JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Computer Algebra Tools for Special Functions in High Order Finite Element Methods

Dissertation

zur Erlangung des akademischen Grades

Doktorin der Technischen Wissenschaften

Angefertigt am *SpezialForschungsBereich F013 - Numerical and Symbolic Scientific Computing*

Begutachter:

Prof. Dr. Peter Paule
Prof. Dr. Joachim Schöberl, RWTH Aachen University

Eingereicht von:

Veronika Pillwein

Linz, March 2008

Abstract

High order finite element methods are one of the most commonly used techniques for obtaining numerical solutions to partial differential equations on non-trivial domains. The given domain is subdivided into simple geometric objects and an approximate solution is computed as a linear combination of locally supported basis functions. Starting from a variational formulation of the partial differential equation the discretization yields a (usually large) system of linear equations that is commonly solved using iterative methods. The performance of the iterative solvers is closely related to the choice of basis functions. Customarily finite element basis functions are defined by means of orthogonal polynomials.

Orthogonal polynomials belong to a class of sequences for which in the past decades various methods for a symbolic treatment have been devised. This symbolic treatment includes automatic proving of known identities as well as automatic finding of new identities, such as algebraic dependencies or closed form representations for symbolic sums. Implementations of these algorithms are available for all major computer algebra systems.

In this thesis we employ these techniques to solve problems that arose in the context of high order finite element methods. As a first application we show how to find and prove automatically recurrence relations for certain edge and vertex based basis functions. Next, we propose families of basis functions yielding a sparse system matrix for some elliptic boundary values problems. We present an algorithmic proof for the sparsity. Furthermore, we prove and extend a positivity conjecture on a weighted sum over Legendre polynomials that was formulated in the context of a convergence proof for a certain high order finite element scheme. In the final chapter, the construction of a stable polynomial projection operator with applications to a posteriori error estimates is presented.

Zusammenfassung

Zur numerischen Lösung partieller Differentialgleichungen auf nontrivialen Gebieten werden bevorzugt high order Finite Elemente Methoden verwendet. Das gegebene Gebiet wird dabei in einfache geometrische Objekte unterteilt und eine Näherungslösung als Linearkombination von lokal definierten Basisfunktionen berechnet. Ausgehend von einer Variationsformulierung der partiellen Differentialgleichung wird das Problem in ein (üblicherweise hochdimensionales) lineares Gleichungssystem überführt. Zur Lösung werden im Allgemeinen iterative Methoden verwendet, deren Konvergenzverhalten von einer geeigneten Wahl der Basisfunktionen abhängt. Üblicherweise werden diese Basisfunktionen aus orthogonalen Polynomen konstruiert.

Orthogonale Polynome gehören zu einer Klasse von Funktionenfolgen, für die in den letzten Jahrzehnten verschiedene Methoden zur symbolischen Behandlung entwickelt wurden, wie das automatische Beweisen bekannter Identitäten aber auch das automatische Finden neuer Identitäten, wie algebraische Abhängigkeiten oder geschlossene Darstellungen symbolischer Summen. Implementierungen dieser Algorithmen sind in allen führenden Computeralgebrasystemen verfügbar.

In der vorliegenden Arbeit setzen wir diese Algorithmen zur Lösung von verschiedenen Problemen ein, die im Zusammenhang mit high order Finite Elemente Methoden aufgetreten sind. Als erste Anwendung zeigen wir, wie Rekursionen für neue kanten- und vertexbasierte Basisfunktionen automatisch gefunden und bewiesen werden können. Weiters präsentieren wir Familien von Basisfunktionen die für bestimmte elliptische Randwertprobleme zu dünnbesetzten Systemmatrizen führen. Der Beweis dafür wird über einen Algorithmus, den wir in Mathematica implementiert haben, erbracht. Außerdem beweisen und erweitern wir eine Vermutung über die Positivität einer gewichteten Summe von Legendre Polynomen, die im Zusammenhang mit dem Konvergenzbeweis für ein bestimmtes Finite Elemente Modell aufgestellt wurde. Abschließend beschreiben wir die Konstruktion eines stabilen, polynomialen Projektionsoperators mit Anwendungen für a posteriori Fehlerschätzer.

Contents

Abstract **i**

Zusammenfassung **iii**

1 Introduction **1**

2 The Finite Element Method **5**

2.1 Function Spaces . . . 5
2.2 Variational Formulation . . . 8
2.3 Basic Principles of the Finite Element Method . . . 9
2.4 Finite Element Basis Functions . . . 11
2.5 Finite Element System Matrix . . . 14
2.6 A One Dimensional Example . . . 16

3 Algorithms for Special Functions **19**

3.1 Holonomic Sequences . . . 19
3.2 Orthogonal Polynomials . . . 23
3.3 Algorithms and RISC Symbolic Summation Packages . . . 26

4 Jacobi Polynomial Identities and Computer Algebra **33**

4.1 Jacobi and Integrated Jacobi Polynomials . . . 33
4.2 Generating mixed difference-differential relations . . . 39

5 Hypergeometric Summation Algorithms for p-FEM **41**

5.1 Edge Based Basis Functions . . . 41
5.2 Low Energy Vertex Based Basis Functions . . . 46

6 Sparse Shape Functions for High Order-FEM **51**

6.1 Motivation . . . 51
6.2 Definition of the Basis Functions . . . 53
6.3 Algorithm for Proving Theorems 6.1 and 6.2 . . . 57
6.4 Numerical Properties and Applications . . . 64
6.5 Further Results . . . 67

7 Positive Jacobi kernel polynomial sums **73**

7.1 Motivation . . . 73
7.2 Related Results and Idea of Proof . . . 75

7.3 Chebyshev Polynomials of First and Second Kind ($\alpha = \pm\frac{1}{2}$) . . . 76
7.4 Jacobi Polynomials $P_n^{(\alpha,\alpha)}(x)$ with $-\frac{1}{2} < \alpha < \frac{1}{2}$ . . . 78
7.5 Envelope for the Oscillating Part . . . 83

8 Stable Polynomial Projectors and Error Estimates 87
8.1 Construction of the Projector . . . 87
8.2 Formulation of the Postprocessing Method . . . 95
8.3 Right Inverse on Tensor Product Elements . . . 97
8.4 Equilibrated Residual Error Estimates . . . 100

A Mathematica Implementations 103
A.1 Fem2D Package . . . 103
A.2 IntJac Package . . . 107

Notation and Symbols 111

Bibliography 112

Index 119

Eidesstattliche Erklärung 119

Curriculum Vitae 121

Chapter 1

Introduction

Many problems in science and engineering are described by partial differential equations on non-trivial domains which, except in special cases, cannot be solved analytically. Numerical methods are required to solve these equations. The finite element method (FEM) [28, 22, 19] has in the last decades become the most popular tool for obtaining solutions of partial differential equations on complicated domains. The main advantage of the finite element method is its general applicability to a huge class of problems, linear as well as nonlinear partial differential equations, coupled systems, varying material coefficients and boundary conditions.

Finite element methods are based on the variational formulation of partial differential equations. The domain of interest is subdivided by simple geometrical objects such as triangles, quadrilaterals, tetrahedra, or hexahedra. The approximate solution is expanded in a (finite) basis of local functions, each supported on a finite number of elements in the subdivision. There are three main strategies to improve the accuracy of the approximate solution.

The classical approach is to use on each element basis functions of a fixed low polynomial degree, say $p = 1, 2$, and to increase the number of elements in the subdivision. This strategy of local or global refinement of the mesh is called the h-version of the finite element method, where h refers to the diameter of the elements in the subdivision. With this approach the approximation error decays algebraically (i.e., with polynomial rate) in the number of unknowns.

An alternative strategy is to keep the mesh fixed and to locally increase the polynomial degree p of the basis functions. This method is called the p-version of the finite element method [76, 78] and, in the case of a smooth solution, this approach leads to exponential convergence with respect to the number of unknowns. But in practical problems usually the solutions are not smooth. In this case the convergence rate of the p-method degenerates again to an algebraic one.

Exponential convergence can be restored by combining both strategies in the hp-version of the finite element method [76, 49, 31]. On parts of the domain where the sought solution is smooth, few coarse elements with basis functions of high polynomial degrees are used, whereas in the presence of singularities, caused, e.g., by re-entrant corners, the polynomial degree is kept low and the mesh is refined locally towards the singularity. The p- and the hp-method are also referred to as high(er) order finite element methods. In Chapter 2 we give a brief introduction to the finite element method.

While the approximation with piecewise polynomial basis functions of high degree requires considerably less parameters [82, 76], the implementation of high order methods is much more involved. Hence, every simplification of the algorithms is most welcome.

In this thesis we consider several aspects of this challenging task, where as a key instrument we apply recently developed computer algebra algorithms for special functions. The crossover point between high order finite element methods and these symbolic algorithms in our work are the finite element basis functions. Basis functions for high order finite element methods are usually defined by means of certain orthogonal polynomials. We are mainly interested in obtaining recurrence relations or simplifications (closed forms) for expressions involving orthogonal polynomials, as well as inequalities entering in, e.g., norm or convergence estimates.

A large number of relations for orthogonal polynomials can be found in literature, see e.g. [1, 69, 79]. Such relations can, however, on demand also be discovered and proven automatically. In the past decades a variety of algorithms has been developed for this purpose and we give a short overview on some of them below.

The orthogonal polynomials considered in this thesis can be represented as sums over hypergeometric terms, i.e., terms satisfying a first order linear difference equation. Given $f(n,k)$, hypergeometric in both n and k, it can be decided algorithmically whether the sum $F(n) = \sum_{k=1}^{n} f(n,k)$ admits a closed form representation as linear combination of hypergeometric terms. Furthermore, if such a closed form exists, it can be computed automatically. The underlying algorithms are due to Gosper [45], Zeilberger [88], and Petkovšek [66]. Based on WZ-theory [85], Wegschaider has implemented a Mathematica package for discovering and proving hypergeometric multi-sum identities [83].

Symbolic summation algorithms, however, exist also for more general expressions than hypergeometric terms such as functions described by so-called holonomic systems of differential-difference equations. Orthogonal polynomials can be described via three term recurrences, or as solutions to differential equations with polynomial coefficients. In other words, they are holonomic and these symbolic summation algorithms are applicable.

The holonomic universe is closed under certain operations and these operations can again be carried out algorithmically using the representation via difference or differential equations. In the univariate case, we refer to the implementations by Salvy and Zimmermann [70] or Mallinger [59].

One approach to symbolic summation due to Zeilberger [87] uses a representation via operators that annihilate the given (possibly multivariate) expression in a certain non-commutative operator algebra. An algorithm for proving and finding identities for holonomic functions using Gröbner basis computations is due to Chyzak [25].

A further generalization of classical symbolic summation algorithms that finds closed forms of expressions involving nested sums and products has been developed by Karr [50, 51]. The sequences covered by Karr's algorithm are called $\Pi\Sigma$-sequences. Schneider [72] has generalized this algorithm in various directions and due to him is also the only known implementation.

Kauers has devised algorithms for finding algebraic dependencies, symbolic summation, deciding zero equivalence and proving inequalities that cover most of the cases mentioned above and extends to a class of "admissible" sequences, see [52]. He has also implemented this algorithm in his package SumCracker [53].

In Chapter 3 we state the precise definition of holonomic sequences and functions and give more details on holonomic closure properties. Section 3.3 contains further descriptions of the implementations of the symbolic summation algorithms that are employed in this thesis. First applications are given in Chapter 4 where Jacobi and integrated Jacobi polynomials are introduced.

Via the finite element discretization the given partial differential equation (in variational form) is transformed into systems of linear equations to be solved for the coefficients of the expansion in the finite element basis. These usually large equation systems are commonly solved by iterative methods, whose performance depends on properties of the system matrix, such as a sparse structure and a small condition number. Also of interest is a fast assemblance of the system matrix, or, to be more precise, a fast implementation of the matrix-vector product. These demands are influenced by a diligent choice of basis functions.

In Chapter 5 the construction of edge and vertex based basis functions for the high order finite element method is presented for which by means of symbolic summation algorithms recurrence relations have been found that allow for a cheap implementation of these basis functions. In [75] these high order basis functions were introduced and it was shown that the application of cheap block-Jacobi preconditioners is efficient for the proposed basis. In the joint paper [10] Schneider's package Sigma [72] was applied to obtain recurrence relations for these basis functions, whereas here we have chosen to present alternative proofs using different algorithms. The edge based basis functions are defined through an extension procedure of functions defined on the real line similar to the approach by Muñoz-Sola [62], for which also cheap recurrence relations can be generated, see Section 5.1.

Next, in Chapter 6, families of basis functions for triangular and tetrahedral p-finite element methods for discretizing elliptic boundary value problems are proposed that yield a sparse system matrix in the case of a piecewise constant coefficient function and a polygonally bounded domain. More precisely, the number of nonzero matrix entries is proportional to the number of unknowns, which allows for a fast evaluation of the system matrix. The sparsity is proven using a program we implemented in the computer algebra system Mathematica. A further consequence of the sparse structure is the preconditioning of the block of cell based basis functions, which is also applicable for uniformly elliptic second order boundary value problems with arbitrary coefficients. The proposed basis functions are extensions of the bases given in [49, 18]. The contents of this chapter are joint work with Sven Beuchler [16, 17].

The problem treated in Chapter 7 arose in a convergence proof for a certain high order finite element scheme [74]. In this context Schöberl conjectured the positivity of a weighted sum of Legendre polynomials. We have extended this conjecture to weighted sums over certain ultraspherical Jacobi polynomials and proven it by combining human insight and application of computer algebra [67]. So far neither a fully human nor a fully automated proof for this result is known.

In the hp-version of the finite element method adaptive mesh refinement is crucial to avoid an unnecessary high number of unknowns. The decision where to increase the polynomial degrees of the basis functions and where to subdivide the elements further is usually based on a posteriori error estimates. In Chapter 8 the construction of a stable polynomial projection operator with applications in equilibrated residual error estimates is presented. The proof of the norm estimates for the projection operator requires less applications of computer algebra than the previous chapters, but it relies on several results given earlier in the thesis. This is joint work with Dietrich Braess and Joachim Schöberl [20].

From the results presented in this thesis it is obvious that computer algebra techniques have a significant impact on the design of high order finite methods and that this research area offers many interesting topics for future investigation.

Acknowledgements

First of all I want to thank my advisors Peter Paule and Joachim Schöberl for guiding my work and sharing their curiosity and enthusiasm. Their mutual interest for each other's research area created a perfect atmosphere for bringing together numerical and symbolic computation. Both are open for discussions with their students at any time despite their tight schedule. Their motivation and energy is reflected in their respective groups with whom it is a pleasure to work.

I am very grateful to Manuel Kauers for introducing me to his SumCracker package, improving my LaTeX skills and general background information.

At the same time I want to thank Sven Beuchler for sharing his insights on high order finite element methods and a pleasant and fruitful cooperation.

Special thanks go to Martin Aigner, Christoph Koutschan, Flavia Stan, and Sabine Zaglmayr for numerous discussions on and beyond Mathematics as well as for proof-reading.

Moreover, I want to thank all my colleagues of the Special Research Area SFB F013 "Numerical and Symbolic Scientific Computing" for the good and inspiring working climate. Last but not least, I want to acknowledge the financial support of the Austrian Science Fund "Fonds zur Förderung der wissenschaftlichen Forschung" FWF under the grant SFB F013/F1301.

Chapter 2

The Finite Element Method

This chapter contains a brief introduction to the finite element method, in particular the high order finite element method. We define the underlying function spaces and introduce some customary notations. The basic principles of the finite element method are described and illustrated with a one dimensional example. We do not intend to give a complete survey, for instance existence and uniqueness results are not discussed at all. For more informations we refer to (higher order) finite element literature such as Braess [19], Brenner and Scott [22], Karniadakis and Sherwin [49], Ciarlet [28], Demkowicz [31], Szabó and Babuška [78] or Schwab [76].

2.1 Function Spaces

Throughout this chapter let the domain Ω be an open and connected subset of $\mathbb{R}^d$, where in applications we assume that $d = 1, 2, 3$. The closure of a set M is denoted by $\overline{M}$. We define $C_0^\infty(\Omega)$ as the space of infinitely differentiable functions with compact support on Ω. Functions in $C_0^\infty(\Omega)$ are supported on a compact subset of Ω with positive distance to the boundary, i.e., they vanish on $\partial\Omega$. We denote by $L^2(\Omega)$ the Lebesgue space of square integrable functions on the domain Ω:

$$L^2(\Omega) = \{f : \Omega \to \mathbb{R} \mid \int_\Omega f(x)^2 \, dx < \infty\}.$$

Note that $f = g$ in $L^2(\Omega)$ means equality almost everywhere, i.e., equal up to sets of measure zero. The function space $L^2(\Omega)$ is a Hilbert space with inner product

$$(f, g)_{L^2(\Omega)} = \int_\Omega f(x)\, g(x) \, dx.$$

We omit specifying the domain whenever no confusion can arise. The norm induced by this inner product is given by

$$\|f\|^2_{L^2(\Omega)} = (f, f)_{L^2(\Omega)} = \int_\Omega f(x)^2 \, dx.$$

More generally, L^p-spaces, $1 \leq p < \infty$, are defined as

$$L^p(\Omega) = \{f : \Omega \to \mathbb{R} \mid \int_\Omega |f(x)|^p \, dx < \infty\}.$$

These spaces are Banach spaces equipped with the norm $\|f\|^p_{L^p(\Omega)} = \int_\Omega |f(x)|^p\,dx$.

Next we introduce Sobolev spaces. For this purpose we need the notion of weak derivatives. To motivate the definition let us first consider a continuously differentiable function $f \in C^1(a,b)$ and a function $v \in C_0^\infty(a,b)$, for some open real interval (a,b). By partial integration one obtains

$$\int_a^b f(x)v'(x)\,dx = -\int_a^b f(x)'v(x)\,dx + \underbrace{[f(x)\,v(x)]_a^b}_{=0}.$$

Since $v(x)$ vanishes at the endpoints of the interval, the boundary terms vanish. Based on this identity the notion of differentiability can be generalized. A function $f \in L^2(a,b)$ is *weakly differentiable*, if there exists a locally integrable function[1] $w \in L^1_{loc}(a,b)$ satisfying

$$\int_a^b f(x)v'(x)\,dx = -\int_a^b w(x)v(x)\,dx$$

for all $v \in C_0^\infty(a,b)$. If such a function $w(x)$ exists, it is unique almost everywhere. In notation we follow common practice and do not differ between classical (strong) and weak derivatives, i.e., we write $f'(x) = w(x)$. Obviously weak and strong derivatives coincide, if the function f is differentiable in the classical sense. Note that functions with jumps, such as the Heavyside function, are not weakly differentiable. Continuous functions that are piecewise differentiable certainly have a weak derivative. For higher derivatives and dimensions $d > 1$ weak derivatives are defined accordingly. Let α be some multi-index, then the αth weak derivative $D^\alpha f$ of a square integrable function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$, $x \mapsto f(x)$, is defined via

$$\int_\Omega D^\alpha f(x)\,v(x)\,dx = (-1)^{|\alpha|}\int_\Omega f(x)D^\alpha v(x)\,dx,$$

for all $v \in C_0^\infty(\overline{\Omega})$, where $|\alpha| = \sum_{i=1}^d \alpha_i$. We are mainly concerned with *weak gradients.* Motivated by Gauss' law the weak gradient $\nabla f \in (L^1_{loc}(\Omega))^d$ for a function $f \in L^2(\Omega)$ is defined as

$$\int_\Omega \nabla f \cdot v(x)\,dx = -\int_\Omega f(x)\,\operatorname{div} v(x)\,dx,$$

for all $v \in (C_0^\infty(\overline{\Omega}))^d$, where the divergence of a vector valued function v is $\operatorname{div} v = \sum_{i=1}^d \frac{dv_i}{dx_i}$. The *Sobolev space* $H^1(\Omega)$ consists of L^2-functions whose weak derivative is component-wise again square integrable, i.e.,

$$H^1(\Omega) = \{f \in L^2(\Omega) \mid \nabla f \in (L^2(\Omega))^d\}.$$

This space is also a Hilbert space equipped with the inner product

$$(f,g)_{H^1(\Omega)} = (f,g)_{L^2(\Omega)} + (\nabla f, \nabla g)_{L^2(\Omega)},$$

inducing the norm

$$\|f\|^2_{H^1(\Omega)} = (f,f)_{L^2(\Omega)} + (\nabla f, \nabla f)_{L^2(\Omega)} = \|f\|^2_{L^2(\Omega)} + |f|^2_{H^1(\Omega)},$$

[1] $w \in L^1_{loc}(\Omega)$ iff, for every compact subset K of Ω, the restriction of w to K is in $L^1(K)$.

where $|\cdot|_{H^1(\Omega)}$ denotes the H^1-semi norm. For higher order derivatives D^α, $|\alpha| \le k$, the Sobolev space $H^k(\Omega)$ is defined accordingly. These Sobolev spaces are again Hilbert spaces with the inner product

$$(f,g)_{H^k(\Omega)} = \sum_{|\alpha|\le k} (D^\alpha f, D^\alpha g)_{L^2(\Omega)},$$

that induces the norm $\|f\|^2_{H^k(\Omega)} = (f,f)_{H^k(\Omega)}$. Viewing the space of square integrable functions as the Sobolev space with square integrable weak derivatives up to the zeroth order explains the customary notations

$$H^0(\Omega) = L^2(\Omega) \qquad \text{and} \qquad (f,g)_0 = (f,g)_{L^2(\Omega)}.$$

The K-interpolation method, cf. [12, 11], gives an equivalent norming of Sobolev spaces $H^s(\Omega)$ that can be extended to fractional orders of differentiation s. The interpolation spaces $[X_0, X_1]_\theta$, $0 < \theta < 1$, of two Banach spaces X_0, X_1, define a scale of spaces lying in a sense between X_0 and X_1. The norm of the interpolation space $[X_0, X_1]_\theta$ is defined by means of the K-functional

$$K(f,t;X_0,X_1) = \inf_{f=f_0+f_1} \big(\|f_0\|_{X_0} + t\,\|f_1\|_{X_1}\big), \qquad t>0,$$

and given by

$$\|f\|_\theta = \left(\int_0^\infty \left[t^{-\theta} K(f,t;X_0,X_1)\right]^2 \frac{dt}{t}\right)^{1/2}.$$

For $X_0 = L^2(\Omega) = H^0(\Omega)$ and $X_1 = H^1(\Omega)$ it can be shown that

$$K(f,t;H^0,H^1) \approx \min\{1,t\}\|f\|_{L^2(\Omega)} + \sup_{|h|\le t} \|f(\cdot+h) - f\|_{L^2(\Omega)}, \tag{2.1}$$

see [11, Chapter 5.4]. For the definition of the H^s-seminorm only the second term on the right hand side of (2.1) is needed. For $s \in (0,1)$ the Sobolev space $H^s(\Omega)$ is defined as

$$H^s(\Omega) = [H^0(\Omega), H^1(\Omega)]_s = \{f \in L^2(\Omega) \mid \|f\|^2_{L^2(\Omega)} + |f|^2_{H^s(\Omega)} < \infty\}.$$

An equivalent renorming is obtained with the seminorm $|\cdot|_{H^s(\Omega)}$ given by

$$|f|_{H^s(\Omega)} = \int_\Omega \int_\Omega \frac{|f(x)-f(y)|^2}{|x-y|^{2s+d}}\, dx\, dy.$$

Next we briefly introduce the spaces needed for treating boundary values of weakly differentiable functions. How to define the *trace* of a function $f \in H^1(\Omega)$ on $\partial\Omega$, i.e., its values at $\partial\Omega$, is not obvious, because $\partial\Omega$ has measure zero in $\mathbb{R}^d$. Still, for $H^s(\Omega)$, with $s > \frac{1}{2}$, the trace of a function on $\partial\Omega$ can be defined uniquely in the L^2-sense.

For the Sobolev space $H^s(\Omega)$, $s > \frac{1}{2}$, we define the space with zero trace on the boundary $\partial\Omega$ as the closure of $C_0^\infty(\Omega)$ in the H^s-norm, i.e.,

$$H_0^s(\Omega) = \overline{C_0^\infty(\Omega)}^{\|\cdot\|_{H^s(\Omega)}}.$$

For $s > \frac{1}{2}$ the *trace operator* tr mapping a function into its restriction on the boundary is a continuous operator from $H^s(\Omega)$ to $H^{s-1/2}(\Omega)$. Conversely there exists a continuous lifting operator $\mathcal{E} : H^{s-1/2}(\Omega) \to H^s(\Omega)$, such that $\operatorname{tr}(\mathcal{E}f) = f$, $f \in H^{s-1/2}(\Omega)$. The space $H^{s-1/2}(\Omega)$ is needed for given boundary values of a partial differential equation in order to assure continuous dependence of the solution on the input data.

We close this section by defining $H^{-s}(\Omega)$ to be the dual space of $H_0^s(\Omega)$, i.e., as the space of continuous linear functionals on $H_0^(s)$. The value of a functional $u \in H^{-s}(\Omega)$ at a function $v \in H_0^s(\Omega)$ are denoted by $\langle u, v\rangle$. The space $H^{-s}(\Omega)$ is equipped with the dual norm

$$\|u\|_{H^{-s}(\Omega)} = \sup_{v \in H_0^s(\Omega)} \frac{\langle u, v\rangle}{\|v\|_{H^s(\Omega)}}.$$

2.2 Variational Formulation

Finite element methods operate on "weak" or "variational" formulations of partial differential equations. We demonstrate how to obtain the variational formulation starting from a classical (strong) formulation of the Poisson equation.

Example 2.1. *Given a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, and a function f we consider the boundary value problem*

$$\begin{aligned} \text{find } u: \qquad -\Delta u &= f, \qquad \text{in } \Omega, \\ u &= 0, \qquad \text{on } \partial\Omega. \end{aligned} \tag{2.2}$$

We seek a solution to this problem in the Hilbert space $V = H_0^1(\Omega)$, which requires the weak formulation of the equation. This formulation is obtained by first multiplying the differential equation by smooth test functions $v \in C_0^\infty(\Omega)$ and integrating over the domain. Via integration by parts we have

$$\int_\Omega \nabla u \nabla v \, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, ds = \int_\Omega f v \, dx,$$

where n denotes the unit outer normal vector of Ω. Since v vanishes on $\partial\Omega$ the boundary integral is zero and we arrive at the variational formulation of (2.2),

$$\text{find } u \in V: \qquad \int_\Omega \nabla u \nabla v \, dx = \int_\Omega f v \, dx, \qquad \text{for all} \quad v \in V. \tag{2.3}$$

The choice $V = H_0^1(\Omega)$ meets the minimal regularity requirements needed for this equation to be well defined, including also the zero boundary condition $u = 0$ on $\partial\Omega$.

The quantities in (2.3) define a symmetric bilinear form $a(\cdot, \cdot)$ and a linear form $F(\cdot)$ respectively. The bilinear form and linear form for this problem are given by

$$a(u, v) = \int_\Omega \nabla u \nabla v \, dx \qquad \text{and} \qquad F(v) = \int_\Omega f v \, dx.$$

Thus we can rewrite (2.3) as

$$\text{find } u \in V: \qquad a(u, v) = F(v) \qquad \text{for all} \quad v \in V,$$

or, by the Riesz representation theorem, in operator notation, defining $A = a(\cdot, v)$, $F = F(v)$,

$$\text{Find } u \in V: \qquad Au = F, \qquad \text{in } V', \tag{2.4}$$

where V' denotes the dual space of V.

The general procedure to obtain a weak formulation is just as demonstrated in Example 2.1. The given partial differential equation is multiplied by some smooth test function, integrated over the domain and partial integration is performed. The solution is then sought in a larger space, namely some Sobolev space, that is determined by the weak formulation to which the original problem was transformed. This change of spaces is also often necessary, since on complicated domains there might not exist smooth solutions $u \in C^2(\Omega)$ at all. In the next section we briefly introduce the method we have chosen for solving such variational problems.

2.3 Basic Principles of the Finite Element Method

The finite element method (FEM) is a special instance of Galerkin methods which provide a general technique to solve problems of the form

$$\text{Find } u \in V: \qquad a(u,v) = F(v) \qquad \forall v \in V, \tag{2.5}$$

where V is some infinite dimensional space, $a(\cdot,\cdot)$ is a given bilinear form and $F(\cdot)$ is a given linear form. Now we approximate V by a sequence of finite dimensional spaces $V_h \subset V$ yielding the discrete problems

$$\text{Find } u_h \in V_h: \qquad a(u_h,v_h) = F(v_h) \qquad \forall v_h \in V_h. \tag{2.6}$$

By solving these systems approximations u_h to the exact solution u of (2.5) are obtained. Here and in the following the subscript h means that we are dealing with some discrete object. Let N_h denote the dimension of V_h. Then V_h is spanned by a set of basis functions $\{\varphi_0, \dots, \varphi_{N_h-1}\}$. We expand the approximate solutions u_h in terms of these basis functions

$$u_h(x) = \sum_{i=0}^{N_h-1} u_i \varphi_i(x).$$

It is now sufficient to consider as test functions these basis functions, i.e., $v_h = \varphi_j$, $0 \le j \le N_h - 1$. Hence we can rewrite (2.6) accordingly:

$$\text{Find } \underline{u} = (u_0, \dots, u_{N_h-1}) \in \mathbb{R}^{N_h}: \qquad \sum_{i=0}^{N_h-1} u_i a(\varphi_i, \varphi_j) = F(\varphi_j), \qquad \forall\, 0 \le j \le N_h - 1.$$

Defining the system matrix A with entries $A_{i,j} = a(\varphi_i, \varphi_j)$ and the vector $\underline{f}$, $f_j = F(\varphi_j)$, we arrive at the linear system,

$$\text{Find } \underline{u} \in \mathbb{R}^{N_h}: \qquad A\underline{u} = \underline{f}. \tag{2.7}$$

If the bilinear form $a(\cdot,\cdot)$ is of the form

$$a(u,v) = \int_\Omega \lambda(x) \nabla u(x) \cdot \nabla v(x)\, dx,$$

for some coefficient function $\lambda(x)$, then the corresponding system matrix is called *stiffness matrix*. This bilinear form appears in the variational formulation of the partial differential equation $\operatorname{div}(\lambda(x) \cdot \nabla u(x)) = f(x)$. The Poisson equation discussed in Example 2.1 is a special case of this equation for $\lambda(x) \equiv 1$. System matrices with entries stemming from the bilinear form $a(u,v) = \int_\Omega \mu(x) u(x) v(x)\, dx$, for some coefficient function $\mu(x)$, are called *mass matrix*.

The finite element method can be characterized as a special Galerkin method with the following properties:

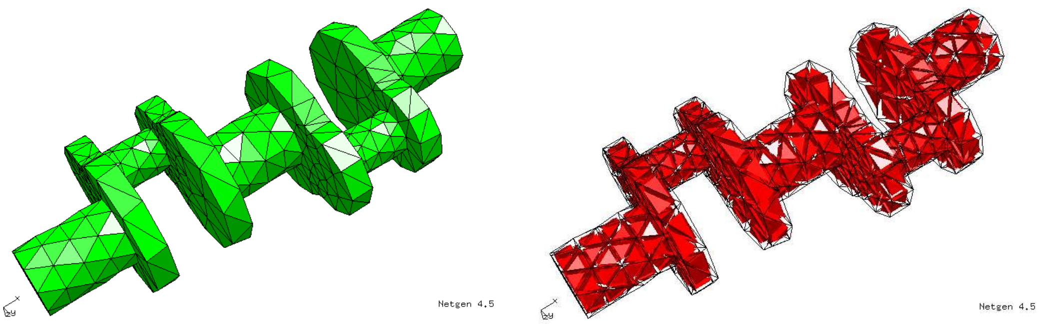


Figure 2.1: Finite element mesh for a crankshaft, left: surface mesh, right: interior tetrahedral elements

- the domain $\overline{\Omega}$ is subdivided into a finite set of simple geometric objects such as lines, triangles, bricks, etc. which are referred to as elements; they are collected in a triangulation $\mathcal{T}_h$ (see below);
- the basis functions φ_i have local support, i.e., they are nonzero on just a finite number of elements of $\mathcal{T}_h$;
- the space V_h consists typically of piecewise polynomial functions.

Assume that the domain $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ is a bounded polygonal or polyhedral domain with Lipschitz continuous boundary[2]. Next we characterize a triangulation.

A *triangulation* (mesh) $\mathcal{T}_h = \{K_i\}_{i \in \mathcal{I}}$ is a finite, non-overlapping partition of Ω into simple geometrical objects K_i. A triangulation is called *admissible* if

1. the elements are non-overlapping, i.e., $\mathring{K}_i \cap \mathring{K}_j = \emptyset,\ i \neq j$;
2. the triangulation $\mathcal{T}_h$ is a covering of $\overline{\Omega}$, i.e., $\overline{\Omega} = \bigcup_{i \in \mathcal{I}} K_i$;
3. the intersection $K_i \cap K_j$ of two different elements $(i \neq j)$ is either empty or a vertex or an edge or a face of both elements.

The index h refers to the mesh size of the triangulation and is related to the diameter of elements of $\mathcal{T}_h$. A triangulation is called *quasi-uniform*, iff there exists $\kappa > 0$ such that each element $K \in \mathcal{T}_h$ contains a circle with radius $\rho_T \geq h/\kappa$. The sets of all vertices, edges, faces and interior cells of elements of the triangulation $\mathcal{T}_h$ are denoted by $\mathcal{V}$, $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{C}$. We only consider admissible triangulations.

One advantage of the finite element method is its ability of solving partial differential equations on non-trivial domains. Figure 2.1 shows a possible mesh for a crankshaft. This mesh was generated by NETGEN, an automatic mesh generator developed by Joachim Schöberl [73].

A key ingredient in the analysis, design and implementation of finite elements is the use of reference elements. The physical elements in the mesh, $K_i \in \mathcal{T}_h$, are defined as images of a

[2] $\partial\Omega$ is a Lipschitz continuous boundary, iff for every $x \in \partial\Omega$ there exists an environment in which $\partial\Omega$ can be represented as graph of a Lipschitz continuous function.

reference element $\hat{K}$ of a (possibly non-linear) isoparametric mapping $F_i : \hat{K} \to K_i$. Several computations such as numerical integration or derivation can be performed a-priori on the reference element and then be mapped onto the actual elements in the mesh.

2.4 Finite Element Basis Functions

In this thesis we consider applications of symbolic summation methods to high order finite element methods, where "high order" refers to using polynomial basis functions up to some "high" degree p. In general one distinguishes three types of finite element methods based on their strategies how to improve the accuracy of the approximate solution, the h-, p- and hp-method.

In the h-version of the finite element method one uses basis functions of fixed (low) polynomial degree p, usually $p = 1, 2$. In order to decrease the error of the numerical solution, the mesh size h is decreased, i.e., the approximate solution is computed on a finer grid. The h-method converges algebraically (i.e., with polynomial rate), that is, the approximation error decays algebraically in the number of unknowns.

An alternative strategy to obtain a higher accuracy is to increase the polynomial degree p of the shape functions on a fixed mesh. This method is called the p-version of the finite element method. If the solution is an analytic function, then using this approach leads to exponential convergence. But in practical applications quite frequently we encounter singularities in the sought solutions. In this situation the convergence of the p-method deteriorates to an algebraic one.

Exponential convergence can be regained by combining both strategies in the hp-version of the finite element method. In case of piecewise analytic solutions involving singularities caused, e.g., by re-entrant corners, edges, etc., the mesh is refined locally towards the singularities. On the finer parts of the grid the basis functions are chosen to be of low polynomial degree, whereas on the coarse parts of the mesh basis functions of higher degree are used.

Concerning polynomial spaces we denote by $P^p(I)$ the space of polynomial functions on the interval $I \subset \mathbb{R}$ with maximal degree $p \geq 0$. The space of polynomials vanishing at the endpoints of I is denoted by $P_0^p(I)$. For simplicial elements $\mathcal{T}$, i.e., triangles or tetrahedra, $P^p(\mathcal{T})$ is the space of polynomials of total degree p, whereas if $\mathcal{T}$ is a quadrilateral or hexahedral element, $P^p(\mathcal{T})$ denotes the space of polynomials of maximal degree p in each variable. If different maximal degrees are used for the variables, we write this explicitely, e.g., $P^{p_1,p_2}(\mathcal{T})$ for $\mathcal{T}$ a quadrilateral and maximal degree p_i for the variable x_i, $i = 1, 2$. The subscript 0 is again used for polynomial functions that vanish at the boundary of $\mathcal{T}$.

In this section we describe the construction of high order finite element basis functions. These basis functions are associated to vertices, edges, faces and cells of $\mathcal{T}_h$ and their defining properties are stated next.

Definition 2.2. *(Higher order basis functions) All basis functions characterized below are piecewise polynomial and continuous.*

- **Vertex based basis functions:** φ_{V_i} *satisfy the nodal basis property, i.e.,* $\varphi_{V_i}(V_j) = \delta_{i,j}$*, for all* $V_i \in \mathcal{V}$*. Basis functions associated to vertices vanish on all faces not containing the associated vertex* V_i*. Usually vertex based basis functions are chosen to be piecewise linear.*

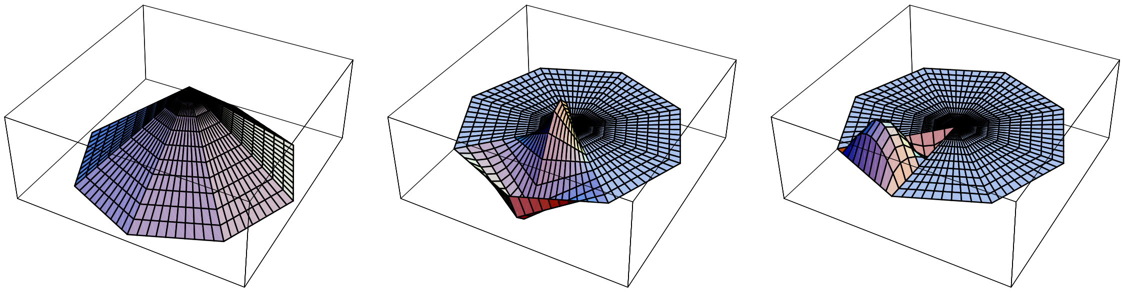


Figure 2.2: Vertex, edge based and cell based basis functions on a triangular mesh

- **Edge based basis functions:** *φ_{E_i}, $E_i \in \mathcal{E}$, span $P_0^p(E_i)$ on the associated edge E_i and vanish on all faces not containing E_i.*
- **Face based basis functions:** *φ_{F_i}, $F_i \in \mathcal{F}$, vanish on all faces except the defining one. They are supported on the two elements sharing F_i.*
- **Cell based basis functions:** *the support of φ_{C_i}, $C_i \in \mathcal{C}$, is the defining element only.*

In the context of the finite element method these basis functions are also called shape functions. We denote the set of all vertex based basis functions as $\Phi_V = \{\varphi_V \ : \ V \in \mathcal{V}\}$ and define $\Phi_E = \{\varphi_E \ : \ E \in \mathcal{E}\}$, $\Phi_F = \{\varphi_F \ : \ F \in \mathcal{F}\}$ and $\Phi_C = \{\varphi_C \ : \ C \in \mathcal{C}\}$ accordingly. The set of all basis functions Φ is given by the vector $\Phi = [\Phi_V, \Phi_E, \Phi_F, \Phi_C] = [\varphi_0, \ldots, \varphi_{N_h-1}]$.

Figure 2.2 shows the three possible types of basis functions in two dimensions for a triangular mesh. Note that this construction allows to vary the polynomial degrees for each vertex, edge, face or cell. Solutions $u = \sum u_i \varphi_i$ are by construction continuous across element interfaces and piecewise differentiable, i.e. u is weakly differentiable. Usually *hierarchic* shape functions are chosen, i.e., the set of shape functions of polynomial degree up to p should be contained in the set of shape functions up to polynomial degree $p+1$. We describe the construction of such a basis starting with the one-dimensional case using the reference element $\hat{I} = [-1, 1]$. The shape functions described below were introduced by Szabó and Babuška [78].

Most commonly orthogonal polynomials are used for the construction of higher order shape functions. In Section 3.2 an overview on orthogonal polynomials is given, for the moment we only need Legendre and integrated Legendre polynomials. Let $P_n(x)$ denote the nth Legendre polynomial. Legendre polynomials are pairwise orthogonal with respect to the L^2-inner product on the interval $[-1, 1]$, i.e.,

$$\int_{-1}^{1} P_i(x) P_j(x)\, dx = \frac{2}{2i+1} \delta_{i,j}, \tag{2.8}$$

where $\delta_{i,j}$ denotes the Kronecker delta. The first few Legendre polynomials are

$$P_0(x) = 1, \qquad P_1(x) = x, \qquad P_2(x) = \frac{1}{2}\left(3x^2 - 1\right), \qquad P_3(x) = \frac{1}{2} x \left(5x^2 - 3\right).$$

Moreover we define the nth integrated Legendre polynomial for $n \geq 2$ and $x \in [-1, 1]$ as

$$L_n(x) = \int_{-1}^{x} P_{n-1}(s)\, ds. \tag{2.9}$$

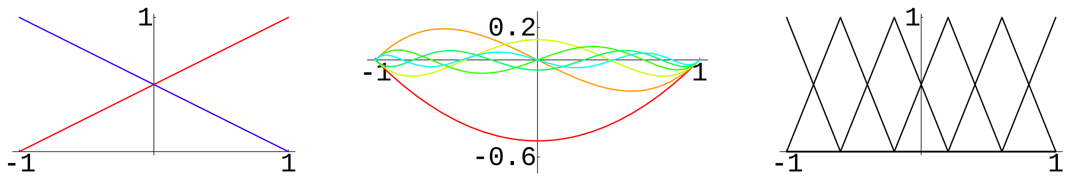


Figure 2.3: Vertex and cell based basis functions on the reference element $[-1, 1]$, $p = 7$, all hat functions on a mesh with five congruent elements

For $n = 0, 1$ we define $L_0(x) = -1$ and $L_1(x) = x$. The first few integrated Legendre polynomials, for $n \geq 2$, are

$$L_2(x) = \frac{1}{2}(x-1)(x+1), \qquad L_3(x) = \frac{1}{2}(x-1)x(x+1), \qquad L_4(x) = \frac{1}{8}(x-1)(x+1)\left(5x^2 - 1\right).$$

Because of (2.8) integrated Legendre polynomials are orthogonal with respect to the inner product of the H^1-semi norm

$$(f, g)_1 = \int_{-1}^{1} f'(x)g'(x)\,dx.$$

Observe also that $L_n(\pm 1) = 0$. For $x = -1$ this is obvious and for $x = 1$ this follows from orthogonality relation (2.8). We define the vertex based shape functions on $\hat{I}$ to be the standard hat functions

$$\varphi_0(x) = \frac{1-x}{2}, \qquad \phi_1(x) = \frac{1+x}{2}, \tag{2.10}$$

and the cell based basis functions as integrated Legendre polynomials,

$$\varphi_i(x) = L_i(x), \qquad 2 \leq i \leq p. \tag{2.11}$$

The family of polynomials $\{\varphi_i\}_{i=0}^{p}$ forms a basis of $P^p(\hat{I})$, see also Figure 2.3.

Next we construct shape functions for two dimensions starting with quadrilateral elements. In the design of these shape functions usually the implicit tensor product structure is exploited.

Let $\hat{Q} = [-1, 1]^2$ be the reference square with vertices and edges numbered as shown in Figure 2.4. Then we define the $2D$ basis functions on $\hat{Q}$ using the previously defined one-dimensional ones as follows:

- Vertex shape functions:

$$\begin{aligned} \varphi^{V_0}(x, y) =& \varphi_0(x)\varphi_0(y), \qquad & \varphi^{V_1}(x, y) =& \ \varphi_1(x)\varphi_0(y), \\ \varphi^{V_2}(x, y) =& \varphi_1(x)\varphi_1(y), \qquad & \varphi^{V_3}(x, y) =& \ \varphi_0(x)\varphi_1(y). \end{aligned} \tag{2.12}$$

- Edge based basis functions:

$$\begin{aligned} \varphi_i^{E_0}(x, y) =& \varphi_i(x)\varphi_0(y), \qquad & \varphi_i^{E_1}(x, y) =& \ \varphi_1(x)\varphi_i(y), \\ \varphi_i^{E_2}(x, y) =& \varphi_i(x)\varphi_1(y), \qquad & \varphi_i^{E_3}(x, y) =& \ \varphi_0(x)\varphi_i(y), \end{aligned} \tag{2.13}$$

  for $2 \leq i \leq p$.

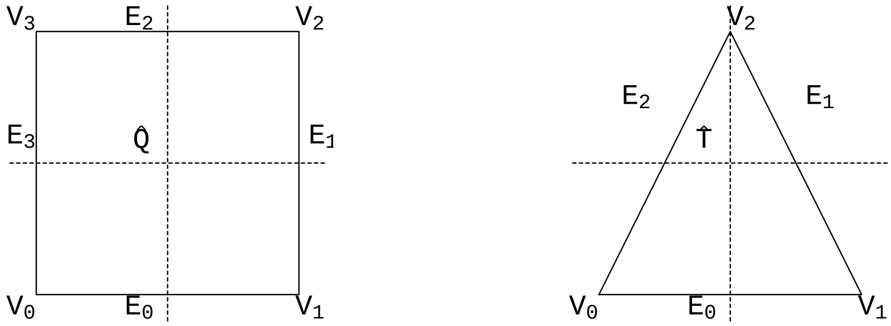


Figure 2.4: Reference square $\hat{Q}$ and reference triangle $\hat{T}$

- Cell based basis functions:

$$\varphi^C_{i,j}(x,y) = \varphi_i(x)\varphi_j(y), \qquad 2 \le i,j \le p. \tag{2.14}$$

It is easily verified that $[\Phi_V, \Phi_E, \Phi_C]$ satisfy the conditions stated in Definition 2.2 using the properties of the one dimensional basis functions.

