose the  $\{|\psi_u(0)\rangle\}$  orthonormal.

In order to get their time evolution governed unequivocally by (3.32), we start to impose restrictions onto the operator  $\hat{h}$ :

- a) We require  $\hat{h}^\dagger = \hat{h}$ , i.e.,  $\hat{h}$  is chosen as Hermitean, so the functions  $|\psi_u(t)\rangle$  will conserve their norm;
- b) the functions are required to remain orthonormal for any time  $t > 0$ :  $\langle \psi_v(t) | \psi_u(t) \rangle = \delta_{uv}$ .

Then Eqn.(3.37) becomes together with (3.32)

$$-i \sum_{j=1}^n \left( \langle \psi_v(t) | \hat{h} | \psi_j(t) \rangle \rho_{ju}(t) - \rho_{vj}(t) \langle \psi_j(t) | \hat{h} | \psi_u(t) \rangle \right) + \dot{\rho}_{vu}(t) = \langle \psi_v(t) | \mathcal{L}\hat{\rho}(t) | \psi_u(t) \rangle. \quad (3.39)$$

Finally, for the  $\rho_{vu}(t)$ ,

$$\dot{\rho}_{vu}(t) = \langle \psi_v(t) | \left\{ \mathcal{L}\hat{\rho}(t) + i [\hat{h}, \hat{\rho}(t)] \right\} | \psi_u(t) \rangle. \quad (3.40)$$

Substituting (3.35) in (3.36), one gets:

$$\sum_{v=1}^n \rho_{vu}(t) | \dot{\psi}_v(t) \rangle = -i \hat{\rho}(t) \hat{h} | \psi_u(t) \rangle + \mathcal{L}\hat{\rho}(t) | \psi_u(t) \rangle - \sum_{v=1}^n | \psi_v(t) \rangle \dot{\rho}_{vu}(t) \quad (3.41)$$

These equations have not determined yet the  $\hat{h}$  completely, so we can include other conditions:

$$\text{c)} \quad \rho_{vu}(t) = \rho_u(t) \delta_{uv}, \quad \forall t.$$

In the MCTDH scheme, this is equivalent to work in the basis of natural single particle functions [124, 125, 126]. Consequently, the eigenvalues (populations) of the density matrix as a function of time  $\rho_u(t)$  are automatically determined as the weights  $\rho_u(t)$ .

With constraint c), Eqn.(3.39) becomes, for  $v \neq u$ :

$$\begin{aligned} 0 &= \dot{\rho}_{vu}(t) \Rightarrow \\ -i \left( \langle \psi_v(t) | \hat{h} | \psi_u(t) \rangle \rho_u(t) - \rho_v(t) \langle \psi_v(t) | \hat{h} | \psi_u(t) \rangle \right) &= \langle \psi_v(t) | \mathcal{L} \hat{\rho}(t) | \psi_u(t) \rangle \Rightarrow \\ \langle \psi_v(t) | \hat{h} | \psi_u(t) \rangle &= \frac{i}{\rho_u(t) - \rho_v(t)} \langle \psi_v(t) | \mathcal{L} \hat{\rho}(t) | \psi_u(t) \rangle. \end{aligned} \quad (3.42)$$

This operation is possible only if the populations for different functions are never the same, which is generally not the case. Nevertheless, the equations can be regularized, as it will be shown in section 3.4.3. Substituting in this equation  $\mathcal{L} \hat{\rho}(t)$  with  $-i [\hat{H}_s, \hat{\rho}(t)] + \mathcal{L}_D(\hat{\rho}(t))$ , and using (3.31) for the Hamiltonian part, we finally obtain:

$$\langle \psi_v(t) | \hat{h} | \psi_u(t) \rangle = \langle \psi_v(t) | \hat{H}_s | \psi_u(t) \rangle + \frac{i}{\rho_u(t) - \rho_v(t)} \langle \psi_v(t) | \mathcal{L}_D(\hat{\rho}(t)) | \psi_u(t) \rangle \quad (3.43)$$

With (3.43) the nondiagonal elements of the operator  $\hat{h}$  are defined. However, we still have to define the diagonal elements and their projection outside the space they span<sup>14</sup>. We can also derive the equations for the  $\rho_u(t)$  for the chosen constraints. Substituting constraint c) in Eqn.(3.39) we get

$$\begin{aligned} -i \left( \langle \psi_u(t) | \hat{h} | \psi_u(t) \rangle \rho_u(t) - \rho_u(t) \langle \psi_u(t) | \hat{h} | \psi_u(t) \rangle \right) + \dot{\rho}_u(t) &= \\ \langle \psi_u(t) | \mathcal{L} \hat{\rho}(t) | \psi_u(t) \rangle. \end{aligned} \quad (3.44)$$

The term in brackets in l.h.s. of Eqn.(3.44) is zero, so

$$\dot{\rho}_u(t) = \langle \psi_u(t) | \mathcal{L} \hat{\rho}(t) | \psi_u(t) \rangle.$$

---

<sup>14</sup>The space spanned by an incomplete set of functions can change in time, so the time evolution operators must have a component outside the  $\{|\psi_u(t)\rangle\}$ .