In the construction of high order basis functions for triangles we use a tensor-product-like structure. This can be obtained by collapsing the quadrilateral to the triangle via the mapping $(x,y) \mapsto (x(1-y),y)$, see [49, 36]. Let the reference triangle $\hat{T}$ be given by the vertices $\{(-1,-1),(1,-1),(0,1)\}$ with vertices and edges according to Figure 2.4. The vertex based basis functions are given as the linear functions satisfying $\varphi_{V_i}(V_j) = \delta_{i,j}$, i.e.,

$$\varphi^{V_0}(x,y) = \frac{1-2x-y}{4}, \quad \varphi^{V_1}(x,y) = \frac{1+2x-y}{4}, \quad \varphi^{V_2}(x,y) = \frac{1+y}{2}. \tag{2.15}$$

The edge based basis functions for the first edge E_0 are given by

$$\varphi^{E_0}_i(x,y) = \varphi_i\left(\frac{2x}{1-y}\right)(1-y)^i, \qquad 2 \le i \le p, \tag{2.16}$$

where $\varphi_i(x)$ are again the previously defined one-dimensional finite element basis functions. The shape functions for the other two edges can be constructed analogously. Finally the cell based basis functions are defined as

$$\varphi^C_{i,j}(x,y) = \varphi_i\left(\frac{2x}{1-y}\right)(1-y)^i\varphi_j(y), \qquad i \ge 2,\ j \ge 1,\ i+j \le p. \tag{2.17}$$

Because of the factor $(1-y)^i$, both edge and cell based basis functions, are polynomials in x and y. The construction principles described above can be carried over to the three dimensional case for prisms, tetrahedra and hexahedra, see e.g. [49, 60].

2.5 Finite Element System Matrix

We have stated the definition of high order shape functions locally on one (reference) element. The global shape functions are defined piecewise and consist of contributions of several elements. When building up the global system matrix $A = (a(\varphi_i, \varphi_j))_{i,j=0}^{N-1}$, first the local element matrices are computed. Their entries are then added up in the corresponding positions of the global system matrix A.

The structure of the system matrix is sparse, since the supports of the basis functions only overlap on a finite number of elements. Based on the ordering of the shape functions $\Phi = [\Phi_V, \Phi_E, \Phi_F, \Phi_C]$ one can write the system matrix (locally and globally) in the following block structure

$$A = \begin{bmatrix} A_{V,V} & A_{V,E} & A_{V,F} & A_{V,C} \\ A_{E,V} & A_{E,E} & A_{E,F} & A_{E,C} \\ A_{F,V} & A_{F,E} & A_{F,F} & A_{F,C} \\ A_{C,V} & A_{C,E} & A_{C,F} & A_{C,C} \end{bmatrix}, \tag{2.18}$$

where $A_{V,V} = [a(\varphi_i^V, \varphi_j^V)]_{i,j}$ and the other blocks are defined analogously. The dimension N of the system matrix depends on the topology of mesh and on the associate polynomial degrees. For instance the mesh depicted in Figure 2.1 consists of 1622 elements. On a single tetrahedron with uniform degree p we have a total of $\frac{1}{6}(p+1)(p+2)(p+3)$ shape functions.

Basically there are two options for solving a linear system, either directly or using iterative solvers. Direct solvers are based on Gaussian elimination exploiting the structure of the system matrix. Because of its sparsity it is possible to perform a reordering of the unknowns such that the bandwidth is reduced and the profile is optimized. Using direct solvers that take the bandwidth into account a lot of computational effort can be saved, see [71]. But usually iterative solvers are preferred, since inverting even a sparse matrix is extremely costly.

The most frequently used iterative solvers are based on conjugate gradient methods. One advantage of iterative solvers is that they need less memory than direct solvers, but the number of iterations increases with the condition number, i.e., the ratio of the largest and the smallest singular value of a matrix,

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

A common strategy to speed up the rate of convergence is to use preconditioned conjugate gradient methods if $\kappa(A)$ is large. The linear system $Au = f$ is multiplied by a regular matrix C^{-1}, called preconditioner, to obtain the system

$$C^{-1}Au = C^{-1}f,$$

that is solved instead of the original problem. The matrix C is chosen such that the condition number of $C^{-1}A$ is small (at least smaller than $\kappa(A)$) and such that the matrix vector product with C^{-1} can be computed efficiently. Setting $C = A$ only one iteration is needed, but its inversion is very costly, whereas setting $C = I$, the identity matrix, is very cheap, but ineffective. So, in a sense, something in between these two choices is needed, for instance $C = \mathrm{diag}(A)$, the diagonal entries of A. Another possibility is solving exactly on the "coarse" level and use diagonal preconditioning on the "finer" level, i.e.,

$$C = \begin{bmatrix} A_{V,V} & 0 & 0 & 0 \\ 0 & \mathrm{diag}(A_{E,E}) & 0 & 0 \\ 0 & 0 & \mathrm{diag}(A_{F,F}) & 0 \\ 0 & 0 & 0 & \mathrm{diag}(A_{C,C}) \end{bmatrix}. \tag{2.19}$$

2.6 A One Dimensional Example

We close this introductory chapter with an elementary example comparing the h- and p-version of the finite element method. Consider the problem given in Example 2.1 in one dimension, i.e.,

$$\text{Find} \quad u \in C^2[0,1]: \qquad -u''(x) = f(x), \qquad u(0) = u(1) = 0, \tag{2.20}$$

where we choose the right hand side $f(x) = \sin(4\pi x)$. The exact solution is easily computed to be $u_{ex}(x) = \frac{1}{16\pi^2}\sin(4\pi x)$. Both methods are compared using the same numbers of unknowns. We choose $p = 8$ and $p = 16$ on a single element for the p-version and hat functions on eight, respectively 16 congruent elements for the h-version.

The variational formulation of (2.20) is

$$\text{Find} \quad u \in H_0^1[0,1]: \qquad a(u,v) = f(v), \qquad \text{for all} \quad v \in H_0^1[0,1],$$

where

$$a(u,v) = \int_0^1 u'(x)v'(x)\,dx, \quad \text{and} \quad f(v) = \int_0^1 f(x)v(x)\,dx.$$

There are several possibilities how to implement Dirichlet boundary conditions. In our example we have homogeneous boundary conditions and in this case the simplest way is to leave out the outermost hat functions, since their coefficients have to be zero anyway. In the case of inhomogeneous Dirichlet boundary conditions the problem is transformed to a homogeneous one. The given function on the boundary is extended properly to the interior of the domain and added to the homogeneous solution.

The next step is to build the system matrix, where we consider the h-version using eight elements. On an arbitrary interior element, say $I_1 = [\frac{1}{8}, \frac{1}{4}]$, the following two parts of hat functions are nonzero:

$$\phi_1(x) = 2 - 8x, \qquad \text{and} \qquad \phi_2(x) = 8x - 1,$$

where the basis functions are labeled according to the defining nodes in the grid from left to right. The element stiffness matrix on I_1 is easily computed as

$$A^{(1)} = [a(\phi_i, \phi_j)]_{i,j=1}^2 = \begin{bmatrix} 8 & -8 \\ -8 & 8 \end{bmatrix}.$$

The local contributions are then added up in the corresponding positions in the global stiffness matrix. Note that in the diagonal the values of two neighbouring elements contribute. The system matrices for both versions for $p = 8$ on one element, respectively $p = 1$ on eight elements are given by

$$A_p = \begin{bmatrix} 4/3 & & & \\ & 4/5 & & \\ & & \ddots & \\ & & & 4/15 \end{bmatrix} \quad \text{and} \quad A_h = \begin{bmatrix} 16 & -8 & & & \\ -8 & 16 & -8 & & \\ & & \ddots & & \\ & & -8 & 16 & -8 \\ & & & -8 & 16 \end{bmatrix}.$$

Here we used the higher order basis functions defined in Section 2.4 in ascending order with respect to their polynomial degrees. The right hand side of the linear system for h- and p-version are given by

$$\underline{f}_h = \tfrac{1}{\pi^2}\,[1, 0, -1, 0, 1, 0, -1]^T\,,$$

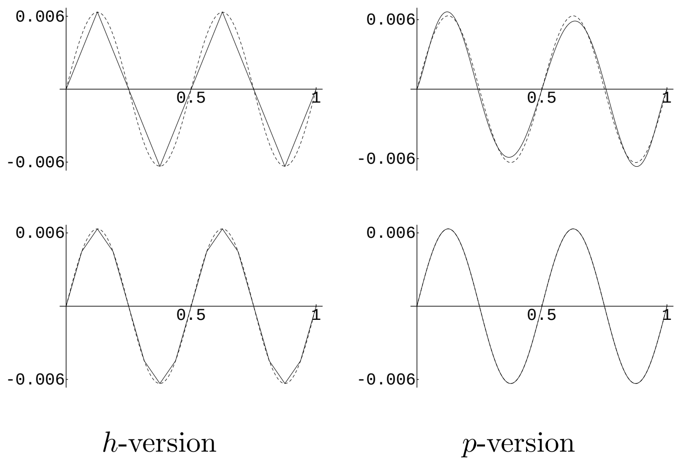


Figure 2.5: Exact (dashed) and approximate (solid) solutions to (2.20), $p = 8$, respectively 8 elements (topline), $p = 16$, respectively 16 elements (bottomline)

and

$$\underline{f}_p = \frac{1}{128\pi^7}\left[0, 48\pi^4, 0, 20\pi^2\left(-21+8\pi^2\right), 0, 21\left(495-240\pi^2+16\pi^4\right), 0\right]^T.$$

Also the entries of this vector are computed element-wise and then added up in the corresponding positions. The coefficients for the numerical solution are then computed as the solution to the linear system

$$A_{p/h}\,\underline{u}_{p/h} = \underline{f}_{p/h}.$$

Figure 2.5 compares the numerical solutions and the exact solution of (2.20). Observe that for $p = 16$ there is no visible difference between the solution of the p-method and $u_{ex}(x)$. This was to be expected, since in our example the solution is analytic.

The computations above were carried out with a finite element package we implemented in Mathematica and with which also figures 2.2 and 2.3 were created. This package is described in Appendix A.1, where also two dimensional examples are given. The principal procedure is the same for dimensions $d = 2, 3$, but the stiffness matrix in the p-version is no longer diagonal for higher dimensions.

Chapter 3

Algorithms for Special Functions

When talking about special functions we refer to orthogonal polynomials, especially Jacobi polynomials. These polynomials appear in the analysis and computations of finite element methods and we apply algorithms for special functions to tackle problems that arise in this context. For this purpose we use the RISC symbolic summation packages that are introduced in this chapter. These packages are implemented in the computer algebra system Mathematica. The software described below as well as a detailed description of the underlying algorithms are available at

`http://www.risc.uni-linz.ac.at/research/combinat/software/`

We do not intend to give a full characterization of either the algorithms or their theoretical background. In the first section holonomic sequences and the concept of generating functions are introduced. Section 3.2 deals with orthogonal polynomials and in Section 3.3 we present the RISC symbolic summation packages. For more details on special functions in general and orthogonal polynomials in particular, we refer to Zeilberger [87], the books of Szegö [79], Andrews, Askey and Roy [5] and Rainville [69] as well as Abramowitz and Stegun [1]. Concerning the Mathematica packages and literature on the implemented algorithms, references can be found in the respective subsections of Section 3.3. A survey article can be found in the chapter "Computer Algebra" [26] of the forthcoming Digital Library of Mathematical Functions (DLMF).

3.1 Holonomic Sequences

In this section we give a rough overview on the framework of objects to which the algorithms presented in Section 3.3 can be applied. We start by introducing the ring of formal power series. Let throughout this chapter K be a field containing $\mathbb{Q}$. Furthermore, let $K[x]$ denote the set of all polynomials in the indeterminate x with coefficients from the field K.

Definition 3.1. *Let z be an indeterminate and let K be a field containing $\mathbb{Q}$. Then we define the ring of formal power series*

$$K[\![z]\!] := \{(c_0, c_1, c_2, \dots) \mid c_n \in K\}.$$

It is common notation to write elements of $K[\![z]\!]$ in power series form:

$$(c_0, c_1, c_2, \dots) = \sum_{n=0}^{\infty} c_n z^n.$$

Addition (subtraction) for elements a, $b \in K[\![z]\!]$ is defined as

$$\sum_{n=0}^{\infty} a_n z^n \pm \sum_{n=0}^{\infty} b_n z^n := \sum_{n=0}^{\infty} (a_n \pm b_n) z^n.$$

Multiplication is defined via the Cauchy product

$$\sum_{n=0}^{\infty} a_n z^n \sum_{n=0}^{\infty} b_n z^n := \sum_{n=0}^{\infty} c_n z^n,$$

where the c_n are given by

$$c_n = \sum_{k=0}^{n} a_k b_{n-k}, \qquad n \geq 0.$$

We refer to an element $f(z) = \sum_{n=0}^{\infty} c_n z^n$ of $K[\![z]\!]$ as the generating function of the sequence $c = (c_0, c_1, c_2, \ldots)$. These elements are viewed rather as formal objects than functions in the classical sense, e.g., convergence is not an issue. Let us stress again that both, the power series $f(z)$ and the sequence c, describe the same objects in $K[\![z]\!]$. An important subclass in $K[\![z]\!]$ are holonomic power series.

Definition 3.2. *A formal power series $f(z) = \sum_{n=0}^{\infty} c_n z^n \in K[\![z]\!]$ is called holonomic iff there exist polynomials $q_0, \ldots, q_m \in K[z]$, not all zero, and $q \in K[z]$ such that*

$$(q_m(z)D^m + \ldots + q_0(z))f(z) = q(z), \tag{3.1}$$

where D^m denotes the mth differential operator. The coefficient sequence $(c_n)_{n=0}^{\infty}$ of a holonomic power series is called a holonomic sequence.

An alternative characterization of holonomic sequences that gives a simple criterion to check for holonomicity is stated in the following lemma.

Lemma 3.3. *A formal power series $f(z) = \sum_{n=0}^{\infty} c_n z^n \in K[\![z]\!]$ is holonomic iff there exist polynomials $q_0, \ldots, q_m \in K[z]$, not all zero, and $q \in K[z]$ such that*

$$q_m(n)c_{n+m} + \ldots + q_0(n)c_n = q(n), \tag{3.2}$$

for all $n \geq 0$.

Example 3.4. *A well known example for a holonomic function is the geometric series $f(z) = \sum_{n\geq 0} z^n$ satisfying the equation $(1-z)f(z) = 1$, $f(0) = 1$. On the sequence level we have $c_{n+1} - c_n = 0$ with $c_0 = 1$.*

Another basic example is the exponential function $f(z) = \sum_{n\geq 0} \frac{z^n}{n!}$ satisfying the differential equation $f'(z) - f(z) = 0$ with $f(0) = 1$. The coefficient sequence $c_n = \frac{1}{n!}$ fulfills the recurrence relation $(n+1)c_{n+1} - c_n = 0$, $c_0 = 1$.

Holonomic power series satisfy certain closure properties that can be formulated either on the power series or on the sequence level. In the next theorem some of these closure properties are stated. Since we do not work with generating functions but only on the sequence level they are formulated in terms of sequences.

Theorem 3.5. *Let $(a_n)_{n\geq 0}, (b_n)_{n\geq 0}$ be holonomic sequences. If c_n is defined as*

- *sum, i.e., $c_n = a_n + b_n$,*
- *Cauchy product, i.e., $c_n = \sum_{k=0}^{n} a_k b_{n-k}$,*
- *Hadamard product, i.e., $c_n = a_n b_n$,*
- *integer shift, i.e., $c_n = a_{n+h}$, $h \in \mathbb{N}$,*
- *partial sum, i.e., $c_n = \sum_{j=0}^{n} a_j$,*
- *forward difference, i.e., $c_n = \Delta_n a_n = a_{n+1} - a_n$,*

then $(c_n)_{n\geq 0}$ is again a holonomic sequence.

In the last theorem we used shift and difference operators. We denote the forward *shift operator* in a variable v by S_v. The forward *difference operator* in n is denoted by Δ_n and can be written as $\Delta_n = S_n - I$, where I is the identity operator.

Let us next introduce a special instance of holonomic sequences, namely hypergeometric sequences.

Definition 3.6. *A sequence $(a_n)_{n\geq 0}$ with elements in K is called hypergeometric over K, if there exist polynomials $p, q \in K[z]$ such that the linear relation*

$$p(n)a_{n+1} + q(n)a_n = 0$$

is satisfied for all $n \geq 0$. A bivariate sequence $(a(n,k))_{n,k\geq 0}$ is called hypergeometric if it is hypergeometric in both variables n and k.

A hypergeometric term $a(n,k)$ is called proper hypergeometric over K if the polynomials p_i, $q_i \in K[z_1, z_2]$ satisfying

$$p_1(n,k)a(n+1,k) + p_0(n,k)a(n,k) = 0, \qquad q_1(n,k)a(n,k+1) + q_0(n,k)a(n,k) = 0,$$

split into integer linear factors of the form $\alpha n + \beta k + \gamma$, α, $\beta \in \mathbb{Z}$, $\gamma \in K$.

The generating function for an important subset of hypergeometric sequences is the hypergeometric ${}_2F_1$-function

$${}_2F_1\left(\begin{matrix} a & b \\ c & \end{matrix}; z\right) = \sum_{k\geq 0} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!}, \tag{3.3}$$

where $(a)_k = a(a+1)\cdot \ldots \cdot (a+k-1)$, $k \geq 1$, and $(a)_0 = 1$, denotes the *Pochhammer symbol* or *rising factorial*. We also use the notation ${}_2F_1(a,b,c;z)$ or $F(a,b,c;z)$. Let a_n be the summand of the hypergeometric function. Then we have

$$\frac{a_{n+1}}{a_n} = \frac{(n+a)(n+b)}{(n+c)(n+1)} z,$$

i.e., a_n is hypergeometric according to Definition 3.6. By the ratio test it hence follows that the series in (3.3) is absolutely convergent for $|z| < 1$, independent of the choice of a, b and

c, as long as c is not zero or a negative integer. The hypergeometric function satisfies Euler's hypergeometric differential equation

$$z(1-z)\frac{d^2y}{dz^2} + [c - (a+b+1+)z]\frac{dy}{dz} - ab\,y = 0. \tag{3.4}$$

It is often referred to as "the" hypergeometric function since it appears in the definition of many classical objects, such as

$$\log(1+x) = x\,{}_2F_1\left(\begin{matrix} 1 \quad 1 \\ 2 \end{matrix}; -x\right).$$

Gauss defined two hypergeometric ${}_2F_1$ functions to be *contiguous* if they have the same power-series variable, if two of the parameters a, b, c are pairwise equal, and if the third pair differs by 1. It is common notation to write $F(a\pm)$ for ${}_2F_1(a \pm 1, b, c; z)$ and analogously for the other parameters. Gauss [41] showed that a hypergeometric function and any two others contiguous to it are linearly related. As Rainville in [69], page 50, observes: "The proof is one of remarkable directness; we prove that the relations exist by obtaining them." We note, however, that, once the identities have been found, the proof follows by simple coefficient comparison, as we illustrate below. An example for a contiguous relation is

$$(b-a)F + aF(a+) - bF(b+) = 0.$$

For the hypergeometric ${}_2F_1$ function there are nine different such relations, if the symmetry in a and b is taken into account. These relations can be iterated and linear relations involving hypergeometric functions with parameters differing by integers are referred to as contiguous relations. By simply shifting the summation index and using $(\lambda)_k = \lambda(\lambda+1)_{k-1}$, one finds for the derivative of the hypergeometric function with respect to z that

$$\frac{dF}{dz}(z) = \frac{ab}{c} F\left(\begin{matrix} a+1 \quad b+1 \\ c+1 \end{matrix}; z\right), \tag{3.5}$$

and

$$z\frac{dF}{dz} = (c-1)(F(c-) - F). \tag{3.6}$$

Let $\langle z^n\rangle F(z)$ denote the coefficient of z^n of $F(z)$. With (3.5) we have for the nth coefficient of the left hand side of (3.6) that

$$\langle z^n\rangle \left(z\frac{dF}{dz}\right) = \frac{ab}{c}\frac{(a+1)_{n-1}(b+1)_{n-1}}{(c+1)_{n-1}(n-1)!} = \frac{(a)_n(b)_n}{(c)_n(n-1)!}.$$

The nth coefficient of the right hand side of (3.6) is given by

$$\langle z^n\rangle\big((c-1)(F(c-) - F)\big) = (c-1)\frac{(a)_n(b)_n}{n!}\left(\frac{1}{(c-1)_n} - \frac{1}{(c)_n}\right) = \frac{(a)_n(b)_n}{(c)_n(n-1)!},$$

which proves (3.6). Later in Section 4.2 we show how to generate this relation using symbolic summation algorithms. The notion "hypergeometric function" can be generalized to p and q other than $(p, q) = (2, 1)$.

Definition 3.7. *Let $p, q \in \mathbb{N}$ and $a_i,\ b_i \in K$. Then the formal power series*

$$ {}_pF_q\left(\begin{matrix} a_1 & \dots & a_p \\ b_1 & \dots & b_q \end{matrix} ; z\right) = \sum_{n\geq 0} \frac{(a_1)_n \cdot \dots \cdot (a_p)_n}{(b_1)_n \cdot \dots \cdot (b_q)_n} \frac{z^n}{n!} $$

is called the hypergeometric ${}_pF_q$ function.

For generalized hypergeometric functions ${}_pF_q$ with $p \leq q+1$, there are $(2p+q)$ linearly independent relations, see [69, Ch.5]. The proof proceeds by classifying the types of contiguous relations and, as in the ${}_2F_1$-case, stating the relations for each class. For more details on hypergeometric functions and contiguous relations see [69, 5].

Observe that, if K is algebraically closed, then the sequence $(c_n)_{n\geq 0}$ is hypergeometric if and only if its generating function is of the form

$$ \sum_{n\geq 0} c_n z^n = c_0\, {}_pF_q\left(\begin{matrix} a_1 & \dots & a_p \\ b_1 & \dots & b_q \end{matrix} ; z\right). $$

3.2 Orthogonal Polynomials

The main cross-over point of the high order finite element method and symbolic summation methods in our work are orthogonal polynomials defining finite element shape functions. This section is a brief introduction to some, mostly well known, basic facts on orthogonal polynomials.

Definition 3.8. *Let $w(x)$ be a nonnegative function on the interval $[a, b]$ (which may be infinite) and assume that moments of all orders exist, i.e., $\int_a^b x^n\, w(x)dx < \infty$ for $n \geq 0$.*

A sequence of polynomials $(p_n(x))_{n=0}^{\infty}$ with real coefficients and with $\deg(p_n(x)) = n$, is called orthogonal with respect to the weight function $w(x)$ if

$$ \int_a^b p_i(x)p_j(x)\, w(x)dx = h_i\delta_{i,j}, \qquad h_i \in \mathbb{R}^+, $$

where $\delta_{i,j}$ denotes the Kronecker delta.

There exist several ways to represent orthogonal polynomials. One possible description is using the coefficients of their three term recurrence. According to the next theorem [5, Theorem 5.2.2] such a representation always exists.

Theorem 3.9. *Any sequence of real orthogonal polynomials $(p_n(x))_{n\geq 0}$ satisfies a three term recurrence with real coefficients $a_n,\ b_n,\ c_n$, with $b_n \neq 0$, of the form*

$$ p_{n+1}(x) = (a_n + b_n x)p_n(x) + c_n p_{n-1}(x), \qquad n \geq 1. \tag{3.7} $$

Furthermore we have, with h_n as in Definition 3.8, that $c_n = \frac{b_n}{b_{n-1}} \frac{h_n}{h_{n-1}}$.

Often the three term recurrence (3.7) can be extended to hold for $n \geq 0$ with $p_{-1}(x) \equiv 0$. Whenever we state relations between orthogonal polynomials involving degrees $n = -1$ it is to be understood in this sense. A prominent family of orthogonal polynomials often used in the definition of high order shape functions are Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$, $\alpha, \beta > -1$ and are treated in the next chapter. Legendre polynomials that were used in the definition of high order shape functions in Section 2.4 also belong to the family of Jacobi polynomials. More precisely, we have, $P_n(x) = P_n^{(0,0)}(x)$.

For $1 \le p < \infty$, let $L_w^p(a,b)$ denote the weighted L^p-space with respect to the weight function $w(x)$, i.e., the class of functions f such that

$$\|f\|_{L_w^p(a,b)}^p = \int_a^b |f|^p w(x)\, dx < \infty.$$

In the following we assume that $[a,b]$ is a finite interval. Let $(p_n(x))_{n\ge 0}$ be a sequence of real orthogonal polynomials on $[a,b]$ with associated weight function $w(x)$. Then the best polynomial approximation for functions $f \in L_w^2(a,b)$ in the weighted L_w^2-norm is given by the polynomial

$$q(x) = \sum_{j=0}^{n} \alpha_j p_j(x), \qquad \text{where} \qquad \alpha_j = \frac{1}{h_j} \int_a^b f(y)\, p_j(y)\, w(y)\, dy, \tag{3.8}$$

with h_j as in Definition 3.8, i.e., the norm $\|f - q\|_{L_w^2(a,b)}$ is minimal when the coefficients α_j are defined as above. Moreover

$$\sum_{j=0}^{n} \alpha_j^2 \le \|f\|_{L_w^2(a,b)}^2. \tag{3.9}$$

The sequence of partial sums $s_n = \sum_{j=0}^{n} \alpha_j^2$ is increasing and bounded. Hence (3.9) also holds in the limit $n \to \infty$. The inequality

$$\sum_{j=0}^{\infty} \alpha_j^2 \le \|f\|_{L_w^2(a,b)}^2.$$

is called Bessel's inequality. Since every function $f \in L_w^2(a,b)$ can be approximated arbitrarily well by a continuous function $g \in C(a,b)$ in the $L_w^2(a,b)$-norm, together with Bessel's inequality and Weierstrass' approximation theorem one obtains the following formula, see [5, Theorem 5.7.4].

Theorem 3.10. *Let $[a,b]$ be a finite interval, $f \in L_w^2(a,b)$ and $(p_n(x))_{n\ge 0}$ be a real orthogonal polynomial sequence in $L_w^2(a,b)$. Then with α_j as in (3.8) we have Parseval's formula:*

$$\sum_{j=0}^{\infty} \alpha_j^2 = \int_a^b f(x)^2\, w(x)\, dx.$$

As a corollary to this theorem we have that $f(x) = \sum_{j\ge 0} \alpha_j p_j(x)$ in $L_w^2(a,b)$ in the sense that the partial sums $q_n(x) = \sum_{j=0}^{n} \alpha_j p_j(x)$ converge to $f(x)$ in the $L_w^2(a,b)$-norm, i.e.,

$$\|q_n - f\|_{L_w^2(a,b)} \to 0 \qquad \text{as} \qquad n \to \infty.$$

More informations and proofs of the stated results can be found e.g. in [5, Ch. 5.7] or [79, Ch. 3]. If $f(x)$ is a polynomial of degree at most m, then we have

$$f(x) = \sum_{j=0}^{m} \alpha_j p_j(x) = \int_a^b \left(\sum_{j=0}^{m} \frac{1}{h_j} p_j(x) p_j(y) \right) f(y)\, w(y)\, dy. \tag{3.10}$$

The expression in brackets defines the sequence of kernel polynomials $(k_n(x,y))_{n\geq 0}$ associated to $(p_n(x))_{n\geq 0}$,

$$k_n(x,y) = \sum_{j=0}^{n} \frac{1}{h_j} p_j(x) p_j(y). \tag{3.11}$$

Identity (3.10) is called the *reproducing property* of kernel polynomials.

From the three term recurrence (3.7) one obtains the following important result, called the Christoffel-Darboux formula [5, Theorem 5.2.4].

Theorem 3.11. *Let $(p_n(x))_{n\geq 0}$ be a family of real orthogonal polynomials and b_n, h_n as in Definition 3.8. Then we have for the associated sequence of kernel polynomials $k_n(x,y)$ that*

$$k_n(x,y) = \frac{1}{b_n h_n} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y}, \tag{3.12}$$

for all $n \geq 0$.

Let us mention two more consequences of the three term recurrence for orthogonal polynomials. The derivatives of some orthogonal polynomials are again orthogonal polynomials. This holds for instance for certain limiting cases of Jacobi polynomials such as Hermite or Laguerre polynomials, see [47, 56]. For Jacobi polynomials derivatives are again Jacobi polynomials with parameters α, β raised by one, i.e.,

$$\frac{d}{dx} P_n^{(\alpha,\beta)}(x) = \frac{n+\alpha+\beta+1}{2} P_{n-1}^{(\alpha+1,\beta+1)}(x), \qquad n \geq 0.$$

Let $(p_n(x))_{n\geq 0}$ be a given sequence of orthogonal polynomials. Then differentiating the three term recurrence (3.7) we can represent $p_n(x)$ in terms of $p_n'(x)$

$$b_n p_n(x) = p_{n+1}'(x) - a_n p_n'(x) - b_n x\, p_n'(x) - c_n p_{n-1}'(x).$$

If the derivatives are again orthogonal polynomials, their three term recurrence can be used to replace $xp_n'(x)$ to obtain a relation between $p_n(x)$ and shifts of $p_n'(x)$, where the coefficients are free of x. Namely, let $(\alpha_n, \beta_n, \gamma_n)$ be the recurrence coefficients for the sequence $(p_n'(x))_{n\geq 1}$, i.e.,

$$p_{n+2}'(x) = (\alpha_n x + \beta_n) p_{n+1}'(x) + \gamma_n p_n'(x).$$

Then we arrive at the following x-free relation between $p_n(x)$ and shifts of its derivative,

$$p_n(x) = A_n p_{n+1}'(x) + B_n p_n'(x) + C_n p_{n-1}'(x),$$

where

$$A_n = \frac{\alpha_{n-1} - b_n}{\alpha_{n-1} b_n},\ B_n = -\frac{a_n \alpha_{n-1} - b_n \beta_{n-1}}{\alpha_{n-1} b_n},\ C_n = -\frac{\alpha_{n-1} c_n - b_n \gamma_{n-1}}{\alpha_{n-1} b_n}.$$

Since the connection coefficients (A_n, B_n, C_n) are not depending on x, this relation can be differentiated with respect to x to obtain relations between $p_n^{(k)}(x)$ and $p_n^{(k+1)}(x)$ for arbitrary k. Their connection coefficients are simply shifts of the coefficients given above.

Another immediate consequence of the three term recurrence is the existence of a bivariate x-free recurrence for the products of orthogonal polynomials. When assembling a system matrix for high order finite elements one has to compute several integrals over products of certain orthogonal polynomials. Having an x-free recurrence relation for the product of these polynomials means having a recurrence relation for the integral over the product. This also holds, when the polynomials are multiplied by some coefficient function not depending on the polynomial degrees.

Lemma 3.12. *Let* $(p_i(x))_{i\geq 0}$ *and* $(q_j(x))_{j\geq 0}$ *be two sequences of orthogonal polynomials with three term recurrences*

$$p_{i+1}(x) = (a_i + b_i x)p_i(x) + c_i p_{i-1}(x), \qquad i \geq 1,$$

and

$$q_{j+1}(x) = (\alpha_j + \beta_j x)q_j(x) + \gamma_j q_{j-1}(x), \qquad j \geq 1.$$

Then the recurrence for $M_{i,j} = \int p_i(x)q_j(x)\,dx$ *is given by*

$$M_{i,j} = \frac{a_i\beta_{j+1} - b_i\alpha_{j+1}}{\beta_{j+1}} M_{i-1,j} + \frac{b_i}{\beta_{j+1}} M_{i-1,j+1} - \frac{\gamma_{j+1}}{\beta_{j+1}} b_i M_{i-1,j-1} + c_i M_{i-2,j}, \tag{3.13}$$

for $i \geq 1,\ j \geq 0$.

Proof. First employ the recurrence relation for p_i to obtain

$$p_{i+1}(x)q_{j+1}(x) = a_i p_i(x)q_{j+1}(x) + b_i p_i(x)\,x\,q_{j+1}(x) + c_i p_{i-1}(x)q_{j+1}(x).$$

Using the recurrence relation for q_j for rewriting $x\,q_{j+1}(x)$ completes the proof. □

Besides using the recurrence coefficients as representation, Jacobi polynomials, as well as other families of orthogonal polynomials, can be characterized in terms of the hypergeometric function. A commonly used representation for Jacobi polynomials is

$$\begin{aligned} P_n^{(\alpha,\beta)}(x) =& \frac{(\alpha+1)_n}{n!}\ {}_2F_1\left(\begin{matrix} -n & n+\alpha+\beta+1 \\ \alpha+1 & \end{matrix}; \frac{1-x}{2}\right) \\ =& \frac{(\alpha+1)_n}{n!} \sum_{k\geq 0} \frac{(-n)_k(n+\alpha+\beta+1)_k}{(\alpha+1)_k k!} \left(\frac{1-x}{2}\right)^k. \end{aligned}$$

Note that the sum above is finite because of the factor $(-n)_k$ which vanishes for $k > n$, assuming $n \in \mathbb{N}$. Because of this natural bound there is no need to specify an upper summation bound. By different power series expansions further representations are obtained. In the next chapter examples are given. We use the hypergeometric sum representation as input when applying some of the RISC symbolic summation packages listed below.

3.3 Algorithms and RISC Symbolic Summation Packages

In this section we introduce the RISC symbolic summation packages implemented in Mathematica that we apply later to problems arising in the context of the high order finite element method. We focus on the functions that are used later in this thesis and do not give a full list of the available features. All these packages (and more) as well as a detailed description of the underlying algorithms are available for download at

`http://www.risc.uni-linz.ac.at/research/combinat/software/`

3.3.1 GeneratingFunctions

The Mathematica package GeneratingFunctions provides routines to manipulate holonomic objects, both on the power series and on the sequence level. It has been implemented by Christian Mallinger as part of his diploma thesis [59]. The Maple predecessor of GeneratingFunctions is the gfun package implemented by Salvy and Zimmermann [70]. The GeneratingFunctions package is loaded by typing:

In[1]:= **<< GeneratingFunctions.m**
GeneratingFunctions Package by Christian Mallinger – © RISC Linz – V 0.67 (03/13/03)

We mostly use Mallinger's guessing-routine GuessRE. This functions finds a homogeneous recurrence relation that is satisfied by elements from a given list. For instance, guessing from the first nine values, the three term recurrence for Legendre polynomials is obtained by

In[2]:= **GuessRE[Table[LegendreP[n, x], {n, 0, 8}], P[n]]**

Out[2]= $\{\{(1+n)P[n]-(3+2n)xP[1+n]+(2+n)P[2+n]==0, P[0]==1, P[1]==x\}, \mathrm{ogf}\}$

The input is a list of objects for which we want to know a linear relation. Optionally also the order of the recurrence and the degree of the coefficient polynomials in n can be specified. Their default bounds are 2 and 3, respectively.

Holonomic closure properties (Theorem 3.5) can be executed in GeneratingFunctions. The implemented operations include REPlus (Addition), RECauchy (Cauchy-product) and REHadamard (Hadamard-product). We consider the following example, where we determine the Cauchy product of the sequence $(a_n = n)_{n\geq 0}$ and the constant sequence $(a_n = 1)_{n\geq 0}$.

In[3]:= **RECauchy[{$a[n+1]-a[n]==0, a[0]==1$},
{$a[n]-2a[1+n]+a[2+n]==0, a[0]==0, a[1]==1$}, $a[n]$]**

Out[3]= $\{a[n]-3a[1+n]+3a[2+n]-a[3+n]==0, a[0]==0, a[1]==1, a[2]==3\}$

Observe regarding the input form that both recurrence relations have to be given in the same variables. According to the definition of the Cauchy product, the result has to be the recurrence relation for $\sum_{k=0}^{n} k$. Solving the above recurrence using the Mathematica built-in command RSolve yields the expected result

In[4]:= **RSolve[%, $a[n], n$]**

Out[4]= $\{\{a[n] \to \frac{n+n^2}{2}\}\}$

Certainly guessing is not proving. There has to be done more for showing that the linear relation, satisfied by a finite number of initial values, holds for the sequence the user is interested in. The key point, however, in our applications is *discovering* new identities. Having found them, giving a proof usually requires only basic arithmetic that can often be done automatically using computer algebra. The other routines presented in this section are rigorous in the sense that their output (based on a correct input) is a full proof.

3.3.2 Gosper, Zeilberger

Gosper's algorithm [45] is designed to find antidifferences of a given hypergeometric term, if they exist, and "impossible" otherwise, i.e. for solving the *telescoping problem* specified by

$$\begin{aligned}\text{Input:}\quad & f(k) \text{ hypergeometric}\\ \text{Output:}\quad & g(k) \text{ hypergeometric: } f(k)=g(k+1)-g(k) \text{ OR "impossible"}\end{aligned}$$

If there exists such a hypergeometric antidifference $g(k)$, then summation over the telescoping identity yields a closed form for the sum

$$\sum_{k=0}^{n} f(k) = \sum_{k=0}^{n} g(k+1)-g(k) = g(n+1)-g(0).$$

Hence Gosper's algorithm even determines whether a hypergeometric term is indefinitely summable or not. Let us discuss next the problem of finding a linear recurrence for a definite sum over a proper hypergeometric term $f(n,k)$

$$S(n) = \sum_{k=-\infty}^{\infty} f(n,k).$$

Here we assume that for fixed n the summand has finite support in k, in the sense that

$$\lim_{k\to\pm\infty} f(n,k) = 0.$$

The task is now to find polynomials $c_0(n),\ldots,c_d(n)$, free of k and not all zero, and $g(n,k)$ satisfying

$$c_0(n)f(n,k)+\ldots+c_d(n)f(n+d,k) = g(n,k+1)-g(n,k). \tag{3.14}$$

The algorithm accomplishing this task is due to Zeilberger [86, 88]. The function $g(n,k)$ is a rational multiple of the summand $f(n,k)$. Hence $g(n,k)$ also has finite support. Thus, summing over the summand recurrence, one obtains a recurrence relation for the sum $S(n)$

$$c_0(n)S(n)+\ldots+c_d(n)S(n+d) = 0.$$

We are using Peter Paule's and Markus Schorn's Mathematica implementation of these algorithms [65]. Their package is loaded by typing

In[5]:= **<< zb.m**

Fast Zeilberger Package by Peter Paule, Markus Schorn, and Axel Riese – © RISC Linz – V 3.43 (03/31/04)

To give an example, we apply Zeilberger's algorithm for finding the three term recurrence for Jacobi polynomials, using the summand of the hypergeometric sum representation as input.

In[6]:= $\mathbf{Zb}\left[\frac{\mathbf{Pochhammer}[\alpha+1,n]}{n!}\frac{\mathbf{Pochhammer}[-n,k]\mathbf{Pochhammer}[n+\alpha+\beta+1,k]}{\mathbf{Pochhammer}[\alpha+1,k]k!}\left(\frac{1-x}{2}\right)^k,\right.$ $\{k,0,n\},n]$

Out[6]= If '$n'$is a natural number, then :

$\{-2(1+\alpha+n)(1+\beta+n)(4+\alpha+\beta+2n)\mathrm{SUM}[n]+(3+\alpha+\beta+2n)(\alpha^2-\beta^2+8x+6\alpha x+\alpha^2x+6\beta x+2\alpha\beta x+\beta^2x+12nx+4\alpha nx+4\beta nx+4n^2x)\mathrm{SUM}[1+n]-2(2+n)(2+\alpha+\beta+n)(2+\alpha+\beta+2n)\mathrm{SUM}[2+n]==0\}$

Typing “Prove[ ]” after Zeilberger’s algorithm returned a recurrence relation, the package delivers a computer proof including the rational function $r(n,k)$ with $g(n,k) = r(n,k)f(n,k)$. In our example we have

$$r(n,k) = \frac{2k(k+\alpha)(n+\alpha+1)(2n+\alpha+\beta+2)(2n+\alpha+\beta+3)(2n+\alpha+\beta+4)}{(k-n-2)(k-n-1)(n+\alpha+\beta+1)}.$$

3.3.3 MultiSum

MultiSum is a Mathematica package for discovering and proving holonomic hypergeometric multi-sum identities. It has been developed by Kurt Wegschaider [83]. In 1945 Mary Fasenmyer, also known as Sister Celine, showed in her PhD-thesis [37] that recurrence relations for hypergeometric single sums $\sum_k f(n,k)$ can be found algorithmically. MultiSum, based on WZ theory [85], uses an efficient generalization of Sister Celine’s technique to find a homogeneous polynomial recurrence relation for hypergeometric multiple sums.

The algorithm proceeds similarly to Zeilberger’s algorithm in the sense that first a recurrence relation for the summand is generated that is free of the summation variables. A recurrence for the given sum is then obtained by summing over both sides of this relation. The package is loaded by typing

In[7]:= **<< MultiSum.m**

MultiSum Package by Kurt Wegschaider and Axel Riese – © RISC Linz – V 1.60 (04/14/04)

The function symbols used by MultiSum for the summand and sum are F and SUM, respectively. To give a simple example we generate the product recurrence (3.13) for Legendre polynomials. As input we use the summands of the hypergeometric sum representation of Jacobi polynomials for $\alpha = \beta = 0$. Since we are interested in a recurrence relation that is free of x we specify this demand using the “FreeOf” option.

In[8]:= $$\mathbf{FindRecurrence}\Big[\frac{\mathbf{Pochhammer}[-n,k]\mathbf{Pochhammer}[n+1,k]}{(k!)^2}\left(\frac{1-x}{2}\right)^k \frac{\mathbf{Pochhammer}[-m,j]\mathbf{Pochhammer}[m+1,j]}{(j!)^2}\left(\frac{1-x}{2}\right)^j, \{m,n\}, \{j,k\}, \mathbf{FreeOf} \to \{x\}\Big][[1]]$$

Out[8]= $$-(m(1+2n)F[-1+m,n,-1+j,-1+k]) + (1+2m)nF[m,-1+n,-1+j,-1+k] + (1+2m)(1+n)F[m,1+n,-1+j,-1+k] - (1+m)(1+2n)F[1+m,n,-1+j,-1+k] == \mathrm{Delta}[j, m(1+2n)F[-1+m,n,-1+j,-1+k] - (1+2m)(1+2n)F[m,n,-1+j,-1+k] + (1+m)(1+2n)F[1+m,n,-1+j,-1+k]] + \mathrm{Delta}[k, -((1+2m)nF[m,-1+n,-1+j,-1+k]) + (1+2m)(1+2n)F[m,n,-1+j,-1+k] - (1+2m)(1+n)F[m,1+n,-1+j,-1+k]]$$

Observe that we specified the summand, the main variables and the summation variables. Additionally it is possible to bound the shifts of both, main and summation variables as well as to specify the maximal degree of the coefficient polynomials. In the output the right hand side is given in terms of forward differences of linear combinations of shifts of the summand. This identity as well as the initial values can easily be checked. The recurrence in the output Out[8] is also called certificate recurrence.

By summing over this certificate recurrence a recurrence for the sum is obtained. This can be done automatically using the SumCertificate command. Continuing our example the recurrence for the double sum $\mathrm{SUM}[n,m] = P_n(x)P_m(x)$ is obtained as follows:

In[9]:= **SumCertificate[%]**

Out[9]= $-((1+2m)n\text{SUM}[-1+n,m])+m(1+2n)\text{SUM}[n,-1+m]+(1+m)(1+2n)\text{SUM}[n,1+m] - (1+2m)(1+n)\text{SUM}[1+n,m]==0$

Note, however, that it is assumed that the summand has finite support and SumCertificate returns a homogeneous recurrence. Hence in certain situations human inspection is needed to pass from the certificate recurrence to the sum recurrence.

3.3.4 SumCracker

SumCracker is a package developed by Manuel Kauers [53] containing algorithmic procedures for treating sequences that are described via certain systems of difference equations (recurrence relations). It can be used for proving known (or conjectured) identities as well as for discovering new identities. The package is loaded by

In[10]:= **<< SumCracker.m**

SumCracker Package by Manuel Kauers – © RISC Linz – V 0.7 2007-02-04

One advantage of this package is that it allows to enter polynomials as symbolic expressions. But it does not deliver proofs that are easily readable by humans such as the algorithms described in the last two sections. The routines that we mostly use are Crack, ZeroSequenceQ, LinearRecurrence and ProveInequality.

Crack rewrites a given expression either in terms of objects contained in the given expression or tries to rewrite it into other objects specified using the Into option. Let us consider an example where we want to find a reformulation of the derivative of Legendre polynomials.

In[11]:= **Crack[D[LegendreP[n + 1, x], x]]//Simplify**

Out[11]= $\frac{(1+n)(-\text{LegendreP}[n,x]+x\text{LegendreP}[1+n,x])}{-1+x^2}$

We have mentioned that derivatives of Jacobi polynomials are again Jacobi polynomials. Hence we try to find a reformulation of $P'_{n+1}(x)$ in terms of Jacobi polynomials $P_n^{(1,1)}(x)$ automatically.

In[12]:= **Crack[D[LegendreP[n + 1, x], x], Into → {n, JacobiP[n, 1, 1, x]}]//Simplify**

Out[12]= $\frac{(2+n)\text{JacobiP}[n,1,1,x]}{2}$

ZeroSequenceQ is a zero-equivalence tester. It can be used to check whether a given relation is true, for instance a result found by guessing. To give an example we verify the Legendre three term recurrence found by GuessRE.

In[13]:= **ZeroSequenceQ[(n + 1)LegendreP[n, x] − (2n + 3)xLegendreP[n + 1, x] + (n + 2)LegendreP[n + 2, x], Variable → n, From → 0]**

Out[13]= True

LinearRecurrence is a special purpose command searching for a (possibly inhomogeneous) linear recurrence for the given sequence. The expressions that may appear in the coefficients are chosen automatically unless specified using the “In” option. The default function symbol of the resulting recurrence is SUM. This setting can be overridden using the “Head” option. As an example we consider generating the Legendre three term recurrence.

In[14]:= **LinearRecurrence[LegendreP[n, x], Head → L, In → {n}]**

Out[14]= $L[2+n] == \frac{(-1-n)L[n]}{2+n} + \frac{(3x+2nx)L[1+n]}{2+n}$

Since SumCracker's internal representation of Legendre polynomials is exactly the three term recurrence, this is a void example. Later, however, this command is used to determine linear relations for nontrivial input.

ProveInequality is an automatic inequality prover constructing an inductive proof using cylindrical algebraic decomposition, see Collins [29] and Gerhold and Kauers [43]. As an example consider the Christoffel-Darboux formula (3.12) for kernel polynomials for the limit $y \to x$,

$$\sum_{j=0}^{n} \frac{1}{h_j} p_j(x)^2 = \lim_{y \to x} \frac{1}{b_n h_n} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y}.$$

It is easily calculated that the right hand side is

$$\frac{1}{b_n h_n}\big(p'_{n+1}(x)p_n(x) - p'_n(x)p_{n+1}(x)\big),$$

which is obviously positive since on the left hand side we are summing over squares with positive coefficients. We demonstrate ProveInequality on the special case of $p_n(x)$ being Legendre polynomials.

In[15]:= **ProveInequality[D[LegendreP[$n+1, x$], x]LegendreP[n, x]
− LegendreP[$n+1, x$]D[LegendreP[n, x], x] > 0, Using → {$-1 \le x \le 1$}, Variable → n]**

Out[15]= True

Note also that when a sum is given as input that within SumCracker the symbol SUM has to be used in order to avoid conflicts with the Mathematica summation command Sum. Furthermore note that SumCracker does not accept definite sums as input, where definite is meant, roughly speaking, in the sense that no summation bound occurs in the summand.

Chapter 4

Jacobi Polynomial Identities and Computer Algebra

For the construction of higher order finite element basis functions usually certain orthogonal polynomials are used. Often these polynomials are Jacobi polynomials or compositions of Jacobi polynomials. One reason therefor is that the weight function associated to Jacobi polynomials pops up naturally in transformations related to finite elements, as will be seen below. For proving the desired properties of these basis functions or operators involving Jacobi polynomials we apply the algorithms described in the last chapter either directly or by using relations that can again be generated automatically using these symbolic summation tools. In Section 3.2 general properties of orthogonal polynomials have been presented. Next, we discuss in particular Jacobi polynomials. For further information we refer to the books of Andrews, Askey and Roy [5], Szegö [79], Rainville [69] or Abramowitz and Stegun [1].

4.1 Jacobi and Integrated Jacobi Polynomials

For α, β real numbers and $-1 < x < 1$, the nth Jacobi polynomial $P_n^{(\alpha,\beta)}(x)$, $n \geq 0$, may be defined via the hypergeometric function as

$$P_n^{(\alpha,\beta)}(x) = \frac{(\alpha+1)_n}{n!}\,{}_2F_1\left(\begin{matrix} -n & n+\alpha+\beta+1 \\ \alpha+1 & \end{matrix};\frac{1-x}{2}\right). \tag{4.1}$$

Using this representation and the hypergeometric differential equation (3.4), immediately a differential equation satisfied by Jacobi polynomials is obtained:

$$(1-x^2)\,y''(x) + [\beta-\alpha-(\alpha+\beta+2)x]\,y'(x) + n(n+\alpha+\beta+1)\,y(x) = 0. \tag{4.2}$$

Jacobi polynomials are not symmetric in the parameters α and β, but they satisfy the relation

$$P_n^{(\alpha,\beta)}(x) = (-1)^n P_n^{(\beta,\alpha)}(-x). \tag{4.3}$$

From this identity and by definition (4.1) another hypergeometric sum representation of Jacobi polynomials follows, namely

$$P_n^{(\alpha,\beta)}(x) = (-1)^n\frac{(\beta+1)_n}{n!}\,{}_2F_1\left(\begin{matrix} -n & n+\alpha+\beta+1 \\ \beta+1 & \end{matrix};\frac{1+x}{2}\right). \tag{4.4}$$

The values of Jacobi polynomials at $x = \pm 1$, that can be read off the representations (4.1) and (4.4), respectively, are

$$P_n^{(\alpha,\beta)}(+1) = \binom{n+\alpha}{n} \qquad \text{and} \qquad P_n^{(\alpha,\beta)}(-1) = (-1)^n \binom{n+\beta}{n}.$$

Already in Section 3.2 we stated that derivatives of Jacobi polynomials are again Jacobi polynomials with shifted parameters, i.e.,

$$\frac{d}{dx} P_n^{(\alpha,\beta)}(x) = \frac{n+\alpha+\beta+1}{2} P_{n-1}^{(\alpha+1,\beta+1)}(x). \tag{4.5}$$

This identity follows from the formula of derivatives of the hypergeometric function (3.5) by substituting $a = -n$, $b = n+\alpha+\beta+1$, $c = \alpha+1$ and $z = \frac{1-x}{2}$. With the same substitution the contiguous relation (3.6) can be reinterpreted in terms of Jacobi polynomials to obtain

$$(1-x)\frac{d}{dx} P_n^{(\alpha,\beta)}(x) = \alpha P_n^{(\alpha,\beta)}(x) - (n+\alpha) P_n^{(\alpha-1,\beta+1)}(x). \tag{4.6}$$

Because of (4.5) this identity relates Jacobi polynomials with shifts in α and β. Further such identities can be generated automatically using MultiSum. Let

$$\text{summand} = \frac{(\alpha+1)_n}{n!} \frac{(-n)_k (n+\alpha+\beta+1)_k}{(\alpha+1)_k k!} z^k,$$

be the summand of the hypergeometric sum representation of Jacobi polynomials. Now execute the FindRecurrence command, bounding the shifts in α and n each by one.

```
In[16]:= FindRecurrence[summand, {n, α}, {1, 1}, k, 0]

Out[16]= {−((β + n)F[−1 + n, α, k]) − (α + β + 2n)F[n, −1 + α, k] + (α + β + n)F[n, α, k] == Delta[k, 0]}
```

Summing over this difference equation (using SumCertificate) and rewriting in terms of Jacobi polynomials yields

$$(2n+\alpha+\beta) P_n^{(\alpha-1,\beta)}(x) = (n+\alpha+\beta) P_n^{(\alpha,\beta)}(x) - (n+\beta) P_{n-1}^{(\alpha,\beta)}(x). \tag{4.7}$$

A mirrored version of (4.7) is obtained by plugging in relation (4.3):

$$(2n+\alpha+\beta) P_n^{(\alpha,\beta-1)}(x) = (n+\alpha+\beta) P_n^{(\alpha,\beta)}(x) + (n+\alpha) P_{n-1}^{(\alpha,\beta)}(x). \tag{4.8}$$

This identity can also be generated using MultiSum by using the summand of (4.4) and bounding the shifts in β and n each by one. For the identities stated so far no range for α and β was specified and these parameters were treated symbolically, although the definitions (4.1) and (4.4) are not valid for nonnegative integer α and β, respectively. But, more general and not using the hypergeometric function, we may write

$$P_n^{(\alpha,\beta)}(x) = \frac{1}{n!} \sum_{k \geq 0} \frac{(-n)_k}{k!} (n+\alpha+\beta+1)_k (\alpha+k+1)_{n-k} \left(\frac{1-x}{2}\right)^k. \tag{4.9}$$

From this form it is obvious that Jacobi polynomials can be defined for all values of the parameters α, β. Another well known representation of Jacobi polynomials is the *Rodrigues formula*. With

$$w_{\alpha,\beta}(x) = \left(\frac{1-x}{2}\right)^\alpha \left(\frac{1+x}{2}\right)^\beta, \tag{4.10}$$

Jacobi polynomials can be written as

$$w_{\alpha,\beta}(x)\,P_n^{(\alpha,\beta)}(x) = \frac{(-1)^n}{n!}\frac{d^n}{dx^n}w_{n+\alpha,n+\beta}(x). \tag{4.11}$$

Next, let $\alpha, \beta > -1$. Under this condition $w_{\alpha,\beta}(x)$ is an integrable, nonnegative function on $[-1, 1]$ for which the moments of all orders exist. Hence it can be viewed as a weight function for orthogonal polynomials and, using the Rodrigues formula, via integration by parts it follows that

$$\int_{-1}^{1} w_{\alpha,\beta}(x)P_i^{(\alpha,\beta)}(x)P_j^{(\alpha,\beta)}(x)\,dx = 0, \qquad i \neq j.$$

Summarizing, we have, for $\alpha, \beta > -1$, that $\big(P_n^{(\alpha,\beta)}(x)\big)_{n\geq 0}$ is a sequence of orthogonal polynomials on the interval $[-1, 1]$ with associated weight function $w_{\alpha,\beta}(x)$ satisfying the orthogonality relation

$$\int_{-1}^{1} P_i^{(\alpha,\beta)}(x)P_j^{(\alpha,\beta)}(x)w_{\alpha,\beta}(x)\,dx = h_i^{\alpha,\beta}\delta_{i,j}. \tag{4.12}$$

The squared weighted $L^2_{w_{\alpha,\beta}}$-norm is given by

$$h_i^{\alpha,\beta} = \frac{2}{2i+\alpha+\beta+1}\frac{\Gamma(i+\alpha+1)\Gamma(i+\beta+1)}{i!\,\Gamma(i+\alpha+\beta+1)}. \tag{4.13}$$

By Theorem 3.9 Jacobi polynomials satisfy a three term recurrence and in Section 3.3.2 it was proven, using Zeilberger's algorithm (see Out[6]), that for $n \geq 0$

$$P_{n+1}^{(\alpha,\beta)}(x) = (a_n + b_n x)P_n^{(\alpha,\beta)}(x) + c_n P_{n-1}^{(\alpha,\beta)}(x), \quad P_{-1}^{(\alpha,\beta)}(x) = 0,\ P_0^{(\alpha,\beta)}(x) = 1,$$

with coefficients

$$\begin{aligned} a_n &= \frac{(2n+\alpha+\beta+1)(\alpha^2-\beta^2)}{2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}, \qquad b_n = \frac{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}{2(n+1)(n+\alpha+\beta+1)}, \\ c_n &= -\frac{(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)}{(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}. \end{aligned} \tag{4.14}$$

The special case of $\alpha = \beta$ is called ultraspherical polynomial or *Gegenbauer polynomial*. The nth Gegenbauer polynomial $C_n^\lambda(x)$, where $\lambda > -\frac{1}{2}$, is commonly defined as

$$C_n^\lambda(x) = \frac{(2\lambda)_n}{(\lambda+\frac{1}{2})_n}P_n^{(\lambda-1/2,\lambda-1/2)}(x).$$

These polynomials are even or odd polynomials according as n is even or odd. One alternative hypergeometric sum representation of Gegenbauer polynomials, other than (4.1) or (4.4), is

$$C_n^\lambda(x) = \frac{(\lambda)_n}{n!}(2x)^n\,{}_2F_1\left(\begin{matrix} -n/2 & (1-n)/2 \\ 1-n-\lambda & \end{matrix};\frac{1}{x^2}\right). \tag{4.15}$$

Legendre polynomials $P_n(x) = P_n^{(0,0)}(x) = C_n^{1/2}(x)$ are also in this class. Recall that Legendre polynomials are orthogonal with respect to the L^2-inner product, i.e., with respect to the weight function $w_{0,0}(x) \equiv 1$. Two further well-known ultraspherical polynomials are the

Chebyshev polynomials of the first ($\alpha = \beta = -\frac{1}{2}$) and of the second kind ($\alpha = \beta = \frac{1}{2}$). They are commonly denoted by $T_n(x)$ and $U_n(x)$, respectively, and defined with the normalization

$$T_n(x) = \frac{P_n^{(-1/2,-1/2)}(x)}{P_n^{(-1/2,-1/2)}(1)}, \qquad \text{and} \qquad U_n(x) = (n+1)\frac{P_n^{(1/2,1/2)}(x)}{P_n^{(1/2,1/2)}(1)}. \tag{4.16}$$

Both satisfy the same recurrence relation with constant coefficients

$$p_n(x) - 2x\, p_{n+1}(x) + p_{n+2}(x) = 0, \qquad n \geq 0,$$

but with different starting values

$$T_0(x) = 1, \quad T_1(x) = x, \qquad \text{and} \qquad U_0(x) = 1, \quad U_1(x) = 2x.$$

Recall that in Section 3.2 a relation connecting orthogonal polynomials and their derivatives (if they are again orthogonal polynomials) with coefficients not depending on x was given. Jacobi polynomials are in this class and we have for $p_n(x) = P_n^{(\alpha,\beta)}(x)$

$$p_n(x) = A_n p'_{n+1}(x) + B_n p'_n(x) + C_n p'_{n-1}(x), \tag{4.17}$$

with coefficients

$$\begin{aligned} A_n &= \frac{2(n+\alpha+\beta+1)}{(2n+\alpha+\beta+2)(2n+\alpha+\beta+1)}, \\ B_n &= (\alpha-\beta)\frac{2}{(2n+\alpha+\beta)(2n+\alpha+\beta+2)}, \\ C_n &= -\frac{2(n+\beta)(n+\alpha)}{(n+\alpha+\beta)(2n+\alpha+\beta+1)(2n+\alpha+\beta)}. \end{aligned} \tag{4.18}$$

For ultraspherical polynomials relation (4.17) further simplifies since the coefficients B_n vanish. For Legendre polynomials in particular it reads as

$$P_n(x) = \frac{1}{2n+1}[P'_{n+1}(x) - P'_{n-1}(x)]. \tag{4.19}$$

Conversely, for the expansion of $P'_{n+1}(x)$ in terms of Legendre polynomials $\lfloor \frac{n}{2} \rfloor$ terms are needed. Let $\gamma_k = (P'_{n+1}, P_k)_0$ denote the coefficients in the expansion

$$P'_{n+1}(x) = \sum_{k=0}^{n} \frac{\gamma_k}{h_k^{0,0}} P_k(x),$$

where the squared L^2-norm $h_k^{0,0} = \frac{2}{2k+1}$ is as in (4.13). Clearly the coefficients γ_k vanish for k such that $n - k \equiv_2 1$. The remaining coefficients are computed by multiplying (4.19) by $P_k(x)$ and integrating over $[-1, 1]$. Since the inner product $(P'_{n-1}, P_n)_0$ vanishes, we have $\gamma_n = 2$. For $0 < k < \lfloor \frac{n}{2} \rfloor$ it follows that $\gamma_{n-2k} = 2$, because of $(P_n, P_{n-2k})_0 = 0$. Altogether this yields

$$P'_{n+1}(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (2n - 4k + 1) P_{n-2k}(x). \tag{4.20}$$

There exists a general formula transforming Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ into Jacobi polynomials $P_n^{(\gamma,\delta)}(x)$, see [5, Ch. 7.1]. The expansion (4.20) can be used to obtain

$$(P_i', P_j')_0 = \int_{-1}^{1} P_i'(x) P_j'(x)\, dx = \begin{cases} 0, & i-j \equiv_2 1, \\ l(l+1), & i-j \equiv_2 0, \quad \text{where } l = \min\{i,j\}. \end{cases} \tag{4.21}$$

This evaluation of the L^2-inner product of derivatives of Legendre polynomials is needed later in Section 8.1. Next we generalize integrated Legendre polynomials, as defined in (2.9), to integrated Jacobi polynomials.

Definition 4.1. *For $\alpha > -1$, $n \geq 1$ define the nth integrated Jacobi polynomial*

$$\hat{p}_n^\alpha(x) = \int_{-1}^{x} P_{n-1}^{(\alpha,0)}(s)\, ds.$$

Integrated Legendre polynomials, $\alpha = 0$, are also denoted by $L_n(x) = \hat{p}_n^0(x)$.

Observe that integrated Jacobi polynomials $\hat{p}_n^\alpha(x)$ vanish at $x = -1$. Using the sum representation (4.4) by simple term-wise integration integrated Jacobi polynomials can be related at once to classical Jacobi polynomials. We have for $n \geq 1$

$$\hat{p}_n^\alpha(x) = (-1)^{n-1}(1+x) \sum_{k\geq 0} \frac{(-n+1)_k (n+\alpha)_k}{(k+1)!k!} \left(\frac{1+x}{2}\right)^k. \tag{4.22}$$

Comparing coefficients of this sum representation to (4.4) yields for $n \geq 1$ the relation

$$\hat{p}_n^\alpha(x) = \frac{1+x}{n} P_{n-1}^{(\alpha-1,1)}(x). \tag{4.23}$$

Integrated Jacobi polynomials $\hat{p}_n^\alpha(x)$ can also be expressed in terms of $P_n^{(\alpha,0)}(x)$. Indeed, integrating (4.17) and using that $P_n^{(\alpha,0)}(-1) = (-1)^n$ yields for $n \geq 1$

$$\begin{aligned} \hat{p}_n^\alpha(x) = {} & \frac{2(n+\alpha)}{(2n+\alpha)(2n+\alpha-1)} P_n^{(\alpha,0)}(x) + \frac{2\alpha}{(2n+\alpha-2)(2n+\alpha)} P_{n-1}^{(\alpha,0)}(x) \\ & - \frac{2(n-1)}{(2n+\alpha-1)(2n+\alpha-2)} P_{n-2}^{(\alpha,0)}(x), \end{aligned}$$

since the contributions at the lower integration bound $x = -1$ cancel. For Legendre polynomials, i.e., $\alpha = \beta = 0$, the identity further simplifies since $B_n = 0$ and we obtain

$$L_{n+1}(x) = \frac{1}{2n+1}\left[P_{n+1}(x) - P_{n-1}(x)\right], \qquad n \geq 1. \tag{4.24}$$

From either of the representations, (4.23) or (4.22), a recurrence relation for integrated Jacobi polynomials can be derived. Applying Zeilberger's algorithm to (4.22) or using SumCracker's LinearRecurrence command on (4.23) yields for $n \geq 2$

$$\begin{aligned} \hat{p}_n^\alpha(x) = {} & \frac{2n+\alpha-3}{2n(n+\alpha-1)(2n+\alpha-4)}((\alpha-2)\alpha + (2n+\alpha-4)(2n+\alpha-2)x)\hat{p}_{n-1}^\alpha(x) \\ & - \frac{2(n-2)(n+\alpha-3)(2n+\alpha-2)}{2n(n+\alpha-1)(2n+\alpha-4)} \hat{p}_{n-2}^\alpha(x), \end{aligned} \tag{4.25}$$

with $\hat{p}_0^\alpha(x) = 1$, $\hat{p}_1^\alpha(x) = 1 + x$ for $\alpha \neq 0$. In the case $\alpha = 0$, i.e., for integrated Legendre polynomials, the recurrence relation has to be extended differently to the left. For $n \geq 2$ with $L_0(x) = -1$ and $L_1(x) = x$ we have

$$L_n(x) = \frac{2n-3}{n} x L_{n-1}(x) - \frac{n-3}{n} L_{n-2}(x).$$

Note, that for this choice $L_1(x) \neq \int_{-1}^x P_0(s)\, ds = 1 + x$. A widely known result of Favard [38] states the converse to Theorem 3.9: polynomials satisfying a three term recurrence of the form (3.7) are orthogonal polynomials. The proof, however, is not constructive and gives no information on either the interval or the associated weight function. For $n \geq 1$ and $\alpha \neq 0$, the rewriting (4.23) suggests orthogonality with respect to the singular weight function $w_{\alpha-1,-1}(x)$. Let $i, j \geq 1$, then we have

$$\int_{-1}^1 w_{\alpha-1,-1}(x) \hat{p}_i^\alpha(x) \hat{p}_j^\alpha(x)\, dx = \frac{4}{ij} \int_{-1}^1 w_{\alpha-1,1}(x) P_{i-1}^{(\alpha-1,1)}(x) P_{j-1}^{(\alpha-1,1)}(x)\, dx = \frac{4}{i^2} h_{i-1}^{\alpha-1,1} \delta_{i,j}.$$

Also identity (4.5) on derivatives of Jacobi polynomials supports viewing $\hat{p}_n^\alpha(x)$ as $P_n^{(\alpha-1,-1)}(x)$. Since $\frac{d}{dx}\hat{p}_n^\alpha(x) = P_{n-1}^{(\alpha,0)}(x)$ it follows that

$$\hat{p}_n^\alpha(x) = \frac{2}{n+\alpha-1} P_n^{(\alpha-1,-1)}(x). \tag{4.26}$$

From this identity with (4.1) or (4.9) another sum representation for integrated Jacobi polynomials can be obtained. Next consider integrated Legendre polynomials. Following the reasoning above, we interprete them as Jacobi polynomials $P_n^{(-1,-1)}(x)$. Integrated Legendre polynomials vanish at $x = \pm 1$ and thus the factor $(1 - x^2)$ can be pulled out. It is reasonable to assume that integrated Legendre polynomials are multiples of Jacobi polynomials and in fact, we have

$$L_n(x) = -\frac{1-x^2}{2(n-1)} P_{n-2}^{(1,1)}(x), \qquad n \geq 2. \tag{4.27}$$

This relation can be generated using SumCracker's Crack command, where we define integrated Legendre polynomials via identity (4.19):

```
In[17]:= Crack[JacobiP[n − 2, 1, 1, x], Into → {n, L[n]},
           Where → {L[n] == 1/(2n − 1)(LegendreP[n, x] − LegendreP[n − 2, x])}]//Simplify
```

Out[17]= $\frac{2(n-1)L[n]}{x^2-1}$

The two representations, (4.27) or (4.19), can also be used to generate the three term recurrence for integrated Legendre polynomials using the LinearRecurrence command. For $n \geq 2$, integrated Legendre polynomials span the space of polynomials on $[-1, 1]$ vanishing at $x = \pm 1$. In Section 2.4 it was already stated that they are orthogonal with respect to the H^1-inner product. Using (4.27) one obtains orthogonality for $n \geq 2$ with respect to the singular weight function $w_{-1,-1}(x)$:

$$\begin{aligned}\int_{-1}^1 w_{-1,-1}(x) L_i(x) L_j(x)\, dx &= \frac{4}{(i-1)(j-1)} \int_{-1}^1 w_{1,1}(x) P_{i-2}^{(1,1)}(x) P_{j-2}^{(1,1)}(x)\, dx \\ &= \frac{4}{(i-1)^2} h_{i-2}^{1,1} \delta_{i,j} = \frac{8}{(i-1)i(2i-1)} \delta_{i,j}.\end{aligned}$$

Summarizing, Jacobi polynomials $P_n^{(\alpha,-1)}(x)$ for $\alpha \geq -1$ are to be understood as integrated Jacobi polynomials as defined above. Observe that with this interpretation the identities (4.7) and (4.8) hold for integrated Jacobi polynomials.

4.2 Generating Mixed Difference-Differential Relations

Most of the identities stated in the previous section could be generated automatically using the algorithms presented in Section 3.3, where often it was possible to choose one's favorite tool. The relations collected in the last section emerged separately in various applications described below as answers to problems such as finding a recurrence relation for cheap evaluation of polynomials or finding a suitable rewriting or simplification of certain expressions. One example for such a relation is (4.6) that was originally needed especially formulated for integrated Jacobi polynomials, i.e.,

$$(\alpha-1)\hat{p}_n^\alpha(x) = (1-x)P_{n-1}^{(\alpha,0)}(x) + 2P_n^{(\alpha-2,0)}(x).$$

We obtained this identity, generally for Jacobi polynomials, from the contiguous relation (3.6) that in turn can be generated automatically using MultiSum. Let

$$\text{summand} = \frac{(a)_k(b)_k}{(c)_k k!} z^k$$

denote the summand of the hypergeometric function. First, use the FindStructureSet command to obtain a structure set of feasible shifts in a, b, c and k:

In[18]:= **FindStructureSet[summand, {a, b, c}, {1, 1, 2}, {k}, {1}, WZ → True]**

This yields amongst others the set $S = \{\{0,0,0,1\},\{1,1,1,0\},\{1,1,2,0\}\}$. For this specific structure set, FindRecurrence and subsequently applying SumCertificate yield

In[19]:= **FindRecurrence[summand, {a, b, c}, {k}, S, WZ → True]**

Out[19]= $\{(-2+c)(-1+c)F[-1+a,-1+b,-2+c,-1+k] - (-2+c)(-1+c)F[-1+a,-1+b,-1+c,-1+k] - (-1+a)(-1+b)zF[a,b,c,-1+k] == \mathrm{Delta}[k, -((-2+c)(-1+c)F[-1+a,-1+b,-2+c,-1+k]) + (-2+c)(-1+c)F[-1+a,-1+b,-1+c,-1+k]]\}$

In[20]:= **SumCertificate[%]**

Out[20]= $\{(-2+c)(-1+c)\mathrm{SUM}[-1+a,-1+b,-2+c] - (-2+c)(-1+c)\mathrm{SUM}[-1+a,-1+b,-1+c] - (-1+a)(-1+b)z\mathrm{SUM}[a,b,c] == 0\}$

This is an example, where human inspection is in place for the passing from Out[19] to Out[20]. The argument of the delta operator in Out[19] can be simplified to

$$\frac{(a)_{k-1}(b)_{k-1}}{(c-1)_k(k-2)!} z^{k-1},$$

which obviously vanishes for $k = 0$ and, for $|z| < 1$, also the limit $k \to \infty$ tends to zero. Thus we have verified the homogeneous recurrence in Out[20], which is just (3.6). In general, contiguous relations, also between non-terminating series, can be found automatically using Wilf-Zeilberger theory, see [64].

For proving (3.6) using MultiSum the additional knowledge (3.5) on derivatives of the hypergeometric function was needed. But it is also possible to derive mixed difference-differential relations automatically as we have seen in Out[11] where Crack was used to rewrite derivatives of Legendre polynomials in terms of Legendre polynomials. Another algorithm suited to find such mixed relations is an extension of Zeilberger's algorithm by Frédéric Chyzak and Bruno Salvy [27]. This algorithm is part of Chyzak's Maple package Mgfun. The underlying idea is to translate the relations into annihilating operators which can be represented in so-called Ore-algebras. In short, these are non-commutative operator algebras containing, e.g., shifts and derivatives. As part of his forthcoming thesis [55] Christoph Koutschan implements this algorithm in Mathematica with a user-friendly interface including further extensions. To illustrate how this algorithm proceeds we consider again the derivation of (4.6), this time with Koutschan's package. As input for the program we use the summand of the hypergeometric sum representation (4.1). An annihilator for Jacobi polynomials in terms of shifts in n, α, β and derivation with respect to x is generated with Takayama's algorithm [81, 80] and is given by

$$\begin{aligned}&(-2(n+1)(n+\alpha+\beta+1)S_n-(1-x^2)(2n+\alpha+\beta+2)D_x\\&+(n+\alpha+\beta+1)(2nx+\alpha x+\beta x+2x+\alpha-\beta),\\&(-\alpha-\beta-n-1)S_\beta-(1-x)D_x+(\alpha+\beta+n+1),\\&(\alpha+\beta+n+1)S_\alpha-(1+x)D_x-(n+\alpha+\beta+1),\\&(1-x^2)D_x^2+(-\alpha x-\beta x-2x-\alpha+\beta)D_x+n(n+\alpha+\beta+1).\end{aligned}$$

By means of Gröbner basis elimination in this non-commutative setting further identities for Jacobi polynomials can be obtained. For instance equation (4.6) can be viewed as an answer to the question: Find a relation between Jacobi polynomials free of β and shifts in n. Eliminating these two variables β and S_n yields, amongst other relations, the sought identity in operator form:

$$(1-x)D_xS_\alpha+(n+\alpha+1)S_\beta-(\alpha+1)S_\alpha.$$

Further examples for applications of Chyzak's algorithm can be found in [24].

Chapter 5

Hypergeometric Summation Algorithms for High Order Finite Elements

Next we describe the construction of simplicial edge and vertex based finite element basis functions, where we concentrate on providing recursion formulas that allow a simple implementation for fast basis function evaluation. The performance of iterative solution methods depends on the condition number of the system matrix, which itself depends on the chosen basis functions. Hence, the goal is to design basis functions minimizing the condition number and which can be computed efficiently. In [75] the construction of basis functions such that the blocks consisting of unknowns connected with one vertex, edge, face or cell of the mesh, are nearly orthogonal among each other is presented. For these basis functions the application of cheap block-Jacobi preconditioners is efficient; for the numerical analysis see [75]. In the joint paper with Becirovic, Paule, Riese, Schneider, and Schöberl [10] by means of symbolic summation algorithms recurrence relations for these edge and vertex based shape functions have been derived by Carsten Schneider using his Sigma package [72]. Below, alternative ways to generate these recurrence relations are given using other symbolic summation algorithms. For more details on Schneider's proofs see [10].

5.1 Edge Based Basis Functions

High order edge based basis functions form a basis of $P_0^p(E)$ on the associated edge and vanish on all faces in the finite element mesh not containing E. Degree-preserving extension procedures for given polynomials $p(x) \in P_0^p(E)$ to triangular edge based basis functions were introduced in Babuška et al. [8], and later simplified and extended to 3D by Muñoz-Sola [62]. Recently Demkowicz, Gopalakrishnan and Schöberl [33, 34, 32] gave a concise construction of polynomial extension operators for tetrahedral p-fem for the spaces $H^1(\Omega)$, $H(\mathrm{curl}, \Omega)$ and $H(\mathrm{div}, \Omega)$.

We consider the reference triangle $\hat{T} = \{(-1,-1), (1,-1), (0,1)\}$ and the lower edge E_0 connecting $(-1,-1)$ and $(1,-1)$ as associated edge. The lifting of a given polynomial $p(x)$

defined on E_0 according to Babuška et al. [8] is defined via the averaging procedure

$$(\mathcal{R}_1 p)(x,y) = \frac{1}{1+y}\int_{x-\frac{1+y}{2}}^{x+\frac{1+y}{2}} p(s)\,ds.$$

Note that $(\mathcal{R}_1 p)(x,y)$ is again polynomial in x and y and that its restriction to E_0 is again $p(x)$. This is easy to be seen via the substitution $s = x + t\frac{1+y}{2}$ yielding

$$(\mathcal{R}_1 p)(x,y) = \frac{1}{2}\int_{-1}^{1} p(x + \frac{1+y}{2}t)\,dt.$$

This extension is bounded as operator from $H^{1/2}(E_0) \to H^1(\hat{T})$. The modification by Muñoz-Sola preserving zero boundary values on the edges E_1 and E_2 is

$$(\mathcal{R}p)(x,y) = (1+2x-y)(1-2x-y)(\mathcal{R}_1 \frac{p}{1-x^2})(x,y). \tag{5.1}$$

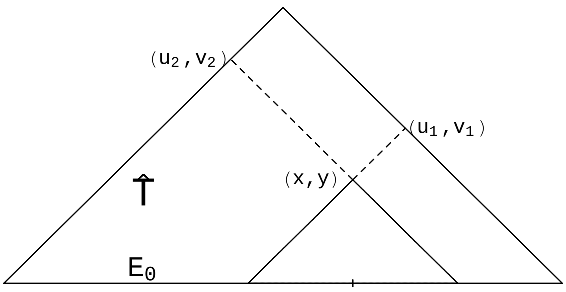
This lifting operator maps polynomials $p \in P_0^p(E_0)$ into the space of polynomials of total degree p on $\hat{T}$ that vanish on $\partial\hat{T}\backslash E_0$. When restricted to the associated edge $\mathcal{R}p(x,y)$ equals $p(x)$. For the well-definedness of this operator we need to introduce another trace space that has not yet been defined in Section 2.1. Let Ω be a domain and Γ be a subset of $\partial\Omega$ with non vanishing $(d-1)$-dimensional measure. The extensions of functions $u \in H_0^{1/2}(\Gamma)$ by zero do not in general belong to $H^{1/2}(\partial\Omega)$. Therefor we define the space

$$H_{00}^{1/2}(\Gamma) = \{u \in H_0^{1/2}(\Gamma) \mid \mathcal{E}u \in H^{1/2}(\partial\Omega)\},$$

where $\mathcal{E}u$ is the extension by zero of u to $\partial\Omega$. This space coincides with the interpolation space $[H_0^1(\Gamma), L^2(\Gamma)]_{1/2}$. It can be shown that $\mathcal{R}$ is bounded as operator from $H_{00}^{1/2}(E_0)$ to $H^1(\hat{T})$, see [8, 62]. The extension operator we are introducing now is defined using a different modification than (5.1) and we consider in particular extending integrated Legendre polynomials. On E_0 integrated Legendre polynomials $L_n(x)$, $2 \le n \le p$, form a basis for $P_0^p(E_0)$. In the first step define

$$\phi_n^{(1)}(x,y) = (\mathcal{R}_1 L_n)(x,y), \qquad n \ge 2.$$

But, although the polynomials L_n, $n \ge 2$, vanish at the boundary of E_0, the extension does not vanish at the upper two edges E_1 and E_2. This can be fixed, e.g., by linear interpolation between the lower and upper two edges.



The coordinates of the interpolation points (u_1,v_1) and (u_2,v_2) are given by

$$u_1(x,y) = \frac{1+2x-y}{4}, \qquad v_1(x,y) = \frac{1-2x+y}{2}$$
$$u_2(x,y) = \frac{-1+2x+y}{4}, \qquad v_2(x,y) = \frac{1+2x+y}{2}.$$

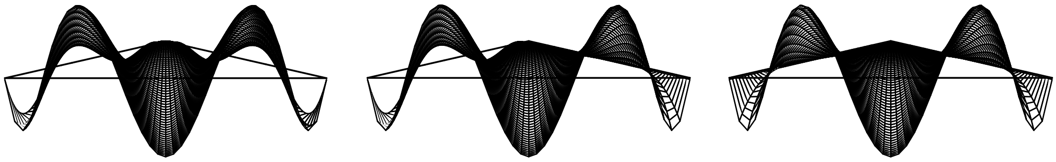


Figure 5.1: $\phi_n^{(1)}(x)$, $\phi_n^{(2)}(x)$ and $\phi_n(x)$ for $n = 6$

With this notation we homogenize $\phi_n^{(1)}$ on the upper right edge as follows

$$\phi_n^{(2)}(x,y) = \phi_n^{(1)}(x,y) - \frac{1+y}{1+v_1}\phi_n^{(1)}(u_1,v_1), \qquad n \geq 2. \tag{5.2}$$

The final extension is then given by

$$\phi_n(x,y) = \phi_n^{(2)}(x,y) - \frac{1+y}{1+v_2}\phi_n^{(2)}(u_2,v_2), \qquad n \geq 2. \tag{5.3}$$

Figure 5.1 illustrates the transition from $\phi_n^{(1)}(x,y)$ to $\phi_n(x,y)$. The resulting extension operator preserves the polynomial order for $n \geq 2$. Furthermore the total extension operator is bounded as operator from $H_{00}^{1/2}(E_0)$ to $H^1(\hat{T})$ and the ϕ_n satisfy homogeneous boundary conditions on the upper two edges E_1 and E_2. Thus they can be chosen as edge based basis functions on the reference triangle $\hat{T}$. By means of symbolic summation methods we have derived the following relation allowing for an efficient computation of the ϕ_n.

Theorem 5.1. *The functions ϕ_n defined in* (5.3) *satisfy the* 5*-term recurrence relation*

$$\phi_n = a_n(v_2 - v_1)\,\phi_{n-1} + (b_n + c_n v_1 v_2)\phi_{n-2} + d_n(v_2 - v_1)\phi_{n-3} + e_n\phi_{n-4}, \tag{5.4}$$

for all $n \geq 5$ with coefficients

$$a_n = \frac{2n-3}{n+1}, \qquad b_n = -\frac{(2n-5)(2n^2-10n+3)}{n(n+1)(2n-7)}, \qquad c_n = \frac{(2n-5)(2n-3)}{n(n+1)},$$
$$d_n = \frac{(n-5)(2n-3)}{n(n+1)}, \qquad e_n = -\frac{(n-6)(n-5)(2n-3)}{n(n+1)(2n-7)}, \qquad .$$

The initial values are given by

$$\begin{aligned}
\phi_1(x,y) &= \frac{x(-2x+y-1)(2x+y-1)}{(-2x+y+3)(2x+y+3)},\\
\phi_2(x,y) &= -\frac{1}{8}(-2x+y-1)(2x+y-1),\\
\phi_3(x,y) &= -\frac{1}{8}x(-2x+y-1)(2x+y-1),\\
\phi_4(x,y) &= \frac{1}{128}(2x-y+1)(2x+y-1)\left(20x^2+3y^2+2y-5\right).
\end{aligned}$$

The initial value $\phi_1(x,y)$ is a rational function, but although this function does not serve as an edge based shape function, it is still the correct extension of the recurrence relation (5.4) to the left. The denominator of $\phi_1(x,y)$ does not vanish on the reference triangle $\hat{T}$.

Remark 5.2. *The coefficients a_n to e_n are computed once and for all and are stored in tables. The evaluation of p basis functions ϕ_n takes just $11p + \mathcal{O}(1)$ floating point operations.*

Proof of Theorem 5.1. For finding and proving recurrence relation (5.4) we utilize holonomic closure properties. Before doing so, we rewrite ϕ_n in terms of Jacobi polynomials $P_n^{(2,2)}(x)$. First recall identity (4.24) connecting integrated Legendre and Legendre polynomials

$$L_n(x) = \frac{1}{2n-1}[P_n(x) - P_{n-2}(x)], \qquad n \geq 2.$$

With this relation $\phi_n^{(1)}(x, y)$ can be expressed in terms of integrated Legendre polynomials. For better readability we change variables to v_1 and v_2 ($x = \frac{v_2 - v_1}{2}$ and $y = v_1 + v_2 - 1$). Thus we obtain

$$\phi_n^{(1)}(x,y) = \frac{1}{v_1+v_2}\frac{1}{2n-1}[L_{n+1}(v_2) - L_{n-1}(v_2) - L_{n+1}(-v_1) + L_{n-1}(-v_1)], \qquad n \geq 2.$$

The right hand side can be simplified further by first expressing integrated Legendre polynomials in terms of Jacobi polynomials $P_n^{(1,1)}(x)$ using (4.27), i.e.,

$$L_{n+1}(x) - L_{n-1}(x) = (x^2-1)\left[\frac{1}{2n}P_{n-1}^{(1,1)}(x) - \frac{1}{2n-4}P_{n-3}^{(1,1)}(x)\right], \qquad n \geq 3$$

By (4.17), Jacobi polynomials can be rewritten in terms of their derivatives which, by (4.5), are again Jacobi polynomials with shifted parameters. Hence the difference $L_{n+1}(x) - L_{n-1}(x)$ can be expressed in terms of $P_n^{(2,2)}(x)$. In fact, applying SumCracker yields

In[21]:= $\mathbf{Crack}[\frac{1}{2n+6}\mathbf{JacobiP}[n+2,1,1,x] - \frac{1}{2n+2}\mathbf{JacobiP}[n,1,1,x],$
$\quad \mathbf{Into} \rightarrow \{n, \mathbf{JacobiP}[n,2,2,x]\}]//\mathbf{Factor}$

Out[21]= $\frac{(5+2n)(-1+x)(1+x)\mathrm{JacobiP}[n,2,2,x]}{4(1+n)(2+n)}$

With this identity we have completed the rewriting of $\phi_n^{(1)}(x, y)$:

$$\phi_n^{(1)}(x,y) = \frac{1}{(2n-4)(2n-2)}\frac{1}{v_1+v_2}\left((1-v_2^2)^2\, P_{n-3}^{(2,2)}(v_2) - (1-v_1^2)^2\, P_{n-3}^{(2,2)}(-v_1)\right), \quad n \geq 3.$$

After performing the correction steps (5.2) and (5.3) we obtain the following representation for ϕ_n:

$$\begin{aligned}\phi_n(x,y) = &\frac{(1-v_1)(1-v_2)}{(2n-4)(2n-2)(v_1+v_2)}[(1-v_2)(1+v_1+2v_2)P_{n-3}^{(2,2)}(v_2) \\ &- (1-v_1)(1+2v_1+v_2)P_{n-3}^{(2,2)}(-v_1)].\end{aligned}$$

With this rewriting of ϕ_n as sum of two (scaled) Jacobi polynomials its recurrence relation can easily be obtained by invoking the REPlus command of Mallinger's GeneratingFunctions package. We use the hypergeometric sum representation (4.1) for Jacobi polynomials and apply Zeilberger's algorithm to obtain the input recurrences for REPlus.

In[22]:= $\mathbf{rec1} = \mathbf{Zb}[\frac{\mathbf{Pochhammer}[-n,k]\mathbf{Pochhammer}[n+5,k]}{k!(k+2)!}\left(\frac{\mathbf{1}+\mathbf{v1}[x,y]}{\mathbf{2}}\right)^k, \{k,0,n\}, n];$

If 'n' is a natural number, then:

Out[22]= $\{2(1+n)\mathrm{SUM}[n]-(7+2n)(-1+2x-y)\mathrm{SUM}[1+n]+2(6+n)\mathrm{SUM}[2+n]==0\}$

and analogously for the second term. The proof is completed by adding these two relations to obtain the recurrence for ϕ_n.

In[23]:= **REPlus[rec1, rec2, SUM[*n*]]**

Out[23]= $4(n+1)(n+2)(2n+11)\mathrm{SUM}(n)-8(n+2)(2n+7)(2n+11)x\mathrm{SUM}(n+1)+(2n+9)\cdot\big(16x^2n^2-4y^2n^2-8yn^2+4n^2+144x^2n-36y^2n-72yn+36n+308x^2-77y^2-154y+47\big)\cdot\mathrm{SUM}(n+2)-8(n+7)(2n+7)(2n+11)x\mathrm{SUM}(n+3)+4(n+7)(n+8)(2n+7)\mathrm{SUM}(n+4)==0$ □

From the first part of the proof of Theorem 5.1 it is obvious that also for the Muñoz-Sola lifting $(\mathcal{R}p)(x,y)$ a recurrence relation can be derived automatically, if $p(x)/(1-x^2)$ is from a family of orthogonal polynomials whose derivatives are again orthogonal polynomials. This is for instance the case if we consider $p(x)=(1-x^2)P_n^{(\alpha,0)}(x)$ for some $\alpha>-1$. With the notation of Definition 4.1, the primary extension operator can be written in terms of integrated Jacobi polynomials

$$\psi_n^\alpha(x,y):=(\mathcal{R}_1P_n^{(\alpha,0)})(x,y)=\frac{1}{v_1+v_2}(\hat{p}_{n+1}^\alpha(v_2)-\hat{p}_{n+1}^\alpha(-v_1)).$$

The recurrence relation for $\psi_n^\alpha(x,y)$ is, as in the last proof, computed by adding the recurrence relations for integrated Jacobi polynomials using REPlus. The resulting recurrence relation is also of order five. Here, we only state the recurrence relation for the special case $\alpha=0$, i.e., $p(x)=(1-x^2)P_n(x)$, since the recurrence coefficients for general α are rather big. The coefficients for the recurrence

$$a_n\psi_n^0(x,y)+b_n\psi_{n+1}^0(x,y)+c_n\psi_{n+2}^0(x,y)+d_n\psi_{n+3}^0(x,y)+e_n\psi_{n+4}^0(x,y)=0,$$

are given by

$$a_n=(n-1)n(2n+5),\qquad b_n=n(2n+1)(2n+5)(v_1-v_2),$$
$$c_n=-(2n+3)(-1+5v_1v_2+2n(3+n)(-1+2v_1v_2)),$$
$$d_n=(n+3)(2n+1)(2n+5)(v_1-v_2),\qquad e_n=(n+3)(n+4)(2n+1).$$

As concluding example we consider the Muñoz-Sola extension of integrated Legendre polynomials. Because of (4.27), integrated Legendre polynomials are $p(x)=(1-x^2)P_n^{(1,1)}(x)$, up to normalization. Analogously to the rewriting in the proof of Theorem 5.1 one obtains

$$\psi_n(x,y):=\big(\mathcal{R}_1P_n^{(1,1)}\big)(x,y)=\frac{1}{v_1+v_2}\frac{2}{n+2}\big(P_{n+1}(v_2)-P_{n+1}(-v_1)\big),$$

and, applying again REPlus, a five term recurrence for $\psi_n(x,y)$ is obtained:

$$a_n\psi_n(x,y)+b_n\psi_{n+1}(x,y)+c_n\psi_{n+2}(x,y)+d_n\psi_{n+3}(x,y)+e_n\psi_{n+4}(x,y)=0,$$

with coefficients:

$$a_n=4(n+2)(n+3)^2(2n+9),\qquad b_n=4(n+3)(n+4)(2n+5)(2n+9)(v_1-v_2),$$
$$c_n=2(n+5)(2n+7)\big(1+(2n+5)(2n+9)(1-2v_1v_2)^2\big),$$
$$d_n=4(n+4)(n+6)(2n+5)(2n+9)(v_1-v_2),\qquad e_n=4(n+4)(n+5)(n+7)(2n+5).$$

The tetrahedral H^1-extension operator investigated in [33] is constructed following the same principles as the triangular edge extension operator described above. First an extension operator similar to Muñoz-Sola is applied followed by a series of correction steps assuring the defining properties of the resulting basis functions. This procedure would also allow for invoking holonomic closure properties, if the extension is applied to certain orthogonal polynomials. Since this extension is defined for three variables and more correction steps are involved, it is to be expected that the resulting recurrence might be rather big.

5.2 Low Energy Vertex Based Basis Functions

The condition number of high order system matrices depends on the maximal degree p of the basis functions. One possibility to reduce this dependence is the use of low energy vertex shape functions. The authors in [75] propose to use vertex based basis functions that are constant along the level sets of the standard hat functions and minimizing the H^1-norm along this class of functions.