The expectation value of the commutator  $[\hat{H}_s, \hat{\rho}(t)]$  for the function  $|\psi_u(t)\rangle$  is zero, so we finally get:

$$\dot{\rho}_u(t) = \langle\psi_u(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle. \quad (3.45)$$

Then we apply constraint c) to (3.41) and get

$$\rho_u(t)|\dot{\psi}_u(t)\rangle = -i\hat{\rho}(t)\hat{h}|\psi_u(t)\rangle + \mathcal{L}\hat{\rho}(t)|\psi_u(t)\rangle - \sum_{v=1}^n \langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle|\psi_v(t)\rangle. \quad (3.46)$$

Substituting in this equation  $\mathcal{L}$  with  $-i[\hat{H}_s, \hat{\rho}(t)] + \mathcal{L}_D(\hat{\rho}(t))$ , as done before, and using (3.31) and (3.43) it follows

$$\rho_u(t)|\dot{\psi}_u(t)\rangle = -i\rho_u(t)\hat{H}_s|\psi_u(t)\rangle + i\rho_u(t)\langle\psi_u(t)|\hat{H}_s|\psi_u(t)\rangle \quad (3.47)$$

$$\begin{aligned} & -i\rho_u(t)|\psi_u(t)\rangle\langle\psi_u(t)|\hat{h}|\psi_u(t)\rangle \\ & + \sum_{v \neq u, v=1}^n |\psi_v(t)\rangle \frac{\rho_v(t)}{\rho_u(t) - \rho_v(t)} \langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle \\ & + \mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle - \langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle|\psi_v(t)\rangle. \end{aligned} \quad (3.48)$$

The diagonal elements of  $\hat{h}$ ,  $\langle\psi_u(t)|\hat{h}|\psi_u(t)\rangle$  do not couple any of the  $\{|\psi_u(t)\rangle\}$  and introduce, therefore, for each function only a phase factor.

Any observable is computed with (3.31). However, only diagonal elements occur because of constraint c), so the phases cancel out one another and can be defined arbitrarily as  $\langle\psi_u(t)|\hat{h}|\psi_u(t)\rangle := \alpha_u(t)$ . Usually  $\alpha(t)$  is set to zero to minimize the phase changes in the functions. Considering that

$$\sum_{v=1}^{\infty} |\psi_v(t)\rangle\langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle = \mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle$$

(if the expansion is complete, the sum is the identity operator  $\hat{I}$ ), and defining the projector on the basis used for the dynamics as

$$\hat{P} := \sum_{v=1}^n |\psi_v(t)\rangle\langle\psi_v(t)|,$$

we can write:

$$\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle = (\hat{I} - \hat{P})\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle + \sum_{v=1}^n |\psi_v(t)\rangle\langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle.$$

The equations of motion for the  $|\psi_u(t)\rangle$  in the form used for the dynamics are finally derived:

$$\begin{aligned} |\dot{\psi}_u(t)\rangle &= -i\hat{H}_s|\psi_u(t)\rangle + i|\psi_u(t)\rangle \left( \langle\psi_u(t)|\hat{H}_s|\psi_u(t)\rangle - \alpha_u(t) \right) \\ &\quad + \sum_{v=1; v \neq u}^n \frac{1}{\rho_u(t) - \rho_v(t)} |\psi_v(t)\rangle\langle\psi_v(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle \\ &\quad + \frac{1}{\rho_u(t)} (\hat{I} - \hat{P}) \mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle . \end{aligned} \quad (3.49)$$

In Eqn.(3.49), the first term on the r.h.s. is due to the Hamiltonian evolution of the system and the second one determines only a phase factor for the wave function  $|\psi_u(t)\rangle$ . The third and fourth terms account for the coupling of the system to the dissipative environment. Eqn.(3.49) and (3.45) are the basic equations of motion in the VWP method, giving the time evolution of the basis functions and their coefficients, respectively.

When a complete set of expansion wave functions is used, summing up (3.45) for every  $u$ , we have

$$\sum_{u=1}^n \rho_u(t) = \sum_{u=1}^n \langle\psi_u(t)|\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle. \quad (3.50)$$

If the number  $n$  goes to  $\infty$ , this is equivalent to the trace of  $\mathcal{L}_D(\hat{\rho}(t))$ . The trace of  $\mathcal{L}_D(\hat{\rho}(t))$  has to be zero for the generator to conserve the norm, so if the expansion is complete, the trace of the density matrix is conserved. For finite  $n$ , however, the trace decreases with increasing time and this loss of norm can be used as a convergence criterion for the proper choice of  $n$ , the number of terms in the expansion (3.31). If  $\mathcal{L}_D = 0$  (the system is subject to no dissipation), the norm is anyway conserved, this being consistent with the unitary character of the Hamiltonian dynamics.