Let in two dimensions the reference triangle $\hat{T}$ be defined as in the last section and let $V = (0, 1)$ be the associated vertex. The vertex based basis functions minimizing the H^1-seminorm are obtained as solutions to the minimization problem

$$\min_{\phi^V(V)=1,\ \phi^V(x,-1)=0} \|\nabla\phi^V\|_{L^2(\hat{T})}.$$

With the ansatz $\phi^V(x,y) = \phi^V(y) \in P^p([-1,1])$, i.e., ϕ^V constant in x-direction, the H^1-seminorm of ϕ^V has the form

$$\|\nabla\phi^V\|^2_{L^2(\hat{T})} = \int_{-1}^{1}\int_{\frac{y-1}{2}}^{\frac{1-y}{2}} [\frac{d}{dy}\phi^V(y)]^2\,dx\,dy = \int_{-1}^{1}(1-y)[\frac{d}{dy}\phi^V(y)]^2\,dy.$$

Analogously in three dimensions, using the reference tetrahedron $\hat{T}$ given by the vertices $(-1,-1,-1)$, $(1,-1,-1)$, $(0,1,-1)$ and $(0,0,1)$ with associated vertex $V = (0,0,1)$, the corresponding ansatz leads to

$$\|\nabla\phi^V\|^2_{L^2(\hat{T})} = \int_{-1}^{1}\frac{(1-z)^2}{2}[\frac{d}{dz}\phi^V(z)]^2\,dz.$$

Thus for dimension $d = 2$, 3 the constrained minimization problems

$$\min_{\substack{v\in P^p(I)\\ v(-1)=0,v(1)=1}} \int_{-1}^{1}(1-s)^{d-1}(v'(s))^2\,ds \tag{5.5}$$

have to be solved. These are strictly convex minimization problems on finite dimensional spaces. Thus there exist unique solutions which we call $v_p^{(d)}(x)$, $d = 2, 3$. To find these solutions we expand $v(x)$ in terms of Jacobi polynomials. The factor $(1-s)^{d-1}$ in (5.5) can be interpreted as the weight function $w_{d-1,0}(x)$ associated to Jacobi polynomials. Hence, it is natural to expand $v(x)$ in terms of integrated Jacobi polynomials $\hat{p}_j^{d-1}(x)$, i.e.,

$$v_p^{(d)}(s) = \sum_{j=1}^{p}\nu_j\,\hat{p}_j^{d-1}(s), \qquad d = 2, 3, \tag{5.6}$$

since with this ansatz the derivative of $v_p^{(d)}(s)$ is just

$$\frac{d}{ds}v_p^{(d)}(s) = \sum_{j=1}^{p} \nu_j \, P_{j-1}^{(d-1,0)}(s).$$

The constrained minimization problem (5.5) can be translated to an algebraic one. Let A be the real $p \times p$ matrix with entries

$$A_{i,j} = \int_{-1}^{1} (1-s)^{d-1} \frac{d}{ds}\hat{p}_i^{d-1}(s) \frac{d}{ds}\hat{p}_j^{d-1}(s) \, ds, \qquad 1 \le i, j \le p,$$

and define the vectors $\underline{b}^0 = (b_1^0, \ldots, b_p^0)$ and $\underline{b}^1 = (b_1^1, \ldots, b_p^1)$ with entries

$$b_i^0 = \hat{p}_i^{d-1}(-1) = 0 \qquad \text{and} \qquad b_i^1 = \hat{p}_i^{d-1}(1) = 2\frac{(d-1)_{i-1}}{i!},$$

where $(a)_n$ denotes again the Pochhammer symbol. With these definitions (5.5) can be transformed into the algebraic constrained minimization problem

$$\min_{\substack{\underline{\nu} \in \mathbb{R}^p \\ \underline{b}^0 \cdot \underline{\nu} = 0, \underline{b}^1 \cdot \underline{\nu} = 1}} \quad \underline{\nu}^T A \, \underline{\nu}.$$

Because of the ansatz (5.6) for $v_p^{(d)}(s)$ using integrated Jacobi polynomials, the matrix entries of A are easily computed using the orthogonality relation (4.12) for Jacobi polynomials. Thus A is a diagonal matrix with entries

$$A_{i,i} = \int_{-1}^{1} (s-1)^{d-1} \big(P_{i-1}^{(d-1,0)}(s)\big)^2 \, ds = \frac{2^d}{2i + d - 2}.$$

Since integrated Jacobi polynomials vanish at $x = -1$ the condition $v(-1) = 0$ is already included in our ansatz. The vector $\underline{b}^1$ has entries $b_i^1 = \frac{2}{i}$ in the two dimensional and $b_i^1 = 2$ in the three dimensional case. With these specific values for A and $\underline{b}^1$ solving the constrained minimization problem for the coefficients $\underline{\nu} = (\nu_1 \ldots, \nu_p)$ yields

$$v_p^{(2)}(x) = \left(\sum_{k=1}^{p} \frac{1}{k} \right)^{-1} \sum_{j=1}^{p} \hat{p}_j^1(x),$$

in the two dimensional case $d = 2$, and

$$v_p^{(3)}(x) = \frac{1}{p(p+2)} \sum_{j=1}^{p} (2j+1) \, \hat{p}_j^2(x),$$

for the three dimensional case $d = 3$. For a fast computation it is sufficient to have good recursive descriptions for

$$u_p^{(2)}(x) := \left(\sum_{k=1}^{p} \frac{1}{k} \right) v_p^{(2)}(x) \qquad \text{and} \qquad u_p^{(3)}(x) := p(p+2) v_p^{(3)}(x),$$

that are stated in the next theorem.

Theorem 5.3. *The functions* $u_p^{(2)}$ *and* $u_p^{(3)}$ *satisfy the recurrence relations*

$$\begin{aligned}
u_1^{(2)}(x) &= \frac{x+1}{2},\\
u_2^{(2)}(x) &= \frac{3}{8}(x+1)^2,\\
u_3^{(2)}(x) &= \frac{1}{24}(x+1)\left(10x^2+5x+7\right),\\
u_p^{(2)}(x) &= \frac{(2p-1)(-1+3x+p^2(1+2x)-p(1+5x))}{p^2(2p-3)}u_{p-1}^{(2)}(x)\\
&\quad -\frac{(1+x-3p(1+x)+p^2(1+2x))}{p^2}u_{p-2}^{(2)}(x)+\frac{(p-2)^2(2p-1)}{p^2(2p-3)}u_{p-3}^{(2)}(x),
\end{aligned}$$

and

$$\begin{aligned}
u_1^{(3)}(x) &= \frac{3(1+x)}{2},\\
u_2^{(3)}(x) &= \frac{1}{2}(1+x)(3+5x),\\
u_p^{(3)}(x) &= \frac{(x(4p^2-1)-1)}{(p+1)(2p-1)}u_{p-1}^{(3)}(x)-\frac{(p-1)(2p+1)}{(p+1)(2p-1)}u_{p-2}^{(3)}(x)+\frac{(2p+1)(1+x)}{(p+1)},
\end{aligned}$$

respectively.

Proof. The recurrence relation for $u_p^{(2)}$ can be proven using holonomic closure properties following the same lines as in the example given in Section 3.3.1. With $c[j]=\hat{p}_{j+1}^1(x)$, i.e. $u_p^{(2)}=\sum_{j=0}^{p-1}c[j]$, only the Cauchy product of the integrated Jacobi polynomials and the constant sequence $c[j]=1$ has to be computed. The recurrence relation for $c[j]=\hat{p}_{j+1}^1(x)$, that can either be generated using Zeilberger's algorithm or read off (4.25), is given by

In[24]:= $\mathbf{rec1 = \{(j+3)^2(j+3)c[j+2]-(j+2)((2j+3)(2j+5)x-1)c[j+1]+(j+1)^2(2j+5)c[j]==0,}$
$\mathbf{c[0]=1+x, c[1]=\frac{1}{4}(1+x)(3x-1)\}}$

In order to obtain the recurrence for the sum we use the RECauchy command of Mallinger's GeneratingFunctions package:

In[25]:= **rec = RECauchy[rec1, {c[j + 1] == c[j], c[0] == 1}, c[j]]**

Out[25]= $\{-(j+3)(2j+7)(x+1)c[j](j+2)^2+(j+3)(2j+5)(x+1)(2xj^2+j^2+13xj+5j+21x+5)c[j+1]$
$-(j+3)(2j+7)(x+1)(2xj^2+j^2+11xj+7j+15x+11)c[j+2]+(j+3)(j+4)^2(2j+5)(x+1)c[j+3]=0,$
$c[0]=\frac{x+1}{2}, c[1]=\frac{3}{8}(x+1)^2, c[2]=\frac{1}{24}(x+1)(10x^2+5x+7)\}$

It is easily checked that this result is the recurrence relation for $c[j]=u_{j+1}^{(2)}$ for $j\geq 0$ as stated in the theorem.

For proving the three term recurrence for $u_p^{(3)}(x)$ we apply Kauers' SumCracker package, where we use the identity (4.26) to rewrite $\hat{p}_j^2(x)=\frac{2}{j+1}P_j^{(1,-1)}(x)$. With Crack a simple closed form for the sum $u_p^{(3)}(x)$ in terms of Legendre polynomials can be generated.

In[26]:= $\mathbf{Crack[SUM[\frac{2k+1}{k+1}JacobiP[k,1,-1,x],\{k,1,p\}],Into\rightarrow\{p,LegendreP[p,x]\}]}$

Out[26]= $\frac{-1-x+\text{LegendreP}[p,x]+\text{LegendreP}[p+1,x]}{-1+x}$

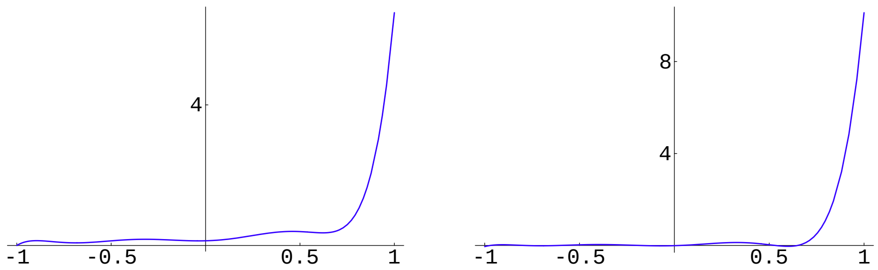


Figure 5.2: $\frac{d}{dx}v_p^{(2)}(x)$ and $\frac{d}{dx}v_p^{(3)}(x)$ for $p = 8$

At this point, we could again invoke holonomic closure properties. But using REPlus to determine a recurrence for $P_p(x) + P_{p+1}(x)$ results in a recurrence relation of order four. Hence instead we use the LinearRecurrence command of SumCracker to obtain

In[27]:= $\mathbf{LinearRecurrence}[\frac{\mathbf{-1 - x + LegendreP[p, x] + LegendreP[p+1, x]}}{\mathbf{-1 + x}}, \mathbf{In \to \{p\}}]$

Out[27]= $$\text{SUM}(p+2) == \frac{4xp^2 + 4p^2 + 16xp + 16p + 15x + 15}{2p^2 + 9p + 9} + \frac{(-2p^2 - 7p - 5)\,\text{SUM}(p)}{2p^2 + 9p + 9} + \frac{(4xp^2 + 16xp + 15x - 1)\,\text{SUM}(p+1)}{2p^2 + 9p + 9}$$

□

Figure 5.2 illustrates the steep descent of the derivatives of these low energy vertex shape functions. Although using REPlus yields a recurrence for the closed form in Out[26] that is too big, GeneratingFunctions can still be used to obtain the three term recurrence for $u_p^{(3)}(x)$. With GuessRE a homogeneous recurrence of order two for the sum of the Legendre polynomials can be found:

In[28]:= **GuessRE[Table[LegendreP[n, x] + LegendreP[n + 1, x]//Factor, {n, 0, 11}], S[n]][[1]]**

Out[28]= $\{(2+n)(7+2n)S[n] + (1 - 35x - 24nx - 4n^2x)S[1+n] + (4+n)(5+2n)S[2+n] == 0,$
$S[0] == 1 + x, S[1] == \frac{1}{2}(x+1)(3x-1)\}$

Here $S[n] = P_{n+1}(x) + P_{n+2}(x)$ for $n \geq 0$. The correctness of this relation for all $n \geq 0$ can easily be verified plugging in the Legendre three term recurrence. Because of $u_{p+1}^{(3)}(x) = (1 + x - S[p])(1 - x)$ and since $S[p]$ satisfies the recurrence above, replacing $S[n]$ in Out[28] by $u_{n+1}^{(3)}(x)$ yields the desired inhomogeneous recurrence

$$\begin{aligned}&(2+n)(7+2n)u_{n+1}^{(3)}(x) + [1 - (2n+5)(2n+7)x]u_{n+2}^{(3)}(x) + (n+4)(2n+5)u_{n+2}^{(3)}(x)\\ =&(2n+5)(2n+7)(1+x).\end{aligned}$$

In [10] Schneider generated,i.e., proved, and used another closed form, namely

$$u_p^{(3)}(x) = \frac{1}{(p+1)(1-x)}\left[pP_p^{(1,-1)}(x) + (p+1)\big(x + 1 - P_{p+1}^{(1,-1)}(x)\big)\right].$$

This identity can also be found using Crack the same way as the rewriting in terms of Legendre polynomials was obtained by leaving out the "Into" option. Since the recurrence relation for the sum of the recurrences for the Jacobi polynomials $P_p^{(1,-1)}(x)$ found with REPlus is also of order four, Schneider followed another strategy to obtain a shorter recurrence. In the first step an additional slack parameter e in the summand was introduced

$$u_p(3)(x) = \sum_{j=1}^{p} \frac{(e-j+p)!}{(-j+p)!} \frac{2j+1}{j+1} P_j^{(1,-1)}(x).$$

Sending this parameter e to 0 the original summand is regained. Applying the GeneratingRecurrence command of his Sigma package yields a recurrence of order three. This relation can be simplified further to the recurrence given in Theorem 5.3 using the Sigma command ReduceRecurrence, for details see [10].

Chapter 6

Triangular and Tetrahedral Shape Functions Using Integrated Jacobi Polynomials

Next we propose bases for the high order finite element method using a triangular or tetrahedral mesh that lead to a sparse stiffness matrix in the case of a piecewise constant diffusion matrix and a polygonally bounded domain. More precisely, the number of nonzero matrix entries is of the same order as the number of unknowns. This sparsity of the system matrix has two direct applications, namely a fast evaluation of the matrix and the preconditioning of the block of cell based basis functions. The latter also works for an uniformly elliptic second order boundary value problem with arbitrary coefficients. For proving the nonzero pattern of the system matrix we explicitely determine the matrix entries with a program that we implemented in Mathematica. The underlying algorithm is described in Section 6.3 and numerical experiments showing the efficiency of the proposed basis functions are given in Section 6.4. The contents of this chapter are joint work with Sven Beuchler [16, 17].

6.1 Motivation

We study the following boundary value problem: Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ be a bounded domain and let $\mathcal{A}$ be a symmetric matrix that is uniformly positive in Ω. Find

$$u \in H^1_{\Gamma_1}(\Omega) = \{u \in H^1(\Omega) \mid u = 0 \text{ on } \Gamma_1\}, \qquad \Gamma_1 \cap \Gamma_2 = \emptyset, \quad \Gamma_1 \cup \Gamma_2 = \partial\Omega,$$

such that

$$a(u, v) = \int_\Omega (\nabla u)^T \mathcal{A} \nabla v \, dx = \int_\Omega f v \, dx + \int_{\Gamma_2} f_1 v \, dx \tag{6.1}$$

holds for all $v \in H^1_{\Gamma_1}(\Omega)$. We denote the global system matrix for this problem by $\mathcal{K}$. The advantage of discretizing by means of the hp-version compared to the pure h-version is that the solution converges faster to the exact solution with respect to the total number of unknowns N. However the choice of a basis Φ in which the element stiffness matrix $\mathcal{K}$ has $\mathcal{O}(N)$ nonzero matrix entries is a difficult question. In the one-dimensional case, e.g., for the differential equation $-u'' + u = f$, in order to get a sparse system matrix primitives of orthogonal polynomials can be used, see e.g. [48] or the example given in Section 2.6.

In the two and three dimensional case, however, the choice of a basis which is optimal with respect to the condition number and sparsity of $\mathcal{K}$ is not so clear. In [9] several bases have been investigated regarding their condition number. In the case of tensor product elements like quadrilaterals and hexahedrons, and a constant diffusion matrix $\mathcal{A}$, tensor products of integrated Legendre polynomials, such as defined in (2.12)-(2.14), can be taken, see [8, 48]. Then the stiffness matrix has $\mathcal{O}(N)$ nonzero matrix entries, see Figure A.1, and $\mathcal{K}$ can be computed in $\mathcal{O}(N)$ operations. However, in the case of a general quadrilateral (hexahedral in 3D) element with nonparallel opposite edges (faces), most of the orthogonality relations of the reference element case disappear and $\mathcal{K}$ has, in general, $\mathcal{O}(p^6)$ matrix entries. Using a quadrature rule, the cost in order to obtain $\mathcal{K}$ is $\mathcal{O}(p^9)$. In [61] tensor products of Lagrangian polynomials on the grid of the Gauss-Lobatto points are proposed. Then the cost for computing $\mathcal{K}$ by a quadrature rule is $\mathcal{O}(p^5)$. This approach can be extended to the tetrahedral case via the Duffy transformation. If the diffusion matrix $\mathcal{A}$ is piecewise constant, the cost for the generation of the stiffness matrix can be reduced to $\mathcal{O}(p^4)$ by the technique of precomputed arrays, see [49, 63].

To give a better intuition for the underlying idea, we first present basis functions that are orthogonal with respect to the L^2-norm on triangles. Let the reference triangle $\hat{T}$ be given by the vertices $(-1,-1)$, $(1,-1)$, $(0,1)$. Dubiner [36] introduced the polynomial functions

$$\psi_{i,j}(x,y) = P_i\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i P_j^{(2i+1,0)}(y), \qquad i,j \geq 0.$$

These functions are orthogonal with respect to the L^2-inner product on $\hat{T}$. By means of the substitution $u = \frac{2x}{1-y}$, also called the Duffy transformation, we have

$$\int_{\hat{T}} \psi_{i,j}(x,y)\psi_{k,l}(x,y)d(x,y) = \int_{-1}^{1} P_i(x)P_k(x)dx \int_{-1}^{1} \left(\frac{1-y}{2}\right)^{i+k+1} P_j^{(2i+1,0)}(y)P_l^{(2k+1,0)}(y)dy.$$

The first integral evaluates to $h_i^{0,0}\delta_{i,k}$ because of the orthogonality of Legendre polynomials. Dubiner chose the parameters α of the Jacobi polynomials to fit the appearing weight function $w_{i+k+1,0}(y) = (\frac{1-y}{2})^{i+k+1}$ for $i = k$. With this one obtains

$$(\psi_{i,j}, \psi_{k,l})_{L^2(\hat{T})} = \int_{\hat{T}} \psi_{i,j}(x,y)\psi_{k,l}(x,y)\, d(x,y) = h_i^{0,0} h_j^{2i+1,0} \delta_{i,k}\delta_{j,l},$$

i.e., the family $\{\psi_{i,j}(x,y)\}_{i,j}$ is orthogonal in $L^2(\hat{T})$. Sherwin and Karniadakis [77] investigated a modification of Dubiner's orthogonal basis for triangular and tetrahedral *hp*-finite element method. The functions $\psi_{i,j}(x,y)$ do not fulfill homogeneous boundary conditions on $\partial\hat{T}$ and are therefore not suited as cell based basis functions. Sherwin and Karniadakis enforced this homogeneity by multiplication of simple bubble functions which induce a change in the parameters of the Jacobi polynomials, i.e.,

$$\tilde{\psi}_{i,j}(x,y) = \frac{1-a^2}{4} P_{i-2}^{(1,1)}(a) \left(\frac{1-b}{2}\right)^i \frac{1+b}{2} P_{j-1}^{(2i-1,1)}(b),$$

with $a = \frac{2x}{1-y}$, $b = y$. They proved a sparse structure for these basis functions for the bilinear form $a(u,v) = (u,v)_{L^2(\hat{T})}$ and for the corresponding extension to three dimensions on a tetrahedron. Sherwin and Karniadakis also observed sparsity of the system matrix for

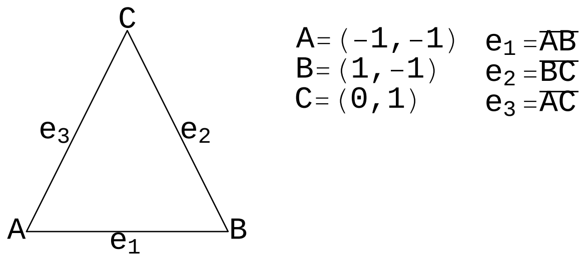


Figure 6.1: Notation of vertices and edges on the reference triangle $\hat{T}$

problem (6.1), but without proof. By applying some of the rewritings stated in Section 4.1 it can be seen that $\tilde{\psi}_{i,j}(x,y)$ is one instance of the family of basis functions that are introduced below, as are the tetrahedral cell based basis functions given in [77]. In [18] Beuchler and Schöberl investigated another basis for triangular high order finite element method based on integrated Jacobi polynomials. They proved the sparsity of the system matrix by explicitely computing the matrix entries. The relations they apply are the basis for the algorithm that we describe below and their results are also covered by our algorithm.

6.2 Definition of the Basis Functions

In this section we define the triangular and tetrahedral basis functions and state the main results on the sparsity of the element stiffness matrix. The parameter p denotes the polynomial degree.

6.2.1 The Triangular Case

Let the reference triangle $\hat{T}$ be given by the vertices $\{A,B,C\} = \{(-1,-1),(1,-1),(0,1)\}$ and edges numbered as shown in Figure 6.1. Using integrated Jacobi polynomials $\hat{p}_n^\alpha(x)$, see Definition 4.1 , we construct the shape functions on the reference element $\hat{T}$. We define the vertex based basis functions as the linear functions satisfying $\phi_i^V(V_j) = \delta_{i,j}$ for vertices $V_i \in \mathcal{V}$,

$$\phi^A(x,y) = \frac{1-2x-y}{4}, \qquad \phi^B(x,y) = \frac{1+2x-y}{4}, \qquad \phi^C(x,y) = \frac{1+y}{2}.$$

Let $\Phi_V = [\phi^A, \phi^B, \phi^C]$ be the basis of the vertex shape functions. Next we construct the edge based basis functions. For the edge e_1 we define

$$\phi_i^{e_1}(x,y) = \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i, \qquad 2 \le i \le p.$$

For the remaining two edges we define for $1 \le i \le p-1$

$$\phi_i^{e_2}(x,y) = \frac{1+2x-y}{2}\hat{p}_i^0(y), \qquad \text{and} \qquad \phi_i^{e_3}(x,y) = \frac{1-2x-y}{2}\hat{p}_i^0(y).$$

By $\Phi_E = [\{\phi_i^{e_1}\}_{i=2}^p, \{\phi_i^{e_2}\}_{i=1}^{p-1}, \{\phi_i^{e_3}\}_{i=1}^{p-1}]$ we denote the vector of all edge based basis functions on $\hat{T}$. The basis functions given so far involved only integrated Legendre polynomials. In the definition of the cell based basis functions we now use Jacobi polynomials with parameter α depending on the degree of the integrated Legendre polynomials analogously to the motivating example. We define the cell based basis functions

$$\phi_{i,j}(x,y) = \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i \hat{p}_j^{2i}(y), \qquad i+j \le p,\ i \ge 2,\ j \ge 1. \tag{6.2}$$

The basis of cell based shape functions is denoted by $\Phi_C = [\phi_{i,j}]_{i,j}$ and the total of all basis functions on $\hat{T}$ is collected in the vector $\Phi = [\phi_V, \phi_E, \phi_C]$.

Let $\mathcal{A} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{12} & \mathcal{A}_{22} \end{bmatrix}$ be a symmetric and positive definite real 2×2-matrix and let

$$\hat{K} = \int_{\hat{T}} (\nabla \Phi(x,y))^T \mathcal{A}\, \nabla \Phi(x,y)\, d(x,y)$$

be the element system matrix on $\hat{T}$ with respect to the basis Φ. According to the partitioning of the basis Φ into vertex, edge and cell based basis functions, the matrix $\hat{K}$ can be split into nine blocks, compare to (2.18). We denote by $\hat{K}_{CC}$ the block built from the cell based basis functions

$$\hat{K}_{CC} = \Big[a_{i,j;k,l}\Big]_{i,k=2;j,l=1}^{i+j\le p;k+l\le p} = \int_{\hat{T}} (\nabla \phi_{i,j}(x,y))^T \mathcal{A}\, (\nabla \phi_{k,l}(x,y))\, d(x,y). \tag{6.3}$$

The element matrix $\hat{K}_{CC}$ has $\frac{1}{2}(p-1)(p-2)$ rows and columns. In the next theorem the main result on the nonzero pattern of the interior block of the element stiffness matrix is formulated.

Theorem 6.1. *Let $\hat{K}_{CC}$ be defined via (6.2) and (6.3). Then the matrix $\hat{K}_{CC}$ has $\mathcal{O}(p^2)$ nonzero matrix entries. More precisely, $a_{i,j;k,l} = 0$ if $|i-k| > 2$ or $|i-k+j-l| > 1$.*

This theorem is proven by explicitely computing the matrix entries using the algorithm described in Section 6.3. This algorithm can also be applied to determine the sparsity pattern of $\hat{K}_{CC}$ when the cell based basis functions are defined involving a parameter $a \ge 0$ as

$$\phi_{i,j}(x,y) = \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i \hat{p}_j^{2i-a}(y), \qquad i+j \le p,\ i \ge 2,\ j \ge 1, \quad 0 \le a \le 4. \tag{6.4}$$

In [18] the case $a = 1$ was investigated and the number of nonzero matrix entries is also $\mathcal{O}(p^2)$. The optimal case, however, with respect to sparsity and condition number of the system matrix, is $a = 0$.

6.2.2 The Tetrahedral Case

Next we consider the three dimensional case and define the shape function on a tetrahedral reference element split into the vertex, edge, face and cell based basis functions. Let $\hat{T}$ be the reference tetrahedron with the vertices A, B, C and D, the edges $e_1, \ldots, e_6$, and the faces $F_1, \ldots, F_4$, see Figure 6.2. The vertex shape functions are defined as the usual hat functions, i.e.,

$$\phi^A(x,y,z) = \frac{1-2y-z-4x}{8}, \qquad \phi^C(x,y,z) = \frac{1+2y-z}{4},$$
$$\phi^B(x,y,z) = \frac{1-2y-z+4x}{8}, \qquad \phi^D(x,y,z) = \frac{1+z}{2}.$$

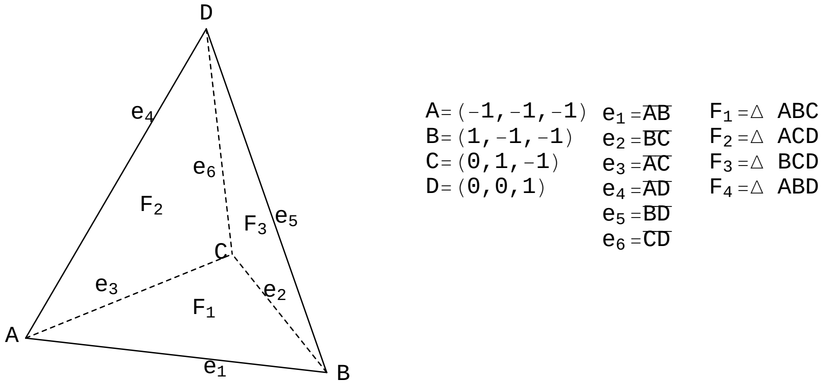


Figure 6.2: Notation of vertices, edges and faces on the reference tetrahedron $\hat{T}$

Let $\Phi_V = [\phi^A, \phi^B, \phi^C, \phi^D]$ denote the basis of vertex based basis functions. The edge based basis functions are defined as

$$\phi_i^{e_1}(x,y,z) = \hat{p}_i^0\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i, \qquad 2 \le i \le p,$$

$$\phi_i^{e_2}(x,y,z) = \frac{1-2y-z+4x}{4}\,\hat{p}_i^0\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^i, \qquad 1 \le i \le p-1,$$

$$\phi_i^{e_3}(x,y,z) = \frac{1-2y-z-4x}{4}\,\hat{p}_i^0\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^i, \qquad 1 \le i \le p-1,$$

$$\phi_i^{e_4}(x,y,z) = \frac{1-2y-z-4x}{4}\,\hat{p}_i^0(z), \qquad 1 \le i \le p-1,$$

$$\phi_i^{e_5}(x,y,z) = \frac{1-2y-z+4x}{4}\,\hat{p}_i^0(z), \qquad 1 \le i \le p-1,$$

$$\phi_i^{e_6}(x,y,z) = \frac{1+2y-z}{2}\,\hat{p}_i^0(z), \qquad 1 \le i \le p-1.$$

We denote by $\Phi_E = [\{\phi_i^{e_1}\}_{i=2}^p, \{\phi_i^{e_2}\}_{i=1}^{p-1}, \{\phi_i^{e_3}\}_{i=1}^{p-1}, \{\phi_i^{e_4}\}_{i=1}^{p-1}, \{\phi_i^{e_5}\}_{i=1}^{p-1}, \{\phi_i^{e_6}\}_{i=1}^{p-1}]$ the vector of all edge based basis functions on the reference tetrahedron $\hat{T}$. The face based basis functions read as

$$\phi_{i,j}^{F_1}(x,y,z) = \hat{p}_i^0\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i \hat{p}_j^{2i-a}\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^j,$$

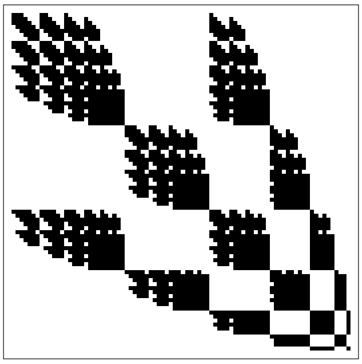
$$i+j \le p,\ 2 \le i,\ 1 \le j,$$

$$\phi_{j,k}^{F_2}(x,y,z) = \frac{1-2y-z-4x}{8}\hat{p}_j^0\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^j \hat{p}_k^{2j-b}(z),$$

$$j+k \le p-1,\ 1 \le j,k,$$

$$\phi_{j,k}^{F_3}(x,y,z) = \frac{1-2y-z+4x}{8}\hat{p}_j^0\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^j \hat{p}_k^{2j-b}(z),$$

$$j+k \le p-1,\ 1 \le j,k,$$



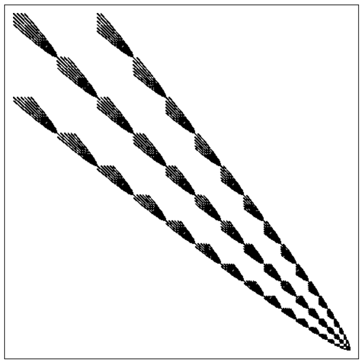


Figure 6.3: Nonzero pattern of the interior block $\hat{K}_{CC}$ for $p = 10$ (left) and $p = 24$ (right) with $a = b = 0$

$$\phi_{i,k}^{F_4}(x,y,z) = \hat{p}_i^0\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i \hat{p}_k^{2i-b}(z), \qquad i+k \le p,\ 2 \le i,\ 1 \le k.$$

By $\Phi_F = [\{\phi_i, j^{F_1}\}_{i,j}, \{\phi_{j,k}^{F_2}\}_{j,k}, \{\phi_{j,k}^{F_3}\}_{j,k}, \{\phi_{i,k}^{F_4}\}_{i,k}]$ we denote the basis of all face based basis functions. The cell based basis functions are defined as

$$\phi_{i,j,k}(x,y,z) = \hat{p}_i^0\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i \hat{p}_j^{2i-a}\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^j \hat{p}_k^{2i+2j-b}(z), \quad i+j+k \le p,\ 2 \le i,\ 1 \le j,k. \tag{6.5}$$

The parameters a, $b \in \mathbb{N}$ satisfy the following assumptions

$$0 \le a \le 4, \quad a \le b \le 6. \tag{6.6}$$

Moreover $\Phi_C = [\phi_{i,j,k}]_{i,j,k} = [\phi_{2,1,1}, \dots, \phi_{2,1,p-3}, \phi_{2,2,1}, \dots, \phi_{p-2,1,1}]$ denotes the basis of cell based basis functions in the indicated order. The basis of all basis functions is denoted by $\Phi = [\Phi_V, \Phi_E, \Phi_F, \Phi_C]$.

Next we define analogously to the previous section the element system matrix $\hat{K}$ and its inner block $\hat{K}_{CC}$ for a symmetric, positive definite real 3×3-diffusion matrix $\mathcal{A}$. The inner block is given by

$$\hat{K}_{CC} = [a_{i,j,k;l,m,n}]_{i,j,k;l,m,n} = \int_{\hat{T}} (\nabla \phi_{i,j,k}(x,y,z))^T \mathcal{A} (\nabla \phi_{l,m,n}(x,y,z))\, d(x,y,z).$$

Now we are in the position to formulate the main theorem of this section.

Theorem 6.2. *The element matrix $\hat{K}$ has $\frac{(p+1)(p+2)(p+3)}{6}$ rows and columns. If condition* (6.6) *is satisfied, each row has a bounded number of nonzero entries and the number of total nonzero entries is $\mathcal{O}(p^3)$. Moreover, the entry $a_{i,j,k;l,m,n}$ of the matrix $\hat{K}_{CC}$ is zero if $|i-l| \in \{0,2\}$, or $|i-l+j-m| > 3+a$, or $|i-l+j-m+k-n| > 2+b$.*

This theorem is proven by explicitely computing the matrix entries using the algorithm described in the next section.

6.3 Algorithm for Proving Theorems 6.1 and 6.2

In this section the algorithm for determining the entries of the element stiffness matrix built from the basis functions proposed in Sections 6.2.1 and 6.2.2 is presented. The computations of the matrix entries are, especially in the three dimensional case, very time and paper consuming. But this task can be shifted to the computer and we implemented the algorithm described below in our program IntJac within Mathematica. The basic computational steps follow the lines carried out in the proof for the sparsity of the stiffness matrix in [18], with some additions that are necessary to treat the optimal case $a = b = 0$ in three dimensions. For these computations several identities for Jacobi and integrated Jacobi polynomials are needed that were introduced in chapter 4. In the next lemma we collect these relations, reinterpreted for the polynomials $P_n^{(\alpha,0)}(x)$ and $\hat{p}_n^\alpha(x)$. To simplify notation we define $p_n^\alpha(x) = P_n^{(\alpha,0)}(x)$.

Lemma 6.3. *Let $p_n^\alpha(x) = P_n^{(\alpha,0)}(x)$ and let $\hat{p}_n^\alpha(x)$ be as defined in section 4.1. Then we have for $n \geq 1$*

$$p_n^{\alpha-1} = \frac{1}{2n+\alpha}[(n+\alpha)p_n^\alpha(x) - n\,p_{n-1}^\alpha(x)], \qquad \alpha > -1, \tag{6.7}$$

$$\begin{aligned} p_{n+1}^\alpha(x) = {} & \frac{2n+\alpha+1}{2(n+1)(n+\alpha+1)(2n+\alpha)}[(2n+\alpha+2)(2n+\alpha)x + \alpha^2]p_n^\alpha(x) \\ & - \frac{n(n+\alpha)(2n+\alpha+2)}{(n+1)(n+\alpha+1)(2n+\alpha)}p_{n-1}^\alpha(x), \qquad \alpha \geq -1, \end{aligned} \tag{6.8}$$

$$\begin{aligned} \hat{p}_n^\alpha(x) = {} & \frac{2(n+\alpha)}{(2n+\alpha-1)(2n+\alpha)}p_n^\alpha(x) + \frac{2\alpha}{(2n+\alpha-2)(2n+\alpha)}p_{n-1}^\alpha(x) \\ & - \frac{2(n-1)}{(2n+\alpha-1)(2n+\alpha-2)}p_{n-2}^\alpha(x), \qquad \alpha \geq -1, \end{aligned} \tag{6.9}$$

$$\hat{p}_n^\alpha(x) = \frac{2}{2n+\alpha-1}[p_n^{\alpha-1}(x) + p_{n-1}^{\alpha-1}(x)], \qquad \alpha > -1, \tag{6.10}$$

$$(\alpha-1)\hat{p}_n^\alpha(x) = (1-x)\,p_{n-1}^\alpha(x) + 2\,p_n^{\alpha-2}(x), \qquad \alpha > 1. \tag{6.11}$$

Proof. These relations have been proven in Section 4.1, hence we only point to the corresponding identities. The first identity is (4.7) with $\beta = 0$. Relation (6.8) is the Jacobi three term recurrence (4.14) with $\beta = 0$. The third identity (6.9) relates integrated Jacobi polynomials with their derivatives, i.e., corresponds to (4.17) with coefficients (4.18) for $\beta = -1$. The next identity (6.10) is (4.8) with $\beta = 0$. Finally (6.11) is obtained from (4.6) by setting $\beta = -1$. □

For sake of simplicity we denote in this chapter the weight function associated to Jacobi polynomials $p_n^\alpha(x)$ by $w_\alpha(x) = w_{\alpha,0}(x) = \left(\frac{1-x}{2}\right)^\alpha$. Recall the orthogonality relation for Jacobi polynomials,

$$\int_{-1}^{1} w_\alpha(x)p_i^\alpha(x)p_j^\alpha(x)\,dx = \frac{2}{2i+\alpha+1}\delta_{i,j}. \tag{6.12}$$

When computing the entries of the stiffness matrix $\hat{K}$, integrals of the form

$$\int_{\hat{T}} D_\zeta\psi_1(\xi)\,D_\eta\psi_2(\xi)\,d\xi \tag{6.13}$$

have to be evaluated, where $\hat{T}$ is the reference triangle or tetrahedron, and ζ, $\eta \in \{x, y\}$, or ζ, $\eta \in \{x, y, z\}$ in the two and three dimensional case, respectively. The functions ψ_1, ψ_2 are the cell based basis functions defined by (6.2) and (6.5). The key idea is to first decouple the integrals using the Duffy transformation and then to rewrite the integrands of the one dimensional integrals such that they can be evaluated using only the orthogonality relation (6.12). In the two dimensional case with reference triangle $\hat{T}$ as defined in Section 6.2.1, we substitute $u = \frac{2x}{1-y}$ to obtain

$$\iint_{\hat{T}} f(x,y)\, d(x,y) = \int_{-1}^{1}\int_{-1}^{1} \frac{1-y}{2}\tilde{f}(u,y)\, du\, dy.$$

For $d = 3$ and the reference tetrahedron depicted in Figure 6.2 we substitute $u = \frac{4x}{1-2y-z}$ and $v = \frac{2y}{1-z}$ to obtain

$$\iiint_{\hat{T}} f(x,y,z)\, d(x,y,z) = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \frac{1-v}{2}\left(\frac{1-z}{2}\right)^2 \tilde{f}(u,v,z)\, du\, dv\, dz.$$

The additional factors emerging because of the substitution are of the form $\left(\frac{1-\zeta}{2}\right)^\alpha$ and only change the appearing weight functions in the integrand. But, when differentiating, additional x and y might be introduced that cannot be absorbed in a weight function. As an example, consider for $d = 2$ the derivative of $\phi_{k,l}(x,y)$ with respect to y:

$$\begin{aligned}\frac{d}{dy}\phi_{k,l}(x,y) =& \frac{1}{2}x\, p^0_{k-1}\left(\frac{2x}{1-y}\right) w_{k-2}(y)\hat{p}^{2k}_l(y) - \frac{k}{2}\hat{p}^0_k\left(\frac{2x}{1-y}\right) w_{k-1}(y)\hat{p}^{2k}_l(y)\\ &+ \hat{p}^0_k\left(\frac{2x}{1-y}\right) w_k(y) p^{2k}_{l-1}(y).\end{aligned}$$

The three term recurrence (6.8) can be used to replace in the first expression on the right hand side $x\, p^0_{k-1}\left(\frac{2x}{1-y}\right)$ and we obtain

$$\begin{aligned}\frac{d}{dy}\phi_{k,l}(x,y) =& \frac{1}{4k-2}\Bigg[(k-1)p^0_{k-2}\left(\frac{2x}{1-y}\right)\hat{p}^{2k}_l(y)w_{k-1}(y) + kp^0_k\left(\frac{2x}{1-y}\right)\hat{p}^{2k}_l(y)w_{k-1}(y)\\ &-(2k-1)\hat{p}^0_k\left(\frac{2x}{1-y}\right)\left(k\hat{p}^{2k}_l(y)w_{k-1}(y) - 2p^{2k}_{l-1}(y)w_k(y)\right)\Bigg].\end{aligned}$$

This elimination is carried out before the Duffy transformation is performed. Let us continue with the example and multiply $\frac{d}{dy}\phi_{k,l}(x,y)$, as rewritten above, with

$$\frac{d}{dx}\phi_{i,j}(x,y) = p^0_{i-1}\left(\frac{2x}{1-y}\right) w_{i-1}(y)\hat{p}^{2i}_j(y).$$

Then altogether, including Duffy transformation, one arrives at the following integrals:

$$\begin{aligned}&\iint_{\hat{T}} \frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)\, d(x,y) =\\ &\quad\frac{(k-1)}{2(2k-1)}\int_{-1}^{1} p^0_{i-1}(x)p^0_{k-2}(x)\, dx \int_{-1}^{1} w_{i+k-1}(y)\hat{p}^{2i}_j(y)\hat{p}^{2k}_l(y)\, dy\\ &\quad+\frac{k}{2(2k-1)}\int_{-1}^{1} p^0_{i-1}(x)p^0_k(x)\, dx \int_{-1}^{1} w_{i+k-1}(y)\hat{p}^{2i}_j(y)\hat{p}^{2k}_l(y)\, dy\\ &\quad-\frac{1}{2}\int_{-1}^{1} p^0_{i-1}(x)\hat{p}^0_k(x)\, dx \int_{-1}^{1} \frac{1}{2}\hat{p}^{2i}_j(y)\left[kw_{i+k-1}(y)\hat{p}^{2k}_l(y) - 2w_{i+k}(y)p^{2k}_{l-1}(y)\right] dy.\end{aligned} \tag{6.14}$$

These operations of taking derivatives of the basis functions, eliminating "disturbing" variables and performing the Duffy transformation are also handed over to the computer. These preparative steps are carried out by the Prepare2DIntegrand and Prepare3DIntegrand commands of our program. The integrals appearing in (6.14) are all of the form

$$\int_{-1}^{1} w_\alpha(u) q_1(u) q_2(u)\, du,$$

where $q_1(u)$, $q_2(u)$ are Jacobi or integrated Jacobi polynomials. Next, these polynomials have to be rewritten in terms of Jacobi polynomials fitting to the appearing weight. Because of the dependence of the parameters, the integrals are processed from left to right, i.e., starting with the integration with respect to x.

For the coefficient matrix $\mathcal{A}$ equal the identity matrix in the tetrahedral case, the integrand has the form

$$\hat{\mathcal{I}} = \frac{d}{dx}\phi_{i,j,k}\frac{d}{dx}\phi_{l,m,n} + \frac{d}{dy}\phi_{i,j,k}\frac{d}{dy}\phi_{l,m,n} + \frac{d}{dz}\phi_{i,j,k}\frac{d}{dz}\phi_{l,m,n}.$$

Executing the Prepare3DIntegrand command on this integrand results in 64 integrands of the form

$$\hat{\mathcal{I}}^{(r)} = c_r\, p_{x,1}\, p_{x,2}\, w_1(y)\, p_{y,1}\, p_{y,2}\, w_2(z)\, p_{z,1}\, p_{z,2},$$

i.e., $\hat{\mathcal{I}} = \sum_{r=1}^{64} \hat{\mathcal{I}}^{(r)}$, where the coefficients c_r are rational in the polynomial degrees, $p_{\zeta,i}$ are Jacobi or integrated Jacobi polynomials and $w_i(\zeta)$ are Jacobi weight functions. The integrands for $\hat{K}$ are listed in Tables 6.1-6.2. In [16] only 21 integrands were used, because another manual rewriting of the derivatives entered the computation. We chose to present here a proof with as little human interaction as necessary. But even the 21 integrands given in [16] illustrate the complexity of the problem of determining the nonzero pattern of the system matrix. The main part of the program handles the rewriting of the integrands until the final evaluation of the integrals. The basic steps of the algorithm for each integration variable are:

1. Collect integrands depending on the current integration variable
2. For each integrand: Rewrite integrated Jacobi polynomials in terms of Jacobi polynomials using (6.9), (6.10), or (6.11)
3. Collect integrands depending on the current integration variable
4. For each integrand: Adjust Jacobi polynomials to appearing weight functions
5. Collect integrands depending on the current integration variable
6. For each integrand: Evaluate integrals using orthogonality relation (6.12)

The two steps of the algorithm that need further explanations are steps 2 and 4. Which of the identities relating integrated Jacobi polynomials and Jacobi polynomials (6.9)- (6.11) have to be used in step 2 depends on the difference $\gamma - \alpha$ of the parameters of $\hat{p}_n^\alpha(\zeta)$ and of the weight function $w_\gamma(\zeta)$.

2. Rewrite $w_\gamma(\zeta)\hat{p}_n^\alpha(\zeta)$ in terms of Jacobi polynomials

   (a) $\gamma - \alpha \geq 0$: transform integrated Jacobi polynomials to Jacobi polynomials with same parameter using (6.9).

	$p_{x,1}$	$p_{x,2}$	w_1	$p_{y,1}$	$p_{y,2}$	w_2	$p_{z,1}$	$p_{z,2}$
$\hat{\mathcal{I}}^{(1)}$	$p_{i-2}^0(x)$	$p_{l-2}^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(2)}$	$p_{i-2}^0(x)$	$p_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(3)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(4)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(5)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(6)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(7)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(8)}$	$p_{i-2}^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(9)}$	$p_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(10)}$	$p_i^0(x)$	$p_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(11)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(12)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(13)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(14)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(15)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(16)}$	$p_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(17)}$	$p_{i-1}^0(x)$	$p_{l-1}^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(18)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(19)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(20)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(21)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l}(y)$	$p_{j-2}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(22)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l}(y)$	$p_{j-1}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(23)}$	$\hat{p}_i^0(x)$	$p_{l-2}^0(x)$	$w_{i+l}(y)$	$p_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(24)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(25)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(26)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(27)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l}(y)$	$p_{j-2}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(28)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l}(y)$	$p_{j-1}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(29)}$	$\hat{p}_i^0(x)$	$p_l^0(x)$	$w_{i+l}(y)$	$p_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(30)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l-1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(31)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(32)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(33)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(34)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(35)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(36)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(37)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$p_{j-2}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(38)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$p_{j-1}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(39)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l}(y)$	$p_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(40)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$

Table 6.1: Integrands for product of x-, y- and z-derivatives and y-derivatives, part I, $\delta = i+j$, $\epsilon = l + m$

	$p_{x,1}$	$p_{x,2}$	w_1	$p_{y,1}$	$p_{y,2}$	w_2	$p_{z,1}$	$p_{z,2}$
$\hat{\mathcal{I}}^{(41)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(42)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(43)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+2}(z)$	$p_{k-1}^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(44)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(45)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(46)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(47)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(48)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(49)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$\hat{p}_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$p_{k-1}^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(50)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-2}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(51)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-2}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(52)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-2}^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(53)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-2}^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(54)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-2}^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(55)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-1}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(56)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-1}^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(57)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-1}^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(58)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-1}^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(59)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_{j-1}^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(60)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(61)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_j^{2i-a}(y)$	$\hat{p}_m^{2l-a}(y)$	$w_{\epsilon+\delta+1}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$p_{n-1}^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(62)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_j^{2i-a}(y)$	$p_{m-2}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(63)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_j^{2i-a}(y)$	$p_{m-1}^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$
$\hat{\mathcal{I}}^{(64)}$	$\hat{p}_i^0(x)$	$\hat{p}_l^0(x)$	$w_{i+l+1}(y)$	$p_j^{2i-a}(y)$	$p_m^{2l-a}(y)$	$w_{\epsilon+\delta}(z)$	$\hat{p}_k^{2\delta-b}(z)$	$\hat{p}_n^{\epsilon-b}(z)$

Table 6.2: Integrands for product of x-, y- and z-derivatives and y-derivatives, part II, $\delta = i+j$, $\epsilon = l+m$

(b) $\gamma - \alpha = -1$: transform integrated Jacobi polynomials to Jacobi polynomials with parameter $\alpha - 1$ using (6.10)

(c) $\gamma - \alpha = -2$: use the mixed relation (6.11) to obtain

$$w_\gamma(\zeta)\hat{p}_n^{\gamma+2}(\zeta) = \frac{2}{\gamma+1}\left(w_\gamma(\zeta)p_n^\gamma(\zeta) + w_{\gamma+1}(\zeta)p_{n-1}^{\gamma+2}(\zeta)\right).$$

This transformation is executed (internally) by the RemovePHAT function. If the integrand contains two integrated Jacobi polynomials RemovePHAT is applied with respect to the original weight function, not taking into account a possible change introduced by relation (6.11). The situation $\gamma - \alpha = -2$, i.e., case 2(c), *only* happens for $a = b$ or $a = 0$. This includes the optimal cell based basis functions with $a = b = 0$. If none of the cases 2(a)-2(c) applies, the algorithms interrupts.

Rewriting the Jacobi polynomials $p_n^\alpha(\zeta)$ in terms of $p_n^\gamma(\zeta)$ fitting to the appearing weights $w_\gamma(\zeta)$ in step 4, means lifting the polynomial parameter α using (6.7) $(\gamma-\alpha)$ times. This transformation is performed by the AdjustToWeight function recursively for each appearing Jacobi polynomial.

4. Rewrite the Jacobi polynomials $p_n^\alpha(\zeta)$ in terms of Jacobi polynomials fitting to the appearing weights $w_\gamma(\zeta)$ $(\gamma-\alpha>0)$ by lifting the polynomial parameter α using (6.7) $(\gamma-\alpha)$-times, i.e., written in explicit form we have

$$p_n^\alpha(\zeta)=\sum_{m=0}^{\gamma-\alpha}(-1)^k\binom{\gamma-\alpha}{m}\frac{(n+\gamma-m)^{\underline{\gamma-\alpha-m}}\,n^{\underline{m}}}{(2n+\gamma-m+1)^{\underline{\gamma-\alpha+1}}}(2n-2m+\gamma+1)p_{n-m}^\gamma(\zeta),$$

where $a^{\underline{k}}=a(a-1)\cdot\ldots\cdot(a-k+1)$ denotes the falling factorial.

If $\gamma-\alpha<0$ the algorithm interrupts. In this step of the algorithm polynomials down to degree $n-\gamma+\alpha$ are introduced. Hence this transformation is a costly one as it increases the number of terms significantly, especially if a, b are far from the ideal case $a=b=0$.

If step 2(c) of the algorithm is executed a further rewriting step might be necessary to avoid abortion in step 4. We comment on this correction step below. First we continue our example of computing the matrix entries for the mixed product in the two dimensional case, where we go through the transformations step by step. After preparing the integrand we arrived at the representation (6.14). We start by integrating with respect to x and in step 2 of the algorithm. In this case only the integrated Legendre polynomials in the third integral need to be rewritten in terms of Legendre polynomials, i.e., the condition in step 2(a) applies.

In[29]:= **RemovePHAT[*p*[*i* − 1, 0, *x*]phat[*k*, 0, *x*]]**

Out[29]= $-\frac{p(i-1,0,x)(p(k-2,0,x)-p(k,0,x))}{2k-1}$

Next, the integrands are collected again yielding for (6.14)

$$\iint_{\hat{T}}\frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)=\int_{-1}^{1}p_{i-1}^0(x)p_k^0(x)\,dx\int_{-1}^{1}\frac{w_{i+k}(y)p_{l-1}^{2k}(y)\hat{p}_j^{2i}(y)}{2k-1}\,dy$$
$$+\int_{-1}^{1}p_{i-1}^0(x)p_{k-2}^0(x)\,dx\int_{-1}^{1}\frac{\hat{p}_j^{2i}(y)\left((2k-1)w_{i+k-1}(y)\hat{p}_l^{2k}(y)-2w_{i+k}(y)p_{l-1}^{2k}(y)\right)}{4k-2}\,dy$$

Step 4 is not executed for the integration with respect to x, because all polynomials already fit to the appearing weight $w_0(x)\equiv 1$. After evaluating the integrals and collecting integrands we have

$$\begin{aligned}\iint_{\hat{T}}\frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)d(x,y)&=\frac{2\delta_{0,-i+k+1}}{(2i-3)(2i-1)}\int_{-1}^{1}w_{2i-1}(y)\hat{p}_j^{2i}(y)p_{l-1}^{2i-2}(y)\,dy\\&-\frac{2\delta_{0,-i+k-1}}{(2i-1)(2i+1)}\int_{-1}^{1}w_{2i+1}(y)\hat{p}_j^{2i}(y)p_{l-1}^{2i+2}(y)\,dy\\&+\frac{\delta_{0,-i+k-1}}{2i-1}\int_{-1}^{1}w_{2i}(y)\hat{p}_j^{2i}(y)\hat{p}_l^{2i+2}(y)\,dy.\end{aligned}\tag{6.15}$$

Next integrated Jacobi polynomials are rewritten in terms of Jacobi polynomials in step 2 of the algorithm. For the first integrand we have $\gamma-\alpha=-1$, i.e., we are in case 2(b) and RemovePHAT rewrites using (6.10).

```
In[30]:= RemovePHAT[p[-1 + l, -2 + 2i, y]phat[j, 2i, y]w[-1 + 2i, y]]
```

Out[30]= $\dfrac{2(p[-1+j,-1+2i,y]+p[j,-1+2i,y])\,p[-1+l,-2+2i,y]\,w[-1+2i,y]}{-1+2i+2j}$

The integrand in the second integral can be reformulated using (6.9), i.e., step 2(a) is executed,

$$w_{2i+1}(y)\hat{p}_j^{2i}(y)p_{l-1}^{2i+2}(y) = w_{2i+1}(y)p_{l-1}^{2i+2}(y)\Big[-\frac{(j-1)p_{j-2}^{2i}(y)}{(i+j-1)(2i+2j-1)} + \frac{ip_{j-1}^{2i}(y)}{(i+j-1)(i+j)} + \frac{(2i+j)p_j^{2i}(y)}{(i+j)(2i+2j-1)}\Big].$$

In the third integral the integrand is a product of two integrated Jacobi polynomials. The first polynomial $\hat{p}_j^{2i}(y)$ is rewritten using (6.9), i.e., step 2(a) is executed again. The second polynomial needs the rewriting (6.11), i.e., we are in case 2(c).

$$w_{2i}(y)\hat{p}_j^{2i}(y)\hat{p}_l^{2i+2}(y) = \frac{2}{2i+1}\left(w_{2i}(y)p_l^{2i}(y) + w_{2i+1}(y)p_{l-1}^{2i+2}(y)\right)\Big[-\frac{(j-1)p_{j-2}^{2i}(y)}{(i+j-1)(2i+2j-1)} + \frac{i\,p_{j-1}^{2i}(y)}{(i+j-1)(i+j)} + \frac{(2i+j)p_j^{2i}(y)}{(i+j)(2i+2j-1)}\Big].$$

In the last two rewritings one obtains combinations of weight function $w_{2i+1}(y)$ and Jacobi polynomials $p_{l-1}^{2i+2}(y)$. These expressions could not be handled by the transformation of step 4, but these terms cancel. A posteriori, we see that this cancellation would have been possible already in (6.15) by using the relation (6.11). However, doing the cancellation *after* transforming the integrated Jacobi polynomials is more efficient for the implementation. The algorithm therefore proceeds with the steps described above. Either way, after applying RemovePHAT to (6.15) and collecting integrands we arrive at

$$\iint_{\hat{T}} \frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)d(x,y) = -\int_{-1}^{1}\frac{2(j-1)\delta_{0,-i+k-1}w_{2i}(y)p_{j-2}^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j-1)(2i+2j-1)}\,dy + \int_{-1}^{1}\frac{2i\delta_{0,-i+k-1}w_{2i}(y)p_{j-1}^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j-1)(i+j)}\,dy + \int_{-1}^{1}\frac{2(2i+j)\delta_{0,-i+k-1}w_{2i}(y)p_j^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j)(2i+2j-1)}\,dy + \int_{-1}^{1}\frac{4\delta_{0,-i+k+1}w_{2i-1}(y)p_{l-1}^{2i-2}(y)p_{j-1}^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)}\,dy + \int_{-1}^{1}\frac{4\delta_{0,-i+k+1}w_{2i-1}(y)p_{l-1}^{2i-2}(y)p_j^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)}\,dy.$$

It remains to adjust the Jacobi polynomials in the integrands to the appearing weights. Doing so for each integrand and collecting terms again yields the following integrand

$$-\frac{2(j-1)\delta_{0,-i+k-1}w_{2i}(y)p_{j-2}^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j-1)(2i+2j-1)} + \frac{2i\delta_{0,-i+k-1}w_{2i}(y)p_{j-1}^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j-1)(i+j)} + \frac{2(2i+j)\delta_{0,-i+k-1}w_{2i}(y)p_j^{2i}(y)p_l^{2i}(y)}{(2i-1)(2i+1)(i+j)(2i+2j-1)} - \frac{4(l-1)\delta_{0,-i+k+1}w_{2i-1}(y)p_{j-1}^{2i-1}(y)p_{l-2}^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)(2i+2l-3)} - \frac{4(l-1)\delta_{0,-i+k+1}w_{2i-1}(y)p_j^{2i-1}(y)p_{l-2}^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)(2i+2l-3)} + \frac{4(2i+l-2)\delta_{0,-i+k+1}w_{2i-1}(y)p_{j-1}^{2i-1}(y)p_{l-1}^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)(2i+2l-3)} + \frac{4(2i+l-2)\delta_{0,-i+k+1}w_{2i-1}(y)p_j^{2i-1}(y)p_{l-1}^{2i-1}(y)}{(2i-3)(2i-1)(2i+2j-1)(2i+2l-3)}.$$

These integrals are easily evaluated using orthogonality relation (6.12). Altogether we obtain, for $c_1, \dots, c_6$ rational functions in i and j,

$$\iint_{\hat{T}} \frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)d(x,y) = c_1\delta_{0,-i+k-1}\delta_{0,l-j} + c_2\delta_{0,-i+k-1}\delta_{0,-j+l+1} + c_3\delta_{0,-i+k-1}\delta_{0,-j+l+2} + c_4\delta_{0,-i+k+1}\delta_{0,-j+l-2} + c_5\delta_{0,-i+k+1}\delta_{0,-j+l-1} + c_6\delta_{0,-i+k+1}\delta_{0,l-j}.$$

From this result one can read off that the integrals over the mixed product of derivatives are nonzero only if $|i-k| = 1$ and $|i-k+j-l| \leq 1$. The concrete values of the rational functions c_i and the output of IntJac when computing (6.14) are given in Appendix A.2.

Let us point out that in the general case, for the triangular shape functions as well as for the tetrahedral shape functions, the exceptional cases $\gamma - \alpha < -2$ in step 2 or $\gamma - \alpha < 0$ in step 4 of the algorithm (that would lead to an interruption of the program) *never* occurs. In the tetrahedral case this is because of the range of parameters a, b, i.e., $0 \leq a \leq 4$ and $a \leq b \leq 6$.

In the three dimensional case for $a = b = 0$ in step 2(c) of the algorithm also terms including $w_\gamma(y)p_j^{\gamma+1}(y)$ are introduced. Most of them cancel as shown in the two dimensional example above. The remaining terms of this form appear only as factors of Jacobi three term recurrences that reduce to zero. These expressions can also be detected automatically and be removed when integrands are collected in step 3. Hence, also in the optimal case, the program does not abort in step 4. For these necessary correction steps more than one of the integrands in Tables 6.1-6.2 are needed. The evaluation of the single integrals leads to a dense matrix. Only the combinations of several of the integrands introduce the cancellations.

We close this section by stating the nonzero pattern of the blocks containing the mixed terms $D_\zeta\phi_{i,j,k}D_\eta\phi_{l,m,n}$ for $(\zeta,\eta) \in \{(x,y),(x,z),(y,z)\}$. After using Prepare3DIntegrand one obtains in total 44 integrands, again all of the form

$$\hat{\mathcal{I}}^{(r)} = c_r\, p_{x,1}\, p_{x,2}\, w_1(y)\, p_{y,1}\, p_{y,2}\, w_2(z)\, p_{z,1}\, p_{z,2}.$$

Applying our program to evaluate these integrals we obtain the following results:

- For $(\zeta,\eta) = (x,y)$ the matrix entries are nonzero if $|i-l| = 1$, $|i-l+j-m| \leq 1+a$ and $|i-l+j-m+k-n| \leq 2+b$.
- For $(\zeta,\eta) = (x,z)$ the matrix entries are nonzero if $|i-l| = 1$, $|i-l+j-m| \leq 2+a$ and $|i-l+j-m+k-n| \leq 2+b$.
- For $(\zeta,\eta) = (x,y)$ the matrix entries are nonzero if $|i-l| \in \{0,2\}$, $|i-l+j-m| \leq 2+a$ and $|i-l+j-m+k-n| \leq 2+b$.

6.4 Numerical Properties and Applications

6.4.1 Properties of the Interior Block of the Stiffness Matrix

Now we briefly summarize the most important computational properties of the proposed basis functions. We assume throughout this section that the coefficient matrix equals the identity

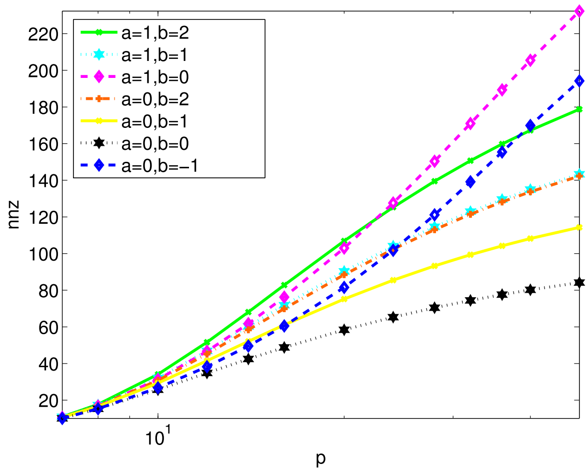


Figure 6.4: Averaged number of the nonzero matrix entries of $\hat{K}_{CC}$ per row

matrix, i.e., $\mathcal{A} = I$. In Figure 6.3 the nonzero pattern of the matrix $\hat{K}_{CC}$ is displayed, i.e., the block of the interior bubbles for the Laplacian, using the basis functions

$$\phi_{i,j,k}(x,y,z) = \hat{p}_i^0 \left(\frac{4x}{1-2y-z}\right) \left(\frac{1-2y-z}{4}\right)^i \hat{p}_j^{2i} \left(\frac{2y}{1-z}\right) \left(\frac{1-z}{2}\right)^j \hat{p}_k^{2i+2j}(z),$$

for $p = 24$. A typical stencil like structure of the nonzero entries can be observed. The average number of nonzero entries per row are bounded by a constant $c_{a,b}$ which is independent of the maximal polynomial degree p. In general one obtains

$$c_{a,b} = 3(2a+7)(2b+5).$$

This constant depends only on the special choice of parameters a and b and is minimal for $a = b = 0$. This minimality is a consequence of the proof of Theorem 6.2. If the condition (6.6) on the parameters a, and b is violated, then the averaged number of nonzero entries increases with p, see Figure 6.4.

Figure 6.5 displays the maximal and the inverse of the minimal eigenvalue of the diagonally preconditioned matrix $\hat{K}_{CC}$. In all cases the maximal eigenvalue is bounded by a constant of about $7, \ldots, 15$. The minimal eigenvalue $\lambda_{\min}$ depends strongly on the choice of a and b. From the numerical results one can conclude that $\lambda_{\min}^{-1}$ grows as $\mathcal{O}(p^{\max\{4,4+2a,4+2b,4+2a+2b\}})$. So, the condition number grows at least with p^4. The optimal order for the condition number can be achieved if $a, b \leq 0$. But, in combination with Theorem 6.2, the basis with $a = b = 0$ should be preferred since it yields the lowest number of nonzero entries and the best condition number.

In the experiments the nonzero matrix entries were computed with a sum factorization algorithm, see [61]. The remaining one dimensional integrals are computed recursively using the product recurrence (3.12) with the three term recurrences for Jacobi and integrated Jacobi polynomials.

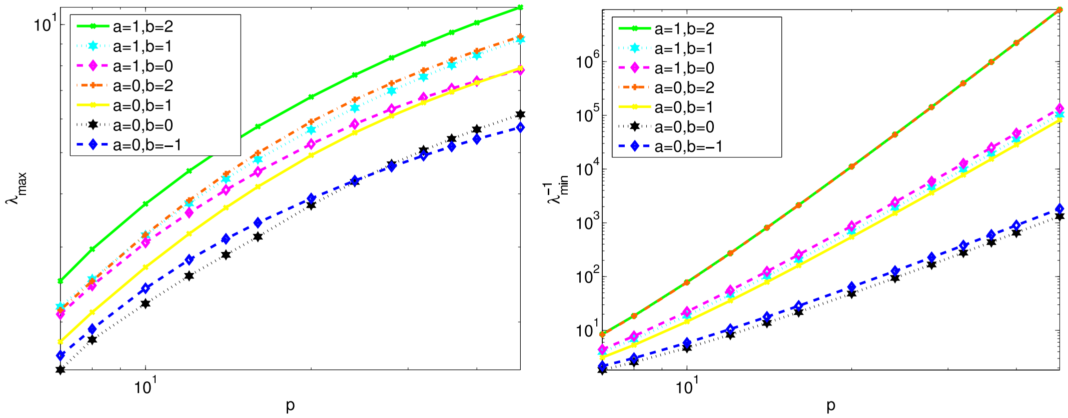


Figure 6.5: Maximal and inverse of minimal eigenvalue of the diagonally preconditioned matrix $\hat{K}_{CC}$

6.4.2 Application: A Preconditioner for the Cell Based Basis Functions

In this section we derive a simple preconditioner for the interior block of the stiffness matrix for the coefficient matrix $\mathcal{A} = I$. It is well known from the literature that preconditioned conjugate gradient methods (pcg-methods) with domain decomposition preconditioners of Dirichlet-Dirichlet-type are among the most efficient iterative solvers for systems arising from variational equations such as (6.1), cf. [3, 8, 48, 54].

The stiffness matrix $\mathcal{K}$ can be written in block structure corresponding to a partition of the basis functions $\Phi = [\Phi_V, \Phi_E, \Phi_F, \Phi_C] = [\Phi_{ext}, \Phi_{int}]$ into exterior degrees of freedom $\Phi_{ext} = [\Phi_V, \Phi_E, \Phi_F]$ and interior degrees of freedom $\Phi_{int} = \Phi_C$. Let

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_{ext} & \mathcal{K}_{ext,int} \\ \mathcal{K}_{int,ext} & \mathcal{K}_{int,ext} \end{bmatrix} = \begin{bmatrix} I & \mathcal{K}_{ext,int}\mathcal{K}_{int}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{S} & 0 \\ 0 & \mathcal{K}_{int} \end{bmatrix} \begin{bmatrix} I & 0 \\ \mathcal{K}_{int}^{-1}\mathcal{K}_{int,ext} & I \end{bmatrix}$$

be this block structure with Schur complement

$$\mathcal{S} = \mathcal{K}_{ext} - \mathcal{K}_{ext,int}\mathcal{K}_{int}^{-1}\mathcal{K}_{int,ext}.$$

Our domain decomposition preconditioner $\mathcal{C}$ for the matrix $\mathcal{K}$ is of the form

$$\mathcal{C} = \begin{bmatrix} I & -\mathcal{E}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{C}_{\mathcal{S}} & 0 \\ 0 & \mathcal{C}_{int} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\mathcal{E} & I \end{bmatrix}, \tag{6.16}$$

where

- $\mathcal{C}_{int}$ is a preconditioner for the inner block $\mathcal{K}_{int}$,
- $\mathcal{C}_{\mathcal{S}}$ is a preconditioner for the Schur complement $\mathcal{S}$, and
- $\mathcal{E}$ is the matrix representation of an extension operator acting from the edges of the elements into the interior.

Preconditioners for the Schur complement have been proposed e.g. in [54]. The papers [2, 8, 62] deal with extension operators for the p-version of the finite element method using triangular or tetrahedral elements. In [14, 46] an algebraic analysis of a preconditioner of type (6.16) is given.

Since the global stiffness matrix is built from the local contributions of the element matrices using a local-to-global connection matrix and since the computations on arbitrary element matrices can be reduced to computations on the reference element, it is sufficient to consider a preconditioner for a single reference element. Now we propose a relatively simple preconditioner $\mathcal{C}_{int}$ for $\mathcal{K}_{int}$ and (based on this) a matrix representation $\mathcal{E}$ for the extension operator of the form (6.16). Let

$$\mathcal{C}_0 = \int_{\hat{T}} (\nabla\Phi(x,y,z))^T \nabla\Phi(x,y,z)\, d(x,y,z)$$

be the stiffness matrix on the reference tetrahedron $\hat{T}$ and consider the block decomposition of $\mathcal{C}_0$

$$\mathcal{C}_0 = \begin{bmatrix} \mathcal{C}_{ext} & \mathcal{C}_{ext,int} \\ \mathcal{C}_{int,ext} & \mathcal{C}_{int} \end{bmatrix}.$$

Theorem 6.4. *Let $\mathcal{C}_\mathcal{S}$ be a preconditioner for the Schur complement such that $\mathcal{C}_\mathcal{S}^{-1}\underline{u}$ requires not more than $\mathcal{O}(p^6)$ operations and such that*

$$c_1\,(\mathcal{C}_\mathcal{S}\underline{v},\underline{v}) \le (\mathcal{S}\underline{v},\underline{v}) \le c_2\,(\mathcal{C}_\mathcal{S}\underline{v},\underline{v}), \qquad \forall \underline{v}, \tag{6.17}$$

for some constants c_1, c_2. Define the preconditioner

$$\mathcal{C}_1 = \begin{bmatrix} I & \mathcal{C}_{ext,int}\mathcal{C}_{int}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{C}_\mathcal{S} & 0 \\ 0 & \mathcal{C}_{int} \end{bmatrix} \begin{bmatrix} I & 0 \\ \mathcal{C}_{int}^{-1}\mathcal{C}_{int,ext} & I \end{bmatrix}.$$

Then $\kappa(\mathcal{C}_1^{-1/2}\mathcal{K}\mathcal{C}_1^{-1/2}) = \mathcal{O}(\frac{c_2}{c_1})$. The operation $\mathcal{C}_1^{-1}\underline{u}$ requires $\mathcal{O}(p^6)$ operations.

Proof. The proof is similar to the proof of Theorem 4.2. in [18]. Using [48] one can prove that $\kappa(\mathcal{C}_0^{-1/2}\mathcal{K}\mathcal{C}_0^{-1/2}) = \mathcal{O}(1)$. Hence the first assertion follows from equation (6.17) immediately.

To prove the complexity argument for $\mathcal{C}_1^{-1}\underline{u}$, we investigate the nonzero pattern for the matrix $\hat{K}_{CC}$ with $\mathcal{A} = I$. Because of Theorem 6.2, see also Figure 6.3, the nonzero pattern has the structure of a 3D-finite difference stencil.

Let $(\mathbb{V},\mathbb{E})$ be the corresponding graph of the matrix $\hat{K}_{CC}$. Then $(\mathbb{V},\mathbb{E})$ has an $\mathcal{O}(N^{2/3})$ separator property and therefore the method of nested dissection, [42], yields a total cost of $\mathcal{O}(N^2) = \mathcal{O}(p^6)$, see [57]. □

6.5 Further Results

All basis functions presented in Sections 6.2.1 and 6.2.2 are of the same form, namely products of certain integrated Jacobi polynomials and Jacobi weight functions. Hence, the algorithm described in Section 6.3 is applicable to compute the entries of all blocks of the element stiffness matrix. In [18] the matrix entries for all blocks for the triangular case and the cell based basis functions (6.4) with $a = 1$ are listed. Here, we do not give the pattern for all blocks, but only state the results for two examples in the tetrahedral case with $a = b = 0$

to support our assertion. First, consider the block $\hat{K}_{F_1C}$ of the stiffness matrix. Let $\hat{K}^{yz}_{F_1C}$ denote the part stemming from the product

$$\int_{\hat{T}} \frac{d}{dy}\phi^{F_1}_{i,j}(x,y,z)\frac{d}{dz}\phi_{l,m,n}(x,y,z)\,d(x,y,z), \qquad \begin{array}{c} 2 \le i,\ 1 \le j,\ i+j \le p \\ 2 \le l,\ 1 \le m,n,\ l+m+n \le p \end{array}.$$

Recall the definition of the face based shape function for $a = 0$,

$$\phi^{F_1}_{i,j}(x,y,z) = \hat{p}^0_i\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i \hat{p}^{2i}_j\left(\frac{2y}{1-z}\right)\left(\frac{1-z}{2}\right)^j.$$

Again derivation and Duffy transformation can be performed automatically yielding integrands of the form

$$\hat{\mathcal{I}} = p_{x,1}p_{x,2}\,w_1(y)p_{y,1}p_{y,2}\,w_2(z)p_{z,1},$$

where the integration with respect to z is particularly simple, because the variable z appears in $\phi^{F_1}_{i,j}$ only in the denominators of the arguments and in the corresponding compensating factors. For these integrals we have that

$$\int_{-1}^{1} w_\alpha(z)p^\alpha_n(z)\,dz = \int_{-1}^{1} w_\alpha(z)p^\alpha_n(z)p^\alpha_0(z)\,dz = \frac{2}{\alpha+1}\delta_{n,0},$$

which can again be evaluated automatically. For $a = b = 0$ the matrix entries of $\hat{K}^{yz}_{F_1C}$ are nonzero if $|i-l| \in \{0,2\}$, $1 \le n \le 4$ and $n-2 \le i+j-l-m \le 2$. Next consider the block $\hat{K}^{zz}_{F_4e_6}$ with entries

$$\int_{\hat{T}} \frac{d}{dz}\phi^{F_4}_{i,j}(x,y,z)\frac{d}{dz}\phi^{e_6}_l(x,y,z)\,d(x,y,z), \qquad \begin{array}{c} 1 \le l \le p \\ 2 \le i,\ 1 \le j,\ i+j \le p \end{array}.$$

After taking derivatives and substituting, integrands of the form

$$\hat{\mathcal{I}} = p_{x,1}\,w_1(y)\,w_2(z)\,p_{z,1}p_{z,2}$$

are obtained. Hence the matrix entries can be determined analogously to the previous example with our program yielding that $\hat{K}^{zz}_{F_4e_6}$ is nonzero if $i = 2$ and $-5 \le j-l \le 1$.

The proposed basis functions also lead to a sparse system matrix for a convection reaction diffusion equation of the form

$$-\operatorname{div}(\mathcal{A}\nabla u) + \underline{b}\cdot\nabla u + cu = f, \tag{6.18}$$

with piecewise constant coefficients $\underline{b}$, c, and piecewise constant diffusion matrix $\mathcal{A}$. The variational formulation of equation (6.18) leads to the bilinear forms

$$a(u,v) = \int_\Omega (\nabla u)^T\mathcal{A}\nabla v\,dx, \quad b(u,v) = \int_\Omega u(\underline{b}\cdot\nabla v),\ dx \quad \text{and} \quad c(u,v) = \int_\Omega c\,uv\,dx.$$

Given these bilinear forms, we define the stiffness matrix $\hat{K}$ as before and, analogously, the convection matrix $\hat{B}$ and the mass matrix $\hat{M}$ via

$$\hat{B} = \int_{\hat{T}} \Phi(x,y,z)^T(\underline{b}\cdot\nabla\Phi(x,y,z))\,d(x,y,z), \qquad \hat{M} = \int_{\hat{T}} c\,\Phi(x,y,z)^T\Phi(x,y,z)\,d(x,y,z).$$

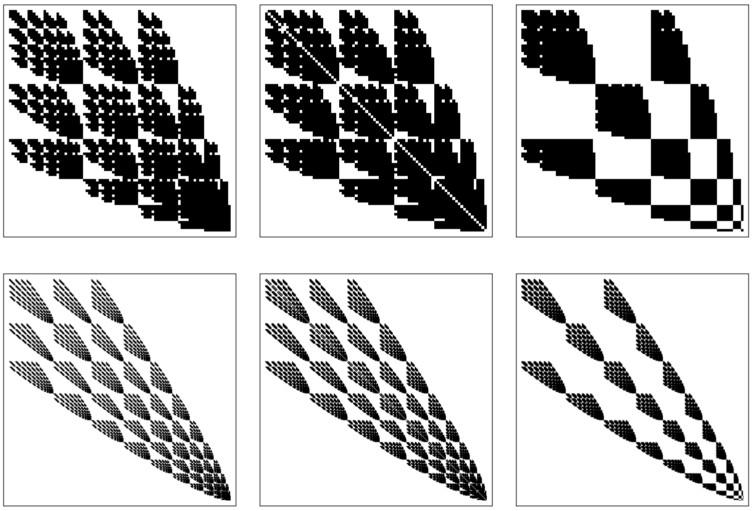


Figure 6.6: Nonzero pattern of the interior blocks $\hat{K}_{CC}$, $\hat{B}_{CC}$, $\hat{M}_{CC}$ (from left to right), Topline: $p = 10$, Bottomline: $p = 16$

These integrands have the same structure as the integrands for the entries of the stiffness matrix. Thus the algorithm is also applicable to determine the nonzero pattern of these matrices and, moreover, it terminates. Let $\hat{B}_{CC}$ and $\hat{M}_{CC}$ denote the interior blocks of the convection and mass matrix, respectively. Regarding the nonzero pattern for $\hat{B}_{CC}$ using the cell based basis functions (6.5) with $a = b = 0$ we obtain:

$$\int_{\hat{T}} \frac{d}{d\zeta}\phi_{i,j,k}(x,y,z)\,\phi_{l,m,n}(x,y,z)\,d(x,y,z) \neq 0, \qquad \zeta \in \{x,y,z\},$$

if $|i-l| \leq 2$ and $|i+j-l-m| \leq 3$ and $|i+j+k-l-m-n| \leq 3$. The corresponding result for the mass matrix is:

$$\int_{\hat{T}} \phi_{i,j,k}(x,y,z)\,\phi_{l,m,n}(x,y,z)\,d(x,y,z) \neq 0$$

if $|i-l| \in \{0,2\}$, $|i+j-l-m| \leq 3$ and $|i+j+k-l-m-n| \leq 4$.

Figure 6.6 shows the nonzero pattern of these matrices for $p = 10$ and $p = 16$. Observe that for these computations the coefficient matrix $\mathcal{A}$ is not chosen to equal the identity matrix. This explains the additional branches in $\hat{K}_{CC}$ that can be observed in this picture compared to Figure 6.3.

Finally, let us discuss using recurrence relations to determine the nonzero pattern of the system matrix. At first we consider the simplest possible case in two dimensions, namely the matrix K^x with entries

$$\begin{aligned} K^x_{i,j;k,l} &= \int_{\hat{T}} \frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dx}\phi_{k,l}(x,y)\,d(x,y) \\ &= \int_{-1}^{1} p^0_{i-1}(x)p^k_{k-1}(x)\,dx \int_{-1}^{1} w_{i+k-1}(y)\hat{p}^{2i}_j(y)\hat{p}^{2k}_l(y)\,dy. \end{aligned}$$

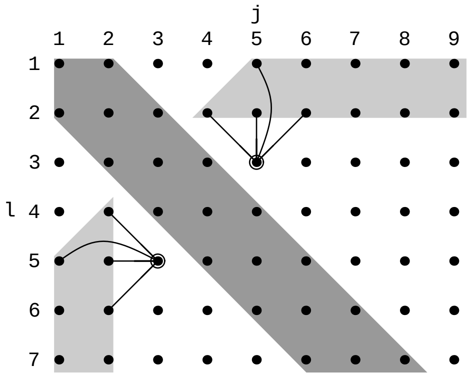


Figure 6.7: Structure of $K^x_{i,j;i,l}$: dark gray: nonzero matrix entries, light gray: initial values for the recurrence relation in the vanishing part

The first integral is easily evaluated using the Legendre orthogonality and we have

$$K^x_{i,j;k,l} = \frac{2}{2i-1}\,\delta_{i,k}\int_{-1}^{1} w_{2i-1}(y)\hat{p}_j^{2i}(y)\hat{p}_l^{2i}(y)\,dy,$$

i.e., $K^x_{i,j;k,l} = 0$ if $i \neq k$. For the remaining integral by Lemma (3.12) there exists a five term recurrence of the form

$$c_1 K^x_{i,j;i,l+1} + c_2 K^x_{i,j+1;i,l} + c_3 K^x_{i,j+1;i,l+1} + c_4 K^x_{i,j+1;i,l+2} + c_5 K^x_{i,j+2;i,l+1} = 0, \tag{6.19}$$

with coefficients c_m rational in i, j and l. This recurrence relation can be generated, e.g., using MultiSum on the summand of the sum representation (4.22) for integrated Jacobi polynomials. Since the matrix $K^x_{i,j;i,l}$ is symmetric in j and l, it is sufficient to consider the upper right triangular matrix. In order to prove that the matrix entries are zero if $|j-l| > 1$, we only need to show that $K^x_{i,1;i,l} = 0$ for $l \geq 3$ and that $K^x_{i,2;i,l} = 0$ for $l \geq 4$. Then it follows from the recurrence relation (6.19) that the remaining entries in the upper right triangular matrix vanish, see Figure 6.7. For computing the initial values of the first two rows, we use again our program and obtain:

$$K^x_{i,1;i,l} = \frac{8(i\delta_{l,2} + (3+2i)\delta_{l,1})}{i(1+i)(-1+2i)(1+2i)(3+2i)}$$

and

$$K^x_{i,2;i,l} = \frac{8(i+2)(2i+5)\delta_{l,1} + 8(2i+1)(2i+5)\delta_{l,2} + 8(i+1)(2i+1)\delta_{l,3}}{(1+i)(2+i)(-1+2i)(1+2i)(3+2i)(5+2i)}.$$

With this we have proven that $K^x_{i,j;k,l} = 0$ if $i \neq k$ and $|i-l| > 1$.

In general, after decoupling the integrals, the system matrix is built from linear combinations of integrands such as

$$\hat{\mathcal{I}}^{2d} = c\,p_{x,1}\,p_{x,2}\,w_1(y)\,p_{y,1}\,p_{y,2}, \quad \text{or,} \quad \hat{\mathcal{I}}^{3d} = c\,p_{x,1}\,p_{x,2}\,w_1(y)\,p_{y,1}\,p_{y,2}\,w_2(z)\,p_{z,1}\,p_{z,2},$$

in the two and three dimensional case, respectively. Each of these univariate products satisfies a product recurrence of the form (3.13) and, by holonomic closure properties, a recurrence relation for the integrand can be determined. The resulting recurrences, however, might be rather big, even for a single integral. For the optimal case $a = b = 0$, combinations of several integrals have to be considered in order to show sparsity of the system matrix. This again increases the order of the resulting recurrences.

Koutschan's package [55] is among the tools that can generate recurrence relations for these integrals and he already obtained first results for the triangular case. Holonomic closure properties include the application of Ore operators, i.e., in our example the derivations D_x and D_y. Hence, annihilating operators for integrands such as $D_x\phi_{i,j}D_y\phi_{k,l}$ can be provided by Koutschan's package, without further rewriting and without decoupling of the integrals. An annihilating ideal obtained for this specific integrand has a size of 1 GB, which also indicates that the generation of a recurrence relation for the entries of the system matrix will be computational expensive, even more in the three dimensional case.

Chapter 7

Positivity of Certain Sums over Jacobi Kernel Polynomials

Whereas in the previous two chapters we were concerned with the construction of high order finite element basis functions, the problems treated in this and the following chapter are different in spirit. In this chapter we show positivity of sums over Jacobi kernel polynomials $k_j^\alpha(x,0)$ on the interval $[-1,1]$ where we consider ultraspherical Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$ with $-\frac{1}{2} \leq \alpha \leq \frac{1}{2}$. This problem originated in a new convergence proof for a certain finite element scheme in the course of which Joachim Schöberl [74] was led to conjecture the inequality

$$\sum_{j=0}^{n}(4j+1)(2n-2j+1)P_{2j}(0)P_{2j}(x) \geq 0 \tag{7.1}$$

for $-1 \leq x \leq 1$ and $n \geq 0$, where $P_n(x)$ denotes the nth Legendre polynomial. This inequality corresponds to setting $\alpha = 0$ in the inequality of Theorem 7.1 that is proven below. No human proof, even for this special case, is known and also asymptotics seem to be difficult [44]. Besides that this problem arose in the context of high order finite element methods, we believe that it is interesting in its own right from a symbolic point of view. We present a proof that is a mixture of traditional reasoning and applications of computer algebra [67].

7.1 Motivation

When constructing a smoothing operator for a high order finite element scheme, Schöberl [74] considered an integral operator that serves as point evaluation when applied to polynomials up to a given degree n. More precisely, he wanted to find a family of polynomials $(\phi_n)_{n\geq 0}$ such that

$$\int_{-1}^{1}\phi_n(x)v(x)\,dx = v(0), \tag{7.2}$$

for all polynomials v with $\deg v \leq n$. Additionally he wanted $(\phi_n)_{n\geq 0}$ to satisfy the norm estimate

$$\|\phi_n\|_{L^1} = \int_{-1}^{1}|\phi_n(x)|\,dx \leq C,$$

where the constant C is independent of n. In Section 3.2 we introduced the kernel polynomial sequence defined as

$$k_n(x,y) = \sum_{j=0}^{n} \frac{1}{h_j} p_j(x) p_j(y), \tag{7.3}$$

for some family of orthogonal polynomials $(p_n(x))_{n\geq 0}$, where $h_n = \int p_n(x)^2 w(x)\,dx$ is the squared, weighted L^2-norm of $p_n(x)$. Recall their reproducing property (3.10)

$$\int k_n(x,y) f(y)\, w(y)\, dy = f(x),$$

for polynomials $f(x)$ with $\deg(f) \leq n$. Let $k_n^\alpha(x,y)$ denote the kernel polynomials for ultraspherical Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$. Then for $\alpha = 0$, i.e., Legendre polynomials, we have

$$\int_{-1}^{1} k_n^0(x,0) v(x)\, dx = v(0), \qquad v \in P^n(-1,1),\ \deg(v) \leq n.$$

Hence a natural candidate for the kernel of the smoothing operator is $\phi_n(x) = k_n^0(x,0)$. But numerical computations suggest that the $k_n^0(x,0)$ are not uniformly bounded in the L^1-norm. So, Schöberl was led to consider a modified ansatz using gliding averages [35],

$$\phi_n(x) = \frac{1}{n+1} \sum_{j=n}^{2n} k_j^0(x,0). \tag{7.4}$$

With this definition $\phi_n(x)$ is a polynomial of degree $2n$ clearly satisfying (7.2). Defining the sum

$$S(n,x) = \frac{1}{n+1} \sum_{j=0}^{n} k_j^0(x,0), \tag{7.5}$$

the polynomials ϕ_n can be rewritten in the form

$$\phi_n(x) = \frac{2n+1}{n+1} S(2n,x) - \frac{n}{n+1} S(n-1,x).$$

Schöberl conjectured that (7.5) is positive for even indices, i.e., $S(2n,x) \geq 0$. If this is true, then the L^1-norm of ϕ_n for odd n can be bounded immediately via

$$\|\phi_n\|_{L^1} \leq \frac{2n+1}{n+1} \int_{-1}^{1} S(2n,x)\, dx + \frac{n}{n+1} \int_{-1}^{1} S(n-1,x)\, dx = \frac{3n+1}{n+1} \leq 3, \quad n \text{ odd}.$$

Observe that for this estimate only the positivity of $S(2n,x)$ and its constant preserving property were needed. After applying the triangle inequality absolute values can be omitted. Since both integrals over $S(2n,x)$ and $S(n-1,x)$ evaluate to 1, the upper bound is established.

Having only an estimate for ϕ_{2n+1} at hand clearly is no obstruction to the application we have in mind since the degree of the smoothing operator can always be raised by one, if needed.

Trying to prove that $S(2n,x) \geq 0$, $x \in [-1,1]$, we observed that this inequality seems to remain valid if we consider more general sums over Jacobi kernel polynomials k_n^α with $\alpha \in [-\frac{1}{2}, \frac{1}{2}]$. Consequently we define

$$S_n^\alpha(x,y) := \sum_{j=0}^{n} k_j^\alpha(x,y).$$

In this notation we have $S(n,x) = (n+1)\, S_n^0(x,0)$. In this chapter we prove the extended conjecture formulated in the following theorem.

Theorem 7.1.

$$S_{2n}^\alpha(x,0) \geq 0 \qquad \textit{for} \qquad -\tfrac{1}{2} \leq \alpha \leq \tfrac{1}{2}, \quad -1 \leq x \leq 1, \quad n \geq 0.$$

Note that for odd degrees, i.e., $S_{2n+1}^\alpha(x,0)$, or for $\alpha \neq -\frac{1}{2}$ and $y \neq 0$, i.e., $S_{2n}^\alpha(x,y)$, the sums are not positive. Because ultraspherical Jacobi polynomials vanish at $x = 0$ for odd degrees and because they are even polynomials for even degrees, see Section 4.1, the sums $S_{2n}^\alpha(x,0)$ are also even polynomials.

7.2 Related Results and Idea of Proof

Let $h_n^\alpha = h_n^{\alpha,\alpha}$ denote the squared, weighted L^2-norm for Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$ with respect to the weight function $w_{\alpha,\alpha}(x) = 2^{-2\alpha}(1-x^2)^\alpha$, i.e.,

$$h_n^\alpha = \int_{-1}^{1} P_n^{(\alpha,\alpha)}(x)^2\, w_{\alpha,\alpha}(x)\, dx = \frac{2}{2n+2\alpha+1} \frac{\Gamma(n+\alpha+1)^2}{n!\,\Gamma(n+2\alpha+1)},$$

see (4.12). Using the definition (7.3) of kernel polynomials, $S_n^\alpha(x,y)$ can be written as the single sum

$$S_n^\alpha(x,y) = \sum_{j=n}^{2n} \sum_{i=0}^{j} \frac{1}{h_i^\alpha} P_i^{(\alpha,\alpha)}(x) P_i^{(\alpha,\alpha)}(y) = \sum_{i=0}^{n} \frac{n-i+1}{h_i^\alpha} P_i^{(\alpha,\alpha)}(x) P_i^{(\alpha,\alpha)}(y).$$

For $\alpha = 0$, n replaced by $2n$ and $y = 0$ this single sum is just the sum in the original formulation of inequality 7.1.

The positivity of trigonometric series as well as their generalizations to Jacobi polynomial series has been considered in many areas of mathematics. One famous example for an inequality of this kind is the Askey-Gasper inequality for the sum $\sum_{k=0}^n P_k^{(\alpha,\beta)}(x)/P_k^{(\beta,\alpha)}(1)$, see [5, 7, 40]. For $\beta = 0$ this sum can be expressed as the square of a hypergeometric function using a formula of Clausen. For $\beta \geq 0$ and $\alpha + \beta > -1$ positivity follows from this result by using a integral representation of Jacobi polynomials. This case also includes Fejér's inequality $\sum_{k=0}^n P_k(x) \geq 0$. Another problem discussed in [7] is determining when the sums

$$\sum_{k=0}^{n} \frac{(\gamma+1)_{n-k}}{(n-k)!} \frac{(2k+\alpha+\beta+1)(\alpha+\beta+1)_k}{k!} \frac{P_k^{(\alpha,\beta)}(x)}{P_k^{(\alpha,\beta)}(1)}$$

are nonnegative for $-1 \leq x \leq 1$. In the ultraspherical case $\alpha = \beta$ with $\gamma = 2\alpha + 3$ nonnegativity can be proven by showing that the generating functions of these sums are products of absolutely monotonic functions, cf. [5] and references therein. None of the techniques mentioned so far, however, are applicable to proving Theorem 7.1, at least not directly.