### 3.4.3 Practical implementation of the VWP equations of motion

The VWP method is a time dependent method, so we have to define initial conditions. If a pure state is used as initial state, one sets  $n$ , the number of basis functions, to some small value<sup>15</sup>. If one starts with a mixed state, one has to use as many functions as necessary to have a representation of the state with an accuracy compatible with the requirements of the propagation. Practically, one has to diagonalize the density matrix and take as many  $|\psi_u(t=0)\rangle$  with their  $\rho_u(t=0)$  as necessary to have an error in the trace  $\sum_{u=1}^n \rho_u(t=0)$  smaller than the error tolerance of the simulation.

After the propagation has been started, one adds a new function as soon as the total norm loss in a timestep exceeds some prespecified accuracy threshold<sup>16</sup> and the smallest  $|\rho_i(t)|$  is larger than this same threshold<sup>17</sup>. If a function has to be added to the ensemble, it is chosen as

$$|\psi_{n+1}(t)\rangle = (\hat{I} - \hat{P}) \sum_{u=1}^n \mathcal{L}_D(\hat{\rho}(t)) |\psi_u(t)\rangle,$$

and normalized.  $\hat{P}$  is again the projector on the subspace presently spanned by the functions at time  $t$ . The new function is generated orthogonal to the other ones (as is required by the algorithm) and in the “direction” where the system is evolving. The initial weight  $\rho_{n+1}(t)$  is chosen smaller than the accuracy required for the dynamics, hence adding no error to the propagation. The weight will be quickly adjusted by the evolution itself.

As mentioned above, in Eqn.(3.49), there is a singularity (a major source of stiff-

---

<sup>15</sup>The notion of “small” depends on the strength of the dissipation. Usually two functions are enough: One is the initial pure state, the other one to reproduce the dissipative dynamics at short times.

<sup>16</sup>The threshold is related to the error tolerance. Not to forget that this is a local error, so the final error is  $\approx \sum_{i=1}^{N_{steps}} |Err_i|$ , where  $N_{steps}$  is the total number of integration steps, and  $|Err_i|$  the error in timestep  $i$ .

<sup>17</sup>It would be of no use to add a function if the population of the last does not increase, the error in this case is likely due to the integrator (for instance due to the use of too long timesteps).

ness, giving rise to very fast unstable dynamics for states with similar populations):

$$\lim_{\rho_u(t) \rightarrow \rho_v(t)} \frac{1}{\rho_u(t) - \rho_v(t)} = \infty.$$

A possible solution for this problem is to use a regularization scheme, and to replace the singular term by

$$\frac{\rho_u(t) - \rho_v(t)}{(\rho_u(t) - \rho_v(t))^2 + \epsilon^2}. \quad (3.51)$$

Here,  $\epsilon$  has to be a function of the difference in population of the two states in order to have a regularization effect, and not to perturb too much the actual dynamics. A convenient choice for  $\epsilon$  is

$$\epsilon = \tilde{\epsilon} \exp \left[ - \left( \frac{\rho_u(t) - \rho_v(t)}{\tilde{\epsilon}} \right)^2 \right].$$

This solution is not completely satisfactory, because contrary to its equivalent counterpart in the MCTDH method [124], it is not only applied to negligibly populated functions [95, 96]. Thus the regularization scheme can become critical. For instance, when many population “crossings” are generated, which is the case in the simulation of the photodesorption processes with both simultaneous excitation and quenching, as described in the next chapter. Moreover, the  $\rho_u(t)$  never really cross during the time evolution<sup>18</sup>, so when a function must become less populated than another one, the two functions must exchange their characteristics, and this can be numerically very inefficient. To tackle this problems, one must be careful in choosing the parameter  $\tilde{\epsilon}$  and the error tolerance required to an integration step. There is not yet a general solution to this problem, because it is pathologically ingrained in the equations of motion of the VWP method. Nevertheless, the problems have been overcome for most of the problems it has been applied to [95, 96].

To implement the equations of motion, first the term  $\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle$  has to be evaluated. The evaluation of this term depends strongly on the dissipation form chosen, but in principle is a known operation, giving a new wavefunction. To determine the time derivative of  $\rho_u(t)$  via (3.45), this state vector has to be integrated with the bra  $\langle\psi_u(t)|$ . The projection on the other states allows to compute the

---

<sup>18</sup>They show the so called avoided crossings [95].

summation term in (3.49), and together with (3.45) the  $-\hat{P}\mathcal{L}_D(\hat{\rho}(t))|\psi_u(t)\rangle$  term; then the resulting terms have to be summed up with the part of the equations of motion depending on the Hamiltonian  $\hat{H}_s$  (which is straightforward to implement) to generate the time derivative of  $|\psi_u(t)\rangle$ .

For the time evolution a predictor–corrector integrator with adjustable timesteps is adopted [95]. During the propagation, the error is controlled. The sources of errors in the VWP method are the following:

- e1) The error in the “diagonalization” of  $\hat{\rho}(t)$  is proportional to the norm loss in the trace (see the end of next section for a detailed discussion), and depends on the number of functions included in the ansatz (3.31).
- e2) The error due to the integrator, which should be added to e1). Usually this error is easily kept smaller than the first one.
- e3) There are the errors coming from the discretization and truncation of the Hilbert space. Their behavior is known from and analogous to ordinary wave packet propagations [13, 127].