Gasper [39] has shown that there exists a non negative function $\kappa_n^{\alpha,\beta}(x,y,z)$ such that

$$P_n^{(\alpha,\beta)}(x) P_n^{(\alpha,\beta)}(y) = \int_{-1}^{1} k_n^{\alpha,\beta}(x,y,z) P_n^{(\alpha,\beta)}(z)\, dz.$$

If the $S_n^\alpha(x,y)$ were positive with the factor $P_j^{(\alpha,\alpha)}(y)$ removed, then it would follow that the sum itself is non negative even for all $y \in [-1,1]$. But, as we already remarked above, this is not the case. Although Gasper's result cannot be applied for proving Theorem 7.1, but it is used in the next chapter for giving a short proof of the positivity of another weighted sum over kernel polynomials.

In Section 3.2 the closed form (3.12) for general orthogonal polynomials was given. For Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$, considered as orthogonal polynomials with respect to the weight function $w_{\alpha,\alpha}(x) = 2^{-2\alpha}(1-x^2)^\alpha$, this closed form reads as

$$k_n^\alpha(x,y) = \frac{c_n^\alpha}{x-y}[P_{n+1}^{(\alpha,\alpha)}(x)P_n^{(\alpha,\alpha)}(y) - P_n^{(\alpha,\alpha)}(x)P_{n+1}^{(\alpha,\alpha)}(y)], \tag{7.6}$$

where

$$c_n^\alpha = \frac{1}{2}\frac{\Gamma(n+2)\Gamma(n+2\alpha+2)}{\Gamma(n+\alpha+1)\Gamma(n+\alpha+2)}.$$

Our proof of Theorem 7.1 uses this closed form representation and is split into two parts. In Section 7.3 we consider the cases $\alpha = \pm\frac{1}{2}$, corresponding to the Chebyshev polynomials of the first and second kind, respectively. The proof of these special cases motivates a decomposition of the sum $S_{2n}^\alpha(x,0)$ which is the key to proving Theorem 7.1 for the remaining part where $-\frac{1}{2} < \alpha < \frac{1}{2}$.

7.3 Chebyshev Polynomials of First and Second Kind ($\alpha = \pm\frac{1}{2}$)

In Section 3.2 we identified Jacobi polynomials with $\alpha = \beta = -\frac{1}{2}$ and $\alpha = \beta = \frac{1}{2}$ with Chebyshev polynomials of the first and second kind up to normalization, see (4.16). We follow common notation and use $T_n(x)$ for Chebyshev polynomials of the first kind and $U_n(x)$ for Chebyshev polynomials of the second kind.

In the case of Chebyshev polynomials of the first kind, the sum $S_n^{-1/2}(x,y)$ is called Fejér kernel and positivity is well known for all $n \geq 0$ and for all x, y in the unit square $[-1,1]^2$, for a short proof see e.g. [84]. Hence we only have to consider the case $\alpha = \frac{1}{2}$. The kernel polynomials for $\alpha = \frac{1}{2}$ expressed in terms of Chebyshev polynomials read as

$$k_n^{1/2}(x,y) = \frac{1}{\pi(x-y)}[U_{n+1}(x)U_n(y) - U_n(x)U_{n+1}(y)].$$

SumCracker yields a closed form for $S_n^{1/2}(x,y)$, namely,

$$S_n^{1/2}(x,y) = \frac{1}{\pi(x-y)^2}[U_{n+1}(x)(xU_n(y) - U_{n+1}(y)) + U_n(x)(yU_{n+1}(y) - U_n(y)) + 1]. \tag{7.7}$$

To prove that $S_{2n}^{1/2}(x,0) \geq 0$ we proceed as follows. Since $U_{2n+1}(0) = 0$ and $U_{2n}(0) = (-1)^n$ we have that

$$S_{2n}^{1/2}(x,0) = \frac{1}{\pi x^2}[1 + (-1)^n x\, U_{2n+1}(x) - (-1)^n U_{2n}(x)].$$

Inspection of the first few polynomials $S_{2n}^{1/2}(x,0)$ suggests that

$$S_{4m}^{1/2}(x,0) = p_{2m}(x)^2 \qquad \text{and} \qquad S_{4m+2}^{1/2}(x,0) = (1-x^2)q_{2m}(x)^2,$$

where $p_{2m}(x)$, $q_{2m}(x)$ are polynomials of degree $2m$ satisfying the relation

$$q_n(x)S_1^{1/2}(x,0) = (p_{n+1}(x) - p_n(x))^2.$$

This claim can be verified by first using GuessRE to guess a recurrence relation for $p_n(x)$. The resulting recurrence can then easily be identified as the three term recurrence for Chebyshev polynomials of the first kind (given the initial values).

Lemma 7.2. *For $m \geq 0$ and $-1 \leq x \leq 1$ we have*

$$S_{4m}^{1/2}(x,0) = \frac{2}{\pi x^2} T_{2m+1}(x)^2,$$

and

$$S_{4m+2}^{1/2}(x,0) = \frac{1}{2\pi x^2(1-x^2)}(T_{2m+3}(x) - T_{2m+1}(x))^2,$$

where $T_m(x)$ are the Chebyshev polynomials of the first kind.

Proof. The closed forms for $S_{4m}^{1/2}(x,0)$ and $S_{4m+2}^{1/2}(x,0)$ can be verified immediately using SumCracker's ZeroSequenceQ command.

```
In[31]:= ZeroSequenceQ[xChebyshevU[4m + 1, x] − ChebyshevU[4m, x] + 1
            − 2ChebyshevT[2m + 1, x]^2]

Out[31]= True

In[32]:= ZeroSequenceQ[−xChebyshevU[4m + 3, x] + ChebyshevU[4m + 2, x] + 1
            − (ChebyshevT[2m + 3, x] − ChebyshevT[2m + 1, x])^2/(2(1 − x^2))]

Out[32]= True
```

□

From these representations it is obvious that the sums $S_{2n}^{1/2}(x,0)$ are nonnegative. While there exists a closed form representation of $S_n^{1/2}(x,y)$, there is no closed form of $S_n^{\alpha}(x,y)$ for general α. Still, examining a derivation of (7.7) using only the three term recurrence satisfied by $U_n(x)$ indicates how to continue dealing with general Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$, $-\frac{1}{2} < \alpha < \frac{1}{2}$.

So, let again $\alpha = \frac{1}{2}$. In order to derive (7.7), we show that $S_n^{1/2}(x,y)$ rewritten according to (7.6) as the sum

$$S_n^{1/2}(x,y) = \frac{1}{\pi(x-y)} \sum_{j=0}^{n} [U_{j+1}(x)U_j(y) - U_j(x)U_{j+1}(y)],$$

is a sum representation which telescopes to the right hand side of (7.7). Because of symmetry it suffices to consider only one part of the sum. For the first part, SumCracker yields

$$(x-y)\sum_{j=0}^{n} U_{j+1}(x)U_j(y) = \tfrac{1}{2}\left(2xU_{n+1}(x)U_n(y) - U_n(x)U_n(y) - U_{n+1}(x)U_{n+1}(y) + 1\right),$$

which implies

$$(x-y)U_{j+1}(x)U_j(y) = \tfrac{1}{2}\Delta_j(2xU_j(x)U_{j-1}(y) - U_{j-1}(x)U_{j-1}(y) - U_j(x)U_j(y)) =: \tfrac{1}{2}\Delta_j G_j(x,y),$$

where Δ_j denotes the difference operator $\Delta_j[\psi(j)] = \psi(j+1) - \psi(j)$. The correctness of this identity can be verified by straight-forward calculation using the three term recurrence for Chebyshev polynomials,

$$U_n(x) - 2xU_{n+1}(x) + U_{n+2}(x) = 0, \qquad U_0(x) = 1,\ U_1(x) = 2x. \tag{7.8}$$

Namely, first we use (7.8) to rewrite $2x\,U_j(x)$ and then, to involve y, we use the same recurrence relation to replace $U_{j-1}(y) + U_{j+1}(y)$. This way we obtain

$$\begin{aligned} G_{j+1}(x,y) - G_j(x,y) &= 2xU_j(y)U_{j+1}(x) - U_{j+1}(x)U_{j+1}(y) \\ &\quad - 2xU_{j-1}(y)U_j(x) + U_{j-1}(x)U_{j-1}(y) \\ &= 2xU_j(y)U_{j+1}(x) - U_{j+1}(x)U_{j+1}(y) - U_{j-1}(y)U_{j+1}(x) \\ &= 2(x-y)U_{j+1}(x)U_j(y). \end{aligned} \tag{7.9}$$

Note that this telescoper only exists because Chebyshev polynomials satisfy a three term recurrence with constant coefficients. The procedure above cannot be generalized to Jacobi polynomials $P_n^{(\alpha,\alpha)}(x)$, $\alpha \neq \pm\frac{1}{2}$, because the polynomial recurrence coefficients do not lead to cancellation. However mimicking the steps of the proof above one obtains a decomposition of $S_{2n}^{\alpha}(x,0)$, $-\frac{1}{2} < \alpha < \frac{1}{2}$, that makes the problem better treatable with our methods.

Since Chebyshev polynomials of the first and second kind satisfy the same recurrence but with different starting values, a closed form for $S_n^{-1/2}(x,y)$ can be computed completely analogously. After minor rewriting one obtains

$$S_n^{-1/2}(x,y) = \frac{1}{\pi(x-y)^2}\left[1 - xy + \tfrac{1}{2}T_{n+2}(x)T_n(y) - T_{n+1}(x)T_{n+1}(y) + \tfrac{1}{2}T_n(x)T_{n+2}(y)\right].$$

7.4 Jacobi Polynomials $P_n^{(\alpha,\alpha)}(x)$ with $-\frac{1}{2} < \alpha < \frac{1}{2}$

In this section we prove Theorem 7.1, i.e., the positivity of $S_{2n}^{\alpha}(x,0)$, $-\frac{1}{2} < \alpha < \frac{1}{2}$, where the sum representation according to (7.6) is given by

$$S_n^{\alpha}(x,y) = \frac{1}{x-y}\sum_{j=0}^{n} c_j^{\alpha}[P_{j+1}^{(\alpha,\alpha)}(x)P_j^{(\alpha,\alpha)}(y) - P_j^{(\alpha,\alpha)}(x)P_{j+1}^{(\alpha,\alpha)}(y)], \tag{7.10}$$

with $c_j^{\alpha} = \frac{1}{2}\frac{\Gamma(j+2)\Gamma(j+2\alpha+2)}{\Gamma(j+\alpha+1)\Gamma(j+\alpha+2)}$. To this end we need several intermediate results starting with a suitable decomposition of $S_n^{\alpha}(x,y)$ which is obtained by following the steps of the derivation (7.9). For this we need the Jacobi three term recurrence. Setting $\alpha = \beta$ in (4.14) yields

$$\begin{aligned} (n+2)(n+2\alpha+2)P_{n+2}^{(\alpha,\alpha)}(x) &= (n+\alpha+2)(2n+2\alpha+3)\,x\,P_{n+1}^{(\alpha,\alpha)}(x) \\ &\quad - (n+\alpha+1)(n+\alpha+2)P_n^{(\alpha,\alpha)}(x) \end{aligned} \tag{7.11}$$

for $n \geq 0$ and the initial values $P_{-1}^{(\alpha,\alpha)}(x) = 0,\ P_0^{(\alpha,\alpha)}(x) = 1$. With this relation we obtain for all $j \geq 0$

$$\begin{aligned}
&(x-y)c_j^\alpha P_{j+1}^{(\alpha,\alpha)}(x)P_j^{(\alpha,\alpha)}(y)\\
&= x\, c_j^\alpha P_{j+1}^{(\alpha,\alpha)}(x)P_j^{(\alpha,\alpha)}(y) - \frac{c_j^\alpha}{(j+\alpha+1)(2j+2\alpha+1)}P_{j+1}^{(\alpha,\alpha)}(x)\\
&\quad\times[(j+\alpha)(j+\alpha+1)P_{j-1}^{(\alpha,\alpha)}(y) + (j+1)(j+2\alpha+1)P_{j+1}^{(\alpha,\alpha)}(y)]\\
&= x\, c_j^\alpha P_{j+1}^{(\alpha,\alpha)}(x)P_j^{(\alpha,\alpha)}(y) - c_j^\alpha\frac{(j+1)(j+2\alpha+1)}{(j+\alpha+1)(2j+2\alpha+1)}P_{j+1}^{(\alpha,\alpha)}(x)P_{j+1}^{(\alpha,\alpha)}(y)\\
&\quad - c_j^\alpha\frac{(j+\alpha)(j+\alpha+1)}{(2j+2\alpha+1)(j+1)(j+2\alpha+1)}P_{j-1}^{(\alpha,\alpha)}(y)\\
&\quad\times[x(2j+2\alpha+1)P_j^{(\alpha,\alpha)}(x) - (j+\alpha)P_{j-1}^{(\alpha,\alpha)}(x)]\\
&= xc_j^\alpha P_{j+1}^{(\alpha,\alpha)}(x)P_j^{(\alpha,\alpha)}(y) - xc_{j-1}^\alpha P_j^{(\alpha,\alpha)}(x)P_{j-1}^{(\alpha,\alpha)}(y)\\
&\quad - c_j^\alpha\frac{(j+1)(j+2\alpha+1)}{(j+\alpha+1)(2j+2\alpha+1)}P_{j+1}^{(\alpha,\alpha)}(x)P_{j+1}^{(\alpha,\alpha)}(y)\\
&\quad + c_j^\alpha\frac{(j+\alpha)^2(j+\alpha+1)}{(j+1)(j+2\alpha+1)(2j+2\alpha+1)}P_{j-1}^{(\alpha,\alpha)}(x)P_{j-1}^{(\alpha,\alpha)}(y).
\end{aligned}$$

Now we plug this identity into Definition (7.10), set $y = 0$ and substitute $n \mapsto 2n$. This gives

$$\begin{aligned}
x^2 S_{2n}^\alpha(x,0) = &\sum_{j=0}^{2n} x\Delta_j[c_{j-1}^\alpha P_j^{(\alpha,\alpha)}(x)P_{j-1}^{(\alpha,\alpha)}(0)]\\
&- 2\sum_{j=0}^{2n} c_j^\alpha\frac{(j+1)(j+2\alpha+1)}{(j+\alpha+1)(2j+2\alpha+1)}P_{j+1}^{(\alpha,\alpha)}(x)P_{j+1}^{(\alpha,\alpha)}(0)\\
&+ 2\sum_{j=0}^{2n} c_j^\alpha\frac{(j+\alpha)^2(j+\alpha+1)}{(j+1)(j+2\alpha+1)(2j+2\alpha+1)}P_{j-1}^{(\alpha,\alpha)}(x)P_{j-1}^{(\alpha,\alpha)}(0),
\end{aligned}$$

The first sum can easily be simplified by telescoping, the second and third sum can be combined by shifting summation indices. We also use the fact that ultraspherical Jacobi polynomials $P_n^{(\alpha,\alpha)}$ of odd degree vanish at $x = 0$. Thus with

$$g_{2n}^\alpha(x,0) = c_{2n}^\alpha\left[x\, P_{2n+1}^{(\alpha,\alpha)}(x) - 2\frac{2n+\alpha+1}{4n+2\alpha+3}P_{2n}^{(\alpha,\alpha)}(x)\right]P_{2n}^{(\alpha,\alpha)}(0)$$

and

$$f_{2n}^\alpha(x,0) = 2(4\alpha^2-1)\sum_{j=0}^{n}\frac{(2j+\alpha+1)c_{2j}^\alpha}{(2j+1)(2j+2\alpha+1)(4j+2\alpha-1)(4j+2\alpha+3)}P_{2j}^{(\alpha,\alpha)}(0)P_{2j}^{(\alpha,\alpha)}(x)$$

we obtain a decomposition of the sum $S_{2n}^\alpha(x,0)$. Note that for Chebyshev polynomials, i.e., $\alpha = \pm\frac{1}{2}$, $f_{2n}^\alpha(x,0)$ collapses to 0 because of the factor $(4\alpha^2-1)$. Only the closed form $g_{2n}^\alpha(x,0)$ survives.

Lemma 7.3.

$$x^2 S_{2n}^\alpha(x,0) = f_{2n}^\alpha(x,0) + g_{2n}^\alpha(x,0), \qquad -\tfrac{1}{2} < \alpha < \tfrac{1}{2},\ -1 \leq x \leq 1,\ n \geq 0.$$

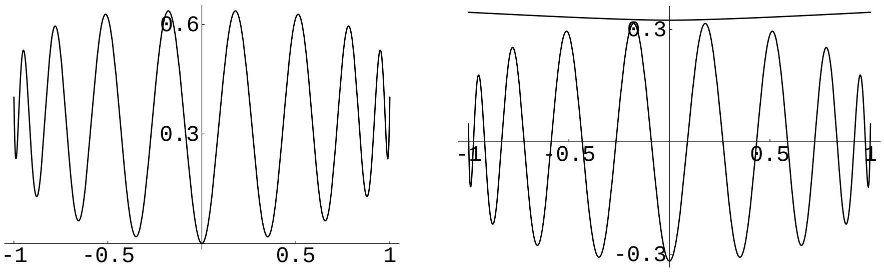


Figure 7.1: left: $x^2 S_{2n}^0(x,0)$ and right: $f_{2n}^0(x,0)$ (red), $g_{2n}^0(x,0)$ (blue) for $n = 8$

As can be seen from Figure 7.1, $g_{2n}^\alpha(x,0)$ contains the main oscillations whereas in $f_{2n}^\alpha(x)$ they are dampened out. In order to prove nonnegativity of $S_{2n}^\alpha(x,0)$ we show that $f_{2n}^\alpha(x,0) + g_{2n}^\alpha(x,0) \geq 0$. This is achieved by estimating the sum $f_{2n}^\alpha(x,0)$ from below. The sum of this lower bound and $g_{2n}^\alpha(x,0)$ can then be shown to be positive with SumCracker's ProveInequality command.

The first step is to define, more generally, f_n^α for arguments $x, y \in [-1,1]$ by

$$f_n^\alpha(x,y) = 2(4\alpha^2 - 1) \sum_{j=0}^{n} \frac{(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P_j^{(\alpha,\alpha)}(x) P_j^{(\alpha,\alpha)}(y).$$

This definition is consistent with that of $f_{2n}^\alpha(x,0)$ above. The coefficient of the Jacobi polynomials inside the sum is positive for $j \geq 1$, hence we have

$$\sum_{j=1}^{n} \frac{(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} [P_j^{(\alpha,\alpha)}(x) - P_j^{(\alpha,\alpha)}(y)]^2 \geq 0,$$

which is equivalent to

$$\begin{aligned} -\sum_{j=0}^{n} \frac{2(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P_j^{(\alpha,\alpha)}(x) P_j^{(\alpha,\alpha)}(y) \geq \\ -\sum_{j=0}^{n} \frac{(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P_j^{(\alpha,\alpha)}(x)^2 \\ -\sum_{j=0}^{n} \frac{(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P_j^{(\alpha,\alpha)}(y)^2 \end{aligned}$$

Since $(1-2\alpha)(1+2\alpha)$ is positive for $-\frac{1}{2} < \alpha < \frac{1}{2}$, both sides of the last inequality can be multiplied with this factor to obtain

Lemma 7.4. *Let* $-\frac{1}{2} < \alpha < \frac{1}{2}$. *Then*

$$f_n^\alpha(x,y) \geq \tfrac{1}{2}(f_n^\alpha(x,x) + f_n^\alpha(y,y)), \qquad n \geq 0,$$

for all $x, y \in [-1,1]$.

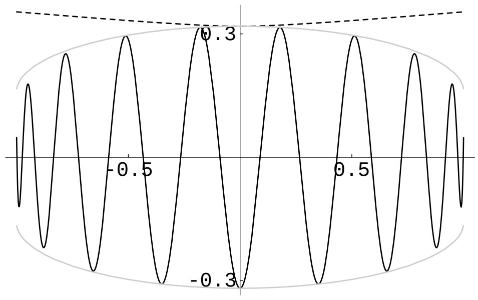


Figure 7.2: solid: $g^0_{2n}(x,0)$, dashed: $f^0_{2n}(x,0)$, gray: $\pm\frac{1}{2}(f^0_{2n}(x,x)+f^0_{2n}(0,0))$

This lower bound has the advantage that we can find a closed form for $f_n^\alpha(x,x)$. Although Kauers' package SumCracker does not find a closed form of $f_n^\alpha(x,x)$ for symbolic α, for specific values of α it succeeds. Guessing on the coefficients of these expressions suggests the closed form stated in the next lemma. The key point, however, is discovering this identity. Once it has been found its validity can be proven fairly easily.

Lemma 7.5.

$$f_n^\alpha(x,x) = 2c_n^\alpha \left[\frac{(n+1)(n+2\alpha+1)}{(n+\alpha+1)(2n+2\alpha+1)} P_{n+1}^{(\alpha,\alpha)}(x)^2 \right.$$
$$\left. -x\, P_n^{(\alpha,\alpha)}(x) P_{n+1}^{(\alpha,\alpha)}(x) + \frac{n+\alpha+1}{2n+2\alpha+3} P_n^{(\alpha,\alpha)}(x)^2 \right],$$

for all $n \geq 0$, $-1 \leq x \leq 1$ *and* $\alpha > -1$. *For* $n=-1$ *we have* $f_{-1}^\alpha(x,x)=0$.

Proof. We prove the identity using ZeroSequenceQ. The coefficients c_n^α are given by their recurrence relation `cdef`.

In[33]:= $$\mathbf{cdef = \{c[k] == \frac{(k+1)(k+2\alpha+1)}{(k+\alpha)(k+\alpha+1)} c[k-1], c[0] == \frac{2^{-2\alpha-1}Gamma[2\alpha+2]}{Gamma[\alpha+1]Gamma[\alpha+2]}\};}$$

In[34]:= $$\mathbf{ZeroSequenceQ[(2\alpha-1)(2\alpha+1)}$$
$$\mathbf{SUM[\frac{(j+\alpha+1)c[j]}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} JacobiP[j,\alpha,\alpha,x]^2, \{j,0,n\}]}$$
$$\mathbf{-\, c[n](\frac{(n+1)(n+2\alpha+1)}{(n+\alpha+1)(2n+2\alpha+1)} JacobiP[n+1,\alpha,\alpha,x]^2}$$
$$\mathbf{-\, x JacobiP[n,\alpha,\alpha,x] JacobiP[n+1,\alpha,\alpha,x] + \frac{n+\alpha+1}{2n+2\alpha+3} JacobiP[n,\alpha,\alpha,x]^2),}$$
$$\mathbf{Where \to cdef]}$$

Out[34]= True

□

Figure 7.2 illustrates how the functions $g_{2n}^\alpha(x,0)$, $f_{2n}^\alpha(x,0)$ and $\frac{1}{2}(f_{2n}^\alpha(x,x)+f_{2n}^\alpha(0,0))$ are related. Lemma 7.5 can also be proven by showing that the closed form is the telescoper for the summand using only the Jacobi three term recurrence. By telescoping and since $f_{-1}^\alpha(x,x)=0$, it suffices to show that

$$\frac{2(4\alpha^2-1)(j+\alpha+1)c_j^\alpha}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P_j^{(\alpha,\alpha)}(x)^2 = f_j^\alpha(x,x) - f_{j-1}^\alpha(x,x).$$

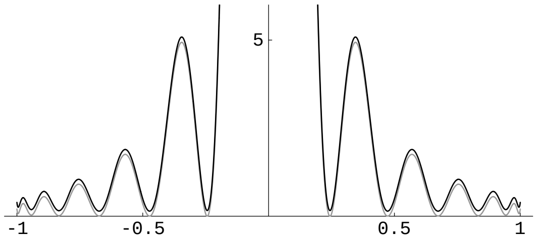


Figure 7.3: gray: $[g^0_{2n}(x,0) + \frac{1}{2}(f^0_{2n}(x,x) + f^0_{2n}(0,0))]/x^2$, black: $S^0_{2n}(x,0)$ for $n = 12$.

Since $c^\alpha_{j-1} = \frac{(j+\alpha)(j+\alpha+1)}{(j+1)(j+2\alpha+1)} c^\alpha_j$ we have

$$\begin{aligned}\frac{1}{2c^\alpha_j}\left(f^\alpha_j(x,x) - f^\alpha_{j-1}(x,x)\right) &= \frac{(4\alpha^2-1)(j+\alpha+1)}{(j+1)(j+2\alpha+1)(2j+2\alpha-1)(2j+2\alpha+3)} P^{(\alpha,\alpha)}_j(x)^2 \\ &+ \left[\frac{(j+1)(j+2\alpha+1)}{(j+\alpha+1)(2j+2\alpha+1)} P^{(\alpha,\alpha)}_{j+1}(x) - xP^{(\alpha,\alpha)}_j(x)\right] P^{(\alpha,\alpha)}_{j+1}(x) \\ &+ \frac{(j+\alpha)(j+\alpha+1)}{(j+1)(j+2\alpha+1)} \left[xP^{(\alpha,\alpha)}_j(x) - \frac{j+\alpha}{2j+2\alpha+1} P^{(\alpha,\alpha)}_{j-1}(x)\right] P^{(\alpha,\alpha)}_{j-1}(x).\end{aligned}$$

By the Jacobi recurrence relation (7.11) the expressions in the last two rows cancel.

Now we collect the previous lemmas to give a proof of Theorem 7.1.

Proof of Theorem 7.1. The cases $\alpha = \pm\frac{1}{2}$ are covered by the results of section 7.3. For $\alpha = -\frac{1}{2}$ Theorem 7.1 follows from well known results on the Fejèr kernel [84] and positivity of $S^{1/2}_{2n}(x,0)$ is obvious from the rewriting stated in Lemma 7.2.

Next we consider $-\frac{1}{2} < \alpha < \frac{1}{2}$. With the decomposition given in Lemma 7.3 and the lower bound from Lemma 7.4 we have

$$x^2 S^\alpha_{2n}(x,0) = g^\alpha_{2n}(x,0) + f^\alpha_{2n}(x,0) \geq g^\alpha_{2n}(x,0) + \tfrac{1}{2}(f^\alpha_{2n}(x,x) + f^\alpha_{2n}(0,0)).$$

To complete the proof it suffices to show positivity of the latter expression. Using Lemma 7.5 we have

$$\begin{aligned}&\frac{1}{c^\alpha_{2n}}[g^\alpha_{2n}(x,0) + \tfrac{1}{2}(f^\alpha_{2n}(x,x) + f^\alpha_{2n}(0,0))] \\ =\ &\frac{(2n+1)(2n+2\alpha+1)}{(2n+\alpha+1)(4n+2\alpha+1)} P^{(\alpha,\alpha)}_{2n+1}(x)^2 - x\, P^{(\alpha,\alpha)}_{2n+1}(x)[P^{(\alpha,\alpha)}_{2n}(x) - P^{(\alpha,\alpha)}_{2n}(0)] \\ &+ \frac{2n+\alpha+1}{4n+2\alpha+3}[P^{(\alpha,\alpha)}_{2n}(x) - P^{(\alpha,\alpha)}_{2n}(0)]^2.\end{aligned} \tag{7.12}$$

We use the ProveInequality command of SumCracker in the following way:

In[35]:= $\mathbf{ProveInequality[\frac{(2n+1)(2n+2\alpha+1)}{(2n+\alpha+1)(4n+2\alpha+1)}JacobiP[2n+1,\alpha,\alpha,x]^2}$
$\mathbf{- xJacobiP[2n+1,\alpha,\alpha,x](JacobiP[2n,\alpha,\alpha,x] - JacobiP[2n,\alpha,\alpha,0])}$
$\mathbf{+ \frac{2n+\alpha+1}{4n+2\alpha+3}(JacobiP[2n,\alpha,\alpha,x] - JacobiP[2n,\alpha,\alpha,0])^2 \geq 0,}$
$\mathbf{Using \to \{-1 \leq x \leq 1,\ -\frac{1}{2} < \alpha < \frac{1}{2}\}, Variable \to n, From \to 0]//Timing}$

Out[35]= {5358.25Second, True} □

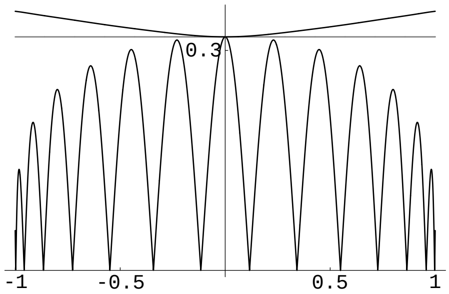


Figure 7.4: $|g_{2n}^0(x,0)|$, $|g_{2n}^0(0,0)| = f_{2n}^0(0,0)$ and $f_{2n}^0(x,0)$ for $n = 6$

The main computational effort in the execution of the ProveInequality lies in the cylindrical decomposition. The condition on α above cannot be removed if we want positivity of (7.12) for $n \geq 0$. It seems though that this expression stays nonnegative for n greater than some lower bound, possibly depending on α.

7.5 Envelope for the Oscillating Part

Taking a closer look at Figure 7.1 it can be noticed that the absolute value of the oscillating part $|g_{2n}^0(x,0)|$ and the sum $f_{2n}^0(x,0)$ are separated by a straight line passing through $f_{2n}^0(0,0) = |g_{2n}^0(0,0)|$, see Figure 7.4. Guided by this observation one idea for proving Theorem 7.1 was to show that $f_{2n}^0(x,0)$ and $|g_{2n}^0(x,0)|$ are bounded from below and above, respectively, by this straight line. The idea for showing boundedness of the oscillating part followed the proof of Theorem 7.3.1 in [79] stating that the relative maxima of Legendre polynomials form a decreasing sequence when x decreases from 1 to 0. In [79] Szegö proves this theorem by computing an envelope for $P_n(x)^2$ and showing its monotonicity. We repeat this result in Theorem 7.6 below. Although this idea did not work out to show positivity of $S_{2n}^0(x,0)$, we believe that the construction of the envelope of $g_{2n}^\alpha(x,0)^2$ is interesting in its own right.

Theorem 7.6. *[79, Theorem 7.3.1] Let $n \geq 2$. The successive relative maxima of $|P_n(x)|$, when x decreases from 1 to 0, form a decreasing sequence. More precisely, if $\mu_1, \mu_2, \ldots, \mu_{[n/2]}$ denote these maxima corresponding to decreasing values of x, we have*

$$1 > \mu_1 > \mu_2 > \ldots > \mu_{[n/2]}.$$

Proof. This result can be proven by means of the differential equation for Legendre polynomials. Setting $\alpha = \beta = 0$ in (4.2) yields the Legendre differential equation

$$(1 - x^2)y''(x) - 2x\,y'(x) + n(n+1)y(x) = 0.$$

For the proof let $h_n(x)$ be defined via

$$n(n+1)h_n(x) = n(n+1)P_n(x)^2 + (1 - x^2)P_n'(x)^2.$$

Then we have $h_n(x) = P_n(x)^2$ if $P_n'(x) = 0$ or $x = \pm 1$. Therefore

$$\max_{-1 \leq x \leq 1} P_n(x)^2 \leq \max_{-1 \leq x \leq 1} h_n(x).$$

Then for the first derivative of $h_n(x)$, using the Legendre differential equation, we have

$$\begin{aligned} n(n+1)h_n'(x) &= 2n(n+1)P_n(x)P_n'(x) - 2xP_n'(x)^2 + 2(1-x^2)P_n'(x)P_n''(x) \\ &= 2\big[n(n+1)P_n(x) - xP_n'(x) + (1-x^2)P_n''(x)\big]P_n'(x) \\ &= 2xP_n'(x)^2. \end{aligned}$$

Hence, $h_n(x)$ is decreasing for $x < 0$ and increasing for $x > 0$. This establishes the statement. □

As a corollary to this theorem it follows that Legendre polynomials attain their maxima at $x = \pm 1$, i.e., $|P_n(x)| \leq |P_n(\pm 1)| = 1$.

Now we apply this procedure for computing an envelope for the oscillating part $g_{2n}^0(x, 0)$. In a first step we consider only the case of Legendre polynomials, i.e., $\alpha = 0$. For computing an envelope it is sufficient to work with the "essential" part of $g_{2n}^0(x, 0)$ which we define as

$$g_n(x) = (4n+3)xP_{2n+1}(x) - 2(2n+1)P_{2n}(x),$$

i.e., we have $(4n+3)g_{2n}^0(x, 0) = c_{2n}^0 P_{2n}(0)g_n(x)$. Now we make an ansatz for the envelope $h_n(x) = h_n^0(x)$ as a generalized version of the envelope in Theorem 7.6,

$$h_n(x) = (a_0 + a_2x^2)g_n(x)^2 + (1-x^2)(b_0 + b_2x^2)g_n'(x)^2. \tag{7.13}$$

Moreover, we also want the derivatives of $h_n(x)$ to be of a special form, hence we also make an ansatz for $h_n'(x)$,

$$h_n^{der}(x) = c_1xg_n(x)^2 + x(d_0 + d_2x^2)g_n'(x)^2. \tag{7.14}$$

Additionally we normalize subject to $h_n(0) = g_n(0)^2$. This refined ansatz for both h_n and h_n^{der} was found after trying a more general ansatz first and observing this special pattern. For specific values of n the coefficients a_0 to d_2 are determined by coefficient comparison of $h_n'(x)$ and $h_n^{der}(x)$. These concrete values are then used as input for Mallinger's GuessRE routine to find recurrence relations for the coefficients. The resulting recurrences can be solved for instance using the Mathematica built-in RSolve command.

Let us consider a concrete example. The solutions for the coefficients $b_0 = b_0(n)$ for $n = 2, \ldots, 10$ are given by

$$S_{b_0} = \left\{\tfrac{1}{31}, \tfrac{1}{57}, \tfrac{1}{91}, \tfrac{1}{133}, \tfrac{1}{183}, \tfrac{1}{241}, \tfrac{1}{307}, \tfrac{1}{381}, \tfrac{1}{463}\right\}.$$

Guessing on this list yields:

In[36]:= $\mathbf{GuessRE}[S_{b_0}, \mathbf{b0}[n]]$

Out[36]= $\{\{(-4n^2 - 22n - 31)\mathrm{b0}[n] + (4n^2 + 30n + 57)\mathrm{b0}[n+1] = 0, \mathrm{b0}[0] = \frac{1}{31}\}, \mathrm{ogf}\}$

Using the RSolve command and shifting the index n properly yields the result stated next. Because of the normalization $h_n(0) = g_n(0)^2$, one has $a_0(n) = 1$. With $\gamma(n) = 4n^2 + 6n + 3$ the remaining coefficients are

$$\begin{aligned} a_2(n) &= -\frac{(4n+3)^2(2n+1)(n+1)}{\gamma(n)(2\gamma(n)-1)}, & c_1(n) &= 2a_2(n), \\ b_0(n) &= \frac{1}{\gamma(n)}, & d_0(n) &= 2b_0(n) + 6b_2(n), \\ b_2(n) &= -\frac{(4n+3)^2}{2\gamma(n)(2\gamma(n)-1)}, & d_2(n) &= -4b_2(n). \end{aligned}$$

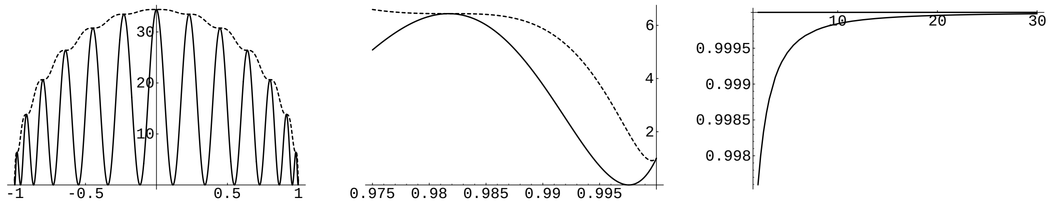


Figure 7.5: $g_n(x)^2$ and envelope $G_n(x)$ for $n = 6$, $\rho(n)$

There are two feasible ways how to obtain this result with GuessRE. Firstly by observing the special form of the denominators involving $\gamma(n)$ and simplifying the input by clearing denominators appropriately. Then guessing is applied to determine the structure of the numerators only. The second way avoids these manipulations and uses GuessRE on the input values directly. In this case the degree bound for the recurrence coefficients has to be increased, see Section 3.3.1, and also the number of input values has to be sufficiently big to obtain an overdetermined equation system.

Next we show that the derivative of $h_n(x)$ coincides with $h_n^{der}(x)$. Differentiating (7.13) and invoking the relations between the coefficients of h_n and h_n^{der} yields

$$\begin{aligned} h_n'(x) &= c_1 x g_n^2(x) + x(d_0 + d_2 x^2) g_n'(x)^2 \\ &\quad + 2(1 + a_2 x^2) g_n(x) g_n'(x) - 4x(b_0 + b_2) g_n'(x)^2 + 2(1 - x^2)(b_0 + b_2 x^2) g_n'(x) g_n''(x). \end{aligned}$$

What is left to be shown is that the second line above vanishes for all $n \geq 2$. Using $\gamma(n)(2\gamma(n) - 1)(b_0(n) + b_2(n)) = 1$ and clearing common factors, this task can be simplified further to showing that $g_n(x)$ satisfies the differential equation

$$(1 + a_2 x^2) g_n(x) - \frac{x}{\gamma(n)(2\gamma(n) - 1)} g_n'(x) + (1 - x^2)(b_0 + b_2 x^2) g_n''(x) = 0. \tag{7.15}$$

But this identity can be verified automatically using SumCrackers's ZeroSequenceQ command. The envelope $G_n(x)$ for $g_n(x)^2$ is consequently defined as

$$G_n(x) = \frac{1}{1 + a_2 x^2} h_n(x),$$

where the denominator $1 + a_2(n)x^2$ is strictly negative on $[-1, 1]$ and its roots are approaching ± 1 from outside the interval in the limit $n \to \infty$. By construction we have that $G_n(x)$ coincides with $g_n(x)^2$ at $x = \pm 1$ and at its extremal points, i.e., $G_n(\pm 1) = g_n(\pm 1)^2$, $G_n(0) = g_n(0)^2$ and $G_n(\bar{x}) = g_n(\bar{x})^2$ for all $\bar{x}$ such that $g_n'(\bar{x}) = 0$.

The next step is showing that the envelope decreases when x passes from 0 to 1. For this purpose we compute its first derivative using the definition of $h_n(x)$ and the special form of $h_n'(x)$, yielding

$$(1 + a_2 x^2)^2 G_n'(x) = x\, g_n'(x)^2 \big(-2a_2(1 - x^2)(b_0 + b_2 x^2) + (1 + a_2 x^2)(d_0 + d_2 x^2)\big).$$

The polynomial on the right hand side is negative for $x \in [-\rho(n), \rho(n)]$, where

$$\rho(n) = \left(\frac{\left(4n^2 + 5n + 2\right)\left(8n^2 + 14n + 7\right) - \sqrt{84n^4 + 252n^3 + 291n^2 + 153n + 31}}{(n+1)(2n+1)(4n+3)^2} \right)^{1/2}.$$

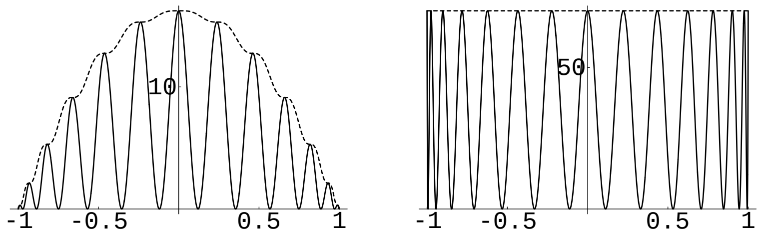


Figure 7.6: $g_n^\alpha(x)^2$ and envelope $G_n^\alpha(x)$ for $\alpha = -\frac{1}{2}$ (left) and $\alpha = \frac{1}{2}$ (right) for $n = 6$

The other roots are not in the interval $[-1, 1]$. This completes the analysis of the envelope and we have that $G_n(x)$ decreases for x passing from 0 to $\rho(n)$, hence the maxima of $g_n(x)^2$ decrease in this interval as well. The middle picture in Figure 7.5 shows the behavior of the envelope at $x = \rho(n)$. The value of $\rho(n)$ approaches 1 as $n \to \infty$. The rightmost picture in Figure 7.5 shows the first few values of $\rho(n)$, indicating its rate of convergence.

Following the same procedure as in the special case $\alpha = 0$ an envelope for arbitrary $\alpha > -1$ for $g_n^\alpha(x)$ can be computed, where

$$g_n^\alpha(x) = (4n + 2\alpha + 3)xP_{2n+1}^{(\alpha,\alpha)}(x) - 2(2n + \alpha + 1)P_{2n}^{(\alpha,\alpha)}(x).$$

The ansatz for $h_n^\alpha(x)$ is completely analogous as in the case $\alpha = 0$, only the coefficients now involve the parameter α. By coefficient comparison and subsequent guessing one obtains with $\gamma^\alpha(n) = 4n^2 + 2(2\alpha + 3)n + (2\alpha + 3)$ that

$$\begin{aligned} a_2^\alpha(n) &= 2(n+1)(2n+2\alpha+1)b_2^\alpha(n), & c_1^\alpha(n) &= 2a_2^\alpha(n), \\ b_0^\alpha(n) &= \frac{1}{\gamma^\alpha(n)}, & d_0^\alpha(n) &= 2(2\alpha+1)b_0^\alpha(n) + 6b_2^\alpha(n), \\ b_2^\alpha(n) &= -\frac{(4n+2\alpha+3)^2}{2\gamma^\alpha(n)(2\gamma\alpha(n) - 1 + 2\alpha)}, & d_2^\alpha(n) &= 4(\alpha-1)b_2^\alpha(n). \end{aligned}$$

Because of the normalization $h_n^\alpha(0) = g_n^\alpha(0)^2$ we have $a_0^\alpha(n) = 1$. The envelope for $g_n^\alpha(x)^2$ is defined as $G_n^\alpha(x) = h_n^\alpha(x)/(1 + a_2^\alpha(n)x^2)$.

In the Chebyshev cases $\alpha = \pm\frac{1}{2}$ we have $g_{2n}^\alpha(x, 0) = x^2 S_{2n}^\alpha(x, 0)$, see Section 7.4. Hence, for $\alpha = \pm\frac{1}{2}$, $g_n^\alpha(x)$ and its envelope $G_n^\alpha(x)$ capture the main information of the sums $S_{2n}^\alpha(x, 0)$, see Figure 7.6.

Chapter 8

Stable Polynomial Projection Operators and a posteriori Error Estimates

Numerical error is intrinsic in computer simulations of physical events. The transformation of a continuum model into a discretized model cannot capture all the information embodied in the models characterized, e.g., by partial differential equations. It also occurs in practical problems that the solution shows less regularity in certain subregions because of the given data. In order to obtain a higher accuracy of the approximate solution in the hp-version of the finite element method locally either the mesh is refined or the polynomial degree of the basis functions is increased. It is desirable, however, not to introduce too many (possibly unnecessary) new variables. In other words, the refinement process, either with respect to the mesh size or with respect to the polynomial degree, has to be adaptive. The decision which refinement to choose (if any) is based on *a posteriori estimates*.

There are various techniques for obtaining good a posteriori error estimates, see Ainsworth and Oden [4], Braess [19], or Demkowicz [31] (for adaptive meshing procedures) and references therein. Below, we turn to a posteriori error estimates for the p-method and the hp-method by the hypercircle method that may be traced back to Prager and Synge [68]. We show that equilibrated residual error estimates are p-robust. A major contribution of this chapter is the construction of a right inverse of the divergence on quadrilateral and hexahedral elements. An essential tool for this construction is a projection operator onto univariate polynomials that is uniformly bounded in the polynomial degree for two norms. In Section 8.1, the composite projection operator is defined and its main properties are proven. Next, the postprocessing which yields the error estimate is described. Section 8.3 contains the construction of right inverse of the divergence operator. Then, in Section 8.4, the proof for the p-robustness of the efficiency estimate is sketched. For more details on this part, we refer to the joint publication [20] with Dietrich Braess and Joachim Schöberl.

8.1 Construction of the Projector

The polynomial projector defined in this section is a composition of an extension operator, a projection operator for polynomials with zero boundary values and a quasi-projection operator. We start by constructing the latter operator. This quasi-projection operator maps

L^2-functions into the space of polynomials up to degree $2n-1$ and reproduces polynomials $q(x)$ with $\deg q(x) \leq n$. Furthermore, it is bounded uniformly in the polynomial degree with respect to both, L^2-, and H^1-norm. The integral kernel of this operator is defined as a generalization of the function $\phi_n(x)$ given by (7.4). Let $0 < \beta_j \leq 1$ and let $k_j^0(x,y)$ denote the jth Legendre kernel polynomial. Then we define, with $B(n) = \sum_{j=0}^{n} \beta_j$,

$$\phi_n(x,y) = \frac{1}{B(2n-1) - B(n-1)} \sum_{j=n}^{2n-1} \beta_j \, k_j^0(x,y).$$

Plugging in the definition of kernel polynomials, see (3.11), and exchanging the order of summation, ϕ_n can be rewritten as

$$\begin{aligned} \phi_n(x,y) &= \sum_{j=0}^{N} \frac{B(N) - B(i-1)}{B(N) - B(n-1)} \frac{2i+1}{2} P_i(x)P_i(y) - \sum_{i=0}^{n} \frac{B(n) - B(i-1)}{B(N) - B(n-1)} \frac{2i+1}{2} P_i(x)P_i(y) \\ &= \sum_{i=n}^{N} \frac{B(N) - B(i-1)}{B(N) - B(n-1)} \frac{2i+1}{2} P_i(x)P_i(y) + \sum_{i=0}^{n-1} \frac{2i+1}{2} P_i(x)P_i(y), \end{aligned}$$

where $N = 2n-1$. Let T_i denote the projection operator mapping a function $u \in L^2(-1,1)$ to the ith term of its Legendre expansion, i.e.,

$$(T_i u)(x) = \frac{2i+1}{2} \int_{-1}^{1} P_i(y)u(y)\, dy \, P_i(x) = \frac{2i+2}{2} (P_i, u)_0 \, P_i(x).$$

Hence, for $u \in L^2(-1,1)$, it follows that

$$\|T_i u\|_{L^2}^2 = \left(\frac{2i+1}{2}\right)^2 (P_i, u)_0^2 \int_{-1}^{1} P_i(x)^2 \, dx = \frac{2i+1}{2} (P_i, u)_0^2. \tag{8.1}$$

Furthermore, recall that $\sum_{i=0}^{\infty} \|T_i u\|_{L^2}^2 = \|u\|_{L^2}^2$ by Parseval's formula (Theorem 3.10). With the notation fixed above and using the rewriting of $\phi_n(x,y)$, we define the quasi-projection operator $\mathcal{S}_n$ with kernel $\phi_n(x,y)$:

$$\mathcal{S}_n = \sum_{i=0}^{2n-1} \gamma_n(i) \, T_i \qquad \text{with} \qquad \gamma_n(i) = \begin{cases} 1, & i \leq n, \\ \frac{B(2n-1) - B(i-1)}{B(2n-1) - B(n-1)}, & n < i \leq 2n-1. \end{cases}$$

For $\beta_j = 1$ and trigonometric polynomials this quasi-projection operator is similar to the operator introduced by de la Vallée-Poussin [30]. Note that $\gamma_n(i) \leq 1$.

Lemma 8.1. *Let the coefficients $0 < \beta_j \leq 1$ be such that*

$$\frac{\beta_j}{B(2n-1) - B(n-1)} \leq \frac{1}{n}, \qquad n \leq j \leq 2n-1. \tag{8.2}$$

Then the quasi-projection operators $\mathcal{S}_n$ satisfy

(i) $\mathcal{S}_n$ reproduce polynomials with degree $\leq n$.

(ii) The operators $\mathcal{S}_n$ are uniformly bounded with respect to n, more precisely we have

$$\|\mathcal{S}_n\|_{L^2} \leq 1 \qquad \text{and} \qquad |\mathcal{S}_n|_{H^1} \leq 3.$$

Proof. The first assertion holds trivially because of the reproducing property of Legendre kernel polynomials. The L^2-norm of $\mathcal{S}_n$ is estimated by exploiting the L^2-orthogonality of Legendre polynomials. With (8.1) and Bessel's inequality one obtains

$$\begin{aligned}\|\mathcal{S}_n u\|_{L^2}^2 &= \sum_{i,j=0}^{2n-1} \gamma_n(i)\gamma_n(j)\frac{2i+1}{2}\frac{2j+1}{2}|(P_i,u)_0|\,|(P_j,u)_0|\,(P_i,P_j)_0 \\ &= \sum_{i=0}^{2n-1} \gamma_n(i)^2\|T_i u\|_{L^2}^2 \le \sum_{i=0}^{2n-1} \|T_i u\|_{L^2}^2 \le \|u\|_{L^2}^2.\end{aligned}$$

In the course of estimating the H^1-seminorm of $\mathcal{S}_n u$ two identities from Chapter 4.1 are needed. Recall equation (4.19) relating Legendre polynomials and their derivatives,

$$P_n(x) = \frac{1}{2n+1}\big[P'_{n+1}(x) - P'_{n-1}(x)\big],$$

and perform partial integration to reformulate $(P_i,u)_0$ in terms of derivatives of u, i.e.,

$$\int_{-1}^{1} P_i(y)u(y)\,dy = -\frac{1}{2i+1}\int_{-1}^{1}[P_{i+1}(y) - P_{i-1}]u'(y)\,dy = -\frac{1}{2i+1}\big[(P_{i+1},u')_0 - (P_{i-1},u')_0\big].$$

The boundary terms above vanish since $P_n(1) = 1$ and $P_n(-1) = (-1)^n$. The other identity that we use is the evaluation (4.21) of the L^2-inner product of derivatives of Legendre polynomials,

$$(P'_i,P'_j)_0 = \int_{-1}^{1} P'_i(x)P'_j(x)\,dx = \begin{cases} 0, & i-j \equiv_2 1, \\ l(l+1), & i-j\equiv_2 0, \quad \text{where } l = \min\{i,j\}.\end{cases}$$

Now we are in the position to start calculating

$$\begin{aligned}|\mathcal{S}_n u|_{H^1}^2 &= \tfrac{1}{4}\sum_{i,j=1}^{2n-1} \gamma_n(i)\gamma_n(j)(2i+1)(2j+1)(P'_i,P'_j)_0(P_i,u)_0(P_j,u)_0 \\ &= \tfrac{1}{4}\sum_{i,j=1}^{2n-1} \gamma_n(i)\gamma_n(j)(P'_i,P'_j)_0(P_{i+1}-P_{i-1},u')_0(P_{j+1}-P_{j-1},u')_0 \\ &=: \tfrac{1}{4}\sum_{i,j=0}^{2n} M_{i,j}(P_i,u')_0(P_j,u')_0.\end{aligned}$$

In the last step, via reordering, a symmetric and positive definite matrix M is defined and we continue estimating,

$$4|\mathcal{S}_n u|_{H^1}^2 \le \sum_{i,j=0}^{2n} |M_{i,j}|\,|(P_i,u')_0||(P_j,u')_0| = 2\sum_{i,j=0}^{2n} \frac{|M_{i,j}|}{\sqrt{2i+1}\sqrt{2j+1}}\|T_i u'\|_{L^2}\|T_j u'\|_{L^2}.$$

Let the normalized matrix $M^{(0)}$ be given by $M^{(0)}_{i,j} = \frac{1}{\sqrt{2i+1}\sqrt{2j+1}}|M_{i,j}|$, for $0 \le i,j \le 2n$, and for $0 \le i \le n$ let $\underline{t}_i = \|T_i u'\|_{L^2}$. Then it follows that

$$|\mathcal{S}_n u|_{H^1}^2 \le \underline{t}^T M^{(0)}\underline{t} \le \rho\big(M^{(0)}\big)\|\underline{t}\|_2^2 = \rho\big(M^{(0)}\big)\sum_{i=0}^{2n}\|T_i u'\|_{L^2}^2 \le \rho\big(M^{(0)}\big)\|u'\|_{L^2}^2.$$

Thus it remains to show that the spectral radius of $M^{(0)}$ is bounded by a constant. It is sufficient to provide an estimate for the row sum-norm of $M^{(0)}$, which is an upper bound for $\rho\big(M^{(0)}\big)$. After reordering, for determining the matrix entries of M only the definition of γ_n and the values of $(P_i', P_j')_0$ are needed. For dealing with the last row it is convenient to introduce the parameter

$$\theta_j = \begin{cases} \beta_{2n-1} + \beta_{2n-2}, & j = 2n-1, \\ \beta_{2n-1}, & j = 2n. \end{cases}$$

We omit the intermediate computations and state only the resulting expressions, starting with the diagonal entries of M:

$$\begin{aligned} M_{i,i} &= 2(2i+1), && 0 \le i < n, \\ M_{i,i} &= 2(2i+1)\gamma_n(i+1) - i(i-1)\frac{\beta_i + \beta_{i-1}}{B(2n-1) - B(n-1)}\big(1 - \gamma_n(i-1)\big), && n \le i < 2n-1, \\ M_{j,j} &= (j-1)j\left(\frac{\theta_j}{B(2n-1) - B(n-1)}\right)^2, && j = 2n-1, 2n. \end{aligned}$$

By (4.21), we have that $M_{i,j} = 0$ if $i - j$ is odd. In the following we silently assume that $i - j$ is even. Furthermore, since M is symmetric, it is sufficient to consider the upper right triangular matrix. Hence, from now on, assume that $i \le j - 2$. The matrix entries in the upper left block vanish, more precisely,

$$M_{i,j} = 0, \qquad \text{for} \quad i < j < n.$$

The nonzero off-diagonal entries are:

$$\begin{aligned} M_{i,n} &= -2(2i+1)\frac{\beta_n}{B(2n-1) - B(n-1)}, && 0 \le i < j, \\ M_{i,j} &= -2(2i+1)\frac{\beta_j + \beta_{j-1}}{B(2n-1) - B(n-1)}, && i < n < j < 2n-1, \\ M_{i,j} &= -2(2i+1)\frac{\theta_j}{B(2n-1) - B(n-1)}, && i < n,\ j = 2n-1, 2n, \\ M_{n,j} &= -\frac{\beta_j + \beta_{j-1}}{B(2n-1) - B(n-1)}\left(2(2n+1) - \frac{(n+1)\beta_n}{B(2n-1) - B(n-1)}\right), \\ & && n < j < 2n-1, \\ M_{n,j} &= \theta_j \frac{(n-1)n\,\gamma_n(n-1) - (n+1)(n+2)\gamma_n(n+1)}{B(2n-1) - B(n-1)}, && j = 2n-1, 2n, \\ M_{i,j} &= -\frac{\beta_j + \beta_{j-1}}{B(2n-1) - B(n-1)}\left(2(2i+1)\gamma_n(i+1) - \frac{(i-1)i\,(\beta_i + \beta_{i-1})}{B(2n-1) - B(n-1)}\right], \\ & && n < i < j < 2n-1, \\ M_{i,j} &= -\left(2(2i+1)\gamma_n(i+1) - \frac{(i-1)i\,(\beta_i + \beta_{i-1})}{B(2n-1) - B(n-1)}\right)\frac{\theta_n}{B(2n-1) - B(n-1)}, \\ & && n < i,\ j = 2n-1, 2n. \end{aligned}$$

Given condition (8.2) on the parameters β_j it is now easily verified that

$$M_{i,i} \le 2(2i+1) \qquad \text{and} \qquad |M_{i,j}| \le \frac{4}{n}\sqrt{2i+1}\sqrt{2j+1}, \quad i \ne j.$$

Since there are at most n nonzero off-diagonal entries in each row, it follows that the row sum of $M^{(0)}$ does not exceed 6, and the proof is complete. ☐

Before continuing with the construction of the polynomial projection operator, let us comment on two possible choices for the parameter β_j. Firstly, let $\beta_j = 1$. This choice corresponds to the operator introduced by de la Vallée-Poussin and, for $y = 0$, to the operator studied in the last chapter. Condition (8.2) is clearly satisfied, since $B(n) = n+1$. This setting has also been used in [20] and the matrix entries of M for this specific choice can be found there.

Secondly, let $\beta_j = \frac{1}{j+1}$ and hence $B(n) = H_{n+1}$, the $(n+1)$th harmonic number. In this setting condition (8.2) is again satisfied. The reason for choosing the weight β_j this way is that the function

$$\psi_n(x,y) = \sum_{j=0}^{n} \frac{1}{j+1} k_j^0(x,y) = \sum_{j=0}^{n} \big(H_{n+1} - H_j\big) \frac{2j+1}{2} P_j(x) P_j(y) \tag{8.3}$$

is nonnegative for all $(x,y) \in [-1,1]^2$, which is proven below. Since $\phi_n(x,y)$ can be expressed as

$$\phi_n(x,y) = \frac{\psi_{2n-1}(x,y) - \psi_n(x,y)}{H_{2n} - H_n}$$

using that $\psi_n(x,y) \geq 0$, the L^1-norm of $\phi_n(\cdot,y)$ can be estimated analogously to the L^1-estimate of $\phi_n(x)$ given in Section 7.1. The positivity of $\psi_n(x,y)$ can be shown using the aforementioned result due to Gasper [39], according to which there exists a nonnegative function $\kappa(x,y,z)$ such that

$$P_n(x)P_n(y) = \int_{-1}^{1} \kappa(x,y,z) P_n(z)\, dz.$$

Hence, changing the order of integration and summation, $\psi_n(x,y)$ can be written as

$$\psi_n(x,y) = \int_{-1}^{1} \kappa(x,y,z) \sum_{i=0}^{n} (H_{n+1} - H_i) \frac{2i+1}{2} P_i(z)\, dz = \int_{-1}^{1} \kappa(x,y,z) \psi_n(z,1)\, dz.$$

In the last step it was used that $P_n(1) = 1$. It remains to show that $\psi_n(z,1) \geq 0$ for $z \in [-1,1]$. But this follows immediately from the closed form of $\psi_n(z,1)$ generated using SumCracker. As input we use the sum representation on the right hand side of (8.3) involving harmonic numbers.

In[37]:= $\mathbf{Crack[HarmonicNumber[n+1]SUM[\frac{2j+1}{2}LegendreP[j,z],\{j,0,n\}] - SUM[HarmonicNumber[j]\frac{2j+1}{2}LegendreP[j,z],\{j,0,n\}]]}$

Out[37]= $\frac{-1+\text{LegendreP}[1+n,z]}{2(-1+z)}$

Since Legendre polynomials are bounded by 1, see Theorem 7.6, this proves the assertion. Alternatively the closed form expression can be obtained using the Christoffel-Darboux formula for kernel polynomials (3.11), which reads for Legendre polynomials as

$$k_n^0(x,y) = \sum_{j=0}^{n} \frac{2j+1}{2} P_j(x) P_j(y) = \frac{n+1}{2} \frac{P_{n+1}(x)P_n(y) - P_n(x)P_{n+1}(y)}{x-y}.$$

Plugging in this formula yields

$$2(1-z)^2\psi_n(z,1) = (1-z)\sum_{i=0}^{n}\big(P_i(z)-P_{i+1}(z)\big) = (1-z)\big(1-P_{n+1}(z)\big).$$

From the positivity of $\psi_n(x,y)$ it follows for the L^1-norm of $\phi_n(\cdot,y)$ that

$$\|\psi_n(\cdot,y)\|_{L^1} \le \frac{1}{H_{2n}-H_n}\left(\int_{-1}^{1}\psi_{2n-1}(x,y)\,dx + \int_{-1}^{1}\psi_n(x,y)\,dx\right) = \frac{H_{2n}+H_n}{H_{2n}-H_n}.$$

The right hand side above behaves asymptotically as $\log n$, because of $H_n \sim \log n$. If the upper summation bound is chosen to be $N = n^2-1$ instead of $N = 2n-1$, then the L^1-norm is bounded by 4. Although for this choice $\frac{\beta_j}{B(N)-B(n-1)} \le \frac{2}{N}$ is not fulfilled, which is the condition corresponding to (8.2), an examination of the proof of Lemma 8.1 shows that the conclusion still holds, possibly with a slightly bigger constant.

The matrix entries are given by the same formulas as stated in the proof with $2n-1$ replaced by n^2-1. The estimate for the diagonal entries carry over directly, but for the off-diagonal entries a more refined analysis is needed. Recall that we defined the normalized matrix $M^{(0)}$ with entries $M^{(0)}_{i,j} = \frac{1}{\sqrt{2i+1}}\frac{1}{\sqrt{2j+1}}|M_{i,j}|$ and consider first the case $j = n$ and $i < j$. These matrix entries can be bounded as follows:

$$|M_{i,n}| \le \frac{2}{H_{n^2}-H_n}\frac{2i+1}{n+1} \le \frac{2}{H_{n^2}-H_n}\frac{\sqrt{2i+1}\sqrt{2n+1}}{n+1}.$$

Since at most n of these values contribute to the row sum-norm, this part of $\|M^{(0)}\|_1$ is bounded by a constant. Next, let $i < n < j < n^2-1$. For the sum over these entries we obtain

$$\begin{aligned}\sum_{j=n}^{n^2-1}|M^{(0)}_{i,j}| &\le \frac{2}{H_{n^2}-H_n}\sum_{j=n}^{n^2-1}\frac{\sqrt{2i+1}\sqrt{2j+1}}{j(j+1)}\\ &\le \frac{C}{H_{n^2}-H_n}\sum_{j=n}^{n^2-1}\frac{1}{j+1} = C.\end{aligned}$$

The boundedness for $n < i < j < n^2-1$ can be shown by the same arguments as well as for the remaining entries of the last two rows. Hence the conclusions of Lemma 8.1 also hold for $\beta = \frac{1}{j+1}$ and $N = n^2-1$. Additionally the L^1-norm of the operator-kernel $\phi_n(\cdot,y)$ is uniformly bounded, with the drawback of having to deal with a much higher maximal polynomial degree.

After this excursion we return to the construction of the projection operator. It remains to find a projector from $P^{2n-1}(-1,1)$ to $P^n(-1,1)$. This is done by separating polynomials with zero boundary values. To this end we define extension operators that provide polynomials of low H^s-norm, $s = 0,1$, to given boundary data:

$$E_n^{(s)}u(x) = \underset{\substack{v\in P^n([-1,1]),\\ v(-1)=u(-1),\, v(1)=u(1)}}{\operatorname{argmin}} \|v\|_s, \qquad s = 0,1.$$

The properties of these operators are summarized in the next lemma.

Lemma 8.2. *There are constants C_s, $s = 0, 1$, independent of n such that for all polynomials u on $[-1, 1]$ with* $\deg(u) \leq 2n - 1$,

$$\|E_n^{(s)} u\|_{H^s} \leq C_s \|u\|_{H^s}, \qquad s = 0, 1.$$

Proof. We restrict ourselves to $n \geq 2$, since we may choose $E_n^{(s)} u = u$ if $n < 2$. For $s = 1$ an optimal solution (up to a constant) preserving boundary values is given by

$$E_n^{(1)} u(x) = u(-1)\frac{1-x}{2} + u(1)\frac{1+x}{2}.$$

The norm estimate follows by the trace theorem, so we have

$$\|E_n^{(1)} u\|_{H^1} \leq |u(-1)| + |u(1)| \leq C\|u\|_{H^1}.$$

In order to determine the L^2-extension we consider the minimization for the left and the right endpoint separately,

$$w^{\pm} = \underset{\substack{v \in P^n([-1,1]),\\ v(\pm 1)=1, v(\mp 1)=0}}{\operatorname{argmin}} \|v\|_{L^2}^2.$$

The solutions to these problems are obtained by the same procedure as applied in the construction of the low energy vertex based shape functions in Section 5.2. The ansatz $v(x) = \sum_{i=0}^{n} v_{\pm,i} P_i(x)$ transforms the constrained minimization problems into algebraic ones,

$$\min \underline{v}^T A \underline{v},$$

with the diagonal matrix $A = \operatorname{diag}\left(\frac{2}{2i+1}\right)_{i=0}^{n}$. The constraints are now

$$\sum_{i=0}^{n} (-1)^i \, v_{-,i} = 1 \qquad \text{and} \qquad \sum_{i=0}^{n} v_{-,i} = 0$$

for the left endpoint and

$$\sum_{i=0}^{n} (-1)^i \, v_{+,i} = 0 \qquad \text{and} \qquad \sum_{i=0}^{n} v_{+,i} = 1$$

for the right endpoint. Solving the algebraic minimization problems yields the coefficients

$$v_{+,i} = \frac{2i+1}{n(n+2)}\left[1 + \frac{(-1)^{n+i+1}}{n+1}\right] \qquad \text{and} \qquad v_{-,i} = (-1)^i v_{+,i}.$$

The total extension operator is then given by

$$E_n^{(0)} u(x) = w^-(x) + w^+(x) = \sum_{i=0}^{n} \big((-1)^i u(-1) + u(1)\big)\, v_{+,i}\, P_i(x).$$

The L^2-norm of the extension can easily be computed by exploiting the L^2-orthogonality of Legendre polynomials. A simple summation shows that

$$\begin{aligned} \|E_n^{(0)} u\|_{L^2}^2 = {} & \frac{2}{(n+1)(n+2)} \left(u(-1)^2 + u(1)^2\right) \\ & + \frac{2}{n(n+1)(n+2)} \left(u(1) - (-1)^n u(-1)\right)^2 . \end{aligned} \tag{8.4}$$

Examining (8.4) we find that $\|E_n^{(0)} u\|_{L^2} \le 3\|E_{2n-1}^{(0)} u\|_{L^2}$. The L^2-norm of the given function is certainly not smaller than the minimal one in $P^{2n-1}(-1,1)$ and thus we obtain

$$\|E_n^{(0)} u\|_{L^2} \le 3\,\|E_{2n-1}^{(0)} u\| \le 3\,\|u\|_{L^2}. \tag{8.5}$$

□

We note that it is necessary to restrict the domain of the extension operators in Lemma 8.2, e.g., to polynomials up to degree $2n-1$. Otherwise we have no bound like (8.5). For this reason the map $\mathcal{S}_n$ enters the analysis. Since the L^2-norm of $E_n^{(0)} u$ is computed exactly, there is also no room for improvement. The upper bound $N = n^2 - 1$, that has been discussed in connection with uniform boundedness of the L^1-norm, cannot be substituted in Lemma 8.2.

Now we define projection operators for functions with zero boundary values. As basis for these polynomials we use scaled integrated Legendre polynomials $\hat{L}_i(x)$, defined as

$$\hat{L}_i(x) = \frac{1}{2}\left((2i-3)(2i-1)(2i+1)\right)^{1/2} L_i(x), \qquad i \ge 2.$$

This normalization has been chosen according to [13, 15], where the following norm estimates for $u = \sum_{i=2}^{M} u_i \hat{L}_i$ have been shown:

$$\|u'\|_{L^2}^2 \approx \sum_{i=2}^{M} i^2\, u_i^2, \qquad \|u\|_{L^2}^2 \approx \sum_{i=2}^{M} \left(\frac{1}{i^2}\, u_i^2 + (u_i - u_{i+2})^2\right). \tag{8.6}$$

We set $P_0^n(-1,1) = P^n(-1,1) \cap H_0^1(-1,1)$ and define the projection operators $\mathcal{R}_n$ for polynomials with zero boundary values:

$$\begin{aligned} \mathcal{R}_n :\ P_0^{2n-1}(-1,1) &\ \rightarrow\ P_0^n(-1,1) \\ u = \sum_{i=2}^{2n-1} u_i \hat{L}_i &\ \mapsto\ \mathcal{R}_n u = \sum_{i=2}^{n} \big(u_i - \frac{1}{n} u_{2n-i+1}\big) \hat{L}_i. \end{aligned}$$

Lemma 8.3. *The (semi-)norms $\|\mathcal{R}_n\|_{L^2}$ and $|\mathcal{R}|_{H^1}$ of the projection operators are uniformly bounded in n.*

Proof. Recalling (8.6) the H^1-estimate is obtained by a straight forward calculation

$$\begin{aligned} |\mathcal{R}_n u|_{H^1}^2 &\approx \sum_{i=2}^{n} i^2 \big(u_i - \frac{1}{n} u_{2n-i+1}\big)^2 \le 2\left(\sum_{i=2}^{n} i^2\, u_i^2 + \sum_{i=2}^{n} \frac{i^2}{n^2} u_{2n-i+1}^2\right) \\ &\le 2\left(\sum_{i=2}^{n} i^2\, u_i^2 + \sum_{i=2}^{n} (2n-i+1)^2 u_{2n-i+1}^2\right) = 2 \sum_{i=2}^{2n-1} i^2\, u_i^2 \preceq \|u'\|_{L^2}^2, \end{aligned}$$

where $\preceq$ refers to $\le$ up to a constant. The boundedness in the L^2-norm follows the same lines. First use (8.6) to express $\|\mathcal{R}_n u\|_{L^2}$, then basic estimates yield

$$\begin{aligned} \|\mathcal{R}_n u\|_{L^2}^2 &\approx \sum_{i=2}^{n} \frac{1}{i^2} \big(u_i - \frac{1}{n} u_{2n-i+1}\big)^2 + \sum_{i=2}^{n} \big((u_i - u_{i+2}) - \frac{1}{n}(u_{2n-i+1} - u_{2n-i-1})\big)^2 \\ &\le 2 \sum_{i=2}^{n} \big(\frac{u_i^2}{i^2} + \frac{u_{2n-i+1}^2}{(2n-i+1)^2}\big) + 2 \sum_{i=2}^{n} \big((u_i - u_{i+2})^2 + (u_{2n-i+1} - u_{2n-i-1})^2\big) \\ &= 2 \sum_{i=2}^{2n-1} \frac{1}{i^2} u_i^2 + (u_i - u_{i+2})^2 \preceq \|u\|_{L^2}^2. \end{aligned}$$

□

Summarizing the results of the previous lemmas we are now in the position to define the composite projection operators and to prove the main result of this section.

Lemma 8.4. *Let $I = [-1, 1]$. There exist projection operators $\mathcal{Q}_n : L^2(I) \to P^n(I)$ which are uniformly bounded in the polynomial degree n with respect to the L^2- and simultaneously the H^1-norm.*

Proof. First define the projection operators $\tilde{\mathcal{R}}_n^{(s)}$ as composition of the operator $\mathcal{R}_n$ of Lemma 8.3 and the extension operators $E_n^{(s)}$ of Lemma 8.2 for $s = 0, 1$ as

$$\tilde{\mathcal{R}}_n^{(s)} = \mathcal{R}_n\big(id - E_n^{(s)}\big) + E_n^{(s)}.$$

Since $\big(E_n^{(1)} - E_n^{(0)}\big)v$ is a polynomial of degree less or equal n for $v \in P^{2n-1}(I)$, the operators $\tilde{\mathcal{R}}_n^{(0)}$ and $\tilde{\mathcal{R}}_n^{(1)}$ coincide. Indeed,

$$\begin{aligned}
\tilde{\mathcal{R}}_n^{(1)}v &= \big(\mathcal{R}_n(id - E_n^{(1)}\big)v + E_n^{(1)}v \\
&= \big(\mathcal{R}_n\big(id - E_n^{(0)}\big) + E_n^{(0)}\big)v + \big(id - \mathcal{R}_n\big)\big(E_n^{(1)} - E_n^{(0)}\big)v \\
&= \tilde{\mathcal{R}}_n^{(0)}v.
\end{aligned}$$

Hence, the norm estimates of the individual operators prove the estimate for $\tilde{\mathcal{R}}_n^{(0)} = \tilde{\mathcal{R}}_n^{(1)}$. Finally we set $\mathcal{Q}_n = \tilde{\mathcal{R}}_n^{(0)}\mathcal{S}_n = \tilde{\mathcal{R}}_n^{(1)}\mathcal{S}_n$ to complete the proof. □

8.2 Formulation of the Postprocessing Method

Let Ω be a polygonal or polyhedral domain in $\mathbb{R}^d$, $d = 2, 3$. We consider the homogeneous Poisson equation (compare to Example 2.1) which in variational form is written as

$$a(u, v) = f(v), \qquad \forall v \in V,$$

with

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx, \qquad f(v) = \int_\Omega f v \, dx.$$

We consider finite element approximation with polynomials of higher order on a quasi-uniform triangulation of a domain Ω with simplicial, quadrilateral or hexahedral elements T. Since the notation for the finite element spaces differ for the triangulations above, we will focus on triangular meshes throughout this chapter unless otherwise stated. The finite element solution is given by

$$a(u_h, v) = f(v), \qquad \forall v \in V_h,$$

and in this case we have

$$V_h = \{v \in V \mid v|_T \in P^{p+1}(T)\}.$$

We assume that f is a piecewise polynomial of degree p. Otherwise f is assumed to be the piecewise approximation of the exact f_{ex}. For sake of simplicity we defined the bilinear form with constant coefficient function $a(x) \equiv 1$. The analysis certainly also works for piecewise constant coefficient functions and could be extended to a highly varying coefficient function satisfying the quasi-monotonicity condition.

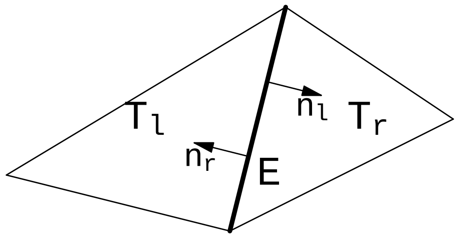


Figure 8.1: Edge E with neighbouring elements T_l and T_r and corresponding outer unit normal vectors n_l and n_r

Let the space $H(\mathrm{div},\Omega)$ for $d=2,3$ be defined as

$$H(\mathrm{div},\Omega) = \{q \in (L^2(\Omega))^d \mid \mathrm{div}\, q \in L^2(\Omega)\}$$

with inner product

$$(p,q)_{\mathrm{div}} = (p,q)_0 + \int_\Omega \mathrm{div}\, p\, \mathrm{div}\, q\, dx.$$

This space again is a Hilbert space with norm induced by the inner product $(\cdot,\cdot)_{\mathrm{div}}$. In the following a postprocessing algorithm is established which provides an equilibrated flux $\sigma \in \Sigma_h \subset H(\mathrm{div})$, i.e., the flux satisfies pointwise

$$\mathrm{div}\, \sigma + f = 0. \tag{8.7}$$

The hypercircle method is the general framework. Cheap implementations of this technique have been described recently, see e.g. [21, 58]. Merely local problems have to be solved that are organized on local patches around nodes of the mesh. This is in contrast to local problems for other classical error estimators that are oriented to elements of the triangulation. We propose the following construction of the flux σ: Following [21] the residual $r \in V'$ defined as

$$\langle r, v\rangle = f(v) - a(u_h, v) = a(u - u_h, v)$$

is written as

$$\langle r, v\rangle = \sum_T (r_T, v)_{L^2(T)} + \sum_E (r_E, v)_{L^2(E)}. \tag{8.8}$$

The local error is assessed via the element residual and via the jumps across element-interfaces. Let n denote the unit outer normal vector of an element T and associated to an edge E let n_l and n_r denote the unit outer normal vector at E of the element to the left and to the right of E, respectively, see Figure 8.1. Then the well known element and edge residuals are given by

$$\begin{aligned} r_T &= f + \mathrm{div}(\nabla u_h), \\ r_E &= \lceil \partial_n u_h \rceil := \frac{\partial u_{h,l}}{\partial n_l} + \frac{\partial u_{h,r}}{\partial n_r}. \end{aligned} \tag{8.9}$$

Here the divergence is taken pointwise on each element. Let ϕ_V denote the hat basis function associated with the vertex V. Its support is the patch $\omega_V = \bigcup\{T \mid V \in \partial T\}$. The family $(\phi_V)_{V\in\mathcal{V}}$ forms a partition of unity subordinated to the union of patches ω_V and the residual is decomposed by this partition of unity, i.e., $\langle r_{\omega_V}, v\rangle := \langle r, \phi_V v\rangle$. Recalling (8.9) the element and edge terms of the residual read as

$$\begin{aligned} r_{\omega_V,T} &= \phi_V\big(f + \operatorname{div}(\nabla u_h)\big),\\ r_{\omega_V,E} &= \phi_V\lceil\partial_n u_h\rceil. \end{aligned} \tag{8.10}$$

By Galerkin orthogonality of the hat basis functions the local residuals are bi-orthogonal to constant functions. The element terms as well as the edge terms in (8.10) are polynomials of degree at most $p+1$. We construct a vector function σ_{ω_V} in the broken Raviart-Thomas space $RT^{p+1}(\omega_V)$ [6],

$$\begin{aligned} RT^p_{-1}(\omega_V) &= \{\tau \in L^2(\omega_V) \mid \tau|_T \in RT^p(T),\ T \subset \omega_V\},\\ \text{with}\qquad RT^p(T) &= \{\tau \mid \tau(x) = q_T + s_T x,\ q_T \in (P^p)^2,\ s_T \in P^p\}, \end{aligned}$$

such that $\operatorname{div}\sigma_{\omega_V} = r_{\omega_V}$. The divergence is understood in the distributional sense and is consistent with (8.9). In combination with the boundary condition, it translates to

$$\begin{aligned} \operatorname{div}_T \sigma_{\omega_V} &= r_{\omega_V,T} &&\text{in}\quad T \subset \omega_V,\\ \lceil\sigma_{\omega_V}\cdot n\rceil &= r_{\omega_V,E} &&\text{at}\quad E \subset \omega_V,\\ \sigma_{\omega_V}\cdot n &= 0 &&\text{on}\quad \partial\omega_V. \end{aligned}$$

The computation of these local fluxes is the crucial step of the equilibration. By adding up all the local fluxes the global correction

$$\sigma_\Delta = \sum_{V\in\mathcal{V}} \sigma_{\omega_V}$$

is obtained which satisfies

$$\operatorname{div}\sigma_\Delta = r.$$

The difference between the discrete flux ∇u_h and the postprocessed flux $\nabla\sigma = u_h + \sigma_\Delta$ provides a true upper bound without generic constant to the error measured in the energy norm, i.e., for the Poisson equation the H^1-seminorm. Specifically, by the theorem of Prager and Synge, cf. [68] or [19, Theorem III.5.1],

$$\|\nabla u - \nabla u_h\|_{L^2} \le \|\nabla u_h - \sigma\|_{L^2} = \|\sigma_\Delta\|_{L^2},$$

i.e., the error estimate is reliable. The p-robust efficiency of this estimate is the content of Section 8.4. Next we present a crucial tool for the proof of this result, namely a right inverse of the divergence operator on tensor product elements. The polynomial projection operator defined in the last section enters in the construction of this inverse. The existence of the inverse can be proven for quadrilateral or hexahedral elements and is left as a conjecture for simplicial elements.

8.3 A Right Inverse of the Divergence Operator on Tensor Product Elements

The estimates in the efficiency proof given in Section 8.4 are based on two ingredients: One is the right inverse of the divergence operator on one element which is the main result of this section. The other one is the extension of normal-traces from edges to elements that has been treated in [32] and is given in the following lemma.

$$\begin{array}{ccc} H^1 & \xrightarrow{d/dx} & L^2 \\ \mathcal{Q}_{p+1}\big\downarrow & & \big\downarrow \tilde{\mathcal{Q}} \\ P^{p+1} & \xrightarrow{d/dx} & P^p \end{array} \qquad \begin{array}{ccc} H(\mathrm{div}) & \xrightarrow{\mathrm{div}} & L^2 \\ \mathcal{Q}^{\Sigma}\big\downarrow & & \big\downarrow \tilde{\mathcal{Q}}_x \otimes \tilde{\mathcal{Q}}_y \\ RT^{[p]} & \xrightarrow{\mathrm{div}} & P^p \end{array}$$

Figure 8.2: Commuting diagram properties of projectors

Lemma 8.5. *Let T be a triangle and let $\gamma \subset \partial T$ be the union of one, two or three edges of T. Let $g_n \in L^2(\gamma)$ be given such that $g_n|_E \in P^p(E)$. If $\gamma = \partial T$ we additionally assume $\int_\gamma g_n = 0$. Then there exists an extension $\sigma_p \in RT^p(T)$ such that*

$$\operatorname{div} \sigma_p = 0 \qquad \textit{and} \qquad \operatorname{tr}_{n,\gamma} \sigma_p = g_n,$$

where $\operatorname{tr}_{n,\gamma}$ denotes the normal trace on γ, and

$$\|\sigma_p\|_{L^2(T)} \leq C \inf_{\substack{\sigma \in L^2(T) \\ \operatorname{div} \sigma = 0, \operatorname{tr}_{n,\gamma} = g_n}} \|\sigma\|_{L^2(T)}.$$

For the construction of the right inverse we turn to rectangular grids. The Raviart-Thomas elements on rectangular grids build the space $RT^{[k]} = P^{k+1,k} \times P^{k,k+1}$.

Theorem 8.6. *Let T be a square or a cube and let $r_T \in P^p(T)$. Then there exists a $\sigma_T \in RT^{[p]}(T)$ such that*

$$\operatorname{div} \sigma_T = r_T \qquad \textit{and} \qquad \|\sigma_T\|_{L^2(T)} \leq C \, \|r_T\|_{H^{-1}(T)}. \tag{8.11}$$

Proof. We restrict ourselves to the two dimensional case and consider the homogeneous Poisson equation

$$\begin{aligned} \Delta w &= r_T && \text{in } T, \\ w &= 0 && \text{on } \partial T. \end{aligned}$$

Since the Laplace operator can be written as the composition $\Delta = \operatorname{div} \nabla$, the flux $\sigma = \nabla w$ satisfies $\operatorname{div} \sigma = r_T$ and $\|\sigma\|_{L^2(T)} = \|r_T\|_{H^{-1}(T)}$. What is left to be done is to project σ into the polynomials.

Let $\mathcal{I}v(x) = \int_{-1}^{x} v(s)\, ds$ and take the one dimensional projector $\mathcal{Q}_{p+1}$ from Lemma 8.4 to define another projector onto $P^p(T)$ by

$$\tilde{\mathcal{Q}} v = (\mathcal{Q}_{p+1} \mathcal{I} v)'.$$

The L^2-boundedness of the operator $\tilde{\mathcal{Q}}$ follows by the H^1-boundedness of the projection operator $\mathcal{Q}_{p+1}$. More precisely, we have

$$\|\tilde{\mathcal{Q}} v\|_{L^2(T)} = \|(\mathcal{Q}_{p+1} \mathcal{I} v)'\|_{L^2(T)} \leq \|(\mathcal{I} v)'\|_{L^2(T)} = \|v\|_{L^2(T)}.$$

The two operators $\tilde{\mathcal{Q}}$ and $\mathcal{Q}_{p+1}$ have the commuting diagram property

$$\tilde{\mathcal{Q}} u' = (\mathcal{Q}_{p+1} u)',$$

see Figure 8.2. The tensor product operator

$$\mathcal{Q}^\Sigma = (\mathcal{Q}_{p+1,x} \otimes \tilde{\mathcal{Q}}_y) \times (\tilde{\mathcal{Q}}_x \otimes \mathcal{Q}_{p+1,y}) : L^2(T) \to RT^{[p]}(T)$$

is bounded in $L^2(Q)$, and it has the commuting diagram property with the divergence, i.e.,

$$\operatorname{div} \mathcal{Q}^\Sigma = (\tilde{\mathcal{Q}}_x \otimes \tilde{\mathcal{Q}}_y) \operatorname{div}.$$

We set $\sigma_T = \mathcal{Q}^\Sigma \sigma$ to complete the proof of the theorem. □

At the moment, an analogous result for the right inverse on simplicial elements can be posed only as a conjecture.

Conjecture 8.7. *Let T be a triangular or tetrahedral element. Let $r_T \in P^p(T)$. Then there exists a $\sigma_T \in RT^p(T)$ such that*

$$\operatorname{div} \sigma_T = r_T \quad \textit{and} \quad \|\sigma_T\|_{L^2(T)} \le C \|r_T\|_{H^{-1}(T)}.$$

This conjecture is supported by the corresponding result for rectangles and also by numerical computations with finite elements of higher order. Table 8.1 contains the constants $C_{p,q}$ such that the inequalities

$$\min_{\substack{\sigma \in BDM^{p+1}(T) \\ \operatorname{div} \sigma = r_T}} \|\sigma\|^2_{L^2(T)} \quad \le \quad C_{p,q} \sup_{v \in P^{p+q}(T) \cap H^1_0(T)} \frac{(v, r_T)^2}{\|v\|^2_{H^1(T)}} \tag{8.12}$$

hold for all $r_T \in P^p(T)$. For the definition of Brezzi-Douglas-Marini-elements (BDM-elements), see [23, Ch. III.3]. The constants can be computed by finding the largest eigenvalue of generalized eigenvalue problems. The discrete H^{-1}-norms in (8.12) approach the H^{-1}-norm from below. Hence,

$$\min_{\substack{\sigma \in BDM^{p+1}(T) \\ \operatorname{div} \sigma = r_T}} \|\sigma\|^2_{L^2(T)} \le C_p \|r_T\|^2_{H^{-1}(T)}$$

with $C_p \le C_{p,q}$. The results indicate that C_p is bounded in p.

p	q = 3	q = 5	q = 8
1	1.81	1.76	1.76
2	2.05	1.92	1.92
4	2.43	1.99	1.99
8	3.23	2.00	2.00
16	4.92	2.38	2.00

Table 8.1: Coefficients C_{pq} in (8.12).

8.4 Equilibrated Residual Error Estimates are p-Robust

Now we are in the position to prove the efficiency of the resulting estimator, i.e., that the overestimation in the error is bounded uniformly in the mesh-size as well as in the polynomial degree. First we note that the $H^{-1}(\omega_V)$-norm of the local residual is bounded by the local error, i.e.,

$$\|r_{\omega_V}\|_{H^{-1}(\omega_V)} \leq C\,\|u-u_h\|_{H^1(\omega_V)}.$$

The proof of the efficiency is completed once we have shown the existence of a σ_{ω_V} on the patch so that

$$\operatorname{div}\sigma_{\omega_V} = r_{\omega_V} \qquad \text{and} \qquad \|\sigma_{\omega_V}\|_{L^2(\omega_V)} \leq \|r_{\omega_V}\|_{H^{-1}(\omega_V)}.$$

For this purpose the right inverse of the divergence given in Theorem 8.6, that applies to distributions of the form (8.8), is needed. In the formulation of the main theorem guaranteeing the efficiency of the a posteriori error estimate for large polynomial degrees, we focus on simplicial meshes, although we have to base the analysis on Conjecture 8.7.

Theorem 8.8. *Let ω_V be the patch of elements around the vertex V and let r be the residual*

$$\langle r, v\rangle = \sum_{T\subset\omega_V}\int_T r_T v + \sum_{E\subset\omega_V}\int_E r_E v,$$

with $r_T \in P^p(T)$ and $r_E \in P^p(E)$. Moreover assume that $\langle r, 1\rangle = 0$. Then there holds

$$\inf_{\substack{\sigma\in RT^p_{-1}\\ \operatorname{div}\sigma = r}} \|\sigma\|L^2(\omega_V) \leq C\,\|r\|_{[H^1(\omega_V)]'},$$

and the constant C is independent of p.

We will only sketch the main ideas of the proof of this theorem. For a full proof including all details as well as a numerical example we refer to [20].

Theorem 8.8 is proven in three steps. First the element residuals are eliminated. On each element of the patch we construct a flux $\sigma_T \in RT^p(T)$ such that $\operatorname{div}_T\sigma_T + r_T = 0$ and the norm estimate of Theorem 8.6 is satisfied. It can be shown that the norm of σ_T on the patch ω_V is bounded by the norm of r in the dual space $H^1(\omega_V)'$.

With $\sigma^{(1)} = \sum_{T\subset\omega_V}\sigma_T$ we define the new residual $r^{(1)} = r - \operatorname{div}\sigma^{(1)}$, where the divergence is understood in the distributional sense. The residual $r^{(1)}$ satisfies

$$\|r^{(1)}\|_{H^1(\omega_V)'} \leq C\,\|r\|_{H^1(\omega_V)'} \qquad \text{and} \qquad \langle r^{(1)}, 1\rangle = 0.$$

Moreover it only contains the edge terms including the edges on $\partial\omega_V$.

In the second step the boundary edge residuals are eliminated. For each triangle T consider the variational boundary value problem

$$(\nabla w, \nabla v) = \int_{E\subset\partial\omega_V} r^{(1)}_E v.$$

At this point set $\sigma_T = \nabla w$ and we have that the L^2-norm of σ_T on T is bounded by the norm of r in the dual space $H^1(\omega_V)'$. Since $r^{(1)}_E$ is a polynomial, by Lemma 8.5 there exists a polynomial $\sigma^{(2)}_T$ whose normal trace on E coincides with $r^{(1)}_E$ and whose L^2-norm on T is bounded by $\|\sigma_T\|_{L^2(T)}$. This construction can be done independently triangle by triangle. Next we subtract the divergences of $\sigma^{(2)} = \sum_T \sigma_T^{(2)}$ to obtain $r^{(2)} = r^{(1)} - \operatorname{div} \sigma^{(2)}$. This new residual contains only edge residuals on the internal edges and satisfies

$$\|r^{(2)}\|_{H^1(\omega_V)'} \leq C \, \|r\|_{H^1(\omega_V)'} \qquad \text{and} \qquad \langle r^{(2)}, 1 \rangle = 0.$$

In the last step the internal edge residuals have to be eliminated. This procedure is similar to the elimination of the boundary edge residuals, but the triangles cannot be treated independently. Hence we circle around the patch, updating the residual each time when passing a triangle. Altogether the residual has been decomposed as a sum of divergences of piecewise polynomials that are bounded as stated in the theorem.

Summarizing we have shown that if the mesh consists of

- affine quadrilateral or hexahedral elements,
- or triangular or tetrahedral elements and Conjecture 8.7 is valid,

then the error estimator is locally efficient, i.e.,

$$\|\sigma_{\omega_V}\|_{L^2(\omega_V)} \leq C \, \|\nabla u - \nabla u_h\|_{L^2(\omega_V)}$$

holds with a constant C that is independent of h and p.

Appendix A

Mathematica Implementations

A.1 Fem2D Package

As a completion to our introduction to the high order finite element method we consider two more examples, this time in two dimensions on a quadrilateral mesh. The following computations are carried out with our Mathematica program Fem2D. We already used its one dimensional counterpart (Fem1D) for the introductory example given in Section 2.6. In Fem2D only quadrilateral meshes, which have to be specified separately, can be used for computations. The input structure is that of meshes generated and exported by Netgen [73]. The description is given below.

A.1.1 Example 1

Let $\Omega = [-1,1]^2$. As first example, consider the boundary value problem: Find u such that

$$\begin{aligned} -\frac{1}{\lambda}\Delta u + \lambda\, u &= f_\lambda, && \text{in } \Omega, \\ u &= 0, && \text{on } \partial\Omega, \end{aligned} \tag{A.1}$$

where $f_\lambda(x,y) = \frac{\lambda^2+2\pi^2}{\lambda}\sin\pi x \sin\pi y$ and λ is a real parameter. The exact solution to this problem is $u_{ex}(x,y) = \sin\pi x\sin\pi y$. The variational formulation for (A.1) is obtained by multiplying the equation by smooth test functions v and integrating over the domain Ω. It reads as

$$\text{Find } u \in H_0^1(\Omega): \qquad a(u,v) = f(v), \qquad \text{for all } v \in H_0^1(\Omega), \tag{A.2}$$

where the bilinear form $a(u,v)$ is given by

$$a(u,v) = \frac{1}{\lambda}\int_\Omega \nabla u \cdot \nabla v\, dx + \lambda \int_\Omega u\, v\, dx,$$

and the linear form $f(v) = \int_\Omega f_\lambda\, v\, dx$. For the discrete approximation of this problem, we use the quadrilateral shape functions defined by (2.12)-(2.14) based on integrated Legendre polynomials. Let $\phi_i \in [\Phi_V, \Phi_E, \Phi_C]$ denote these basis functions and define the stiffness matrix K and the mass matrix M with entries

$$K_{i,j} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j\, dx, \qquad \text{and} \qquad M_{i,j} = \int_\Omega \phi_i\, \phi_j\, dx.$$

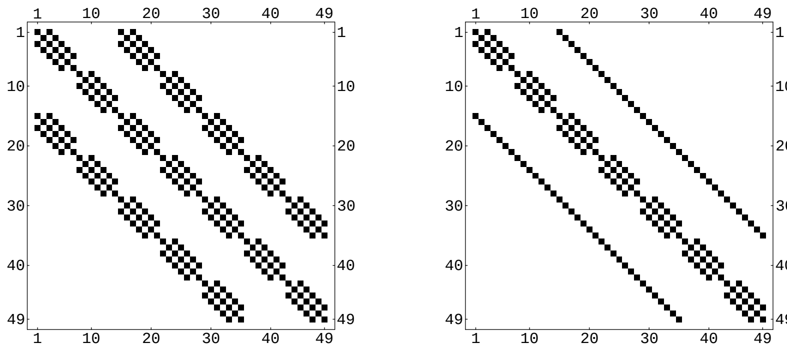


Figure A.1: Nonzero pattern of the interior block of mass matrix (left) and stiffness matrix (right) for $p = 8$

For the right hand side we define the vector $\underline{f}$ with entries $f_i = \int_\Omega f_\lambda \phi_i \, dx$. With this notation starting from the variational formulation of (A.1) we arrive at the linear system

$$\text{Find } \underline{u} \in \mathbb{R}^N : \qquad \left(\frac{1}{\lambda} K + \lambda M\right) \underline{u} = \underline{f}. \tag{A.3}$$

As we already remarked at the end of Section 2.6, the stiffness matrix is no longer a diagonal matrix in the p-version of the finite element method for higher dimensions, not even for a quadrilateral mesh. Figure A.1 shows the nonzero pattern of the block built from the cell based basis functions for the mass and the stiffness matrix. We compute the approximate solutions to (A.2) for $\lambda = 10^{-5}$ on a mesh consisting of 64 congruent square elements with polynomial degree $p = 2$ and on a mesh consisting of a single element with $p = 16$. For both options the total number of unknowns is 289. The first step in the computations is to read in the mesh information using the command:

In[38]:= **GetMeshInformation[”sqfine.mesh”]**

To visualize the mesh use **Show[MeshGraphics[ ]]**. The mesh information is read from a file, where in the first line the total number of vertices is given, followed by the coordinates of the vertices. Next, the total number of elements is given, followed by the vertices surrounding this element, where the vertices are accessed by their number in the vertex-list. The first entry in each element-line is “1”, indicating that the vertices are connected by straight lines. The boundary of the element is a directed graph and the order of the vertices has to be such that the interior of the element is on the left of this graph. Next the edges in the mesh are listed, where again the first line gives the total number of edges. The entry “1” in the first position of the following lines indicates that no curved elements are used and is followed by the two vertices defining an edge. The information for the mesh consisting of the single element $[-1, 1]^2$ is given by:

4		
	−1	−1
	1	−1
	1	1
	−1	1

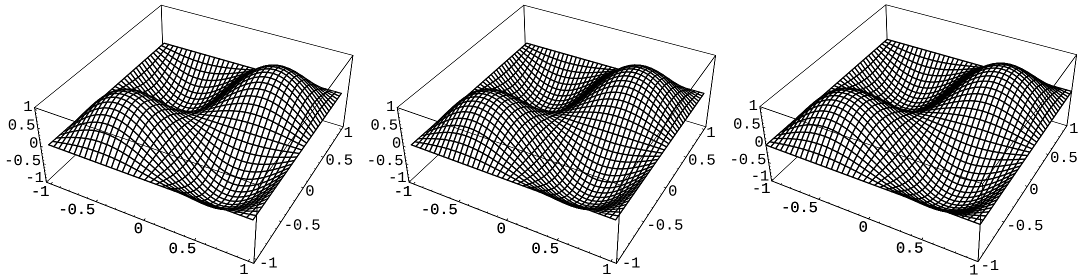


Figure A.2: Approximate solution for $p = 16$ on one element (left), for $p = 2$ on 64 elements (middle) and exact solution (right)

1					
	1	1	2	3	4
4					
	1	1	2		
	1	2	3		
	1	3	4		
	1	4	1		

The next step is to compute the right hand side vector rhs $= \underline{f}$ for a given polynomial degree p using the basis functions defined via integrated Legendre polynomials, which is specified by "Recurrence[ ]":

```
In[39]:= rhs = ComputeSourceVector[f, p, Recurrence[ ]];
```

The linear equation system (A.3) is solved iteratively using the (preconditioned) conjugate gradient method with the "(P)CGMethod" command:

```
In[40]:= sol = PCGMethod[{1/λ&, λ&}, p, "STIFF + MASS", {Recurrence[], DRecurrence[]}, rhs];
```

The key-word "STIFF+MASS" specifies which bilinear forms are used and the corresponding coefficient functions are given in the first argument. Per default the preconditioner (2.19) is used. The computations with $p = 16$ on a single element need less iterations (≈ 20) than the compuations with $p = 2$ on the finer mesh (≈ 50). The solutions can be visualized using

```
In[41]:= VisualizeFunction[sol[[1]], p, n, Recurrence[ ]]
```

The first component of the vector "sol" contains the solution vector $\underline{u}$. The second argument of "VisualizeFunction" is the polynomial degree and the third argument specifies the number of points used for plotting on each element in each direction. Figure A.2 compares the two approximate solutions obtained with Fem2D to the exact solution.

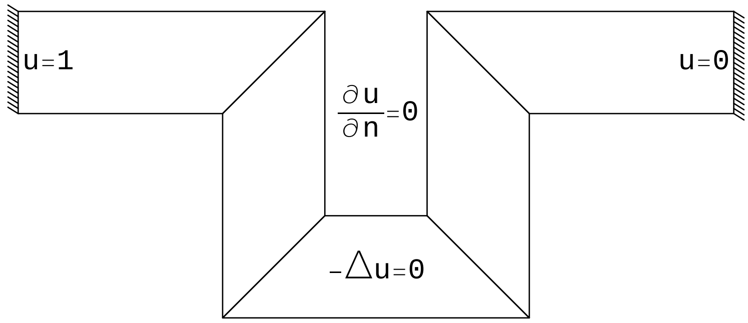


Figure A.3: Geometry and mesh for (A.4)

A.1.2 Example 2

As second example, consider the boundary value problem: Find u such that

$$\begin{aligned} -\Delta u &= 0, && \text{in } \Omega, \\ u &= 0, && \text{on } \Gamma_0, \\ u &= 1, && \text{on } \Gamma_1, \\ \frac{\partial u}{\partial n} &= 0, && \text{on } \Gamma_2. \end{aligned} \tag{A.4}$$

Here n denotes the unit outer normal vector on $\partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ and $\frac{\partial u}{\partial n}$ is the directional derivative along n. The first two boundary conditions are called Dirichlet boundary conditions and the third one Neumann boundary condition. The domain Ω and the boundaries Γ_i are as shown in Figure A.3. This problem describes a flow through the domain Ω, where the homogeneous Neumann boundary condition on Γ_2 specifies an isolation. The variational formulation of (A.4) is obtained by multiplying the differential equation by a smooth test function and integrating over the domain. By Green's formula we have

$$-\int_\Omega \Delta u\, v\, dx = \int_\Omega \nabla u \cdot \nabla v\, dx - \int_{\partial\Omega} \frac{\partial u}{\partial n} v\, ds.$$

The integral over the boundary of Ω can be split according to our splitting of $\partial\Omega$, i.e.,

$$\int_{\partial\Omega} \frac{\partial u}{\partial n} v\, ds = \int_{\Gamma_0} \frac{\partial u}{\partial n} v\, ds + \int_{\Gamma_1} \frac{\partial u}{\partial n} v\, ds + \int_{\Gamma_2} \frac{\partial u}{\partial n} v\, ds = 0.$$

The first two integrals vanish because we choose the test functions

$$v \in \{ f \in H^1(\Omega) \mid f = 0 \text{ on } \Gamma_0 \text{ and } f = 0 \text{ on } \Gamma_1 \},$$

whereas the third integral vanishes because of the zero Neumann boundary condition. Observe that for this type of problem the Neumann boundary condition enters naturally in the variational formulation. Even if we assume a boundary condition of the form $\frac{\partial u}{\partial n} = g$, $g \neq 0$, then $\int_{\Gamma_2} g\, v\, ds$ can be included in the right hand side $\underline{f}$. This is why Neumann boundary conditions are often referred to as "natural boundary conditions". The Dirichlet boundary conditions on the other hand, have to be demanded separately, e.g., by including them in the

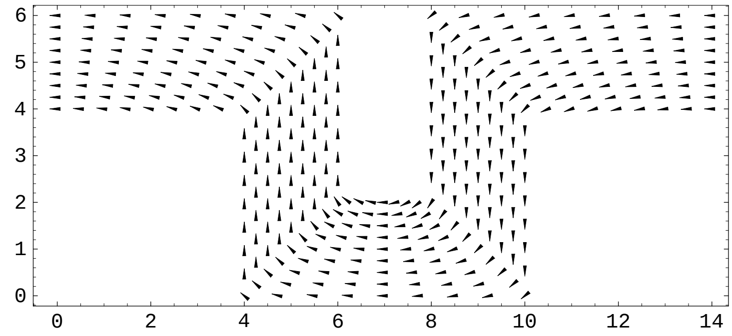


Figure A.4: Gradient field of the solution to (A.4)

definition of the solution space. Hence they are also called "essential boundary conditions". Note, however, that this classification is problem dependent.

The gradient field of the solution to this problem is obtained by the "VisualizeGradient-Field" command and can be plotted using the Mathematica built-in "ListPlotVectorField" function:

In[42]:= **gf = VisualizeGradientField[sol[[1]], *p*, *n*, Recurrence[], DRecurrence[]];**

In[43]:= **ListPlotVectorField[gf, ScaleFactor → 0.2, Frame → True, HeadWidth → 0.3, HeadLength → 0.015]**

The latter command delivers the output shown in Figure A.4.

A.2 IntJac Package

In this section we follow the notations of Chapter 6. Recall the definition of a family of cell based basis functions defined on a triangular mesh,

$$\phi_{i,j}(x,y) = \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i \hat{p}_j^{2i-a}(y), \qquad i+j \le p,\ i \ge 2,\ j \ge 1, \quad 0 \le a \le 4,$$

where $\hat{p}_n^\alpha(\zeta)$ are integrated Jacobi polynomials. We demonstrate our program that executes the algorithm described in Section 6.3 by computing the integrals

$$\int_{\hat{T}} \frac{d}{dx}\phi_{i,j}(x,y)\frac{d}{dy}\phi_{k,l}(x,y)\, d(x,y), \tag{A.5}$$

for $a = 0$ and $a = 1$. Taking derivatives of the basis functions and decoupling the integrals using the Duffy transformation is performed by the Prepare2DIntegrand command. First, we consider the case $a = 0$:

In[44]:= **ComputeMatrixEntries[Prepare2DIntegrand[phi[*i*, *j*, *x*, *y*], phi[*k*, *l*, *x*, *y*], {*x*, *y*}, {*x*, *y*}], *x*, *y*]**

1. Collecting integrands depending on x
   → finished collecting (0.013998 Second)
   → 3 integrands
2. Rewriting integrated Jacobi polynomials in terms of Jacobi polynomials
   Case 2(a) for phat[k, 0, x]
   → finished rewriting (0.003999 Second)

3. Collecting integrands depending on x
   → finished collecting (0.003 Second)
   → 2 integrands
6. Evaluate integrals using Jacobi orthogonality relation
   → finished evaluating (0.002999 Second)
1. Collecting integrands depending on y
   → finished collecting (0.001 Second)
   → 3 integrands
2. Rewriting integrated Jacobi polynomials in terms of Jacobi polynomials
   Case 2(b) for phat[j, 2 i, y] w[-1 + 2 i, y]
   Case 2(a) for phat[j, 2 i, y] w[2 i, y]
   Case 2(c) for phat[l, 2 + 2 i, y] w[2 i, y]
   Case 2(a) for phat[j, 2 i, y] w[1 + 2 i, y]
   → finished rewriting (0.049992 Second)
3. Collecting integrands depending on y
   → finished collecting (0.017998 Second)
   → 5 integrands
4. Adjusting Jacobi polynomials to appearing weights functions
   → finished adjusting (0.007998 Second)
6. Evaluate 7 integrals using Jacobi orthogonality relation
   → finished evaluating (0.006999 Second)

Out[44]=
$$-\frac{4(j+1)\delta(0,-i+k+1)\delta(0,-j+l-2)}{(2i-3)(2i-1)(i+j)(2i+2j-1)(2i+2j+1)}+\frac{4(i-1)\delta(0,-i+k+1)\delta(0,-j+l-1)}{(2i-3)(2i-1)(i+j-1)(i+j)(2i+2j-1)}$$
$$+\frac{4(2i+j)\delta(0,-i+k-1)\delta(0,l-j)}{(2i-1)(2i+1)(i+j)(2i+2j-1)(2i+2j+1)}+\frac{4(2i+j-2)\delta(0,-i+k+1)\delta(0,l-j)}{(2i-3)(2i-1)(i+j-1)(2i+2j-3)(2i+2j-1)}$$
$$+\frac{4i\delta(0,-i+k-1)\delta(0,-j+l+1)}{(2i-1)(2i+1)(i+j-1)(i+j)(2i+2j-1)}-\frac{4(j-1)\delta(0,-i+k-1)\delta(0,-j+l+2)}{(2i-1)(2i+1)(i+j-1)(2i+2j-3)(2i+2j-1)}$$

Collecting the integrands in step 5 of the algorithm is in the implementation included in step 4. For the evaluation of the integrals with respect to x, steps 4 and 5 do not have to be executed. Next, we compute (A.5) for $a = 1$:

In[45]:= **ComputeMatrixEntries[Prepare2DIntegrand[phi[*i*, *j*, *x*, *y*], phi[*k*, *l*, *x*, *y*], {*x*, *y*}, {*x*, *y*}], *x*, *y*]**

1. Collecting integrands depending on x
   → finished collecting (0.005999 Second)
   → 3 integrands
2. Rewriting integrated Jacobi polynomials in terms of Jacobi polynomials
   Case 2(a) for phat[k, 0, x]
   → finished rewriting (0.002 Second)
3. Collecting integrands depending on x
   → finished collecting (0.002 Second)
   → 2 integrands
6. Evaluate integrals using Jacobi orthogonality relation
   → finished evaluating (0.001 Second)
1. Collecting integrands depending on y
   → finished collecting (0. Second)
   → 3 integrands
2. Rewriting integrated Jacobi polynomials in terms of Jacobi polynomials

Case 2(a) for phat[j, -1 + 2 i, y] w[-1 + 2 i, y]
Case 2(a) for phat[j, -1 + 2 i, y] w[2 i, y]
Case 2(b) for phat[l, 1 + 2 i, y] w[2 i, y]
Case 2(a) for phat[j, -1 + 2 i, y] w[1 + 2 i, y]
→ finished rewriting (0.050993 Second)

3. Collecting integrands depending on y
   → finished collecting (0.010999 Second)
   → 12 integrands
4. Adjusting Jacobi polynomials to appearing weights functions
   → finished adjusting (0.028995 Second)
6. Evaluate 22 integrals using Jacobi orthogonality relation
   → finished evaluating (0.013998 Second)

Out[45]=
$$\frac{(j+1)(j+2)(2i+j-1)\delta(0,-i+k+1)\delta(0,-j+l-3)}{(2i-3)(2i-1)(i+j-1)(i+j)(i+j+1)(2i+2j-1)(2i+2j+1)}$$
$$-\frac{2(j+1)\left(8i^2+6ji-16i+2j^2-5j+6\right)\delta(0,-i+k+1)\delta(0,-j+l-2)}{(2i-3)(2i-1)(i+j-1)(i+j)(2i+2j-3)(2i+2j-1)(2i+2j+1)}$$
$$-\frac{j(2i+j-1)(2i+j)\delta(0,-i+k-1)\delta(0,-j+l-1)}{(2i-1)(2i+1)(i+j-1)(i+j)(i+j+1)(2i+2j-1)(2i+2j+1)}$$
$$+\frac{2(i-2)\left(4i^2+2ji-8i+j^2-2j+3\right)\delta(0,-i+k+1)\delta(0,-j+l-1)}{(2i-3)(2i-1)(i+j-2)(i+j-1)(i+j)(2i+2j-3)(2i+2j-1)}$$
$$+\frac{2(2i+j-1)\left(4i^2+2ji-2i+2j^2-j-2\right)\delta(0,-i+k-1)\delta(0,l-j)}{(2i-1)(2i+1)(i+j-1)(i+j)(2i+2j-3)(2i+2j-1)(2i+2j+1)}$$
$$+\frac{2(2i+j-3)\left(4i^2+2ji-10i+2j^2-3j+4\right)\delta(0,-i+k+1)\delta(0,l-j)}{(2i-3)(2i-1)(i+j-2)(i+j-1)(2i+2j-5)(2i+2j-3)(2i+2j-1)}$$
$$+\frac{2(i-1)\left(4i^2+2ji-2i+j^2-2j\right)\delta(0,-i+k-1)\delta(0,-j+l+1)}{(2i-1)(2i+1)(i+j-2)(i+j-1)(i+j)(2i+2j-3)(2i+2j-1)}$$
$$-\frac{(j-1)(2i+j-4)(2i+j-3)\delta(0,-i+k+1)\delta(0,-j+l+1)}{(2i-3)(2i-1)(i+j-3)(i+j-2)(i+j-1)(2i+2j-5)(2i+2j-3)}$$
$$-\frac{2(j-1)\left(8i^2+6ji-12i+2j^2-7j+4\right)\delta(0,-i+k-1)\delta(0,-j+l+2)}{(2i-1)(2i+1)(i+j-2)(i+j-1)(2i+2j-5)(2i+2j-3)(2i+2j-1)}$$
$$+\frac{(j-2)(j-1)(2i+j-2)\delta(0,-i+k-1)\delta(0,-j+l+3)}{(2i-1)(2i+1)(i+j-3)(i+j-2)(i+j-1)(2i+2j-5)(2i+2j-3)}$$

In this computation we observe that case 2(c) never occurs and that the rewriting in step 4 introduces more terms compared to the previous example. As we remarked in Section 6.3, in the three dimensional case, e.g., for $a = b = 0$ in step 4 of the algorithm a further rewriting is needed that reduces Jacobi three term recurrences to zero. This correction step is necessary for the integration with respect to y of the integral

$$\int_{\hat{T}} \frac{d}{dz}\phi_{i,j,k}(x,y,z)\frac{d}{dz}\phi_{l,m,n}(x,y,z)\, d(x,y,z).$$

The output of our program for this evaluation, showing only the main steps, reads as:

In[46]:= **int1 = xComputeMatrixEntries[Prepare3DIntegrand[phi[i, j, k, x, y, z], phi[l, m, n, x, y, z], {x, y, z}, {z, z}], x, y, z];**

In[47]:= **int2 = yComputeMatrixEntries[int1, y, z];**

1. Collecting integrands depending on y
   → finished collecting (7.32046 Second)
   → 64 integrands
2. Rewriting integrated Jacobi polynomials in terms of Jacobi polynomials

→ finished rewriting (0.432027 Second)

3. Collecting integrands depending on y
   → finished collecting (0.772048 Second)
   → 58 integrands
4. Adjusting Jacobi polynomials to appearing weights functions
   reducing terms using the Jacobi three term recurrence (1.33608 Second)
   → finished adjusting (3.98025 Second)
6. Evaluate, 70 integrals using Jacobi orthogonality relation
   → finished evaluating (2.19614 Second)

Notation and Symbols

$\mathbb{N} = \{0, 1, 2, \ldots\}$	-	The set of natural numbers
$\mathbb{Z}, \mathbb{Q}, \mathbb{R}$	-	Sets of integers, rational, real numbers
$\mathbb{R}^d$	-	Set of real vectors $\underline{x} = (x_1, x_2, \ldots, x_d)^T$
$\underline{u}$	-	Vector $\underline{u} \in \mathbb{R}^d$
Ω	-	Bounded domain (open and connected subset of $\mathbb{R}^d$)
$\partial\Omega$	-	Boundary of the domain Ω
$K[x]$	-	Ring of polynomials in x with coefficients in K
$K[\![x]\!]$	-	Ring of formal power series
$\deg p$	-	Total degree of a polynomial p
$\deg_x p$	-	Degree of a polynomial p w.r.t. the variable x
$(a)_n$	-	Pochhammer symbol or rising factorial, $(a)_n = a(a+1) \cdot \ldots \cdot (a+n-1)$
$a^{\underline{n}}$	-	Falling factorial, $a^{\underline{n}} = a(a-1) \cdot \ldots \cdot (a-n+1)$
${}_pF_q\left(\begin{matrix} a_1 & \ldots & a_p \\ b_1 & \ldots & b_q \end{matrix}; z\right)$	-	The generalized hypergeometric function, ${}_pF_q\left(\begin{matrix} a_1 & \ldots & a_p \\ b_1 & \ldots & b_q \end{matrix}; z\right) = \sum_{n \geq 0} \frac{(a_1)_n \cdot \ldots \cdot (a_p)_n}{(b_1)_n \cdot \ldots \cdot (b_q)_n} \frac{z^n}{n!}$
$F(a, b, c; z)$	-	The hypergeometric ${}_2F_1$-function, $F(a, b, c; z) = {}_2F_1\left(\begin{matrix} a & b \\ c & \end{matrix}; z\right)$
H_n	-	Harmonic number, $H_n = \sum_{j=1}^{n} \frac{1}{j}$
$P^p(T)$, T simplex	-	Space of polynomial functions of total degree p defined over T
$P^p(Q)$, Q quadrilateral or hexahedron	-	Space of polynmomial functions of maximal degree p defined over Q
$P^{p_1,p_2}(Q)$, Q quadrilateral	-	Space of polynomials p with $deg_{x_i} p \leq p_i$ defined over Q
$P_n^{(\alpha,\beta)}(x), P_n(x), C_n^{\lambda}(x)$	-	The families of Jacobi, Legendre and Gegenbauer polynomials, respectively
$T_n(x), U_n(x)$	-	The families of Chebyshev polynomials of the first and second kind, respectively
$L_n(x)$	-	nth integrated Legendre polynomial
$\hat{p}_n^{\alpha}(x)$	-	nth integrated Jacobi polynomial
$p_n^{\alpha}(x)$	-	Jacobi polynomial $P_n^{(\alpha,0)}(x)$

$w_{\alpha,\beta}(x) = \left(\frac{1-x}{2}\right)^{\alpha}\left(\frac{1+x}{2}\right)^{\beta}$	-	Weight function associated to Jacobi polynomials, $w_{\alpha}(x) = w_{\alpha,0}(x)$
Δ_n	-	The forward difference operator in n
S_n	-	The forward shift operator in n, $S_n f(n) = f(n+1)$
∇	-	Gradient operator, $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_d}\right)$, $x \in \mathbb{R}^d$
div	-	Divergence operator, $\operatorname{div} f(x) = \sum_{i=1}^{d} \frac{\partial f_i(x)}{\partial x_i}$ for a vector valued function f and $x \in \mathbb{R}^d$
Δ	-	Laplace operator, $\Delta f(x) = \sum_{i=1}^{d} \frac{\partial^2 f(x)}{\partial x_i^2}$
$C_0^{\infty}(\Omega)$	-	The space of infinitely differentiable functions with compact support on Ω
$L^p(\Omega)$	-	$L^p(\Omega) = \{f : \Omega \to \mathbb{R} \mid \int_{\Omega} \lvert f(x)\rvert^p \, dx < \infty\}$
$\lVert \cdot \rVert_{L^p(\Omega)}$	-	$\lVert f \rVert_{L^p(\Omega)}^p = \int_{\Omega} \lvert f(x)\rvert^p \, dx$
$(\cdot,\cdot)_0 = (\cdot,\cdot)_{L^2(\Omega)}$	-	$(f,g)_0 = (f,g)_{L^2(\Omega)} = \int_{\Omega} f(x)g(x)\, dx$
$\lVert \cdot \rVert_{L^2(\Omega)}, \lVert \cdot \rVert_0$	-	$\lVert f \rVert_{L^2(\Omega)}^2 = \lVert f \rVert_0^2 = (f,f)_0$
$H^1(\Omega)$	-	$H^1(\Omega) = \{f \in L^2(\Omega) \mid \nabla f \in (L^2(\Omega))^d\}$, $\Omega \subset \mathbb{R}^d$
$\lVert \cdot \rVert_{H^1(\Omega)}, \lVert \cdot \rVert_1$	-	$\lVert f \rVert_{H^1(\Omega)}^2 = \lVert f \rVert_1^2 = (f,f)_0 + (\nabla f, \nabla f)_0$
$(\cdot,\cdot)_{H^1(\Omega)}, (\cdot,\cdot)_1$	-	$(f,g)_{H^1(\Omega)} = (f,g)_1 = (f,g)_0 + (\nabla f, \nabla g)_0$

Bibliography

[1] M. Abramowitz and I. Stegun, editors. *Handbook of mathematical functions.* Dover-Publications, 1965.

[2] M. Ainsworth and L. Demkowicz. Explicit polynomial preserving trace liftings on a triangle. TICAM Report 03-47, 2003.

[3] M. Ainsworth and B. Guo. An additive Schwarz preconditioner for p-version boundary element approximation of the hypersingular operator in three dimensions. *Numer. Math.*, 85(3):343–366, 2000.

[4] M. Ainsworth and J.T. Oden. *A posteriori error estimation in finite element analysis.* Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.

[5] G.E. Andrews, R. Askey, and R. Roy. *Special Functions.* Encyclopedia of Mathematics and its Applications 71. Cambridge University Press, 2000.

[6] D.N. Arnold and F. Brezzi. Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *RAIRO Modél. Math. Anal. Numér.*, 19(1):7–32, 1985.

[7] R. Askey and G. Gasper. Positive Jacobi polynomial sums. II. *Amer. J. Math.*, 98(3):709–737, 1976.

[8] I. Babuška, A.W. Craig, J. Mandel, and J. Pitkäranta. Efficient preconditioning for the p version of the finite element method in $\mathbb{R}^2$. *SIAM Journal of Numerical Analysis*, 28:624–661, 1991.

[9] I. Babuška, M. Griebel, and J. Pitkäranta. The problem of selecting the shape functions for a p-type finite elemnt. *Int. Journ. Num. Meth. Eng.*, 28:1891–1908, 1989.

[10] A. Becirovic, P. Paule, V. Pillwein, A. Riese, C. Schneider, and J. Schöberl. Hypergeometric Summation Algorithms for High Order Finite Elements. *Computing*, 78(3):235–249, 2006.

[11] C. Bennett and R. Sharpley. *Interpolation of operators*, volume 129 of *Pure and Applied Mathematics.* Academic Press Inc., Boston, MA, 1988.

[12] J. Bergh and J. Löfström. *Interpolation Spaces. An Introduction*, volume 223 of *Grundlehren der mathematischen Wissenschaften.* Springer Berlin Heidelberg New York, 1976.

[13] S. Beuchler. A domain decomposition preconditioner for p-FEM discretizations of two-dimensional elliptic problems. *Computing*, 74(4):299–317, 2005.

[14] S. Beuchler. Extension operators on tensor product structures in two and three dimensions. *SIAM J. Sci. Comput.*, 26(5):1776–1795 (electronic), 2005.

[15] S. Beuchler and D. Braess. Improvements for some condition number estimates for preconditioned system in p-FEM. *Numer. Linear Algebra Appl.*, 13(7):573–588, 2006.

[16] S. Beuchler and V. Pillwein. Sparse shape functions for tetrahedral p-FEM using integrated Jacobi polynomials. *Computing*, 80(4):345–375, 2007.

[17] S. Beuchler and V. Pillwein. Completions to sparse shape functions for triangular and tetrahedral p-FEM. In *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 435–442, Heidelberg, 2008. Springer.

[18] S. Beuchler and J. Schöberl. New shape functions for triangular p-fem using integrated Jacobi polynomials. *Numer. Math*, 103:339–366, 2006.

[19] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer Berlin Heidelberg, 3rd edition, 2003.

[20] D. Braess, V. Pillwein, and J. Schöberl. Equilibrated Residual Error Estimates are p-Robust. Technical report, 2008. submitted.

[21] D. Braess and J. Schöberl. Equilibrated residual error estimates for Maxwell's equations. *Math. Comp.* to appear.

[22] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer New York, 2nd edition, 2002.

[23] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Flement Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991.

[24] F. Chyzak. Gröbner bases, symbolic summation and symbolic integration. In *Gröbner bases and applications (Linz, 1998)*, volume 251 of *London Math. Soc. Lecture Note Ser.*, pages 32–60. Cambridge Univ. Press, Cambridge, 1998.

[25] F. Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Math.*, 217(1-3):115–134, 2000. Formal power series and algebraic combinatorics (Vienna, 1997).

[26] F. Chyzak and P. Paule. Computer Algebra. In D. Lozier, editor, *The Digital Library of Mathematical Functions (DLMF)*, pages n/a–n/a. National Institute of Standards and Technology (NIST), Gaithersburg, U.S.A., 2008. In preparation.

[27] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *J. Symbolic Comput.*, 26(2):187–227, 1998.

[28] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North Holland, Amsterdam, 1978.

[29] G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, pages 134–183. Lecture Notes in Comput. Sci., Vol. 33. Springer, Berlin, 1975.

[30] Ch.J. de la Vallée-Poussin. *Leçons sur l'approximation des fonctions d'une variable réelle.* Paris: Gauthier-Villars, 1919.

[31] L. Demkowicz. *Computing with hp Finite Elements I. One- and Two-Dimensional Elliptic and Maxwell Problems.* CRC Press, Taylor and Francis, 2006.

[32] L. Demkowicz, J. Gopalakrishnan, and J. Schöberl. Polynomial extension operators. Part III. In preparation.

[33] L. Demkowicz, J. Gopalakrishnan, and J. Schöberl. Polynomial extension operators. Part I. RICAM-Report 2007-15, Radon Insttiute for Computational and Applied Mathematics, J. Kepler Univ. Linz, 2007.

[34] L. Demkowicz, J. Gopalakrishnan, and J. Schöberl. Polynomial extension operators. Part II. RICAM-Report 2007-16, Radon Insttiute for Computational and Applied Mathematics, J. Kepler Univ. Linz, 2007.

[35] R.A. DeVore and G.G. Lorentz. *Constructive approximation*, volume 303 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 1993.

[36] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

[37] M.C. Fasenmyer. *Some generalized hypergeometric polynomials.* PhD thesis, University of Michigan, 1945.

[38] J. Favard. Sur les polynomes de Tchebicheff. *Comptes Rendus Acad. Sci. Paris*, 200:2052–2053, 1935.

[39] G. Gasper. Positivity and the convolution structure for Jacobi series. *Ann. of Math. (2)*, 93:112–118, 1971.

[40] G. Gasper. Positive sums of the classical orthogonal polynomials. *SIAM J. Math. Anal.*, 8(3):423–447, 1977.

[41] C.F. Gauss. Disquisitiones generales circa seriem infinitam. *Comm. Soc. Reg. Gött. II, Werke*, 3:123–162, 1812.

[42] A. George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973. Collection of articles dedicated to the memory of George E. Forsythe.

[43] S. Gerhold and M. Kauers. A Procedure for Proving Special Function Inequalities Involving a Discrete Parameter. In *Proceedings of ISSAC '05*, pages 156–162. ACM Press, 2005.

[44] S. Gerhold, M. Kauers, and J. Schöberl. On a Conjectured Inequality for a Sum of Legendre Polynomials. Technical Report 2006-11, SFB F013, 2006.

[45] R.W. Gosper. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA*, 75:40–42, 1978.

[46] G. Haase, U. Langer, and A. Meyer. Domain decomposition preconditioners with inexact subdomain solvers. *J. Numer. Linear Algebra Appl.*, 1(1):27–41, 1992.

[47] W. Hahn. Über die Jacobischen Polynome und zwei verwandte Polynomklassen. *Mathematische Zeitschrift*, 39:634–638, 1935.

[48] S. Jensen and V.G. Korneev. On domain decomposition preconditioning in the hierarchical *p*-version of the finite element method. *Comput. Methods Appl. Mech. Eng.*, 150(1–4):27–41, 1991.

[49] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for CFD.* Numerical Mathematics and Scientific Computation. Oxford University Press, New York, Oxford, 1999.

[50] M. Karr. Summation in finite terms. *J. Assoc. Comput. Mach.*, 28(2):305–350, 1981.

[51] M. Karr. Theory of summation in finite terms. *J. Symbolic Comput.*, 1(3):303–315, 1985.

[52] M. Kauers. *Algorithms for Nonlinear Higher Order Difference Equations.* PhD thesis, RISC-Linz, October 2005.

[53] M. Kauers. SumCracker – A Package for Manipulating Symbolic Sums and Related Objects. *Journal of Symbolic Computation*, 41(9):1039–1057, 2006. Available at http://www.risc.uni-linz.ac.at/publications/.

[54] V. Korneev, U. Langer, and L.S. Xanthis. On fast domain decomposition solving procedures for *hp*-discretizations of 3-D elliptic problems. *Comput. Methods Appl. Math.*, 3(4):536–559 (electronic), 2003.

[55] C. Koutschan. *Computer Algebra Algorithms for ∂-Finite and Holonomic Functions.* PhD thesis, 2008. In preparation.

[56] H.L. Krall. On derivatives of orthogonal polynomials. *Bulletin of the AMS*, 42:423–428, 1936.

[57] R.J. Lipton, D.J. Rose, and R.E. Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.*, 16(2):346–358, 1979.

[58] R. Luce and B.I. Wohlmuth. A local a posteriori error estimator based on equilibrated fluxes. *SIAM J. Numer. Anal.*, 42(4):1394–1414 (electronic), 2004.

[59] C. Mallinger. Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences. Master's thesis, RISC, J. Kepler University, August 1996. Available at http://www.risc.uni-linz.ac.at/publications/.

[60] J.M. Melenk. *hp-Finite Element Methods for Singular Perturbations*, volume 1796 of *Lecture notes in mathematics.* Springer Berlin Heidelberg, 2002.

[61] J.M. Melenk, K. Gerdes, and C. Schwab. Fully discrete *hp*-finite elements: Fast quadrature. *Comput. Methods Appl. Mech. Engrg.*, 190:4339–4364, 2001.

[62] R. Muñoz-Sola. Polynomial liftings on a tetrahedron and applications to the *hp*-version of the finite element method in three dimensions. *SIAM Journal of Numerical Analysis*, 34(1):282–314, 1997.

[63] S.A. Orszag. Spectral methods for problems in complex geometries. *J. Comp. Phys.*, 37(1):70–92, 1980.

[64] P. Paule. Contiguous relations and creative telescoping. Preprint.

[65] P. Paule and M. Schorn. A Mathematica Version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities. *Journal of Symbolic Computation*, 20:673–698, 1995.

[66] M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symbolic Comput.*, 14(2-3):243–264, 1992.

[67] V. Pillwein. Positivity of certain sums over Jacobi kernel polynomials. *Adv. in Appl. Math.*, 2008. doi:10.1016/j.aam.2007.12.001.

[68] W. Prager and J.L. Synge. Approximations in elasticity based on the concept of function space. *Quart. Appl. Math.*, 5:241–269, 1947.

[69] E.D. Rainville. *Special Functions*. Chelsea Publishing Co., Bronx, N.Y., first edition, 1971.

[70] B. Salvy and P. Zimmermann. Gfun: A package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20:163–177, 1994.

[71] O. Schenk and K. Gärtner. Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO. *Journal of Future Generation Computer Systems*, 20(3):475–487, 2004.

[72] C. Schneider. A new Sigma approach to multi-summation. *Adv. in Appl. Math.*, 34(4):740–767, 2005.

[73] J. Schöberl. Netgen - an advancing font 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1:41–52, 1997.

[74] J. Schöberl. A posteriori error estimates for Maxwell Equations. *Math. Comp*, 77(262):633–649, 2008.

[75] J. Schöberl, J.M. Melenk, C. Pechstein, and S. Zaglmayr. Additive Schwarz preconditioning for *p*-version triangular and tetrahedral finite elements. *IMA Journal of Numerical Analysis.*

[76] Ch. Schwab. *p- and hp-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 1998.

[77] J.S. Sherwin and E.M. Karniadakis. A new triangular and tetrahedral basis for high-order (*hp*) finite element methods. *Internat. J. Numer. Methods Engrg.*, 38(22):3775–3802, 1995.

[78] B.A. Szabó and I. Babuška. *Finite Element Analysis.* John Wiley & Sons, 1991.

[79] G. Szegö. *Orthogonal Polynomials.* AMS Colloquium Publications, Volume XXIII. 3 edition, 1974.

[80] N. Takayama. An Algorithm of Constructing the Integral of a Module - an Infinite Dimensional Analog of Gröbner Basis. In *ISSAC*, pages 206–211, 1990.

[81] N. Takayama. Gröbner Basis, Integration and Transcendental Functions. In *ISSAC*, pages 152–156, 1990.

[82] T. C. Warburton, S. J. Sherwin, and G. E. Karniadakis. Basis functions for triangular and quadrilateral high-order elements. *SIAM J. Sci. Comput.*, 20(5):1671–1695 (electronic), 1999.

[83] K. Wegschaider. Computer Generated Proofs of Binomial Multi-Sum Identities. Master's thesis, RISC, J. Kepler University, May 1997.

[84] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske. The kernel polynomial method. *Rev. Modern Phys.*, 78(1):275–306, 2006.

[85] H.S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Invent. Math.*, 108(3):575–633, 1992.

[86] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete. Math.*, 80:207–211, 1990.

[87] D. Zeilberger. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math.*, 32(3):321–368, 1990.

[88] D. Zeilberger. The method of creative telescoping. *J. Symbolic Computation*, 11:195–204, 1991.

Index

AdjustToWeight *(IntJac command)*, 62

Bessel's inequality, 24, 89

Cauchy product, 20, 21, 27, 48
certificate recurrence, 29
Chebyshev polynomial, 36, 76–79, 86
Christoffel-Darboux formula, 25, 31, 91
ComputeMatrixEntries *(IntJac command)*, 107, 108
condition number, 15, 41, 46, 52, 54, 66–67
contiguous, 22
contiguous relation, 22, 23, 34, 39
Crack *(SumCracker command)*, 30, 38, 40, 48, 49

definite sum, 31
difference operator, 21
Dirichlet boundary condition, 16, 106
divergence, 6, 87, 96, 97, 99–101
Duffy transformation, 52, 58, 59, 68, 107

falling factorial, 62
Fasenmyer's algorithm, 29
Fejér kernel, 76
Fem1D, 17, 103
Fem2D, 103–107
FindRecurrence *(MultiSum command)*, 29, 34, 39
FindStructureSet *(MultiSum command)*, 39
finite element basis functions, 11–14
finite element method, 9–11
formal power series, 19, 20, 23
FreeOf *(MultiSum option)*, 29

Gegenbauer polynomial, 35
generating function, 20, 23
GeneratingFunctions, 27, 44, 49
  GuessRE *(command)*, 27, 49, 77, 84, 85
  RECauchy *(command)*, 27, 48
  REHadamard *(command)*, 27
  REPlus *(command)*, 27, 44, 45, 49, 50
Gosper's algorithm, 28
GuessRE *(GeneratingFunctions command)*, 27, 49, 77, 84, 85

h-method, 11

Hadamard product, 21
harmonic number, 91
Head *(SumCracker option)*, 30, 31
holonomic, 20
holonomic power series, 20
holonomic sequence, 20
hp-method, 11
hypergeometric, 21
hypergeometric ${}_pF_q$ function, 23
hypergeometric function, 21, 75

In *(SumCracker option)*, 30, 31
integrated Jacobi polynomial, 37–39, 45–48, 53–59, 61, 63, 65, 68–70, 107
integrated Legendre polynomial, 12, 13, 37, 38, 42, 44, 45, 53, 62, 94, 103, 105
integrated Legendre polynomials, 52
interpolation space, 7, 42
IntJac, 57, 64
  AdjustToWeight *(command)*, 62
  ComputeMatrixEntries *(command)*, 107, 108
  Prepare2DIntegrand *(command)*, 59, 107
  Prepare3DIntegrand *(command)*, 59, 64, 109
  RemovePHAT *(command)*, 61–63
Into *(SumCracker option)*, 49

Jacobi polynomial, 23, 24, 30, 33–39, 44, 45, 47–50, 52, 57, 63, 73–83, 86

kernel polynomial, 25, 73–75, 88, 89

Legendre differential equation, 83
Legendre polynomial, 12, 23, 27, 29–31, 35–37, 40, 48, 49, 52, 62, 73, 75, 83–85, 88, 89, 91, 93
LinearRecurrence *(SumCracker command)*, 30, 31, 37, 38, 49

mass matrix, 9, 68, 103
MultiSum, 29, 34, 70
  FindRecurrence *(command)*, 29, 34, 39
  FindStructureSet *(command)*, 39
  FreeOf *(option)*, 29
  SumCertificate *(command)*, 29, 30, 34, 39

Neumann boundary condition, 106

orthogonal polynomial, 23–26

p-method, 11
Parseval's formula, 24
Pochhammer symbol, 21, 47
polynomial
- Chebyshev-, 36, 76–78, 86
- Gegenbauer-, 35
- integrated Legendre-, 38
- integrated Jacobi-, 37–39, 45–48, 53–59, 61, 63, 65, 68–70, 107
- integrated Legendre-, 12, 13, 37, 38, 42, 44, 45, 52, 54, 62, 94, 103, 105
- Jacobi-, 23, 24, 33–39, 44, 45, 47–50, 52, 57, 73–78, 86
- Legendre-, 12, 23, 27, 29–31, 35–37, 40, 44, 48, 49, 52, 62, 73, 75, 83–85, 88, 89, 91, 93
- ultraspherical-, 35, 36, 73–86

preconditioner, 15, 41, 66–67
Prepare2DIntegrand *(IntJac command)*, 59, 107
Prepare3DIntegrand *(IntJac command)*, 59, 64, 109
proper hypergeometric, 21
ProveInequality *(SumCracker command)*, 30, 31, 80, 82, 83

Raviart-Thomas, 97, 98
RECauchy *(GeneratingFunctions command)*, 27, 48
REHadamard *(GeneratingFunctions command)*, 27
RemovePHAT *(IntJac command)*, 61–63
REPlus *(GeneratingFunctions command)*, 27, 44, 45, 49, 50
reproducing property, 25, 74, 89
rising factorial, 21
Rodrigues formula, 34

Schur complement, 66, 67
shape function, 12
shift operator, 21
Sigma, 41, 50
Sobolev space, 6
stiffness matrix, 9, 57, 68, 103
SUM *(SumCracker expression)*, 30, 31
SumCertificate *(MultiSum command)*, 29, 30, 34, 39
SumCracker, 30, 31, 44, 76, 77, 81, 91
- Crack *(command)*, 30, 38, 40, 48, 49
- Head *(option)*, 30, 31
- In *(option)*, 30, 31
- Into *(option)*, 49
- LinearRecurrence *(command)*, 30, 31, 37, 38, 49
- ProveInequality *(command)*, 30, 31, 80, 82, 83
- SUM *(expression)*, 30, 31
- Where *(option)*, 38
- ZeroSequenceQ *(command)*, 30, 77, 81, 85

trace, 7
trace operator, 7
trace space, 42
triangulation, 10
- quasi-uniform-, 10, 95

ultraspherical polynomial, 35, 36, 73–86

variational formulation, 8, 9, 16, 68, 95, 103, 104, 106

weak gradient, 6
weakly differentiable, 6
weight function, 23, 24, 57–59, 61, 63, 75, 76
Where *(SumCracker option)*, 38

Zeilberger's algorithm, 28, 35, 37, 40, 44, 48
ZeroSequenceQ *(SumCracker command)*, 30, 77, 81, 85

Eidesstattliche Erklärung

Ich erkläre an Eides statt, daß ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, March, 2008

. . . .

Veronika Pillwein

Curriculum Vitae

Personal data

Name	Veronika Pillwein
Date and place of birth	March 19, 1975, Vienna, Austria
Nationality	Austria

Affiliation

SpezialForschungsBereich SFB F013
Numerical and Symbolic Scientific Computing

Johannes Kepler Universität Linz
Altenberger Straße 69
4040 Linz
AUSTRIA

www.sfb013.uni-linz.ac.at
veronika.pillwein@sfb013.uni-linz.ac.at

Education

06/1993	Matura (high school graduation) at BG Gmunden
10/1993–01/2004	Studying Technical Mathematics at the Johannes Kepler University, Linz *Diploma thesis:* "Interpolation von Sobolevräumen" *Thesis advisor:* Prof. Paul Müller
10/2004–04/2008	Doctorate studies at the Spezialforschungsbereich SFB013 "Numerical and Symbolic Scientific Computing", Johannes Kepler University, Linz